



# Université Paris Cité

## **Mise en place VPN IPsec, sous forme de TP.**

**Etudiant 1: Yohann Letellier 22317638**

**Etudiant 2: Ivan KRIVOKUCA 22306432**

**Etudiant 3: Ziad HASNI 51805965**

**Etudiant 4: Yacine Ahmed-Yahia**

**Professeur: Monsieur Khatoun**

**Année: 2024-2025**

**Master 2 Cybersécurité**

# *Sommaire.*

<b>Sommaire.....</b>	<b>2</b>
<b>I. Prérequis.....</b>	<b>3</b>
<b>II. Mise en place de l'environnement.....</b>	<b>4</b>
1) Créer deux réseaux privés hôtes.....	4
2) Installer une machine virtuelle Debian.....	5
3) Cloner la machine virtuelle Debian.....	9
4) Installer et configurer la 1ère machine PfSense.....	12
5) Installer et configurer la 2ème machine PfSense.....	21
6) Vérifier la bonne installation de l'environnement.....	22
<b>III. Mise en place du VPN IPsec.....</b>	<b>24</b>
1) Mise en place sur la 1ère passerelle PFsense.....	24
2) Mise en place sur la 2ème passerelle PFsense.....	26
3) Connexion des deux passerelles.....	26
4) Vérifier le bon fonctionnement.....	27

# ***1. Prérequis***

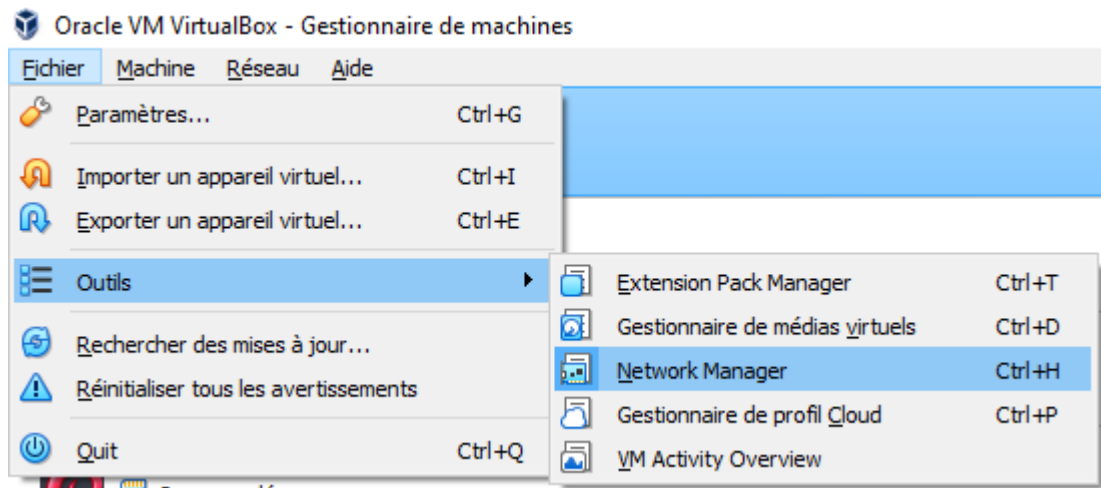
Les prérequis pour ce TP sont:

- Au minimum 10 Go de RAM et 85 Go de stockage.
- VirtualBox.
- Iso Debian 12.
- Iso PFSense.

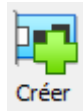
## 11. Mise en place de l'environnement

### 1) Créer deux réseaux privés hôtes

Aller dans Fichier -> Outils -> Network Manager.



Cliquer sur “Créer”.



