

Docker compose

Etape 1 : Création sans docker-compose

Par exemple, pour créer un conteneur nginx, avec un volume spécifique et un réseau spécifique, nous devrions faire :

1. Créer un volume

```
docker volume create --name vol_nginx
```

```
vagrant@mydocker:~$ docker volume create --name vol_nginx
vol_nginx
```

2. Affichez les volumes

```
docker volume ls
```

```
vagrant@mydocker:~$ docker volume ls
DRIVER          VOLUME NAME
local           vol_nginx
```

3. Créez un réseau

```
docker network create net_nginx
```

```
vagrant@mydocker:~$ docker network create net_nginx
0378cec87ced67268bc93a774b7f305992eb3f3c246de96e5defde6e9c429b
```

4. Affichez les réseaux

```
docker network ls
```

```
vagrant@mydocker:~$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
35437d685315        bridge              bridge              local
eca043e15001        host                host                local
0378cec87ced        net_nginx           bridge              local
1ddb38e11d1         none                null                local
```

5. Lancer le conteneur

```
docker run -d -p 80:80 \
    -v vol_nginx:/var/www \
    --network net_nginx \
    -e VARIABLE=truc \
    --name nginx nginx
```

```
vagrant@mydocker:~$ docker run -d -p 80:80 \  
> -v vol_nginx:/var/www \  
> --network net_nginx \  
> -e VARIABLE=truc \  
> --name nginx nginx  
f9bc5dbbf8074a7f96cbfc0b80c4df585a0e443d2ad93f58870f84cef22df699  
vagrant@mydocker:~$
```

6. Vérifiez dans un navigateur



7. Supprimez les ressources

Conteneurs

```
docker container stop $(docker container ls -aq)
```

```
docker container rm $(docker container ls -aq)
```

Volumes

```
docker volume rm $(docker volume ls -q)
```

Réseau

```
docker network rm net_nginx
```

Etape 2 : Comparaison avec docker-compose, on commence par le fichier

Créez un répertoire compose, dans ce répertoire créez un fichier **docker-compose.yml**

Ajouter le contenu suivant :

```
version: '2'

volumes:
  vol_nginx:

networks:
  net_nginx:

services:
  nginx:
    image: nginx
    container_name: nginx
    volumes:
      - vol_nginx:/var/www
    environment:
      - VARIABLE=truc
    ports:
      - "80:80"
    networks:
      - net_nginx
```

Puis on lance le tout :

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

```
vagrant@mydocker:~/compose$ vagrant@mydocker:~/compose$ docker-compose up -d
Creating network "compose_net_nginx" with the default driver
Creating volume "compose_vol_nginx" with default driver
Creating nginx ...
Creating nginx ... done
```

Vérifiez dans un navigateur



Etape 3 : MySQL + GUI

<https://docs.docker.com/samples/library/mysql/>

1. Créez un répertoire ex : **doc-compose**

Dans ce répertoire créez un fichier **docker-compose.yml** et ajouter le contenu suivant

docker-compose.yml

```
version: '2.0'
services:
  adminer:
    image: adminer
    restart: always
    ports:
      - 8080:8080
  db:
    image: mysql:5.6
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: azerty
```

Exécution

Ouvrir un terminal **T1** dans le répertoire **doc-compose**

```
docker-compose up
```

Ouvrir un autre terminal **T2**

```
docker ps
```

```
vagrant@mydocker:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0c84654cf087	adminer	"entrypoint.sh docke..."	2 minutes ago	Up 2 minutes	0.0.0.0:8080->8080/tcp	docompose_adminer_1
458db1a45aba	mysql:5.6	"docker-entrypoint.s..."	2 minutes ago	Up 2 minutes	3306/tcp	docompose_db_1

```
docker exec -it <id container mysql> bash
```

```
See 'docker --help'
vagrant@mydocker:~$ docker exec -it 45 bash
root@458db1a45aba:/#
```

```
mysql -u root -p
```

```
root@458db1a45aba:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.6.45 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

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>
```

```
show databases;
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
+-----+
3 rows in set (0.34 sec)

mysql>
```

```
Exit
```

```
mysql> exit
Bye
root@458db1a45aba:/#
```

Ouvrir un navigateur sur l'url : <http://192.168.2.10:8080/>

← → ↻ ⚠ Non sécurisé | 192.168.2.10:8080

Langue: Français ▼

Adminer 4.7.3

Authentification

Système	MySQL ▼
Serveur	db
Utilisateur	root
Mot de passe	*****
Base de données	

☒ Authentification ☐ Authentification permanente

Nous obtenons

Langue: Français ▼

Adminer 4.7.3

MySQL » db

Déconnexion

DB: ▼

Requête SQL Importer Exporter

Sélectionner la base de données

Créer une base de données Privilèges Liste des processus Variables Statut

Version de MySQL : 5.6.45 via l'extension PHP PDO_MySQL

Authentié en tant que : root@172.18.0.3

Base de données - Rafraîchir	Interclassement	Tables	Taille	Calcul
<input type="checkbox"/> information_schema	utf8_general_ci	?	?	?
<input type="checkbox"/> mysql	latin1_swedish_ci	?	?	?
<input type="checkbox"/> performance_schema	utf8_general_ci	?	?	?

Sélectionnée(s) (0)
Supprimer

Revenir dans le terminal T1

```
ctrl+c  
docker-compose down  
docker ps
```