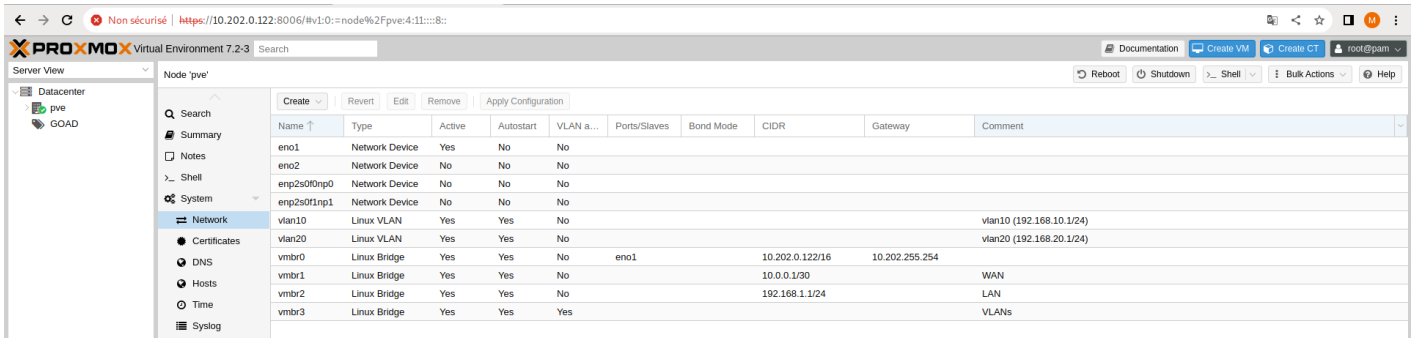


Installations supplémentaires : Pfsense, Packer, Terraform & Ansible

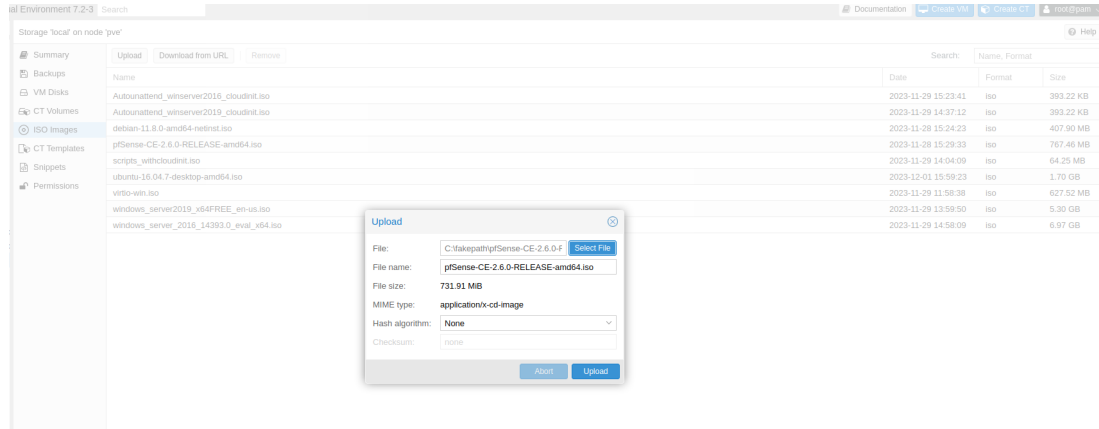
## Installation de GOAD

### 1. Première étape l'installation de Pfsense : IP 192.168.1.2

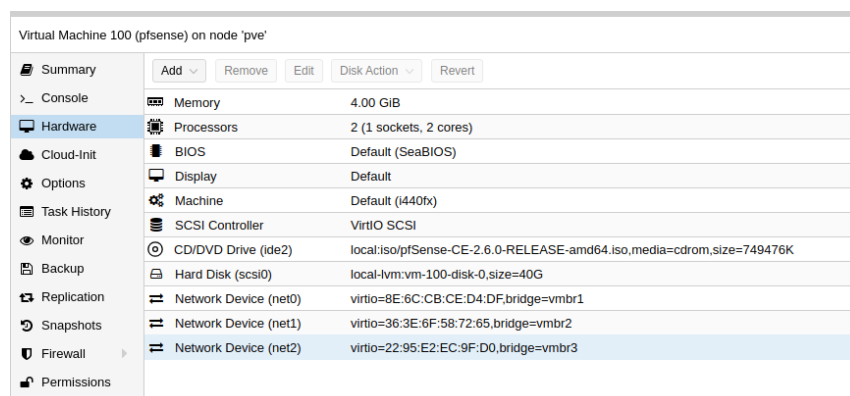
- On commence par ajouter des réseaux WAN, LAN et VLAN's au Proxmox.



