Labo 10 - Docker compose

Docker compose est un chef de cuisine qui permet de créer une recette en y mettant dedans tous le conteneurs nécessaires à la fabrication du plat final

Nous allons monter un serveur Wordpress (serveur Web + base de données)

Tout d'abord il faut créer le fichier docker-compose.yml

Pour plus d'infos sur le format .yaml, il suffit d'aller sur le sur leur site https://yaml.org/

La première ligne correspond à la dernière version de docker compose qui est rétro-compatible avec les anciennes versions

```
sudo vi docker-compose.yml
```

On peut retrouver plusieurs sections dans ce fichier :

version: "3.8" -> obligatoire (version actuelle de docker-compose)
services: -> obligatoire (les conteneurs dont nous aurons besoin)
volumes: -> dépend de vos besoins
networks: -> dépend de vos besoins

Voici donc le texte à copier-coller dans votre fichier docker-compose.yml

> Attention : pensez à respecter les indentations !!!!

```
version: "3.8"
services: #nom du conteneur
  wordpress:
    image: wordpress:4.9 #image à téléchager sur le docker hub avec le tag
    ports:
      - 80:80 #ports à ouvrir entre l'hôte et le conteneur
    environment: #variables d'environnements nécessaires pour la création du conteneur prises
sur le docker hub www.hub.docker.com
      - WORDPRESS_DB_HOST=db #nom du conteneur de la base de données
      - WORDPRESS_DB_USER=tata #utilisateur qui aura été créé plus bas
      - WORDPRESS_DB_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
      - WORDPRESS_DB_NAME=wordp #nom de la basse de données
    networks:
```

```
      - galaxie #nom du réseau auquel sera connecté le conteneur
  db:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=tata_yoyo #mot de passe root de la base de données
      - MYSQL_DATABASE=wordp #nom de la base de données
      - MYSQL_USER=tata #utilisateur de la base de données
      - MYSQL_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
    networks:
      - galaxie
#volumes: #à ce stade la section volume sera utilisée plus tard
networks:
  galaxie: #création réseau bridgé cloné
```

nancre@DOCKER:~

```
version: "3.8"
services: #nom du conteneur
  wordpress:
    image: wordpress:4.9 #image à téléchager sur le docker hub avec le tag
    ports:
      - 80:80 #ports à ouvrir entre l'hôte et le coneneur
    environment: #variables d'environnements nécessaires pour la création du conteneur prises sur le docker hub www.hub.docker.com
      - WORDPRESS_DB_HOST=db #nom du conteneur de la base de données
      - WORDPRESS_DB_USER=tata #utilisateur qui aura été créé plus bas
      - WORDPRESS_DB_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
      - WORDPRESS_DB_NAME=wordp #nom de la basse de données
    networks:
      - galaxie #nom du réseau auquel sera connecté le conteneur
  db:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=tata_yoyo #mot de passe root de la base de données
      - MYSQL_DATABASE=wordp #nom de la base de données
      - MYSQL_USER=tata #utilisateur de la base de données
      - MYSQL_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
    networks:
      - galaxie
#volumes: #à ce stade la section volume sera utilisée plus tard
networks:
  galaxie: #création réseau bridgé cloné
~
~
```

