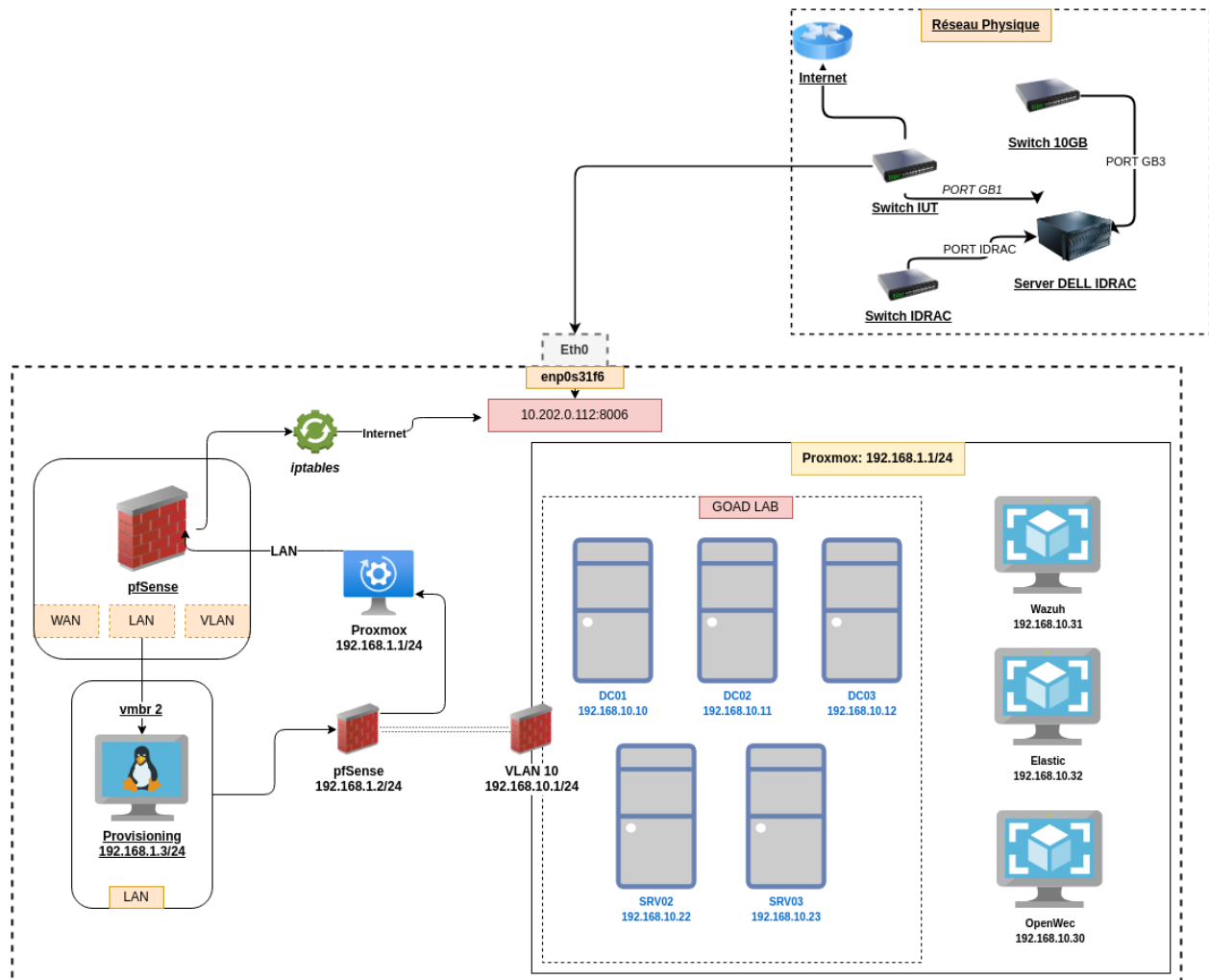


GOAD PROXMOX

Schéma Réseaux



Explication:

D'après le schéma on peut voir comment notre infrastructure est déployée sur proxmox. On a utilisé pfSense comme notre firewall pour que notre infrastructure soit déployée correctement et sur un réseau local. On a créé 3 bridges sur notre proxmox:

vmbr1	Linux Bridge	Yes	Yes	No	10.0.0.1/30	WAN
vmbr2	Linux Bridge	Yes	Yes	No	192.168.1.1/24	LAN
vmbr3	Linux Bridge	Yes	Yes	Yes		