Configurer les 2 réseaux comme suit:

Adapteur	Serveur DHCP
<input type="radio"/> Configurer la carte automatiquement	
<input checked="" type="radio"/> Configurer la carte manuellement	
Adresse IPv4 :	1.2.3.4
Masque réseau IPv4 :	255.255.255.0
Adresse IPv6 :	fe80::8d5f:6cb7:72f8:32a0
IPv6 Prefix Length:	64

Adapter    Serveur DHCP

☐ Configurer la carte automatiquement

☒ Configurer la carte manuellement

Adresse IPv4 : 7.8.9.1

Masque réseau IPv4 : 255.255.255.0

Adresse IPv6 : fe80::ecdb:c346:c336:f557

IPv6 Prefix Length: 64

Ceci donne deux réseaux privés, déconnectés l'un de l'autre:  
1.2.3.0/24 et 7.8.9.0/24

## 2) Installer une machine virtuelle Debian

Cliquer sur “Nouvelle”.



Donner pour nom “Site 1”. Donner le lien de l'ISO et le dossier où sera installée la VM. Sélectionner “Skip Unattended Installation”. Enfin, cliquer sur “Suivant”

Crée une machine virtuelle

### Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Nom : Site 1 ✓

Folder: E:\VM

ISO Image: E:\VM\debian-12.4.0-amd64-netinst.iso

Edition:

Type : Linux 64

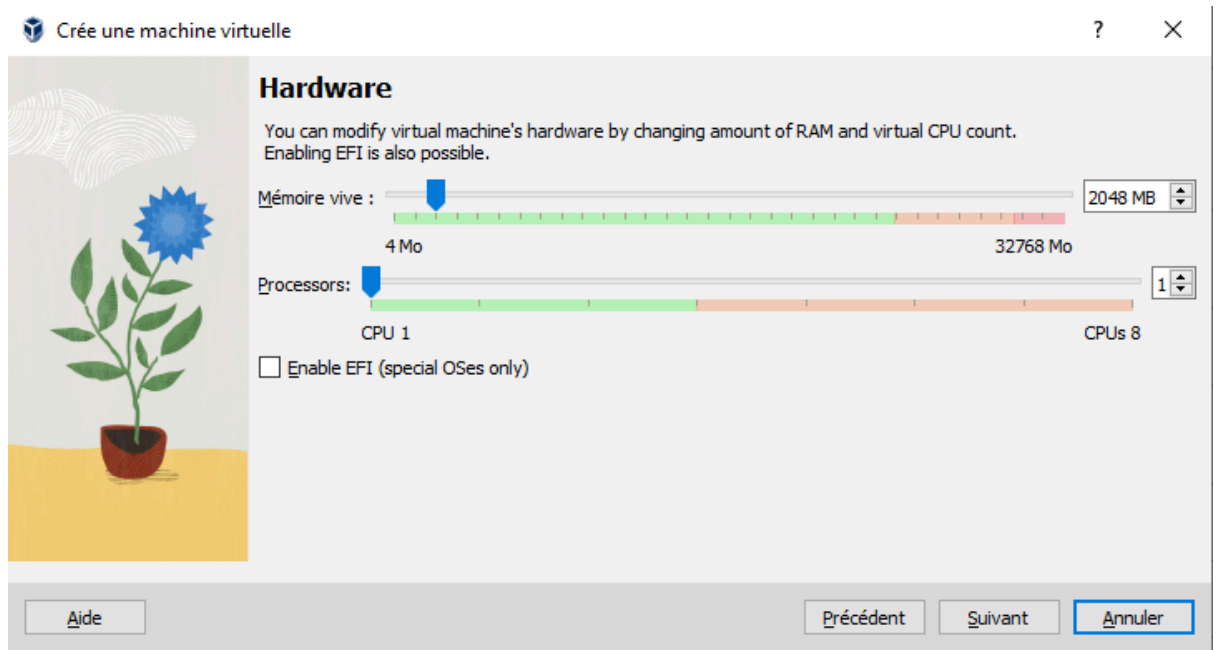
Version : Debian (64-bit)