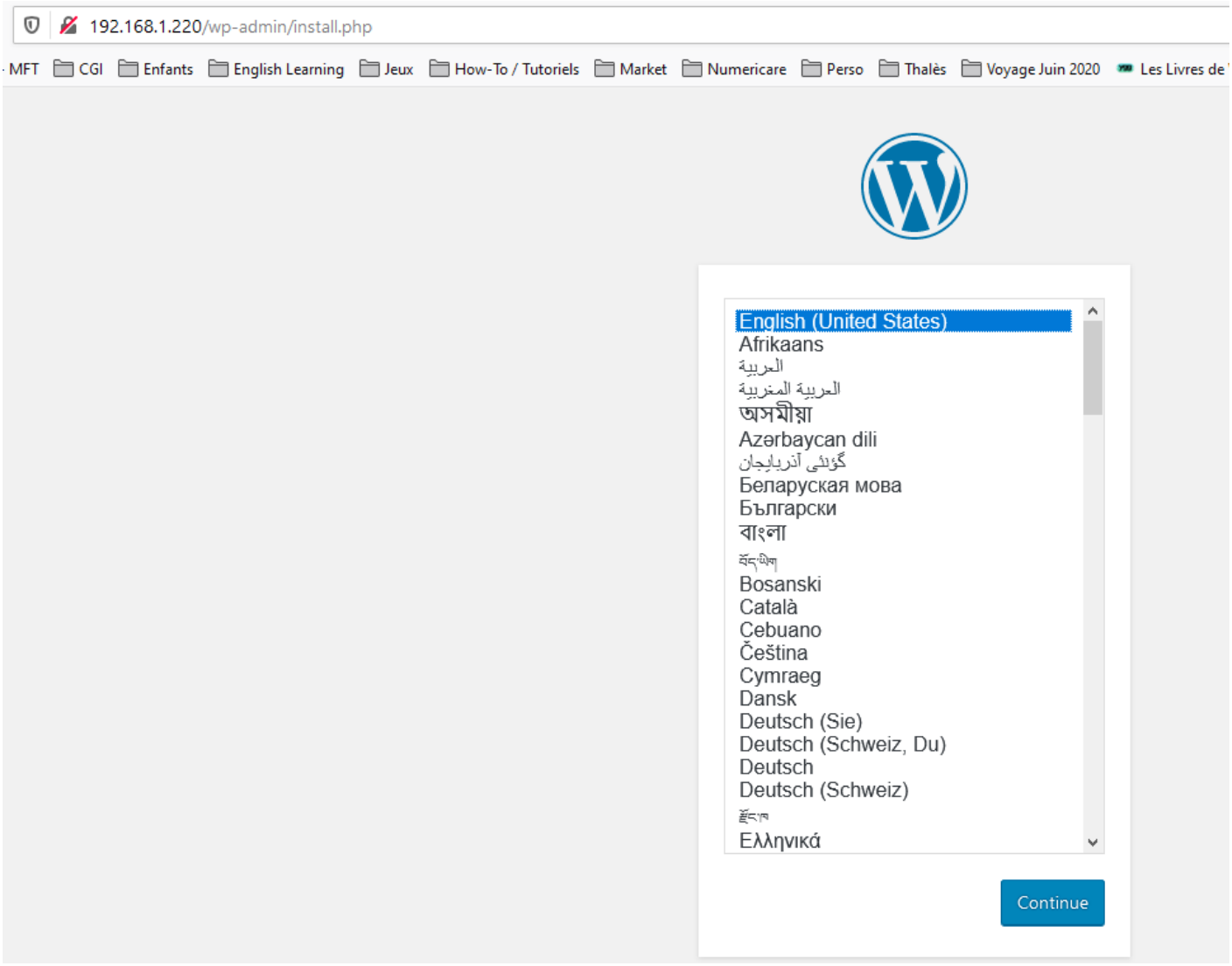
Pour lancer la création de mon environnement

```
docker-compose up
```

