

Travaux Pratiques "Principes et outils pour le DevOps"

TP 2 – Docker – Conteneurs et images

Les dépôts initiaux se trouvent à l'adresse <https://github.com/CERI-M1-DevOps>

Ce second TP correspond au dépôt **TPConteneursEtImages**, que vous allez forker en cliquant sur le lien proposé dans le README. Vous travaillez ensuite localement dans un clone sur votre poste de travail.

Travail à rendre pour chacun des TP

L'intérêt de différents TP est de vous faire comprendre et retenir les différentes commandes et outils de base de Docker.

Pour chaque TP, vous produirez un rapport au format pdf qui présente les réponses et explications des différentes étapes réalisées, que vous pousserez sur votre dépôt.

Pour produire le rapport, vous pouvez dans un premier temps vous aider d'éléments/outils sur le net pour avoir les réponses aux questions, mais vous tâcherez dans un second temps d'avoir les solutions par vous-même.

Lorsqu'une question mentionne "Constatez", cela signifie que vous devez présenter ce qui se passe, et expliquer pourquoi : pour cela, vous devez produire une explication personnelle, et non pas un copier-coller de ce qu'un outil du net vous aura suggéré. Il s'agit de restituer ce que vous avez compris, c'est un travail individuel et personnel.

Si besoin, vous pourrez consulter un aide-mémoire des commandes Docker :

<https://devhints.io/docker>

Environnement de travail

Vous allez travailler dans une machine virtuelle Ubuntu, sur laquelle Docker sera installé. , Au cours des TP, lorsque on mentionne la "machine hôte", il s'agit du poste de travail qui accueille la machine virtuelle VM. Lorsque, par contre, on parle de "machine virtuelle" ou VM, il s'agit de celle sur laquelle Docker est installé.

Dans ce premier TP, le rapport à produire ne concerne que l'utilisation de Docker (à partir de la partie 2), et non pas la première partie mise en place et configurations.

Partie 1 – Mise en place d'une machine virtuelle et configurations minimales

- Téléchargez le fichier vmdocker.ova à l'adresse suivante

<https://cloud.univ-avignon.fr/index.php/s/sSC26NT46ZQzM9a>

pour mettre en place la machine virtuelle dans laquelle vous allez travailler.

Le nom de l'utilisateur ainsi que son mot de passe que vous utiliserez sont `dockeruser`.

- Ajoutez un alias `ipa` pour la commande `ip -c -br a`. Que fait cette commande ?

- Vérifiez que l'accès réseau est possible en lançant 3 ping consécutifs sur `quad9`.

- On va accéder à la VM depuis la machine hôte. Quelle est l'adresse IP de cette VM ? Vérifiez en "pingant".
- Vérifiez le statut du serveur ssh.
- Ouvrez un terminal sur la machine hôte et établissez une connexion en ssh pour l'utilisateur
- Installez l'outil jq, processeur de json en ligne de commande.

Partie 2 – Installation et démarrage de Docker

A - Allez sur <https://docs.docker.com/engine/install/ubuntu/>

Choisir la version "Installation à partir du repo apt" et n'effectuez que les étapes 1 et 2.

<https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>

- Utilisez une commande docker pour afficher la version (client et serveur !)
- Comment voir les composants du daemon Docker qui tournent ?
- Quels sont les services/socket utilisés par docker ? Quel utilisateur a démarré ces services ?
- Que faire pour arrêter docker ? Quel est le statut du socket ?
- Que faire pour désactiver/réactiver docker ?
- Vous constatez que les commandes docker s'utilisent en mode sudo. Pour permettre à l'utilisateur dockeruser d'exécuter les commandes docker sans passer par sudo, effectuez les étapes suivantes :
 - créez un groupe nommé docker
 - passez en root dans son répertoire home
 - ajoutez (append) l'utilisateur dockeruser au groupe docker
 - puis redémarrer la VM
 - réessayez d'afficher la version sans passer en sudo.

Partie 3 – Premières manipulations de conteneurs et d'images

Docker propose une commande help pour afficher les commandes autorisées, et chaque commande possède aussi une option help.

Vous pouvez aussi vous aider de la commande Unix apropos, combinée avec un grep docker pour trouver une commande qui correspond à votre recherche. Par exemple, pour trouver une commande docker qui permet de rechercher (search – faites appel à vos connaissances en anglais...) des éléments, vous pouvez exécuter : `apropos search | grep ^docker`.

Vous trouverez aussi plus de détails sur le site officiel de Docker à l'adresse : <https://docs.docker.com/reference/cli/docker/>

- Quel est le répertoire dans lequel Docker stocke ses objets.
- Quels sont les différentes catégories d'objets Docker qui peuvent être stockés ?
- Consultez le contenu du répertoire approprié qui contient les conteneurs : combien y en a-t-il pour l'instant ?

