lxc.container.conf-2.0.5

Fichiers de configuration de conteneur pour LXC

Les conteneurs linux sont toujours créé avant être utilisés. Cette création définis un jeu de ressources système à virtualiser/isoler quand un processus utilise le conteneur. Par défaut, les pid, sysv ipc et points de montage sont virtualisés et isolés. Les autres ressources système sont partagées entre les conteneurs, tant qu'ils sont explicitement définis dans la configuration.

Configuration

lxc.include Spécifie un fichier de configuration à inclure

lxc.arch Autoriser un jeu d'architectures pour le conteneur, par exemple lancer des binaires 32bits dans un hôte 64bits

lxc.utsname Spécifie le hostname pour le conteneur

lxc.haltsignal Spécifie un signal utilisé pour arrêter le conteneur (utilisé par lxc-stop) Défaut : SIGPWR

lxc.rebootsignal Spécifie un signal utilisé pour redémarrer le conteneur (utilisé par lxc-stop) Défaut : SIGINT

lxc.stopsignal Spécifie un signal utilisé pour stopper le conteneur (utilisé par lxc-stop) Défaut : SIGKILL

lxc.init_cmd Chemin absolu dans le rootfs du conteneur du binaire à utiliser comme init. Défaut : /sbin/init

lxc.init_uid UID à utiliser dans un namespace user pour init. Défaut : 0

lxc.init_gid GID à utiliser dans un namespace user pour init. Défaut : 0

lxc.ephemeral (bool) Spécifie si un conteneur est détruit à l'arrêt

lxc.network Peut être utilisé sans valeur pour supprimer toutes valeurs précédentes.

lxc.network.type Spécifie le type de réseau à utiliser pour le conteneur. Chaque fois qu'un lxc.network.type est trouvé, une nouvelle configuration réseau commence et permet de définir plusieurs types de réseaux :

- **none** Le conteneur partage l'espace réseau de l'hôte
- **empty** Ne créé que l'interface loopback
- **veth** Créé une paire ethernet virtuel, dont la partie dans l'hôte est attaché à un bridge.
- **vlan** interface vlan lié avec l'interface
- **macvlan** lie une interface macvlan
- **phys** Une interface déjà existante est assignée au conteneur

lxc.network.vlan.id Pour le type vlan, indique le numéro du vlan

lxc.network.flags Àction à prendre pour le réseau (up=active l'interface)

lxc.network.link Spécifie l'interface à utiliser pour le trafic réseau réel (type phys ou macvlan)

lxc.network.mtu mtu max pour cette interface

lxc.network.name Nom de l'interface

lxc.network.hwaddr Adresse mac pour l'interface virtuelle

lxc.network.ipv4 IPv4 de l'interface

lxc.network.ipv4.gateway passerelle de l'interface

lxc.network.ipv6 IPv6 pour l'interface

lxc.network.ipv6.gateway Passerelle ipv6 pour l'interface

lxc.network.script.up script à exécuter après la création/configuration de l'interface (côté hôte).

lxc.network.script.down script à exécuter après la avant de détruire l'interface (côté hôte).

lxc.pts Si définis, le conteneur aura une nouvelle instance de pseudo tty. La valeur spécifie le nombre de tty permis.

lxc.console.logfile Chemin vers un fichier où la sortie de la console est écrite

lxc.console Chemin vers un périphérique auquel la console est attachée. Option dangereuse.

lxc.tty Spécifie le nombre de tty à rendre disponible dans le conteneur

lxc.devttydir Spécifie un répertoire sous /dev sous lequel créer les périphériques de console du conteneur. les pty Unix98 sont créés sur l'hôte et lié dans les périphériques attendus dans le conteneur. Par défaut, il sont liés sur /dev/console et /dev/ttyN. Cela peut empêcher des misesà jours. En changeant d'emplacement sous /dev, ils seront liés symboliquement.

lxc.autodev Par défaut lxc ne créé que quelques liens symboliques dans /dev du conteneur (fd,stdin,stdout,stderr) ais ne créé pas automatiquement les entrées de périphérique. À 1, lxc monte un nouveau tmpfs sous /dev (limité à 500k) et définis les périphériques minimum requis. À 0, empêche de monter et remplir /dev

lxc.kmsg Active /dev/kmrsg comme lien vers /dev/console. Défaut : 0

lxc.mount Les points de montage sont privés au conteneur. Spécifie l'emplacement d'un fichier fstab contenant les informations de montage.