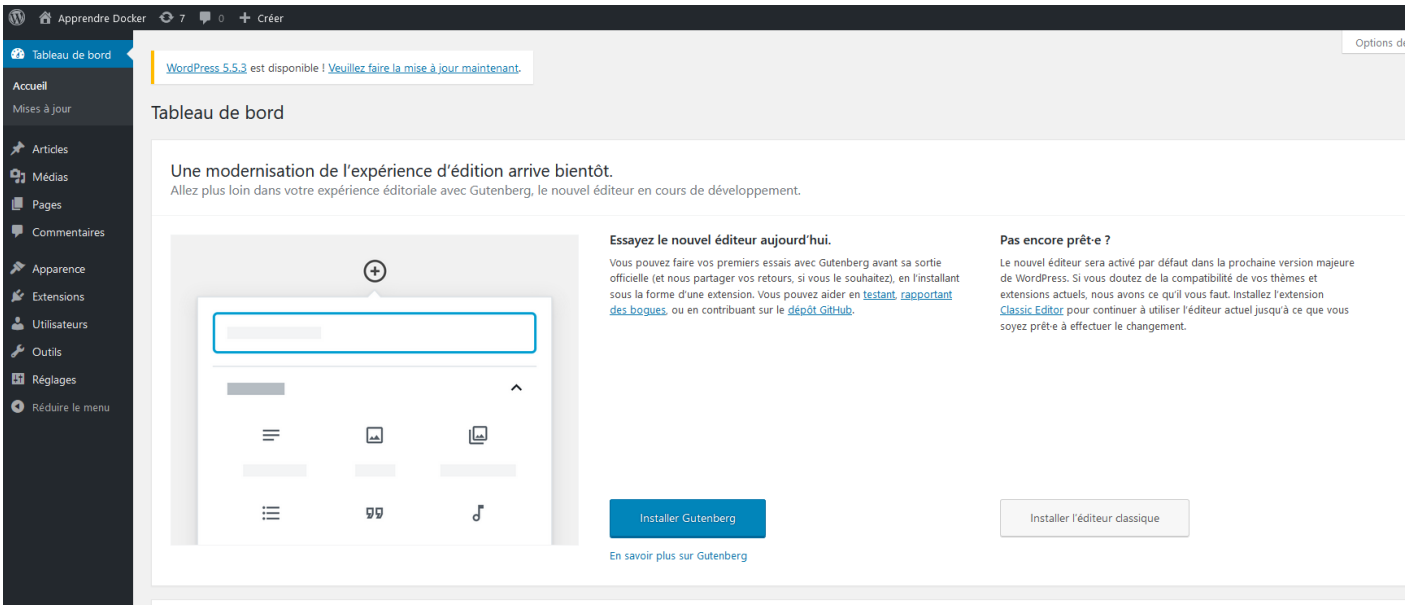
nancre@DOCKER:~

```
[nancre@DOCKER ~]$ docker-compose up
Creating network "nancre_galaxie" with the default driver
Pulling wordpress (wordpress:4.9)...
4.9: Pulling from library/wordpress
a5a6f2f73cd8: Pull complete
633e0d1cd2a3: Pull complete
fcdfdf7118ba: Pull complete
4e7dc76b1769: Pull complete
c425447c8835: Pull complete
75780b7b9977: Pull complete
33ed51bc30e8: Pull complete
7c4215700bc4: Pull complete
ef55a760eb7a: Pull complete
d982e3946ac5: Pull complete
a38e2fdf4f50: Pull complete
09f702917a0a: Pull complete
df1d46358537: Pull complete
9b0d5695ec42: Pull complete
7abe1c1f0479: Pull complete
db1df7737fbd: Pull complete
da5248206256: Pull complete
e850a08a7c7e: Pull complete
e5e7ecd1752b: Pull complete
Digest: sha256:7e476394586459bb622d3f37448cd07e703ec6906257d232542f2f51ff073da7
Status: Downloaded newer image for wordpress:4.9
Pulling db (mysql:5.7)...
5.7: Pulling from library/mysql
bb79b6b2107f: Already exists
49e22f6fb9f7: Pull complete
842b1255668c: Pull complete
9f48d1f43000: Pull complete
c693f0615bce: Pull complete
8a621b9dbed2: Pull complete
0807d32aef13: Pull complete
f15d42f48bd9: Pull complete
098ceecc0c8d: Pull complete
b6fead9737bc: Pull complete
351d223d3d76: Pull complete
Digest: sha256:4d2b34e99c14edb99cdd95ddad4d9aa7ea3f2c4405ff0c3509a29dc40bcb10ef
Status: Downloaded newer image for mysql:5.7
Creating nancre_wordpress_1 ... done
Creating nancre_db_1        ... done
Attaching to nancre_db_1, nancre_wordpress_1
db_1         | 2020-11-17 13:09:20+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Ser
wordpress_1  | WordPress not found in /var/www/html - copying now...
db_1         | 2020-11-17 13:09:20+00:00 [Note] [Entrypoint]: Switching to dedicated user 'my
db_1         | 2020-11-17 13:09:20+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Ser
db_1         | 2020-11-17 13:09:20+00:00 [Note] [Entrypoint]: Initializing database files
db_1         | 2020-11-17T13:09:20.425438Z 0 [Warning] TIMESTAMP with implicit DEFAULT value
wordpress_1  | Complete! WordPress has been successfully copied to /var/www/html
db_1         | 2020-11-17T13:09:20.616817Z 0 [Warning] InnoDB: New log files created, LSN=457
db_1         | 2020-11-17T13:09:20.665372Z 0 [Warning] InnoDB: Creating foreign key constrain
db_1         | 2020-11-17T13:09:20.724620Z 0 [Warning] No existing UUID has been found, so we
db_1         | 2020-11-17T13:09:20.725896Z 0 [Warning] Gtid table is not ready to be used. Ta
wordpress_1  |
wordpress_1  | MySQL Connection Error: (2002) Connection refused
wordpress_1  |
wordpress_1  | Warning: mysqli::__construct(): (HY000/2002): Connection refused in Standard i
```

