TP 2 : Docker et Docker Compose

Prérequis

Installer Docker sur la machine.

- Sur Ubuntu

```
sudo apt install -y docker.io
sudo apt install -y docker-compose-v2
```

- Sur MacOS, avec Brew :

```
brew install docker
```

- Sur Windows ? Utilisez une VM...

Télécharger Docker Desktop (`https://www.docker.com/products/docker-desktop/`)

1 Premier conteneur

Après avoir démarré Docker Desktop, ouvrez un terminal pour exécuter les commandes du TP.

- Lancez votre premier conteneur avec l'image officielle `hello-world` :

  ```
  docker run hello-world
  ```

  Observez le message affiché et expliquez ce qui se passe.

- Listez les conteneurs existants avec `docker ps -a`.

2 Explorer l'image Alpine

1) Téléchargez l'image Alpine :

```
docker pull alpine
```

2) Lancez un conteneur interactif basé sur Alpine :

```
docker run -it alpine /bin/sh
```

3) Explorez le système de fichiers (commande `ls`, `cat`, etc.).

4) Essayez de trouver un noyau Linux dans le conteneur. Que remarquez-vous ?

Note : vous pouvez quitter l'environnement interactif d'Alpine en utilisant `exit`.

3 Manipulation d'images

1) Listez les images disponibles localement avec `docker images`

2) Téléchargez l'image `busybox` avec la commande `docker pull busybox`

3) Inspectez cette nouvelle image avec `docker image inspect busybox`.

4) Supprimez une image inutilisée avec `docker image rm busybox`

4 Construire une image avec un Dockerfile

1) Créez un fichier `Dockerfile` contenant les instructions suivantes :

```
FROM alpine:latest
WORKDIR /app
COPY hello.sh .
RUN chmod +x hello.sh
CMD ["./hello.sh"]
```

2) Créez un script `hello.sh` qui affiche `"Hello Docker!"`.

3) Construisez l'image :

```
docker build -t monapp:latest .
```

4) Lancez l'image et vérifiez le résultat.

5 Bonnes pratiques

1) Modifiez votre Dockerfile pour installer `curl`, puis supprimez les fichiers temporaires afin de réduire la taille de l'image.

2) Comparez la taille de l'image avant et après l'optimisation.

6 Déploiement avec Docker Compose

1) Créez un fichier `compose.yaml` décrivant deux services :

- `web` basé sur l'image `nginx:latest`, exposant le port 8080.
- `db` basé sur l'image `mariadb:latest`, avec un mot de passe administrateur défini par une variable d'environnement.

2) Lancez la pile avec :

```
docker-compose up -d
```

3) Vérifiez que le serveur Nginx est accessible sur `http://localhost:8080`.

4) Arrêtez et supprimez les conteneurs avec :

```
docker-compose down
```