**TP DOCKERCOMPOSE**

Dans ce TP nous allons voir comment configurer le déploiement de nos deux services, Nginx et Bind, en un seul fichier. Cela a pour but de faciliter le déploiement de votre solution. Egalement le fait d'utiliser un fichier de configuration à appliquer au lieux de démarrer vos containers en ligne de commande vous permettra de plus facilement garder une trace de vos manipulations afin de débugger plus facilement si quelque chose se passe mal ou alors afin de modifier facilement votre solution.

**I/Mise en place**

Pour faire ce TP vous devez avoir votre arborescence complète pour les deux services :

* /home/<votre\_utilisateur>/nginx/code/intra
* /home/<votre\_utilisateur>/nginx/code/wiki
* /home/<votre\_utilisateur>/nginx/conf
* /home/<votre\_utilisateur>/bind/data

Le but de votre compose est de déployer une solution clé en main qui fonctionne directement. Il vous faut donc une configuration opérationnelle des container nginx et bind. Récupérez la totalité du contenu du dossier de configuration de votre dns et placez la dans le dossier /home/<votre\_utilisateur>/bind/data (vous trouverez l'emplacement du dossier de configuration dans les explication du dépôt github de la solution ☺)

**II/Rédaction du compose**

Dans votre répertoire personnel créez le fichier docker-compose.yml. Ce fichier va décrire le démarrage de tous les services composants notre solution. Pour chaque service nous allons indiquer le nom du service, le nom de l'image à utiliser, le nom du container à générer ainsi que les volumes et les ports à attribuer. En l'occurrence dans notre cas il faudra attribuer, pour le service web, les volumes nginx/code/intra et nginx/code/wiki au répertoires /intra et /wiki du container ( /!\ le fichier attend des chemins relatifs à son propre emplacement pour les chemins sources). Il faudra également attribuer les ports 80 et 8080 de l'image a des ports de la machine physique. Pour le service DNS il faudra attribuer le répertoire bind/data au répertoire contenant les données de bind dans le container. Il faudra également attribuer les différents ports du service.

Avec le fichier docker-compose.yml correctement rédigé vous pouvez démarrer votre solution avec la commande docker compose up (en étant dans le dossier où se trouve le fichier docker-compose.yml). Comme pour le docker run, l'option -d vous permet de détacher la solution de votre terminal.

La commande docker compose ps permet d'afficher les solutions. Si vous modifiez l'un des services de votre fichier docker-compose.yml et que vous refaites un docker compose up -d docker va simplement redéployer le service que vous avez modifié sans stopper les autres services de la solution.