

Création de conteneur LXC

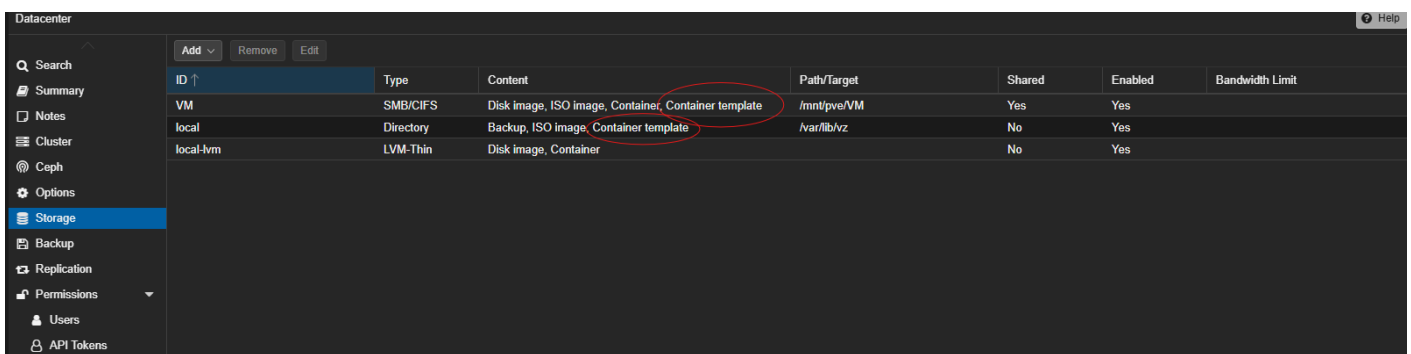
Les conteneurs LXC (Linux Containers) est une solution de virtualisation légère permettant d'exécuter plusieurs systèmes Linux isolés sur un même hôte. Contrairement aux machines virtuelles classiques, ils partagent le noyau de l'hôte, ce qui les rend plus rapides et moins gourmands en ressources. Cela se base également sur des template déjà disponible dans proxmox, à téléchargé pour une mise en place rapide.

- [Récupération d'une template](#)
- [Mise en place d'un conteneur](#)
- [Bonus Mise en place de la Haute Disponibilité](#)

Récupération d'une template

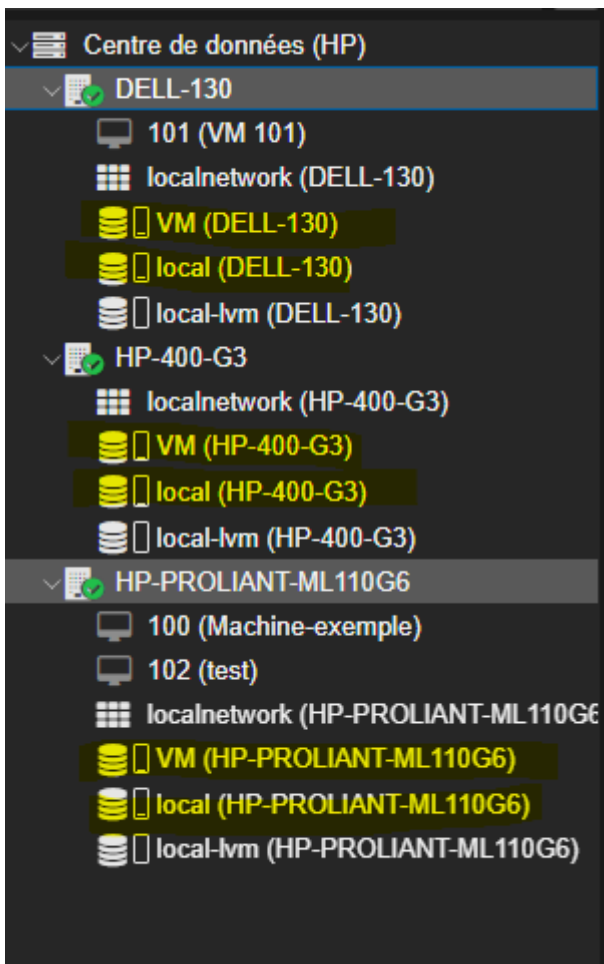
Une template, c'est une images préconfigurées d'OS Linux. Ils accélèrent la création de conteneurs en fournissant un système de base prêt à l'emploi. Proxmox offre des templates officiels et permet la création de modèles personnalisés que nous allons découvrir ici.