☒ Skip Unattended Installation

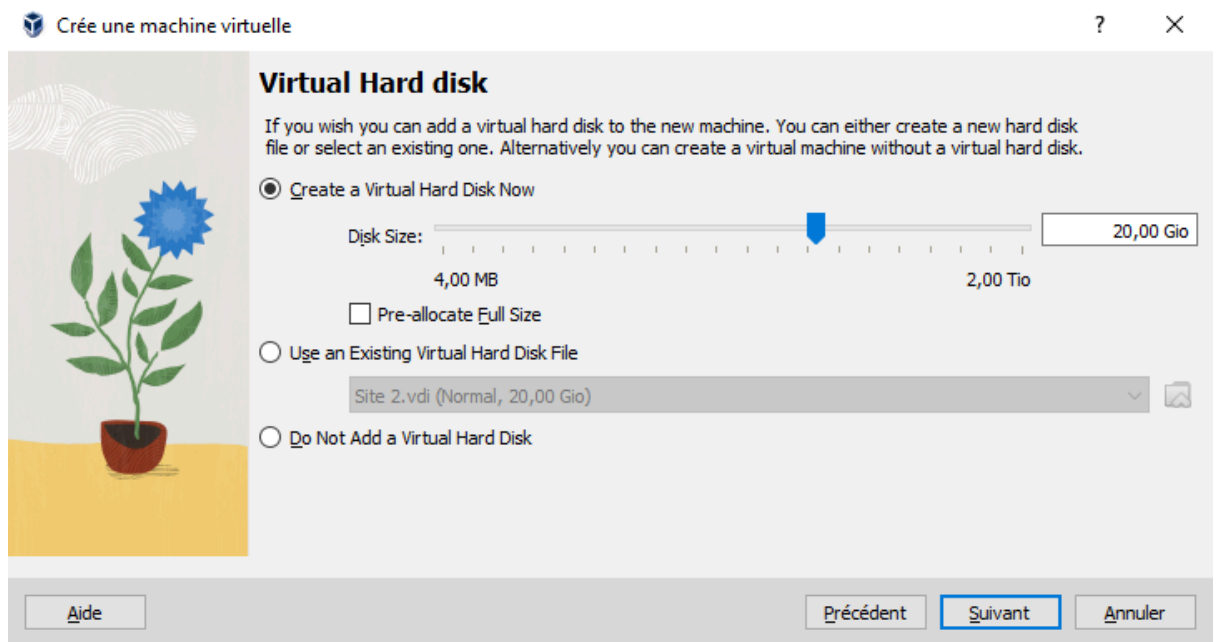
*You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.*

Aide    Mode expert    Précédent    **Suivant**    Annuler

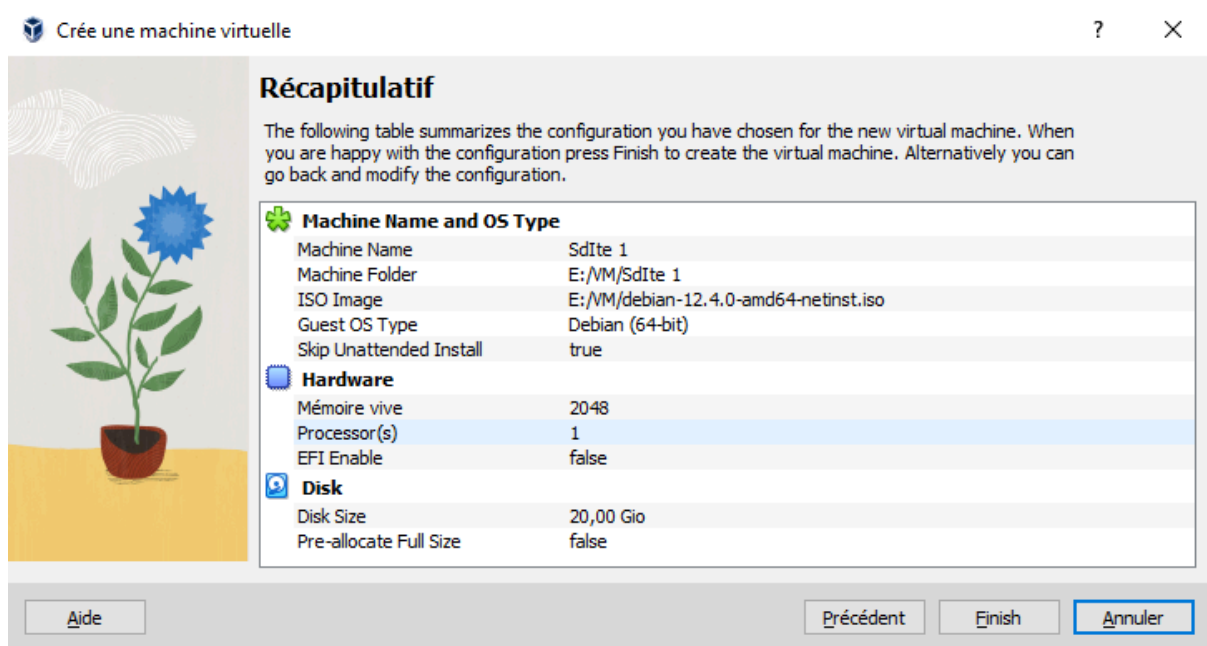
Sélectionner 1 CPU et 2048 MB de RAM, et cliquer sur suivant.



