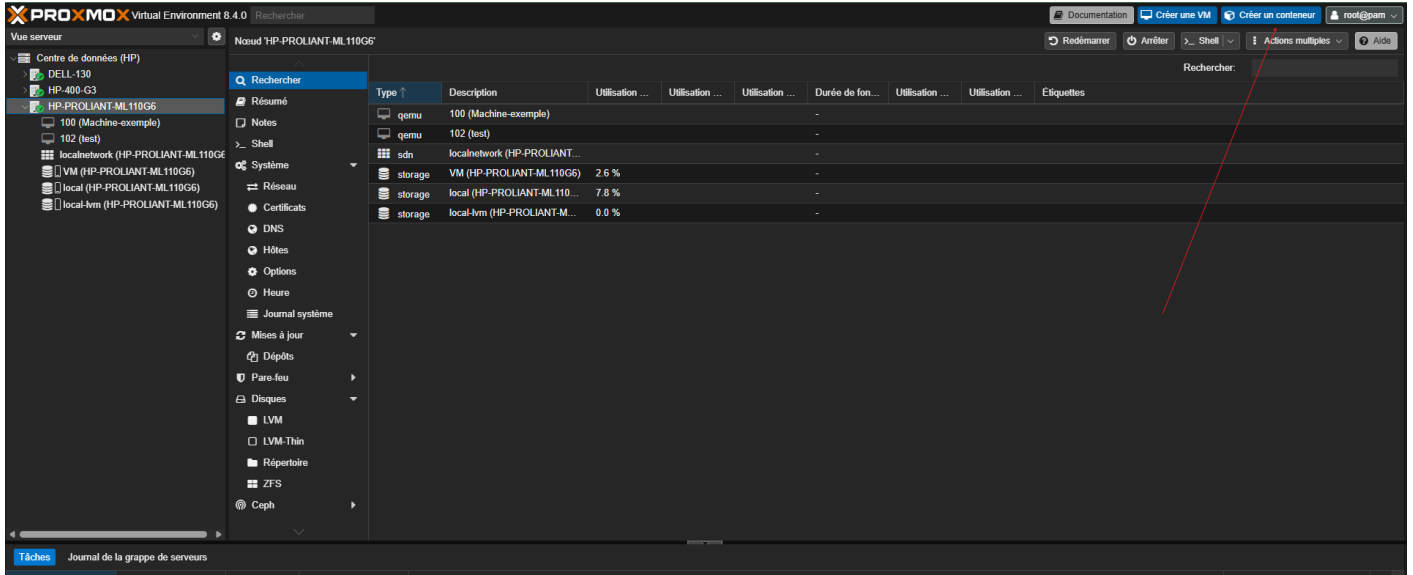


Mise en place d'un conteneur

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

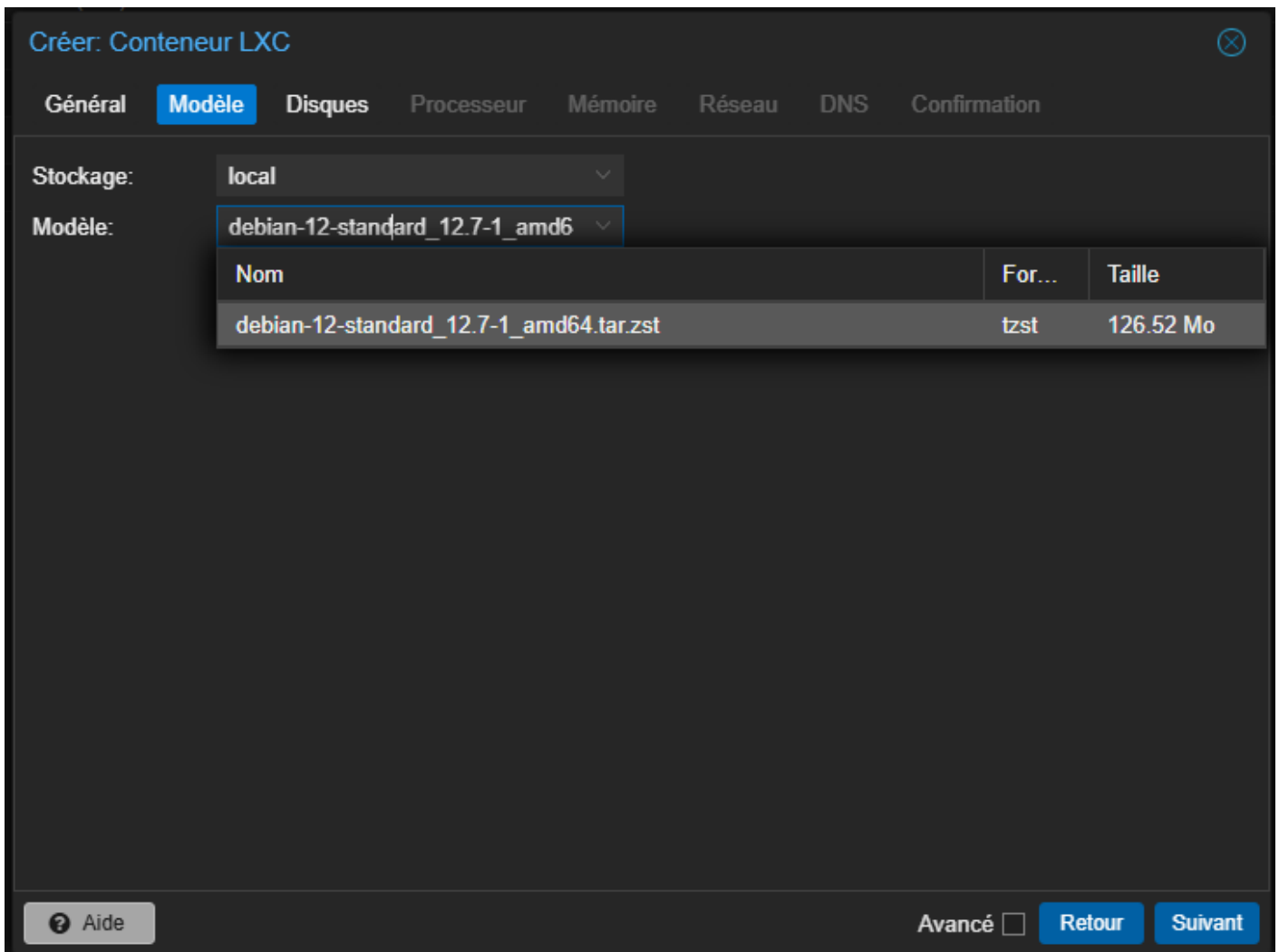
Nœud:	HP-PROLIANT-ML110G6	Pool de ressources:	
CT ID:	103	Mot de passe:	*****
Nom d'hôte:	neofetch	Confirmer le mot de passe:	*****
Conteneur non privilégié:	<input checked="" type="checkbox"/>	Clef(s) SSH publique(s):	
Imbriqué:	<input checked="" type="checkbox"/>		

