

TD1 : Namespaces, Cgroups, et OverlayFS sous Linux

Objectif : Le but de ce TP est d'explorer les fonctionnalités du noyau Linux qui permettent la virtualisation des OS, à savoir les namespaces, les cgroups, et le système de fichiers de type overlay.

Prérequis :

- Un système Linux (par ex. Ubuntu/Debian).
- Privilèges root ou sudo.

Exercice 1 :

Objectif : Isoler les processus à l'aide des namespaces pour mieux comprendre comment les conteneurs isolent les ressources.

1. En utilisant la commande `unshare`, créez un nouveau namespace réseau (`net`) et `pid`.
2. Vous êtes maintenant dans un nouveau namespace `pid` et réseau. Vérifier que le processus de PID 1 est dans le nouveau namespace.
3. Lister les interfaces réseaux présente de le nouveau namespace.
4. Quitter le namespace et revenir au shell parent.

Exercice 2 :

Objectif : Utiliser cgroup v2 pour limiter les ressources d'un processus (CPU, cpubset, mémoire).

1. Vérifions que cgroup v2 est bien activé.
2. Installer le paquet `cgroup-tools` (si non installé).
3. Installer le paquet `stress`.
4. Créez un sous-répertoire `moncroupe` dans `/sys/fs/cgroup` qui servira à limiter les ressources d'un processus.
5. Limitez l'utilisation du CPU à 20 % en utilisant le contrôleur `cpu.max` dans cgroup v2. Les valeurs sont définies comme `quota :period` (quotas en microsecondes).
6. Limitez l'utilisation de la mémoire à 15 Mo
7. Qu'est-ce que `cpuset` dans le contexte de cgroup v2, et comment permet-il de mieux contrôler l'utilisation du CPU par les processus ?
8. Activer le contrôleur `cpuset`
9. Limitez le processus à certains CPU spécifiques (par exemple, les CPU 0 et 1). Le paramètre `cpuset.cpus` définit les coeurs que le processus peut utiliser.
10. Déplacez le shell courant dans le cgroup `moncroupe` pour appliquer les limites définies.
11. tester les limitations en exécutant un processus (`stree`) gourmand en ressources. Observez son comportement avec `htop` ou `top`. Vous devriez voir qu'il ne s'exécute que sur les CPU 0 et 1.

Exercice 3 :

Objectif : Utiliser OverlayFS pour créer un système de fichiers à couches.

1. Dans le répertoire TP1, créez les sous-répertoires `lower`, `upper`, `work` et `merged`. Le répertoire `lower` sera la couche de base, et `upper` sera la couche modifiable.
2. Peuplez la couche `lower` par trois fichiers `lfic1.txt`, `lfic2.txt` et `dup.txt` dont les contenus respectives sont "Le fichier 1 dans la couche inférieure", "Le fichier 2 dans la couche inférieure" et "Le fichier duplique dans la couche inférieure"
3. Peuplez la couche `lower` par trois fichiers `ufic1.txt`, `ufic2.txt` et `dup.txt` dont les contenus respectives sont "Le fichier 1 dans la couche supérieure", "Le fichier 2 dans la couche supérieure" et "Le fichier duplique dans la couche supérieure"
4. Monter le système de fichiers OverlayFS.
5. Quel est le contenu du répertoire `merged`
6. Affichez le contenu du fichier `dup.txt`. Que constatez-vous ?
7. Modifiez le fichier `lfic1.txt` en Le fichier 1 dans la couche inférieure modifie
8. Affichez le contenu du fichier `lfic1.txt` du répertoire `lower`. Qu'en concluez-vous ? Où se trouve le fichier modifié ?
9. Supprimer les fichiers `lfic2.txt` et `ufic2.txt`.
10. Est-ce que c'est fichiers ont été supprimés dans les couches `lower` et `upper` ? Comment peut-on savoir si un fichier dans la couche modifiable a été supprimé ?

Exercice 4 :

Objectif : Nettoyage.

1. Démonter le système de fichiers OverlayFS `merged`, puis supprimez les répertoires `lower`, `upper`, `work` et `merged`
2. Supprimer le cgroup `v2 mongroupe`

TP1 : Construction d'un conteneur à partir de zéro

Ce TP vous permet de comprendre les bases de la création d'un conteneur à partir de zéro en utilisant des outils bas niveaux du noyau Linux. Vous utiliserez les namespaces pour isoler l'environnement, OverlayFS pour créer un système de fichiers en couches, et pivot_root pour réassigner la racine du système de fichiers. Ces techniques sont utilisées par des technologies de conteneurisation telles que LXC, Docker pour fournir des environnements isolés et efficaces.

Exercice 1 :

Préparation de l'environnement

1. Dans le répertoire TP1 créez une sous-répertoire **container** pour le conteneur.
2. Installez **debootstrap** pour initialiser un système de fichiers minimal.
3. Initialisez un système de fichiers minimal Debian minimal dans le répertoire **container**.
4. Utilisez **unshare** pour créer un nouveau namespace. qui isolera le réseau, les processus, et le système de fichiers ...
5. Utiliser OverlayFS pour simuler un système de fichiers en couches. Utiliser les noms de répertoires (container (lower), upper, work et merged)
6. Montez le système de fichiers OverlayFS. La couche basse est celle qui contient le système de fichiers minimal : le répertoire **container**
7. Expliquez en quoi la commande **pivot_root** est essentielle lors de la création d'un conteneur. En quoi diffère-t-elle de **chroot**, et pourquoi **pivot_root** est-il souvent préféré pour l'isolement des conteneurs ?
8. Créez un répertoire temporaire pour pouvoir exécuter la commande **pivot_root**
9. Utilisez **pivot_root** pour déplacer la racine de **merged** à **old_root**
10. Monter les systèmes de fichiers nécessaires : **proc**, **sys**, **tmp**
11. Recouvrer le shell courant par un **bash**.
12. Ecrire un script shell permettant la création d'un conteneur à partir de zéro.
13. Ecrire un programme C permettant la création d'un conteneur à partir de zéro.