Pour pouvoir téléchargé des template de conteneur, il faut vérifier qu'il y a les autorisations sur les répertoire. Pour cela, on va ce rendre dans "**Datacenter>Storage**" et regarder si on voit "**container template**" / "**Modèle de conteneur**" :



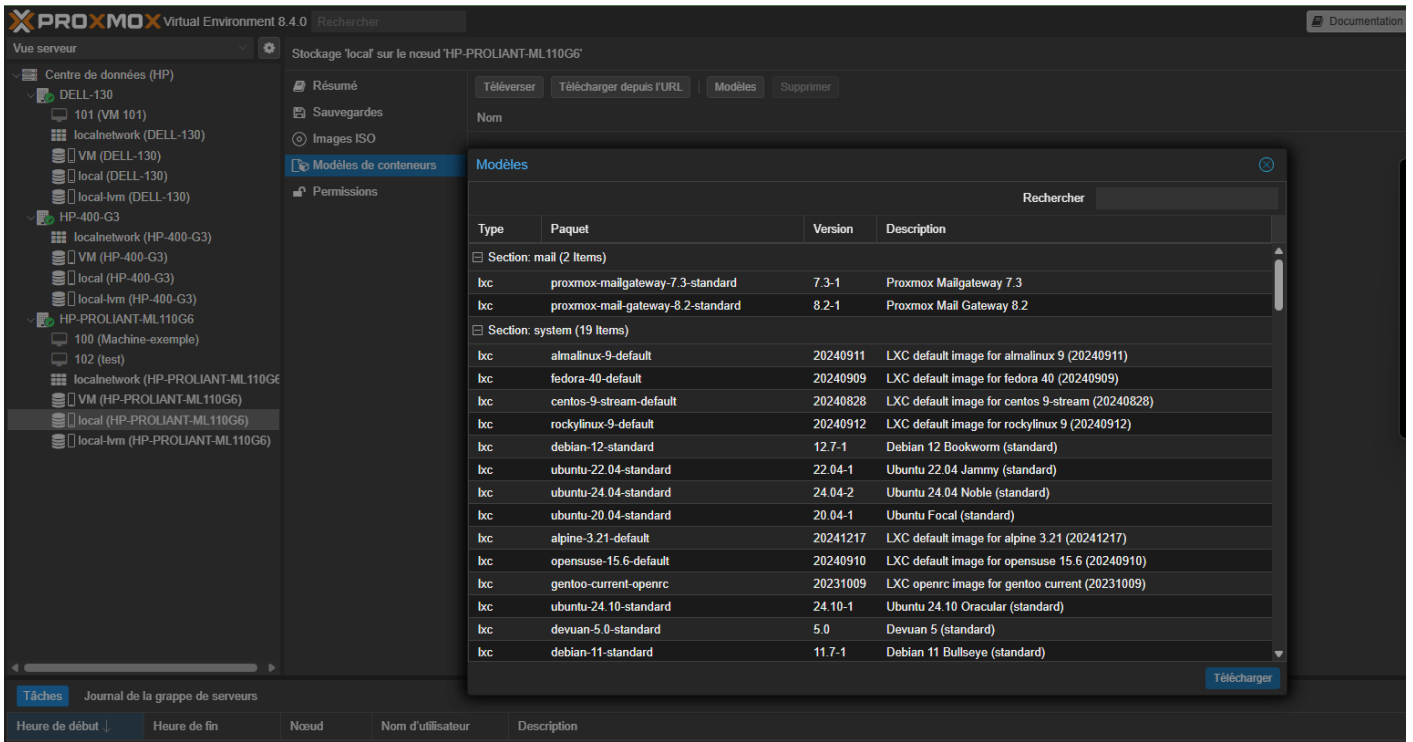
ID ↑	Type	Content	Path/Target	Shared	Enabled	Bandwidth Limit
VM	SMB/CIFS	Disk image, ISO image, Container, Container template	/mnt/pve/VM	Yes	Yes	
local	Directory	Backup, ISO image, Container template	/var/lib/yz	No	Yes	
local-lvm	LVM-Thin	Disk image, Container		No	Yes	

Dans mon cas "VM" et "local" peuvent avoir des templates et c'est pour cela que je vais aller sur VM ou local pour téléchargé une template :

