




# Conteneurs Linux

Anthony BUSSON



## Conteneurs: les outils linux

- ▀ Les conteneurs sont basés sur des fonctionnalités du noyau Linux
  - ▀ namespace
  - ▀ cgroups
  - ▀ setccomp

## namespace

- ▀ Depuis 2002
  - ▀ Linux 2.4.19
- ▀ 8 différents namespace depuis la version 5.6 du noyau Linux (7 sous Ubuntu)
- ▀ Chaque type de namespace définit ce que le processus ou groupe de processus peut accéder et modifie le comportement des appels systèmes associés
  - ▀ mount: montage des périphériques / de systèmes de fichiers
  - ▀ process ID (pid)
  - ▀ network: table de routage, ports, iptable, etc.
  - ▀ ipc (inter processes communication)
  - ▀ UTS (Unix Time Sharing): host and domain name
  - ▀ user ID: mapping des ID
  - ▀ time namespace: le temps peut être différent d'un namespace à l'autre (pas sous ubuntu)
  - ▀ Control group namespace (obsolete)

## Principe

- ▀ Un processus hérite des namespace de son père
- ▀ Si il ne le partage pas (voir commande unshare), un nouveau namespace est créer
  - ▀ Tous ses fils appartiendront au même namespace (sauf nouvel appel à unshare)
- ▀ Les appels systèmes peuvent avoir un comportement différent dans leur namespace:
  - ▀ Isolation
  - ▀ Privilège
- ▀ Par exemple, un namespace peut isoler un ensemble de fonctionnalités:
  - ▀ PID: processus
  - ▀ hostname: hostname
  - ▀ mount: ensemble de points de montage
  - ▀ network: table de routage, numéros de ports, /proc/net, etc.
  - ▀ etc.

## Identifiant d'un namespace

- Un namespace est identifié par un numéro d'inode.
- Tous les processus appartiennent à un namespace pour chaque catégorie (7 sous les versions actuelles de Linux/Ubuntu)
- Chaque processus a un lien symbolique vers chacun de ses namespaces
  - L'inode pointé a un numéro (d'inode) et un nom qui identifie le namespace
  - Tous les processus appartenant au même namespace ont ce même lien symbolique

/proc/<pid>/ns/<leLien>

```
busson@ubuntu: ~
busson@ubuntu:~$ sudo ls -l /proc/2298/ns/
total 0
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 cgroup -> 'cgroup:[4026531835]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 ipc -> 'ipc:[4026531839]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 mnt -> 'mnt:[4026531840]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 net -> 'net:[4026531992]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 pid -> 'pid:[4026531836]'
lrwxrwxrwx 1 busson busson 0 nov. 20 06:01 pid_for_children -> 'pid:[4026531836]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 user -> 'user:[4026531837]'
lrwxrwxrwx 1 busson busson 0 nov. 20 05:57 uts -> 'uts:[4026531838]'
busson@ubuntu:~$
```

## lsns

- La commande lsns liste les namespace disponible

```
busson@ubuntu:~$ lsns
  NS TYPE   NPROCS  PID USER  COMMAND
4026531835 cgroup    63 1757 busson /lib/systemd/systemd --user
4026531836 pid        63 1757 busson /lib/systemd/systemd --user
4026531837 user       63 1757 busson /lib/systemd/systemd --user
4026531838 uts        63 1757 busson /lib/systemd/systemd --user
4026531839 ipc        63 1757 busson /lib/systemd/systemd --user
4026531840 mnt        63 1757 busson /lib/systemd/systemd --user
4026531992 net        63 1757 busson /lib/systemd/systemd --user
```

