Tp : Docker compose

Etape 1 : Installation du docker-compose

- Sous Linux : exécutez les commandes

#sudo curl –L "https://github.com/docker/compose/releases/download/1.29.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

#sudo chmod +x /usr/local/bin/docker-compose

Pour tester l'installation, exécutez :

#docker-compose version

- Sous windows : il est déjà installé avec docker desktop
- Sous MacOs : il est déjà installé avec docker desktop

Etape 2 : Création de deux conteneurs

- Un conteneur base de donnée (par exp mysql)
- Un conteneur pour la création d'un site web (par exemple wordpress)

Etape 3 : Création d'un fichier docker-compose

Commencez d'abord par créer un fichier et nommez le docker-compose.yml :

```
version: '3'

services:

  db:

    image: mysql:5.7

    volumes:

      - db_data:/var/lib/mysql

    restart: always

    environment:

      MYSQL_ROOT_PASSWORD: somewordpress

      MYSQL_DATABASE: wordpress

      MYSQL_USER: wordpress

      MYSQL_PASSWORD: wordpress
```

// Ce service décrit notre conteneur mysql :

- On va utiliser une image mysql qui existe sur docker hub
- On définit le volume qui va nous permettre de stocker l'ensemble du contenu du dossier
- On définit l'option de redémarrage
- On définit les différentes variables d'environnement nécessaires pour le bon fonctionnement de Mysql

```
wordpress:
  depends_on:
    - db
  image: wordpress:latest
  ports:
    - "8000:80"
  restart: always
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wordpress
    WORDPRESS_DB_PASSWORD: wordpress
    WORDPRESS_DB_NAME: wordpress
```

// Ce service nout permet de décrire notre conteneurWordpress:

- depends_on, nous permet de créer une dépendance entre deux conteneurs. Ainsi, Docker démarrera le service db avant de démarrer le service Wordpress.
- ports, permet de dire à Docker Compose qu'on veut exposer un port de notre machine hôte vers notre conteneur, et ainsi le rendre accessible depuis l'extérieur.

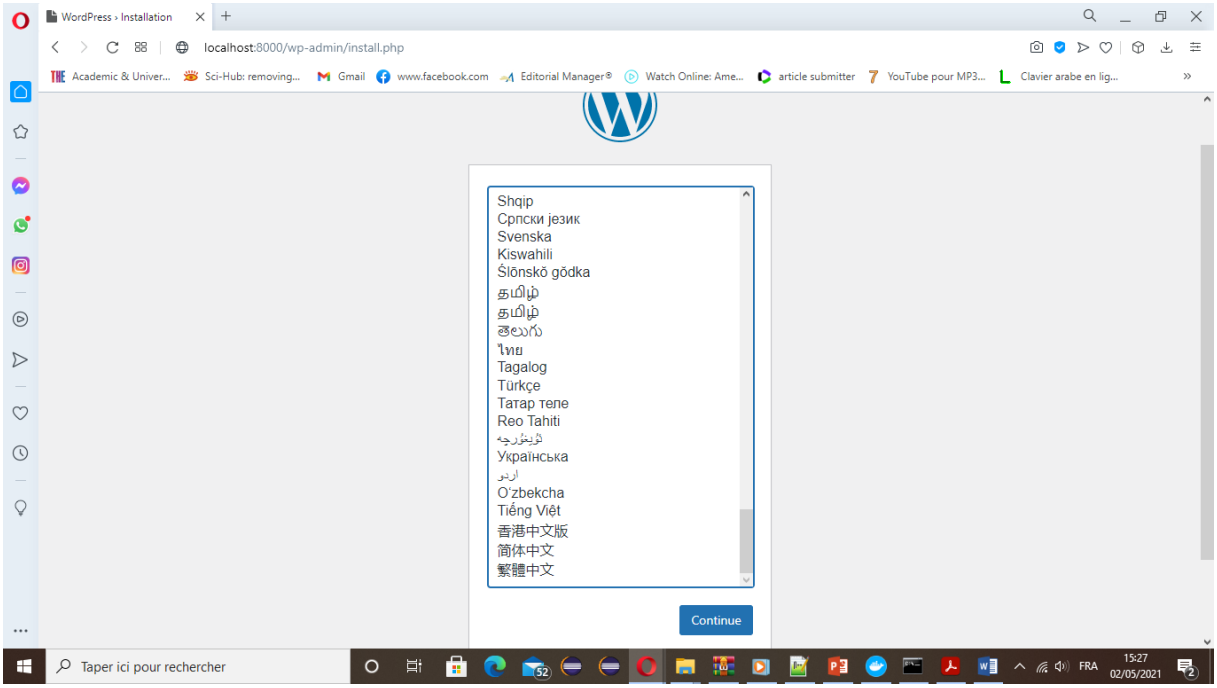
```
volumes:
  db_data: {}
```

// Volumes spécifier le dossier de stockage des données.

Pour lancer votre docker-compose, executez la commande :

#docker-compose up –d

Puis celui-ci lance les deux conteneurs sur votre système. Pour accéder aux conteneurs vous pourrez voir le résultat dans votre navigateur: http://localhost:8000/wp-admin/install.php



Etape 4 : Des instructions pour manipuler docker-compose

Pour exécuter les services du docker-compose.yml

#docker-compose up

-d : Exécuter les conteneurs en arrière-plan

Pour lister des conteneurs du Docker Compose

#docker-compose ls

-a ou --all : afficher aussi les conteneurs stoppés

Pour voir Sorties/erreurs des conteneurs du Docker Compose

#docker-compose logs

-f : suivre en permanence les logs du conteneur

-t : afficher la date et l'heure de la réception de la ligne de log

--tail=<NOMBRE DE LIGNE> = nombre de lignes à afficher à partir de la fin pour chaque conteneur.

Pour tuer les conteneurs du Docker Compose

#docker-compose kill

Pour stopper les conteneurs du Docker Compose

#docker-compose stop

-t ou --timeout : spécifier un timeout en seconde avant le stop (par défaut : 10s)

Pour démarrer les conteneurs du Docker Compose

#docker-compose start

Pour arrêtez les conteneurs et supprimer les conteneurs, réseaux, volumes, et les images

#docker-compose down

-t ou --timeout : spécifier un timeout en seconde avant la suppression (par défaut : 10s)

Pour supprimer des conteneurs stoppés du Docker Compose

#docker-compose rm

-f ou --force : forcer la suppression

Pour lister les images utilisées dans le docker-compose.yml

#docker-compose images

Exercice :

- Créer une image (Dockerfile) qui sera un serveur apache avec du php qui aura la capacité de se connecter à un serveur MySql extérieur.
- Faire un pull d'une image mysql.
- Créer votre fichier docker-compose.