

NOUVEAU

GUIDE PRATIQUE

VOLUME 1

XPROXMOX



cloud-init

AUTOMATISER LA CREATION D'UNE VM AVEC

CLOUD-INIT

L'acquisition des compétences par l'exemple

POINTS CLES DU TUTORIEL



Installation pas à pas



Préparation de l'environnement de travail



Préparation Cloud-Init (sécurité et outils)



Création manuelle d'un « template » Cloud-Init



Création manuelle d'un clone intégral et tests de fonctionnement

Note :

Ce guide a été testé avec © Proxmox 9.1.1 et Debian 13 (« Trixie »).

L'objectif de ce guide est d'automatiser la création de machines virtuelles Debian 13 sur Proxmox 9 à l'aide du module « cloud-init ».

1 – PREPARATION DE LA MACHINE DEBIAN QUI SERVIRA DE MODELE

Dans votre hyperviseur Proxmox, créez la machine virtuelle Debian suivante :

MACHINE DEBIAN 13 (qui servira pour la création du « template »)

Résumé	Ajouter	Supprimer	Éditer	Action disque	Revenir en arrière
Console	Mémoire	2.00 Gio [balloon=0] [allow-ksm=0]			
Matériel	Processeurs	2 (1 sockets, 2 cores) [x86-64-v2-AES]			
Cloud-Init	BIOS	OVMF (UEFI)			
Options	Affichage	Par défaut			
Historique des tâches	Machine	q35			
Moniteur	Contrôleur SCSI	VirtIO SCSI single			
Sauvegarde	Lecteur CD/DVD (ide2)	local:iso/debian13-2.iso,media=cdrom,size=784M			
Réplication	Disque dur (scsi0)	STOCKAGE-VMs:vm-1000-disk-1,cache=writeback,discard=on,iotread=1,size=20G,ssd=1			
Instantanés	Carte réseau (net0)	virtio=BC:24:11:B9:F7:9A,bridge=lanSIO2			
	Disque EFI	STOCKAGE-VMs:vm-1000-disk-0,efitype=4m,ms-cert=2023,pre-enrolled-keys=1,size=1M			

Remarques : la machine Debian a été créée avec un Bios de type « OVMF » avec gestion du démarrage UEFI plus moderne et rapide que la version de Bios par défaut et avec une machine de type « q35 ».

CONFIGURATION DE LA MÉMOIRE VIVE

Mémoire (MiB): 2048

Mémoire minimale (MiB): 2048

Partages: Par défaut (100%)

Élasticité mémoire (ballooning): ☐

Allow KSM: ☐

Désactiver le « ballooning » et le « KSM » au niveau de la mémoire vive pour éviter que la machine ait des lenteurs ou fige lorsque l'espace de stockage est de type « ZFS ».

CONFIGURATION DU DISQUE DUR VIRTUEL

Disque Bande passante

Image disque: STOCKAGE-VMs:vm-1000- Cache:

Abandonner: ☒

IO thread: ☒

Émulation de SSD: ☒

En lecture seule: ☐

Sauvegarde: ☒

Passer la réplication: ☐

E-S asynchrones: Par défaut (io_uring)

Activer « Abandonner », « IO thread » et « Emulation SSD »

La case « **Abandonner** » permet de n'utiliser que l'espace réellement occupé par la machine virtuelle sur l'espace de stockage. Par exemple, si votre disque de base est de 30 Go et que votre machine n'utilise que 15 Go d'espace, seuls ces 15 Go seront véritablement utilisés sur l'espace de stockage.

2 – INSTALLATION ET CONFIGURATION D'UNE MACHINE DEBIAN 13.2 « CLI »

Préparez une machine Debian 13, sans interface graphique, qui nous servira de base pour notre futur « template ». **Pensez à installer SSH, « sudo » et « curl » sur la machine.** Connectez-la à votre LAN afin de voir si tout est opérationnel (adresse IP allouée dynamiquement par IPFire et accès internet) :

Vérifiez que votre machine a bien reçu une adresse IP dynamique dans l'étendue du VLAN tagué (ici le VLAN 10) et que vous pouvez sortir sur Internet.

3 – PREPARATION « CLOUD-INIT » DE LA MACHINE DEBIAN 13

1^{ère} étape :

Il faut maintenant « préparer » la machine Debian en installant des « outils » sur la machine Debian 13 à l'aide des commandes suivantes (ici, nous en profitons pour installer « curl » et « sudo ») :

```
sudo apt update && apt upgrade -y  
sudo apt install qemu-guest-agent cloud-init curl sudo -y
```

Activez et démarrez « **qemu-guest-agent** » avec les commandes suivantes :

```
sudo systemctl enable qemu-guest-agent  
sudo systemctl start qemu-guest-agent
```

Le « guest agent » permet à Proxmox de communiquer avec la VM (IP, sorties, live status). « Cloud-Init » permet l'injection automatique de configuration.