[Charger le fichier de clef SSH](#)

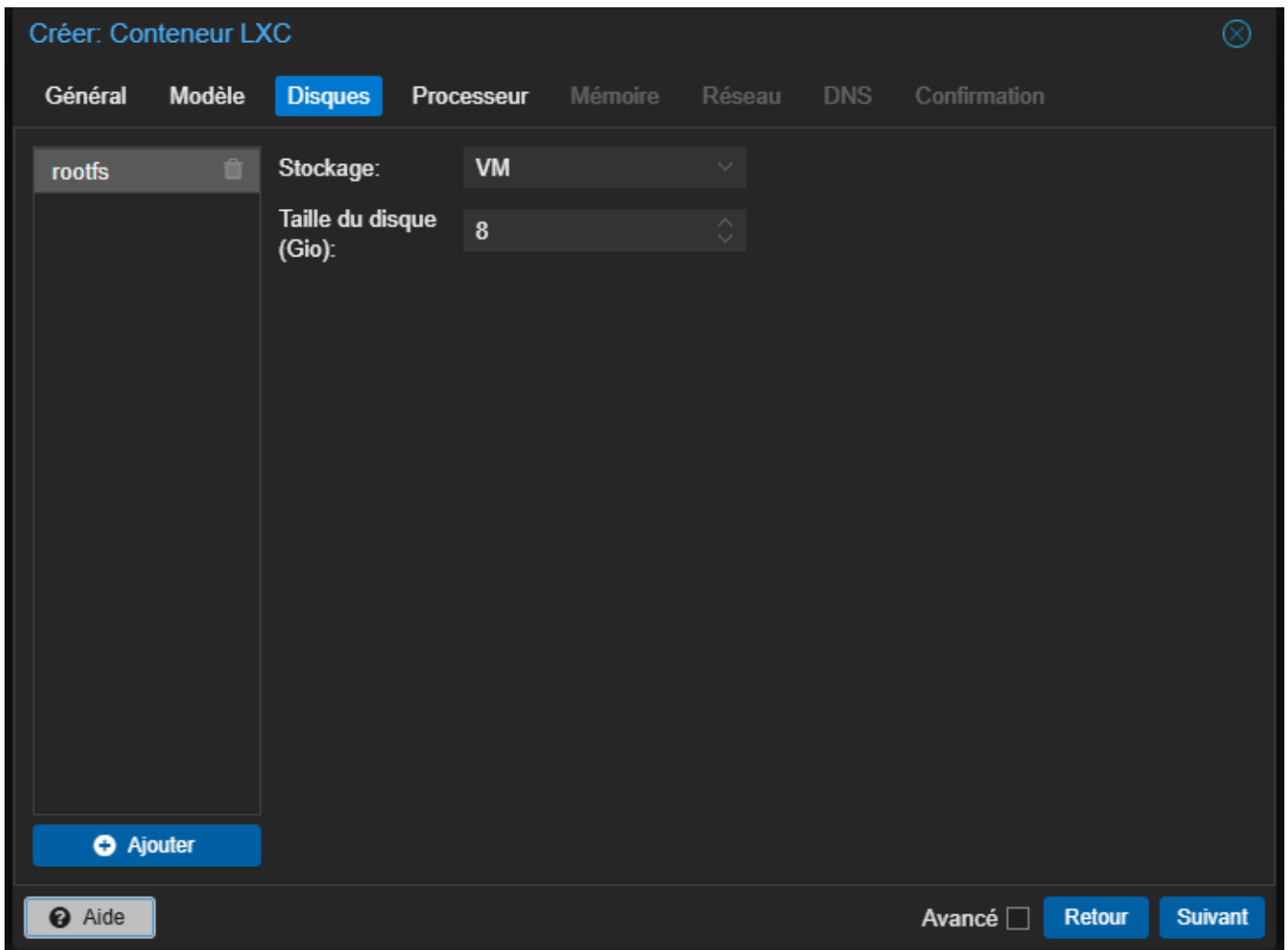
[? Aide](#) Avancé [Retour](#) [Suivant](#)