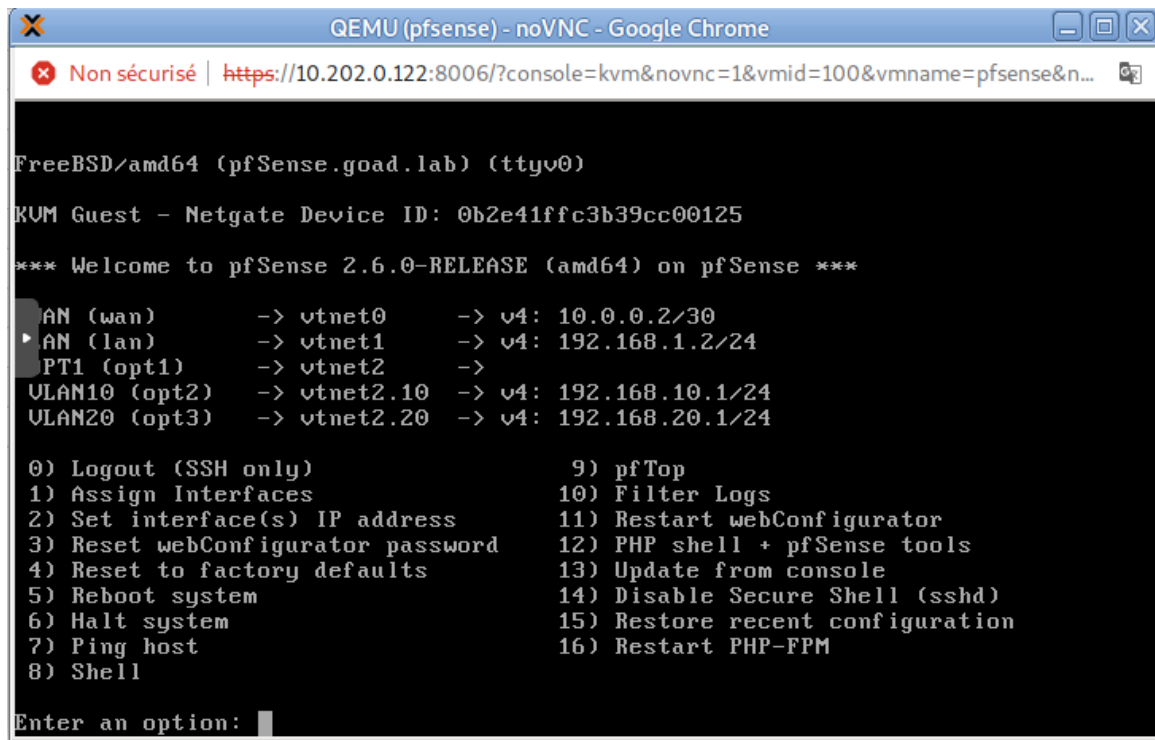
- Une fois que l'ajout est terminé on applique la configuration avec le "Apply configuration".
- Ensuite on installe l'ISO de pfSense sur Proxmox.



- On crée la VM et on lui ajoute des Network :



- On peut maintenant démarrer la VM et suivre l'installation.
- Une fois terminé on va configurer les premiers WAN, LAN et VLAN depuis cette VM, avec le menu "1) Assign Interfaces"



```

QEMU (pfsense) - noVNC - Google Chrome
Non sécurisé | https://10.202.0.122:8006/?console=kvm&novnc=1&vmid=100&vmname=pfsense&n...

FreeBSD/amd64 (pfSense.goad.lab) (ttyv0)
KVM Guest - Netgate Device ID: 0b2e41ffc3b39cc00125

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

  AN (wan)      -> vtnet0      -> v4: 10.0.0.2/30
  AN (lan)      -> vtnet1      -> v4: 192.168.1.2/24
  PT1 (opt1)    -> vtnet2      ->
  VLAN10 (opt2) -> vtnet2.10  -> v4: 192.168.10.1/24
  VLAN20 (opt3) -> vtnet2.20  -> v4: 192.168.20.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults    13) Update from console
5) Reboot system              14) Disable Secure Shell (sshd)
6) Halt system                15) Restore recent configuration
7) Ping host                  16) Restart PHP-FPM
8) Shell

Enter an option:

