

# Inception - Document traduit en français

## Chapitre I

### Preamble

## Chapitre II

### Introduction

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

## Chapitre III

### Directives 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 dossier `srcs`.
- Un `Makefile` est également requis et doit être situé à la racine de votre répertoire. Il doit configurer votre application entière (c'est-à-dire qu'il doit construire les images Docker en utilisant `docker-compose.yml`).
- Ce sujet nécessite de mettre en pratique des concepts que, selon votre formation, vous n'avez peut-être pas encore appris. Par conséquent, nous vous conseillons de lire beaucoup de documentation relative à l'utilisation de Docker, ainsi que tout ce que vous trouverez utile pour accomplir cette tâche.

## Chapitre IV

### Partie obligatoire

Ce projet consiste à configurer une petite infrastructure composée de différents services selon des règles spécifiques.

Le projet entier doit être réalisé dans une machine virtuelle. Vous devez utiliser `docker compose`.

Chaque image Docker doit avoir le même nom que son service correspondant. Chaque service doit fonctionner dans un conteneur dédié. Pour des raisons de performance, les conteneurs doivent être construits soit à partir de la version avant-dernière 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-meme les images Docker de votre projet. Il est donc interdit de tirer des images Docker prefabriquees, ainsi que d'utiliser des services tels que DockerHub (Alpine/Debian etant exclus de cette regle).

Vous devez ensuite configurer :

- Un conteneur Docker contenant NGINX avec TLSv1.2 ou TLSv1.3 uniquement.
- Un conteneur Docker contenant WordPress + php-fpm (il doit etre installe et configure) sans nginx.
- Un conteneur Docker contenant MariaDB sans nginx.
- Un volume contenant votre base de donnees WordPress.
- Un second volume contenant les fichiers de votre site WordPress.
- Un reseau Docker etablissant la connexion entre vos conteneurs.

Vos conteneurs doivent redemarrer en cas de crash. Un conteneur Docker n'est pas une machine virtuelle. Ainsi, il n'est pas recommande d'utiliser des astuces comme 'tail -f' et ainsi de suite pour le faire fonctionner. Lisez sur le fonctionnement des demons et s'il est judicieux de les utiliser ou non.

Bien entendu, l'utilisation de network: host ou --link ou links: est interdite. La ligne network doit etre presente dans votre fichier docker-compose.yml. Vos conteneurs ne doivent pas etre demarres avec une commande executant une boucle infinie. Cela s'applique egalement a toute commande utilisee comme point d'entree ou utilisee dans des scripts de point d'entree. Les patchs hacky suivants sont interdits : tail -f, bash, sleep infinity, while true.

Lisez sur PID 1 et les meilleures pratiques pour ecrire des Dockerfiles.

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

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

Pour simplifier les choses, vous devez configurer votre nom de domaine afin qu'il pointe vers votre adresse IP locale. Ce nom de domaine doit etre login.42.fr. Encore une fois, 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.

Le dernier tag est interdit. Aucun mot de passe ne doit etre present dans vos Dockerfiles. Il est obligatoire d'utiliser des

variables d'environnement. Il est également fortement recommande d'utiliser un fichier .env pour stocker les variables d'environnement. Le fichier .env doit etre situe a la racine du repertoire srcs.

Votre conteneur NGINX doit etre le seul point d'entree dans votre infrastructure via le port 443 uniquement, en utilisant le protocole TLSv1.2 ou TLSv1.3.

Voici un exemple de diagramme du resultat attendu :

(exemple de structure de repertoire)

Pour des raisons evidentes de securite, tous les identifiants, cle API, variables d'environnement, etc. doivent etre sauvegardes localement dans un fichier .env et ignores par git. Des identifiants stockes publiquement entraineront directement l'echec du projet.

## Chapitre V

### Partie bonus

Pour ce projet, la partie bonus est concue pour etre simple.

Un Dockerfile doit etre ecrit pour chaque service supplementaire. Ainsi, chacun d'entre eux fonctionnera dans son propre conteneur et aura, si necessaire, son volume dedie.

Liste des bonus :

- Configurer un cache redis pour votre site WordPress afin de gerer correctement le cache.
- Configurer un conteneur de serveur FTP pointant vers le volume de votre site WordPress.
- Creer un site web statique simple dans la langue de votre choix, sauf PHP (oui, PHP est exclu !). Par exemple, un site vitrine ou un site de presentation de votre CV.
- Configurer Adminer.
- Configurer un service de votre choix que vous trouvez utile. Pendant la defense, vous devrez justifier votre choix.

Pour completer la partie bonus, vous avez la possibilite de configurer des services supplementaires. Dans ce cas, vous pouvez ouvrir plus de ports pour repondre a vos besoins.

La partie bonus ne sera evaluee que si la partie obligatoire est parfaite. Parfait signifie que la partie obligatoire a ete

integralement realisee et fonctionne sans dysfonctionnement. Si vous n'avez pas rempli TOUTES les exigences obligatoires, votre partie bonus ne sera pas evaluee du tout.

## Chapitre VI

### Soumission et evaluation par les pairs

Rendez votre projet dans votre depot Git comme d'habitude. Seul le travail a l'interieur de votre depot sera evalue pendant la defense. N'hésitez pas a verifier deux fois les noms de vos dossiers et fichiers pour vous assurer qu'ils sont corrects.