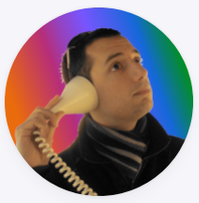


Zwindler's Reflection

Blog d'un startuper 👨‍💻 punk à chat 🐱

Rechercher

Cherchez un article, une publication, etc.

TABLE DES MATIÈRES

Système Virtualisation

Lancer des containers Docker avec Proxmox VE (et LXC)

Ecrit par ~ zwindler ~

07 Nov 2022 5 minutes de lecture

Introduction

J'ai trouvé plusieurs tutos pour installer Docker (engine) **à l'intérieur** d'un container LXC sous Proxmox VE pour ensuite lancer des containers Docker dedans. Le souci c'est que vous devez vous logger **dans** le container LXC avant de pouvoir interagir avec vos containers Docker.

On peut aussi installer Docker directement sur Proxmox VE mais ce n'est pas trop conseillé et on n'a pas la facilité d'usage de LXC avec la GUI de Proxmox.

Et en fait, il se trouve qu'on peut lancer des images Docker **comme l'OS** d'un container LXC et ainsi avoir des Docker pilotés par Proxmox. On aura donc pas un comportement 100% identique à si vous aviez fait un `docker run` sur votre machine.

Cependant, on gagnera un échange un container totalement identique à vos autres containers LXC de votre cluster (avec toutes les fonctions associées) mais lancé avec une image Docker.

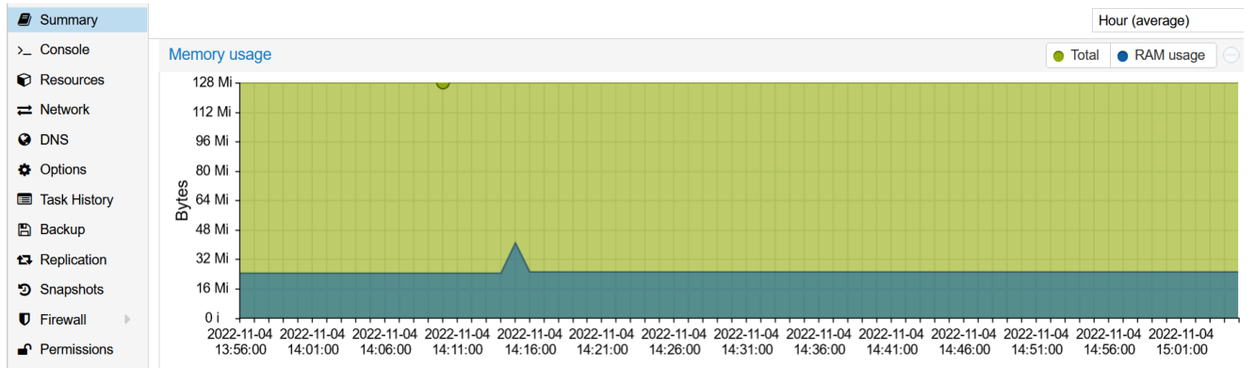
Mais d'abord je vous propose de revenir un peu en arrière avant de vous montrer comment...

Contexte

Vous le savez, je fais du Proxmox VE depuis un moment. Proxmox VE est une super distribution clé en main et production ready de virtualisation de serveurs (c'est même de l'hyperconvergé si on active la partie Ceph).

Il y a plein de petites choses que j'aime avec Proxmox VE, et l'une d'elle est qu'on peut créer des containers Linux avec LXC plutôt que des machines virtuelles QEMU. J'avais fait un article pour en parler.

A l'usage (sauf paramétrage kernel spécifique) le container LXC se comporte de manière strictement identique à une VM dans Proxmox, mais pour une fraction des ressources ! On a un OS, on installe des applications dessus, comme une VM.



> Un frontal nginx pour plusieurs applications. Le container LXC complet ne consomme que quelques Mo

Certes c'est beaucoup moins bien isolé qu'une vraie VM (car c'est un container, on est isolé des autres processus, mais on tourne sur le kernel de l'hyperviseur directement).

Mais avec si peu de besoins en ressources, je peux héberger de nombreuses petites applications dans des contextes différents avec de très petites machines physiques (des Atom 4Go de RAM à 6€ par mois chez OneProviders) ce qui serait impossible avec une vraie VM.

C'est bien, mais pas suffisant

Cependant, LXC n'est pas la techno de containerisation la plus connue/hype. Depuis des années sur le forum, à chaque fois que des gens demande le support de Docker dans Proxmox VE, on les envoie balader. Pas toujours très gentiment d'ailleurs...

Pourtant, il y a plusieurs raisons pour avoir envie de lancer des containers Docker sur son infra Proxmox :

- D'abord, Docker n'est pas autant des workloads stateless et/ou éphémères qu'on le croit. Beaucoup d'applications containerisées auraient toute leur place sur un cluster de virtualisation.
- Ensuite parce que beaucoup d'éditeurs de logiciels sont devenus faignants. Certains ne gèrent plus les processus d'installation (ou alors juste sur un OS) et fournissent juste une image Docker qu'ils maintiennent mise à jour.