S'il n'y a pas d'erreur, rendez-vous sur votre navigateur web en tapant l'ip de votre host



En suivant les étapes d'installation, on a bien un wordpress fonctionnel



Si nous quittons l'environnement, les conteneurs vont s'arrêter. Pour que les conteneurs soit lancés en tâche de fond

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

```
wordpress_1  | 192.168.1.202 - - [17/Nov/2020:13:18:10 +0000] "POST /wp
^CGracefully stopping... (press Ctrl+C again to force)
Stopping nancre_db_1        ... done
Stopping nancre_wordpress_1 ... done
[nancre@DOCKER ~]$ sudo docker-compose up -d
[sudo] password for nancre:
Sorry, try again.
[sudo] password for nancre:
sudo: docker-compose: command not found
[nancre@DOCKER ~]$ docker-compose up -d
Starting nancre_db_1        ... done
Starting nancre_wordpress_1 ... done
[nancre@DOCKER ~]$
```

Pour arrêter les conteneurs sans les supprimer

```
docker-compose stop
```

Pour les démarrer

```
docker compose start
```

Pour supprimer tout un environnement

```
docker-compose down
```

nancre@DOCKER:~

```
[nancre@DOCKER ~]$ docker-compose stop
Stopping nancre_db_1        ... done
Stopping nancre_wordpress_1 ... done
[nancre@DOCKER ~]$ docker-compose start
Starting wordpress ... done
Starting db        ... done
[nancre@DOCKER ~]$ docker-compose down
Stopping nancre_db_1        ... done
Stopping nancre_wordpress_1 ... done
Removing nancre_db_1        ... done
Removing nancre_wordpress_1 ... done
Removing network nancre_galaxie
[nancre@DOCKER ~]$
```

> Cette commande supprime tout sauf les volumes

Voyons justement comment personnaliser wordpress avec les volumes
Le chemin de base de wordpress se trouve dans /var/www/html
Et celui de mysql dans /var/lib/mysql

