

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 13.2 (« Trixie »).

L'objectif de ce guide est d'automatiser la création de machines virtuelles Debian 13.2 sur Proxmox 9.1 à l'aide du module « cloud-init » et d'un script d'automatisation. Le script de déploiement devra proposer :

- La génération d'un numéro ID automatique avec incrémentation
- Un nom de machine (hostname) lié au nom d'utilisateur
- un mot de passe personnalisé
- l'activation du service SSH avec un port personnalisé (port 22 + numéro du VLAN)
- la connexion automatique de la machine au VLAN pfSense sélectionné
- l'attribution dynamique d'une adresse IP en fonction du VLAN choisi

1 – PREPARATION DE L'ENVIRONNEMENT DE TRAVAIL

Dans votre hyperviseur Proxmox, créez les 3 machines virtuelles suivantes :

- 1 machine virtuelle pfSense avec une interface **WAN** et une interface **LAN** qui gèrera les VLANs
- 1 machine virtuelle Debian 13.2 avec une interface graphique pour la gestion de pfSense
- 1 machine virtuelle Debian 13.2 de base (sans interface graphique)

Caractéristiques des machines :

MACHINE pfsense

>_ Console	Mémoire	2.00 Gio [balloon=0] [allow-kvm=0]
Matériel	Processeurs	2 (1 sockets, 2 cores) [host]
Cloud-Init	BIOS	Par défaut (SeaBIOS)
Options	Affichage	Par défaut
Historique des tâches	Machine	Par défaut (i440fx)
Moniteur	Contrôleur SCSI	VirtIO SCSI single
Sauvegarde	Lecteur CD/DVD (ide2)	local:iso/pfsense-2-7-2.iso,media=cdrom,size=854172K
Réplication	Disque dur (scsi0)	STOCKAGE-VMs:vm-100-disk-0,cache=writeback,discard=on,iothread=1,size=32G
Instantanés	Carte réseau (net0)	virtio=BC:24:11:B0:C6:50,bridge=vbr0
	Carte réseau (net1)	virtio=BC:24:11:5E:F9:65,bridge=lanSIO2

Configuration des interfaces WAN et LAN sur pfSense (exemple) :

```
*** Welcome to pfSense 2.8.1-RELEASE (amd64) on pfSenseSI02 ***

WAN (wan)          -> vtnet0          -> v4: 192.168.168.140/24
LAN (lan)           -> vtnet1          -> v4: 192.168.100.254/24
VLAN10_ALPHA (opt1) -> vtnet1.10       -> v4: 10.0.10.254/24
VLAN20_BEN (opt2)   -> vtnet1.20       -> v4: 10.0.20.254/24
VLAN30_ZAC (opt3)   -> vtnet1.30       -> v4: 10.0.30.254/24
VLAN40_KERIAN (opt4) -> vtnet1.40       -> v4: 10.0.40.254/24
VLAN50_PAULC (opt5) -> vtnet1.50       -> v4: 10.0.50.254/24
VLAN60_HUGO (opt6)  -> vtnet1.60       -> v4: 10.0.60.254/24
VLAN70_PAULR (opt7) -> vtnet1.70       -> v4: 10.0.70.254/24
VLAN80_MALO (opt8)  -> vtnet1.80       -> v4: 10.0.80.254/24
VLAN100_PROF (opt9) -> vtnet1.100      -> v4: 10.0.100.254/24
```

MACHINE DEBIAN 13 (avec une interface graphique pour la gestion de pfSense)

Résumé	Ajouter	Supprimer	Éditer	Action disque	Revenir en arrière
Console	Mémoire	4.00 Gio [balloon=0] [allow-kvm=0]			
Matériel	Processeurs	2 (1 sockets, 2 cores) [host]			
Cloud-Init	BIOS	Par défaut (SeaBIOS)			
Options	Affichage	Par défaut			
Historique des tâches	Machine	Par défaut (i440fx)			
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-101-disk-0,discard=on,iotthread=1,size=32G,ssd=1			
	Carte réseau (net0)	virtio=BC:24:11:72:05:AB,bridge=lanSIO2			

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-kvm=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,iotthread=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 ».

