

TP « Handle » Docker– 2exercices – Durée approx. ~ 1h30min.

A l'aide du support de cours et des éventuels mémentos réalisez les exercices suivants :

(Tout code ou implémentation compilant ou non sera étudié)

Le présent sujet de TP comporte 2 pages

I – Proposer un service

Le but de cet exercice est de mettre en œuvre un conteneur docker proposant un service Jenkins.

La démarche se déroule en quatre temps à savoir :

1. *La récupération de l'image jenkins depuis le hub docker.*
2. *Le démarrage du conteneur proposant le service.*
3. *La vérification de la disponibilité du service.*
4. *L'arrêt du conteneur associé.*

- Pour réaliser la récupération de l'image, téléchargez la dernière image docker **jenkins** depuis internet.
 - Vous pouvez alternativement charger l'image depuis un fichier avec la commande adéquate.
- Lors du démarrage du service **jenkins** veuillez penser à exposer le port associé au service afin de le rendre accessible depuis votre machine hôte.
- Une fois le conteneur démarré, vous devez être à même de pouvoir accéder au service depuis votre machine hôte grâce à une url de la syntaxe : http://IP_MACHINE_HOTE:8080/

II – Service « from scratch »

Le but de cet exercice est de reconstruire un conteneur docker proposant un service Jenkins à partir d'une base Tomcat.

La démarche se déroule en quatre temps à savoir :

1. *La récupération des ressources.*
2. *La rédaction d'un fichier Dockerfile.*
3. *La construction de l'image.*
4. *Le test du service.*

- Pour réaliser la récupération de l'image, téléchargez les ressources nécessaires à la construction de l'image à savoir la dernière image docker **tomcat** ainsi que l'archiver web (**war**) jenkins depuis internet.
 - Vous pouvez alternativement charger l'image depuis un fichier avec la commande adéquate.
 - Vous trouverez l'archive web de l'application jenkins sur le site <https://jenkins.io/>
- Afin de construire une nouvelle image docker appuyez-vous sur la rédaction d'un fichier Dockerfile en spécifiant :
 - Le fait que l'on se repose sur l'image **Tomcat** en tant que base de départ.
 - Le fait que l'archive **war** doit être disponible au sein de l'image générée.
 - Le fait que le port d'écoute **8080** réseau soit déclaré.
- Une fois le conteneur construit assurez-vous que celui-ci soit fonctionnel en démarrant ce dernier et vérifiant le fait de pouvoir accéder.

IV – *Rappels*

Approche incrémentale du développement

Pour obtenir les résultats attendus aux différents exercices, veuillez toujours appliquer une approche incrémentale en termes d'ajout de code/fonctionnalité.

Par exemple: une approche incrémentale pour ce type d'exercice "ls-like" serait :

1. La récupération des paramètres.
2. Tester fichier/répertoire.
3. Parcourir les éléments du répertoire en affichant leur nom.
4. Alimenter chaque fichier avec une information supplémentaire : permission / taille / propriétaire...

Documentation

Pour obtenir des informations ou de la documentation ayez le réflexe d'utiliser les pages du manuel.

Par exemple:

- man 3 stat / man 2 open / man 2 readdir / man errno

Gestion des erreurs

Afin d'avoir une gestion des erreurs la plus précise possible ayez le réflexe d'utiliser les codes retours **ERRNO** spécifiés dans les pages de manuel

Par exemple:

- | | |
|------------------|---|
| ➤ EEXIST | File exists (POSIX.1) |
| ➤ EFAULT | Bad address (POSIX.1) |
| ➤ EISDIR | Is a directory (POSIX.1) |
| ➤ ENOTDIR | Not a directory (POSIX.1) |
| ➤ ELOOP | Too many levels of symbolic links (POSIX.1) |