lxc.mount.entry Spécifie un point de montage correspondant à une ligne dans le fichier fstab

lxc.mount.auto Spécifie quels systèmes de fichiers kernel devraient être automatiquement montés :

proc :mixed /proc est monté en rw, mais /proc/sys et /proc/sysrq-trigger sont montés en ro

proc :rw /proc est monté en rw

sys :mixed /sys en ro, /sys/devices/virtual/net en rw

sys :ro monté en ro

sys :rw monté en rw

cgroup :mixed Monte un tmpfs sous /sys/fs/cgroup, créé les répertoires pour toutes les hiérarchies auquel le conteneur est ajouté, créé les sous-répertoires avec le nom du cgroup, et mount le cgroup du conteneur dans ce répertoire. Le conteneur sera capable d'écrire dans son propre cgroup mais pas les parents, vu qu'ils sont remontés en ro

cgroup :ro tout en ro

group :rw en rw

cgroup-full :mixed Un peu plus simplifié que cgroup :mixed, peut laisser fuiter des information dans le conteneur.

group-full :ro tout en ro

cgroup-full :rw tout en rw

lxc.rootfs Emplacement du système de fichier racine pour le conteneur. Peut être un fichier image, un répertoire, ou un périphériques block. Non spécifié, partage celui de l'hôte

lxc.rootfs.mount Où monter récursivement lxc.rootfs avant le pivot.

lxc.rootfs.options Options de montage supplémentaires pour rootfs

lxc.rootfs.backend Spécifie le type de backend rootfs à utiliser (dir, zfs,...).

lxc.cgroup.[subsystem name] Spécifie la valeur cgroup à définir.

lxc.cap.drop Capabilities à supprimer dans le conteneur, sans le préfix CAP.

lxc.cap.keep Spécifie la capabilitie à conserver dans le conteneur. none supprime toutes les capabilities

lxc.aa_profile Spécifie le profile apparmor sous lequel le conteneur devrait fonctionner. unconfined désactive apparmor. unchanged conserve le profile du conteneur parent.

lxc.aa_allow_incomplete les profiles apparmor sont basés sur des chemins. Donc des restrictions de fichier nécessitent des restrictions de montage. Cependant, ces restrictions de montage ne sont pas encore implémentées dans le kernel. À 0, le conteneur n'est pas démarré si le kernel n'a pas les fonctions de montage apparmor, donc un regression après une mise à jours kernel sera détectée.

lxc.se_context Spécifie le contexte SELinux sous lequel le conteneur devrait être lancé ou unconfined_t. (ex : system_u :system_r :lxc_t :s0 :c22)

lxc.seccomp Spécifie un fichier contenant la configuration seccomp à charger avant que le conteneur démarre.

lxc.id_map Un conteneur peut être démarré dans un namespace user privé avec un mappage. Contient 4 champs séparés par des espaces :

1 'u', 'g', ou 'b' pour both

2 L'UID/GID vu dans le namespace

3 L'UID/GID vu dans l'hôte

4 Le nombre d'uid/gid consécutifs à mapper

hooks

Les hooks de conteneur sont des programmes ou des scripts qui peuvent être exécutés à divers moments dans la durée de vie d'un conteneur. Quand un hook est exécuté, des informations sont passées en argument et via les variables d'environnement :

- Le nom du conteneur
- Section (toujours 'lxc')
- Type de hook (ex : 'clone' ou 'pre-mount')
- Arguments additionnels.

LXC_NAME Nom du conteneur

LXC_ROOTFS_MOUNT Chemin du rootfs monté

LXC_CONFIG_FILE Chemin du fichier de configuration de conteneur

LXC_SRC_NAME Dans le cas d'un hook clone, c'est le nom du conteneur original

LXC_ROOTFS_PATH valeur de lxc.rootfs pour le conteneur

lxc.hook.pre-start Un hook à lancer dans l'espace de nom de l'hôte avant que le ttys du conteneur, consoles ou mountages soient effectués.

lxc.hook.pre-mount Un hook à lancer dans l'espace de nom fs du conteneur avant que le rootfs ne soit up. Permet la manipulation du rootfs (ex : fs chiffré)

lxc.hook.mount Un hook à lancer dans l'espace de nom du conteneur après que le montage ait été effectué, mais avant le pivot_root

lxc.hook.autodev Un hook à lancer dans l'espace de nom du conteneur après que le montage ait été effectué, mais avant le pivot_root si lxc.autodev == 1. Permet d'assister le remplissage de /dev

lxc.hook.start Un hook à lancer dans l'espace de nom du conteneur immédiatement après l'exécution du init. Nécessite un programme dans le conteneur

lxc.hook.stop Un hook à lancer dans l'espace de nom de l'hôte après que le conteneur ait été éteind. Pour chaque espace de nom un argument est passé au hook contenant le type de namespace et un fichier qui peut être utilisé pour obtenir un fd vers le namespace correspondant.

