

Sommaire

Installation de LXC et bridge-utils
Installation d'un pont virtuel
Création d'un conteneur type
Configuration du conteneur

LXC et bridge-utils

Sur la machine hôte il faut dans un premier temps mettre à jours les packets, installer lxc puis installer bridge-utils avec les commandes :

```
apt-get update
apt-get upgrade
apt-get install lxc
apt-get install bridge-utils
```

L'outil lxc permet de créer des conteneur.

L'outil bridge-utils quand à lui permet de créer un pont virtuel entre le réseau physique et le réseau virtuel que nous créerons pour y mettre des conteneurs.

Configuration du pont virtuel

Après avoir fait l'installation de bridge-utils il faut configurer le pont virtuel afin que nous puissions avoir accès aux conteneurs depuis l'extérieur.

Pour que cela soit permanent il faut dans un premier modifier le fichier rc.local avec la commande

```
nano /etc/rc.local
```

Puis une fois dedans il faut faire la création du pont virtuel avec "brctl addbr br0" puis enlever l'adresse IP a l'interface réseau physique avec "ifconfig eno1 0.0.0.0", il faut ensuite donner l'adresse IP de l'interface réseau au pont virtuel avec "ifconfig br0 10.31.80.254/20 up" puis ajouter la passerelle par défaut avec "route add default gw 10.31.80.254". Après cela fait il faut dire que le pont virtuel est "lié" à l'interface réseau physique avec "brctl addif br0 eno1". L'avant-dernière ligne est pour le nom de domaine. La dernière ligne est pour retourner le 0 a la sortie c'est plus propre.

```
brctl addbr br0
ifconfig eno1 0.0.0.0
ifconfig br0 10.31.80.254/20 up
route add default gw 10.31.80.254
brctl addif br0 eno1
echo "nameserver 8.8.8.8"> /etc/resolv.conf
exit 0
```

nous pouvons vérifier si notre pont c'est bien créer avec les commandes.

```
brctl show  
ifconfig
```

Création et configuration du container template

dans un premier temps nous allons créer un conteneur qui nous servira par la suite de modèle que nous pourrons par la suite dupliquer pour créer autant de conteneur que le réseau nous permet de faire. Il faudra cependant bien penser à changer l'adresse IP après avoir fait la duplication.

Pour la création du conteneur template nous utiliserons la commande

```
lxc-create -n template -t debian -- -r bullseyes
```

notre conteneur template est maintenant créé. Il faut maintenant le démarrer avec

```
lxc-start template
```

on fait un

```
lxc-info template
```

pour vérifier l'état du conteneur et voir toutes les infos nécessaires comme le nom, le statut, l'IP et d'autres infos utiles. À ce stade le conteneur est démarré mais si l'on redémarre le serveur il ne se redémarrera pas il faut donc ajouter les commandes dans le fichier config avec la commande

```
nano /var/lib/lxc/XXX/config  
#XXX : nom du conteneur (ici template)
```

Mais cette méthode nous oblige à le faire à chaque fois que nous dupliquons le conteneur template. Nous pouvons donc directement modifier le fichier `/var/lib/lxc/template/config` afin que lorsque nous dupliquons le template le prochain conteneur est déjà cette config. Il faut ensuite ajouter dans ce fichier la directive suivante

```
lxc.start.auto = 1
```

Lorsque tout cela est fait l'on peut accéder au conteneur pour finir de le configurer correctement, pour cela nous utilisons la commande

```
lxc-attach template
```

Nous nous trouvons donc dans le conteneur. commençons par configurer son adresse IP de manière permanente en inscrivant les commandes dans le fichier `/etc/network/interfaces`.

```
nano /etc/network/interfaces  
#les commandes ci après sont à écrire dans le fichier  
auto lo
```

```
iface lo inet loopback

auto eth0
iface eth0 inet static #par défaut c'est dhcp il faut bien pensé a
passer cette ligne en static sinon l'IP ne sera pas fixe
    address 10.31.80.2/20
    gateway 10.31.80.254
    dns-nameservers 8.8.8.8
```

Maintenant que l'adresse IP est configuré et que le conteneur à accès à internet on peut faire l'installation des outils nécessaire avec les commandes

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

on peut maintenant configurer de l'heure sur le conteneur template

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

Il faut maintenant configurer le sudo si il y a besoin de ssh dans le conteneur

```
adduser std
usermod -a -G sudo std
```

Il faut maintenant faire la vérification du conteneur template

```
ctrl d
lxc-stop template
lxc-info template
ssh sio@10.31.80.2
password
sudo -i
```

Duplication du container template en container web

Si tout est bon alors on peut copier le conteneur template en conteneur web grâce à la commande

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

Il faut alors configurer le conteneur web, pour cela nous rentrons dans le conteneur puis on change l'adresse IP

```
lxc-attach web
nano /etc/network/interfaces
auto lo
```

```
iface lo inet loopback
auto eth0
iface eth0 inet static
address 10.31.80.80/20
gateway 10.31.80.254
dns-nameservers 8.8.8.8
```

Après avoir modifier l'IP on peut faire l'installation d'Apache2 dans notre conteneur web afin de le transformer en server, pour cela il faut faire

```
apt install apache2
```

Afin de pouvoir vérifier si tout fonction correctement il faut d'abord supprimer la page index.html qui se trouve dans /var/www/html puis créer une page web dans le répertoire /var/www/html en faisant

```
nano /var/www/html/index.html
#faire un peu de html
```

L'on se rend ensuite sur un navigateur on tape dans la barre de recherche l'IP du conteneur/index.html si l'on voit se que l'on a écrit alors tout est bon et on a fini!!!!

From:

<https://sisr2.beaupeyrat.com/> - Documentations SIO2 option SISR

Permanent link:

<https://sisr2.beaupeyrat.com/doku.php?id=sisr1-g5:mission1.2>

Last update: **2024/03/08 11:46**

