UNIVERSITÉ DE RENNES 1

ESIR ECOLE SUPERIEURE D'INGENIEURS DE RENNES

26/10/2018

Docker – TD2

> *« J'atteste que ce travail est original, qu'il indique de façon appropriée tous les emprunts, et qu'il fait référence de façon appropriée à chaque source utilisée »*

Léo Guilpain – ESIR 3 - IoT

Table des matières

Introduction

Le but de ce TP est de d'héberger chaque site dans son propre conteneur docker.

Partie 1 : Interconnexion de conteneurs

Question 01

```
leo@leo:~/ESIR3/TP2$ mkdir td_docker
leo@leo:~/ESIR3/TP2$ cd td_docker/
leo@leo:~/ESIR3/TP2/td_docker$ mkdir proxy
leo@leo:~/ESIR3/TP2/td_docker$ mkdir site1
leo@leo:~/ESIR3/TP2/td_docker$ mkdir site2
leo@leo:~/ESIR3/TP2/td_docker$ cd proxy/
leo@leo:~/ESIR3/TP2/td_docker/proxy$ touch Dockerfile
leo@leo:~/ESIR3/TP2/td_docker/proxy$ touche site1.conf
La commande « touche » est introuvable, vouliez-vous dire :
 La commande « touch » du paquet « coreutils » (main)
touche : commande introuvable
leo@leo:~/ESIR3/TP2/td_docker/proxy$ touch site1.conf
leo@leo:~/ESIR3/TP2/td_docker/proxy$ touch site2.conf
leo@leo:~/ESIR3/TP2/td_docker/proxy$ cd ..
leo@leo:~/ESIR3/TP2/td_docker$ cd site1
leo@leo:~/ESIR3/TP2/td_docker/site1$ touch index.html
leo@leo:~/ESIR3/TP2/td_docker/site1$ cd ..
leo@leo:~/ESIR3/TP2/td_docker$ cd site2
leo@leo:~/ESIR3/TP2/td_docker/site2$ touch index.html
leo@leo:~/ESIR3/TP2/td_docker/site2$
```

Figure 1 : Création des dossiers et fichiers

Question 02

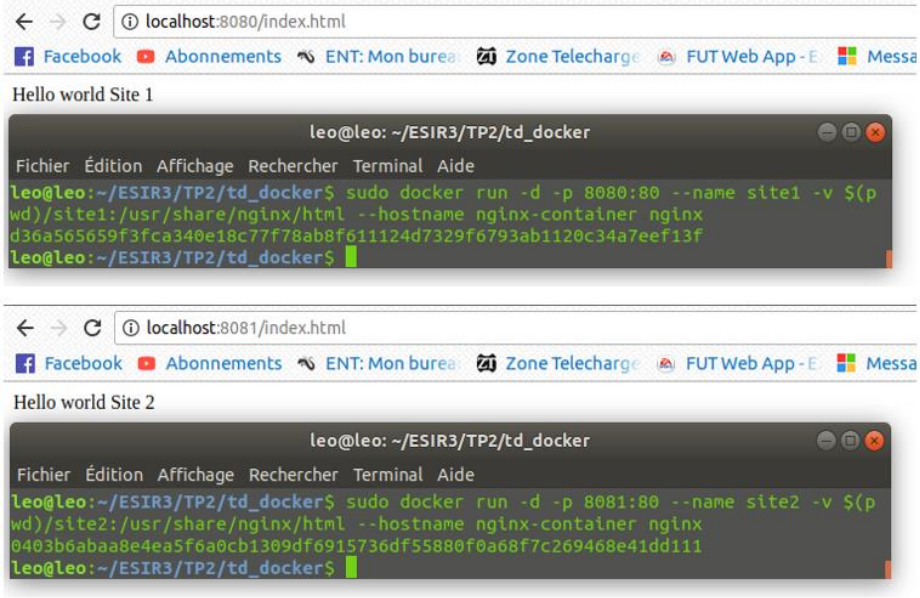


Figure 2 : Visualisation des sites

Question 03

Dans un premier temps on récupère les adresses IP à l'aide de la commande ci-dessous. On obtient pour le site 1 : **172.17.0.2** et pour le site 2 : **172.17.0.3**

```
leo@leo:~/ESIR3/TP2/td_docker$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' site1
172.17.0.2
leo@leo:~/ESIR3/TP2/td_docker$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' site2
172.17.0.3
```

Figure 3 : Récupération adresses IP

On configure le fichier « site1.conf » comme ceci :

Ouvrir — site1.conf — ~/ESIR3/TP2/td2-LeoGui/td2/td_docker/proxy

```
server {
    listen 80;
    server_name  localhost;

    location / {
        proxy_pass http://172.17.0.2:8080;
    }

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

Figure 4 : Site1.conf

À noter que le fichier « site2.conf » est configuré de façon similaire en modifiant l'adresse IP et le numéro de port.

Ensuite on modifie le fichier Dockerfile de la même façon que dans le TP précédent.

