

Labo web infrastructure

Etape 1 : Serveur HTTP statique

Infrastructure

Dans le répertoire `./docker-images/apache-php-image/`

Dossier `src/` -> contient les fichiers sources du site web

Dockerfile :

```
FROM php:7.2-apache
COPY src/ /var/www/html/
```

FROM : Instruction utilisée pour spécifier le nom de l'image docker et start le processus

COPY : Instruction utilisée pour copier un fichier ou un répertoire de notre hôte vers l'image Docker. Copie l'élément dans le contexte d'instruction de docker

Tester et lancer

- cloner le repository : `git@github.com:Otabiel/web_Infrastructure.git` SSH `h`
- `https://github.com/Otabiel/web_Infrastructure.git` HTTPS
- Ce déplacer à la racine du projet
- Lancer la commande `docker build -t nom_a_choix docker-images/apache-php-image/` pour construire l'image docker du serveur apache avec le contenu du site, du dossier `src/`
- - Linux : `docker run -d nom_a_choix`
 - Autres : `docker run -d -p 9090:80 nom_a_choix` (Le numéro de port peut être changé)
- Il faut récupérer le `NAMES` de notre docker avec la commande suivante : `docker ps`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9de7265f1d93	res/apache	"docker-php-entrypoi..."	2 minutes ago	Up 2 minutes	80/tcp	blissful_almeida

- On récupère l'adresse IP avec la commande `docker logs NAMES`

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
[Fri May 28 10:52:58.218561 2021] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.2.34 configured -- resuming normal operations
[Fri May 28 10:52:58.218613 2021] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Dans cet exemple l'adresse ip est 172.17.0.2

- Pour modifier la configuration apache dockerisé, il faut lancer la commande `docker exec -it NAMES /bin/bash`. Ceci va nous lancer une session du terminal sur le docker. Ensuite, on peut se déplacer dans `/etc/apach2` avec la commande `cd` et modifier ce qu'on souhaite.

Etape 2 : HTTP dynamique

Infrasctructure

Dans le répertoire `./docker-images/express-image/`

dossier `src/` qui contient le fichier source de l'application JavaScript

Dockerfile :

```
FROM node:15.13.0
COPY src/ /opt/app
CMD ["node", "/opt/app/index.js"]
```

Tester et lancer

- Il faut consruire les dépendances node avec la commande `npm install` depuis le répertoire `./docker-images/express-image/src/`. (si on n'a pas la commande `npm -> https://www.npmjs.com/`)
- Ensuite il faut construire le container avec la commande `docker build -t nom_a_choix docker-images/express-image/` depuis la racine du projet.
- Lancer le container avec la commande `docker run -d -p 9091:3000 nom_a_choix`.
- Lister les adresse ip avec la commande `ip addr` pour trouver celle de docker.
- Accéder à l'adresse ip de docker via un navigateur comme tel `http://172.17.0.1:9091`
- Pour terminer l'exécution du container, utiliser la commande `docker kill NAMES`. `NAMES` se trouve avec la commande `docker ps`.

Etape 3 : Proxy

Infrastructure

Dans le répertoire `./docker-images/apache-reverse-proxy`

Dockerfile :

```
FROM php:7.2-apache
COPY conf/ /etc/apache2

RUN a2enmod proxy proxy_http
RUN a2ensite 000-* 001-*
```

Script build et script run permettant de construire et d'exécuter les containers.

Un dossier `conf/` contenant les configuration pour les sites.

Attention : les IPs sont hardcoder et doivent donc toujours être vérifiées pour correspondre au contenu du fichier de configuration `001-reverse-proxy.conf` contenu dans `./docker-images/apache-reverse-proxy/conf/sites-available`.

Tester et lancer

- Avant de faire quoique ce soit, vérifier qu'aucun autre container soit actif avec la commande `docker ps`. Vu qu'on a hardcodé les adresse IP, cela pourrait poser problème si d'autres containers utilise les adresse IP hardcodées déjà utilisées. Pour kill les container en cours, utiliser la commande `docker kill NAMES`. Enfin, on clean le tout avec la commande `docker rm docker ps -qa`.
- Exécuter le script `./docker-images/apache-reverse-proxy/build.sh` pour construire les containers.
- Exécuter le script `./docker-images/apache-reverse-proxy/run.sh` pour run les containers.
- Depuis un terminal (linux) aller modifier le fichier hosts en lançant `sudo !!` ou `sudo nano /etc/hosts` et ajouter une ligne contenant notre adresse IP et le nom du domaine tel que : `192.168.1.123 address.res.ch`.
- Accéder au site dynamique express `http://address.res.ch:8080/api/address/`.