2^{ème} étape :

S'assurer du datasource Proxmox (**OBLIGATOIRE**). En effet, Cloud-Init doit savoir qu'il doit lire le disque « cidata » (cloud-init) fourni par Proxmox. Pour cela, créez le fichier « **90-datasource.cfg** » avec la commande suivante :

```
sudo echo "datasource_list: [ NoCloud, ConfigDrive ]" | sudo tee /etc/cloud/cloud.cfg.d/90-datasource.cfg
```

Vérifiez que le fichier est bien créé avec la commande suivante :

```
sudo cat /etc/cloud/cloud.cfg.d/90-datasource.cfg
```

Vous devez obtenir ceci :

```
root@debian13-base:~# cat /etc/cloud/cloud.cfg.d/90-datasource.cfg  
datasource_list: [ NoCloud, ConfigDrive ]
```

3^{ème} étape :

Désactivez la configuration réseau interne de Cloud-Init car **Cloud-Init ne doit pas réécrire le réseau** → c'est **systemd-networkd** le service réseau de Debian 13 qui s'en charge. Pour cela, saisissez la commande suivante :

```
sudo echo "network: {config: disabled}" | sudo tee /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

Vérifiez que le service réseau de Cloud-Init a bien été désactivé avec la commande suivante :

```
sudo cat /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

```
root@debian13-base:~# cat /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
network: {config: disabled}
```

4^{ème} étape :

Nettoyez l'état Cloud-Init (important avant conversion en template) avec la commande suivante :

```
sudo cloud-init clean --logs
```

5^{ème} étape :

Supprimez les clés SSH host (pour éviter les clones identiques) avec les commandes suivantes :

```
sudo rm -f /etc/ssh/ssh_host_*
sudo dpkg-reconfigure openssh-server
```

Vous devriez obtenir ceci :

```
root@debian13-base:~# dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
3072 SHA256:L9ytnuNdaWygUDGgKmyVqVTib0W1/Lx++5mz1ja2ajw root@debian13-base (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:n1DWSLbm6vj13wvpefUyy5qzb0J0Ui2Q0ylIo9L6aY root@debian13-base (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:gWMYwEitRAoLW0FFvNC5RY0IBmQr4+L2ekNJC/hW8eE root@debian13-base (ED25519)
ssh.socket is a disabled or a static unit not running, not starting it.
root@debian13-base:~# _
```

6^{ème} étape :

Vérifiez que « **systemd-networkd** » est bien actif sur la machine Debian 13 avec la commande suivante :

```
sudo systemctl status systemd-networkd
```

Vous devez obtenir ceci :

```
systemd-networkd.service - Network Configuration
Loaded: loaded (/usr/lib/systemd/system/systemd-networkd.service; disabled; preset: enabled)
Active: inactive (dead)
TriggeredBy: systemd-networkd.socket
Docs: man:systemd-networkd.service(8)
      man:org.freedesktop.network1(5)
FD Store: 0 (limit: 512)
root@debian13-base:~#
```

Remarque : si « **systemd-networkd** » n'est pas actif, saisissez les commandes suivantes :

```
sudo systemctl enable systemd-networkd
sudo systemctl start systemd-networkd
```

7^{ème} étape :

Lancez un « nettoyage » final avec les commandes suivantes :

```
sudo apt autoremove -y  
sudo apt clean
```

8^{ème} étape (fin) :

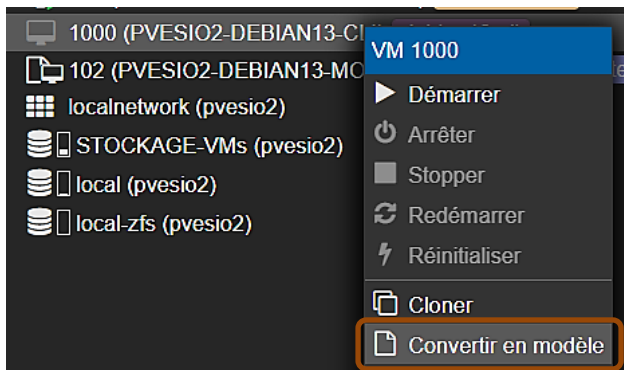
Arrêtez « proprement » la machine Debian 13 avec la commande suivante :

```
sudo shutdown -h now
```

