TP « Master » Docker– 2exercices – Durée approx. ~ 2h00min.

A l'aide du support de cours et des éventuels mémentos réalisez les exercices suivants :

(Tout code ou implémentation compilant ou non sera étudié)

Le présent sujet de TP comporte 4 pages

*I – **Proposer un service d'infrastructure via conteneur***

Le but de cet exercice est de mettre en œuvre un conteneur docker proposant une fonctionnalité essentielle à l'infrastructure, à savoir la fourniture d'adresses IP via DHCP.

La démarche se déroule en cinq étapes à savoir :

1. *La récupération de l'image networkboot/dhcpd depuis le hub docker.*
2. *La création du fichier de configuration pour le service*
3. *Le démarrage du conteneur avec les paramètres appropriés.*
4. *La vérification de la disponibilité du service.*
5. *L'arrêt du conteneur associé.*

- Pour réaliser la récupération de l'image, téléchargez la dernière image docker *networkboot/dhcpd* depuis internet.
  - https://hub.docker.com/r/networkboot/dhcpd/
  - Vous pouvez alternativement charger l'image depuis un fichier avec la commande adéquate.
  - Vous pouvez alternativement créer une image depuis une image d'une distribution classique alpine/centos/ubuntu etc.

- Lors du démarrage du service il est nécessaire de remplir les préconditions suivantes :
  - Associer le fichier de configuration décrivant les paramètres réseau ainsi que définir la stack/pile réseau à associer au conteneur.
    - D'après la documentation Le fichier de configuration */data/dhcpd.conf* se situe au sein du dossier
    - La stack/pile réseau dans ce cas de figure est celle de la machine hôte (cf paramètre : ***--network***)

Exemple de fichier de configuration :

```
authoritative;

default-lease-time 7200; # Duree du bail DHCP
max-lease-time 7200;     # Duree max du bail

subnet 10.16.64.0 netmask 255.255.255.0 {
          option routers 10.16.64.254;                # IP routeur
          option subnet-mask 255.255.255.0;           # Masque de sous-reseau
          range 10.16.64.10 10.16.64.67;              # Plage addressable (si restriction
          option broadcast-address 10.16.64.255;      # IP de broadcast
          option domain-name-servers 10.16.64.253;    # IP serveur DNS
          option domain-name "smb137.lecture.cnam";   # Suffixe DNS par defaut de l'hote
          option domain-search "smb137.lecture.cnam"; # Suffixe DNS par defaut lors de recheches (ping etc)
}
```

- Avoir une adresse IP dans le sous-réseau associé à la description réalisée dans le fichier de configuration :

Pour rappel : les commandes suivantes permettent d'ajouter une adresse IP à une interface existante

ip addr add **IP**/**CIDR** dev **IFACE**
ifconfig **IFACE**:**0 IP** netmask **NETMASK** up

Exemples: ip addr add **10.16.64.1**/**24** dev **eth0**
ifconfig **eth0**:**0 10.16.64.1** netmask **255.255.255.0** up

- Une fois le conteneur démarré, vous devez être à même de pouvoir solliciter le service DHCP via :
  - L'ajout d'une nouvelle carte réseau
  - L'ajout d'une nouvelle adresse IP à une carte existante

Pour rappel : la commande dhclient permet de solliciter le service dhcpd

Exemple : dhclient -4 eth0 => permet d'obtenir une IPv4 auprès du service pour l'interface eth0

- La vérification de la disponibilité de service s'effectue simplement par l'exécution de la commande : ip addr show

⇨ Une nouvelle adresse IP dans le sous-réseau associé doit désormais être présente.

*II – **Proposer une stack applicative « wordpress »***

Le but de cet exercice est de mettre en œuvre une stack applicative wordpress synchronisant 2tiers apache/php et mysql.

La démarche se déroule en cinq étapes à savoir :

1. *La récupération des images depuis le hub docker.*
2. *La création du fichier de configuration docker-compose pour le service*
3. *Le démarrage de la stack.*
4. *La vérification de la disponibilité du service.*
5. *L'arrêt du conteneur associé.*

- Pour réaliser la récupération des images, téléchargez les dernières images docker *mysql* et *wordpress* depuis internet.
  - Vous pouvez alternativement charger les images depuis des fichiers avec la commande adéquate.

- La seconde étape consiste à créer un fichier *docker-compose.yml* au sein duquel nous allons retrouver pour chaque conteneur :
  - L'image utilisée
  - L'ensemble des variables d'environnement nécessaires
  - Les ports et volumes à mapper

- ***Attention** : vérifiez la présence de la commande docker-compose, en cas d'absence de cette dernière suivez la procédure l'installation : https://docs.docker.com/compose/install/*

  Dans notre cas il va falloir s'intéresser aux deux documentations suivantes :
  - MySQL : https://hub.docker.com/_/mysql/
  - Wordpress : https://hub.docker.com/_/wordpress/

- L'étape numéro 3 consiste à démarrer la pile applicative via la commande docker-compose ainsi que le fichier précédemment écrit.
  - Vous pouvez vous référer à la documentation de la commande pour ce faire

- L'étape numéro quatre consiste à vérifier l'état du service proposé. Pour ce faire il est nécessaire de procéder en deux temps :
  - Vérifier l'état des conteneurs avec les options adéquates de la commande docker
  - Vérifier l'accessibilité de l'application via un navigateur.

*IV – **Rappels***

Approche incrémentale du développement

Pour obtenir les résultats attendus aux différents exercices, veuillez toujours appliquer une approche incrémentale en termes d'ajout de code/fonctionnalité.

Par exemple: une approche incrémentale pour ce type d'exercice "ls-like" serait :

1. La récupération des paramètres.
2. Tester fichier/répertoire.
3. Parcourir les éléments du répertoire en affichant leur nom.
4. Alimenter chaque fichier avec une information supplémentaire : permission / taille / propriétaire…

Documentation

Pour obtenir des informations ou de la documentation ayez le réflexe d'utiliser les pages du manuel.

Par exemple :

- man docker
- https://docs.docker.com/
- https://github.com/docker/docker.github.io/blob/master/compose/compose-file/index.md

Gestion des erreurs

Afin d'avoir une gestion des erreurs la plus précise possible ayez les réflexes :

- D'associer un terminal et d'afficher la sortie standard du conteneur sur le terminal.
- De parcourir les fichiers de logs au sein du conteneur ciblé.