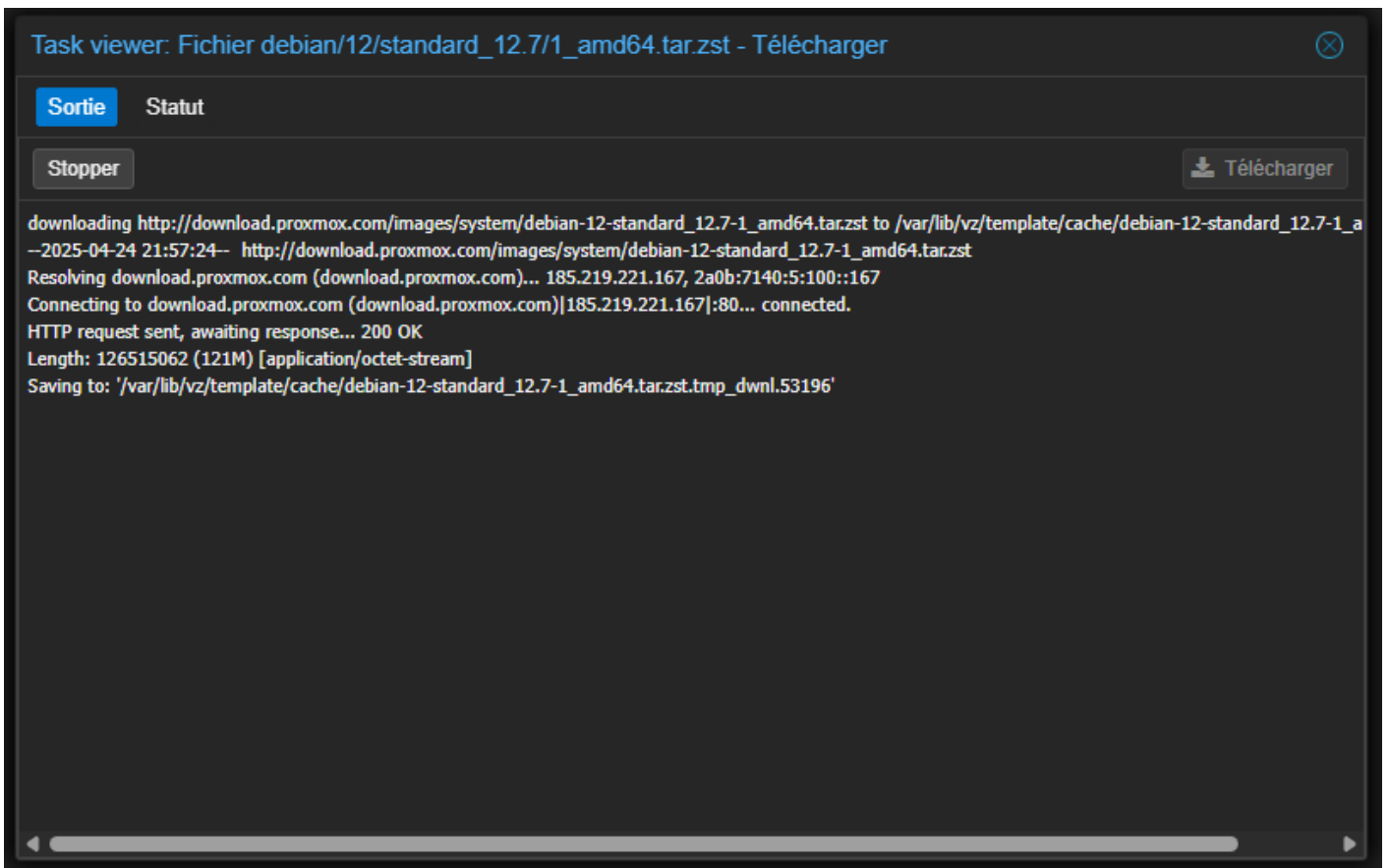


Ici, j'ai 3 proxmox relier mais cela n'est pas obligatoire pour créé un conteneur. C'est pour cela que je vais tous faire sur HP-PROLIANT-ML110G6.

Pour mon cas, je vais dans local puis je clique sur "**Modèles**" / "**template**" et on peut voir tous les modèles disponibles :



On a un vaste choix de template, que se soit ubuntu, debian, opensuse, etc et aussi de l'autre service de proxmox pour les mails. Pour mon cas, je vais prendre debian 12 pour l'exemple :



J'ai donc sélectionné debien 12 puis fais "**download**" / "**téléchargement**" puis il proxmox lance le téléchargement.