Sur notre serveur, les machines virtuelles sont stockées dans un espace de stockage nommé « STOCKAGE-VMs ». Ce stockage est composé de 2 disques SSD de 480 Go qui ont été configuré en RAID 1 dans Proxmox :

ID ↑	Type	Contenu	Chemin d'accès/Cible	Partagé	Activé
STOCKAGE-VMs	ZFS	Image disque, Conteneur		Non	Oui

Statut: STOCKAGE-VMs

Recharger

État de santé

ONLINE

Erreurs

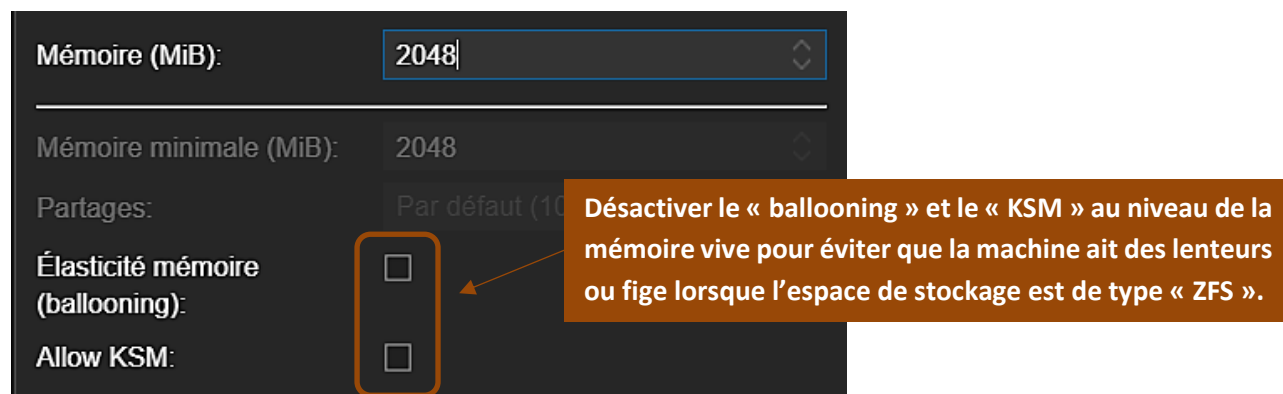
No known data errors

Périphériques

Nom	État de santé	READ	WRITE	CKSUM
STOCKAGE-VMs	ONLINE	0	0	0
mirror-0	ONLINE	0	0	0
/dev/disk/by-id/ata-CT500BX500SSD...	ONLINE	0	0	0
/dev/disk/by-id/ata-CT500BX500SSD...	ONLINE	0	0	0

Etant donné que le type de stockage est « **ZFS** » (raid mirroring géré par Proxmox), il est conseillé d'activer/désactiver les options suivantes dans la machine virtuelle Debian 13.2 qui servira de base à la création du « **template** » :