```
FROM nginx
COPY ./site1.conf etc/nginx/conf.d/site1.conf
COPY ./site2.conf etc/nginx/conf.d/site2.conf
VOLUME /etc/nginx
```

Figure 5 : Dockerfle

Question 04

Une fois le fichier Dockerfile configuré, il suffit de lancer la commande « docker build -t myproxy . » :

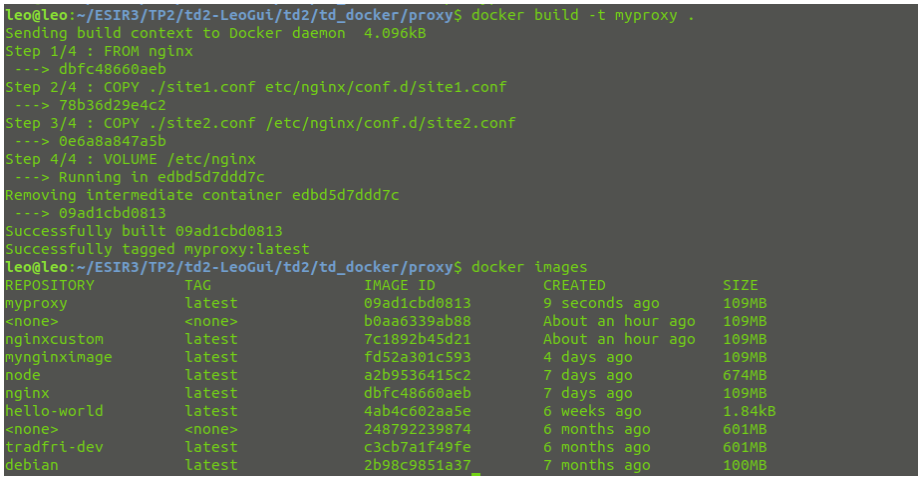


Figure 6 : Build de l'image

On voit bien grâce à “docker images” que l’image a été correctement créée.

Pour créer l’image et lancer le conteneur, il suffit de faire comme ceci :

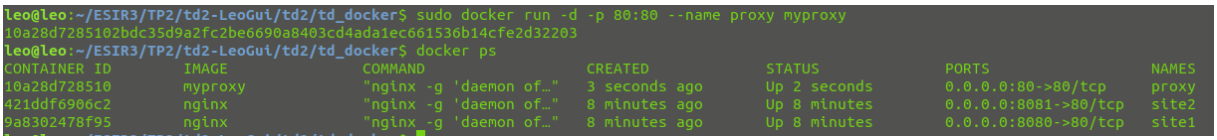


Figure 7 : Lancement du conteneur

On a lancé le conteneur « proxy » à partir de l’image « myproxy »

Question 05

On modifie le fichier hosts qui se trouve sur ordinateur à l’aide de la commande « vim »

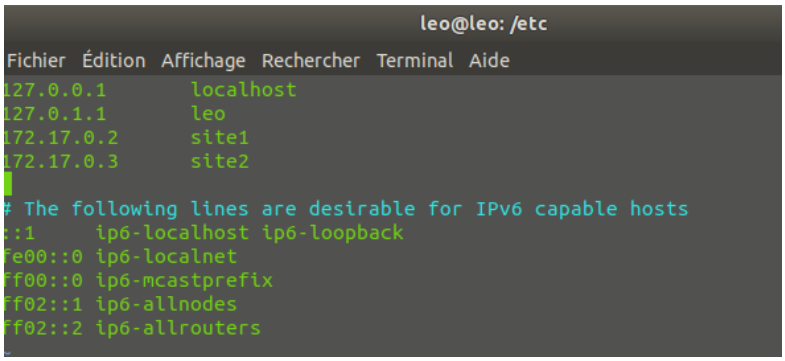


Figure 8 : etc/host

Ensuite lorsque l'on fait la commande "curl http://site1" et "curl http://site2" , on obtient ceci :

Figure 9 : curl site1

Les fichiers « index.html » des deux sites ont bien été liés correctement car on peut maintenant y avoir accès.

Question 06

Le problème de cette solution c'est qu'il faut connaître l'adresse IP des deux sites. Il faut également configurer sur l'ordinateur les adresses IP. Donc si ces dernières changent, la connexion ne se fait plus. De plus, les adresses IP et les numéros de ports sont stockés dans les fichiers de configuration, cela peut donc engendrer des problèmes de confidentialité mais surtout des problèmes de sécurisation.

Docker Link :

Question 01

On effectue la commande ci-dessous pour lier les conteneurs :

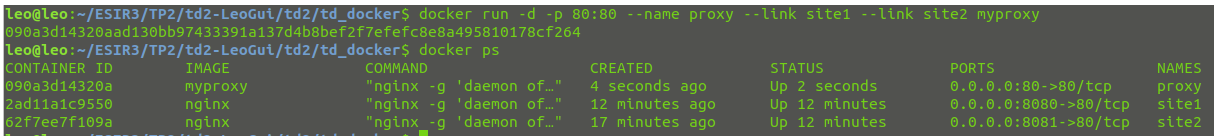


Figure 10 : lien entre les conteneurs

Ensuite on regarde les variables d'environnement :