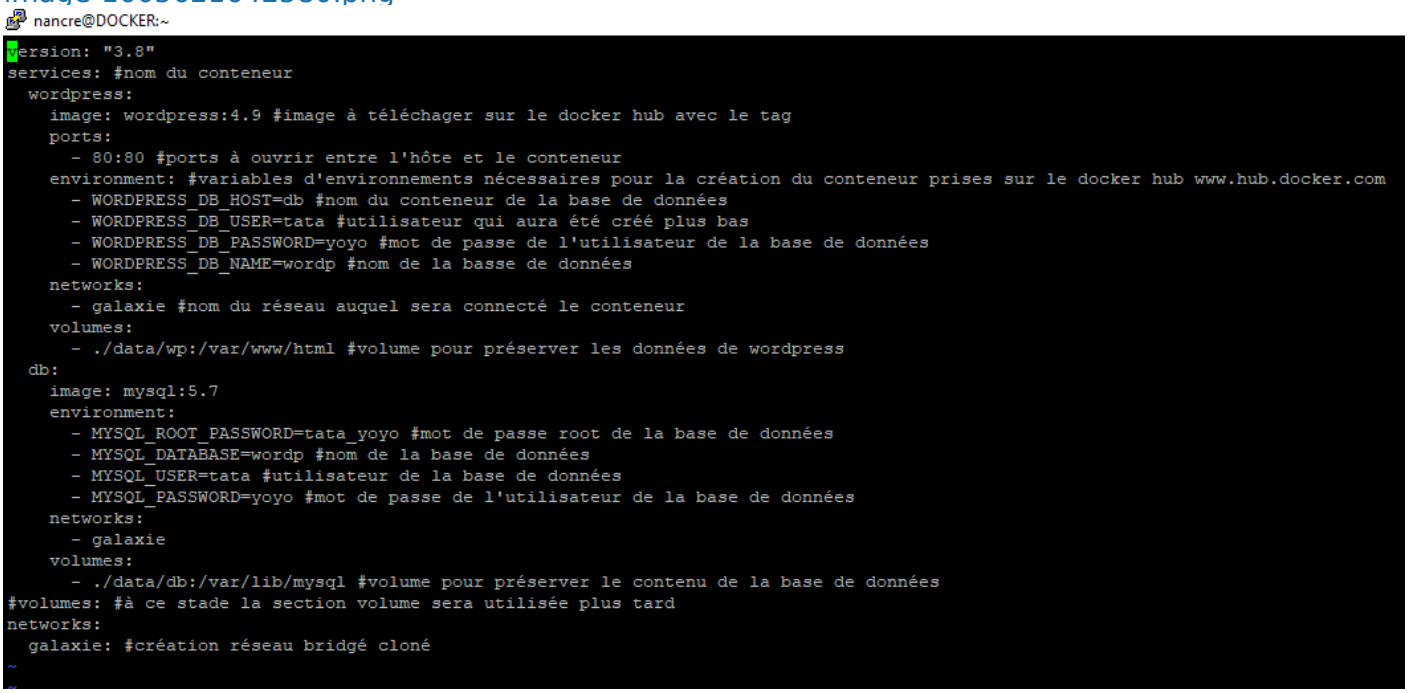
Nous allons commencer par la création de volumes mappés

Editons le fichier docker-compose.yml

```
sudo vi docker-compose.yml
```

```
version: "3.8"
services: #nom du conteneur
  wordpress:
    image: wordpress:4.9 #image à téléchager sur le docker hub avec le tag
    ports:
      - 80:80 #ports à ouvrir entre l'hôte et le conteneur
    environment: #variables d'environnements nécessaires pour la création du conteneur prises
sur le docker hub www.hub.docker.com
      - WORDPRESS_DB_HOST=db #nom du conteneur de la base de données
      - WORDPRESS_DB_USER=tata #utilisateur qui aura été créé plus bas
      - WORDPRESS_DB_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
      - WORDPRESS_DB_NAME=wordp #nom de la basse de données
    networks:
      - galaxie #nom du réseau auquel sera connecté le conteneur
    volumes:
      - ./data/wp:/var/www/html #volume pour préserver les données de wordpress
 db:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=tata_yoyo #mot de passe root de la base de données
      - MYSQL_DATABASE=wordp #nom de la base de données
      - MYSQL_USER=tata #utilisateur de la base de données
      - MYSQL_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
    networks:
      - galaxie
    volumes:
      - ./data/db:/var/lib/mysql
#volumes: #à ce stade la section volume sera utilisée plus tard
networks:
  galaxie: #création réseau bridgé cloné
```