CONFIGURATION DE LA MÉMOIRE VIVE



Mémoire (MiB): 2048

Mémoire minimale (MiB): 2048

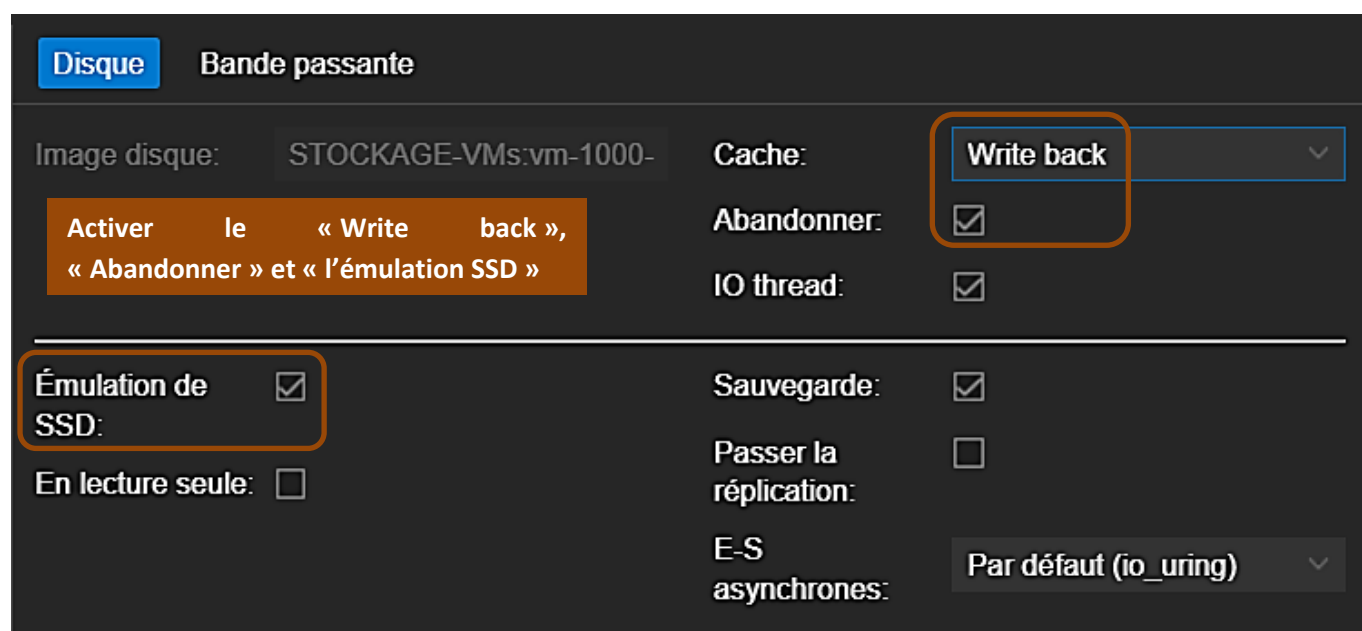
Partages: Par défaut (1024)

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: Write back

Abandonner: ☒

IO thread: ☒

Émulation de SSD: ☒

En lecture seule: ☐

Sauvegarde: ☒

Passer la réplication: ☐

E-S asynchrones: Par défaut (io_uring)

Activer le « Write back », « Abandonner » et « l'émulation SSD »

Remarque : si votre stockage est de type « **ZFS** » (raid Proxmox), il est vivement recommandé d'activer le cache du disque dur virtuel en mode « **Write Back** ».

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 ZFS.

2 – INSTALLATION DE pfSENSE ET CONFIGURATION DE pfSENSE AVEC UNE MACHINE DEBIAN GRAPHIQUE

A l’aide de vos connaissances/compétences, préparez votre routeur/pare-feu pfSense (voir tutoriels sur notre site <https://tutos-info.fr>). Une fois votre pfSense installé, configurez-le à l’aide de votre machine Debian graphique.

Exemple de configuration des interfaces :

Interface Assignments

Interface Groups

Wireless

VLANs

QinQs

PPPs

GREs

GIFs

Bridges

LAGGs

Interface	Network port
WAN	vtnet0 (bc:24:11:b0:c6:50)
LAN	vtnet1 (bc:24:11:5e:f9:65)
vlan10_ALPHA	VLAN 10 on vtnet1 - lan (vlanALPHA)
vlan20_BEN	VLAN 20 on vtnet1 - lan (vlanBEN)
vlan30_ZAC	VLAN 30 on vtnet1 - lan (vlanZAC)
vlan40_KERIAN	VLAN 40 on vtnet1 - lan (vlanKERIAN)
vlan50_PAULC	VLAN 50 on vtnet1 - lan (vlanPAULC)
vlan60_HUGO	VLAN 60 on vtnet1 - lan (vlanHUGO)
vlan70_PAULR	VLAN 70 on vtnet1 - lan (vlanPAULR)
vlan80_MALO	VLAN 80 on vtnet1 - lan (vlanMALO)
vlan100_PROF	VLAN 100 on vtnet1 - lan (vlanPROF)

Nous avons une interface « WAN » pour la sortie sur Internet et une interface « LAN » dans laquelle nous avons ajouté plusieurs VLANs (1 VLA par étudiant).

