

Manipulation Docker

Le but de ce TP va être de manipuler des instances docker et/ou des containers et d'approprier les commandes de base.

Attention : certaines commandes comme le démarrage d'un container peuvent prendre parfois jusqu'à 30 secondes, voir un peu plus, soyez patient :)

1) Connection sur la machine distante ou on doit travailler

Pour se connecter sur la machine distante, il faut utiliser une connexion ssh. C'est une connexion sécurisée qui permet de ne pas faire passer le mot de passe en clair sur le réseau.

Vous pouvez utiliser le programme « putty » sur windows ou « ssh » sur linux.

La connexion se fait sur une machine dont l'adresse IP est 195.83.161.135. Pour que chaque étudiant ait sa propre machine, les ports ssh de connexion diffèrent. Lors de la mise en place du TP, l'enseignant vous a donné un numéro. Ce numéro est important car il va être utilisé pour le port ssh de connexion et le mot de passe.

Pour résumer:

IP: 195.83.161.135
port: 122<votre numéro sur 2 digits>
login: student
mot de passe est: student

Ex:

Si votre numéro est 86 alors le port sera 12286

Une fois sur votre machine distante, vous pouvez l'administrer totalement. Vous devez absolument changer le mot de passe par défaut le plus rapidement possible.

Pour pouvoir noter le TP, il faut que vous mettiez dans un fichier nommé /root/info.txt votre nom et prénom. Attention, pas de fichier = zéro !

Tout ce qui est noté sur fond bleu ... (xxx réponse à fournir)...correspond à une question à laquelle vous devez répondre dans le document de rendu.

A la fin de la séance, vous devez rendre ce document dans la zone de rendu « Rendu séance », même si vous n'avez pas fini le TP.

2) Manipulation de container (suite):

Dans le contexte du TP précédent, dites comment on peut constater qu'on est bien dans un container (guest OS) et de type CentOS, ou tout autre container, et pas dans la machine initiale qui vous est assignée.

Mettez cette information dans le document de rendu (1ère réponse à fournir).

Lancez un container CentOS en mode interactif, puis quittez le (Attention, pas forcément évident en fonction de la commande utilisé pour lancer le container).

Dites comment vous avez fait pour quitter dans le document de rendu (2ème réponse à fournir).

Dans la question suivante, on va lancer le même container mais cette fois ci en arrière-plan et non-interactif. Pour vérifier que le container tourne bien en arrière-plan, nous allons faire (sur une seule ligne de commande), un shell qui tourne en boucle et affiche une chaîne de caractère et aussi qui fait une pause de 1 seconde entre chaque boucle sinon votre processeur va être occupé à 100% !

Avant de traiter cette question, posons-nous la question de pourquoi lancer l'instance docker en arrière plan avec un shell qui boucle sur lui même, au lieu juste de lancer un le container avec le shell à exécuter (3ème réponse à fournir) ?

Maintenant pour traiter la question, je vous propose de commencer par faire la ligne de commande en shell et la tester sur la machine qui vous est assignée. Une fois que cela fonctionne, on peut ensuite tester en lançant la commande shell via le container docker.

Mettez cette commande et la sortie écran produite lorsque vous avez rentré cette commande dans le document de rendu (4ème réponse à fournir).

Constatez qu'aucun téléchargement ne se fait car l'image a déjà été téléchargée lors de l'exercice précédent et a été conservé dans un cache.

Trouvez maintenant les commandes docker qui permettent de :

- Afficher la liste des containers actifs (5ème réponse à fournir).
- Afficher l'ensemble des containers (y compris ceux qui sont stoppés ou terminés) (6ème réponse à fournir).

Mettez toutes ces commandes et les sorties écrans associés dans le document de rendu.

Cherchez comment afficher la sortie écran (output) du container précédemment lancé, et mettez la commande et l'output dans le document de rendu (7ème réponse à fournir).

Trouvez comment stopper un container et mettez la commande dans le document de rendu (8ème réponse à fournir).

Lorsqu'on fait des opérations sur un container docker, il est possible de nommer le container docker de plusieurs manières (son nom, etc...).

Quels sont les identifiants possibles qu'on peut utiliser pour préciser le container lorsqu'on veut l'arrêter ?

Mettez ces identifiants dans le document de rendu (9ème question) (Attention, on ne demande pas ici l'identifiant précis de votre container, mais d'une manière générale avec quels éléments on peut identifier un container pour des opération de start, stop , etc...)

Maintenant on vous demande de redémarrer le container que vous avez arrêté précédemment et de vous y attacher (Attention: parfois il faut faire un retour chariot pour que vous récupériez la main dans votre container).

Quelle commande avez vous fait pour relancer le container arrêté précédemment ? (10ème question)

Quelle commande avez-vous fait pour vous attacher au container ? (11ème question)

Vérifier en utilisant la commande qui permet de lister tous les containers, que votre container a bien démarré.

Mettre la commande de verification dans le document de rendu (12ème réponse à fournir).

Arrêtez ce container.

On va maintenant chercher une image de la distribution Busybox.

Trouvez la/es commande(s) qui permettent de la télécharger seulement (et pas de l'exécuter).

Mettez cette/ces commande(s) et la sortie écran produite dans le document de rendu (13ème réponse à fournir).

Busybox est utilisé dans des systèmes où on a besoin d'un shell minimal, essayez maintenant de lancer le container busybox avec le shell bash. Est-ce que ça fonctionne ?

Expliquez le résultat dans le document de rendu (14ème réponse à fournir).

Maintenant que vous en savez un peu plus sur Busybox, lancez un shell en interactif sur le container Busybox.

Mettez cette commande et la sortie écran produite lorsque vous avez rentré cette commande dans le document de rendu (15ème réponse à fournir).

Nous allons maintenant essayer de modifier une image existante, pour cela on va faire plusieurs étapes:

- Récupérer l'image qu'on veut modifier
- Lancer l'image de manière interactive pour être dans un container
- Modifier le contenu de ce container
- Sauvegarder le container en image avec un nom que vous choisirez

Essayez de faire ça sur une image busybox, en rajoutant dans le container, le fichier /home/essai.txt

Expliquez dans le document de rendu comment vous avez fait ? (16ème réponse à fournir).

Bien sur, il faut tester si l'image que vous avez modifiée a vraiment été modifiée, pour cela lancer un container à partir de l'image modifiée, et regardez si le fichier /home/essai.txt est bien présent.

Expliquez dans le document de rendu les différentes étapes de vérification. (17ème réponse à fournir).

Vous allez maintenant effacer l'image modifiée que vous avez créé. Indiquez la commande utilisée pour trouver et effacer cette image. (18ème réponse à fournir).

Relancez un container busybox et vérifiez que c'est bien l'image d'origine qui s'est lancé et pas celle modifiée (qui ne doit plus exister).

Pour rappel, un container n'est ni plus ni moins qu'un système de fichier, trouvez quelle commande docker utiliser pour lister le contenu des fichiers présent dans une image docker. ATTENTION, il ne faut pas lancer d'instance de l'image docker. (19ème réponse à fournir).

Faire fonctionner le jeu <https://hub.docker.com/r/kazhar/mazingame/> . Expliquez sur le document de rendu ce que vous avez fait pour que cela fonctionne. (20ème réponse à fournir).