image-1605621642580.png



Tapez la commande pour recréer l'environnement et la nouvelle installation de wordpress

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

Donc en tapant la commande docker-compose down, les conteneurs seront arrêtés et supprimés mais les données conservées dans le volume spécifié (ici le /data)

Voyons maintenant le comportement avec les volumes managés

Editons de nouveau le fichier docker-compose.yml

```
version: "3.8"
services: #nom du conteneur
  wordpress:
    image: wordpress:4.9 #image à téléchager sur le docker hub avec le tag
    ports:
      - 80:80 #ports à ouvrir entre l'hôte et le conteneur
    environment: #variables d'environnements nécessaires pour la création du conteneur prises
sur le docker hub www.hub.docker.com
      - WORDPRESS_DB_HOST=db #nom du conteneur de la base de données
      - WORDPRESS_DB_USER=tata #utilisateur qui aura été créé plus bas
      - WORDPRESS_DB_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
      - WORDPRESS_DB_NAME=wordp #nom de la basse de données
    networks:
      - galaxie #nom du réseau auquel sera connecté le conteneur
    volumes:
```

```
      - wp:/var/www/html #volume pour préserver les données de wordpress
  db:
     image: mysql:5.7
     environment:
       - MYSQL_ROOT_PASSWORD=tata_yoyo #mot de passe root de la base de données
       - MYSQL_DATABASE=wordp #nom de la base de données
       - MYSQL_USER=tata #utilisateur de la base de données
       - MYSQL_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
     networks:
       - galaxie
     volumes:
       - db:/var/lib/mysql
 volumes: #ici ce sont des volumes managés que vous modifierez plus haut
   wp:
   db:
 networks:
   galaxie: #création réseau bridgé cloné
```

nancre@DOCKER:~

```
version: "3.8"
services: #nom du conteneur
  wordpress:
    image: wordpress:4.9 #image à téléchager sur le docker hub avec le tag
    ports:
      - 80:80 #ports à ouvrir entre l'hôte et le conteneur
    environment: #variables d'environnements nécessaires pour la création du conteneur prises sur le docker hub www.hub.docker.com
      - WORDPRESS_DB_HOST=db #nom du conteneur de la base de données
      - WORDPRESS_DB_USER=tata #utilisateur qui aura été créé plus bas
      - WORDPRESS_DB_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
      - WORDPRESS_DB_NAME=wordp #nom de la basse de données
    networks:
      - galaxie #nom du réseau auquel sera connecté le conteneur
    volumes:
      - wp:/var/www/html #volume pour préserver les données de wordpress
  db:
    image: mysql:5.7
    environment:
      - MYSQL_ROOT_PASSWORD=tata_yoyo #mot de passe root de la base de données
      - MYSQL_DATABASE=wordp #nom de la base de données
      - MYSQL_USER=tata #utilisateur de la base de données
      - MYSQL_PASSWORD=yoyo #mot de passe de l'utilisateur de la base de données
    networks:
      - galaxie
    volumes:
      - db:/var/lib/mysql #volume pour préserver le contenu de la base de données
volumes: #ici ce sont des volumes managés que vous modifierez plus haut
  wp:
  db:
networks:
  galaxie: #création réseau bridgé cloné
~
```

