LXC, Comment ca marche?

KostiTeam

October 24, 2016

Contents

1 La virtualisation

La virtualisation est le fait de créer une version virtuelle d'une entité physique, ces versions virtuelles sont alors appelées Machines virtuelles ou VM (Virtual Machine). Nous pouvons prendre comme exemple Virtual Box, qui est un outil de virtualisation permettant d'émuler un système d'exploitation (linux ou autre). Les différentes ressources de la machine hôte sont alors partagées et allouées dynamiquement aux différentes machines virtuelles par des logiciels appelés hyperviseur.

Ces machines virtuelles ont un défaut pour nous: elles simulent tout le materiel d'une machine (processeur, RAM...), ce qui les rend moins performante, et moins flexible au niveau du partage de ressources. Nous prefererons donc utiliser des VE (Virtual Environement) aux VM. Les VE permetttent de ne simuler que le systeme d'exploitation, et de partager le même noyeau qu la machine hôte, et donc, de répartir les ressources entre machine hôte et les differents environements virtuels. Cette solution est plus légere et flexible en terme de consomation et de partage de ressources.

LXC est un outil de virtualisation permettant de créer des environnements virtuelles, différents systemes d'exploitations sont mis a disposition (Ubuntu, Debian...). Ces Environements sont appelées containers. Le partage des ressources est assuré par l'outil Cgroups du noyau, qui permet de limiter, compter et isoler l'utilisation des ressources.

Chaque environnement virtuel est isolée, de la même manière que l'isolement d'un programme avec "*chroot*": chaque environnement est créé de manière à ce qu'ils naient pas accès au système d'exploitation de la machine hôte. En revanche, la machine hôte, elle, a accès aux machines virtuelles. Cette isolation entre les machines virtuelles et la machine hôte, permet de garantir une certaine sécurité.

2 Les containers, leur fonctionnement

Les environnements virtuels, ou, containers (conteneurs en francais) doivent, en premier temps, être créés à l'aide de la commande "*lxc-create ...*" (voir notices pour plus de précisions sur les commandes). De nombreux systèmes d'exploitation seront alors proposés, (nous avons choisi de prendre Debian, Jessie, i386). Par défaut, les machines créées ne sont pas configurées: elles n'ont pas d'interfaces, elles n'ont pas de compilateur, et les utilitaires préinstallés sont très rudimentaire (pas de ping, ifconfig, tcpdump...).

Chaque container possède un fichier de configuration situé à l'emplacement suivant: "*/var/lib/lxc/<nom du container>/config*". Il est possible de configurer de nombreux aspect du container dans ce fichier (par exemple: modifier

les variables d'environnement, ou changer le nom d'hôte ("*hostname*") du container. Voir "*man lxc*" pour en savoir plus). Ce fichier va notamment permettre de paramétrer la configuration réseau du container à son démarrage (c'est ce qui va nous intéresser en priorité avec ce fichier).

Les containers se lancent avec la commande "*lxc-start ...*"; il est préférable de les lancer en démons (en arrière-plan), puis d'y "*attacher*" un terminal avec "*lxc-attach*", afin d'être connecté en super utilisateur (ou root) car, premièrement, par défaut, aucun profil d'utilisateur n'est créé sur le container, et, de plus, cela permet d'avoir l'entier contrôle du container afin de, par exemple, modifier son adresse ipv4.

3 Bridges et fonctionnement réseau

3.1 fichier config

Comme expliqué plus haut, les paramètres réseaux du container peuvent être modifiés via le fichier config. Dans ce fichier, chaque "*block*" permet de définir une interface. Un block commence toujours par la définition du type de réseau, nous allons donc nous intéresser tout d'abord au type de réseau ("*lxc.network.type*"). Plusieurs types de réseaux sont disponibles, mais, nous allons choisir le type veth, qui signifie Virtual Ethernet, ce type permet d'établir un lien entre l'interface virtuelle du container et un pont, préalablement créé sur la machine host (nous verrons cette liaison plus en détail par la suite).

Ce fichier permet aussi d'établir des adresses ipv4 ("*lxc.network.ipv4*"), ipv6 ("*lxc.network.ipv6*"), et mac ("*lxc.network.hwaddr*"), ainsi que leurs Broadcast. Cela permet de mettre en place le réseau virtuel avant de lancer les machines; bien que ces paramètres puissent être modifiés une fois la machine lancée à l'aide de l'outil "*ifconfig*". Attention, les tables de routage ne peuvent pas être configurées d'avance, il faut les configurer une fois le VE lancé.

Le dernier paramètre que nous verrons pour ce fichier est le lien ("*lxc.network.link*"); ce paramètre va permettre d'indiquer à quel bridge nous voulons connecter notre interface.

3.2 Bridges

Les bridges (ou pont en francais) sont des équipements réseaux qui permettent de relier deux (ou plus) interfaces de manière complètement transparente: en observant les paquets qui transitent, le pont peut connaitre les adresses mac des interfaces, et ainsi, rediriger les paquets. Les ponts peuvent par exemple, être utilisés pour rediriger une connexion Ethernet: une machine se connecte

en Ethernet, une seconde machine se connecte à la première, elles établissent un pont entre deux de leurs interfaces (une interface de la première machine, et une interface de la deuxième machine), et, ainsi, la deuxième machine peut avoir accès à internet.

Lorsqu'un VE se lance, une interface se créé sur la machine hôte pour chaque interface présente sur le VE. Ces interfaces créées ont un noms qui commencent toujours par "*VETH*" suivit de quatre caractères. Ces interfaces sur la machine hôte representent les interfaces de l'environement virtuels, et vons nous permettre des reliers les environnements entre eux; il est possible de voir la correspondance entre les interfaces d'un container et les interfaces de la machine hôte grace a la commande "*lxc-info ...*".

Il est donc necessaire de relier ces interfaces sur la machine hôte a un même bridge pour que les containers puissent communiquer!