

Création d'un Cluster LXD sur Raspberry

Table of Contents

- [Création d'un Cluster LXD sur Raspberry](#)
 - [Matériel](#)
 - [Installation](#)
 - [Configuration de l'appliance LXD](#)
 - [Configuration du noeud maître](#)
 - [Configuration des autres noeuds](#)
 - [Opération](#)
 - [Conclusion](#)

Il est possible sur des cartes ARM très bon marché comme le Raspberry Pi de mettre en place un cluster domestique très compact, silencieux, mais étonnamment puissant, capable d'exécuter des conteneurs ou des machines virtuelles, accessible depuis n'importe quel système de votre réseau.

Ces cartes sont devenues de plus en plus puissantes au fil des ans et sont désormais même capables de faire fonctionner des machines virtuelles complètes. Combiné avec la capacité de LXD à regrouper des systèmes, il est maintenant plus facile que jamais de configurer un cluster qui peut être facilement développé à l'avenir.

Matériel

Un cluster LXD fonctionnera bien sur les modèles 2 Go ou 4 Go, mais si on veut exécuter des machines virtuelles, il faut s'en tenir aux modèles 4 Go ou 8 Go.

Les cartes sont connectées au même réseau et on a besoin d'un écran compatible HDMI et d'un clavier USB pour la configuration initiale. Une fois cela fait, tout peut être fait à distance via SSH et l'API LXD.

Il est certainement possible de faire fonctionner cela sur une seule carte ou sur bien plus de 3 cartes, mais 3 est le nombre minimum pour que la base de données LXD soit hautement disponible.



Dans la configuration décrite, on utilise simplement un fichier loop sur la carte microSD. C'est un stockage plutôt lent et petit. Pour utiliser un stockage externe il faut spécifier "disque ou partition vide existant" lors de la configuration du cluster LXD ci-dessous.

Installation

Tout d'abord, télécharger l'image de l'appliance souhaitée et utiliser l'une des méthodes décrites [ici](#) pour graver l'image sur la carte SD.



Ubuntu a publié une appliance LXD dédiée

<https://cdimage.ubuntu.com/ubuntu-core/appliances/lxd-core18-pi.img.xz> ciblant à la fois le Raspberry Pi 4 et les systèmes Intel traditionnels.



Les articles suivants présentent la création détaillée de clusters (sur debian) qui peuvent être utilisés comme base:

- [Cluster Kubernetes sur Raspberry](#)
- [Création d'un cluster Raspberry PI avec ClusterHat.](#)
- [Création d'un cluster HPC sur Raspberry](#) :DONE:

Retirer la carte microSD de l'ordinateur et l'insérer dans le Raspberry Pi,

Allumer le Pi en le connectant à l'alimentation USB et se connecter en SSH, le Pi affichera une commande ssh semblable à ceci:

```
ssh <nom d'utilisateur Ubuntu SS0>@<adresse IP de l'appareil>
```

Avec Raspian Buster pour le Raspberry Pi, il est désormais possible d'installer LXD avec snapd:

```
sudo apt-get install snapd bridge-utils  
sudo snap install core lxd
```



[Deskpi Pro](#) est un kit matériel pour convertir une Raspberry PI 4 standard à partir d'un SBC nu, avec un stockage limité, en un mini PC complet avec un bouton d'alimentation, un refroidissement, de meilleurs ports et un Mass Storage via SATA USB3 ou M.2. Pour l'installer sur Raspbian 64bit il faut installer git en premier `apt-get update && apt-get -y install git` puis procéder ainsi:

```
cd ~  
git clone https://github.com/DeskPi-Team/deskpi.git  
cd ~/deskpi/  
./install.sh
```

Configuration de l'appliance LXD



Lorsqu'une configuration a déjà été automatiquement appliquée (ce qui est le cas de l'appliance Ubuntu), il faut annuler une partie celle-ci, en exécutant sur chacune des cartes :



```
sudo lxc profile device remove default root
sudo lxc profile device remove default eth0
sudo lxc storage delete local
sudo lxc config unset core.https_address
```

Configuration du noeud maître

Se connecter en ssh et exécuter :

```
sudo lxc config set core.trust_password <SOME PASSWORD>
```

Pour que les appareils puissent s'authentifier automatiquement il faut générer clés Secure Shell (SSH) :

```
ssh-keygen -t rsa
```

- Cela démarre le processus de génération de clé, l'utilitaire ssh-keygen invite à indiquer où stocker la clé
- Appuyer sur la touche ENTER pour accepter l'emplacement par défaut. L'utilitaire ssh-keygen demande une phrase de passe
- Taper une phrase de passe

On dispose maintenant d'une clé publique et privée que l'on peut utiliser pour s'authentifier, récupérer la clé générée en exécutant la commande suivante:

```
cat ~/.ssh/id_rsa.pub
```

