

TP : Docker Compose

Dans ce lab, nous allons illustrer l'utilisation de Docker Compose et lancer l'application Voting App de Docker. Cette application est très utilisée pour des présentations et démos.

L'application Voting App est une application micro-services composées de 5 services.

- **vote:** front-end permettant à un utilisateur de voter entre 2 options
- **redis:** database redis dans laquelle sont stockés les votes
- **worker:** service qui récupère les votes depuis redis et met les résultats dans une database postgres
- **db:** database postgres dans laquelle sont stockés les résultats
- **result:** front-end permettant de visualiser les résultats

1. Clone du repository GitHub

Avec les commandes ci-dessous, récupérez le code de l'application depuis le repository GitHub.

```
$ git clone https://github.com/docker/example-voting-app
$ cd example-voting-app
$ ls
```

Plusieurs fichiers, au format Docker Compose, sont disponibles. Chacun décrivant l'application avec une complexité plus ou moins importante..

Nous utilisons ici le fichier docker-compose.yml qui est le fichier par défaut. Ce fichier est très intéressant car il définit également des volumes et networks en plus des services.

2. Lancement de l'application

Lancer l'application à partir du fichier docker-compose.yml est très simple, il suffit de lancer la commande suivante:

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

Note: avec l'option -d, l'application est lancée en background.

Les étapes réalisées lors du lancement de l'application sont les suivantes:

création du volume

création des networks

construction des images pour les services custom (vote, worker, result) et récupération des images redis et postgres

lancement de containers pour chaque service

3. Les containers

Avec la commande suivante, listez les containers qui ont été lancés.

Le nom de container est préfixé par le répertoire dans lequel le fichier Compose est présent.

```
$ docker-compose ps
```

4. Les volumes

Listez les volumes avec la CLI, et vérifiez que le volume défini dans le fichier docker-compose.yml est présent.

`docker volume ls`

Le nom du volume est préfixé par le répertoire dans lequel le fichier Compose est présent.

5. Les networks

Listez les networks avec la CLI. Les deux networks définis dans le fichier `docker-compose.yml` sont présents.

`$ docker network ls`

De même que pour le volume, leur nom est préfixé par le nom du répertoire.

Note: comme nous sommes dans le contexte d'un hôte unique (et non dans le contexte d'un cluster Swarm), le driver utilisé pour la création de ces networks est du type bridge. Il permet la communication entre les containers tournant sur une même machine..

6. Utilisation de l'application

Nous pouvons maintenant accéder à l'application:

nous effectuons un choix entre les 2 options depuis l'interface de vote à l'adresse `http://localhost:5000`.

nous visualisons le résultat depuis l'interface de résultats à l'adresse `http://localhost:5001`

7. Scaling du service worker

Par défaut, un container est instantié pour chaque service. Il est possible, avec la commande `scale` de changer ce comportement et de scaler un service une fois qu'il est lancé.

Avec la commande suivante, augmenter le nombre de worker à 2.

`$ docker-compose scale worker=2`

Listez les containers présent. Les 2 containers relatifs au service worker sont présent.

`$ docker-compose ps`

Notes: il n'est pas possible de scaler les services `vote` et `result` car ils spécifient tous les 2 un port, plusieurs containers ne peuvent pas utiliser le même port de la machine hôte
il n'est pas non plus possible de scaler les services `db` et `redis` car ils spécifient tous les 2 l'option `container_name`, plusieurs containers ne peuvent pas avoir le même nom.

`docker-compose scale redis=2`

ERROR: for redis Cannot create container for service redis: Conflict. The container name "/redis"

...

...

8. Suppression de l'application

Avec la commande suivante, stoppez l'application. Cette commande supprime l'ensemble des éléments créés précédemment (volume et networks).

`$ docker-compose down`

Résumé

Cet exemple illustre l'utilisation de Docker Compose sur l'exemple bien connu de la Voting App dans le cadre d'un hôte unique.