La machine Debian graphique avec laquelle nous administrons pfSense est connectée à l’interface LAN et n’ait pas dans un VLAN.

Interface Assignments

Interface Groups

Wireless

VLANs

QinQs

PPPs

GREs

GIFs

VLAN Interfaces

Interface	VLAN tag	Priority	Description
vtnet1 (lan)	10	Plusieurs VLANs ont été créés ici avec l’architecture réseau suivante : VLAN10 : 10.0.10.254/24 DHCP VLAN10 : 10.0.10.100 → 10.0.10.150 VLAN20 : 10.0.20.254/24 DHCP VLAN10 : 10.0.20.100 → 10.0.20.150 Etc. pour les autres VLANs	vlanALPHA
vtnet1 (lan)	20		vlanBEN
vtnet1 (lan)	30		vlanZAC
vtnet1 (lan)	40		vlanKERIAN
vtnet1 (lan)	50		vlanPAULC
vtnet1 (lan)	60		vlanHUGO
vtnet1 (lan)	70		vlanPAULR
vtnet1 (lan)	80		vlanMALO
vtnet1 (lan)	100		vlanPROF

Remarque : Pensez à activer les services DHCP sur chaque VLANs.

Configuration des règles de pare-feu :

Vérifiez que le « NAT OUTBOUND » est bien en mode « Automatic outbound NAT » :

Firewall / NAT / Outbound

Port Forward1:1OutboundNPT

Outbound NAT Mode

Mode

Automatic outbound NAT rule generation.
(IPsec passthrough included)

Hybrid Outbound NAT rule generation.
(Automatic Outbound NAT + rules below)

Manual Outbound NAT rule generation.
(AON - Advanced Outbound NAT)

Disable Outbound NAT rule generation.
(No Outbound NAT rules)

Configuration des règles de pare-feu pour le VLAN10 :

- Commencez par bloquer les communications « inter-vlans » (règles « Block » en rouge au début)
- Configurez des règles « Pass » pour le VLAN10 vers pfSense et « Any » afin de sortir sur Internet (règles « Pass » en vert, à mettre à la fin) :

FloatingWANLANVLAN10_ALPHAVLAN20_BENVLAN30_ZACVLAN40_KERIANVLAN50_PAULCVLAN60_HUGOVLAN70_PAULRVLAN80_MALOVLAN100_PROF

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN20_BEN address	*	*	none			
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN30_ZAC address	*	*	none			
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN40_KERIAN address	*	*				
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN50_PAULC address	*	*				
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN60_HUGO address	*	*				
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN70_PAULR address	*	*				
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN80_MALO address	*	*				
☐	✗ 0/0 B	IPv4 *	VLAN10_ALPHA subnets	*	VLAN100_PROF address	*	*				
☐	✓ 0/5 KiB	IPv4 *	VLAN10_ALPHA subnets	*	This Firewall (self)	*	*	none			
☐	✓ 0/112 KiB	IPv4 *	VLAN10_ALPHA subnets	*	*	*	*	none			

Ici, nous empêchons la communication inter-vlan afin de créer des « bacs à sable » séparés pour chaque étudiant (règles de type « Block »).

- Activez le service DHCP sur chaque VLAN :

SettingsWANLANVLAN10_ALPHAVLAN20_BENVLAN30_ZACVLAN40_KERIANVLAN50_PAULCVLAN60_HUGOVLAN70_PAULRVLAN80_MALOVLAN100_PROF

General Settings

DHCP Backend

Kea DHCP

Enable

☒ Enable DHCP server on VLAN10_ALPHA interface

Deny Unknown Clients

Allow all clients

When set to **Allow all clients**, any DHCP client will get an IP address within this scope/range on this interface. If set to **Allow known clients from any interface**, any DHCP client with a MAC address listed in a static mapping on **any** scope(s)/interface(s) will get an IP address. If set to **Allow known clients from only this interface**, only MAC addresses listed in static mappings on this interface will get an IP address within this scope/range.