```

- On va se connecter a son interface graphique avec un forwarding depuis le PVE de proxmox pour y accéder depuis notre pc en localhost sur le port 8082 avec la commande : `ssh -L 8082:192.168.1.2:80 root@10.202.0.122`



```

test@202-4:~$ ssh -L 8082:192.168.1.2:80 root@10.202.0.122
root@10.202.0.122's password:
Linux pve 5.15.30-2-pve #1 SMP PVE 5.15.30-3 (Fri, 22 Apr 2022 18:08:27 +0200) x
86_64

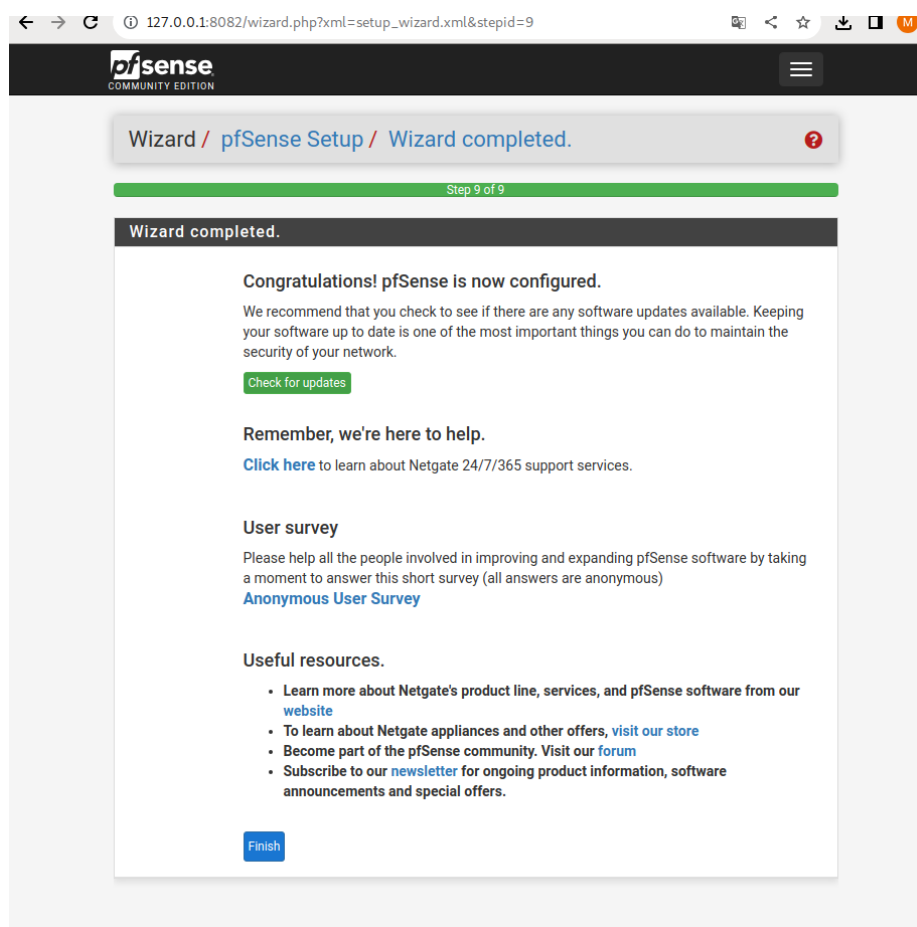
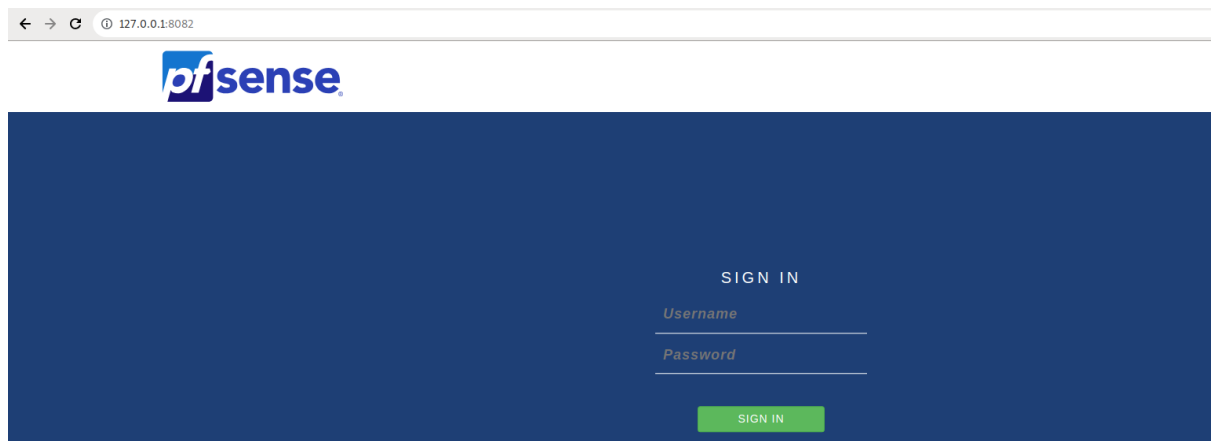
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  7 18:23:46 2023 from 10.202.4.1
root@pve:~#

```

- On peut maintenant accéder à l'interface graphique de pfsense et continuer l'installation

IP : 127.0.0.1:8082



- Maintenant on va pouvoir configurer les Firewall (LAN) :

**pfSense** COMMUNITY EDITION System ▾ Interfaces ▾ Firewall ▾ Services ▾ VPN ▾ Status ▾ Diagnostics ▾ Help ▾

Firewall / Rules / Edit

### Edit Firewall Rule

**Action**   
 Choose what to do with packets that match the criteria specified below.  
 Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

**Disabled** ☐ Disable this rule  
 Set this option to disable this rule without removing it from the list.

**Interface**   
 Choose the interface from which packets must come to match this rule.

**Address Family**   
 Select the Internet Protocol version this rule applies to.

**Protocol**   
 Choose which IP protocol this rule should match.

**Source**

**Source** ☐ Invert match   /

**Destination**

**Destination** ☐ Invert match   /

**Extra Options**

**Log** ☐ Log packets that are handled by this rule  
 Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).

**Description**

**pfSense** COMMUNITY EDITION System ▾ Interfaces ▾ Firewall ▾ Services ▾ VPN ▾ Status ▾ Diagnostics ▾ Help ▾

Firewall / Rules / LAN

Floating WAN LAN VLAN10 VLAN20

### Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	3 / 1.37 MiB	*	*	*	LAN Address	80 22	*	*		Anti-Lockout Rule	
<input type="checkbox"/>	0 / 1.50 GiB	IPv4+6 *	LAN net	*	*	*	*	none		Default allow LAN to any rule	

- On va ensuite ajouter des règles de NAT sur notre Proxmox :

```
echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

- Si cela renvoi 1 c'est que le forwarding est activé

- Autoriser icmp

```
iptables -t nat -A PREROUTING -i vmbr0 -p icmp -j ACCEPT
```

- Autoriser ssh

```
iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 22 -j ACCEPT
```

- Autoriser proxmox web

```
iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 8006 -j ACCEPT
```

- Tout rediriger vers pfsense

```
iptables -t nat -A PREROUTING -i vmbr0 -j DNAT --to 10.0.0.2
```

- Ajout SNAT WAN -> ip public

