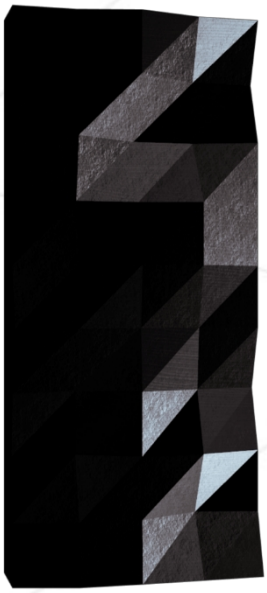


Inception

Résumé : *Ce document est un exercice relatif à l'administration du système.*

Version : 2

Contenu

Chapitre I

Préambule

Chapitre II

Introduction

Ce projet vise à élargir vos connaissances en matière d'administration système en utilisant Docker. Vous allez virtualiser plusieurs images Docker, en les créant dans votre nouvelle machine virtuelle personnelle.

Chapitre III

Orientations générales

- Ce projet doit être réalisé sur une machine virtuelle.
- Tous les fichiers nécessaires à la configuration de votre projet doivent être placés dans un répertoire srcs
  dossier.
- Un Makefile est également nécessaire et doit être situé à la racine de votre répertoire. Il doit configurer l'ensemble de votre application (c'est-à-dire qu'il doit construire les images Docker à l'aide de docker-compose.yml).
- Ce sujet nécessite la mise en pratique de concepts que vous n'avez peut-être pas encore appris, en fonction de votre niveau d'expérience. Par conséquent, nous vous conseillons de ne pas hésiter à lire beaucoup de documentation relative à l'utilisation de Docker, ainsi que tout ce qui vous sera utile pour réaliser ce travail.

Chapitre IV Partie obligatoire

Ce projet consiste à mettre en place une petite infrastructure composée de différents services selon des règles spécifiques. L'ensemble du projet doit être réalisé dans une machine virtuelle. Vous devez utiliser docker compose.

Chaque image Docker doit porter le même nom que le service correspondant.
Chaque service doit être exécuté dans un conteneur dédié.
Pour des questions de performance, les conteneurs doivent être construits à partir de l'avant-dernière version stable d'Alpine ou de Debian. Le choix vous appartient.
Vous devez également écrire vos propres Dockerfiles, un par service. Les Dockerfiles doivent être appelés dans votre docker-compose.yml par votre Makefile.
Cela signifie que vous devez construire vous-même les images Docker de votre projet. Il est ensuite interdit de tirer des images Docker prêtes à l'emploi, ainsi que d'utiliser des services tels que DockerHub (Alpine/Debian étant exclu de cette règle).

Vous devez ensuite l'installer :

- Un conteneur Docker qui contient NGINX avec TLSv1.2 ou TLSv1.3 uniquement.
- Un conteneur Docker qui contient WordPress + php-fpm (il doit être installé et configuré) uniquement sans nginx.
- Un conteneur Docker qui contient uniquement MariaDB sans nginx.
- Un volume qui contient votre base de données WordPress.
- Un deuxième volume qui contient les fichiers de votre site web WordPress.
- Un réseau docker qui établit la connexion entre vos conteneurs.

Vos conteneurs doivent redémarrer en cas de panne.

Un conteneur Docker n'est pas une machine virtuelle. Par conséquent, il n'est pas recommandé d'utiliser un correctif pirate basé sur "tail -f" et ainsi de suite lorsque vous essayez de l'exécuter. Découvrez comment fonctionnent les démons et si c'est une bonne idée de les utiliser ou non.

Bien entendu, l'utilisation de network : host ou --link ou links : est interdite. La ligne network doit être présente dans votre fichier docker-compose.yml. Vos conteneurs ne doivent pas être démarrés avec une commande exécutant une boucle infinie. Ceci s'applique donc également à toute commande utilisée comme point d'entrée, ou utilisée dans des scripts de point d'entrée. Voici quelques patches interdits : tail -f, bash, sleep infinity, while true.

En savoir plus sur le PID 1 et les meilleures pratiques pour écrire des Dockerfiles.