On doit donc se contenter de VMs dans Proxmox pour lancer des containers Docker (cf mon point du début).

LXC supporte les images OCI

Coup de théâtre. Il y a quelques jours, j'ai découvert que LXC, le moteur de containerisation de Proxmox VE, était compatible avec le format OCI.

- buzzwrd.me - Creating LXC containers from docker and OCI images

Pour ceux qui ne connaissent pas OCI, il s'agit d'une organisation visant à créer un standard pour unifie la façon de stocker les containers.

> The Open Container Initiative is an open governance structure for the express purpose of creating open industry standards around container formats and runtimes. opencontainers.org/

Et Docker respecte ce format : on peut donc théoriquement lancer depuis LXC des containers dont l'image de base est une image Docker.

> Je suis profondément choqué (Réf "Jean François Coppé" pour ceux qui n'ont pas le mème)

Bon, quand est-ce qu'on commence ?

Disclaimer: ce qui suit est de la bidouille. Rien n'est supporté et je ne le conseille pas en production. Il est probable qu'il y ait aussi des limitations (notamment pour ce qui est stockage).

D'abord, il manque quelques dépendances pour que la fonction OCI de LXC soit utilisable sous Proxmox VE :

```
sudo apt install skopeo umoci jq
```

Ensuite, par défaut, les containers LXC s'attachent à un bridge linux qui s'appelle `lxcbr0`. Cependant ce bridge n'existe probablement pas sur votre installation de Proxmox VE (pas installé par défaut).

En *quick and dirty*, on peut juste remplacer le nom par défaut, même si idéalement il faudrait trouver le flag dans la CLI pour changer le bridge. En fonction de comment vous avez configuré le réseau dans votre serveur Proxmox VE, vous allez probablement utiliser `vmbr0` ou `vmbr1` :

```
sed -i 's/lxcbr0/vmbr1/g' /etc/lxc/default.conf
```

Maintenant qu'on a tout, on peut lancer nos containers LXC à partir d'images docker. Ici je lance l'image officielle `alpine:latest`, mais j'aurais pu prendre n'importe quelle autre :

```
lxc-create 500 -t oci -- --url docker://alpine:latest
```

J'ai créé un container appelé "500" (car les VMs et les containers LXC dans proxmox VE sont appelés avec des nombres). L'image est téléchargée puis le container créé :

```
Getting image source signatures
Copying blob 213ec9aee27d [--------------------------------------] 0.0b / 0.0b
Copying config 29f453b10b done
Writing manifest to image destination
Storing signatures
Unpacking the rootfs
```

On peut maintenant le démarrer (avec `lxc-execute` on lance le container et ouvrir un prompt directement dedans, mais on a aussi `lxc-start` et `lxc-attach`)

```
lxc-execute 500
```

```
/ # cat /etc/os-release
NAME="Alpine Linux"
ID=alpine
VERSION_ID=3.16.2
PRETTY_NAME="Alpine Linux v3.16"
HOME_URL="https://alpinelinux.org/"
BUG_REPORT_URL="https://gitlab.alpinelinux.org/alpine/aports/-/issues"
```

Tel quel, Proxmox VE ne sait pas que notre container existe et il n'apparaitra pas dans l'UI. Mais ça fonctionne ;-).

Bonus

On peut même tricher et faire croire à Proxmox que c'est un container LXC qu'il a créé lui-même en créant un fichier `500.conf`, dans le dossier `/etc/pve/lxc/`

Les seules lignes obligatoires sont les suivantes :

- arch (amd64 sauf si vous êtes sur ARM)
- cores (la limite de CPUs à imposer au container)
- hostname
- memory (la limite de RAM à imposer au container)
- ostype
- rootfs

```
cat /etc/pve/lxc/500.conf
arch: amd64
cores: 1
hostname: docker-alpine
memory: 128
ostype: alpine
rootfs: /var/lib/lxc/500/rootfs
```

Une fois le fichier créé, automatiquement le container apparait dans la liste des VMs de l'hôte. On peut le démarrer et ça fonctionne :)

```
Welcome to Alpine Linux 3.16
Kernel 5.15.35-1-pve on an x86_64 (/dev/tty1)

docker-alpine login:
```

Le container pourra être démarré, arrêté, etc. Le tout depuis la GUI de Proxmox VE :)

Have fun!

Source

- Proxmox VE - Linux Container
- Setup and Install Docker in a Promox 7 LXC Conainer
- buzzwrd.me - Creating LXC containers from docker and OCI images

LXC Docker Proxmox VE QEMU

CONTENUS LIÉS

Je déconseille ngin
tant

Généré avec Hugo
Thème **Stack** conçu par Jimmy