Mission 1.2 : Installation et Configuration de Conteneurs LXC

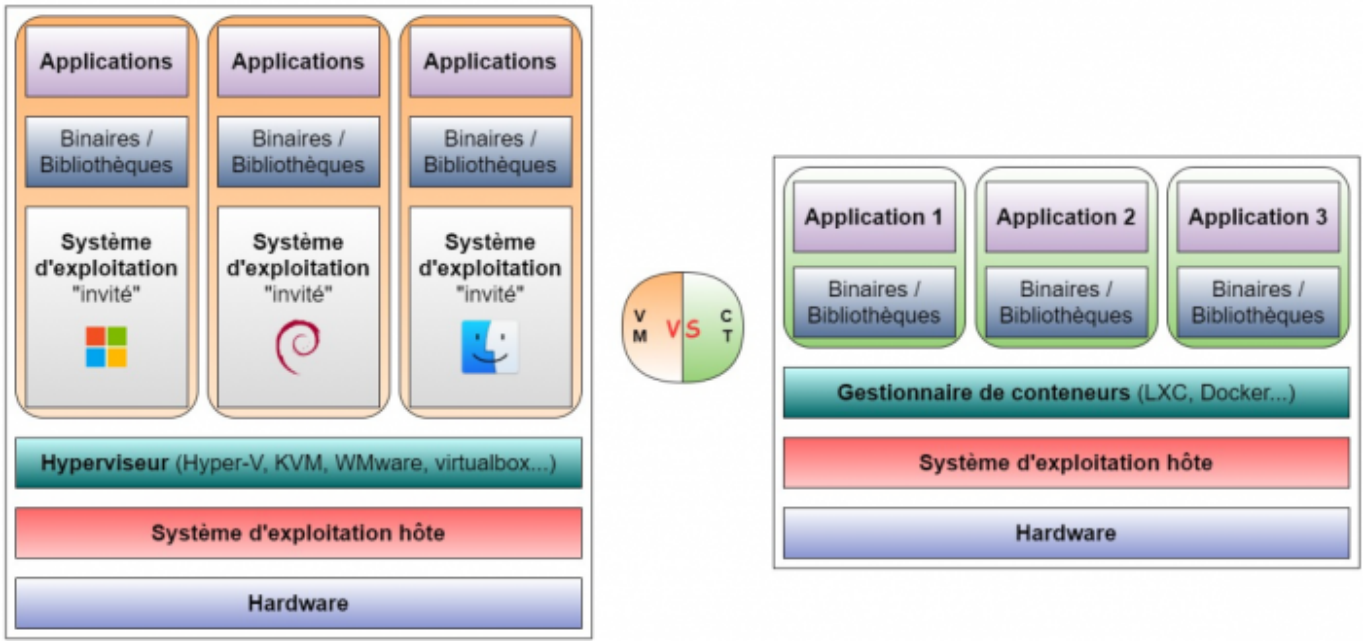
0. Pré-requis

Afin de pouvoir réaliser cette mission il faut :

- Un serveur sous Debian 12
- Un routeur sous Debian 12
- La configuration IP des deux machines conforme au cahier des charges,
- L'accès SSH par clés depuis la salle de cours au serveur et au routeur.

1. Introduction

L'objectif de cette mission est de mettre en place un système de conteneurisation sur le serveur de l'association m2l.
Nous utiliserons pour cela LXC (LinuX Containers), afin de pouvoir héberger les futurs services serveurs dans des conteneurs isolés les uns des autres et ayant leur propre adressage IP.



Contrairement à la virtualisation lourde que nous avons déjà utilisée au premier semestre avec notamment VirtualBox, les conteneurs sont une forme de virtualisation légère pour lequel c'est le système d'exploitation qui est virtualisé et non la machine entière. Cela signifie comme on peut le voir dans le schéma qu'il n'est pas nécessaire d'installer un système d'exploitation dans un conteneur à la différence des machines virtuelles. C'est le système d'exploitation de la machine hôte qui est utilisé par les conteneurs.

Par rapport aux machines virtuelles, les conteneurs sont :

- Moins volumineux,
- Plus rapide,
- Moins gourmands en ressources,
- Mais un peu moins bien isolés.

2. Mise en Place de LXC

2.0. Mémo LXC

Voici quelques commandes lxc qui seront très utiles pour manipuler les conteneurs.

