

Virtualisation

Professeur : Florent Gluck

Assistant : Sebastien Chassot

April 13, 2022

Lab - LXC

Introduction

Le but de ce travail pratique est de vous familiariser avec la manipulation des containers LXC.

Préparation

Tous les travaux pratiques jusqu'à la fin du semestre sont à réaliser sur la machine distante qui vous a été fournie.

Ceux-ci nécessitent un volume de stockage important. Vos machines disposent de nombreux disques totalisant un espace de stockage de 110GB. Vous allez commencer ici par reconfigurer votre support de stockage avec LVM afin d'utiliser cet espace à bon escient.

Le système de fichiers racine de 20GB doit rester inchangé. Par contre, supprimez les contenus des 3 disques supplémentaires et créez un volume groupe de 90GB avec ces 3 disques. Créez-y un volume logique de la capacité totale du volume groupe. Créez alors un système de fichiers de type ext4 au sein de ce volume groupe et montez le dans `/mnt/data`. Créez ensuite le répertoire vide `/mnt/data/lxc`, puis créez le lien symbolique `/var/lib/lxc` afin qu'il pointe dessus.

Une fois ces opérations réalisées, procédez sur votre machine distante à l'installation des packages pour utiliser les containers LXC.

Création de containers privilégiés

Dans cette section, vous allez créer des containers privilégiés en utilisant les templates propres aux distributions supportées par LXC, puis ensuite via le template `download`.

- Quels sont les templates disponibles sur votre système ?

Commencez par créer un container privilégié basé sur la distribution Ubuntu Impish (21.10), en utilisant le template `ubuntu` et mesurez le temps nécessaire jusqu'à ce que la création du container soit terminée.

- Quel temps avez-vous obtenu ?

Créez un deuxième container basé sur la même distribution, mais en utilisant le template `download`. Mesurez également le temps de création du container.

- Quel temps avez-vous obtenu ?
- Pour quelle raison les temps obtenus sont-ils si différents ? Développez.

Créez un nouveau container basé à nouveau sur le template `ubuntu` (impish) et mesurez le temps de création.

- Quel est le temps mesuré et pourquoi la création de ce container est beaucoup plus rapide qu'auparavant ?

Listez les containers installés sur votre système pour vérifier qu'ils ont été créés avec succès.

- Dans quels états se trouvent-ils ?
- A quel chemin sur la machine hôte se trouve l'arborescence du système de fichiers racine (*rootfs*) du premier container ?
- Quelle est la taille du système de fichiers racine de ce premier container ?

Processus

Démarrez un container et connectez-y vous avec `lxc-attach` et `lxc-console`.

- Quelle est la différence entre ces deux commandes ?

Connectez-vous à un container avec `lxc-console` et authentifiez vous avec l'utilisateur `ubuntu`. Dans le container exécutez le programme `vimtutor`.

- Dans le container, à quel utilisateur appartient `vimtutor` et quel est son PID (aide: `ps aux | grep ...`) ?
- Sur la machine hôte, à quel utilisateur appartient `vimtutor` et quel est son PID ?
- Analysez et expliquez les différences observées pour `vimtutor` entre dans le container et sur la machine hôte.

Quittez `vimtutor`, puis lancez un shell root avec `sudo -s`. Exécutez alors à nouveau `vimtutor`, mais en root cette fois-ci.

- Dans le container, à quel utilisateur appartient `vimtutor` et quel est son PID ?
- Sur la machine hôte, à quel utilisateur appartient `vimtutor` et quel est son PID ?
- A nouveau, analysez et expliquez les différences observées pour `vimtutor` entre dans le container et sur la machine hôte.

Dans le container, quittez `vimtutor`. Listez maintenant tous les processus en cours.

- Quel est le nombre de processus en cours dans le container (aide: `wc -l`) ?
- A titre de comparaison, combien de processus sont en cours sur la machine hôte ?

Dans un shell de la machine hôte, exécutez `vimtutor`.

- Dans le container, voyez-vous le processus `vimtutor` ? Comprenez-vous pourquoi ?

Exécution non-priviliée

Créez maintenant un nouveau container, mais cette fois-ci de type **non-privilié**. Bien sûr, n'oubliez pas au préalable de modifier la configuration système comme nécessaire.

Démarrez le container, puis attachez-vous à celui-ci. Vous devriez vous retrouver dans un shell root dans le container.

Exécutez à nouveau `vimtutor`. Dans le container, vérifiez que le processus `vimtutor` appartient bien à l'utilisateur root.

- Sur la machine hôte, inspectez le processus `vimtutor`. Quel est le UID du processus ? Que pouvez-vous alors en conclure ?

Limites

On désire maintenant imposer une limite sur le nombre de CPU utilisables ainsi que sur la mémoire vive à disposition dans le container. Modifiez alors la configuration de votre container pour qu'il n'utilise que 64MB de RAM. Votre machine hôte possède un seul CPU, mais modifiez toutefois la configuration de votre container afin que celui-ci n'utilise explicitement qu'un seul coeur (ainsi, sur un hôte à plusieurs coeurs, seul un coeur sera utilisé).

- Quel fichier avez-vous modifié et quelles modifications y avez-vous apportées ?
- Comment pouvez-vous vérifier que le nombre de CPU effectifs est bien correct (aide: **stress**) ?
- Même question pour la quantité de RAM disponible ?