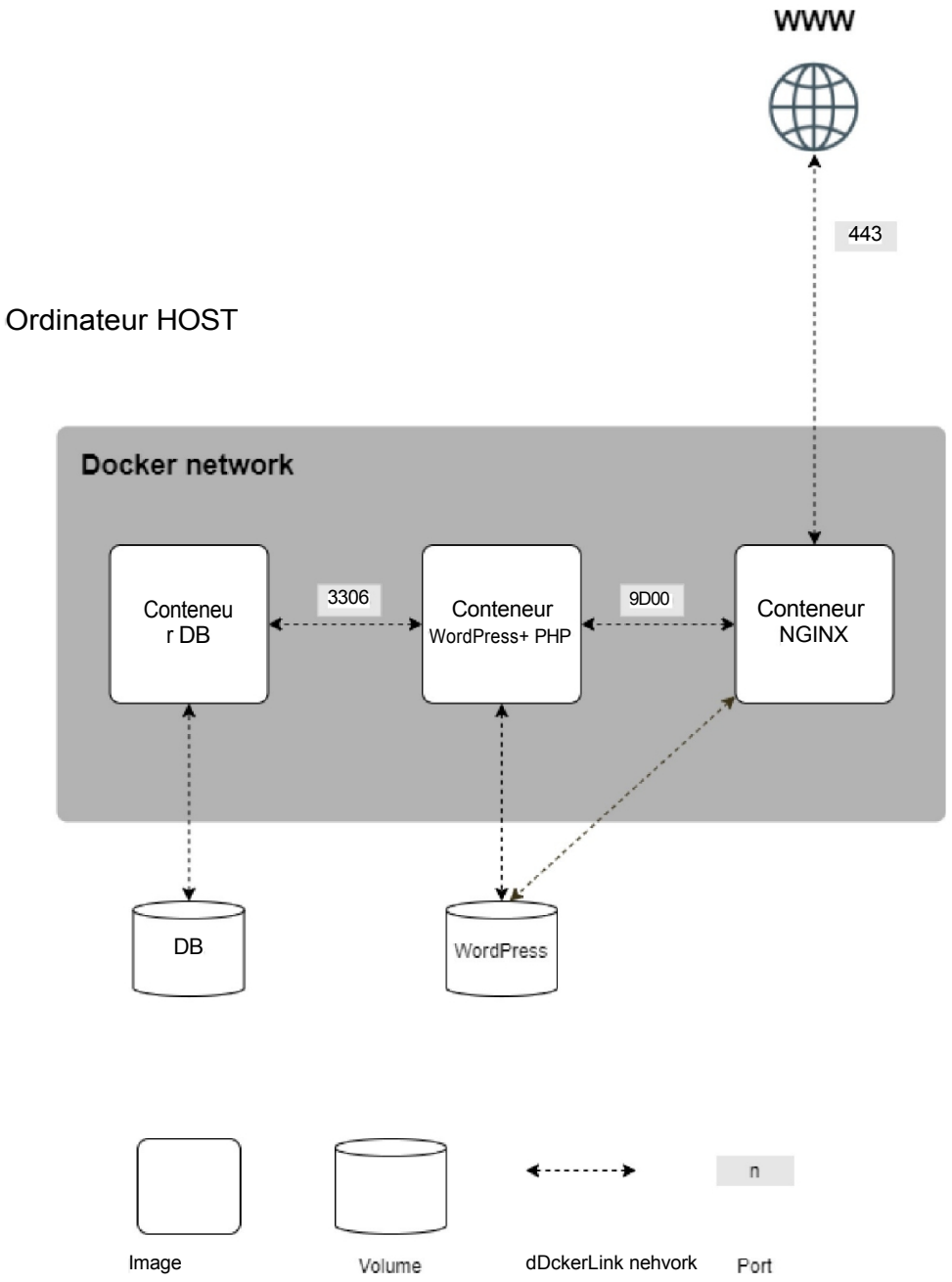
- Dans votre base de données WordPress, il doit y avoir deux utilisateurs, l'un d'eux étant l'administrateur. Le nom d'utilisateur de l'administrateur ne peut pas contenir admin/Admin ou admin- istrateur/Administrator (par exemple, admin, administrateur, Administrator, admin-123, etc.)

Vos volumes seront disponibles dans le dossier /home/login/data de la machine hôte utilisant Docker. Bien entendu, vous devez remplacer le login par le vôtre.

