

Comment convertir une image KVM en conteneur LXC

Table of Contents

- Étape 1 : créer un nouveau conteneur lxc vide
- Étape 2 : copier l'image KVM dans le conteneur LXC
- Étape 3 : préparer les nœuds de périphérique du conteneur et corriger fstab
- Étape 4 : créer le fichier de configuration LXC
- Étape 5 : démarrer le conteneur
- Annexe 1 : améliorer le conteneur..

KVM (**K**ernel-based **V**irtual **M**achine) est une technologie de virtualisation Open Source intégrée à Linux. Avec **KVM**, on peut transformer Linux en un hyperviseur qui permet à une machine hôte d'exécuter plusieurs environnements virtuels isolés, appelés invités ou machines virtuelles, mais pour de nombreux cas d'utilisation, **LXC** est une alternative plus performante et légère.

En passant à **LXC**, on peut réduire l'utilisation globale de la mémoire - le principal avantage est que les processus au sein d'un conteneur **LXC** sont des processus distincts au sein du système hôte. Cela devrait permettre au système hôte de gérer la mémoire (cache, tampons, swap, etc.) plus efficacement.

Étape 1 : créer un nouveau conteneur lxc vide

Tout d'abord, créer un nouveau conteneur LXC vide avec une configuration par défaut, nommé par exemple 'imap':

```
lxc-create --name imap
```

Les conteneurs de LXC sont créés dans `/var/lib/lxc`, où un nouveau conteneur `/var/lib/lxc/imap` peut être trouvé. Le conteneur contient pour l'instant deux entrées : `config` pour sa configuration et un répertoire `rootfs` vide qui contiendra bientôt le système de fichiers du conteneur.

Étape 2 : copier l'image KVM dans le conteneur LXC

Chaque volume LVM reflète un disque dur physique, y compris les tables de partition, etc. Cela empêche de simplement monter l'image racine du KVM.

Tout d'abord, récupérer la table de partition de l'image KVM :

```
sudo fdisk -l /dev/vg0/imap.img
```

```
Disk /dev/vg0/imap.img: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00022ea3
```

	Device	Boot	Start	End	Blocks	Id	System
	/dev/vg0/imap.img1	*	2048	209713151	104855552	83	Linux

Ainsi, le véritable système de fichiers commence au secteur 2048 et chaque secteur a une longueur de 512 octets. Cela permet de calculer le décalage de ces partitions dans l'image et de monter la partition en boucle dans le système de fichiers hôtes :

```
mount -o loop,offset=$((2048*512)) /dev/vg0/imap.img /mnt-source
```

On peut maintenant copier le contenu de l'ancienne image KVM dans le nouveau conteneur :

```
rsync -av /mnt-source/* /var/lib/lxc/imap/rootfs
```

Étape 3 : préparer les nœuds de périphérique du conteneur et corriger fstab

LXC ne prend pas en charge **udev**, il faut donc créer les nœuds de périphérique du conteneur. Pour simplifier cela, on peut utiliser le script bash suivant et le copier dans `/usr/local/sbin/create-lxc-nodes.sh`

```
#!/bin/bash
ROOT=$(pwd)
DEV=${ROOT}/dev
mv ${DEV} ${DEV}.old
mkdir -p ${DEV}
mknod -m 666 ${DEV}/null c 1 3
mknod -m 666 ${DEV}/zero c 1 5
mknod -m 666 ${DEV}/random c 1 8
mknod -m 666 ${DEV}/urandom c 1 9
mkdir -m 755 ${DEV}/pts
mkdir -m 1777 ${DEV}/shm
mknod -m 666 ${DEV}/tty c 5 0
mknod -m 600 ${DEV}/console c 5 1
mknod -m 666 ${DEV}/tty0 c 4 0
mknod -m 666 ${DEV}/tty1 c 4 1
mknod -m 666 ${DEV}/tty2 c 4 2
mknod -m 666 ${DEV}/tty3 c 4 3
mknod -m 666 ${DEV}/tty4 c 4 4
mknod -m 666 ${DEV}/tty5 c 4 5
mknod -m 666 ${DEV}/tty6 c 4 6
mknod -m 666 ${DEV}/full c 1 7
mknod -m 600 ${DEV}/initctl p
```

```
mknod -m 666 ${DEV}/ptmx c 5 2
```