Ignore Client Identifiers

☐ Do not record a unique identifier (UID) in client lease data if present in the client DHCP request

This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.

Exemple d'étendue DHCP configurée pour le VLAN10 :

Primary Address Pool	
Subnet	10.0.10.0/24
Subnet Range	10.0.10.1 - 10.0.10.254
Address Pool Range	From 10.0.10.100 To 10.0.10.150
The specified range for this pool must not be within the range configured on any other address pool for this interface.	

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

Préparez une machine Debian 13.2, sans interface graphique, qui nous servira de base pour notre futur « template ». **Pensez à installer SSH sur la machine.** Connectez-la à un VLAN (par exemple le VLAN10) afin de voir si tout est opérationnel (adresse IP allouée dynamiquement par pfSENSE et accès internet) :

Pont (bridge):	lanSIO2	Modèle:	VirtIO (paravirtualisé)
Étiquette de VLAN:	10	Adresse MAC:	BC:24:11:B9:F7:9A
Pare-feu:	☐		

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 :

```
root@debian13-base:~# 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
ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:b9:f7:9a brd ff:ff:ff:ff:ff:ff
    altname enp6s18
    altname enxhc2411b9f79a
    inet 10.0.10.105/24 brd 10.0.10.255 scope global dynamic noprefixroute ens18
        valid_lft 7149sec preferred_lft 6249sec
    inet6 fe80::b9de:d78c:a6fe:aa20/64 scope link
        valid_lft forever preferred_lft forever
root@debian13-base:~# _
```

Test de ping :

```
root@debian13-base:~# ping -c3 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=9.09 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=9.37 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=8.62 ms

--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 8.616/9.023/9.368/0.310 ms
root@debian13-base:~# _
```

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 ») :

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

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

```
systemctl enable qemu-guest-agent  
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 :

```
echo "datasource_list: [ NoCloud, ConfigDrive ]" \> /etc/cloud/cloud.cfg.d/90-datasource.cfg
```

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

```
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 :

```
echo "network: {config: disabled}" \> /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 :

```
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 :

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

5^{ème} étape :

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

```
rm -f /etc/ssh/ssh_host_*  
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:gWMYwEitRAoLW0FFvNCSRY0IBmQr4+LZekNJC/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 :

```
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 :

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

7^{ème} étape :

Lancez un « nettoyage » final avec les commandes suivantes :

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

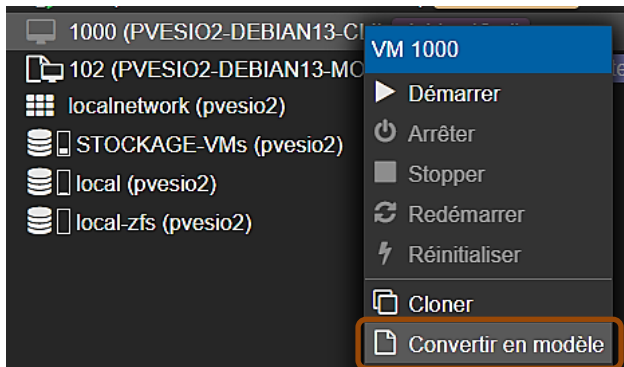
8^{ème} étape (fin) :

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

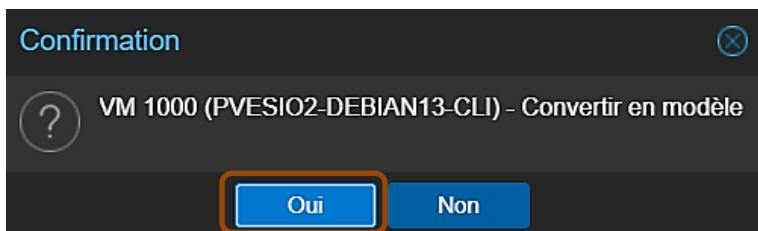
```
shutdown -h now
```

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** » :

