

Inception

Résumé :

Ce document est un exercice lié à l'administration système.

Version : 2.1

Sommaire

I. Préambule 2

II. Introduction 3

III. Directives générales 4

IV. Partie obligatoire 5

V. Partie bonus 9

VI. Soumission et évaluation par les pairs 10

Chapitre I

Préambule

Chapitre II

Introduction

Ce projet vise à élargir vos connaissances en administration système en utilisant Docker.

Vous allez virtualiser plusieurs images Docker, 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 l'intégralité de votre application (c'est-à-dire, il doit construire les images Docker en utilisant docker-compose.yml).

Ce sujet nécessite la mise en pratique de concepts que, selon votre parcours, vous n'avez peut-être pas encore appris. Par conséquent, nous vous conseillons de ne pas hésiter à lire beaucoup de documentation liée à l'utilisation de Docker, ainsi que tout ce que vous trouverez utile pour réaliser cette mission.

## Chapitre IV

### Partie obligatoire

Ce projet consiste à mettre en place une petite infrastructure composée de différents services sous 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 son service correspondant.

Chaque service doit s'exécuter dans un conteneur dédié.

Pour des raisons de performance, les conteneurs doivent être construits soit à partir de la version stable avant-dernière 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 donc interdit de récupérer des images Docker toutes faites, ainsi que d'utiliser des services tels que DockerHub (Alpine/Debian étant exclus de cette règle).

Vous devez ensuite mettre en place :

Un conteneur Docker contenant NGINX avec uniquement TLSv1.2 ou TLSv1.3.

Un conteneur Docker contenant uniquement WordPress + php-fpm (il doit être installé et configuré)

sans nginx.

Un conteneur Docker contenant uniquement MariaDB sans nginx.

Un volume contenant votre base de données WordPress.

Un deuxième volume contenant les fichiers de votre site WordPress.

Un réseau Docker établissant la connexion entre vos conteneurs.

Vos conteneurs doivent redémarrer en cas de crash.

Un conteneur Docker n'est pas une machine virtuelle. Il n'est donc pas recommandé d'utiliser des solutions de contournement basées sur `tail -f` et autres pour essayer de le faire fonctionner.

Renseignez-vous sur le fonctionnement des démons et sur l'opportunité de les utiliser ou non.

Bien entendu, l'utilisation des options réseau : `host` ou `--link` ou `links` est interdite. La ligne réseau doit être présente dans votre fichier `docker-compose.yml`.

Vos conteneurs ne doivent pas être démarrés avec une commande exécutant une boucle infinie. Cela s'applique également à toute commande utilisée comme point d'entrée, ou dans les scripts de point d'entrée.

Voici quelques solutions de contournement interdites : `tail -f`, `bash`, `sleep infinity`, `while true`.

Renseignez-vous sur PID 1 et les bonnes 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 doit pas contenir `admin/Admin` ou `administrator/Administrator` (par exemple, `admin`, `administrator`, `Admin`, `admin-123`, etc.).

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

Pour simplifier les choses, vous devez configurer votre nom de domaine pour qu'il pointe vers votre adresse IP locale. Ce nom de domaine doit être `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 de `wil`.

L'utilisation du dernier tag 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.

## Chapitre V

### Partie bonus

Pour ce projet, la partie bonus se veut simple.

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

Liste des bonus :

Mettre en place un cache redis pour votre site WordPress afin de gérer correctement le cache.

Mettre en place un conteneur serveur FTP pointant vers le volume de votre site WordPress.

Créer 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 pour présenter votre CV.

Mettre en place Adminer.

Mettre en place un service de votre choix que vous jugez utile. Pendant 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 pour répondre à 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 fonctionne sans dysfonctionnement. Si vous n'avez pas rempli TOUS les critères obligatoires, votre partie bonus ne sera pas évaluée du tout.

## Chapitre VI

## Soumission et évaluation par les pairs

Remettez votre devoir dans votre dépôt Git comme d'habitude. Seul le travail à l'intérieur de 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.