

LAB 6 - Gestion du réseau et des données pour les conteneurs

Partie 1 : Mappage des ports

Exemple 1 : Créez un conteneur en se basant sur la description suivante :

- Image : nodejs/node
- Port à publier :
 - Sur le host : 8080
 - Dans le conteneur : 80
- Nom du conteneur : testAppNode

Exemple 2 : Créez un conteneur basé sur l'image jenkins:2.60.3 en mappant les port 8080 et 50000.

```
docker run --name serveurCI-CD --publish 8080:8080 --publish 50000:50000  
jenkins:2.60.3
```

Exemple 3 : Depuis le docker hub, cherchez une image correspondante à la version 7.9 du **sonarqube** et créez un conteneur pour mapper les ports nécessaires.

```
sudo docker run --publish 9000:9000 --name sonar3 sonarqube
```

Partie 2 : Persistance des données

Exemple 1 : Définition d'un volume dans un Dockerfile

- Créez le Dockerfile suivant

```
FROM alpine:3.8  
VOLUME ["/data"]
```

- Construisez l'image imgvol à partir de ce Dockerfile

```
[root@localhost dockerfiles]# docker build -t imgvol -f Dockerfile-volume .
```

- lancez un shell interactif dans un container, nommé c2, basé sur l'image imgvol.

```
[root@localhost dockerfiles]# docker run -it --name c2 imgvol
```

- créez le fichier /data/hello.txt dans le conteneur

```
/ # echo "Ceci un fichier hello" > /data/hello.txt
```

- inspectez pour récupérer la clé **Mounts** afin d'avoir le chemin d'accès du volume sur la machine hôte.

```
[root@localhost dockerfiles]# docker inspect c2
/var/lib/docker/volumes/bfb0ee2404a940743950ec61c86d428aeff5ed05328aff122aa
f92f6d5ea8195/_data
```

- Supprimez le conteneur créé

```
[root@localhost dockerfiles]# rm -f
/var/lib/docker/volumes/bfb0ee2404a940743950ec61c86d428aeff5ed05328aff122aa
f92f6d5ea8195/_data/hello.txt
```

- Vérifiez l'existence du fichier hello.txt

```
/ # ls /data/
```

Exemple 2 : Définition d'un volume au lancement d'un container

- Lancez un container avec les caractéristiques suivantes:
 - basé sur l'image alpine:3.8
 - nommé c3
 - exécution en background (option -d)
 - définition de /data en tant que volume (option -v)
 - spécification d'une commande qui écrit dans le volume ci-dessus

```
[root@localhost dockerfiles]# docker run -d --name c3 -v /data alpine:3.8
sh -c 'ping 8.8.8.8 >> /data/ping.txt'
```

- Inspectez le container et repérez notamment le chemin d'accès du volume sur la machine hôte.

```
[root@localhost dockerfiles]# docker inspect c3
```

- Supprimez le container c3.

```
[root@localhost dockerfiles]# docker rm c3
```

Exemple 3 : Utilisation des volumes via la CLI

- Créez un volume nommé html
- Listez les volumes existants et vérifiez que le volume html est présent
- Inspectez le volume créé
- Lancer un container basé sur nginx et monter le volume html sur le point de montage /usr/share/nginx/html du container. En publiant également le port 80 du container sur le port 8080 de l'hôte.

```
[root@localhost dockerfiles]# docker run --name nginxServer1 -p 8080:80 -d  
-v html:/usr/share/nginx/html nginx
```

- Affichez le contenu du volume html.

```
[root@localhost dockerfiles]# ls /var/lib/docker/volumes/html/_data/  
50x.html  index.html
```