Le **vmbr1** ça nous sert comme WAN, **vmbr2** on l' utilise comme LAN (pour la machine provisioning qu'on expliquera à la suite) et aussi **vmbr3** pour les vlans.

vlan10	Linux VLAN	Yes	Yes	No	192.168.10.100/24	VLAN10 (192.168.10.1/24)
vlan20	Linux VLAN	Yes	Yes	No		VLAN10 (192.168.20.1/24)

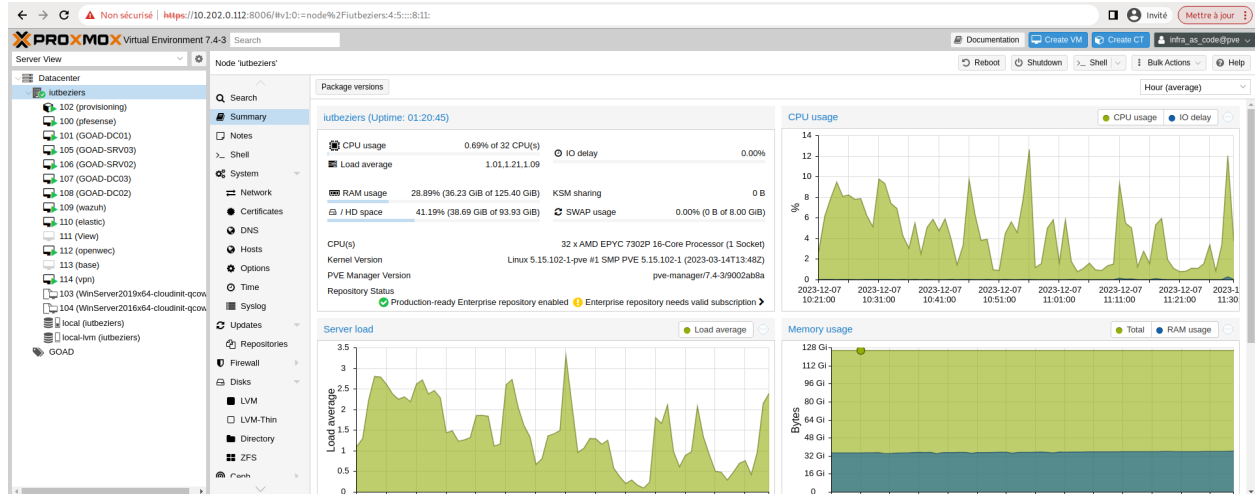
Ensuite on a créé Linux vlans et on les a associé a notre bridge vbr3.

On a déployé une machine virtuelle ubuntu qu'on l'appelé provisioning car c'est à travers de cette machine qu'on va déployer GOAD sur proxmox, cette machine est en réseau LAN ainsi que notre environnement proxmox sera en réseau LAN. On a préféré déployer toutes nos

Lucas, Antoine, Elouan

machines virtuelles en VLAN 10, on peut voir aussi que le GOAD LAB est en VLAN 10 ainsi que le reste d'autres machines.

Deployment GOAD



Configuration iptables sur notre pve:

```
root@iutbeziers:~# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
ACCEPT     icmp -- anywhere              anywhere
ACCEPT     tcp  -- anywhere              anywhere      tcp dpt:ssh
ACCEPT     tcp  -- anywhere              anywhere      tcp dpt:8006
DNAT       all  -- anywhere              anywhere      to:10.0.0.2

Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
SNAT       all  -- 10.0.0.0/30          anywhere      to:10.202.0.112
```

User & Token:

pveum user token add infra_as_code@pve packer -expire 0 -privsep 0 -comment "token"

```
root@iutbeziers:~# pveum user token add infra_as_code@pve packer -expire 0 -privsep 0 -comment "token"
```

key	value
full-tokenid	infra_as_code@pve!packer
info	{"comment":"token","expire":"0","privsep":"0"}
value	85ac8f5a-8e5e-4d8d-a5d2-92d60dc10de8

```
root@iutbeziers:~# pveum user list
```

userid	comment	email	enable	expire	firstname	groups	keys	lastname	realm-type	tokens
infra_as_code@pve			1	0					pve	
root@pam		goad@proxmox.com	1	0					pam	

Avant le déploiement c'est important de créer un user dans notre cas c'est *"infra_as_code"* et on a créé un token pour le même.