Mémo : Commandes utiles pour LXC

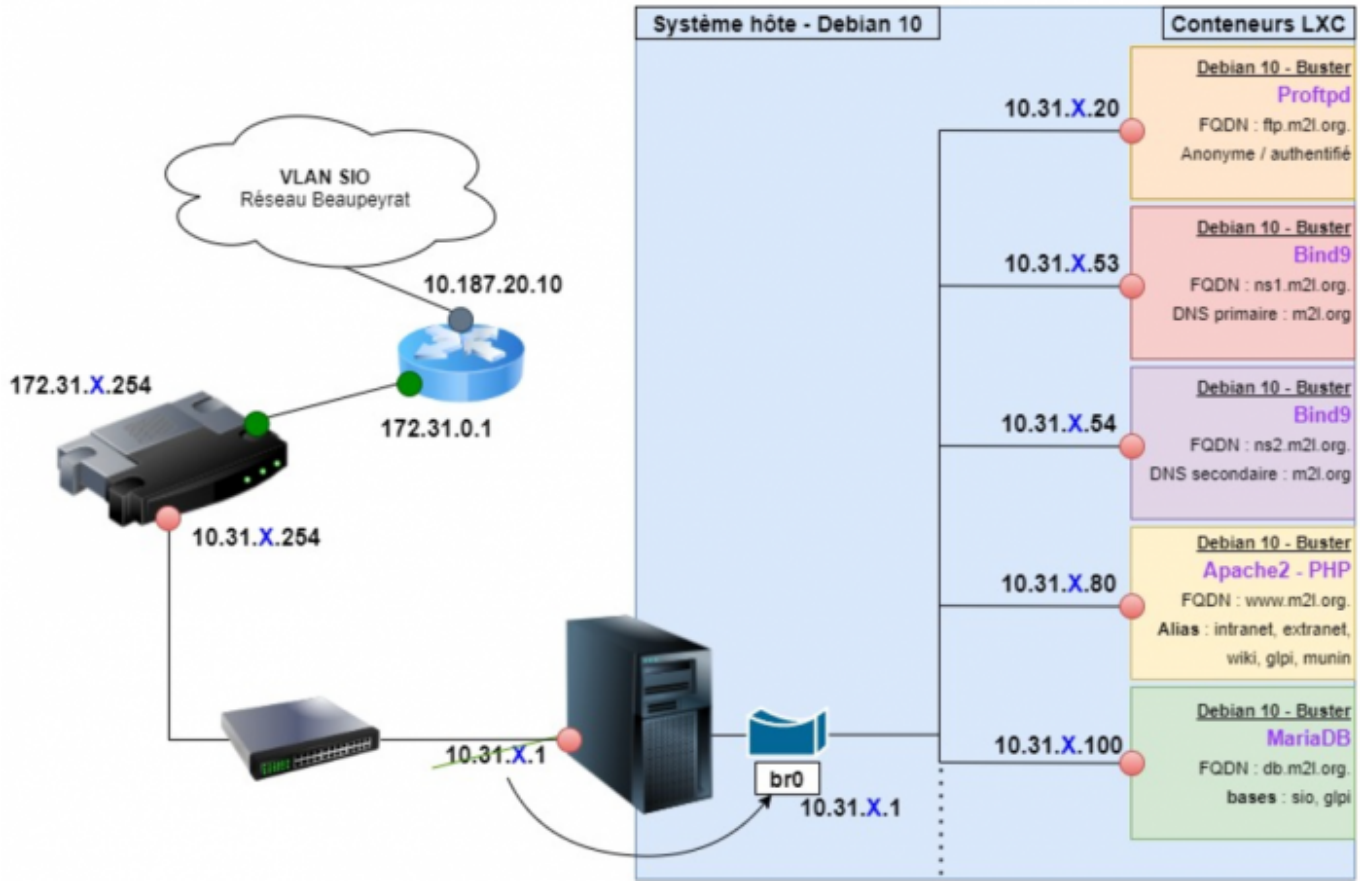
- **lxc-start [-n] <container>**
  - lxc-start template
- **lxc-stop [-n] <container>**
  - lxc-stop template
- **lxc-copy -n <container> -N <nouveau_container>**
  - lxc-copy -n template -N web
- **lxc-destroy [-n] <container>**
  - lxc-destroy web
- **lxc-checkconfig**
- **lxc-attach [-n] <container>**
  - lxc-attach template
- **lxc-info [-n] <container>**
  - lxc-info template
- **lxc-ls**

2.1. Préambule

Les conteneurs sont placés sur le même réseau « public » que l'hôte (notre serveur) (10.31.16.0/20)

- Comme pour le serveur, leur accès à Internet est assuré, par le routeur.
- L'accès aux services qu'ils offrent depuis l'extérieur est assuré grâce à leurs adresses IP

Schéma du réseau



Pour réaliser la configuration réseau telle que présentée dans le schéma ci-dessus, il est nécessaire de créer un pont (***bridge***) virtuel pour interconnecter les différents conteneurs ainsi que l'hôte.