Voilà la template téléchargé et présente dans "Local" :

Task viewer: Fichier debian/12/standard_12.7/1_amd64.tar.zst - Télécharger

Sortie Statut

Stopper Télécharger

```
downloading http://download.proxmox.com/images/system/debian-12-standard_12.7-1_amd64.tar.zst to /var/lib/vz/template/cache/debian-12-standard_12.7-1
--2025-04-24 21:57:24-- http://download.proxmox.com/images/system/debian-12-standard_12.7-1_amd64.tar.zst
Resolving download.proxmox.com (download.proxmox.com)... 185.219.221.167, 2a0b:7140:5:100::167
Connecting to download.proxmox.com (download.proxmox.com)[185.219.221.167]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 126515062 (121M) [application/octet-stream]
Saving to: '/var/lib/vz/template/cache/debian-12-standard_12.7-1_amd64.tar.zst.tmp_dwnl.53196'
 0K ..... 26% 1.28M 70s
32768K ..... 53% 1.28M 44s
65536K ..... 79% 1.28M 19s
98304K ..... 100% 1.29M=94s
2025-04-24 21:58:58 (1.28 MB/s) - '/var/lib/vz/template/cache/debian-12-standard_12.7-1_amd64.tar.zst.tmp_dwnl.53196' saved [126515062/126515062]
calculating checksum...OK, checksum verified
download of 'http://download.proxmox.com/images/system/debian-12-standard_12.7-1_amd64.tar.zst' to '/var/lib/vz/template/cache/debian-12-standard_12.7-1'
TASK OK
```

'HP-PROLIANT-ML110G6'

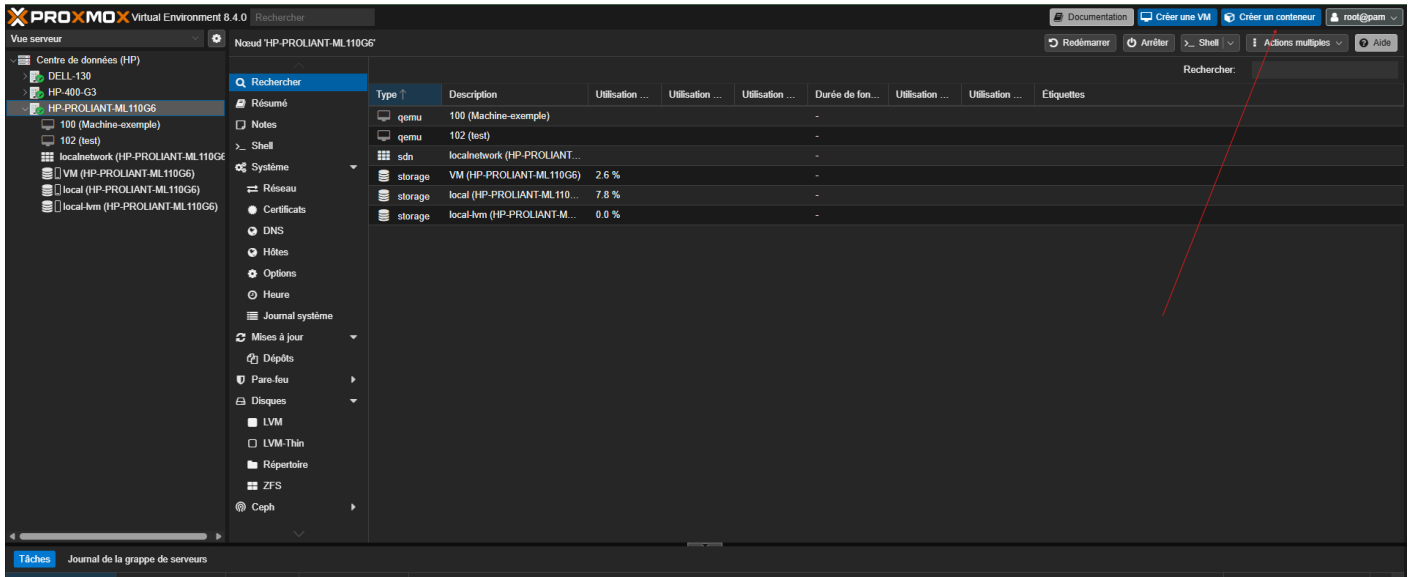
Téléverser | Télécharger depuis l'URL | Modèles | Supprimer