lxc.hook.post-stop Un hook à lancer dans l'espace de nom de l'hôte après que le conteneur ait été stoppé.

lxc.hook.clone Un hook à lancer quand le conteneur est cloné.

lxc.hook.destroy Un hook à lancer quand le conteneur est détruit

Des variables d'environnement sont disponible aux hooks pour fournir des informations de configuration. Toutes les variables ne sont pas valides dans tous les contextes. En particulier, tous les chemins sont relatifs au système hôte et donc non valides durant lxc.hook.start

LXC_NAME Nom du conteneur

LXC_CONFIG_FILE Chemin du fichier de configuration de conteneur

LXC_CONSOLE Chemin de la console de sortie du conteneur si non null

LXC_CONSOLE_PATH Chemin de la console de sortie de log du conteneur si non null

LXC_ROOTFS_MOUNT Chemin du rootfs monté dans l'hôte

LXC_ROOTFS_PATH Valeur de lxc.rootfs pour le conteneur

LXC_SRC_NAME Dans le cas d'un hook clone, c'est le nom du conteneur original

LXC_TARGET Seulement pour le hook stop. Vaut stop ou reboot

LXC_CGNS_AWARE Non définis, cette version de lxc ne gère pas les namespaces cgroup. Ne garantit pas que ces namespaces sont activés dans le kernel.

lxc.loglevel Niveau de log de 0 (trace) à 8 (fatal)

lxc.logfile Fichier où logger les infos

lxc.start.auto (bool) Indique si le conteneur devrait être auto-démarré.

lxc.start.delay Délai d'attente avant que le conteneur suivant ne soit démarré

lxc.start.order Un entier utilisé pour trier les conteneurs durant l'auto-démarrage

lxc.monitor.unshare Si non 0, le namespace de montage sera non partagé avec l'hôte avant l'initialisation du conteneur (avant le hook pre-start). Nécessite CAP_SYS_ADMIN.

lxc.group Clé multivalué pour placer le conteneur dans un groupe.

lxc.environment Permet de passer des variables d'environnement au conteneur. Noter que ces variables sont visibles dans l'hôte dans /proc/PID/environ

Exemples de configuration

Exemple de configuration réseau :

```
lxc.utsname = myhostname
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.name = eth0
lxc.network.hwaddr = 4a:49:43:49:79:bf
lxc.network.ipv4 = 10.2.3.5/24 10.2.3.255
lxc.network.ipv6 = 2003:db8:1:0:214:1234:fe0b:3597
```

uid/gid mapping

```
lxc.id_map = u 0 100000 10000
lxc.id_map = g 0 100000 10000
```

cgroup

```
lxc.cgroup.cpuset.cpus = 0,1
lxc.cgroup.cpu.shares = 1234
lxc.cgroup.devices.deny = a
lxc.cgroup.devices.allow = c 1:3 rw
lxc.cgroup.devices.allow = b 8:0 rw
```

Exemple de configuration complexe :

```
lxc.utsname = complex
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = br0
lxc.network.hwaddr = 4a:49:43:49:79:bf
lxc.network.ipv4 = 10.2.3.5/24 10.2.3.255
lxc.network.ipv6 = 2003:db8:1:0:214:1234:fe0b:3597
lxc.network.ipv6 = 2003:db8:1:0:214:5432:feab:3588
lxc.network.type = macvlan
lxc.network.flags = up
lxc.network.link = eth0
lxc.network.hwaddr = 4a:49:43:49:79:bd
lxc.network.ipv4 = 10.2.3.4/24
lxc.network.ipv4 = 192.168.10.125/24
lxc.network.ipv6 = 2003:db8:1:0:214:1234:fe0b:3596
lxc.network.type = phys
lxc.network.flags = up
lxc.network.link = dummy0
lxc.network.hwaddr = 4a:49:43:49:79:ff
lxc.network.ipv4 = 10.2.3.6/24
lxc.network.ipv6 = 2003:db8:1:0:214:1234:fe0b:3297
```

```
lxc.cgroup.cpuset.cpus = 0,1
lxc.cgroup.cpu.shares = 1234
lxc.cgroup.devices.deny = a
lxc.cgroup.devices.allow = c 1:3 rw
lxc.cgroup.devices.allow = b 8:0 rw
lxc.mount = /etc/fstab.complex
lxc.mount.entry = /lib /root/myrootfs/lib none ro,bind 0 0
lxc.rootfs = /mnt/rootfs.complex
lxc.cap.drop = sys_module mknod setuid net_raw
lxc.cap.drop = mac_override
```