

LXC Debian Jessie

Virtualisation facile !

samedi 5 septembre 2015, par [Pierrox](#)

Installation de LXC

1. **apt-get install** lxc lxcctl

Création d'un container

1. `lxc-create -n NomContainer -t debian`

- -n nom du container
- -t template de l'OS à utiliser

Bien noter le mot de passe root qui s'affichera à la fin de la création du container lxc.

Configuration du réseau sur l'hôte

Il y a plusieurs façons de configurer le réseau, pour ma part je préfère créer un (ou plusieurs) sous-réseaux pour mes containers qui seront "natter" par la machine hôte.

Pour illustrer, on pourrait dire que les sous-réseaux sont cloisonnés à l'intérieur de l'hôte qui fera office de pare-feu vers l'extérieur.

Autoriser "l'ip forwarding"

1. **nano** /etc/sysctl.conf
2. `net.ipv4.ip_forward=1`

[Télécharger](#)

Recharger la configuration du système pour prendre en compte la modification.

1. `sysctl -p`

Configurer l'interface qui servira de bridge

La paquet bridge-utils est nécessaire, il est normalement installé en dépendance de lxc, au cas ou :

1. **apt-get install** bridge-utils

Paramétrage de /etc/network/interfaces

```
# The loopback network interface
auto lo
iface lo inet loopback

# Configuration du Bridge pour le sous réseau virtualisé
auto br0
iface br0 inet static
    address 192.168.100.254
    netmask 255.255.255.0
    bridge_ports none
```

Redémarrer le service réseau afin de prendre en compte la nouvelle interface br0

1. service networking restart

Vérifier que l'interface réseau est bien présente :

`ifconfig` devrait renvoyer quelque chose comme ça

1. br0 Link encap:Ethernet HWaddr 22:ce:06:00:1d:7f
2. inet adr:192.168.100.254 Bcast:192.168.100.255 Masque:255.255.255.0
3. UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
4. RX packets:0 errors:0 dropped:0 overruns:0 frame:0
5. TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
6. collisions:0 lg file transmission:0
7. RX bytes:0 (0.0 B) TX bytes:3915 (3.8 KiB)

[Télécharger](#)

Activer le partage de connexion internet pour les containers LXC :

On va donner l'accès internet aux containers LXC à travers wlan0 (On verra comment automatiser ça sur un hôte au profil itinérant qui se connecte sur des interfaces diverses et variées : wlan0 eth0 usb0...)

1. iptables -t nat -F POSTROUTING
2. iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE

[Télécharger](#)

Configuration du réseau sur les containers LXC

Attention ! il ne s'agit pas de paramétrer le réseau à l'intérieur du container mais d'éditer un fichier de configuration qui se chargera de configurer le réseau au démarrage de celui-ci.

Par défaut, Debian installe les nouveaux containers dans **/var/lib/lxc** ; ce dossier contient tous les

containers créés, qui eux-même contiennent des fichiers de logs, de configs et le dossier **rootfs** qui comme son nom l'indique, correspond à la racine du système virtualisé (/).

Afin de configurer le réseau sur notre container LXC, nous allons éditer son fichier de config **/var/lib/lxc/NomContainer**

Extrait de la configuration réseau d'un container **/var/lib/lxc/NomContainer/config**

1. `lxc.network.type = veth`
2. `lxc.network.name = veth0`
3. `lxc.network.flags = up`
4. `lxc.network.link = br0`
5. `lxc.network.veth.pair = veth0`
6. `lxc.network.ipv4 = 192.168.100.1/24`
7. `lxc.network.ipv4.gateway = 192.168.100.254`

[Télécharger](#)

Démarrage du container et test du réseau

1. `lxc-start -n NomContainer -d`

- `-n` nom du container
- `-d` en mode daemon

La commande pour contrôler si notre container LXC est bien démarré :

1. `lxc-ls -f`

Devrait retourner

1. NAME STATE IPV4 IPV6 AUTOSTART
2. -----
3. NomContainer RUNNING 192.168.100.1 - NO

[Télécharger](#)

Sur l'hôte un petit ping 192.168.100.1 devrait retourner :

1. PING 192.168.100.1 (192.168.100.1) 56(84) bytes of data.
2. 64 bytes from 192.168.100.1: `icmp_seq=1 ttl=64 time=0.087 ms`

[Télécharger](#)

On va se connecter directement au container LXC

1. lxc-console -n NomContainer

Se connecter au container, j'espère que vous avez bien noté le mot de passe root lors de sa création sinon vous pouvez toujours [consulter l'article dédié au changement de mot de passe root d'un container LXC](#).

Une fois connecté, on va tester l'accès internet en utilisant apt pour installer ping afin de "pinguer" la machine hôte :

1. root@NomContainer:~#**apt-get install** inetutils-ping
2. Les NOUVEAUX paquets suivants seront **installés** :
3. inetutils-ping
4. (...)
5. Paramétrage de inetutils-ping (2:1.9.2.39.3a460-3) ..
6. root@NomContainer:~#**ping** 192.168.100.254
7. PING 192.168.100.254 (192.168.100.254): 56 data bytes
8. 64 bytes from 192.168.100.254: **icmp_seq=0 ttl=64 time=0,163 ms**
9. 64 bytes from 192.168.100.254: **icmp_seq=1 ttl=64 time=0,163 ms**

[Télécharger](#)

Et voilà, Juste ça marche !

P.-S.

Pour quitter un container, utilisez la combinaison de touche `ctrl+a` puis `q`