VM Provisioning:

Dans notre provisioning on a cloné le dossier GOAD. Ensuite on a effectué quelques modifications pour l'installation sur proxmox:

/GOAD/ad/GOAD/providers/proxmox/terraform/variables.tf

```
root@provisioning:~/GOAD/ad/GOAD/providers/proxmox/terraform# cat 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 = "root123"
}

variable "pm_node" {
  default = "iutbeziers"
}

variable "pm_pool" {
  default = "GOAD"
}

variable "pm_full_clone" {
  default = true
}
```

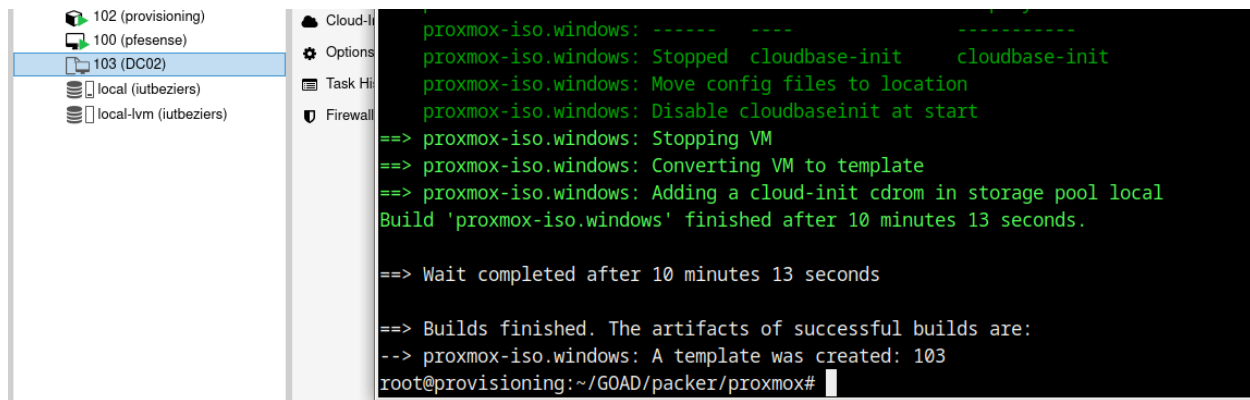
Lancement du script packer:

Avant de lancer le script packer on a modifié le fichier de configuration:

cd /GOAD/packer/proxmox/config.auto.pkrvars.hcl

```
root@provisioning:~/GOAD/packer/proxmox# cat config.auto.pkrvars.hcl
proxmox_url      = "https://192.168.1.1:8006/api2/json"
proxmox_username = "infra_as_code@pve"
proxmox_password = "root123"
proxmox_skip_tls_verify = "true"
proxmox_node     = "iutbeziers"
proxmox_pool     = "GOAD"
proxmox_iso_storage = "local"
proxmox_vm_storage = "local-lvm"
root@provisioning:~/GOAD/packer/proxmox#
```

Ensuite on lance le script:



