Docker : Création d'image

Etape 1 : Création d'une image à partir d'un container

1. Lancez un container basé sur une image alpine, en mode interactif, et en lui donnant le nom c1

Solution

La commande suivante permet de créer le container demandé. On utilise l'option **--name** pour spécifier le nom du container.

```
docker run -it --name c1 alpine
```

2. Lancez la commande **curl google.com**

   Qu'observez-vous ?

Solution

L'utilitaire curl n'est pas disponible dans une image alpine, il faut l'installer.

```
/ # curl google.com
```

```
/ # curl google.com
/bin/sh: curl: not found
```

3. Installez curl à l'aide du gestionnaire de package **apk**

Solution

La commande suivante permet de mettre à jour la liste des packages et d'installer l'utilitaire curl:

```
apk update && apk add curl
```

4. Quittez le container avec CTRL-P + Q (pour ne pas killer le processus de PID 1)

   Créer l'image myping à partir du container c1

Solution

Afin de créer l'image myping à partir du container c1, il faut utiliser la commande suivante

```
docker commit c1 myping
```

5. Lancer un shell sh dans un container basé sur l'image **myping**

Solution

La commande suivante permet de lancer un shell **sh** dans un container basé sur l'image **myping**. Même si l'on ne précise pas la commande **sh**, la commande utilisée par défaut est celle de l'image à partir de laquelle **myping** a été créée, c'est à dire **alpine**.

```
docker run -ti myping
```

6. Depuis le shell, lancer curl sur google.com

Solution

```
# curl google.com
```

7. Supprimer l'image myping

Solution

```
docker container stop $(docker container ls -aq)
docker container rm $(docker container ls -aq)
docker rmi myping
```

Etape 2 : Création d'une image à partir d'un Dockerfile

1. Créez un fichier Dockerfile avec les instructions suivantes

   - Image de base : ubuntu:16.04
   - Installation de l'utilitaire ping. Assurez-vous de mettre à jour la liste des dépots (apt-get update) et également d'utiliser l'option -y lors de l'installation du package iputils-ping (apt-get install -y iputils-ping)
   - Instruction ENTRYPOINT définie comme étant la commande **ping**
   - Instruction CMD définie par l'adresse IP **8.8.8.8** correspondants à un serveur DNS de Google

Solution

Le Dockerfile demandé est le suivant

```
FROM ubuntu:16.04
RUN apt-get update -y && apt-get install -y iputils-ping
ENTRYPOINT ["ping"]
CMD ["8.8.8.8"]
```

2. A partir de ce Dockerfile, créez l'image **ping:1.0**

Solution

La commande suivante permet de créer l'image **ping:1.0**

```
docker image build -t ping:1.0 .
```

3. Lancez un container basé sur l'image **ping:1.0** sans lui donner de commande

   Qu'observez-vous ?

Solution

La commande suivante permet de lancer un container basé sur l'image ping:1.0 sans précisez de commande après le nom de l'image

```
docker container run ping:1.0
```

Le processus lancé dans le container correspond à la concaténation des instructions ENTRYPOINT et CMD définies dans le Dockerfile, à savoir **ping 8.8.8.8 .**

4. Lancez un container basé sur l'image **ping:1.0** en lui spécifiant la commande **www.google.com**

   Que s'est-il passé ?

Solution

La commande suivante permet de lancer un container basé sur l'image ping:1.0 en lui spécifiant l'argument www.google.com en tant que commande

```
docker container run ping:1.0 www.google.com
```

Comme un argument (**www.google.com**) a été spécifié à la suite du nom de l'image (**ping:1.0**), celui-ci à redéfini l'instruction **CMD** spécifiée dans le Dockerfile. Le processus lancé dans le container correspond à la concaténation de l'instruction ENTRYPOINT (celle du Dockerfile) et du paramètre spécifié sur la ligne de commande, c'est à dire à ping www.google.com

5. Supprimer l'image ping

Solution

```
docker container stop $(docker container ls -aq)
docker container rm $(docker container ls -aq)
docker rmi ping :1.0
```