Nom

debian-12-standard_12.7-1_amd64.tar.zst

Mise en place d'un conteneur

Pour créer un conteneur, je vais donc faire "Créer un conteneur" :



Suite à cela, une fenêtre s'ouvre :

Créer: Conteneur LXC ✕

Général | Modèle | Disques | Processeur | Mémoire | Réseau | DNS | Confirmation

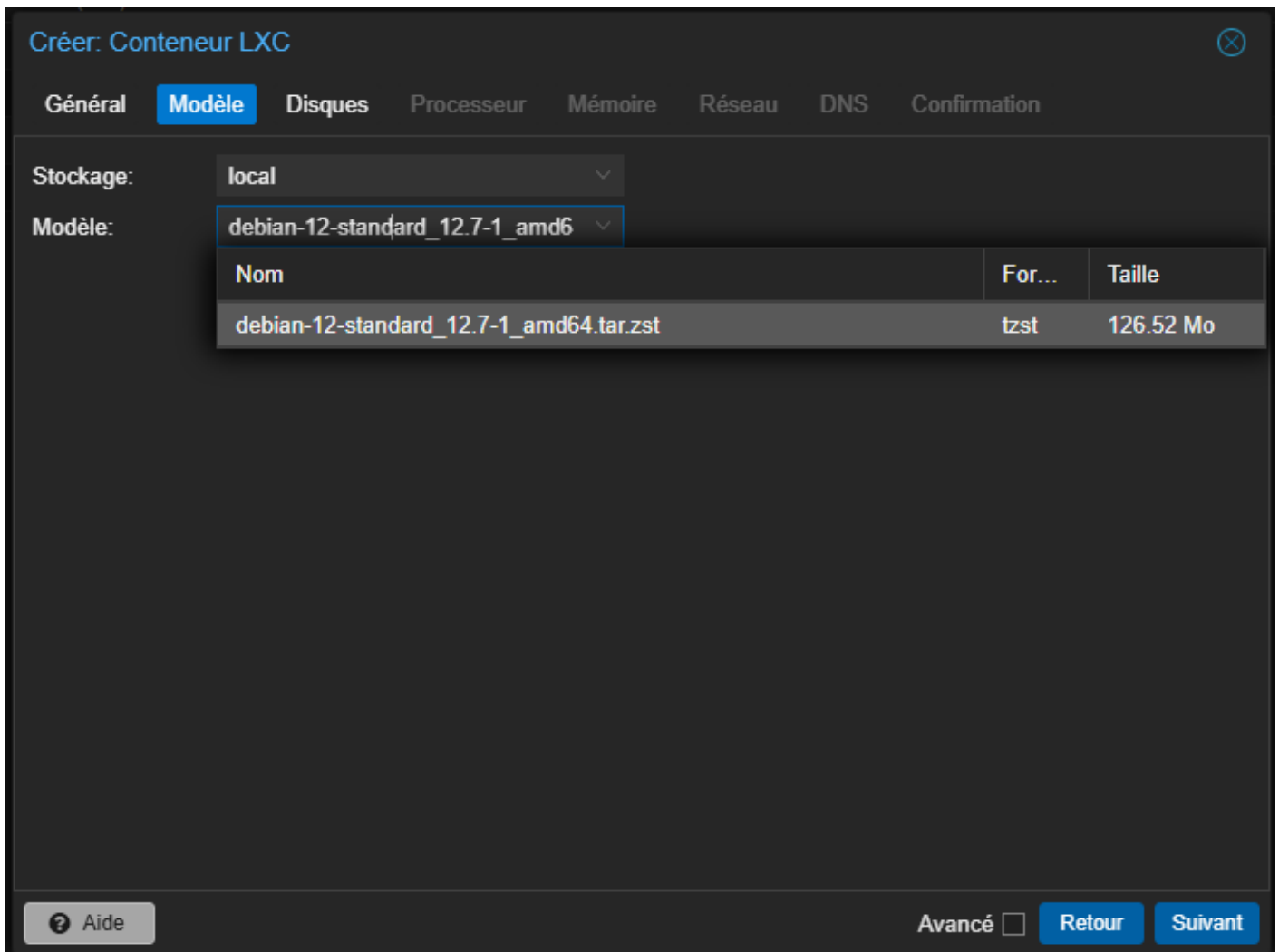
Nœud:	HP-PROLIANT-ML110G6	Pool de ressources:	
CT ID:	103	Mot de passe:	*****
Nom d'hôte:	neofetch	Confirmer le mot de passe:	*****
Conteneur non privilégié:	<input checked="" type="checkbox"/>	Clef(s) SSH publique(s):	
Imbriqué:	<input checked="" type="checkbox"/>		