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.

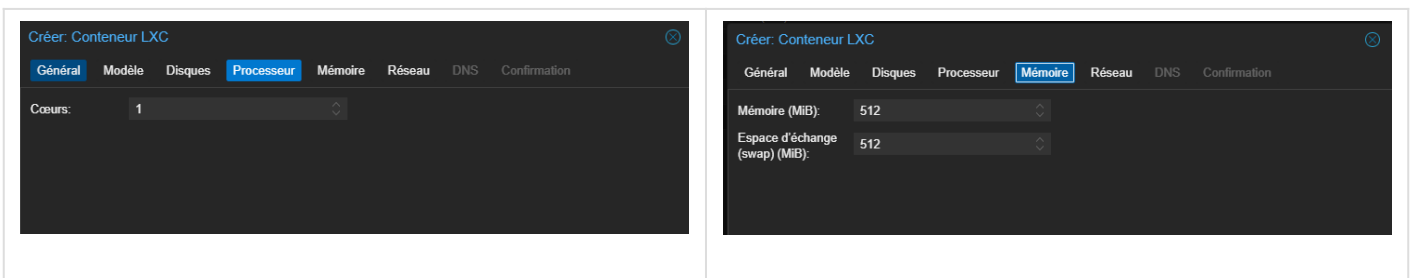
Puis on passe à l'onglet "Modèle" où nous sélectionnons le stockage où se trouve le modèle téléchargé précédemment, puis on le choisit :



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.



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 :



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:	<input type="text" value="eth0"/>	IPv4:	<input checked="" type="radio"/> Statique <input type="radio"/> DHCP
Adresse MAC:	<input type="text" value="auto"/>	IPv4/CIDR:	<input type="text" value="192.168.3.21/24"/>
Pont (bridge):	<input type="text" value="vibr0"/>	Passerelle (IPv4):	<input type="text" value="192.168.3.1"/>
Étiquette de VLAN:	<input type="text" value="aucun VLAN"/>	IPv6:	<input type="radio"/> Statique <input checked="" type="radio"/> DHCP <input type="radio"/> SLAAC
Pare-feu:	<input checked="" type="checkbox"/>	IPv6/CIDR:	
		Passerelle (IPv6):	

Avancé

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 ✕

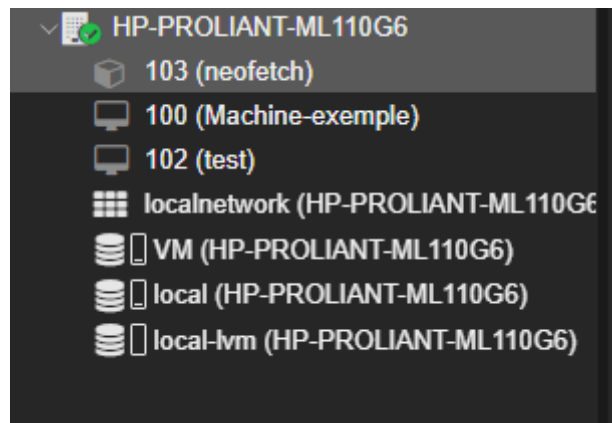
Général Modèle Disques Processeur Mémoire Réseau DNS **Confirmation**

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
 $$;         Y$b.     ,dP'    Resolution: 1152x720
 Y$$        `."Y$$$$P"'    Terminal: /dev/tty1
 `$$b      "-.____
 `Y$$
 `Y$$
  $$b.
  `Y$$b.
   "Y$b._
    `""

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

Voilà donc un conteneur totalement fonctionnelle sous debian 12.

Revision #6
Created 2025-04-24 20:23:12 UTC by Renard
Updated 2025-04-24 21:36:21 UTC by Renard