Sur le premier noeud (le noeud maître), exécuter `sudo lxd init` et suivre les étapes comme indiqué ci-dessous :

```
Would you like to use LXD clustering? (yes/no) [default=no]: yes
What name should be used to identify this node in the cluster?
[default=localhost]: rpi01
What IP address or DNS name should be used to reach this node?
[default=10.166.11.235]:
Are you joining an existing cluster? (yes/no) [default=no]:
Setup password authentication on the cluster? (yes/no) [default=yes]:
Trust password for new clients:
Again:
Do you want to configure a new local storage pool? (yes/no) [default=yes]:
Name of the storage backend to use (btrfs, dir, lvm) [default=btrfs]:
```

```
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty disk or partition? (yes/no)
[default=no]:
Size in GB of the new loop device (1GB minimum) [default=5GB]: 20GB
Do you want to configure a new remote storage pool? (yes/no) [default=no]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to configure LXD to use an existing bridge or host interface?
(yes/no) [default=no]: yes
Name of the existing bridge or host interface: eth0
Would you like stale cached images to be updated automatically? (yes/no)
[default=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no)
[default=no]:
```

Pour que les appareils puissent s'authentifier automatiquement il faut générer clés Secure Shell (SSH):

```
ssh-keygen -t rsa
```

- Cela démarre le processus de génération de clé, l'utilitaire ssh-keygen invite à indiquer où stocker la clé
- Appuyer sur la touche ENTER pour accepter l'emplacement par défaut. L'utilitaire ssh-keygen demande une phrase de passe
- Taper une phrase de passe

On dispose maintenant d'une clé publique et privée que l'on peut utiliser pour s'authentifier, récupérer la clé générée en exécutant la commande suivante:

```
cat ~/.ssh/id_rsa.pub
```

Voilà, on a un cluster (d'un système) avec la mise en réseau et le stockage configurés.

Configuration des autres noeuds

Maintenant, il faut joindre les autres cartes en exécutant également `sudo lxd init` dessus :

```
stgraber@localhost:~$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]: yes
What name should be used to identify this node in the cluster?
[default=localhost]: rpi02
What IP address or DNS name should be used to reach this node?
[default=10.166.11.92]:
Are you joining an existing cluster? (yes/no) [default=no]: yes
IP address or FQDN of an existing cluster node: 10.166.11.235
Cluster fingerprint:
b9d2523a4935474c4a52f16ceb8a44e80907143e219a3248fbb9f5ac5d53d926
You can validate this fingerprint by running "lxc info" locally on an
```

```

existing node.
Is this the correct fingerprint? (yes/no) [default=no]: yes
Cluster trust password:
All existing data is lost when joining a cluster, continue? (yes/no)
[default=no] yes
Choose "source" property for storage pool "local":
Choose "size" property for storage pool "local": 20GB
Would you like a YAML "lxd init" preseed to be printed? (yes/no)
[default=no]:
stgraber@localhost:~$

```

Il faut mettre la clé ssh copiée précédemment dans le fichier ~/.ssh/authorized_keys:

```
echo [paste the ssh key here] >> ~/.ssh/authorized_keys
```

Et valider que tout semble bon en exécutant `sudo lxc cluster list` sur l'un d'entre eux :

```

stgraber@localhost:~$ sudo lxc cluster list
+-----+-----+-----+-----+-----+
| NAME | URL | DATABASE | STATE | MESSAGE |
| ARCHITECTURE |
+-----+-----+-----+-----+
| rpi01 | https://10.166.11.235:8443 | YES | ONLINE | fully operational |
| aarch64 |
+-----+-----+-----+-----+
| rpi02 | https://10.166.11.92:8443 | YES | ONLINE | fully operational |
| aarch64 |
+-----+-----+-----+-----+
| rpi03 | https://10.166.11.200:8443 | YES | ONLINE | fully operational |
| aarch64 |
+-----+-----+-----+-----+
stgraber@localhost:~$

```

Il y a de fortes chances qu'on n'ait plus jamais besoin de SSH sur ces cartes, tout le reste peut maintenant être fait à distance via le client de ligne de commande LXD ou l'API à partir de n'importe quel système Linux, Windows ou macOS.

