Virtualisation Avancée

TP1

Etape 1

a)

La différence observée dans les logs de docker est qu'il ne pull plus l'image du container.

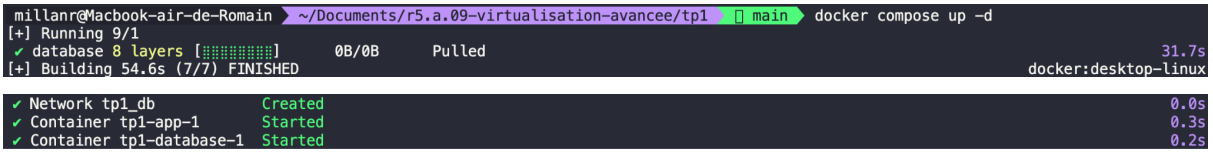
J'en déduis donc, que lorsqu'un container n'est pas disponible en local, l'image du container est téléchargée depuis internet.

d)

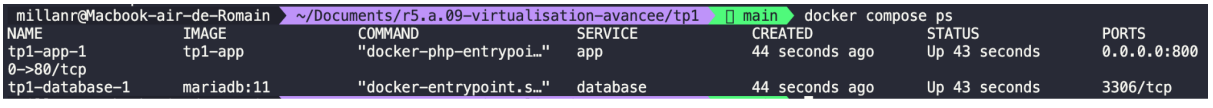
Dans le docker-compose, 2 containers seront lancés, et les ports exposées seront:

- 8000

e)



f)



g)

```yaml
version: '3' # version du fichier docker-compose


services: # Liste des services
database: # Service n°1 (Base de données)
image: mariadb:11 # Image du 1er service
environment: # Liste des variables d'environnement dans le container
```

```
- MARIADB_ROOT_PASSWORD=changeme # Variable d'environement 'MARIADB_ROOT_PASSWORD'
avec ca valeur
networks: # Liste des networks
- db # Utilisateur du network 'db'

app: # Service n°2 (Web)
build: # Consigne pour crée le container
context: . # Ou se situe le context du docker-compose
dockerfile_inline: | # C'est un Dockerfile mais directement écrit dans le
docker-compose au lieu d'avoir un autre fichier distinct
FROM php:7.2-apache
RUN docker-php-ext-install pdo pdo_mysql
RUN docker-php-ext-enable pdo pdo_mysql
volumes: # Liste des volumes du container
- ./app/src:/var/www/html/ # Associe le dossier ./app/src au dossier /var/www/html
dans le container
ports: # Liste des ports ouvert
- "8000:80" # Le port 8000 est liée au port 80 du container
networks: # Liste des networks
- db # Utilisateur du network 'db'

networks: # Liste des networks possible
db: # Création du network 'db'
```

Etape n°2

h)

Les pages PHP de l'application se trouve dans le dossier 'app/src' disponible dans le dossier 'tp1'

- config.php: Fichier de configuration pour les accès à la base de données
- helloworld.php: Première page, contenant un début de page HTML et des explications concernant les commentaires.
- index.php: Affiche les informations php avec la fonction phpinfo().
- init.php: Initialise la table 'user' dans la base de données et remplis 3 tuples dans celle-ci.
- readAll.php: Affiche tous les utilisateurs connus dans la base de données, si il se passe une erreur cela affiche l'erreur en plus.

k)

Lors de l'extinction du container 'mariadb' qui nous sert de Base de Données, les données ne sont pas sauvegardées. Donc cela cause une suppression de toutes les données dans celle-ci.

l)

```
volumes:
- ./database-data:/var/lib/mysql
```

m)

En ayant ajouté le stockage des données, je ne perd plus des données insérées dans la base de données lors de l'extinction du container.

Etape n°3

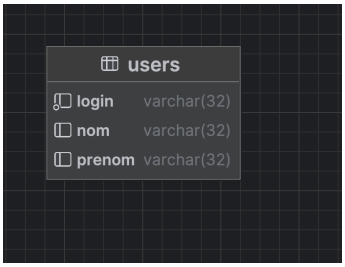
n)

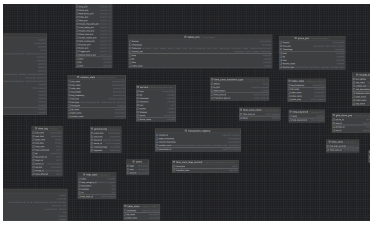
```
millanr@Macbook-air-de-Romain ~/Documents/r5.a.09-virtualisation-avancee/tp1  main ± docker exec -it tp1-app-1 bash
root@760ac7693366:/var/www/html#
```

o)

Grâce à la commande 'nmap database', je peux vérifier que le port 3306 est bien ouvert.

q)





	login	nom	prenom
1	bl	Leponge	Bob
2	jdo	Doe	John
3	aa	Alice	Alice
4	millanr	MILLAN	Romain

Etape n°4

r)

```
app: # Service n°2 (Web)
build: # Consigne pour crée le container
context: . # Ou se situe le context du dockerfile.
dockerfile_inline: | # Permet de crée un dockerfile directement dans le
docker-compose.
FROM php:8.2-apache
RUN docker-php-ext-install pdo pdo_mysql
RUN docker-php-ext-enable pdo pdo_mysql
```

Cela fonctionne bien en modifiant la version de php.

s)

Utilisation de PHP 5.4:

```
app: # Service n°2 (Web)
build: # Consigne pour crée le container
context: . # Ou se situe le context du dockerfile.
dockerfile_inline: | # Permet de crée un dockerfile directement dans le
docker-compose.
FROM php:5.4-apache
RUN docker-php-ext-install pdo pdo_mysql
RUN docker-php-ext-enable pdo pdo_mysql
```

Utilisation de Mariadb 10:

```
services: # Liste des services
database: # Service n°1 (Base de données)
image: mariadb:10 # Image du 1er service
```

En modifiant les 2 versions, l'application fonctionne très bien.

Etape 5

t)

Pour passer de mariadb à postgres j'ai modifier le container dans le docker-compose comme ceci:

```
database: # Service n°1 (Base de données)
image: postgres
environment:
```

```
POSTGRES_USER: root
POSTGRES_PASSWORD: changeme
PGDATA: /data/postgres
volumes:
- ./postgres:/data/postgres
ports:
- "5432:5432"
networks: # Liste des networks
- db # Utilisateur du network 'db'
```

De plus j'ai dus activer les driver dans le service web:

```
app: # Service n°2 (Web)
build: # Consigne pour crée le container
context: . # Ou se situe le context du dockerfile.
dockerfile_inline: | # Permet de crée un dockerfile directement dans le
docker-compose.
FROM php:5.4-apache
RUN docker-php-ext-install pdo pdo_pgsql
RUN docker-php-ext-enable pdo pdo_pgsql
```

u)

Tout le code fonctionne bien comme à l'étape n°2

Pour la persistance des données il m'as fut modifier le volumes ainsi qu'ajouter une variable d'environnement pour indiquer à postgres de stocker les données dans un dossier particulier:

```
PGDATA: /data/postgres
volumes:
- ./postgres:/data/postgres
```

Les développeurs ont toujours accès à la base de données mais maintenant avec le port '5432'