

DOCKER

TP DOCKER

Le but de ce TP est de vous faire découvrir les différentes commandes vous permettant de gérer vos conteneurs.

Pour se faire il vous faudra une VM linux neuve avec la configuration réseau fonctionnelle. (CF: Cours Linux)

I/Installation

Une fois votre machine configurée il vous faudra installer les différents paquets officiels de docker. Pour l'installer sur ma machine Debian j'ai simplement tapé "install docker debian" sur google et j'ai pris le lien du site officiel : (adaptez votre recherche en fonction de la distribution que vous utilisez)

<https://docs.docker.com/engine/install/debian/>

L'installation de docker s'arrête quand la procédure vous demande de lancer la commande `sudo docker run hello-world`.

Revoyons ensemble les différentes étapes de la procédure :

```
- sudo apt-get remove docker docker-engine docker.io containerd runc
```

Cette commande va purger votre machine de tout paquet docker précédemment installé afin de partir sur de bases saines.

```
- sudo apt-get update
- sudo apt-get install \
  ca-certificates \
  curl \
  gnupg \
  lsb-release
```

Ces commandes visent à mettre à jours la liste de paquets de vos dépôts ainsi qu'à installer certains paquets nécessaires à la suite de la procédure. (les "\n" à la fin des lignes permettent juste d'écrire une commande sur plusieurs lignes)

```
-
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

DOCKER

- sudo mkdir -p /etc/apt/keyrings
- curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

Cette commande permet de récupérer la clé de déchiffrement du dépôt et de l'inscrire dans le fichier /usr/share/keyrings/docker-archive-keyring.gpg.

- echo \
- "deb [arch=\$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
- https://download.docker.com/linux/debian \
- \$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

Cette commande permet de rajouter le dépôt officiel de docker dans la liste de vos dépôts, en spécifiant sa clé de déchiffrement.

- sudo apt-get update
- sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin

Et enfin on met à jours la liste des paquets de vos dépôts et on installe les paquets nécessaires au bon fonctionnement de Docker.

Pour tester que votre installation à bien fonctionné la procédure vous propose de lancer votre premier container avec la commande :