> **Outil**
>
> Pour le bridge, nous utiliserons l'outil **bridge-utils**

```
PS C:\WINDOWS\system32> ssh root@10.31.80.1
Enter passphrase for key 'C:\Users\HP/.ssh/id_rsa':
Linux srv-g5 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 13 11:43:22 2025 from 10.31.95.254
root@srv-g5:~# ifconfig
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.31.80.1  netmask 255.255.240.0  broadcast 10.31.95.255
        inet6 fe80::e5:a2ff:fef1:65a8  prefixlen 64  scopeid 0x20<link>
        ether 02:e5:a2:f1:65:a8  txqueuelen 1000  (Ethernet)
        RX packets 123  bytes 14591 (14.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 81  bytes 13972 (13.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::4a0f:cfff:fe43:977f  prefixlen 64  scopeid 0x20<link>
        ether 48:0f:cf:43:97:7f  txqueuelen 1000  (Ethernet)
        RX packets 123  bytes 16805 (16.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 93  bytes 16216 (15.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xe1000000-e1020000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Boucle locale)
```

2.2. Installation de LXC et bridge-utils

Je commence par mettre à jour et installer les paquets nécessaires sur l'hôte :

```
# On met à jour la liste des paquets
root@srv-g1:~# apt update
# On met à jour les paquets installés
root@srv-g1:~# apt upgrade
# La commande suivante permet d'installer les paquets lxc et bridge-utils
depuis les dépôt Debian
root@srv-g1:~# apt install lxc bridge-utils
```

Mémo : Bridge-utils

La commande **brctl** (bridge control) est utilisée pour gérer les ponts réseau (network bridges) sur les systèmes basés sur Linux. Voici les principales sous-commandes disponibles :

- **brctl addbr <bridge>**
  - Crée un nouveau pont réseau avec le nom spécifié.
  - Exemple : brctl addbr br0
- **brctl delbr <bridge>**
  - Supprime un pont réseau existant.
  - Exemple : brctl delbr br0
- **brctl addif <bridge> <interface>**
  - Ajoute une interface réseau au pont spécifié.
  - Exemple : brctl addif br0 eth0
- **brctl delif <bridge> <interface>**
  - Retire une interface réseau du pont spécifié.
  - Exemple : brctl delif br0 eth0
- **brctl show**
  - Affiche la liste des ponts existants et les interfaces qui y sont connectées.
  - Exemple : brctl show

Vérification de l'installation de LXC

2.3. Configuration Réseau

Pour réaliser la configuration réseau (incluant donc le nouveau bridge), je **modifie** le fichier rc.local du serveur.

```
# On édite le fichier rc.local avec nano
root@srv-g1:~# nano /etc/rc.local
```

rc.local

```
 #!/bin/sh -e
 # On crée le bridge virtuel
 brctl addbr br0
# On enlève l'adresse IP de l'interface physique du serveur
 ifconfig eno1 0.0.0.0
# On enlève l'adresse IP de l'interface physique du serveur
 ifconfig br0 10.31.80.1/20
# On connecte l'interface physique eno1 du serveur à notre bridge
brctl addif br0 eno1
# On définit la passerelle par défaut du serveur /!\ Il s'agit de notre
```

```
routeur
# On utilise le resolver public de google pour la résolution de noms
echo "nameserver 8.8.8.8" > /etc/resolv.conf route add defaut gw
10.31.95.254

exit 0
```

callout type=“warning” title=“Attention : Application des modifications” icon=“glyphicon glyphicon-exclamation-sign”> Pour appliquer les modifications faites, il faut **rebooter** la machine. </callout>