```
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ docker exec proxy printenv
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=090a3d14320a
SITE1_PORT=tcp://172.17.0.2:80
SITE1_PORT_80_TCP=tcp://172.17.0.2:80
SITE1_PORT_80_TCP_ADDR=172.17.0.2
SITE1_PORT_80_TCP_PORT=80
SITE1_PORT_80_TCP_PROTO=tcp
SITE1_NAME=/proxy/site1
SITE1_ENV_NGINX_VERSION=1.15.5-1~stretch
SITE1_ENV_NJS_VERSION=1.15.5.0.2.4-1~stretch
SITE2_PORT=tcp://172.17.0.3:80
SITE2_PORT_80_TCP=tcp://172.17.0.3:80
SITE2_PORT_80_TCP_ADDR=172.17.0.3
SITE2_PORT_80_TCP_PORT=80
SITE2_PORT_80_TCP_PROTO=tcp
SITE2_NAME=/proxy/site2
SITE2_ENV_NGINX_VERSION=1.15.5-1~stretch
SITE2_ENV_NJS_VERSION=1.15.5.0.2.4-1~stretch
NGINX_VERSION=1.15.5-1~stretch
NJS_VERSION=1.15.5.0.2.4-1~stretch
HOME=/root
```

Figure 11 : Variables d'environnement

Comme prévu, nous avons des liens entre les différents conteneurs. Par exemple, nous avons accès aux différentes adresses IP. Nous allons donc pouvoir faire des références à ces variables.

Question 02

Pour avoir accès au shell du conteneur "proxy", on effectue la commande suivante :

"sudo docker exec -it proxy bash".

Ensuite après avoir mis à jour, on télécharge et installe "vim" pour pouvoir modifier le fichier en cas de besoin.

En naviguant dans les répertoires du conteneur « proxy », et surtout dans le répertoire « etc/host », on peut voir que les adresses IP ont bien été récupérées correctement.

```
leo@leo: ~/ESIR3/TP2/td2-LeoGui/td2/td_docker
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      site1 nginx-container
172.17.0.3      site2 nginx-container
172.17.0.4      090a3d14320a
```

Figure 12 : etc/host

Question 03

site2.conf

```
server {
    listen 80;
    server_name  site2;

    location / {
        proxy_pass http://site2;
        }

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

site1.conf

```
server {
    listen 80;
    server_name  site1;

    location / {
        proxy_pass http://site1;
    }

    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   /usr/share/nginx/html;
    }
}
```

Figure 13 : Fichiers de configuration

On a ici, dans les deux fichiers de configuration, remplacé l'adresse IP ainsi que le numéro de port par le nom des conteneurs. Comme le lien a été fait auparavant, le conteneur « proxy » a accès à ces variables. Cela permet également une meilleure sécurisation des données.