```
iptables -t nat -A POSTROUTING -o vmbr0 -j SNAT -s 10.0.0.0/30 --to-source 10.202.0.122
```

- On oublie pas d'enregistrer ces règles dans le fichiers /etc/network/save-iptables

- On va maintenant ajouter les interfaces des VLANs dans pfsense :

pfSense COMMUNITY EDITION

System ▾ Interfaces ▾ Firewall ▾ Services ▾ VPN ▾ Status ▾ Diagnostics ▾ Help ▾

Interfaces / VLANs

Interface Assignments Interface Groups Wireless **VLANs** QinQs PPPs GREs GIFs Bridges LAGGs

VLAN Interfaces				
Interface	VLAN tag	Priority	Description	Actions
vtnet2 (opt1)	10		VLAN10	
vtnet2 (opt1)	20		VLAN20	

Add

- On va ajouter un server dhcp pour les VLANs, en renseignant la range IP :

The screenshot shows the pfSense web interface for the DHCP Server configuration of the VLAN10 interface. The breadcrumb trail is Services / DHCP Server / VLAN10. The 'VLAN10' tab is selected under the 'General Options' section.

**General Options**

- Enable:** ☒ Enable DHCP server on VLAN10 interface
- BOOTP:** ☐ Ignore BOOTP queries
- 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 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 below (i.e. for this interface) will get an IP address within this scope/range.
- Ignore denied clients:** ☐ Denied clients will be ignored rather than rejected. This option is not compatible with failover and cannot be enabled when a Failover Peer IP address is configured.
- Ignore client identifiers:** ☐ If a client includes a unique identifier in its DHCP request, that UID will not be recorded in its lease. 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.
- Subnet:** 192.168.10.0
- Subnet mask:** 255.255.255.0
- Available range:** 192.168.10.1 - 192.168.10.254
- Range:** From 192.168.10.100 To 192.168.10.254

- On fait la même chose avec le VLAN20, en modifiant le range d'adresse également.
- On va mettre en place un alias dans le firewall pour les VLANS

The screenshot shows the pfSense web interface for the Firewall Aliases configuration. The breadcrumb trail is Firewall / Aliases / IP. The 'IP' tab is selected.

**Firewall Aliases IP**

Name	Values	Description	Actions
INTERNAL	192.168.1.1/16, 10.0.0.1/30, 10.10.10.0/24		

Buttons: Add Import

- On ajoute les règles Firewall pour le WAN et le VLAN10 :

The screenshot shows the PfSense Firewall Rules configuration for the WAN interface. The 'Rules' tab is selected, and the 'WAN' interface is chosen. The table lists the following rules:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 2.09 MiB	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogus networks	[Settings]
0 / 0 B	IPv4 TCP	10.0.0.1	*	192.168.1.3	22 (SSH)	*	none			[Anchor] [Edit] [Copy] [Delete]
0 / 0 B	IPv4 TCP	10.0.0.1	*	LAN address	80 (HTTP)	*	none			[Anchor] [Edit] [Copy] [Delete]

Buttons at the bottom: Add, Add, Delete, Save, Separator.

The screenshot shows the PfSense Firewall Rules configuration for the VLAN10 interface. The 'Rules' tab is selected, and the 'VLAN10' interface is chosen. The table lists the following rules:

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0 / 2.09 MiB	*	Reserved Not assigned by IANA	*	*	*	*	*	*	Block bogus networks	[Settings]
101 / 17.29 GiB	IPv4 *	VLAN10 net	*	! INTERNAL	*	*	none			[Anchor] [Edit] [Copy] [Delete]

Buttons at the bottom: Add, Add, Delete, Save, Separator.

- Résultat des interfaces Pfsense :

Interfaces			
WAN	↑	10Gbase-T <full-duplex>	10.0.0.2
LAN	↑	10Gbase-T <full-duplex>	192.168.1.2
VLAN10	↑	10Gbase-T <full-duplex>	192.168.10.1
VLAN20	↑	10Gbase-T <full-duplex>	192.168.20.1

## 2. Installation de Provisioning

- On va créer un template CT : provisioning

The screenshot shows the Proxmox Virtual Environment 7.2-3 interface. The 'Server View' is selected, and the 'pve' node is chosen. A new container is being created, named '101 (provisioning)'. The 'Container 101 (provisioning) on node 'pve'' page is displayed, showing the 'Summary' tab. The 'Memory' is set to 4.00 G. Buttons for 'Add', 'Edit', 'Remove', and 'Vol' are visible.

- On va ajouter une règle dans le firewall Pfsense pour autoriser le ssh de provisioning.
- Sur Provisioning on va y installer :

- Packer :

```
root@provisioning:~# packer -v
1.9.4
root@provisioning:~#
```

- Terraform

```
root@provisioning:~# terraform -version
Terraform v1.6.4
on linux_amd64
root@provisioning:~#
```

- Ansible :

```
root@provisioning:~# ansible-galaxy --version
ansible-galaxy [core 2.12.6]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible-galaxy
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.1.2
  libyaml = True
root@provisioning:~#
```

```
root@provisioning:~# ansible --version
ansible [core 2.12.6]
  config file = None
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.10/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/local/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0]
  jinja version = 3.1.2
  libyaml = True
root@provisioning:~#
```



### 3. Installation de Packer

- On télécharge les ISO des serveurs Windows sur le Proxmox

Name	Date	Format	Size
Autosart2022_windows2022_cloud64.iso	2023-11-29 15:23:42	iso	393.22 MB
Autosart2022_windows2022_cloud64.iso	2023-11-29 14:37:12	iso	393.22 MB
ubuntu-22.04-amd64-netinst.iso	2023-11-29 15:24:23	iso	407.90 MB
ubuntu-22.04-amd64-netinst.iso	2023-11-29 15:29:33	iso	787.48 MB
ubuntu-22.04.7-desktop-amd64.iso	2023-11-29 14:54:08	iso	64.25 MB
ubuntu-22.04.7-desktop-amd64.iso	2023-12-01 15:58:25	iso	1.70 GB
ubuntu-22.04.7-desktop-amd64.iso	2023-11-29 13:58:38	iso	627.82 MB
windows_server_2022_1439310_english_x64.iso	2023-11-29 13:59:58	iso	5.30 GB
windows_server_2022_1439310_english_x64.iso	2023-11-29 14:58:08	iso	6.97 GB

- On va créer un user dédiés à Packer depuis le terminal du pve :