[Charger le fichier de clef SSH](#)

[? Aide](#) Avancé [Retour](#) [Suivant](#)

On donne donc un nom au conteneur dans "Nom d'hôte", moi j'ai choisi de mettre neofetch et on choisit un mot de passe qui fera minimum 5 caractères.


Puis on passe à l'onglet "Modèle" où nous sélectionnons le stockage où se trouve le modèle téléchargé précédemment, puis on le choisit :




Pour mon exemple, je vais laissé 8go de stackage qui sera suffisant mais vous êtes libres de mettre la taille que vous voulez. Aussi "VM" est un stockage partagé pour de la [HA \(haute disponibilité\) configuré précement](#) mais vous pouvez aussi le mettre sur un autre stockage.


Créer: Conteneur LXC

Général Modèle **Disques** Processeur Mémoire Réseau DNS Confirmation

rootfs  Stockage: VM

Taille du disque (Gio): 8

 Ajouter

 Aide Avancé **Retour** **Suivant**

Processeur et mémoire pour mon exemple resterons aussi par défaut mais libre à vous de mettre plus haut que ce qui est de base. Cela dépend surtout de votre besoin :

Créer: Conteneur LXC

Général Modèle Disques **Processeur** Mémoire Réseau DNS Confirmation

Cœurs: 1

Créer: Conteneur LXC

Général Modèle Disques Processeur **Mémoire** Réseau DNS Confirmation

Mémoire (MiB): 512

Espace d'échange (swap) (MiB): 512

Pour la partie réseau, je vais mettre en static pour l'ipv4 et DHCP pour l'ipv6, donc définir moi même l'adresse ip en fixe pour la v4 mais si vous ne voulez pas vous embêter, libre à vous de mettre sur DHCP. J'ai donc mit une ip libre de mon réseau, ici 192.168.3.21 en 255.255.255.0 (d'où le /24) et la passerelle 192.168.3.1 (mon routeur) :

Créer: Conteneur LXC

Général Modèle Disques Processeur Mémoire Réseau DNS Confirmation

Nom: eth0 IPv4: Statique DHCP

Adresse MAC: auto IPv4/CIDR: 192.168.3.21/24

Pont (bridge): vubr0 Passerelle (IPv4): 192.168.3.1

Étiquette de VLAN: aucun VLAN IPv6: Statique DHCP SLAAC

Pare-feu: IPv6/CIDR: Passerelle (IPv6):

Aide Avancé Retour Suivant

La partie DNS, je n'y touche pas, il utilisera le DNS de l'hôte et on voit un résumé de la configuration dans "confirmation" puis on finit par faire "**Terminer**" :

Créer: Conteneur LXC ✕

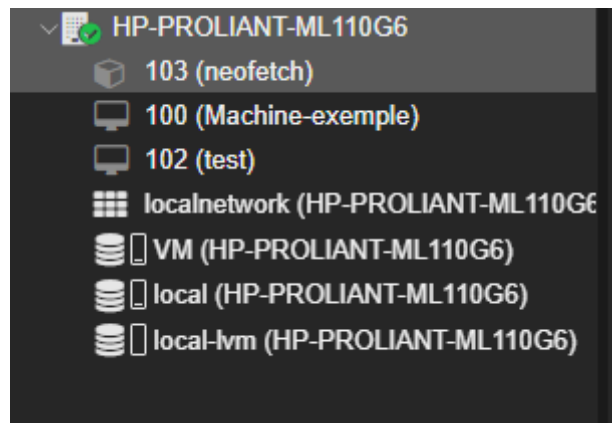
Général Modèle Disques Processeur Mémoire Réseau DNS **Confirmation**

Key ↑	Value
cores	1
features	nesting=1
hostname	neofetch
memory	512
net0	name=eth0,bridge=vmbbr0,firewall=1,ip=192.168.3.21/24,gw=192.168.3.1,ip6=dhcp
nodename	HP-PROLIANT-ML110G6
ostemplate	local:vztmpl/debian-12-standard_12.7-1_amd64.tar.zst
pool	
rootfs	VM:8
ssh-public-keys	
swap	512
unprivileged	1
vmid	103

Démarrer après création

Avancé **Retour** **Terminer**

Une fois finie d'être créée, elle apparaît avec les VM :



Je vais donc démarrer ce conteneur puis montrer son fonctionnement.