Question 4

Il faut donc refaire toutes les manipulations. Après ceci, on se rend compte du bon fonctionnement en réalisant la commande "curl http://site1". Cette dernière renvoie la même chose que précédemment alors que l'on a modifié les fichiers de configuration. Cela signifie donc que le lien a bien été fait correctement. Désormais le proxy connait les noms de domaines "site1" et "site2"

```
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ sudo docker exec -it proxy bash
root@7599c05f3cdd:/# cd etc
root@7599c05f3cdd:/etc# cat hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      site1 nginx-container
172.17.0.3      site2 nginx-container
172.17.0.4      7599c05f3cdd
root@7599c05f3cdd:/etc# exit
exit
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ curl http://site1
<html>
<header><title>Site 1</title></header>
<body>
Hello world Site 1
</body>
</html>
```

Figure 14 : curl site1

Partie II. Docker-compose

Question 01

```
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ touch docker-compose.yml
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ cd site1
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker/site1$ touch Dockerfile
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker/site1$ cd ..
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker$ cd site2
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker/site2$ touch Dockerfile
leo@leo:~/ESIR3/TP2/td2-LeoGui/td2/td_docker/site2$
```

Figure 15 : Création des fichiers

Question 02

On modifie le fichier “docker-compose.yml” comme ceci :

Ouvrir — docker-compose.yml — ~/ESIR3/TP2/td2-LeoGui/td2/td_docker

```
proxy:
 build: ./proxy
 container_name: proxy
 ports:
  - 80:80
 links:
  - site1
  - site2

site1:
 image: nginx:latest
 container_name: site1
 volumes:
  - './site1:/usr/share/nginx/html'
 ports:
  - 8080:80

site2:
 image: nginx:latest
 container_name: site2
 volumes:
  - './site2:/usr/share/nginx/html'
 ports:
  - 8081:80
```

Figure 16 : docker-compose.yml

Le premier bloc correspond au conteneur “proxy”. On le build à partir du fichier “Dockerfile” se trouvant dans le dossier proxy. Ce conteneur sera lié au conteneur “site1” et “site2” et sera lancé sur le port “80”.

Le conteneur “site1” sera lancé à partir de l’image nginx et sera monté à partir du dossier “site1”. Il sera lancé sur le port 8080.

Le conteneur “site2” sera lancé à partir de l’image nginx et sera monté à partir du dossier “site2”. Il sera lancé sur le port 8081.

Question 03

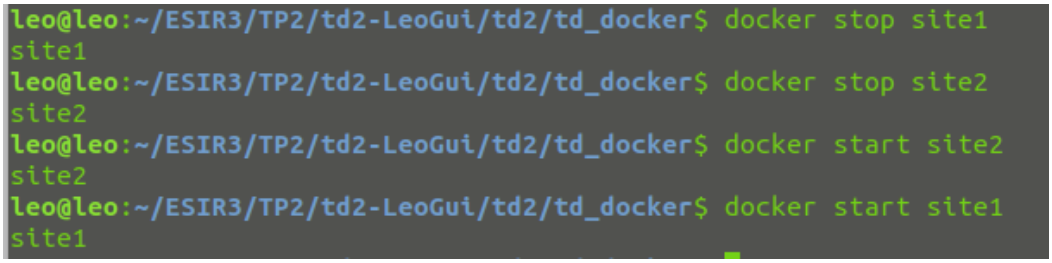
Figure 17 : docker -compose up

Lorsqu'on lance la commande "docker-compose up --build", le build se fait correctement.

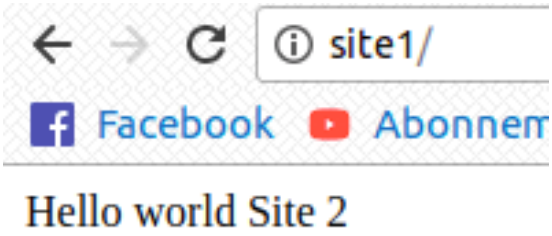
Ensuite on a la ligne "attaching to site2,site1,proxy". Il suffit ensuite sur le navigateur de taper "http://site1" ou "http://site2" et nous avons bien accès à l'index.html du site. Comme on peut le voir dans l'invite de commande, les opérations se font correctement.

Question 04

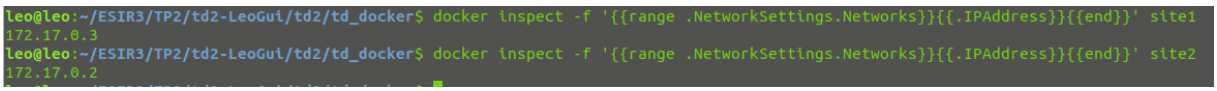
On effectue les commandes suivantes :



Ensuite lorsque l'on tape sur internet "http://site1", on obtient l'index.html du site 2



Pour comprendre pourquoi ce fonctionnement, j'ai regardé les deux adresses IP des conteneurs :



Comme on peut le voir, elles ont été inversées. C'est donc pour cela que les index.html sont inversés.

Conclusion

Dans ce TP sur Docker, nous avons donc réussi à héberger chaque site dans un conteneur docker différents.