- Utilisez une commande pour rechercher des images, essayez avec l'image hello-world
- Faites exécuter un conteneur qui correspond à l'image ayant le plus d'étoiles. Faites une copie d'écran qui enregistre les étapes réalisées par docker.
- Vérifiez maintenant le contenu du répertoire des conteneurs. Combien y a-t-il de conteneurs ?
- Vérifiez aussi le contenu du répertoire des images
- Quel est le sha256 de l'image ?
- Utilisez la commande docker (sans option) qui permet de lister les conteneurs en exécution.
- Combien de conteneurs qui s'exécutent ?
- Quel est l'alias de la commande que vous venez d'utiliser ?

Remarque : par défaut, si on ne précise pas d'objet, docker considère que l'opération demandée s'applique aux conteneurs.

- Utilisez la commande docker qui permet de lister les images. Quel est l'identifiant de l'image utilisée ?
- Réexécutez le conteneur et comparez à la précédente exécution : expliquez.
- Combien de conteneurs s'exécutent et combien sont stockés localement.
- Essayez de supprimer l'image et expliquez. Ne la supprimez pas finalement.
- Utilisez une commande qui permet de lister tous les conteneurs et pas uniquement ceux en exécution et observez leur statut.
- Quel est le nom de ces conteneurs ?
- Utilisez une option pour afficher l'information de façon non tronquée.
- Exécutez une nouvelle fois l'image, puis supprimez ce dernier conteneur en utilisant son nom.
- Exécutez une nouvelle fois l'image en utilisant son sha256 en donnant un nom au nouveau conteneur, puis constatez.
- Faites ce qu'il faut pour supprimer l'image (sans forcer).
- Utilisez une commande qui donne des infos globales sur le système et observez le nombre de conteneurs et d'images.
- Quelle commande (et options) docker pouvez-vous utiliser pour n'afficher que les IDs des conteneurs ?
- Utilisez une possibilité du bash pour exploiter les résultats de cette dernière commande afin de supprimer tous les conteneurs en une seule commande.
- Supprimez maintenant l'image et constatez.
- Faites à nouveau exécuter un conteneur, que vous nommerez hello1, pour la même image hello-world, que se passe-t-il ?
- Quel est l'identifiant de l'image ? Constatez.

- Créez un conteneur, que vous nommerez hello2, mais sans le démarrer, puis observez son statut.
- Démarrez-le ensuite en utilisant son nom et constatez son statut ensuite. Quel est l’affichage ?
- Utilisez maintenant l’option qui permettra d’attacher l’entrée et la sortie standard pour démarrer ce conteneur hello2.
- Utilisez la même option pour démarrer le conteneur hello1.
- Exécutez maintenant un nouveau conteneur nommé hello3, mais en faisant en sorte qu’il n’affiche pas d’information sur la sortie standard (en background donc). Constatez dans la liste (totale) des conteneurs.
- Exécutez maintenant un nouveau conteneur nommé hello4, mais en faisant en sorte que ce conteneur ait totalement disparu après son exécution. Constatez dans la liste (totale) des conteneurs.

Vous pouvez supprimer l’image hello-world avant de passer à la partie suivante.

Partie 4 – DockerHub

Rendez-vous à l’adresse de ce registre public : <https://hub.docker.com/>

- Combien de différents types d’images trouve-t-on sur ce registre public, comment se fait la classification ?
- Retrouvez l’image officielle de l’OS Ubuntu.
- Consultez les différentes versions proposées, comment les distingue-t-on ?
- Quelle est la version la plus récente ? Quelle différence a-t-elle avec la version latest ? Quels sont leurs identifiants respectifs ?
- Quelles sont les vulnérabilités de la version la plus récente ? Et celles de la version latest ?
- Observez comment ces vulnérabilités sont classées.
- Comparez la version latest avec la version noble : quel pourrait être l’intérêt de cette situation ?
- En cliquant sur le tag de chacune des 2, consultez les couches de ces images, et observez à partir de quand elles diffèrent.
- Placez-vous dans le répertoire `/var/lib/docker/image/overlay2/layerdb/sha256` et listez son contenu.
- Téléchargez l’image ubuntu : par défaut, quelle est celle qui est téléchargée
- Téléchargez maintenant l’image ubuntu la plus récente.
- Utilisez un filtre pour n’afficher que les images référençant ubuntu.
- Identifiez la commande qui est lancée lorsqu’on exécute un conteneur à partir de l’image ubuntu.

Pour la suite, vous pouvez supprimer l’image ubuntu:rolling.

Partie 5 – Interagir avec un conteneur

Rappel : la séquence de caractères Ctrl-D saisie sur l'entrée d'une commande Unix a pour effet de fermer son entrée standard. Si cette commande attend ses informations de l'entrée standard, elle n'a donc plus rien à faire, et par conséquent, elle se termine. Faites un essai en ouvrant un terminal sur la machine hôte, puis tapez Ctrl-D et constatez.