Et nous relançons la commande suivant pour recréer les conteneurs avec les volumes managés

```
docker-compose up
```

```
db_1          | 2020-11-17T14:21:55.356400Z 0 [Note] InnoDB: Using CPU crc32 instructions
db_1          | 2020-11-17T14:21:55.357995Z 0 [Note] InnoDB: Initializing buffer pool, total size = 1
db_1          | 2020-11-17T14:21:55.364549Z 0 [Note] InnoDB: Completed initialization of buffer pool
db_1          | 2020-11-17T14:21:55.366493Z 0 [Note] InnoDB: If the mysqld execution user is authoriz
db_1          | 2020-11-17T14:21:55.377845Z 0 [Note] InnoDB: Highest supported file format is Barracu
db_1          | 2020-11-17T14:21:55.385838Z 0 [Note] InnoDB: Creating shared tablespace for temporary
db_1          | 2020-11-17T14:21:55.385938Z 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. P
db_1          | 2020-11-17T14:21:55.398400Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
db_1          | 2020-11-17T14:21:55.399007Z 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 re
db_1          | 2020-11-17T14:21:55.399022Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are acti
db_1          | 2020-11-17T14:21:55.399276Z 0 [Note] InnoDB: Waiting for purge to start
db_1          | 2020-11-17T14:21:55.449422Z 0 [Note] InnoDB: 5.7.32 started; log sequence number 1260
db_1          | 2020-11-17T14:21:55.449675Z 0 [Note] Plugin 'FEDERATED' is disabled.
db_1          | 2020-11-17T14:21:55.449929Z 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mys
db_1          | 2020-11-17T14:21:55.452654Z 0 [Note] InnoDB: Buffer pool(s) load completed at 201117
db_1          | 2020-11-17T14:21:55.456275Z 0 [Note] Found ca.pem, server-cert.pem and server-key.pem
db_1          | 2020-11-17T14:21:55.456293Z 0 [Note] Skipping generation of SSL certificates as certi
db_1          | 2020-11-17T14:21:55.456817Z 0 [Warning] CA certificate ca.pem is self signed.
db_1          | 2020-11-17T14:21:55.456869Z 0 [Note] Skipping generation of RSA key pair as key files
db_1          | 2020-11-17T14:21:55.457497Z 0 [Note] Server hostname (bind-address): '*'; port: 3306
db_1          | 2020-11-17T14:21:55.457560Z 0 [Note] IPv6 is available.
db_1          | 2020-11-17T14:21:55.457573Z 0 [Note]   - '::' resolves to '::';
db_1          | 2020-11-17T14:21:55.457590Z 0 [Note] Server socket created on IP: '::'.
db_1          | 2020-11-17T14:21:55.458260Z 0 [Warning] Insecure configuration for --pid-file: Locati
db_1          | 2020-11-17T14:21:55.465604Z 0 [Note] Event Scheduler: Loaded 0 events
db_1          | 2020-11-17T14:21:55.465757Z 0 [Note] mysqld: ready for connections.
db_1          | Version: '5.7.32'  socket: '/var/run/mysqld/mysqld.sock'  port: 3306  MySQL Community
wordpress_1   | AH00558: apache2: Could not reliably determine the server's fully qualified domain na
wordpress_1   | AH00558: apache2: Could not reliably determine the server's fully qualified domain na
wordpress_1   | [Tue Nov 17 14:21:57.741367 2020] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25
wordpress_1   | [Tue Nov 17 14:21:57.741433 2020] [core:notice] [pid 1] AH00094: Command line: 'apach
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:08 +0000] "GET / HTTP/1.1" 302 378 "-" "Mozilla/
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:08 +0000] "GET /wp-admin/install.php HTTP/1.1" 2
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:09 +0000] "GET /wp-includes/css/buttons.min.css?
01 Firefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:13 +0000] "POST /wp-admin/install.php?step=1 HTT
.0"
wordpress_1   | sh: 1: -t: not found
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:35 +0000] "POST /wp-admin/install.php?step=2 HTT
efox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:38 +0000] "GET /wp-login.php HTTP/1.1" 200 1559
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:38 +0000] "GET /wp-admin/load-styles.php?c=0&dir
s NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:39 +0000] "POST /wp-login.php HTTP/1.1" 302 1106
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:40 +0000] "GET /wp-admin/ HTTP/1.1" 200 17505 "h
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:44 +0000] "GET /wp-includes/js/wp-emoji-release.
irefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:44 +0000] "GET /wp-admin/admin-ajax.php?action=w
NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:44 +0000] "GET /wp-admin/admin-ajax.php?action=w
NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:44 +0000] "GET /wp-admin/admin-ajax.php?action=d
0.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0"
wordpress_1   | 192.168.1.202 - - [17/Nov/2020:14:23:44 +0000] "POST /wp-admin/admin-ajax.php HTTP/1.
```