```
pveum useradd infra_as_code@pve
pveum passwd infra_as_code@pve
```

- En lui ajoutant les rôles nécessaires, on pourra le vérifier depuis l'interface web de Proxmox.

sé | <https://10.202.0.122:8006/#v1:0:18:4:11:=contentIso:::20:14>

User name ↑	Realm ↑	Enabled	Expire	Name	TFA	Comment
infra_as_code	pve	Yes	never		No	
root	pam	Yes	never		No	

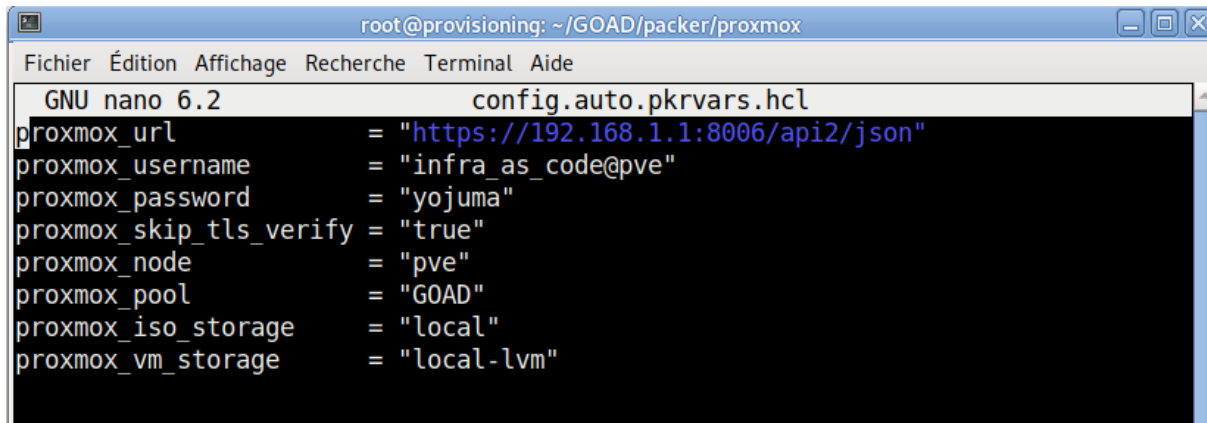
- On va se connecter a Provisioning depuis le pve on va SSH provisioning avec sa clé  
ssh : root@pve:/home# ssh -i provisioning root@192.168.1.3

```
root@pve:/home# ssh -i provisioning root@192.168.1.3
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.30-2-pve x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Thu Dec  7 10:50:50 2023 from 192.168.1.1
root@provisioning:~#
```

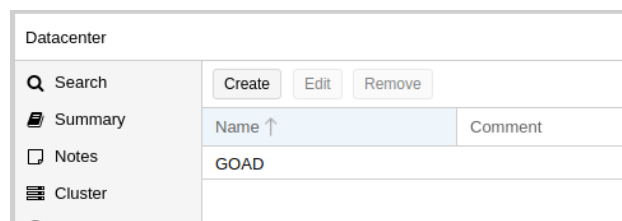
- Après avoir git clone le dossier GOAD sur provisioning. On va configurer les fichiers Packer pour l'installation du GOAD :
  - 1er fichiers a changer c'est : config.auto.pkrvars.hcl



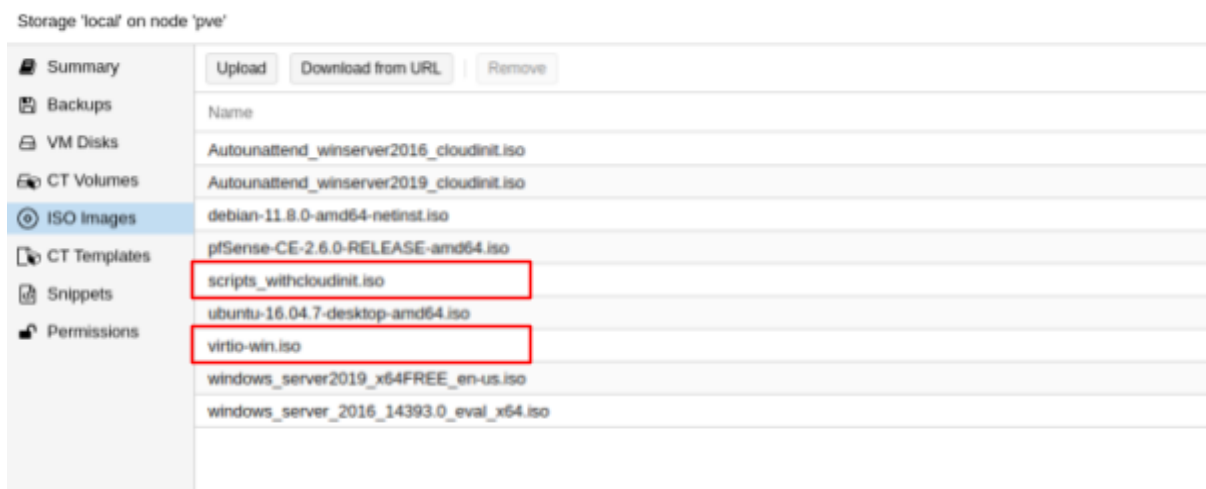
```

root@provisioning: ~/GOAD/packer/proxmox
Fichier Édition Affichage Recherche Terminal Aide
GNU nano 6.2 config.auto.pkrvars.hcl
proxmox_url      = "https://192.168.1.1:8006/api2/json"
proxmox_username = "infra_as_code@pve"
proxmox_password = "yojuma"
proxmox_skip_tls_verify = "true"
proxmox_node     = "pve"
proxmox_pool     = "GOAD"
proxmox_iso_storage = "local"
proxmox_vm_storage = "local-lvm"
  
```

- On y renseigner l'IP de proxmox pour la vm Provisioning, l'user et son mdp, le node, le pool "GOAD" créer comme ceci :



- Maintenant on va lancer le script pour créer un fichier iso : ./build\_proxmox\_iso.sh  
Ce script va permettre de créer 2 ISO.



- Puis on lance les script pour créer les serveur Windows :  
 packer init .  
 packer validate -var-file=windows\_server2019\_proxmox\_cloudinit.pkvars.hcl .  
 packer build -var-file=windows\_server2019\_proxmox\_cloudinit.pkvars.hcl .