Utiliser ce script pour créer tous les nœuds de devices nécessaires :

```
cd /var/lib/lxc/imap/rootfs  
/usr/local/sbin/create-lxc-nodes.sh
```

Comme tous les systèmes de fichiers ont déjà été préparés par le système hôte, le système d'initialisation du système invité n'a pas besoin d'effectuer de travail pendant le démarrage (en fait, cela pourrait plutôt être nuisible). Pour éviter tout problème, commenter chaque ligne dans le fichier de configuration "etc/fstab" des invités.

Étape 4 : créer le fichier de configuration LXC

La configuration de chaque conteneur LXC est stockée dans le fichier "config" qui se trouve autour du répertoire "rootfs". Crée un nouveau :

```
# Modèle utilisé pour créer ce conteneur : (null)  
# Paramètres passés au modèle :  
lxc.mount.entry = proc proc proc nodev,noexec,nosuid 0 0  
lxc.mount.entry = sysfs sys sysfs defaults 0 0  
lxc.tty = 2  
lxc.pts = 1024  
lxc.cgroup.devices.deny = a  
lxc.cgroup.devices.allow = c 1:3 rwm  
lxc.cgroup.devices.allow = c 1:5 rwm  
lxc.cgroup.devices.allow = c 5:1 rwm  
lxc.cgroup.devices.allow = c 5:0 rwm  
lxc.cgroup.devices.allow = c 4:0 rwm  
lxc.cgroup.devices.allow = c 4:1 rwm  
lxc.cgroup.devices.allow = c 1:9 rwm  
lxc.cgroup.devices.allow = c 1:8 rwm  
lxc.cgroup.devices.allow = c 136:* rwm  
lxc.cgroup.devices.allow = c 5:2 rwm  
lxc.cgroup.devices.allow = c 254:0 rm  
lxc.utsname = imap  
lxc.network.type = veth  
lxc.network.flags = up  
lxc.network.link = virbr0  
lxc.network.hwaddr = 00:16:3e:f1:35:ab  
lxc.network.ipv4.gateway = 192.168.122.1  
lxc.network.ipv4 = 192.168.122.15  
lxc.cap.drop = sys_module  
lxc.cap.drop = mac_admin  
lxc.cap.drop = mac_override  
lxc.cap.drop = sys_time  
lxc.rootfs = /var/lib/lxc/imap/rootfs
```

- **lxc.mount.entry**: monte les systèmes de fichiers proc et sys dans l'invité (
- **lxc.cgroup.devices.allow**: autorise quelques périphériques (principalement des terminaux)
- **lxc.utsname**: définit le nom du conteneur
- **lxc.rootfs**: définit le chemin pour son système de fichiers racine (/var/lib/lxc/imap/rootfs).
- **lxc.network**: définit la configuration du réseau (ipv4 définit l'adresse IP à laquelle le conteneur sera attribué par le serveur DHCP interne). Il faut fournir une adresse MAC unique (hwaddr) à chaque conteneur.

Étape 5 : démarrer le conteneur

Démarrer le conteneur en arrière-plan

```
lxc-start --name imap -d
```

On peut alors s'y connecter en SSH

```
ssh 192.168.122.15
```

Annexe 1 : améliorer le conteneur..

Il y a des choses qui ne sont plus nécessaires. Toutes les commandes sont entrées dans le conteneur (qui est un système Debian 7.0 BTW) !

```
apt-get remove --purge acpid acpi  
update-rc.d -f hwclock.sh remove  
update-rc.d -f mountall.sh remove  
update-rc.d -f checkfs.sh remove  
update-rc.d -f udev remove
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=howto:convert-kvm-to-lxc>

Last update: **2025/02/19 10:59**

