

TP : Docker Compose

Introduction

Dans ce TP nous allons aborder Docker Compose, l'outil intégré à Docker pour gérer des stacks applicatives.

Nous montrerons comment :

- décrire une stack
- lancer une stack
- gérer les volumes et les réseaux
- lier les applications dans une stack
- upgrader notre application

Pré-requis :

- Connaître les bases de la CLI Docker
- Connaître les paramètres principaux pour lancer un conteneur

Installation

Deux méthodes d'installation sont disponibles, l'une via les packages pip, l'autre via les releases obtenues sur la page GitHub du projet docker-compose. Nous utiliserons cette dernière :

```
# curl -L "https://github.com/docker/compose/releases/download/v2.2.2/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
# chmod +x /usr/local/bin/docker-compose  
# docker-compose --version  
docker-compose version v2.2.2, build b31ff33
```

Une version supérieure à v2.2.2 conviendra aussi.

Préparation de l'environnement

Afin de travailler dans un environnement vierge et ne impacter d'autres stacks docker-compose, créez un dossier et placez vous à l'intérieur. Toutes les commandes qui suivront seront lancées depuis ce répertoire.

```
~ $ mkdir mystack  
~ $ cd mystack
```

Notre stack

Nous utiliserons l'application gitea comme démonstrateur. gitea est un serveur Git écrit en Go concurrent de Gitlab ou Bitbucket par exemple. L'application se compose du serveur lui même ainsi que d'une base de données. Ces deux parties seront déployées par Docker Compose. Copier/coller cette stack dans un fichier docker-compose.yml

```

version: '2'
services:
  gitea_server:
    image: gitea/gitea:1.19
    ports:
      - "3000:3000"
      - "10022:22"
    volumes:
      - gitea_server_data:/data
    networks:
      - gitea_network
  gitea_db:
    image: mysql:8
    volumes:
      - gitea_db_data:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=password
      - MYSQL_DATABASE=gitea
    networks:
      gitea_network:
        aliases:
          - mysql
volumes:
  gitea_server_data:
    driver: local
  gitea_db_data:
    driver: local
networks:
  gitea_network:

```

Avant de lancer votre stack, essayez de retrouver :

- la version de gitea utilisée
- La façon dont MySQL récupère son master password
- Le nom que va prendre MySQL grâce au service discovery de Docker
- Le port SSH sur lequel votre serveur gitea écoutera

Vous pouvez maintenant lancer votre stack :

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

Vérifiez que votre stack est correctement lancée :

```

$ docker-compose ps
Name
Command State Ports
-----
10dockercomposegitea_gitea_db_1  docker-entrypoint.sh mysqld Up    3306/tcp
10dockercomposegitea_gitea_server_1 /app/gitea/docker/start.sh ... Up
0.0.0.0:10022->22/tcp, 0.0.0.0:3000->3000/tcp

```

Vérifiez que :

- Les volumes ont bien été créés
- Trouvez leur emplacement sur votre système hôte
- Le réseau a bien été créé
- Les conteneurs sont bien démarrés et visibles par le démon Docker

Configuration de gitea

Rendez vous sur l'adresse de votre daemon Docker au port en écoute par gitea, probablement localhost:3000 et procédez à l'installation de gitea

- Sur quelle adresse se trouve votre container MySQL ?
- Quel est le password root de la base de données ?

Il n'est pas utile de remplir toutes les informations. Seule la partie présentée dans le screenshot a besoin

d'être mise à jour en fonction de votre docker-compose.yml.

Une fois l'installation terminée, vous devez vous créer un compte et ensuite vous loguer.

Créer un repository et ajoutez y un fichier quelconque. Nous souhaitons simplement tester la persistance des données.

Vérification de la persistance

Notre application fonctionne et nous afin de vérifier que nos conteneurs ne disposent pas de données

locales, nous allons les détruire :

```
$ docker-compose stop
$ docker-compose rm -f
```

- Vérifiez que les volumes sont toujours présents sur le système
- Vérifiez que vous avez bien perdu l'accès à l'application

On peut maintenant relancer notre application

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

- Vérifiez que vous avez récupéré l'accès à l'application
- Vérifiez que les données que vous y aviez mis y sont toujours

Upgrade de l'application

Notre serveur gitea est en version 1.19.

Mais une version 1.20 est disponible.

Nous allons modifier notre fichier docker-compose.yml et changer le tag de notre image gitea/gitea pour

utiliser la version 1.20 et relancer notre stack.

```
$ sed -i 's/1.19/1.20/g' docker-compose.yml
$ docker-compose stop
$ docker-compose rm -f
$ docker-compose up -d
```

Si vous retournez sur la page web gitea, vous devriez voir la version changer :

Bien entendu, Docker ne rend pas cette mise à jour magique. Il convient de tester cet upgrade dans un

environnement de test pour s'assurer que la nouvelle version n'impacte pas le schéma de la base de

données par exemple.