Vérifications

```
root@srv-g1:~# ifconfig
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.31.16.1  netmask 255.255.240.0  broadcast 10.31.31.255
        inet6 fe80::68fd:78ff:feff:66bf  prefixlen 64  scopeid 0x20<link>
        ether 6a:fd:78:ff:66:bf  txqueuelen 1000  (Ethernet)
        RX packets 75666  bytes 69845021 (66.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 62421  bytes 12962920 (12.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::4a0f:cfff:fe43:9793  prefixlen 64  scopeid 0x20<link>
        ether 48:0f:cf:43:97:93  txqueuelen 1000  (Ethernet)
        RX packets 333733  bytes 396657068 (378.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 198841  bytes 38016420 (36.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xe1000000-e1020000
root@srv-g1:~# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
0.0.0.0         10.31.31.254    0.0.0.0         UG    0      0        0 br0
10.0.3.0        0.0.0.0         255.255.255.0   U     0      0        0 lxcbr0
10.31.16.0      0.0.0.0         255.255.240.0   U     0      0        0 br0
root@srv-g1:~# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.6afd78ff66bf       no              eno1
                                                        veth0WileE
                                                        vethOhhAFP
                                                        vethso3jjK
root@srv-g1:~# ping -c2 www.sautour.fr
PING phoenix.sautour.fr (51.159.34.166) 56(84) bytes of data.
64 bytes from sautour.fr (51.159.34.166): icmp_seq=1 ttl=47 time=36.3 ms
64 bytes from sautour.fr (51.159.34.166): icmp_seq=2 ttl=47 time=36.5 ms

--- phoenix.sautour.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 36.281/36.391/36.502/0.110 ms
```

Résultats

- L'adresse IP et le masque de sous-réseau sont conformes,
- La passerelle par défaut est bonne,
- Le bridge est bien créé et l'interface eno1 est linkée,
- L'accès réseau est opérationnel.

3. Création d'un conteneur template

3.0. Configuration par défaut pour les conteneurs

Pour modifier la configuration par défaut des conteneurs, on modifie le fichier /etc/lxc/default.conf
Le but ici est de lier automatiquement les interfaces réseau virtuelles des conteneurs avec le bridge br0.

```
root@srv-g1:~# nano /etc/lxc/default.conf
```

default.conf

```
# Type d'interface réseau dans le conteneur : ici virtual ethernet
lxc.net.0.type = veth
# Lien de l'interface du conteneur avec le bridge br0
lxc.net.0.link = br0
# On monte l'interface par défaut
lxc.net.0.flags = up
# Nom de l'interface réseau dans le conteneur : eth0 #whynot
lxc.net.0.name = eth0
# Later...
lxc.apparmor.profile = generated
lxc.apparmor.allow_nesting = 1
# Démarrage automatique du conteneur au démarrage du serveur
lxc.start.auto = 1
```

Ainsi à la création des nouveaux conteneurs, ils prondront automatiquement cette configuration par défaut.
La configuration particulière d'un conteneur se trouve dans le fichier **/var/lib/lxc/XXXXX/config**
(XXXXX étant le nom du conteneur)

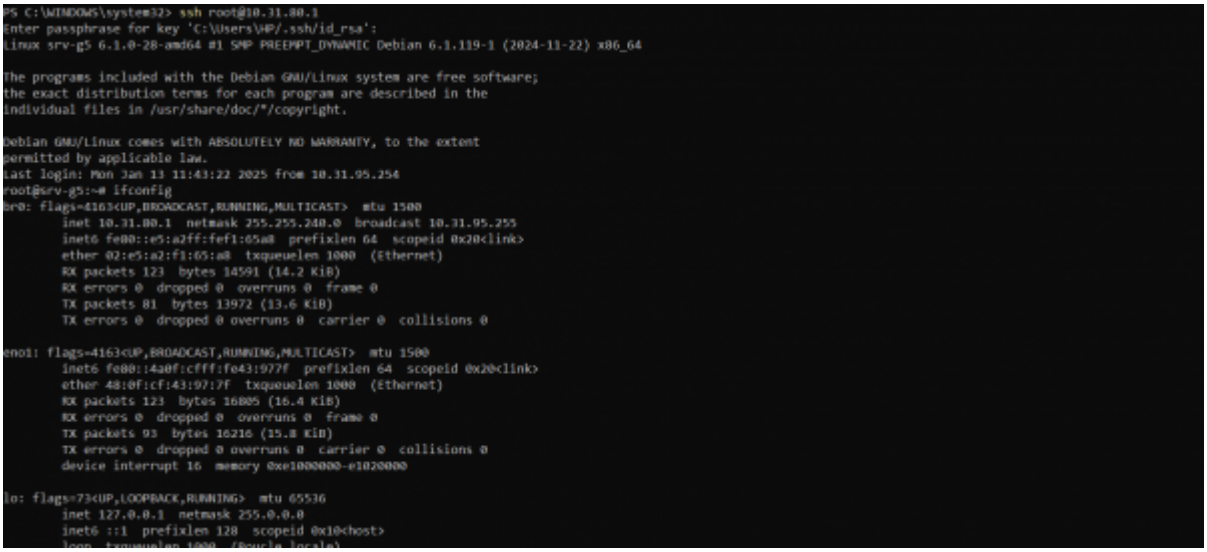
3.1. Création du conteneur template

Pour créer un conteneur template basé sur Debian 12 #Bookworm, j'utilise la commande suivante :