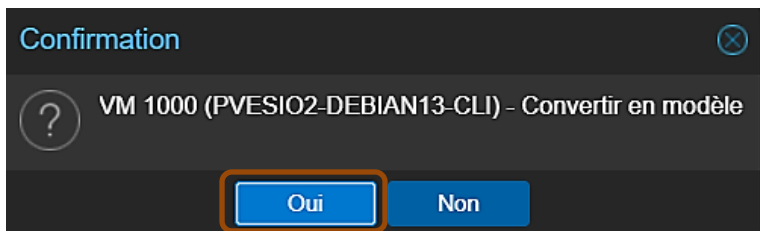
4 – CONVERSION DE LA MACHINE DEBIAN 13 DE BASE EN « TEMPLATE » CLOUD-INIT (modèle)

1^{ère} étape :

- Faites un clic droit sur votre machine virtuelle Debian 13
- Cliquez sur « **Convertir en modèle** »



- Confirmez la conversion en cliquant le bouton « **Oui** » :

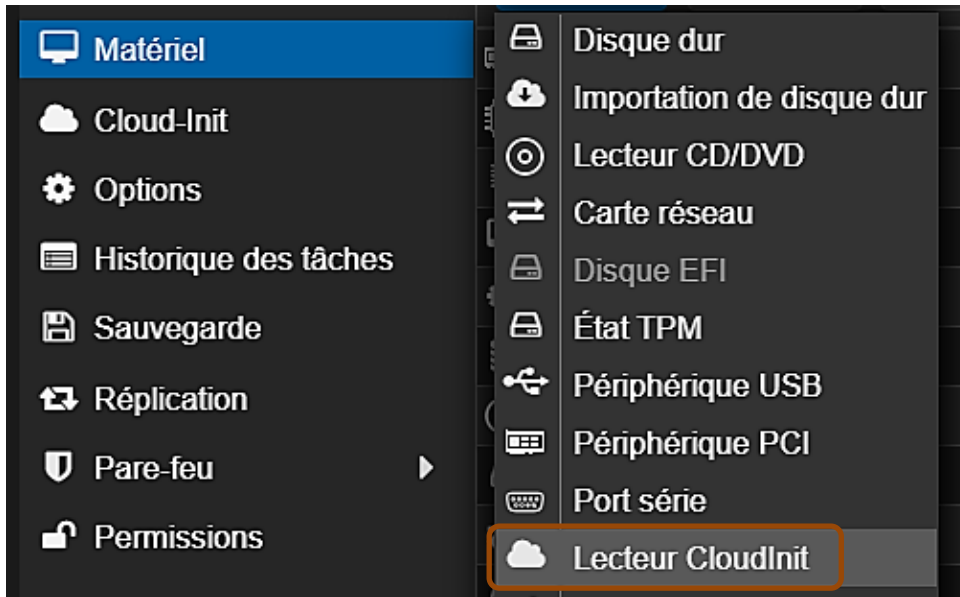


La conversion est assez rapide et l'icône de votre machine a changé d'aspect :

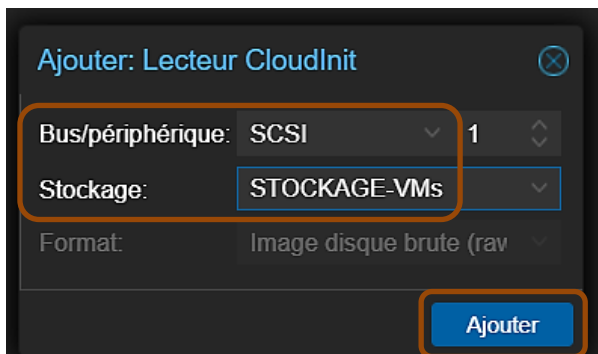


2^{ème} étape :

- Ajoutez le disque « Cloud-Init » au template ; pour cela :
 - Cliquez sur le template et sur « **Matériel** » dans le volet de droite
 - Cliquez le bouton « **Ajouter** » et cliquez sur « **Lecteur CloudInit** » :



- Sélectionnez le « **Bus/périphérique** » ; ici nous avons choisi « **SCSI** »
- Sélectionnez votre stockage dans lequel les machines virtuelles sont créées
- Cliquez le bouton « **Ajouter** » :