Spécifier 20 Go de taille de disque et cliquer sur “Suivant”.

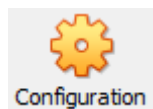


Cliquer ensuite sur “Finish”.

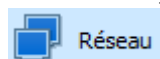


Enfin, se laisser guider dans l'installation pas à pas de Debian.

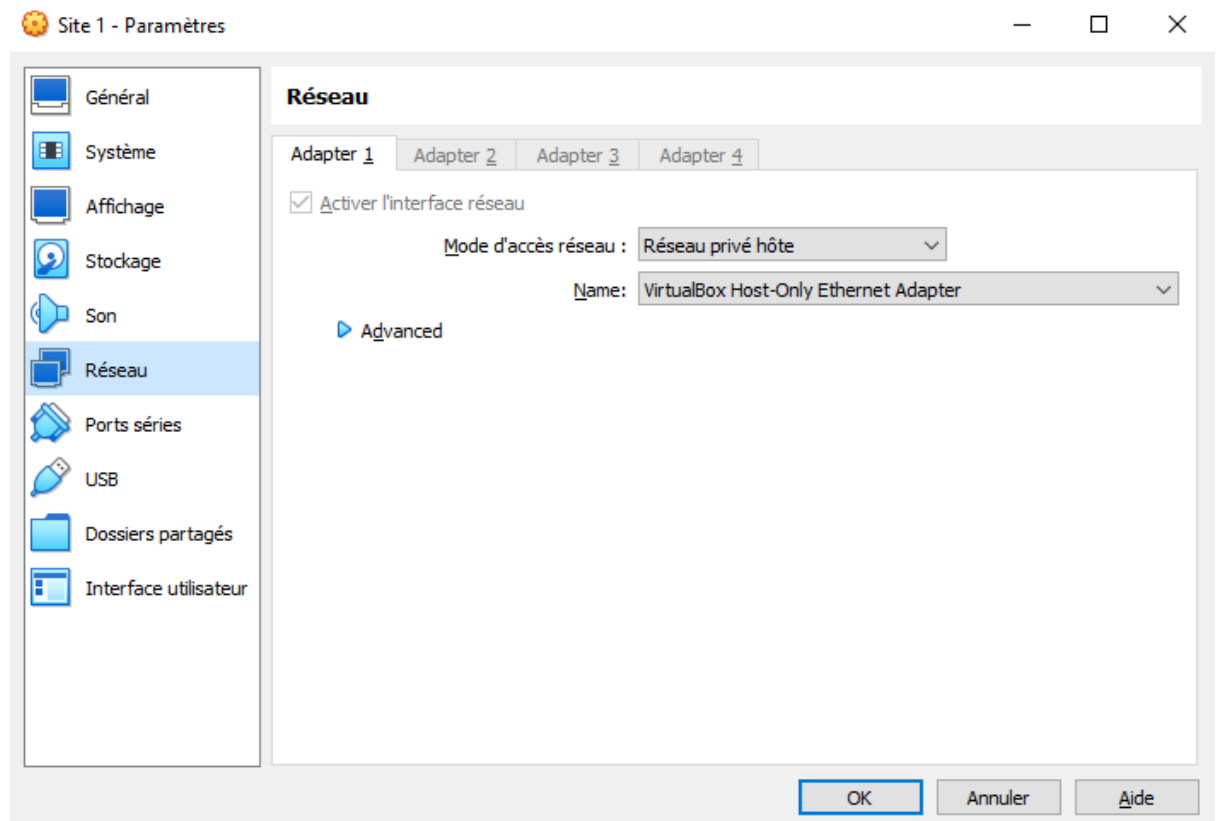
Une fois Debian installé, aller dans les paramètres de la VM.