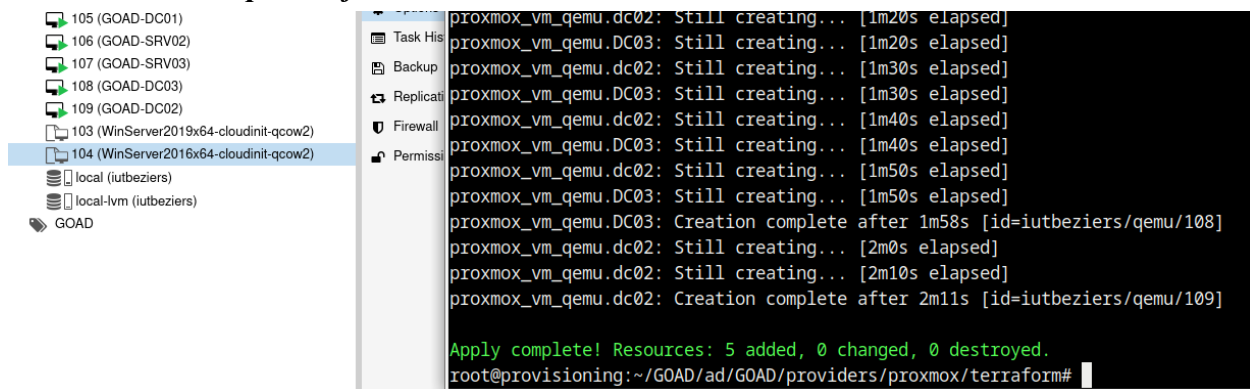
```
proxmox-iso.windows: -----
proxmox-iso.windows: Stopped cloudbase-init cloudbase-init
proxmox-iso.windows: Move config files to location
proxmox-iso.windows: Disable cloudbaseinit at start
==> proxmox-iso.windows: Stopping VM
==> proxmox-iso.windows: Converting VM to template
==> proxmox-iso.windows: Adding a cloud-init cdrom in storage pool local
Build 'proxmox-iso.windows' finished after 10 minutes 13 seconds.

==> Wait completed after 10 minutes 13 seconds

==> Builds finished. The artifacts of successful builds are:
--> proxmox-iso.windows: A template was created: 103
root@provisioning:~/GOAD/packer/proxmox#
```

Il a créé à la fin les templates windows pour déployer les machines sur terraform.

Lancement du script terraform:



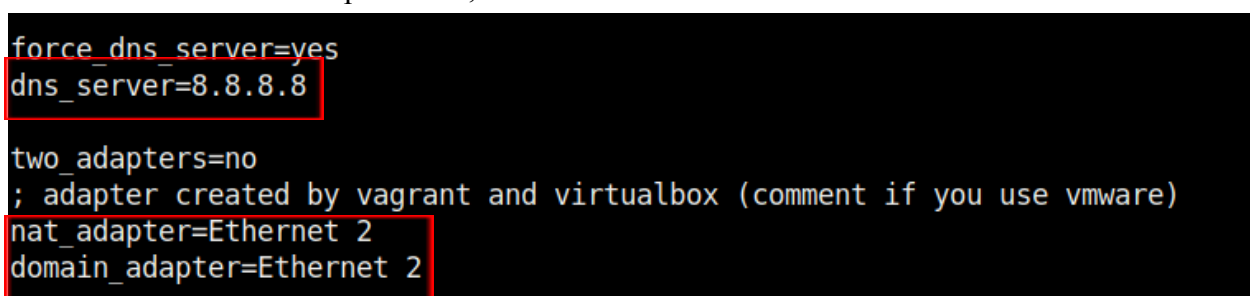
```
proxmox_vm_qemu.dc02: Still creating... [1m20s elapsed]
proxmox_vm_qemu.dc03: Still creating... [1m20s elapsed]
proxmox_vm_qemu.dc02: Still creating... [1m30s elapsed]
proxmox_vm_qemu.dc03: Still creating... [1m30s elapsed]
proxmox_vm_qemu.dc02: Still creating... [1m40s elapsed]
proxmox_vm_qemu.dc03: Still creating... [1m40s elapsed]
proxmox_vm_qemu.dc02: Still creating... [1m50s elapsed]
proxmox_vm_qemu.dc03: Still creating... [1m50s elapsed]
proxmox_vm_qemu.dc03: Creation complete after 1m58s [id=iutbeziers/qemu/108]
proxmox_vm_qemu.dc02: Still creating... [2m0s elapsed]
proxmox_vm_qemu.dc02: Still creating... [2m10s elapsed]
proxmox_vm_qemu.dc02: Creation complete after 2m11s [id=iutbeziers/qemu/109]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
root@provisioning:~/GOAD/ad/GOAD/providers/proxmox/terraform#
```

Le script terraform déploie les 5 machines GOAD par les templates créés grâce au packer.

Lancement du script Ansible:

Avant le lancement du script ansible, on a fait des modifications dans l'inventaire:



```
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 2
domain_adapter=Ethernet 2
```

Ethernet 2 car nos machines windows sont dans la même interface et DNS 8.8.8.8 pour pouvoir avoir l'accès à l'internet sur les vms windows.

