

Installation de LXC

Projet m2l.org

Table des matières

Les conteneurs	3
Qu'est-ce qu'un conteneur ?.....	3
Qu'est-ce que LXC ?.....	3
Installation de LXC et reconfiguration du réseau	4
Qu'est-ce qu'un bridge ?	4
Installation de LXC.....	4
Création du conteneur template.....	6
Création du conteneur web.....	8
Commandes LXC utiles	10



Les conteneurs

Qu'est-ce qu'un conteneur ?

Un conteneur est un type de technologie de virtualisation.

Les conteneurs sont des zones isolées les unes des autres qui ont pour rôle de stocker des applications. Ils partagent le système d'exploitation hôte mais ont leur propre espace de fichiers, leur propre mémoire et leurs propres processus, ce qui les rend indépendants les uns des autres et par conséquent faciles à déployer, à échanger et à gérer.

Qu'est-ce que LXC ?

LXC (ou Linux Containers) est une plateforme de conteneurs Open Source qui fournit un ensemble d'outils, de modèles de bibliothèques et de liaisons de langage. LXC dispose d'une interface en ligne de commande simple qui facilite la prise en main pour les nouveaux utilisateurs.

Il contient un environnement de virtualisation au niveau du système d'exploitation qu'il est possible d'installer sur de nombreux systèmes basés sur Linux. Il est possible d'y accéder dans le référentiel du package de distributions Linux.



Installation de LXC et reconfiguration du réseau

Pour installer des conteneurs, nous allons devoir faire l'installation des outils LXC nécessaires et créer un bridge qui reliera notre serveur à nos conteneurs

Qu'est-ce qu'un bridge ?

Lorsqu'un conteneur est créé, par défaut, il est configuré pour être dans un réseau isolé, ce qui signifie qu'il ne peut pas accéder directement aux ressources du réseau externe. Un bridge permet donc aux conteneurs de communiquer avec d'autres machines ou réseaux externes en dehors de l'hôte.

Un bridge, ou pont réseau, agit comme une passerelle entre le réseau externe et les conteneurs. Il permet aux conteneurs d'obtenir une adresse IP sur le réseau local et de partager la connexion réseau de l'hôte. De cette manière, les conteneurs peuvent communiquer avec d'autres machines sur le réseau, accéder à Internet, et être accessibles depuis d'autres ordinateurs.

Installation de LXC

Pour installer l'outil LXC, nous entrerons les commandes suivantes :

```
apt update  
apt upgrade  
apt install lxc
```

Nous vérifions ensuite la configuration :

```
lxc-checkconfig
```

Pour créer le pont nécessaire à la communication entre les conteneurs et le serveur, nous installons l'outil bridge-utils :

```
apt install bridge-utils
```

Nous pouvons maintenant créer notre pont :



```
brctl addbr br0
```

Nous vérifions que le pont soit créé :

```
root@srv-g6:~# brctl show
bridge name      bridge id                STP enabled  interfaces
br0              8000.a2e124b08e14        no           eno1
                veth5VHN0w
```

```
root@srv-g6:~# ifconfig -a
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.31.96.1 netmask 255.255.240.0 broadcast 10.31.111.255
    inet6 fe80::a0e1:24ff:feb0:8e14 prefixlen 64 scopeid 0x20<link>
    ether a2:e1:24:b0:8e:14 txqueuelen 1000 (Ethernet)
    RX packets 135605 bytes 118811242 (113.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 93109 bytes 12389045 (11.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nous devons maintenant faire en sorte que le pont joue le rôle de carte réseau en lui donnant l'adresse IP du serveur et en enlevant cette dernière à la carte réseau :

```
ifconfig eno1 0.0.0.0
ifconfig br0 10.30.96.1/20 up
route add default gw 10.31.96.254
brctl addif br0 eno1
```

Afin que les modifications soient effectuées à chaque redémarrage du serveur, nous allons modifier son fichier /etc/rc.local :

```
#!/bin/sh -e

brctl addbr br0

ifconfig eno1 0.0.0.0
ifconfig br0 10.31.96.1/20 up
route add default gw 10.31.96.254
brctl addif br0 eno1

echo "nameserver 8.8.8.8" > /etc/resolv.conf

exit 0
```

La configuration du pont est maintenant terminée.



Création du conteneur template

Nous allons créer un conteneur template sur lequel on souhaite installer tous les outils par défaut de nos conteneurs afin de pouvoir de copier et modifier pour les conteneurs à venir. Pour ce faire, nous allons le créer à partir d'une image de Debian 11 appelé Bullseye :

```
host# lxc-create -n template -t debian -- -r bullseye
```

```
root@srv-g6:~# lxc-ls  
template web
```

Attention : Lorsque les commandes commencent par host#, il faut les exécuter directement dans le serveur. Si elles commencent par name#, il faut les exécuter dans le conteneur en question.

Nous démarrons ensuite notre conteneur :

```
host# lxc-start template
```

Pour accéder à notre conteneur, nous devons entrer la commande :