Dans les paramètres, aller dans "Réseau".



Sélectionner comme mode d'accès "Réseau privé hôte" et comme réseau le 1er.



Allumer ensuite la VM. Configurer le réseau de la manière suivante:



Annuler

Filaire

Appliquer

Détails
Identité
IPv4
IPv6
Sécurité

Méthode IPv4

☐ Automatique (DHCP)
☐ Réseau local seulement
☒ Manuel
☐ Désactiver
☐ Partagée avec d'autres ordinateurs

Adresses

Adresse	Masque de réseau	Passerelle	
1.2.3.11	255.255.255.0	1.2.3.10	✕
			✕

DNS

Automatique ☒

Séparer les adresses IP avec des virgules

Routes

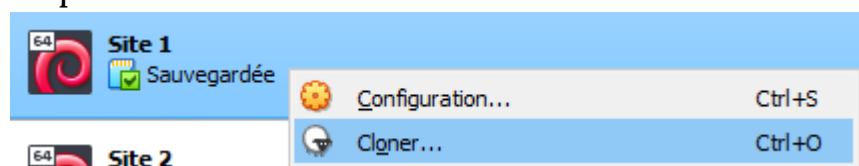
Automatique ☒

Adresse	Masque de réseau	Passerelle	Métrique	
				✕

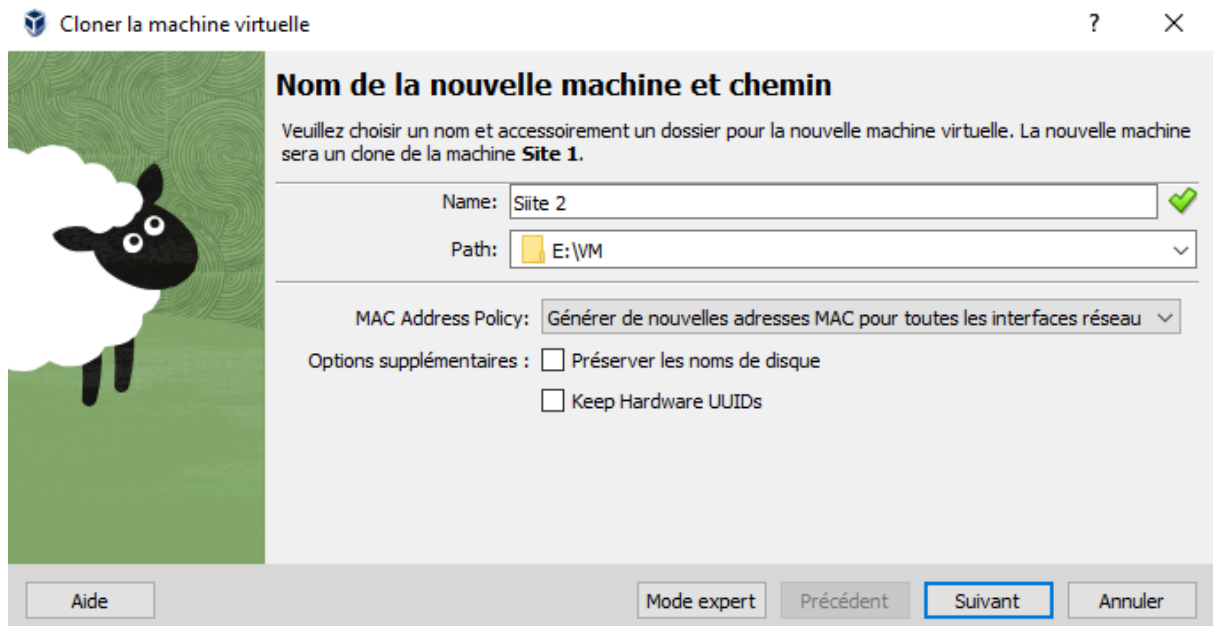
### 3) Cloner la machine virtuelle Debian