Lucas, Antoine, Elouan

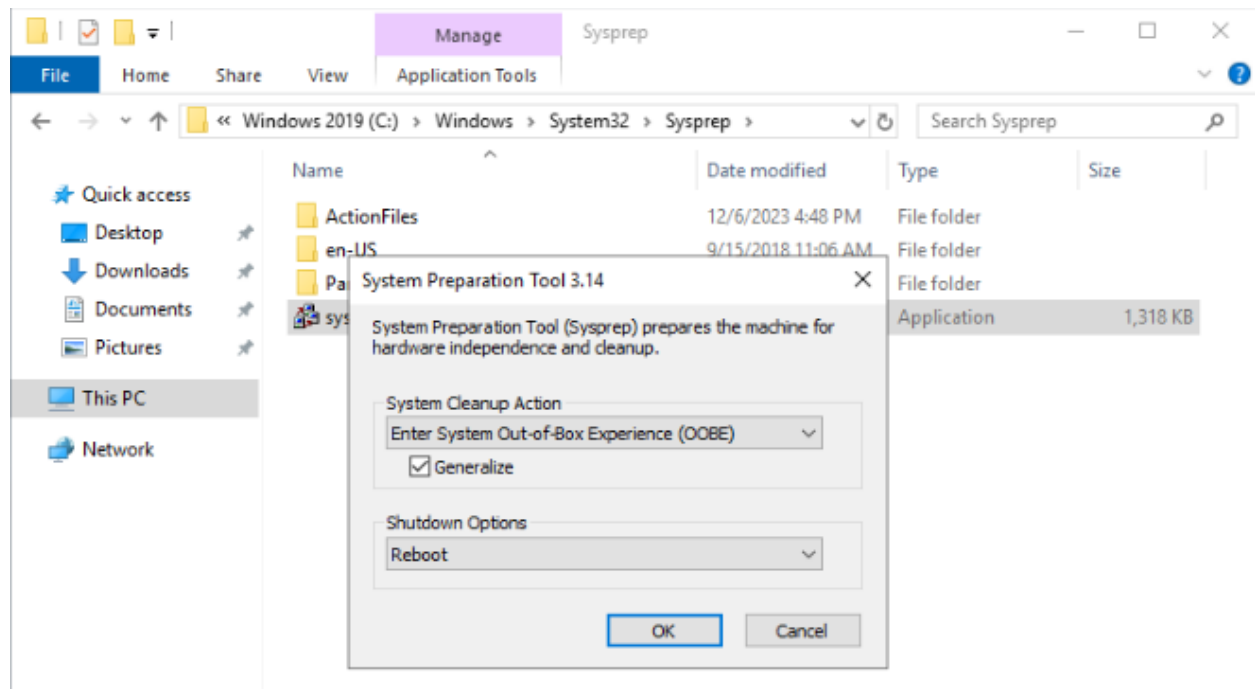
Installation de requirements:

```
root@provisioning:~/G0AD/ansible# ansible-galaxy install -r requirements.yml
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/ansible-windows-1.11.0.tar.gz to /root/.ansi
i/ansible-windows-1.11.0-8akydg85
Installing 'ansible.windows:1.11.0' to '/root/.ansible/collections/ansible_collections/ansible/windows'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/community-windows-1.11.0.tar.gz to /root/.an
q5i/community-windows-1.11.0-u2i_3ukn
ansible.windows:1.11.0 was installed successfully
Installing 'community.windows:1.11.0' to '/root/.ansible/collections/ansible_collections/community/windows'
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/chocolatey-chocolatey-1.5.1.tar.gz to /root/
9deq5i/chocolatey-chocolatey-1.5.1-3302qnb
community.windows:1.11.0 was installed successfully
Downloading https://galaxy.ansible.com/api/v3/plugin/ansible/content/published/collections/artifacts/community-general-8.0.2.tar.gz to /root/.ans
5i/community-general-8.0.2-te4nch0m
Installing 'chocolatey.chocolatey:1.5.1' to '/root/.ansible/collections/ansible_collections/chocolatey/chocolatey'
chocolatey.chocolatey:1.5.1 was installed successfully
Installing 'community.general:8.0.2' to '/root/.ansible/collections/ansible_collections/community/general'
community.general:8.0.2 was installed successfully
root@provisioning:~/G0AD/ansible#
```