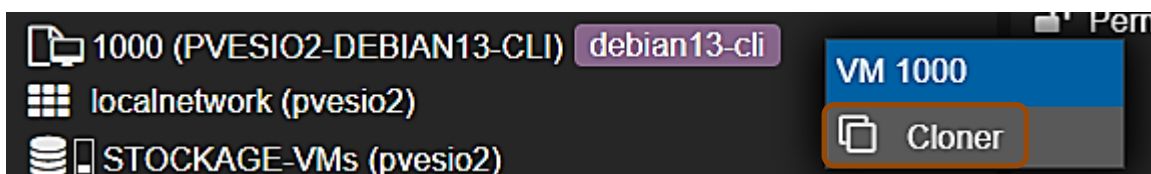


Vérifiez que le lecteur « **CloudInit** » est bien présent :



5 – LANCEMENT DU CLONE MANUELLEMENT

- Lancez le clone en faisant un clic droit sur le template et en cliquant « **Cloner** » :



- Complétez la fenêtre en saisissant un « **VM ID** », un « **Nom** » et en sélectionnant le mode « **Clone intégral** »
- Cliquez le bouton « **Cloner** » et patientez :

Clone VM Template 1000

Nœud cible: pvesio2 Mode: Clone intégral

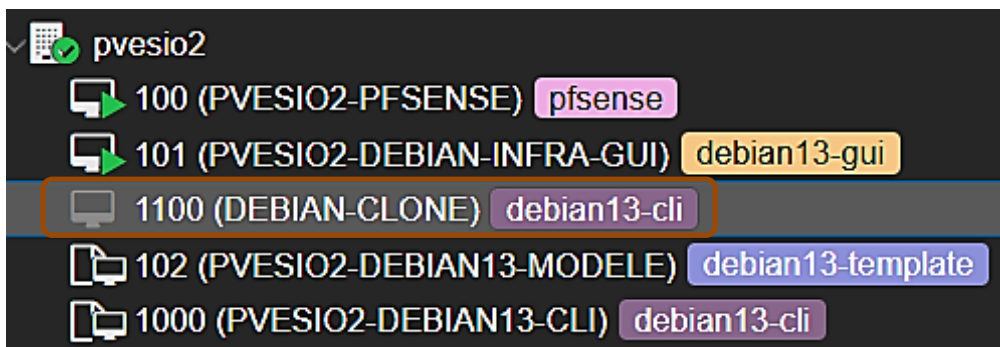
VM ID: 1100 Stockage cible: Identique à la source

Nom: DEBIAN-CLONE Format: Image au format QEMU

Pool de ressources:

Aide Cloner

Au bout de quelques instants une machine est générée :



- Cliquez sur votre clone et, dans la fenêtre de droite, cliquez sur « **Cloud-Init** » ; la fenêtre affiche ceci :

Machine virtuelle 1100 (DEBIAN-CLONE) sur le nœud pvesio2 debian13-cli

Résumé Console Matériel **Cloud-Init** Options Historique des tâches Moniteur

Supprimer Éditer Régénérer l'image

Utilisateur	Par défaut
Mot de passe	aucun
Domaine DNS	utiliser les valeurs de l'hôte
Serveurs DNS	utiliser les valeurs de l'hôte
Clef SSH publique	aucun
Mettre à jour les paquets	Oui
Configuration IP (net0)	

- Double-cliquez sur « **Utilisateur** » et saisissez un utilisateur pour la machine clonée :

Éditer: Utilisateur

Utilisateur: sid

OK

- Faites de même pour le mot de passe et cliquez « **OK** » :

Editer: Mot de passe

Mot de passe:

OK

- Double-cliquez sur « **Configuration IP (net0)** » et sélectionnez « **DHCP** » :

- Assurez-vous que dans « **Matériel** » votre machine est bien connectée au LAN :

- Faites démarrer la machine et identifiez-vous avec les identifiants saisis précédemment :

```
Debian GNU/Linux 13 DEBIAN-CLONE tty1
DEBIAN-CLONE login: sio
Password:
```

La session s'ouvre ; vérifiez que vous avez bien reçu une adresse IP dynamique dans votre LAN :

```
sio@DEBIAN-CLONE:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:60:d9:a4 brd ff:ff:ff:ff:ff:ff
    altname enp6s18
    altname enxbc241160d9a4
    inet 10.0.10.106/24 brd 10.0.10.255 scope global dynamic noprefixroute ens18
        valid_lft 7087sec preferred_lft 6187sec
    inet6 fe80::9630:c4cc:ee7:7bd3/64 scope link
        valid_lft forever preferred_lft forever
sio@DEBIAN-CLONE:~$ _
```

La machine est prête !