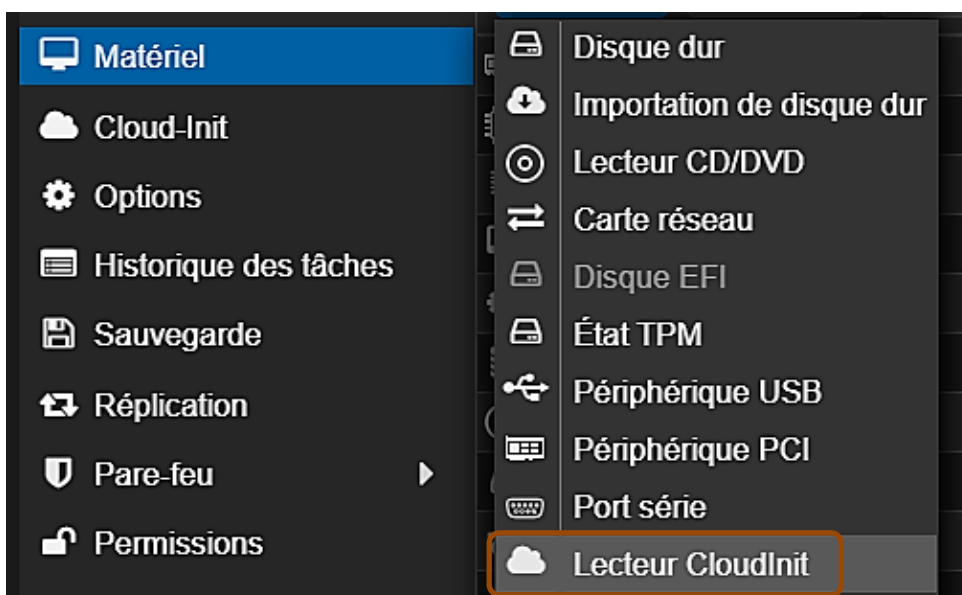


La conversion est assez rapide et l'icône de votre machine a changé d'aspect :

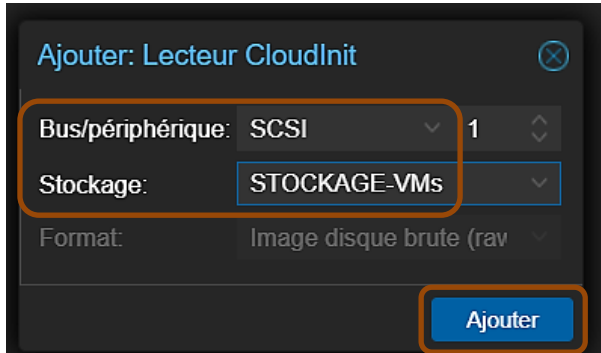


2^{ème} étape :

- 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** » :

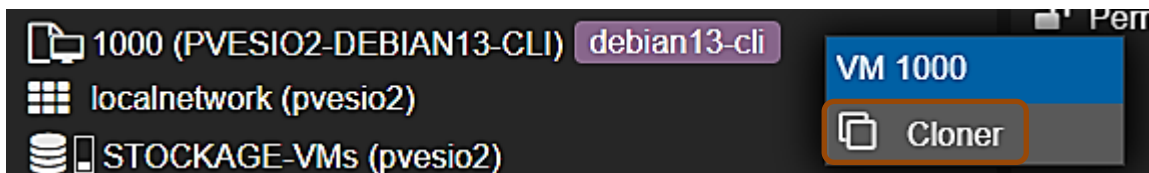


Vérifiez que le lecteur « **CloudInit** » est bien présent :