De base, il n'y a que root et root n'est pas autorisé pour le SSH, je vous invite donc à créer un utilisateur pour SSH. Je me suis donc connecté depuis la console de Proxmox :

```
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify (mogrify) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-im6 (mogrify-im6) in auto mode
Setting up libgs10-common (10.0.0~dfsg-11+deb12u7) ...
Setting up chafa (1.12.4-1+b1) ...
Setting up imagemagick (8:6.9.11.60+dfsg-1.6+deb12u2) ...
Setting up libgs10:amd64 (10.0.0~dfsg-11+deb12u7) ...
Setting up libspectre1:amd64 (0.2.12-1) ...
Setting up ghostscript (10.0.0~dfsg-11+deb12u7) ...
Setting up libimlib2:amd64 (1.10.0-4+deb12u1) ...
Setting up w3m-img (0.5.3+git20230121-2) ...
Setting up caca-utils (0.99.beta20-3) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for mailcap (3.70+nmul) ...
Processing triggers for libc-bin (2.36-9+deb12u8) ...

Progress: [ 99%] [#####]
#####
#####
Processing triggers for libgdk-pixbuf-2.0-0:amd64 (2.42.10+dfsg-1+deb12u1) ...
root@neofetch:~# neofetch
-bash: neofetch: command not found
root@neofetch:~# neofetch

  ,met$$$$$gg.          root@neofetch
 ,g$$$$$$$$$$$$$$P.    -----
 ,g$$P"      ""Y$$$.    OS: Debian GNU/Linux 12 (bookworm) x86_64
 ,$$P'      `$$$$.      Host: ProLiant ML110 G6
',$$P      ,ggs.       `$$b:  Kernel: 6.8.12-9-pve
`d$$'      , $P"      .  $$$  Uptime: 29 mins
 $$P       d$'      ,  $$P  Packages: 387 (dpkg)
 $$:      $$$.     -   ,d$$' Shell: bash 5.2.15
 $$;      Y$b.     _   ,d$P' Resolution: 1152x720
 Y$$      `."Y$$$$P"' Terminal: /dev/tty1
 `$$b     "-._      CPU: Intel Xeon X3430 (1) @ 2.394GHz
 `Y$$     _         GPU: 1c:00.0 Mat
 `Y$$     $$$b.     Memory: 39MiB /
  `Y$$b.
   `Y$$b.
    "Y$b._
     `""

-3 libgs-common libgs10 libgs10-common libharfb
root@neofetch:~#
```

Voilà donc un conteneur totalement fonctionnelle sous debian 12.

Bonus Mise en place de la Haute Disponibilité

Un conteneur peut avoir aussi de la haute disponibilité. En me basant sur une configuration faite précédemment "[3- Création d'un cluster](#)".

Je vais donc dans "**Datacenter**" / "**Centre de données**" puis "**HA**" :

ID	État	Nœud	Nom	Nombre m...	Nombre m...	Groupe	Description
vm:100	stopped	HP-PROLI...	Machine-e...	3	3	HA	

Je vais donc sélectionner pour "VM" 103 (qui est neofetch), le groupe celui que j'avais créé "HA" puis je met le nombre de déménagement et redémarrages maximal à 3 :

Ajouter: Ressource: Conteneur/Machine virtuelle

VM: 103 Groupe: HA

Nombre maximum de redémarrages: 3 État de la demande: started

Nombre max. de déménagements: 3

Commentaire:

Ajouter

Voilà donc rajouté :

Ajouter Editer Supprimer						
ID	État	Nœud	Nom	Nombre m...	Nombre m...	Groupe
vm:100	stopped	HP-PROLI...	Machine-e...	3	3	HA
ct:103	started	HP-PROLI...	neofetch	3	3	HA

Test :

J'ai donc débranché le câble réseau du serveur HP proliant ml110 G5 et on voit que la HP 400 prend bien le relais (ce qui fait que c'est lent c'est mon routeur de test)