Lors de l'exécution du script ansible, on a rencontré des difficultés car la machine DC02 ne prenait pas de domain child, on ne pouvait pas avancer à la tâche suivante.

```
TASK [member_server : Add member server] *****
[fatal: [srv02]: FAILED! => ('changed': true, 'msg': 'failed to join domain: Computer 'castelblack' failed to join domain 'north.sevenkingdoms.local' from its current workgroup 'WORKGROUP' wi
th following error message: The specified domain either does not exist or could not be contacted.', 'reboot_required': false)
ok: [srv03]
PLAY [play workstations AD configuration] *****
```

Du coup pour résoudre ce problème on a fait un sysprep comme ça il nous a mis la machine dc02 en défaut et on a relancé le script et ça a marché.



Lucas, Antoine, Elouan

On peut voir qu'après elle a pris le domaine child.

Processor: AMD EPYC 7302P 16-Core Processor 2.99 GHz

Installed memory (RAM): 3.02 GB

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name: winterfell [Change settings](#)

Full computer name: winterfell.north.sevenkingdoms.local

Computer description:

Domain: north.sevenkingdoms.local

OPEN VPN

On a utilisé OPENVPN comme VPN, notre firewall pfSense intègre déjà OPENVPN pour le faire fonctionner on a fait:

- Création d'un nouvel utilisateur:

Users					
	Username	Full name	Status	Groups	Actions
<input type="checkbox"/>	admin	System Administrator	✓	admins	
<input type="checkbox"/>	vpn-mayfly		✓		
				Add	Delete

On a associé cet utilisateur au certificat qu'on a créé.

- Création de certificat

Certificate Authorities						
Name	Internal	Issuer	Certificates	Distinguished Name	In Use	Actions
VLAN10-OPENVPN	✓	self-signed	3	CN=vlan10-ca Valid From: Thu, 30 Nov 2023 08:41:00 +0000 Valid Until: Sun, 27 Nov 2033 08:41:00 +0000		

- Ajouter les regles de sortie dans notre firewall

Floating

WAN

LAN

VLAN10

VLAN20

OpenVPN

Rules (Drag to Change Order)

<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	✓	0/0 B	IPv4 *	*	*	VLAN10 net	*	*	none	allow vlan1 access	

Lucas, Antoine, Elouan

On a rajouté une règle d'entrée:

Firewall / Rules / WAN											
Floating WAN LAN VLAN10 VLAN20 OpenVPN											
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/709 KiB	*	Reserved Not assigned by IANA	*	*	*	*	*		Block bogon networks	
IN											
<input checked="" type="checkbox"/>	5/17.70 MiB	IPv4 UDP	*	*	WAN address	2137	*	none		goad-openvpn	
OUT											
<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	10.0.0.1	*	192.168.1.3	22 (SSH)	*	none		Allow SSH	
<input checked="" type="checkbox"/>	0/0 B	IPv4 TCP	10.0.0.1	*	LAN address	80 (HTTP)	*	none			

Test:

```
test@232-22:~/Téléchargements$ sudo openvpn pfSense-UDP4-2137-vpn-mayfly-config.ovpn
2023-12-08 16:31:02 OpenVPN 2.5.1 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on May 14 2021
2023-12-08 16:31:02 library versions: OpenSSL 1.1.1n 15 Mar 2022, LZ0 2.10
🔑 Enter Auth Username: vpn-mayfly
🔑 Enter Auth Password: *****
2023-12-08 16:31:15 TCP/UDP: Preserving recently used remote address: [AF_INET]10.202.0.112:2137
2023-12-08 16:31:15 UDPv4 link local: (not bound)
2023-12-08 16:31:15 UDPv4 link remote: [AF_INET]10.202.0.112:2137
2023-12-08 16:31:15 [vpn.goad.lab] Peer Connection Initiated with [AF_INET]10.202.0.112:2137
2023-12-08 16:31:16 TUN/TAP device tun0 opened
2023-12-08 16:31:16 net_iface_mtu_set: mtu 1500 for tun0
2023-12-08 16:31:16 net_iface_up: set tun0 up
2023-12-08 16:31:16 net_addr_ptp_v4_add: 10.10.10.6 peer 10.10.10.5 dev tun0
2023-12-08 16:31:16 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to prevent this
2023-12-08 16:31:16 Initialization Sequence Completed
```