```

root@provisioning:~/GOAD/packer/proxmox# packer init .
root@provisioning:~/GOAD/packer/proxmox# packer validate -var-file=windows_server2019_proxmox_cloudinit.pkvars.hcl .
The configuration is valid.
root@provisioning:~/GOAD/packer/proxmox# packer build -var-file=windows_server2019_proxmox_cloudinit.pkvars.hcl .
proxmox-iso.windows: output will be in this color.

==> proxmox-iso.windows: Retrieving additional ISO
==> proxmox-iso.windows: Trying ./iso/Autounattend_winserver2019_cloudinit.iso
==> proxmox-iso.windows: Trying ./iso/Autounattend_winserver2019_cloudinit.iso?checksum=sha256%3A8cab834d77cbbae18a3d912473cdadcf98aaa4fa205e8b7bc8621c956cc40c0
6cc40c0
==> proxmox-iso.windows: ./iso/Autounattend_winserver2019_cloudinit.iso?checksum=sha256%3A8cab834d77cbbae18a3d912473cdadcf98aaa4fa205e8b7bc8621c956cc40c0
=> /root/GOAD/packer/proxmox/iso/Autounattend_winserver2019_cloudinit.iso
proxmox-iso.windows: Uploaded ISO to local:iso/Autounattend_winserver2019_cloudinit.iso
==> proxmox-iso.windows: Creating VM
==> proxmox-iso.windows: No VM ID given, getting next free from Proxmox
==> proxmox-iso.windows: Starting VM
==> proxmox-iso.windows: Waiting for WinRM to become available...

```

```

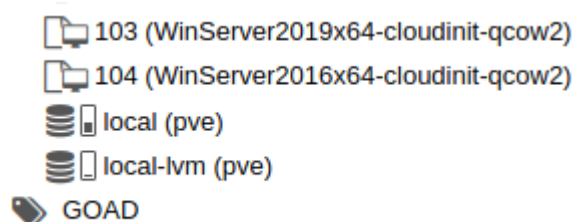
==> proxmox-iso.windows: Provisioning with Powershell...
==> proxmox-iso.windows: Provisioning with powershell script: ./scripts/sysprep/cloudbase-init-p2.ps1
proxmox-iso.windows: Show cloudinit service
proxmox-iso.windows:
proxmox-iso.windows: Status   Name                DisplayName
proxmox-iso.windows: -----
proxmox-iso.windows: Stopped cloudbase-init cloudbase-init
proxmox-iso.windows: Move config files to location
proxmox-iso.windows: Disable cloudbaseinit at start
==> proxmox-iso.windows: Stopping VM
==> proxmox-iso.windows: Converting VM to template
==> proxmox-iso.windows: Adding a cloud-init cdrom in storage pool local
Build 'proxmox-iso.windows' finished after 8 minutes 27 seconds.

==> Wait completed after 8 minutes 27 seconds

==> Builds finished. The artifacts of successful builds are:
--> proxmox-iso.windows: A template was created: 104
root@provisioning:~/GOAD/packer/proxmox# cd /root/GOAD/ad/GOAD/providers/proxmox/terraform
root@provisioning:~/GOAD/ad/GOAD/providers/proxmox/terraform# cp variables.template variables.tf

```

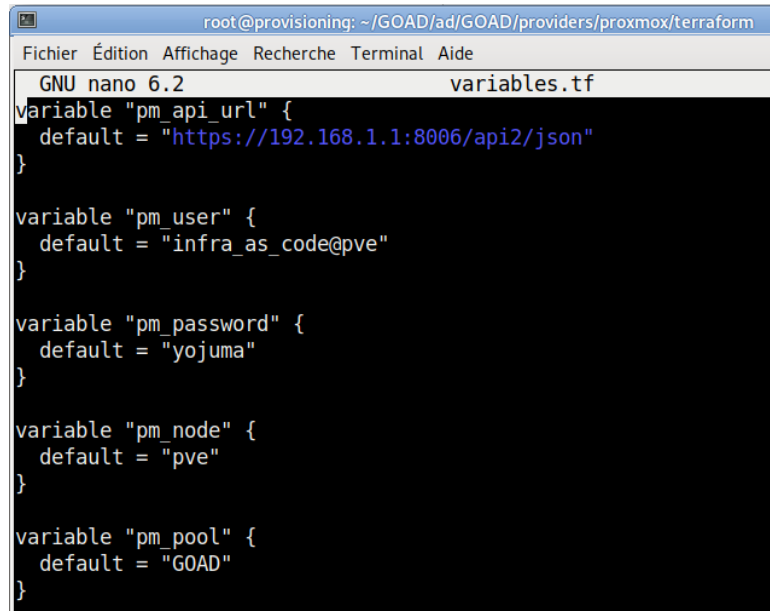
- Résultat :



## 4. Installation de Terraform pour installer les machines GOAD

- On va modifier le fichier : variables.tf

Dans : /root/GOAD/ad/GOAD/providers/proxmox/terraform



