## **Installation de Docker : Réf syllabus section 6.8.1**

|  |
| --- |
| # désinstaller les anciennes versions  apt-get remove docker docker-engine docker.io  # ajouter le dépôt Docker et le support HTTPS pour le dépôt  apt-get update  apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common  # ajouter la clé du dépôt  curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -  # finalement ajout du dépôt  add-apt-repository "deb https://download.docker.com/linux/debian $(lsb\_release -cs) stable"  # installation  apt-get update  apt-get install docker-ce  #vérification de l’installation  docker run hello-world |

ATTENTION aux espaces lors du copier-coller.

## **Déploiement d’un site HTML 🡺 1 conteneur**

1. Création d’un **Dockerfile.**
2. Il faut dans un premier temps rédiger un **Dockerfile** avec un éditeur sous windows :

|  |
| --- |
| FROM debian  MAINTAINER Shayan Amini  RUN apt-get update && apt-get upgrade -y  RUN apt-get install apache2 -y  COPY siteHTML.conf /etc/apache2/sites-available/000-default.conf  COPY ./siteHTML /var/www/siteHTML  #RUN a2ensite siteHTML  #RUN echo "127.0.0.1 siteHTML" >> /etc/hosts  EXPOSE 80  CMD apachectl -D FOREGROUND  # apache fonctionne en background -> container exited  #CMD service apache2 restart |

COPY siteHTML.conf /etc/apache2/sites-available/000-default.conf permet de copier notre vHost dans le vHost par défaut et donc plus besoin d’activer à chaque fois ceci.

COPY ./siteHTML /var/www/siteHTML permet de copier le contenu du siteHTML dans le répertoire connu par Apache.

1. Création d’un **VirtualHost. 🡺** siteHTML.conf

|  |
| --- |
| <VirtualHost \*:80>  # pas de servername -> site par défaut  #ServerName siteHTML  ServerAdmin webmaster@localhost  DocumentRoot /var/www/siteHTML  ErrorLog ${APACHE\_LOG\_DIR}/siteHTML\_errors.log  CustomLog ${APACHE\_LOG\_DIR}/siteHTML\_access.log combined  <Directory /var/www/siteHTML>  Require all granted  AllowOverride all  </Directory>  </VirtualHost> |

1. Il faut à présent transférer les fichiers édités sur le serveur Debian, pour cela on utilise le logiciel **WinScp** étant un logiciel de transfert de fichiers utilisant le *SSH* et le protocole *FTP.*
2. Créer un répertoire nommé *Docker* dans */home/shayan/* qui contiendra tous les sites déployé.

cd /home/shayan

mkdir Docker

1. Une image contenant capture d’écran

   Description générée automatiquementSe connecter avec un compte autre que celui de root (Ex : shayan) et glisser-déposer tous les fichiers nécessaires dans *Docker.*

L’arborescence ressemblerait à celle-ci :

/home/shayan

Docker

siteHTMLDocker

siteHTML

Dockerfile

siteHTML.conf

1. Création d’une image :

|  |
| --- |
| docker build -t imagesitehtml . |

-t 🡪 nom qu’on donne à l’image

4. Création d’un conteneur à partir de l’image

|  |
| --- |
| docker run -d -p 8000:80 --name contsitehtml imagesitehtml |

-p : port forwarding

--name : donner un nom au conteneur

**Commandes utiles**

Lister les images : docker images

Lister les conteneurs : docker container ls

\*\*\*Il est utile de nettoyer de temps en temps les images surtout lorsqu’il y a des erreur du type

« image existante » :

|  |
| --- |
| # pour y voir plus clair, il est utile de  # stopper et supprimer tous les conteneurs  docker stop $(docker ps -aq)  docker rm $(docker ps -aq) |

Finalement, on test le déploiement avec le navigateur lynx : lynx localhost:8000

## **Déploiement du site PHP 🡺 2 conteneurs**

Pour ce deuxième exercice, nous allons nous rapprocher d’une architecture micro-service. L’idée est de déployer le site des bonnes nouvelles(site PHP-MySQL) via 2 conteneurs. Le premier conteneur contiendra le serveur Web (apache) et le code source tandis que le deuxième conteneur contiendra la base de données. Nous allons également tirer profit du Docker Hub en utilisant des images déjà optimisées et prêtes à l’emploi