```
sudo docker run hello-world
```

En retour de cette commande vous devriez avoir un message vous indiquant que l'installation de docker s'est correctement déroulé.

Afin de pouvoir se passer du sudo devant vos commandes vous pouvez ajouter votre utilisateur au groupe docker (pensez à vous déconnecter/reconnecter pour appliquer la modification utilisateur).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

DOCKER

II/Docker search

Nous allons maintenant mettre en place un container permettant de déployer un serveur web avec php. En utilisant la commande `docker search php` vous pouvez constater qu'il existe énormément d'images ayant php dans le nom. Cela vient du fait que docker est un outil open source et à forte tendance communautaire donc beaucoup de gens publient leur images docker ayant diverses fonctionnalités. Nous allons utiliser l'image officiel php mais pas la version de base. En effet la version de base de l'image php permet de démarrer un container pour exécuter des scripts php. Nous utiliserons du coup la version `php:7.2-apache` comprenant également un service web afin de diffuser un site utilisant du php.

III/Démarrer un container

Pour démarrer une image docker nous utiliserons la commande `docker run <nom_image>`

Par exemple : `docker run php:7.2-apache`

En l'état cette commande pose quelques problèmes. Tout d'abord elle ne vous rend pas la main sur votre terminal. Il reste bloqué sur les logs de votre container. Pour remédier à ça il faut rajouter l'option `-d` à notre commande docker run. Avec l'option `-d` vous récupérez la main sur votre terminal mais votre solution ne fonctionne toujours pas car la commande n'est toujours pas complète. Si vous essayer d'accéder à l'adresse ip de votre VM en http votre navigateur ne devrait rien vous afficher. En effet à aucun moment vous n'avez spécifié quel port de votre machine doit permettre d'accéder à votre container. Pour se faire vous devez utiliser l'option `-p` qui permet d'indiquer quel port de votre VM permet d'accéder à votre container. Un serveur web diffuse par défaut son service sur le port 80, il faudra donc faire un lien entre un port de votre VM et le port 80 de votre container. Nous choisirons le port 8080 de la VM. Nous rajouterons également l'option `--name` pour donner un nom à notre container :

`docker run -d -p 8080:80 --name web php:7.2-apache`

Vous pouvez maintenant accéder à l'adresse de votre VM sur le port 8080 pour voir un jolie message d'erreur du serveur car votre container ne contient pas de site web.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

DOCKER

Faisons maintenant un peu de ménages dans vos containers.

Grâce à la commande `docker ps -a` vous pouvez afficher vos différents containers.

La commande `docker stop <id/nom_container>` vous permet de stopper un container et la commande `docker rm <id/nom_container>` permet de le supprimer.

Sachez que vous n'êtes pas obligé de saisir tout l'ID du container. Si 2 caractères suffisent à identifier votre container vous pouvez ne taper que ces 2 caractères pour le mentionner.

Exemple : container 1 ID : 548A321E4
 container 2 ID : 54996DFA5

Docker stop 54 => arrêtera les 2 containers
docker stop 548 => arrêtera le premier container

IV/Interagir avec un container

Vous disposez maintenant d'un container hébergeant un service web php qui est accessible sur le port 8080 de votre VM.

Ce container n'a actuellement pas de site web à diffuser c'est pour ça que vous obtenez une page d'erreur lors de l'accès par navigateur. Vous avez la possibilité de transférer des fichiers depuis votre VM vers votre container avec la commande `docker cp`.

Récupérez un de vos projets php que vous mettrez dans un dossier src dans votre répertoire personnel de votre VM.

Placez-vous dans le dossier src et tapez la commande `docker cp . web:/var/www/html`. Cette commande permet de copier tout le répertoire courant (".") dans le dossier /var/www/html du container appelé web.

Nous allons vérifier que la commande s'est bien effectuée en exécutant la commande `ls /var/www/html` dans notre container. Pour se faire nous allons utiliser la commande `docker exec`.

`docker exec web ls /var/www/html`

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

DOCKER

Vous devriez voir votre projet dans le résultat de la commande. Effectuez également la vérification en accédant à l'adresse ip de votre VM sur le port 8080 via votre navigateur.

`http://<ip_de_la_vm>:8080`

V/ Dockerfile

Vous disposez maintenant d'un container php diffusant votre site web mais si un problème survient ou si vous devez changer de serveur vous devrez de nouveau démarrer votre container et y copier les fichiers de votre site. Nous allons maintenant faire en sorte de créer une image php qui contiendra directement tout le code de votre site. De cette manière si vous devez recréer un container vous n'aurez qu'à utiliser votre image dans le docker run et votre container diffusera directement votre projet.

Pour se faire nous allons rédiger un fichier appelé Dockerfile que nous allons compiler en une image.

Nous utiliserons deux instructions : FROM et COPY. FROM permet d'indiquer à partir de quelle image on va se baser et COPY permet de copier des fichiers dans l'image :

`FROM php:7.2-apache`

`COPY src/ /var/www/html`

`RUN docker-php-ext-install pdo_mysql`

Une fois ce fichier rédigé vous pouvez taper la commande `docker build -t monimage` pour compiler le Dockerfile qui se trouve dans le repertoire courant (".") en lui donnant le nom "monimage" ("-t").

Démarrez maintenant votre container avec la même commande `docker run` que précédemment en remplaçant simplement le nom de l'image à utiliser par `monimage`

Vous pouvez constater que vous pouvez toujours retrouver votre site web à l'adresse de votre VM sur le port 8080.

VI/ Docker compose

Votre solution fonctionne mais si votre projet nécessite une base de donnée il n'est pas totalement opérationnel. Nous allons donc rajouter un container mariadb ayant la configuration nécessaire. Nous pourrions démarrer manuellement un container de base de données mais nous allons plutôt rédiger un fichier qui va permettre de démarrer nos deux containers simultanément. Ce fichier s'appelle docker-compose.yml et est rédigé en YAML (Yet Another Markup Language).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

DOCKER

Nous déclarerons dans ce fichiers deux services, un par container que nous voulons démarrer, et dans chaque service nous indiquerons tout ce que nous aurions utilisé dans les commande docker run.

