

# TP #2

## Etape 0

Stopper et supprimer tous les containers existants

Créer un répertoire docker-tp2

Initialiser un repository git dans ce répertoire

Chaque étape doit être dans un sous-répertoire nommé "etapeX" (X étant le numéro de l'étape)

Il peut être utile au fur et à mesure des étapes d'organiser les fichiers dans des sous-répertoires (config, src, ... par exemple).

Le TP sera à rendre sous la forme d'un repository github dont vous me donnerez le lien (et les droits de consultation ;)

## Etape 1

2 containers nommés comme suit :

- HTTP : 1 container avec un serveur HTTP qui écoute sur le port 8080
- SCRIPT : 1 container avec un interpréteur PHP (plus le protocole FPM pour NGINX)

Une page index.php qui lorsqu'elle est appelée exécute la fonction php\_info() et qui sera situé dans les containers dans le répertoire /app.

Test de validité de l'exercice : avec un navigateur voir le résultat de l'exécution du php\_info()

Décommenter et remplacer les lignes 30 à 36 par les suivantes :

```
location ~ /\.php$ {
    root           /app;
    fastcgi_pass   script:9000;
    fastcgi_index  index.php;
    fastcgi_param  SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    include        fastcgi_params;
}
```

## Etape 2

3 containers nommés comme suit :

- HTTP : 1 container avec un serveur HTTP qui écoute sur le port 8080
- SCRIPT : 1 container avec un interpréteur PHP (plus le protocole FPM pour NGINX)

- DATA : 1 container avec un serveur de base données SQL (MariaDB, MySQL, PostgreSQL, ...)

Une page test\_bdd.php qui lorsqu'elle est appelée va executer 2 requêtes CRUD (Request : lecture, Create Update Delete : écriture) au minimum sur le serveur SQL : 1 lecture et 1 écriture

Test de validité de l'exercice : avec un navigateur voir le résultat de l'exécution de la page en retournant un résultat différent et dépendant du contenu de la base de données à chaque refresh de la page

## Etape 3

3 containers nommés comme suit :

- HTTP : 1 container avec un serveur HTTP qui écoute sur le port 8080
- SCRIPT : 1 container avec un interpréteur PHP (plus le protocole FPM pour NGINX)
- DATA : 1 container avec un serveur de base données SQL (MariaDB, MySQL, PostgreSQL, ...)

Remplacer la/les pages PHP simples par un package Wordpress complet.

Test de validité de l'exercice : avec un navigateur voir l'interface d'admin/installation de Wordpress afin de finaliser l'installation de celui-ci

## Etape 4

Convertir la configuration de l'étape 3 en Docker Compose

Test de validité de l'exercice : identique à l'étape 3