TP1 : Introduction à Docker et Cycle de Vie

IUT Sénart-Fontainebleau

2025

1 Objectifs

— Installer Docker
— Comprendre les concepts de base de Docker (images, conteneurs)
— Maîtriser le cycle de vie d'un conteneur
— Savoir analyser et débugger un conteneur

Préparation du rendu

1.1 Rendu

Tous les commandes utilisées doivent se trouver sur dans votre **projet git docker**.

Ce projet git être privée et partagé avec `https://grond.iut-fbleau.fr/pierront` (Maxime Pierront)

Il faut m'envoyer un mail à l'adresse suivante : **contact@pierrontmaxi.me** avec dans l'object **obligatoirement** cette chaine de caractère sinon je ne verrai pas votre mail : [**BUT2FI-DOCKER**]

Le mail devra contenir :

— Votre nom et prénom
— Le lien vers votre dépôt git

1.2 Organisation du dépôt git

Votre dépôt devra être organisé de la manière suivante :

— Un fichier README.md expliquant le contenu du dépôt
— Un dossier TP1 contenant :
  — Un fichier COMMANDS.md listant toutes les commandes utilisées avec leurs résultats
  — Un fichier QUESTIONS.md contenant les réponses aux questions de compréhension

— Un schéma du cycle de vie Docker (format image)

---

2 Installation

2.1 Installation de Docker

1. Se rendre sur le site officiel : `https://docs.docker.com/engine/install/`
2. Suivre la procédure correspondant à votre système d'exploitation

2.2 Vérification de l'installation

Exécuter les commandes suivantes :

```
docker --version
docker run hello-world
```

3 Premiers pas avec Docker

3.1 Images Docker

3.1.1 Lister les images disponibles

```
docker images
```

3.1.2 Télécharger une image

```
docker pull nginx
```

3.1.3 Rechercher une image

```
docker search ubuntu
```

Note : Noter les différentes colonnes de la sortie (STARS, OFFICIAL, AUTOMATED)

3.2 Gestion basique des conteneurs

1. Créer un premier conteneur :

```
docker run nginx
```

2. Observer la différence en mode détaché :

```
docker run -d nginx
```

4 Cycle de vie des conteneurs

4.1 États des conteneurs

4.1.1 Commandes d'observation

```
docker ps       # conteneurs en cours d'ex cution
docker ps -a    # tous les conteneurs
```

4.1.2 Exercice pratique

1. Créer trois conteneurs nginx avec des noms différents :

```
docker run -d --name web1 nginx
docker run -d --name web2 nginx
docker run -d --name web3 nginx
```

2. Observer et noter les états de chaque conteneur
3. Identifier les colonnes importantes dans la sortie de docker ps

4.2 Manipulation des états

Pratiquer les commandes suivantes sur les conteneurs créés :

```
docker stop web1
docker start web1
docker restart web2
docker pause web3
docker unpause web3
docker kill web1
```

Pour chaque commande, observer et noter le changement d'état

5 Inspection et Debug

5.1 Logs et monitoring

```
# Observer les logs
docker logs [container-name]

# Suivre les logs en temps r el
docker logs -f [container-name]

# Statistiques des conteneurs
docker stats
```

5.2 Inspection détaillée

```
# Inspecter un conteneur
docker inspect [container-name]

# Ex cuter des commandes dans un conteneur
docker exec -it [container-name] /bin/bash
```

6 Exercice final

Créer un environnement de test complet :

1. Créer un conteneur nginx en mode détaché avec le nom "web-test"
2. Pratiquer toutes les commandes du cycle de vie vues précédemment
3. Analyser les logs et les différents états
4. Nettoyer l'environnement (suppression des conteneurs)

7 Travaux Pratiques Supplémentaires

7.1 Rapatrier l'image officielle du serveur web apache (httpd)

Sur docker hub, vous devez retrouver la commande pour rapatrier cette version de l'image apache (httpd) :

`alpine3.17`

7.2 Créer un volume

On va utiliser la commande suivante :

`docker volume`

pour créer le volume nommé `volume_serveur_web`

Dans ce volume y déposer un fichier nommé « index.html » dans lequel l'étudiant va inscrire son nom.

7.3 Démarrer un conteneur

On va utiliser la commande suivant pour démarrer 3 conteneurs avec notre image apache :

`docker run`

L'étudiant doit démarrer un conteneur pour répondre à chacun de ces cas :

— Démarrer un conteneur en mode détaché nommé tp21

— Démarrer un conteneur en mode détaché en montant le volume précédemment créé nommé tp22

— Démarrer un conteneur en mode détaché en montant le volume précédemment créé et en exposant le port 80 nommé tp23

8 Questions de compréhension

Répondre aux questions suivantes :

1. Quelle est la différence entre une image et un conteneur ?
2. Expliquez la différence entre `docker stop` et `docker kill`
3. Pourquoi utilise-t-on le mode détaché ?
4. Comment vérifier la santé d'un conteneur ?
5. Quels sont les différents états possibles d'un conteneur ?

9 Livrables attendus

1. Un document listant toutes les commandes utilisées avec leurs résultats
2. Les réponses aux questions de compréhension
3. Un schéma du cycle de vie Docker

Conseils

— Prenez des notes sur les commandes utilisées

— Documentez les erreurs rencontrées et leurs solutions

— N'hésitez pas à consulter la documentation officielle : `https://docs.docker.com/`

10 Questions de compréhension

Répondre aux questions suivantes :

1. Quelle est la différence entre une image et un conteneur ?

2. Expliquez la différence entre `docker stop` et `docker kill`
3. Pourquoi utilise-t-on le mode détaché ?
4. Comment vérifier la santé d'un conteneur ?
5. Quels sont les différents états possibles d'un conteneur ?

11 Livrables attendus

1. Un document listant toutes les commandes utilisées avec leurs résultats
2. Les réponses aux questions de compréhension

Conseils

— Prenez des notes sur les commandes utilisées
— Documentez les erreurs rencontrées et leurs solutions
— N'hésitez pas à consulter la documentation officielle : `https://docs.docker.com/`