> **D'autres images de conteneurs sont disponibles**
> Liste des images disponibles : https://images.linuxcontainers.org/

```
root@srv-g1:~# lxc-create -n template -t debian -- -r bookworm
```

> **Résultat attendu**



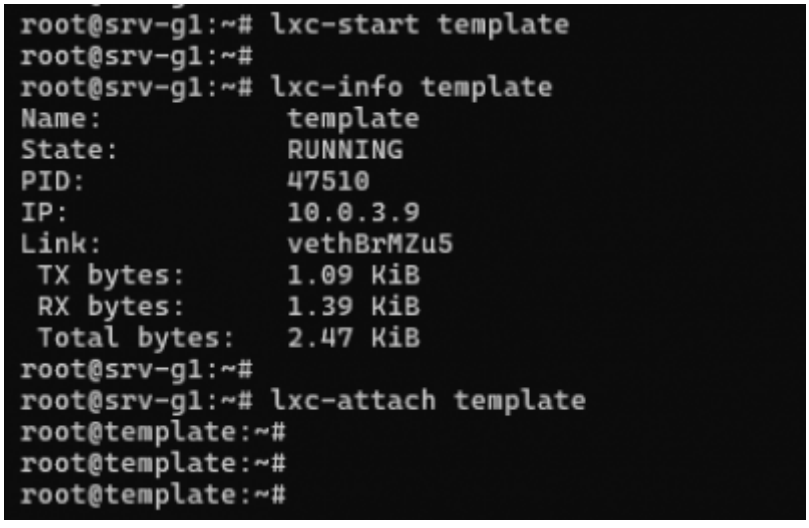
Je peux ensuite démarrer le conteneur avec :

```
root@srv-g1:~# lxc-start -n template
```

Pour accéder au conteneur, j'utilise :

```
root@srv-g1:~# lxc-attach -n template
```

Ce qui donne...



Attention !

On remarque bien qu'une fois que l'on a exécuté la commande lxc-attach, le prompt a changé, le nom de la machine a changé. On est désormais dans le conteneur et plus sur le serveur.

3.2. Configuration réseau du conteneur

Contrairement à la méthode de configuration IP du serveur, je configure l'interface réseau du conteneur `template` en modifiant le fichier `/etc/network/interfaces` :

```
root@template:~# nano /etc/network/interfaces
```

J'ajoute les lignes suivantes :

bash

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.31.80.80/20
    gateway 10.31.95.254
    dns-nameservers 8.8.8.8
```

Redémarrage du service réseau

Pour appliquer la configuration réseau il faut redémarrer le service réseau.

```
root@template:~# systemctl restart networking
```

Ensuite, je vérifie la configuration réseau avec :

```
root@template:~# ifconfig
```

```
root@template:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.31.16.2  netmask 255.255.240.0  broadcast 10.31.31.255
        inet6 fe80::216:3eff:fedf:8f45  prefixlen 64  scopeid 0x20<link>
        ether 00:16:3e:df:8f:45  txqueuelen 1000  (Ethernet)
        RX packets 13021  bytes 10284359 (9.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3197  bytes 216247 (211.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@template:~# ping -c2 google.fr
PING google.fr (142.251.37.163) 56(84) bytes of data.
64 bytes from mrs09s14-in-f3.1e100.net (142.251.37.163): icmp_seq=1 ttl=114 time=13.2 ms
64 bytes from mrs09s14-in-f3.1e100.net (142.251.37.163): icmp_seq=2 ttl=114 time=12.8 ms