Opération

Maintenant, sur n'importe quel système sur lequel on souhaite utiliser ce cluster nouvellement configuré, exécuter (en mettant à jour l'adresse IP pour qu'elle corresponde à celle de l'une des cartes) :

```
stgraber@castiana:~$ lxc remote add my-cluster 10.166.11.235
Certificate fingerprint:
b9d2523a4935474c4a52f16ceb8a44e80907143e219a3248fbb9f5ac5d53d926
ok (y/n)? y
Admin password for my-cluster:
Client certificate stored at server: my-cluster
stgraber@castiana:~$ lxc remote switch my-cluster
stgraber@castiana:~$ lxc cluster list
```

NAME	URL	DATABASE	STATE	MESSAGE
ARCHITECTURE	FAILURE DOMAIN			
rpi01	https://10.166.11.235:8443	YES	ONLINE	fully operational
aarch64				
rpi02	https://10.166.11.92:8443	YES	ONLINE	fully operational
aarch64				
rpi03	https://10.166.11.200:8443	YES	ONLINE	fully operational
aarch64				

Chaque fois qu'on souhaite interagir avec un système local plutôt qu'avec le cluster distant, exécuter simplement :

```
lxc remote switch local
```



Étant donné qu'aucune image de machine virtuelle Arm 64 bits n'est actuellement configurée pour le démarrage sécurisé, on peut le désactiver complètement avec :

```
lxc profile set default security.secureboot false
```

Et maintenant, on peut exécuter quelques conteneurs et machines virtuelles :

```
lxc launch images:alpine/edge c1
lxc launch images:archlinux c2
lxc launch images:ubuntu/18.04 c3
lxc launch images:ubuntu/20.04/cloud v1 --vm
lxc launch images:fedora/32/cloud v2 --vm
lxc launch images:debian/11/cloud v3 --vm
```

Les opérations de lancement initiales seront plutôt lentes, surtout si on utilise la carte microSD principale pour le stockage. Cependant, la création de plusieurs de ces instances devrait alors être assez rapide grâce à l'image déjà prête.

```
stgraber@castiana:~$ lxc list
```

+-----+-----+-----+-----+-----+				
+-----+-----+-----+-----+-----+				
NAME	STATE	IPV4	IPV6	
TYPE		SNAPSHOTS	LOCATION	
+-----+-----+-----+-----+-----+				
c1	RUNNING	10.166.11.113 (eth0)		
fd42:4c81:5770:1eaf:216:3eff:fed7:35c0 (eth0)		CONTAINER		0
rpi01				
+-----+-----+-----+-----+-----+				
c2	RUNNING	10.166.11.88 (eth0)		
fd42:4c81:5770:1eaf:216:3eff:feaa:81ac (eth0)		CONTAINER		0
rpi02				
+-----+-----+-----+-----+-----+				
c3	RUNNING	10.166.11.146 (eth0)		
fd42:4c81:5770:1eaf:216:3eff:fe79:75a5 (eth0)		CONTAINER		0
rpi03				
+-----+-----+-----+-----+-----+				
v1	RUNNING	10.166.11.200 (enp5s0)	fd42:4c81:5770:1eaf:216:3eff:fe61:1ad6 (enp5s0)	
VIRTUAL-MACHINE		0	rpi01	
+-----+-----+-----+-----+-----+				
v2	RUNNING	10.166.11.238 (enp5s0)		
fd42:4c81:5770:1eaf:216:3eff:fe8d:e6ae (enp5s0)		VIRTUAL-MACHINE		0
rpi02				
+-----+-----+-----+-----+-----+				
v3	RUNNING	10.166.11.33 (enp5s0)		
fd42:4c81:5770:1eaf:216:3eff:fe86:526a (enp5s0)		VIRTUAL-MACHINE		0
rpi03				
+-----+-----+-----+-----+-----+				
+-----+-----+-----+-----+-----+				

À partir de là, tout devrait sembler assez normal. On peut utiliser `lxc exec` pour exécuter directement des commandes à l'intérieur de ces instances, `lxc console` pour accéder à la console texte ou VGA, ...

Conclusion

LXD constitue une solution très simple et flexible pour configurer un laboratoire, que ce soit à la maison avec des cartes très Raspberry Pi, dans le cloud en utilisant des instances de cloud public ou

sur tout matériel de rechange dont vous disposez.

L'ajout de serveurs supplémentaires à un cluster est simple et rapide et les clusters LXD prennent même en charge les architectures mixtes, ce qui vous permet de mélanger Raspberry Pi 4 et Intel NUC en un seul cluster capable d'exécuter à la fois les charges de travail Intel et Arm.

Tout se comporte à peu près de la même manière que sur votre ordinateur portable ou de bureau, mais il est désormais accessible à plusieurs utilisateurs à partir de n'importe quel système avec accès au réseau.

À partir de là, on peut passer à une configuration de cluster beaucoup plus grande, en utilisant des projets pour gérer plusieurs utilisations distinctes du cluster, en attachant un stockage distant, en utilisant un réseau virtuel, en intégrant avec MAAS ou avec Canonical RBAC, ... Il existe de nombreuses options qui peuvent être progressivement ajouté à une configuration comme celle-ci lorsqu'on en ressent le besoin.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-cluster-lxd>

Last update: **2025/02/19 10:59**