```
host# lxc-attach template
```

Nous allons attribuer une IP temporaire et une route par défaut à notre conteneur pour pouvoir, par la suite, installer des outils :

```
template# ifconfig eth0 10.31.96.2 up  
template# route add default gw 10.31.96.254
```

Pour que l'IP et la route par défaut soient configurées au lancement du conteneur, nous allons les insérer dans le fichier /etc/network/interfaces :

```
template# nano /etc/network/interfaces :
```



```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.31.96.2/20
    gateway 10.31.96.254
    dns-nameservers 8.8.8.8
```

```
root@template:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.31.96.2 netmask 255.255.240.0 broadcast 10.31.111.255
    inet6 fe80::216:3eff:fee4:90d5 prefixlen 64 scopeid 0x20<link>
    ether 00:16:3e:e4:90:d5 txqueuelen 1000 (Ethernet)
    RX packets 53 bytes 10359 (10.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 936 (936.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Nous vérifions que notre conteneur ait accès à internet avec la commande suivante :

```
root@template:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=12.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=12.8 ms
```

Nous allons maintenant mettre le conteneur à jour et installer les outils nécessaires :

```
template# apt update
template# apt upgrade
template# apt install sudo net-tools tcpdump nano iputils-ping dbus
```

Nous modifions la timezone :

```
template# ln -fs /usr/share/zoneinfo/Europe/Paris /etc/localtime
template# dpkg-reconfigure -f noninteractive tzdata
```

Nous créons un nouvel utilisateur qui aura les autorisations sudo :

```
template# adduser sio
template# usermod -a -G sudo sio
```

La configuration du conteneur template est terminée. Nous pouvons maintenant l'éteindre.



Création du conteneur web

Le conteneur web sera notre conteneur de sites internet. Pour le créer, nous commençons par copier le conteneur template en conteneur web :

```
host# lxc-copy -n template -N web
```

Nous changeons ensuite les paramètres du conteneur pour que ce dernier se lance automatiquement au redémarrage du serveur :

```
host# lxc.start.auto = 1
```

Nous pouvons entrer cette directive dans le fichier `/var/lib/lxc/web/config` pour la rendre permanente. Ensuite, nous pouvons répéter la configuration du serveur template pour la configuration IP en changeant l'adresse IP par `10.31.96.80/20`.

```
root@web:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.31.96.80 netmask 255.255.240.0 broadcast 10.31.111.255
    inet6 fe80::216:3eff:fe9f:4ebe prefixlen 64 scopeid 0x20<link>
    ether 00:16:3e:9f:4e:be txqueuelen 1000 (Ethernet)
    RX packets 30288 bytes 8788433 (8.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 192 bytes 14295 (13.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Une fois le conteneur configuré, nous vérifions que le conteneur puisse se connecter à Internet, puis nous installons Apache dessus :

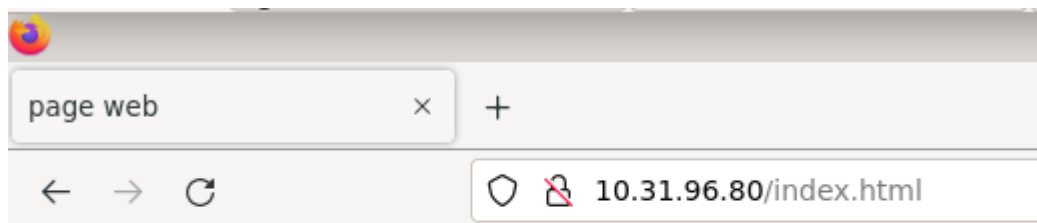
```
web# ping 8.8.8.8
web# apt install apache2
```

```
root@web:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=13.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=12.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=12.8 ms
```




```
ii  apache2                2.4.54-1~deb11u1
ii  apache2-bin             2.4.54-1~deb11u1
ii  apache2-data             2.4.54-1~deb11u1
ii  apache2-utils            2.4.54-1~deb11u1
```

Nous modifions ensuite la page HTML du site internet par défaut (/var/www/html/index.html) par un site personnalisé et nous testons depuis le serveur si nous avons accès à ce dernier (10.31.96.80/index.html) :



Ceci est une page web.



Commandes LXC utiles

`lxc-start [-n] <conteneur>` : démarrer un conteneur

`lxc-stop [-n] <conteneur>` : éteindre un conteneur

`lxc-copy -n <conteneur> -N <nouveau_conteneur>` : copier un conteneur

`lxc-destroy [-n] <conteneur>` : supprimer un conteneur

`lxc-checkconfig` : vérifier le kernel actuel d'LXC

`lxc-attach [-n] <conteneur>` : accéder à un conteneur

`lxc-info [-n] <conteneur>` : afficher les informations d'un conteneur

`lxc-ls` : lister les conteneurs créés

Astuce : Pour sortir d'un conteneur et revenir sur le serveur, il suffit de faire CTRL + D

