

TP Docker

1. Instructions d'installation de Docker

Pour installer Docker sur votre machine :

1. Rendez-vous sur le site officiel : <https://docs.docker.com/get-docker/>
2. Choisissez votre système d'exploitation (Windows, macOS, Linux).
3. Téléchargez et installez Docker Desktop ou Docker Engine selon votre configuration.
4. Vérifiez l'installation avec la commande :
`docker --version`

2. Objectif pédagogique

L'objectif de ce TP est de vous familiariser avec les concepts fondamentaux de Docker, la création et la gestion de conteneurs, l'utilisation des images, ainsi que les commandes essentielles pour manipuler l'écosystème Docker.

Rendu attendu :

Le dépôt git contenant :

- Un readme avec l'ensemble des tâches réalisés
- Les fichiers par TP organisé par dossier. Avec l'arborescence suivante :
 - TP Docker
 - Fichier TP
 - Readme.md

3. Exercice : Manipulation de base des conteneurs

Étapes :

1. Vérifiez la version de Docker :
`docker --version`
→ Affiche la version installée de Docker.
2. Listez les images disponibles sur votre machine :
`docker images`
→ Montre toutes les images stockées localement.

3. Téléchargez une image depuis Docker Hub :
`docker pull hello-world`
→ Télécharge l'image officielle "hello-world".
4. Exécutez un conteneur à partir de l'image :
`docker run hello-world`
→ Lance un conteneur qui affiche un message de bienvenue.
5. Listez les conteneurs en cours d'exécution :
`docker ps`
→ Montre les conteneurs actifs.
6. Listez tous les conteneurs (actifs et stoppés) :
`docker ps -a`
→ Affiche tout l'historique des conteneurs.
7. Supprimez un conteneur :
`docker rm <ID_conteneur>`
8. Supprimez une image :
`docker rmi <ID_image>`

4. Exercice : Création d'un serveur web avec Docker

1. Téléchargez l'image officielle Nginx :
`docker pull nginx`
2. Lancez un conteneur Nginx en arrière-plan :
`docker run -d -p 8080:80 --name mon_nginx nginx`
→ Démarre un serveur web accessible sur <http://localhost:8080>.
3. Vérifiez que le conteneur est actif :
`docker ps`
4. Accédez à la page dans votre navigateur pour voir la page par défaut de Nginx.
5. Arrêtez le conteneur :
`docker stop mon_nginx`

6. Supprimez le conteneur :

```
docker rm mon_nginx
```

5. Exercice : Déploiement d'une application Python Flask

Objectif : Créer et lancer une application web simple avec Flask à l'aide de Docker.

Consignes :

- Créez un fichier app.py contenant une application Flask minimale. (hello world)
- Écrivez un Dockerfile pour construire l'image de l'application.
- Construisez l'image et lancez un conteneur.
- Testez l'application depuis votre navigateur.

Durée estimée : 1h.

6. Exercice : Utilisation de docker compose

Objectif : Déployer une application complexe avec Docker.

Consignes :

- Compléter votre fichier app.py afin de lui ajouter une connexion à un bdd mongoDB
- Écrivez un fichier docker compose avec vos deux conteneurs
- Lancer le docker compose puis à minima vérifier que la connexion c'est bien effectuée sur la base de données

Durée estimée : 1h.

7. Cheatsheet Docker

Commande	Description
docker --version	Affiche la version installée de Docker
docker images	Liste les images disponibles localement
docker pull <image>	Télécharge une image depuis Docker Hub
docker run <image>	Lance un conteneur à partir d'une image
docker ps	Liste les conteneurs en cours d'exécution
docker ps -a	Liste tous les conteneurs, actifs et stoppés
docker stop <id nom>	Arrête un conteneur en cours
docker rm <id nom>	Supprime un conteneur
docker rmi <id nom>	Supprime une image
docker exec -it <id nom> bash	Ouvre un terminal interactif dans un conteneur
docker build -t <nom_image> .	Construit une image à partir d'un Dockerfile