

Admin Infra : Exercices

1 Préliminaires

Nous utiliserons la machine Debian sur Azure pour ces exercices.

Lorsqu'un mot de passe est demandé, nous vous conseillons de mettre « azerty1.» par facilité.

Pour ce cours, Nous réutiliserons le serveur Linux déjà installé.

2 Exercice 1 : Manipuler Docker-compose

1. Consulter le syllabus à la section 10.10 docker-compose
2. Installer docker-compose sur votre machine Linux
3. Créer une image Docker avec Apache et le site HTML (HTTP) via un docker-compose.yml
 1. Réutilisez ce que vous avez fait (Dockerfile) à la séance précédente
 2. Les solutions de la séance précédente sont disponibles sur Moodle si vous n'avez pas vos solutions
 3. Au niveau de la création des conteneurs, vous ferez un mapping de port (port forwarding) entre votre machine hôte(8091) et le conteneur (80). Ainsi le site HTML sera accessible depuis votre machine hôte sur le port 8091
4. Tester votre solution : `lynx localhost:8091`

2.1 Indications utiles

Il n'est pas nécessaire d'utiliser les volumes pour cet exercice.

Attention au format YAML qui nécessite une indentation homogène via des espaces (pas de tabulation!).

Il est utile de supprimer les conteneurs et images problématiques pour plus de lisibilité.

3 Exercice 2 : 3 conteneurs

Pour ce deuxième exercice, nous allons créer une architecture micro-service. L'idée est de déployer le site des bonnes nouvelles(site PHP-MySQL) via 3 conteneurs. Le premier conteneur

contiendra la base de données(mariadb:10.4), le deuxième conteneur contiendra le serveur Web (apache) et le code source et le troisième conteneur phpmyadmin (phpmyadmin:4.8). Nous allons également tirer profit du Docker Hub en utilisant des images déjà optimisées et prêtes à l'emploi.

1. Le premier conteneur sera créé à partir de l'image mariadb:10.4 présent sur le DockerHub. Regardez la documentation sur le DockerHub pour définir le mot de passe root ainsi que la base de données bdbn. Ceci doit être cohérent avec le fichier Db.class.php.
2. Le deuxième conteneur (apache) sera créé à partir du Dockerfile fait à la séance précédente. Il sera exposé sur le port 8092.
3. Le troisième conteneur sera créé à partir de l'image phpmyadmin/phpmyadmin:4.7 présent sur le DockerHub. Regardez la documentation sur le DockerHub pour savoir comment connecter celui-ci à mariadb. Il sera exposé sur le port 9002. Celui-ci permettra d'injecter le fichier bdbn.sql pour créer la base de données.
4. L'arborescence sera donc la suivante :
 1. sitePHPDockerCompose
 - Dockerfile (Création du conteneur apache)
 - bdbn.sql (fichier de création de la DB disponible sur Moodle)
 - siteBonneNouvelles (site php disponible sur Moodle)
 - sitePHP.conf (vhost disponible sur Moodle)
 - docker-compose.yml
5. Tester votre solution sur votre machine physique
 1. Vous devez avoir accès phpmyadmin pour injecter/importer le fichier bdbn.sql
 2. Vous devez voir dans le navigateur le site des bonnes nouvelles hébergé dans votre conteneur.
 3. La page « les livres » doit être fonctionnelle si votre conteneur DB a été bien fait.

3.1 Indications utiles

Il est nécessaire d'utiliser les volumes pour cet exercice.