```

root@provisioning: ~/GOAD/ad/GOAD/providers/proxmox/terraform
Fichier  Édition  Affichage  Recherche  Terminal  Aide
GNU nano 6.2                                variables.tf
variable "pm_api_url" {
  default = "https://192.168.1.1:8006/api2/json"
}

variable "pm_user" {
  default = "infra_as_code@pve"
}

variable "pm_password" {
  default = "yojuma"
}

variable "pm_node" {
  default = "pve"
}

variable "pm_pool" {
  default = "GOAD"
}

```

- Donc pour lancer l'installation des VM on fait :

terraform init

terraform plan -out goad.plan

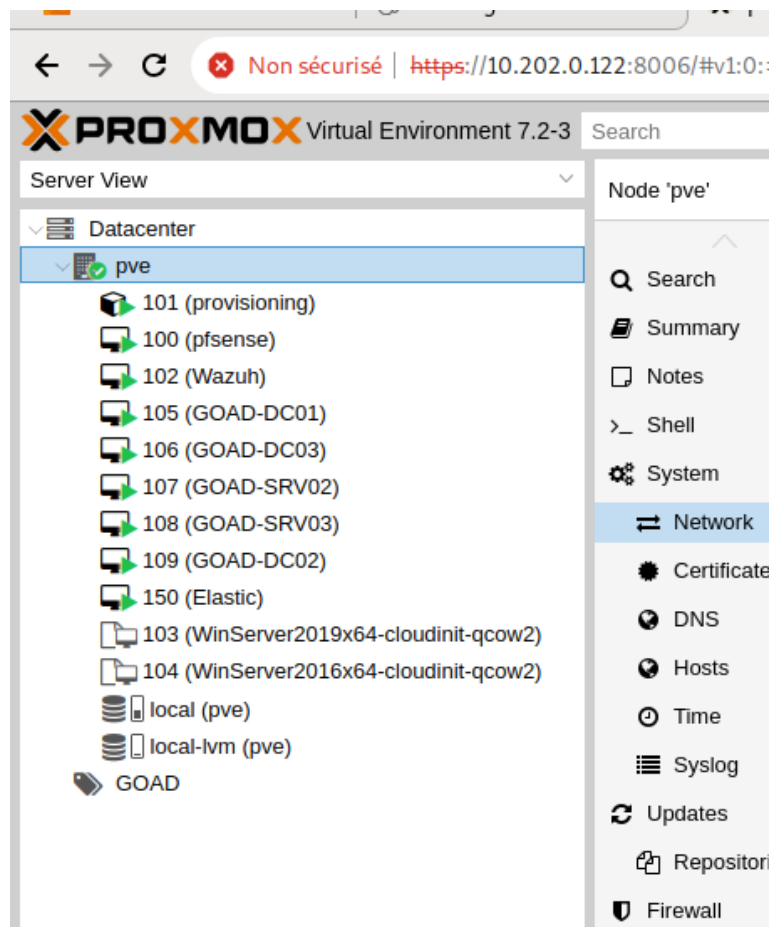
terraform apply "goad.plan"

```

proxmox_vm_qemu.dc01: Still creating... [1m0s elapsed]
proxmox_vm_qemu.dc02: Creation complete after 1m3s [id=pve/qemu/107]
proxmox_vm_qemu.srv02: Still creating... [1m10s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m10s elapsed]
proxmox_vm_qemu.dc01: Still creating... [1m10s elapsed]
proxmox_vm_qemu.dc01: Creation complete after 1m17s [id=pve/qemu/106]
proxmox_vm_qemu.srv02: Still creating... [1m20s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m20s elapsed]
proxmox_vm_qemu.srv02: Still creating... [1m30s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m30s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m40s elapsed]
proxmox_vm_qemu.srv02: Still creating... [1m40s elapsed]
proxmox_vm_qemu.srv02: Still creating... [1m50s elapsed]
proxmox_vm_qemu.srv03: Still creating... [1m50s elapsed]
proxmox_vm_qemu.srv03: Creation complete after 1m57s [id=pve/qemu/109]
proxmox_vm_qemu.srv02: Still creating... [2m0s elapsed]
proxmox_vm_qemu.srv02: Still creating... [2m10s elapsed]
proxmox_vm_qemu.srv02: Still creating... [2m20s elapsed]
proxmox_vm_qemu.srv02: Still creating... [2m30s elapsed]
proxmox_vm_qemu.srv02: Still creating... [2m40s elapsed]
proxmox_vm_qemu.srv02: Still creating... [2m50s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m0s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m10s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m20s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m30s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m40s elapsed]
proxmox_vm_qemu.srv02: Still creating... [3m50s elapsed]
proxmox_vm_qemu.srv02: Still creating... [4m0s elapsed]
proxmox_vm_qemu.srv02: Still creating... [4m10s elapsed]

```

- On peut vérifier sur Proxmox maintenant :



## 5. Installation de Ansible pour GOAD

- On va modifier le fichier "inventory", on y changera l'IP du DNS et le nom des adaptateurs réseaux des VM qui était pas les bon dans le fichier, voilà le résultat des changements :

```
root@provisioning: ~/GOAD/ad/GOAD/providers/proxmox
Fichier  Édition  Affichage  Recherche  Terminal  Aide
GNU nano 6.2                                inventory
elk ansible_host=192.168.10.50 ansible_connection=ssh

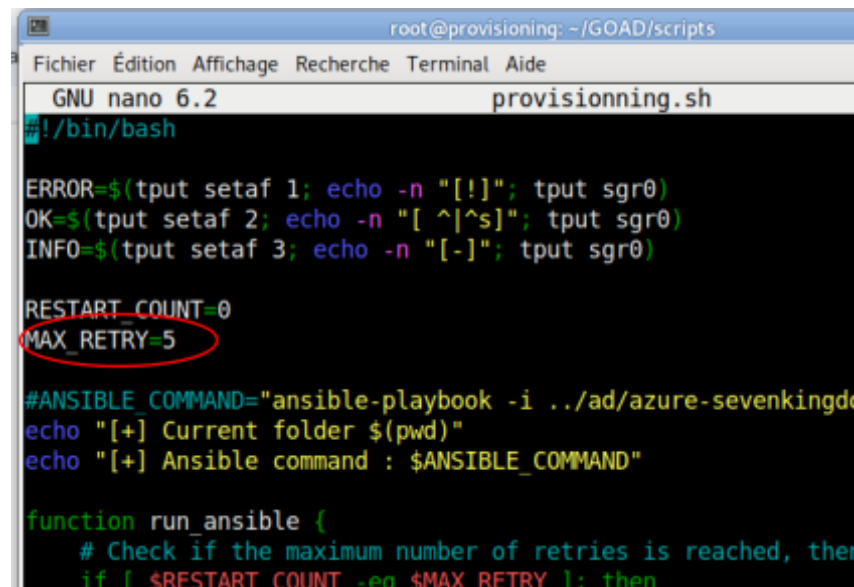
[all:vars]
; domain_name : folder inside ad/
domain_name=GOAD