Identifiant du namespace: c'est un inode

Type de namespace  
Voir slide précédent

Nombre de processus dans ce namespace

Plus petit PID dans ce ns

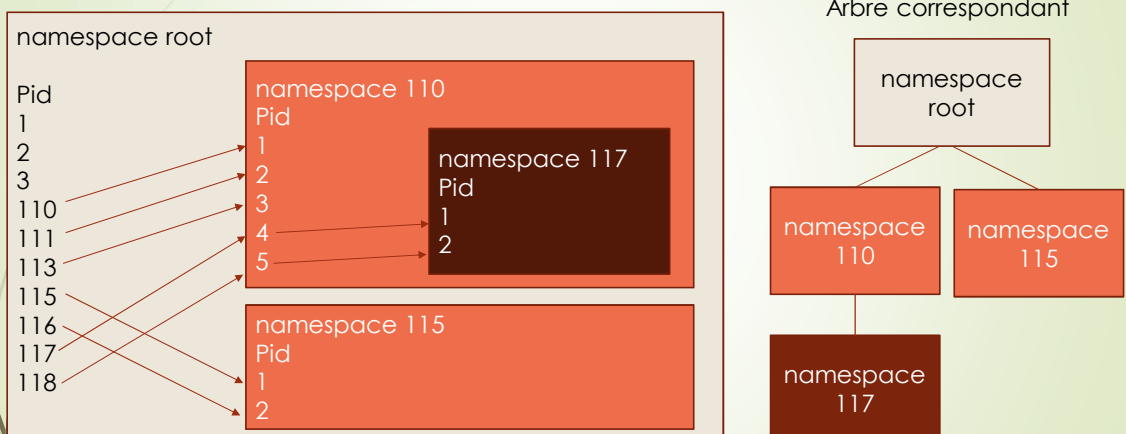
Commande (processus) pour lequel ces ns ont été affichés.

## Namespace pid

- Isolation des processus
  - Lorsqu'un processus est assigné a un nouveau namespace « pid »
    - Il récupère le pid 1 dans son propre namespace
    - Ses fils auront les pid 2, 3, etc.
    - Il ne peut voir ou interagir avec les processus des namespace pid « enfants »
  - Les processus ne peuvent interagir qu'avec les processus de leur propre namespace ou des namespace enfants
    - Interagir = utiliser des appels systèmes avec eux (signaux, etc.)

## Namespace pid: détail (1)

- Les namespace sont emboîtés (nested)



Le processus de pid 117 a 3 PID: un dans propre namespace (1 ici), et un dans chacun des namespaces de niveau supérieur (4 et 117).

## Namespace mount

- Isole les points de montages pour chacun des « mount » namespace
- Chaque namespace « mount » a sa liste de points de montage
- Intérêt: monter des parties du système qui seront propre au processus de ce namespace (et pas visible par les autres processus).
  - On peut donc avoir des dossiers qui sont les mêmes que ceux du SE et d'autres qui sont locales au namespace (suivant ce qui a été monté).

## Namespace UTS et user

- Namespace UTS: Permet de changer le hostname et domainname
  - Et aussi le NIS name (peu utilisé)
- namespace user (exemple du « root mapping »):
  - Associe l'uid 0 / gid 0 au processus qui créer le nouveau namespace
  - L'ensemble des namespace créer en même temps appartiennent à ce nouveau namespace user
  - Les capabilities de ces nouveaux namespace sont accessibles par cet utilisateur root (hostname, network, etc.)
  - Pour les autres namespaces, le root local n'a pas les droits.
  - Fonctionnement: mapping entre l'uid/gid du namespace et celui du host.

## Exemple namespace user: le root mapping

PROC  
1

ns user 1

ns net 1

ns pid 1

ns mount 1

Appartiennent tous au ns user 1  
Les droits/capabilités sont associés  
aux utilisateurs/root du ns user 1



Création d'un processus fils avec la création  
de nouveaux namespace: user et net.

PROC  
2

ns user 2

ns net 2

ns pid 1

ns mount 1

Pour les appels-systèmes touchant  
aux ns user et net, on regarde les  
privileges de l'utilisateur dans le  
name space user 2.  
Pour les appels systèmes des ns pid  
et mount on regarde les privileges  
de l'utilisateur dans le namespace  
user 1.

## namespace net

- ▀ Permet d'isoler les aspects réseau des processus.
- ▀ Pour chaque net namespace, les tables suivantes sont construites:
  - ▀ Une table de routage, une table arp
  - ▀ Règles de filtrage (iptables)
  - ▀ Etc.
- ▀ Les commandes doivent s'appliquer au bon namespace

## Les commandes

- lsns: affiche les ns d'un processus
- unshare:
  - associe un namespace à un processus existant
  - créer un processus avec un ou des namespaces différents
- nsenter:
  - permet de taper des commandes associés à certains namespace
    - Changer l'utilisateur
    - Monter des systèmes de fichiers ou périphériques
    - Changer le hostname
    - Etc.