- Lancez un conteneur à partir de l'image ubuntu et constatez son statut. Essayez de redémarrer ce même conteneur en interactif. Enfin supprimez le.
- Lancez, à partir de l'image ubuntu, un conteneur nommé os_ubuntu en interactif et attaché à un terminal.
- Quelle est la commande qui correspond au processus de PID 1 de ce conteneur ?
- Dans cet OS, exécutez les commandes whoami, pwd, ls et hostname. Notez son ID.
- Ouvrez un deuxième terminal, connectez-vous en ssh sur la VM et observez le statut du conteneur en cours d'exécution. Que constatez-vous (à part le statut) ?
- Revenez dans le conteneur sur le premier terminal. Déplacez-vous dans le répertoire home.
- Quittez ce conteneur avec la commande unix classique puis observez le statut du conteneur.
- Démarrez ce conteneur en interactif. Dans quel répertoire vous trouvez-vous ? Pourquoi ?
- Dans le second terminal, utilisez une commande qui permet d'inspecter le conteneur. Constatez qu'il est en cours d'exécution. Retrouvez son Pid.
- Vous pouvez aussi essayer d'utiliser jq pour obtenir cette information, par exemple voir : <https://blog.madrzejewski.com/jq-traiter-parser-json-shell-cli/> (ou tout lien de votre choix)
- Dans ce même terminal, retrouvez le processus dont le PID est celui que vous venez d'identifier et constatez.
- Revenez dans le conteneur (dans le premier terminal) et déplacez-vous à nouveau dans home.
- Quittez maintenant ce conteneur en utilisant la combinaison de touches Ctrl-p Ctrl-q. Quel est maintenant le statut du conteneur ?
- Attachez le terminal au conteneur. Quel est le répertoire courant ? Ajoutez-y un fichier avec un contenu quelconque.
- Quittez à nouveau ce conteneur sans l'arrêter, puis utilisez une commande docker pour voir les différences dans le système de fichiers du conteneur.
- Utilisez une commande docker pour exécuter la commande Unix hostname dans ce conteneur (sans le passer en foreground).
- Utilisez une commande docker pour exécuter la commande Unix bash en interactif dans ce conteneur. Vérifiez le nombre de processus bash qui tournent dans le conteneur.
- Quittez à nouveau ce conteneur sans l'arrêter, puis utilisez une commande docker qui affiche les processus du conteneur.
- Revenez dans le conteneur et arrêtez-le en le quittant.

- Lancez maintenant un nouveau conteneur nommé ll à partir de l'image ubuntu dont la commande est maintenant ls -l.
- Redémarrez ce conteneur pour obtenir le même affichage.
- Lancez un nouveau conteneur nommé ps avec la commande ps aux, mais en faisant en sorte que ce conteneur disparaisse après son exécution. Constatez le PID et constatez le statut.
- Lancez un nouveau conteneur nommé salut avec la commande echo Bonjour, puis relancez ce conteneur.
- Lancez un nouveau conteneur avec une commande infinie (sh -c "while true ; sleep 3600 ; done"), puis faites en sorte de le supprimer.
- Démarrez le conteneur os_ubuntu en interactif. Dans ce shell, lancez une commande qui affiche salut toutes les 3 secondes.
- Dans le deuxième terminal connecté en ssh à la VM, utilisez une commande docker pour vous attacher au conteneur qui tourne dans le premier terminal puis constatez. Interrompez le processus qui effectue le salut, lancez une commande basique (par exemple ls) et constatez dans l'autre terminal.
- Quittez ce conteneur en utilisant la commande bash exit avec un code retour non nul. Constatez le statut de ce conteneur.
- Utilisez une commande docker pour voir le log du conteneur os_ubuntu.
- Inspectez le conteneur à la recherche du fichier contenant son journal (log). Vous pouvez le consulter avec jq.
- Utilisez une commande docker pour supprimer tous les conteneurs arrêtés.

A la fin de cette partie, arrêter et supprimer tous les conteneurs actifs (une seule commande).

Partie 6 – Inspection et manipulation d'images

Téléchargez l'image python:3.9.

- Combien de couches ont été téléchargées ?
- Observez les couches de cette image (il existe aussi une option non tronquée)
- Quelle est la dernière couche ?
- Puis téléchargez python : 3.14.0a1. Combien de couches ont été téléchargées ? Expliquez.
- Utilisez l'inspection d'une image et l'outil jq pour afficher les couches des 2 images python, observez les couches communes.
- Recréez un nouveau conteneur os_ubuntu à partir de l'image ubuntu et ajoutez dans le répertoire home un nouveau fichier, avec un contenu quelconque. Quittez-le sans l'arrêter. Utilisez une commande docker pour exporter le système de fichiers dans une archive tar.
- Créez une image image_ubuntu_with_file en important l'archive. Comment est-elle taggée ? Observez ses couches (son historique).

- Créez un conteneur à partir de cette image. Si vous avez un bash, vous pouvez vérifier que le fichier créé précédemment existe.
- A partir du conteneur `os_ubuntu`, committez maintenant une image `image2` et consultez ses couches. Utilisez maintenant cette image pour créer un conteneur.

A la fin de cette partie, il n'est plus nécessaire de conserver les images créées.

Partie 7 – Quelques informations générales

- Utilisez une commande docker pour consulter la consommation des conteneurs en exécution.
- Utilisez une commande docker pour voir la consommation disque des différents objets docker. Essayez aussi la version détaillée.