# Faire et gérer un Linux Container (LXC) (Ubuntu 17.04)

Les Linux Containers permettent de faire des machines virtuelles pour compartimenter vos machines (dans notre cas les serveurs Bor et Heimdall). Ici on va voir comment les mettre en place facilement.

Dans notre réseau, il existe deux types de LXC. Celles connectées sur le réseau ouvert 10.61.16.0/24 et celles connectées sur un réseau fermé dont le coeur est au sein de BOR 10.61.15.0/24. La plupart du temps, les LXC seront sur le réseau fermé (10.61.15.0/24). Pour y accéder de l'extérieur, on mettra des règles de routage sur BOR

## Configuration du réseau et installation des dépendances



Si la machine est déjà configurée pour recevoir des LXC, passez à l'étape suivante

On va commencer par configurer le réseau de la machine hôte pour pouvoir attribuer une ip à chaque machine virtuelle quand elle seront crées. Ainsi, on crée un bridge **br0** pour le réseau ouvert (16.0) et un bridge **br1** pour le réseau fermé (15.0) sur la machine hôte. La configuration présentée est celle de BOR qui gère le routage (ce qui veut dire que tous les services sur le réseau en 15.0 seront accessibles via son ip 10.61.16.10). Si vous souhaitez une configuration un peu différente, à vous d'adapter, il existe de très simple tutoriels sur internet.

En règle général, sauf exception (LXC Meili, LXC Calendrier), on mettra toujours les LXC sur le réseau en 15.0 pour plus de sécurité.

```
$> sudo nano /etc/network/interfaces
```

/etc/network/interfaces

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
# This is an autoconfigured IPv6 interface
#auto eno1
#iface eno1 inet dhcp

#reseau en .16 vers exterieur
auto br0
iface br0 inet static
    address 10.61.16.10
```

```
    netmask 255.255.255.0
    gateway 10.61.16.254
    bridge-ifaces eno1
    bridge-ports eno1
    up ifconfig eno1 up

iface eno1 inet manual

#reseau en .15 ferme et relie a aucune interface
auto br1
iface br1 inet static
    address 10.61.15.1
    netmask 255.255.255.0
    network 10.61.15.0
    broadcast 10.61.15.255
    bridge_ports none
```

Cette configuration précise que l'interface primaire à utiliser pour le bridge br0 est eno1 (à vous d'adapter en fonction de la machine).

On va maintenant installer les dépendances nécéssaires sur la machine qui va héberger les machines virtuelles.

```
$> sudo apt-get install lxc lxc-templates wget bridge-utils
```

On va configurer notre réseau manuellement, donc éditer le fichier suivant :

```
$> sudo nano /etc/default/lxc-net
```

Et passez **“USE_LXC_BRIDGE” à “false”**, comme ça on informe le LXC qu'il va utiliser par défaut le réseau fermé sur le bridge que nous avons configuré précédemment. (br1)

```
USE_LXC_BRIDGE="false"
```

Maintenant éditez le fichier suivant, pour changer le lien réseau de lxc avec le bridge **br1** que nous avons crée sur la machine hôte. Ce bridge va nous permettre d'attribuer une ip à chaque machine virtuelle sur le réseau en 15.0

```
$> sudo nano /etc/lxc/default.conf
```

/etc/lxc/default.conf

```
lxc.network.type = veth
lxc.network.link = br1  # Nom de notre bridge
lxc.network.flags = up
lxc.network.hwaddr = 00:16:3e:xx:xx:xx
```

Il est nécéssaire redémarrer la machine pour vérifier que tout fonctionne correctement avant

d'installation notre première machine virtuelle.

```
$> sudo reboot
```

# Création d'un service sur un Linux Container (LXC)



Ici est seulement donné un exemple de création de LXC portant le nom de **bor_apache**, c'est à vous d'adapter la configuration en fonction du service que vous voulez proposer.

## Création de la machine virtuelle

Si vous lisez ceci, je suppose que toutes les dépendances sont installées. On va créer dans cette partie une machine virtuelle à qui on donnera le nom de **bor_apache** (pour un mettre un serveur apache). A vous d'adapter le nom en fonction du service que vous voulez proposer



Vous devez absolument respecter la convention fixée par le GInfo nomdelamachine_service pour le nom de votre LXC

En tapant cette commande dans la console, vous pouvez voir tous les modèles de système d'exploitation possibles pour votre LXC :

```
$> sudo ls /usr/share/lxc/templates/
```

```
 lxc-alpine    lxc-archlinux  lxc-centos  lxc-debian    lxc-fedora  lxc-
openmandriva  lxc-oracle  lxc-slackware   lxc-sshd    lxc-ubuntu-cloud
lxc-altlinux  lxc-busybox    lxc-cirros  lxc-download  lxc-gentoo  lxc-
opensuse      lxc-plamo   lxc-sparclinux  lxc-ubuntu
```



Vous devez être connecté à internet pour exécuter cette commande : Se connecter sur le portail captif en ligne de commande

Nous allons créer notre machine virtuelle sous Ubuntu, donc on tape cette commande :