L’arborescence devrait ressembler à :

/home/shayan/Docker

◦ sitePHPDocker

▪ contsitephp -> Premier conteneur (serveur web et le code source)

• Dockerfile

• siteBonneNouvelles (site php disponible sur Moodle)

• sitePHP.conf (vhost disponible sur Moodle)

Et aussi Db.class.php (dispo dans siteBonneNouvelles/models) qui doit être modifié

▪ contdbphp -> Deuxième conteneur (Base de données)

• Dockerfile

• bdbn.sql (fichier de création de la DB disponible sur Moodle)

## **Premier conteneur 🡺 Serveur web et le code source**

1. Comme toujours, création d’un **Dockerfile**.
2. Rédaction d’un Dockerfile sous windows avec un éditeur :

|  |
| --- |
| FROM php:7.2.8-apache-stretch  RUN docker-php-ext-install pdo\_mysql  MAINTAINER Olivier Choquet  EXPOSE 80  COPY sitePHP.conf /etc/apache2/sites-available/000-default.conf  COPY siteBonneNouvelles /var/www/sitePHP  # modifier éventuellement le nom du conteneur et mdp de passe dans Db.class.php  # docker utilisera au niveau réseau le nom du conteneur pour atteindre le conteneur de la DB  COPY Db.class.php /var/www/sitePHP/models/Db.class.php  CMD apache2ctl -D FOREGROUND |

Modifier le fichier Db.class.php dans le dossier Models pour y remplacer l'ancienne ligne avec cette ligne:

*$this->\_db = new PDO('mysql:host=****contsitephpdb****;dbname=bdbn', 'root',* ***'ipl'****);*

Ensuite copier ce fichier et le mettre à l'endroit où DockerFile du dossier **contsitephp** se situe.

2. Création d’un **VirtualHost.** sitePHP.conf

|  |
| --- |
| <VirtualHost \*:80>  # pas de servername -> site par défaut  #ServerName sitePHP  ServerAdmin webmaster@localhost  DocumentRoot /var/www/sitePHP  ErrorLog ${APACHE\_LOG\_DIR}/sitePHP\_errors.log  CustomLog ${APACHE\_LOG\_DIR}/sitePHP\_access.log combined  <Directory /var/www/sitePHP>  Require all granted  AllowOverride all  </Directory>  </VirtualHost> |

3. Transférer ses fichiers avec un compte, autre que celui de root (Ex :shayan)

## **Deuxième conteneur 🡺 La base de données**

1. Création d’un **Dockerfile**.

|  |
| --- |
| FROM mysql  MAINTAINER Olivier Choquet  ENV MYSQL\_DATABASE=bdbn  ENV MYSQL\_ROOT\_PASSWORD=ipl  COPY bdbn.sql /docker-entrypoint-initdb.d |

Transférer ses fichiers avec un compte, autre que celui de root (Ex :shayan)

## Exécution

Se placer dans le répertoire sitePHPDocker :

création image site php code 🡺 docker build -t imgsitephp contsitephp/ .

création image site php db  🡺 docker build -t imgsitephpdb contdbphp/ .

création du conteneur avec la DB 🡺 docker run -d --name contsitephpdb imgsitephpdb

création du conteneur sitephp 🡺 docker run -d -p 8001:80 --name contsitephp imgsitephp

A ce stade lynx localhost affiche le site php mais erreur dans le menu "Livres" car connexion à la DB

# Erreur : Name or service not know -> erreur réseau

### **Connexion des conteneurs**

Il faut connecter les conteneurs -> les placer dans un même réseau :

création de notre réseau 🡺 docker network create sitephpnet

connecter le premier conteneur 🡺 docker network connect sitephpnet contsitephp

connecter le deuxième conteneur 🡺 docker network connect sitephpnet contsitephpdb

**Commandes utiles**

Lister les images : docker images

Lister les conteneurs : docker container ls

\*\*\*Il est utile de nettoyer de temps en temps les images surtout lorsqu’il y a des erreur du type

« image existante » :

|  |
| --- |
| # pour y voir plus clair, il est utile de  # stopper et supprimer tous les conteneurs  docker stop $(docker ps -aq)  docker rm $(docker ps -aq) |

$docker system prune –a => supprime tout (images, contenaires, réseaux inutilisés)