Activité : créer un environnement LAMP avec Docker-Compose

Présentation

Docker permet la création de conteneurs en proposant les fonctionnalités suivantes :

- création de **conteneur personnalisé** en utilisant un fichier **Dockerfile** ;
- **exposition** d'un service en choisisant le **port local** de l'ordinateur ;
- gestion de la **persistance des données** en gérant des **volumes** ;
- **liaison** des conteneurs entre eux quand une **architecture en microservices**, utilise un conteneur spécifique par service (service Web, service de base de donés, etc.)

Cependant :

- La création d'un conteneur nécessite alors **plusieurs paramèters** lors de son **lancement**.
- Il faut **lancer manuellement** les conteneurs et les **uns après les autres**.

> **Docker-compose** est donc un outil permettant de créer et de lancer des architectures avec plusieurs conteneurs en une seule commande.

L'architecture à lancer est décrite au préalable dans un seul fichier au format YAML qui contient toute la configuration nécessaire pour chacun des conteneurs (port exposé, volume à gérer, etc.)

Objectif

Réaliser le même environnement LAMP avec 3 conteneurs en utilisant Docker-compose :

- Conteneur Docker **php** (serveur Web) personnalisé pour gérer les accès en **php** à la base de données MariaDB ;
- Conteneur Docker **mariadb** (base de données) ;
- Conteneur Docker **phpmyadmin** (site Web d'administration de la base de données).

Les fichiers du site Web et de la base de données sont enregistrés en local en utilisant le principe des volumes de Docker.

Syntaxe du fichier docker-compose.yml

Un fichier docker-compose.yml doit être créé avec les instructions nécessaires dont voici celles qui sont à utiliser :

- Le mot-clé **service** : permet de définir les différents conteneurs qui seront activés.
- L'instruction **image** : permet de préciser l'imagfe à utiliser pour créer le conteneur
- L'instruction **depends_on** : permet de définir le lien avec un autre conteneur
- L'instruction **volume** : met en œuvre le mécanisme de persistance des données
- L'instruction **ports** : met de préciser le port local d'écoute du service
- L'instruction **environment** : permet de définir des variables d'environnement dans le conteneur

Commandes Docker et Docker-compose de création des trois conteneurs

Création du serveur de base de données MariaDB

- nom du conteneur : **servbdd**
- base de données en écoute sur le port par défaut : **3306** ;
- mot de passe du compte root de MariaDB précisé dans une variable d'environnement MARIADB*ROOT*PASSWORD : **MARIADB*ROOT*PASSWORD: passwordmariadb** ;
- dossier local pour la persistance des bases de données du conteneur : **c:\www\html\siteweb-bdd**

Voici les paramètres de la commande Docker et les équivalents à écrire dans le fichier Docker-compose.yml

Docker	Docker-compose.yml
`docker run -d --name servbdd -v c:\www\html\siteweb-bdd:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=passwordmariadb mariadb`	`services: servbdd: image: mariadb volumes: - c:\www\html\siteweb-bdd:/var/lib/mysql environment: MARIADB_ROOT_PASSWORD: passwordmariadb`

Création du conteneur phpmyadmin :

- nom du conteneur : **phpmyadmin**
- port local d'écoute pour accéder au site Web phpmyadmin : **8080** ;
- liaison avec le conteneur **servbdd** en renseignant le **nom du serveur MariaDB** dans la variable d'environnement **PMA*HOST : PMA_HOST: servbdd. Voici les paramètres de la commande Docker et les équivalents à écrire dans le fichier Docker-compose.yml ^ Docker ^ Docker-compose.yml ^ |<code>docker run -d –name phpmyadmin –link servbdd:servbdd -p 8080:80 -e PMAHOST=servbdd phpmyadmin</code>|<code> services: phpmyadmin: image: phpmyadmin dependson: - servbdd ports: - 8080:80 environment: PMA_HOST=servbdd </code>| ====Création du serveur Web avec les paramètres suivants :==== * nom du conteneur : servweb * image Docker personnalisée (basé sur l'image php:8.1-apache) : charles/php. * mappage du site Web sur le port local : 8001 ; * dossier local contenant les fichiers du site Web : c:\www\html\siteweb; * dossier local pour la persistance des logs du serveur Apache : c:\www\log ; Voici les paramètres de la commande Docker et les équivalents à écrire dans le fichier Docker-compose.yml ^ Docker ^ Docker-compose.yml ^ |<code>docker run -d –name servweb –link servbdd:servbdd -p 8001:80 -v c:\www\html\siteweb:/var/www/html -v c:\www\log:/var/log/apache2 charles/php</code>|<code> services: servweb: image: charles/php depends_on: - servbdd ports: - 8001:80 volumes: - c:\www\html\siteweb:/var/www/html - c:\www\log:/var/log/apache2 </code>| ===== Le fichier docker-compose.yml complet de l'architecture LAMP===== <code file docker-compose.yml> version: '3' services: # base de données MariaDB servbdd: image: mariadb volumes: - c:\www\html\nolark-bdd:/var/lib/mysql environment: MARIADBROOTPASSWORD: passwordmariadb # site Web phpmyadmin phpmyadmin: image: phpmyadmin dependson: - servbdd ports: - 8080:80 environment: PMAHOST: servbdd # site Web servweb: image: charles/php dependson: - servbdd ports: - 8001:80 volumes: - c:\www\html\nolark:/var/www/html - c:\www\html\nolarklogs:/var/log/apache2 </code> ===== Exécution des commandes du fichier docker-compose.yml===== La commande docker-compose doit être lancée dans le dossier où se trouve le fichier docker-compose.yml. ==== Lancement de tous les conteneurs avec Docker-compose==== <code> docker-composer up -d </code> Résultat de la commande : <code> PS C:\www\html> docker-compose up -d [+] Running 4/4 - Network html_default Created 0.1s - Container html-servbdd-1 Started 1.0s - Container html-phpmyadmin-1 Started 1.8s - Container html-servweb-1 Started 1.9s </code> ==== Arrêt de tous les conteneurs avec Docker-compose==== <code> docker-composer down </code> ===== Activité à faire =====***

  Mission :

  - Créer un fichier docker-compose.yml qui permet de créer un site Wordpress avec la base de données **MariaDB**.

  Voici les variables d'environnement nécessaires pour le **conteneur MariaDB** :

  - MARIADB*ROOT*PASSWORD: passwordmariadb
  - MARIADB*DATABASE: wordpress * MARIADB*USER: wordpress
  - MARIADB_PASSWORD: wordpress

  Voici les variables d'environnement nécessaires pour le **conteneur Wordpress** :

  - WORDPRESS*DB*HOST: servbdd
  - WORDPRESS*DB*NAME: wordpress
  - WORDPRESS*DB*USER: wordpress
  - WORDPRESS*DB*PASSWORD: wordpress

====== Retour Accueil Docker ====== * Docker