Pas de surprise tout fonctionne et on repart sur une nouvelle installation de wordpress
Arrêtons le tout et voyons si les volumes ont bien été créés

```
sudo docker volume ls
```

nancre@DOCKER:~

```
[nancre@DOCKER ~]$ sudo docker volume ls
[sudo] password for nancre:
DRIVER              VOLUME NAME
local               94279462079f15b8ccbaa41985aa984ec0f487ef2278a5a7905e7e06208e8ed8
local               a508alad7d87deb041d93c223f53afc4ee4258a8b2848f3083df8a88d06e696e
local               mes_donnees
local               mes_donnees1
local               nancre_db
local               nancre_wp
[nancre@DOCKER ~]$
```

En faisant un docker-compose down, les conteneurs seront supprimés mais pas les volumes

nancre@DOCKER:~

```
[nancre@DOCKER ~]$ docker-compose down
Removing nancre_wordpress_1 ... done
Removing nancre_db_1        ... done
Removing network nancre_galaxie
[nancre@DOCKER ~]$ sudo docker volume ls
DRIVER              VOLUME NAME
local               94279462079f15b8ccbaa41985aa984ec0f487ef2278a5a7905e7e06208e8ed8
local               a508alad7d87deb041d93c223f53afc4ee4258a8b2848f3083df8a88d06e696e
local               mes_donnees
local               mes_donnees1
local               nancre_db
local               nancre_wp
[nancre@DOCKER ~]$
```

Cependant il est possible de forcer la suppression de ces volumes

```
docker-compose down -v
```

nancre@DOCKER:~

```
[nancre@DOCKER ~]$ docker-compose down -v
Removing network nancre_galaxie
WARNING: Network nancre_galaxie not found.
Removing volume nancre_wp
Removing volume nancre_db
[nancre@DOCKER ~]$ sudo docker volume ls
DRIVER              VOLUME NAME
local               94279462079f15b8ccbaa41985aa984ec0f487ef2278a5a7905e7e06208e8ed8
local               a508alad7d87deb041d93c223f53afc4ee4258a8b2848f3083df8a88d06e696e
local               mes_donnees
local               mes_donnees1
[nancre@DOCKER ~]$
```

> Cela supprime les volumes managés créés à partir du docker-compose.yml mais pas les autres volumes managés créés avec docker

Pour finir, il est possible de modifier un paramètre dans le fichier docker-compose.yml en cas d'arrêt ou redémarrage de docker.

Il s'agit de restart:

restart: always #permet de redémarrer automatiquement le conteneur quel que soit l'état dans lequel il était avant l'arrêt ou le redémarrage de docker
restart: no #empêche de redémarrer automatiquement le conteneur quel que soit l'état dans lequel il était avant l'arrêt ou le redémarrage de docker
restart: unless-stopped #permet de redémarrer automatiquement le conteneur s'il était démarré avant l'arrêt ou le redémarrage de docker

---

Revision #1
Created 2026-04-24 13:18:55 UTC by Nico là
Updated 2026-04-24 13:19:11 UTC by Nico là