force_dns_server=yes
dns_server=8.8.8.8

two_adapters=no
; adapter created by vagrant and virtualbox (comment if you use vmware)
nat_adapter=Ethernet Instance 0 2
domain_adapter=Ethernet Instance 0 2

; adapter created by vagrant and vmware (uncomment if you use vmware)
; nat_adapter=Ethernet0
; domain_adapter=Ethernet1
```

- Avant de lancer Ansible sur les VM, le script "provisioning.sh" qui permet de lancer tous les scripts permettant l'installation, au départ il essaie 3 fois d'installer Ansible sur les vm s'il y a une erreur, avant de s'arrêter.  
Moi je l'ai modifié pour avoir plus d'essai avant de réussir l'installation car j'avais une erreur qui persistait, et en ajoutant un nombre d'essai plus élevé j'ai réussi :



```

root@provisioning: ~/GOAD/scripts
GNU nano 6.2 provisioning.sh
#!/bin/bash

ERROR=$(tput setaf 1; echo -n "[!]"; tput sgr0)
OK=$(tput setaf 2; echo -n "[ ^|^s]"; tput sgr0)
INFO=$(tput setaf 3; echo -n "[-]"; tput sgr0)

RESTART_COUNT=0
MAX_RETRY=5

#ANSIBLE_COMMAND="ansible-playbook -i ../ad/azure-sevenkingdo
echo "[+] Current folder $(pwd)"
echo "[+] Ansible command : $ANSIBLE_COMMAND"

function run_ansible {
    # Check if the maximum number of retries is reached, then
    if [ $RESTART_COUNT -eq $MAX_RETRY ]; then

```

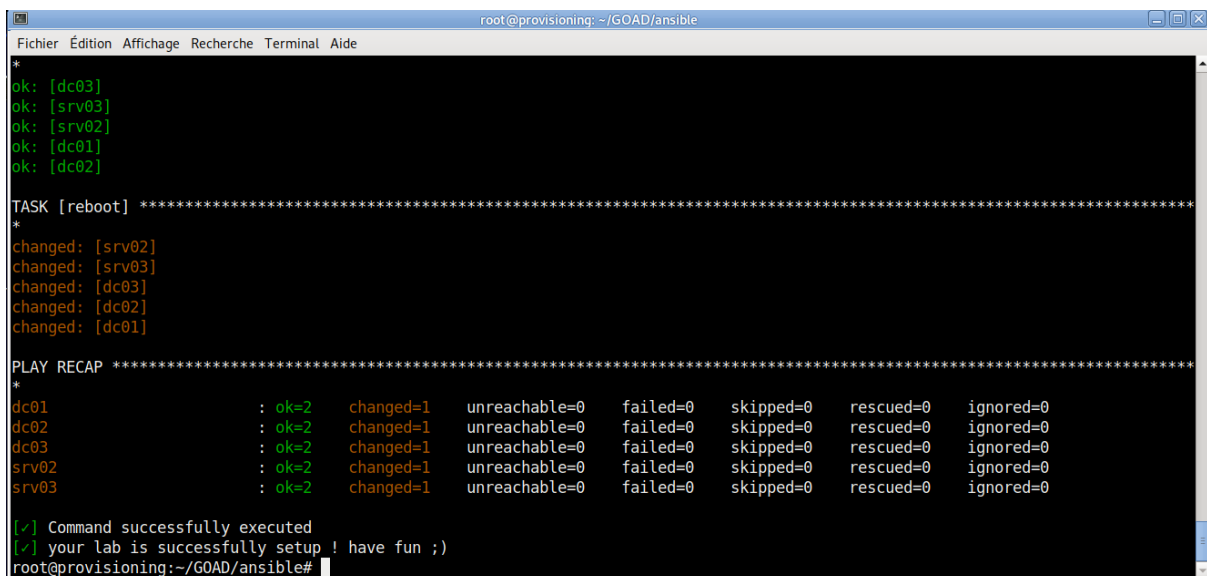
- Donc on lance le script :

```

cd /root/GOAD/ansible
export ANSIBLE_COMMAND="ansible-playbook -i ../ad/GOAD/data/inventory -i
../ad/GOAD/providers/proxmox/inventory"
../scripts/provisionning.sh

```

- Et à la fin de l'installation on a ce résultat :



```

root@provisioning: ~/GOAD/ansible
Fichier Édition Affichage Recherche Terminal Aide
*
ok: [dc03]
ok: [srv03]
ok: [srv02]
ok: [dc01]
ok: [dc02]

TASK [reboot] *****
*
changed: [srv02]
changed: [srv03]
changed: [dc03]
changed: [dc02]
changed: [dc01]

PLAY RECAP *****
*
dc01      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc02      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
dc03      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv02     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv03     : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[✓] Command successfully executed
[✓] your lab is successfully setup ! have fun ;)
root@provisioning:~/GOAD/ansible#

```

- L'installation de GOAD est terminée.

Réalisé par Mathéo Balazuc.