## cgroups (control groups)

- Depuis 2008 (Linux version 2.6.24)
  - Août 2020: 5.8.5
- Limite l'usage des ressources pour un ensemble de processus
  - Priorisation: limite l'usage CPU
  - Mémoire: alloue un volume maximum de mémoire au groupe
  - Comptabilité: compte le temps CPU
  - Isolation: un groupe de processus sera associés à des namespace (déjà fait).
- Actuellement: cgroup version 2

## cgroups: fonctionnement

- Les cgroups sont créés/supprimés au travers de dossiers dans le répertoire suivant (pour le cpu ici / cela peut être memory aussi):

```
/sys/fs/cgroup/cpu/
```

- Le cgroup initial incluant tous les processus est le cgroup « / » se trouvant à la racine de /sys/fs/cgroup/cpu
- La création d'un cgroup peut se faire avec mkdir:

```
mkdir /sys/fs/cgroup/cpu/myCgroup
```

- Pour ajouter un processus à ce cgroup, il faut ajouter son pid dans le fichier

```
/sys/fs/cgroup/cpu/myCgroup/cgroup.procs
```

Les fils appartiennent aux cgroups de leur père.

## cgroups: création des limites

- CPU:
  - On alloue des ressources sur une base globale de 1024
  - Par exemple: 512 limitera à 50% du CPU
  - Alloué dans le fichier /sys/fs/cgroup/cpu/myCgroup/cpu.shares
- Mémoire:
  - On alloue en nombre d'octets
  - /sys/fs/cgroup/memory/myCgroup/memory.limit\_in\_bytes

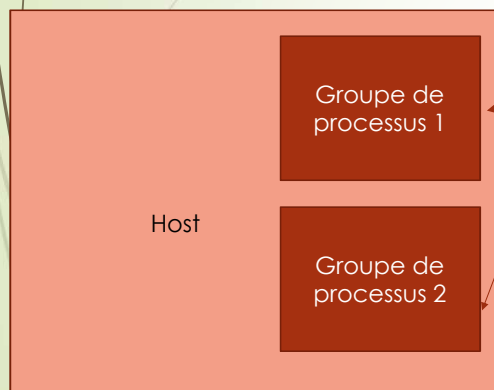


## Sécurité

- ▀ L'appel système `setccomp(2)`:
  - ▀ Filtre les appels systèmes qui peuvent être exécutés par un processus
  - ▀ Liste blanche
  - ▀ Liste noire
  - ▀ Une liste préétablie: `read`, `write`, `exit`, etc.

Les fils récupèrent les mêmes filtres que leur père.

## Conclusion



Un processus a initié une isolation:

- Montages (mount) isolés
- Isolation des processus fils
- Isolation du réseau
- Hostname propre
- Privilèges limités au namespace

namespaces

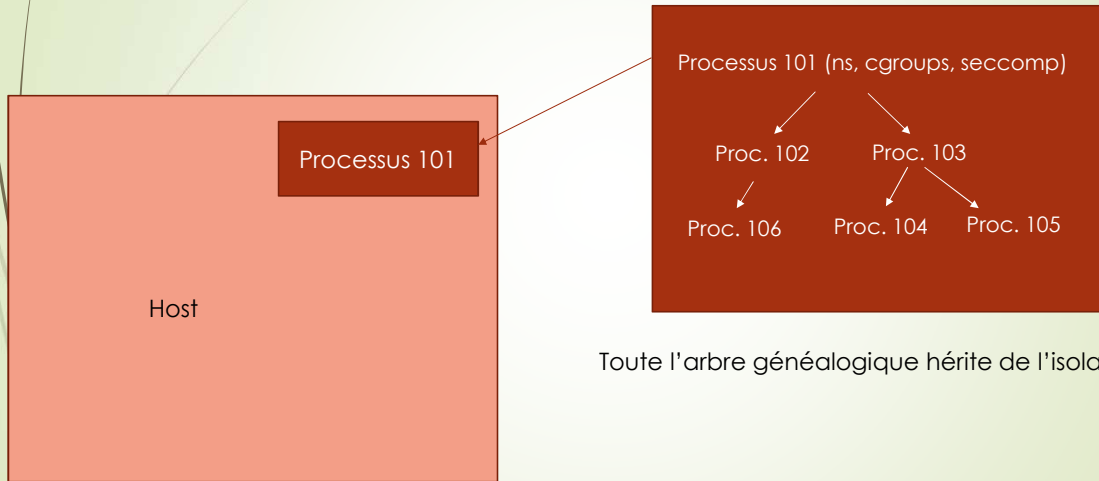
- Allocation limitée de CPU et mémoire

cgroups

- Limitation des appels systèmes possibles

seccomp

## Conclusion



Toute l'arbre généalogique hérite de l'isolation.