--- google.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 12.846/13.011/13.176/0.165 ms
```

La configuration IP est conforme aux attendus et la connectivité réseau du conteneur est fonctionnelle

3.3. Installation des Outils de Base dans le conteneur

Ensuite, j'installe les outils de base à l'intérieur du conteneur :

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

Les outils installés sont les suivants :

- **sudo** : Outil permettant d'exécuter des commandes avec des privilèges administratifs.
- **net-tools** : Ensemble d'outils en ligne de commande pour la gestion des réseaux, comme ifconfig, netstat, etc.
- **tcpdump** : Outil permettant de capturer et analyser les paquets réseau en temps réel.
- **nano** : Éditeur de texte simple et convivial en ligne de commande.
- **iputils-ping** : Outil de diagnostic réseau pour tester la connectivité avec un hôte distant via des paquets ICMP.
- **dbus** : Système de bus de messages pour la communication inter-processus dans les systèmes Unix/Linux.

4. Tests de déploiement d'un conteneur web

4.0. Copie et Configuration du Conteneur

Je copie le conteneur `template` pour créer un nouveau conteneur `web` :

```
lxc-stop -n template
lxc-copy -n template -N web
lxc-start -n web
lxc-attach -n web
```

Je change le nom d'hôte du conteneur `web` :

```
hostnamectl set-hostname web
nano /etc/hosts
```

Je remplace le nom `template` ou `debian` par `web`.

4.1. Configuration Réseau du Conteneur `web`

Je modifie le fichier `/etc/network/interfaces` du conteneur `web` :

```
nano /etc/network/interfaces
```

J'ajoute les lignes suivantes :

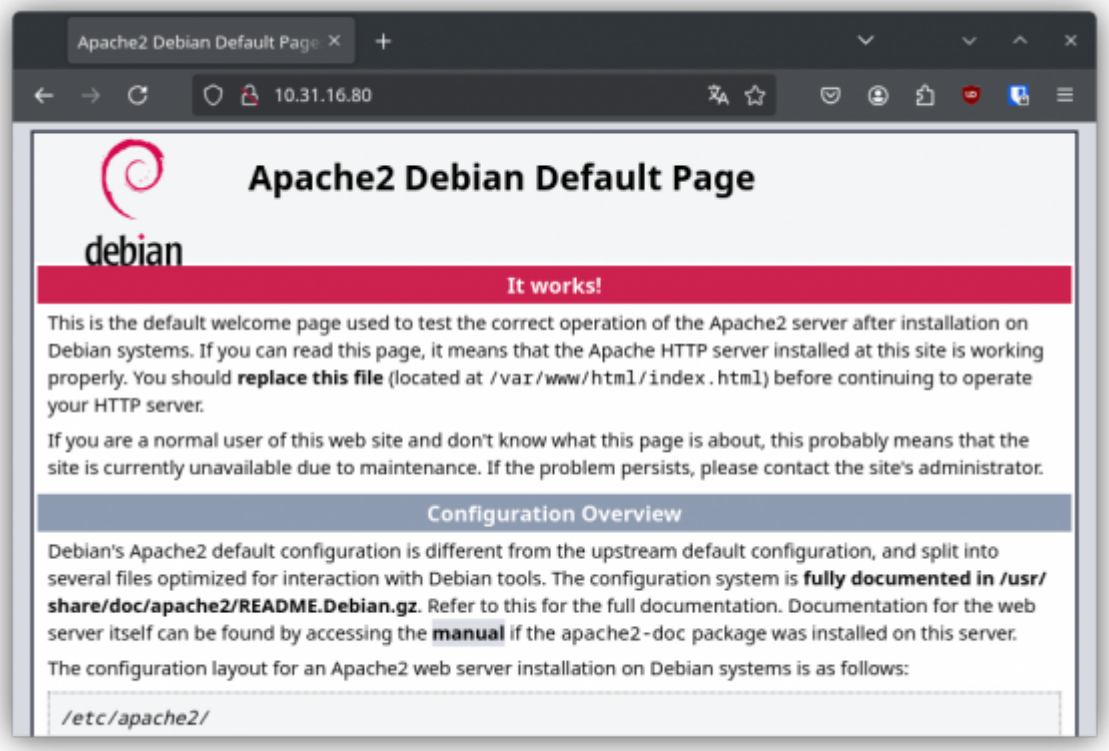
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.31.80.80/20
    gateway 10.31.95.254
    dns-nameservers 8.8.8.8
```

4.2. Installation d'Apache et Test

J'installe Apache dans le conteneur `web` et je crée une page web :

```
root@web:apt install apache2
echo "<html><body><h1>Page Web sur le Conteneur Web</h1></body></html>" >
/var/www/html/index.html
```

Je teste l'accès à cette page depuis un navigateur web en tapant **http://10.31.16.80**.



Le site apache2 par défaut est accessible à distance depuis le réseau de Beaupeyrat .
Tout fonctionne comme attendu.