Tun0:

On peut voir que la porte tun0 est ouverte (WAN) grâce à la configuration VPN.

```
tun0                UNKNOWN          10.10.10.6 peer 10.10.10.5/32 fe80::67b2:b154:84cb:a20b/64
```


Suricata IPS:

On a lancé un docker suricata sur le node Proxmox qui récupère toutes les alertes et les stocke sur un fichier json. On les a monté sur la machine elastic directement pour pouvoir les afficher, pour faire cela on a utilisé le partage nfs.

```
root@iutbeziers:~# docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS        NAMES
4f40aacb10c6   jasonish/suricata:latest "/docker-entrypoint..." 4 hours ago    Up 4 hours    ports        flamboyant_archimedes
```

```
GNU nano 5.4
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes          hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4           gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes     gss/krb5i(rw,sync,no_subtree_check)
#
/var/log/suricata/ 192.168.10.32(rw,sync,no_root_squash,subtree_check,insecure)
```

```
root@iutbeziers:~# showmount -e
Export list for iutbeziers:
/var/log/suricata 192.168.10.32
```

Ensuite sur la machine elastic:

```
root@elastic:/var/log/suricata# mount -t nfs -o nfsvers=3 192.168.10.100:/var/log/suricata/ /var/log/suricata/
root@elastic:/var/log/suricata# ls
eve.json fast.log stats.log suricata.log
root@elastic:/var/log/suricata#
```

Après on l'a monté sur le client, l'adresse du serveur c'est 192.168.10.100 et pas 10.202.0.112 car notre machine elastic elle est sur vlan10.

Python view:

On a développé un script python pour la visualisation des données dans le fichier log suricata eve.json. Pour cela on a opté pour l'utilisation de la bibliothèque pandas et json.

```
root@iutbeziers:~/suricata-resources# python3 recup.py
timestamp      flow_id  in_iface  event_type  src_ip  src_port  ... tcp.flags  tc  tcp.syn  tcp.fin  tcp.psh  tcp.ack  tcp.state
0  2023-12-08T11:44:56.041577+0100  9.332318e+14  vmbr0  dcerpc  10.202.0.186  51820.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
1  2023-12-08T11:44:56.977877+0100  1.879583e+15  vmbr0  tls  10.202.11.20  56474.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
2  2023-12-08T11:45:00.977795+0100  9.582242e+14  vmbr0  tls  10.202.11.20  56482.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
3  2023-12-08T11:45:00.981635+0100  4.084254e+14  vmbr0  tls  10.202.11.20  56488.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
4  2023-12-08T11:45:03.796289+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
5  2023-12-08T11:45:04.089338+0100  9.332318e+14  vmbr0  dcerpc  10.202.0.186  51820.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
6  2023-12-08T11:45:09.127674+0100  9.332318e+14  vmbr0  dcerpc  10.202.0.186  51820.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
7  2023-12-08T11:45:11.796721+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
8  2023-12-08T11:45:19.797108+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
9  2023-12-08T11:45:27.797477+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
10 2023-12-08T11:45:35.797841+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
11 2023-12-08T11:45:39.151120+0100  1.995970e+15  vmbr0  flow  fe80:0000:0000:0000:b27b:25ff:fe26:9907  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
12 2023-12-08T11:45:43.798219+0100  NaN  NaN  stats  NaN  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
13 2023-12-08T11:45:44.227592+0100  5.395379e+14  vmbr0  flow  10.202.255.254  67.0  ...  NaN  NaN  NaN  NaN  NaN  NaN
14 2023-12-08T11:45:44.230296+0100  1.669879e+15  vmbr0  flow  0000:0000:0000:0000:0000:0000:0000:0000  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
15 2023-12-08T11:45:44.837572+0100  9.215838e+14  vmbr0  flow  0000:0000:0000:0000:0000:0000:0000:0000  NaN  ...  NaN  NaN  NaN  NaN  NaN  NaN
```


Sous forme graphique:

