II. b) Docker Compose

Objectif

L'objectif de cet exercice est de revenir sur le déployement de plusieurs container d'un seul coup à l'aide de Docker Compose.

Taches à réaliser

Lors de cette exercice, nous avons pour objectif de décrire une stack applicative web à l'aide d'un docker-compose.yml. Cette stack devra comprendre à minima :

- Un site web frontend de votre choix
- Une backend ou API appelé par le site web
- Une base de donnée appelée par l'API
- Un reverse proxy seul point d'entrée du network

1. Frontend

Cela sera surement la base de votre stack, il éxiste de nombreuse application open-source fournissant un moyen d'installation Docker par docker compose, soit créatifs, herbégez une application de votre choix.

Voici deux liens utiles pour rechercher des applications à self-host :

AlternativeTo

Vous permet de chercher des alternatives à vos sites ou logiciels préférés. Pensez à rechercher une solution open-source et self-host.

awesome-selfhosted

-> version web

Awesome-Selfhosted est une liste de solutions gratuites et self-hosted.

Il y a de nombreuses catégories, n'hésitez pas à passez du temps à rechercher ce qui vous intéresse.

Pensez à chercher la mention "Docker" pour ne pas avoir de mauvaise surprise (sauf si vous désirez refaire toute la partie Dockerfile pour un projet open-source ...)

2. Backend

Le backend de votre solution va surement dépendre de votre frontend de choix.

3. Database

Pour la base de données, suivant la solution frontend choisi, il est parfois possible de choisir quel moteur de base de données utiliser, n'hésitez pas à choisir celle avec laquelle vous avez le plus d'habitude.

4. Reverse Proxy

Le reverse proxy permet de recevoir l'ensemble des appels à un nom de domain donné, `localhost` dans notre cas.

Lors de cet exercice, je ne veux voir que les ports du reverse proxy ouvert. L'ensemble des communications entre frontend, le backend et l'API doivent passer par le network docker ou par le reverse proxy.

L'implementation d'un reverse proxy est primordiale lors du déploiements de nombreuses applications sur le même serveur host et permet donc la distinction des services par sous-domaines par exemple.

Pour le reverse proxy, voir la documentation nginx par exemple.

Revision #4
Created 26 October 2023 14:31:31 by Victor BADIN-LIMOU

Updated 27 October 2023 09:50:01 by Victor BADIN-LIMOU