```
$> sudo lxc-create -t ubuntu -n bor_apache -- --packages "nano,lynx,openssh-
server,wget,net-tools"
```

Le paramètre -n précise le nom donné à la machine virtuelle et -t le template sur lequel on sa base pour la créer. Le –packages installe une suite de packets nécéssaires pour la suite. Ensuite, vous verrez un long retour de commande, le temps que la machine se crée, patientez..

## Démarrage de la machine virtuelle

Pour démarrer la machine que vous venez de créer tapez :

```
$> sudo lxc-start -n bor_apache -d
```

Vous pouvez maintenant accéder directement en root à la machine via la commande :

```
$> sudo lxc-attach -n bor_apache
```

Vous pouvez maintenant configurer cette machine avec le routeur pour lui attribuer une ip en fonction de son adresse mac, vous pourrez ensuite accéder à cette machine via ssh comme pour toutes les autres, qu'elle soient physiques ou virtuelles.



Ne ratez pas l'étape suivante : on va supprimer l'utilisateur par défaut et ajouter un utilisateur **ginfo** courant qui aura les droits de sudo. On supprime ensuite l'utilisateur par défaut “ubuntu”

```
$> sudo adduser ginfo
$> sudo usermod -g sudo ginfo
$> sudo deluser ubuntu
$> sudo rm -r /home/ubuntu
```




Pour quitter la machine virtuelle, tapez exit

## Configuration du réseau de la machine virtuelle

On va commencer par attribuer à notre LXC une ip sur le réseau fermé (la procédure est quasi identique pour rajouter un service sur le réseau ouvert, il faut juste remplacer les 15 par les 16 et surtout faire utiliser aux LXC br0 et non br1). Pour cela, on édite le fichier `/etc/network/interfaces` **en étant connecté sur le LXC!** et on attribue l'ip que l'on souhaite (ici 10.61.15.26)

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.61.15.26
    netmask 255.255.255.0
    dns-nameservers 10.61.15.1
    gateway 10.61.15.1
```

**Etape facultative**

On peut suivre le tutoriel suivant pour rendre notre LXC accessible depuis les PC de l'école NAT : Accéder à un service LXC du réseau fermé depuis l'extérieur (réseau de l'école). Dans le cas de notre LXC apache, cela est évidemment nécessaire, on redirigera les ports externes de BOR 80/443 vers 10.61.15.26.

### Régler l'heure sur la LXC

Par défaut la LXC n'est pas à l'heure... Puis qu'elle n'est pas sur la bonne timezone. (Allez savoir pourquoi)

On commence par entrer dans la LXC et régler l'heure actuelle.

```
$> rm /etc/localtime
$> ln -s /usr/share/zoneinfo/Europe/Paris /etc/localtime
```

On rend ensuite cela définitif en édidant le fichier suivant */etc/timezone* (`nano /etc/timezone`) pour y mettre juste :

/etc/timezone

```
Europe/Paris
```

### Démarrer la LXC automatiquement au démarrage et modifier ses paramètres

Il faut modifier la configuration **/var/lib/lxc/bor_apache/config** concernant la machine virtuelle :

```
$> sudo nano /var/lib/lxc/bor_apache/config
```

Pour démarrer la machine au boot on ajoute la ligne suivante :

```
$> lxc.start.auto = 1
```

Dans ce fichier de configuration on peut en profiter par exemple pour modifier l'adresse mac du LXC.

## Commandes utiles et gestion des linux containers (LXC)

### Voir les informations d'un LXC

Pour voir la liste des LXC sur le système :

```
$> sudo lxc-ls -f
```

Pour voir la liste des LXC allumées :

```
$> sudo lxc-ls --active -f
```

## Eteindre un LXC

Pour éteindre le container **bor_apache** par exemple, tapez :

```
$> sudo lxc-stop -n bor_apache
```

## Détruire un LXC

Pour détruire le container **bor_apache** par exemple, tapez :

```
$> sudo lxc-destroy -n bor_apache
```

## Renommer un LXC

Pour renommer le container **pikachu** en **bor_pikachu** par exemple, on commence par arrêter le container :

```
$> sudo lxc-stop -n pikachu
```

Ensuite on renomme le dossier contenant le container :

```
$> sudo mv /var/lib/lxc/pikachu /var/lib/lxc/bor_pikachu
```

On modifie ensuite le fichier de configuration du container :

```
$> sudo nano /var/lib/lxc/bor_pikachu/config
```

Il faut modifier les 3 lignes suivantes en modifiant ici **pikachu** en **bor_pikachu**

```
lxc.utsname = pikachu
lxc.rootfs = /var/lib/lxc/pikachu/rootfs
lxc.mount = /var/lib/lxc/pikachu/fstab
```



La deuxième ligne peut contenir *dir*, dans ce cas là ne pas modifier.

On finit par redémarrer le container :

```
$> sudo lxc-start -n bor_pikachu
```

Ensuite il faut modifier le hostname de la lxc, pour cela **on s'y connecte** et on modifie le fichier

*/etc/hostname*

```
$> sudo nano /etc/hostname
```

Il suffit d'y mettre le nouveau nom, ici **bor_pikachu** par exemple.

From:
https://wiki.centrale-med.fr/ginfo/ - **Wiki GInfo**

Permanent link:
**https://wiki.centrale-med.fr/ginfo/musee:tutoriels:faire_un_lxc**

Last update: **16/10/2018 01:48**