```
version: '3.8'

services:
  web:
    build: .
    ports:
      - 8080:80
  bdd:
    image: mariadb
    environment:
      MARIADB_ROOT_PASSWORD: <votre_mot_de_passe_root>
      MARIADB_DATABASE: <nom_de_votre_database>
    volumes:
      - db:/var/lib/mysql

volumes:
  db:
```

Quelques précisions sur ce fichiers:

version: '3.8' est la version de YAML que nous utilisons.

web: est le nom du premier services que nous déclarons

build: . permet d'indiquer que pour démarrer le container il faudra compiler le Dockerfile présent dans le répertoire courant

ports: permet d'indiquer les ports à exposer

bdd: est le nom du deuxième service que nous déclarons

image: permet d'indiquer l'image à utiliser pour notre container

environment: permet de déclarer des variables d'environnement pour notre container (pensez à adapter la valeur de ces variables à votre projet)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

DOCKER

volumes: - db:/var/lib/mysql permet de créer un volume docker appelé db permettant de sauvegarder la configuration de votre base de données.

En l'état votre docker compose fonctionne. Vous pouvez démarrer votre solution avec la commande `docker compose up -d`. Cette commande va donc démarrer votre container php en le diffusant sur le port 8080 de votre VM ainsi que votre container Mariadb en créant la base de donnée que vous avez indiqué dans la variable MARIADB_DATABASE.

Il ne reste plus qu'une seule chose à gérer pour que votre projet soit totalement opérationnel : les tables de la base de donnée.

Pour structurer votre base avec les tables nécessaires à votre projet vous pouvez rédiger un script sql de création de tables et nous indiquerons dans notre docker compose de lancer ce script au démarrage du container.

Dans ce script, que vous nommerez `init.sql` et que vous placerez au même endroit où se situe votre fichier docker-compose.yml, vous indiquerez toutes les commandes sql de création de table dont vous avez besoin. Il faudra ensuite rajouter une ligne dans votre docker compose, dans la catégorie volumes de votre service bdd pour renseigner ce fichier :

```
- ./init.sql:/docker-entrypoint-initdb.d/init.sql
```

Cette ligne permet d'indiquer que le script `init.sql` se trouvant dans ce répertoire (celui du docker-compose.yml) doit être mis dans le dossier `/docker-entrypoint-initdb.d` de votre container. Les scripts placés dans ce dossier devront être exécuté au démarrage du container.

Vous pouvez maintenant éteindre le docker compose, si vous l'aviez lancé avant de rajouter le script dans le fichier, avec la commande `docker compose down -v`. Le "-v" permet d'indiquer a docker de supprimer également les volumes qui étaient liés à la solution. Vous pouvez maintenant démarrer votre solution avec la commande `docker compose up -d`. Si la commande s'exécute sans erreurs vérifiez que vos containers on bien démarrés avec la commande `docker ps -a` en vous assurant qu'ils sont tous les deux en statut "up". S'ils ne sont pas tous démarrés c'est qu'un problème est survenue au démarrage. Pour savoir d'où vient le problème vous pouvez éteindre votre compose (`docker compose down -v`) et le redémarrer sans le "-d" afin de voir s'afficher les logs et pouvoir identifier la source du problème. Une fois que votre solution démarre correctement vous pouvez accéder à votre site web sur le port 8080 de votre VM.

Pour finir je vous invite à publier votre répertoire de projet (contenant votre Dockerfile, votre docker-compose.yml, votre script init.sql ainsi que votre dossier de code source) sur un dépôt github. De

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
Maxime PRZYBYLO	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		

Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

DOCKER

cette manière depuis n'importe quel serveur linux disposant de docker vous pouvez pull votre solution et la déployer dessus.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Maxime PRZYBYLO</u>	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / DEVELOPPEMENT		