LAB 2 : Manipulation des conteneurs avec le CLI

Exercice 1 : Hello from Alpine

Le but de ce premier exercice est de lancer des containers basés sur l'image alpine

1. Lancez un container basé sur **alpine** en lui fournissant la commande **echo hello**
2. Quelles sont les étapes effectuées par le docker daemon ?
3. Lancez un container basé sur alpine sans lui spécifier de commande. Qu'observez-vous ?

Exercice 2 : manipulation des commandes récurrentes

Le but de cet exercice est de manipulern des commandes récurrentes

1. Exécutez les commandes suivantes

```
$ docker run -it --name=test1 alpine:latest date

$ docker run -it --name=test1 alpine:latest date
```

2. Pourquoi ça ne marche pas?
3. Que pouvez-vous faire pour les faire fonctionner tous les deux (il y a au moins deux façons)?

Exercice 3 : shell intéractif

Le but de cet exercice est de lancer des containers en mode interactif

1. Lancez un container basé sur alpine en mode interactif sans lui spécifier de commande
2. Que s'est-il passé ?
3. Quelle est la commande par défaut d'un container basé sur alpine ?
4. Naviguez dans le système de fichiers
5. Utilisez le gestionnaire de package d'alpine (apk) pour ajouter un package

```
$ apk update

$ apk add curl
```

Exercice 4 : foreground / background

Le but de cet exercice est de créer des containers en foreground et en background

1. Lancez un container basé sur alpine en lui spécifiant la commande **`ping 8.8.8.8`**
2. Arrêter le container avec CTRL-C

Le container est-il toujours en cours d'exécution ?

Note: vous pouvez utiliser la commande **`docker ps`** que nous détaillerons dans l'une des prochaines lectures), et qui permet de lister les containers qui tournent sur la machine.

3. Lancez un container en mode interactif en lui spécifiant la commande **`ping 8.8.8.8`**
4. Arrêter le container avec CTRL-P CTRL-Q

Le container est-il toujours en cours d'exécution ?

5. Lancez un container en background, toujours en lui spécifiant la commande **`ping 8.8.8.8`**

Le container est-il toujours en cours d'exécution ?

Exercice 5 : liste des containers

Le but de cet exercice est de montrer les différentes options pour lister les containers du système

1. Listez les containers en cours d'exécution
2. Est-ce que tous les containers que vous avez créés sont listés ?
3. Utilisez l'option -a pour voir également les containers qui ont été stoppés
4. Utilisez l'option -q pour ne lister que les IDs des containers (en cours d'exécution ou stoppés)

Exercice 6 : inspection d'un container

Le but de cet exercice est l'inspection d'un container

1. Lancez, en background, un nouveau container basé sur nginx:1.14 en exposant le port 80 du container.
2. Notez l'identifiant du container retourné par la commande précédente.
3. Inspectez le container en utilisant son identifiant
4. En utilisant le format Go template, récupérez le nom et l'IP du container
5. Manipuler les Go template pour récupérer d'autres informations
6. Pour plus de pratique, comment déterminer rapidement et succinctement l'identifiant de l'image et la date de création d'une image Alpine?

Exercice 7 : exec dans un container

Le but de cet exercice est de montrer comment lancer un processus dans un container existant.

1. Lancez un container en background, basé sur l'image alpine. Spécifiez la commande **`ping 8.8.8.8`** et le nom ping avec l'option --name

2. Observez les logs du container en utilisant l'ID retourné par la commande précédente ou bien le nom du container

Quittez la commande de logs avec CTRL-C

3. Lancez un shell sh, en mode interactif, dans le container précédent

4. Listez les processus du container

Qu'observez vous par rapport aux identifiants des processus ?

Exercice 8 : cleanup

Le but de cet exercice est de stopper et de supprimer les containers existants

1. Listez tous les containers (actifs et inactifs)

1. Stoppez tous les containers encore actifs en fournissant la liste des IDs à la commande stop

2. Vérifiez qu'il n'y a plus de containers actifs

3. Listez les containers arrêtés

4. Supprimez tous les containers

5. Vérifiez qu'il n'y a plus de containers

Exercice 9 : Installer LAMP dans un conteneur ubuntu

Le but de cet exercice est d'installer LAMP dans un conteneur ubuntu

L'acronyme LAMP désigne un ensemble de quatre technologies open source : un système d'exploitation Linux, un serveur web Apache, un système de bases de données MySQL et le langage de programmation PHP. L'objectif de cet exercice est d'installer et configurer ces quatre briques dans un conteneur Docker en se basant sur l'image **Ubuntu 18.04** ainsi que d'installer phpMyAdmin pour l'administration graphique de la base de données MySQL.

1. Vérifier que le moteur Docker est en cours d'exécution

2. Chercher l'image de conteneur en question

3. Télécharger l'image en local

4. Lancer un conteneur en se basant sur l'image déjà télécharger
5. Mettre à jour les sources listes du conteneur
6. Installer le serveur web Apache 2 dans le conteur en question
7. Démarrer le serveur web Apache
8. Consulter la page index.php en utilisant l'adresse IP de conteneur
9. Installer le SGBD MySQL dans le conteneur en question
10. Sécuriser l'accès au SGBD MySQL

Exercice 10 : Wordpess

Installer l'application Wordpress sur le conteneur créé dans l'exercice précédent.

Exercice 11 : Mise en place d'une chaine d'intégration continue

Le but de cet exercice est la mise en place de la chaine d'intégration continue sur des conteneurs docker.

Dans des conteneurs basés sur l'image Centos 7, ré-implémenter les composants de la chaine d'intégration continu :

- Java
- Maven
- Jenkins
- Artifacory