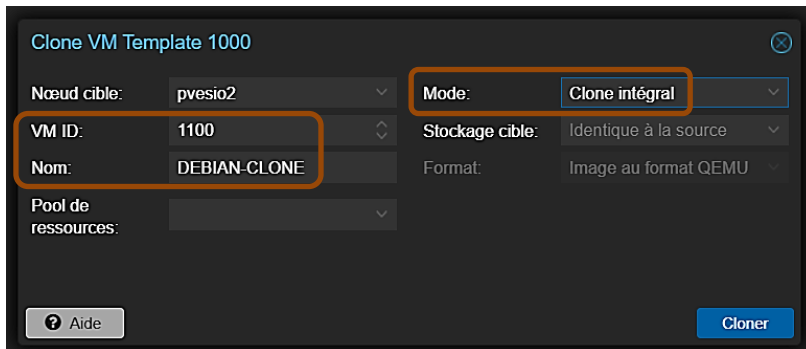


6 – LANCEMENT DU CLONE MANUELLEMENT

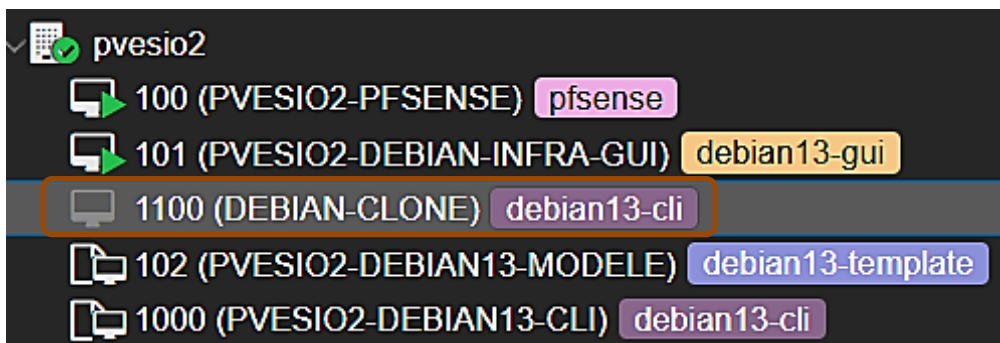
- Lancez le clone en faisant un clic droit sur le template et en cliquant « **Cloner** » :



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



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	Supprimer	Éditer	Régénérer l'image
Console	Utilisateur	Par défaut	
Matériel	Mot de passe	aucun	
Cloud-Init	Domaine DNS	utiliser les valeurs de l'hôte	
Options	Serveurs DNS	utiliser les valeurs de l'hôte	
Historique des tâches	Clef SSH publique	aucun	
Moniteur	Mettre à jour les paquets	Oui	
	Configuration IP (net0)		

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

Éditer: Utilisateur

Utilisateur:

OK

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

Éditer: Mot de passe

Mot de passe:

OK

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

Éditer: Configuration réseau

Carte réseau: **net0**

IPv4: ☐ Statique ☒ DHCP

IPv6: ☒ Statique ☐ DHCP ☐ Autoconfigur

IPv4/CIDR:

IPv6/CIDR:

Passerelle (IPv4):

Passerelle (IPv6):

OK

- Assurez-vous que dans « **Matériel** » votre VLAN10 a bien été tagué :

Éditer: Carte réseau

Pont (bridge):

Modèle:

Étiquette de VLAN:

Adresse MAC:

Pare-feu: ☐

- 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 VLAN :

```
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 enx0c241160d9a4
    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:~$ _
```

- Faites un test de ping vers un DNS pour vérifier votre connexion à Internet :

```
sio@DEBIAN-CLONE:~$ ping -c3 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=9.58 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=9.22 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=8.59 ms

--- 1.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 8.589/9.128/9.576/0.408 ms
sio@DEBIAN-CLONE:~$ ping -c3 1.1.1.1
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.
64 bytes from 1.1.1.1: icmp_seq=1 ttl=55 time=9.11 ms
64 bytes from 1.1.1.1: icmp_seq=2 ttl=55 time=9.02 ms
64 bytes from 1.1.1.1: icmp_seq=3 ttl=55 time=9.33 ms
```

La machine est prête !