Pour simplifier les choses, vous devez configurer votre nom de domaine de manière à ce qu'il pointe vers votre adresse IP locale.
Ce nom de domaine doit être login.42.fr. Là encore, vous devez utiliser votre propre login. Par exemple, si votre login est wil, wil.42.fr redirigera vers l'adresse IP pointant vers le site web de wil.

La dernière balise est interdite.
Aucun mot de passe ne doit être présent dans vos Dockerfiles. Il est obligatoire d'utiliser des variables d'environnement.
Il est également fortement recommandé d'utiliser un fichier .env pour stocker les variables d'environnement. Le fichier .env doit être situé à la racine du répertoire srcs.
Votre conteneur NGINX doit être le seul point d'entrée dans votre infrastructure via le port 443 uniquement, en utilisant le protocole TLSv1.2 ou TLSv1.3.

Voici un exemple de diagramme du résultat attendu :



Vous trouverez ci-dessous un exemple de la structure de répertoire attendue :

```
$> ls -alR
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxrwt 17 wil wil 4096 avril 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Makefile
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 srcs

./srcs :
total XX
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 ...
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 docker-compose.yml
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .env
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 exigences

./srcs/requirements :
total XX
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 3 wil wil 4096 avril 42 20:42 ..
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 bonus
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 mariadb
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 nginx
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 outils
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 wordpress

./srcs/requirements/mariadb :
total XX
drwxrwxr-x 4 wil wil 4096 avril 42 20:45 .
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 ...
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 conf
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 tools
[...]
./srcs/requirements/nginx :
total XX
drwxrwxr-x 4 wil wil 4096 avril 42 20:42 .
drwxrwxr-x 5 wil wil 4096 avril 42 20:42 ...
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 conf
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
-rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil wil 4096 avril 42 20:42 tools
[...]

$> cat srcs/.env
DOMAIN_NAME=wil.42.fr
# certificats
CERTS_=./XXXXXXXXXXX
# MYSQL SETUP
MYSQL_ROOT_PASSWORD=XXXXXXXXXXX
MYSQL_USER=XXXXXXXXXXX
MYSQL_PASSWORD=XXXXXXXXXXX
[...]
$>
```

Pour des raisons évidentes de sécurité, toutes les informations d'identification, clés API, variables env, etc... doivent être sauvegardées localement dans un fichier .env et ignorées par git. Les informations d'identification stockées publiquement vous conduiront directement à l'échec du projet.

Chapitre V

Partie bonus

Pour ce projet, la partie bonus se veut simple.

Un fichier Dockerfile doit être écrit pour chaque service supplémentaire. Ainsi, chacun d'entre eux s'exécutera dans son propre conteneur et disposera, si nécessaire, de son volume dédié.

Liste de bonus :

- Configurez le cache redis pour votre site WordPress afin de gérer correctement le cache.
- Mettez en place un serveur FTP pointant vers le volume de votre site WordPress.
- Créez un site web statique simple dans le langage de votre choix sauf PHP (Oui, PHP est exclu !). Par exemple, un site vitrine ou un site de présentation de votre CV.
- Configurer Adminer.
- Mettez en place un service de votre choix que vous jugez utile. Lors de la soutenance, vous devrez justifier votre choix.

Pour compléter la partie bonus, vous avez la possibilité de mettre en place des services supplémentaires. Dans ce cas, vous pouvez ouvrir plus de ports en fonction de vos besoins.

La partie bonus ne sera évaluée que si la partie obligatoire est PARFAITE. Parfait signifie que la partie obligatoire a été intégralement réalisée et qu'elle fonctionne sans dysfonctionnement. Si vous n'avez pas satisfait à TOUTES les exigences obligatoires, votre partie bonus ne sera pas évaluée du tout.

Chapitre VI

Soumission et évaluation par les pairs

Rendez votre travail dans votre dépôt Git comme d'habitude. Seul le travail contenu dans votre dépôt sera évalué lors de la soutenance. N'hésitez pas à vérifier les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.

```
16D85ACC441674FBA2DF65190663EC3C3C258FEA065D090A715F1B62F5A57F0B75403
61668BD6823E2F873124B7E59B5CE94BB7ABD71CD01F65B959E14A3838E414F1E871
F7D91730B
```