Pour faire le 2ème site ainsi que les 2 serveurs VPN, il faut cloner la machine, afin de ne pas avoir à la réinstaller.

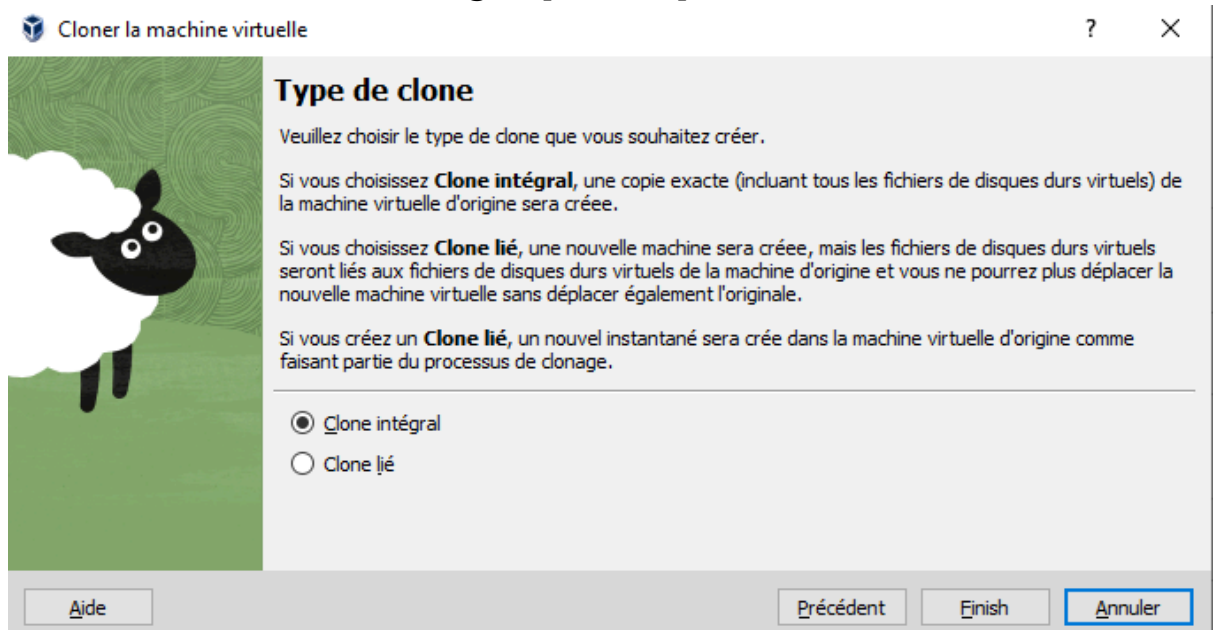
Pour le 2ème site, aller sur la VM Site 1, faire un clic droit et cliquer sur “cloner”.



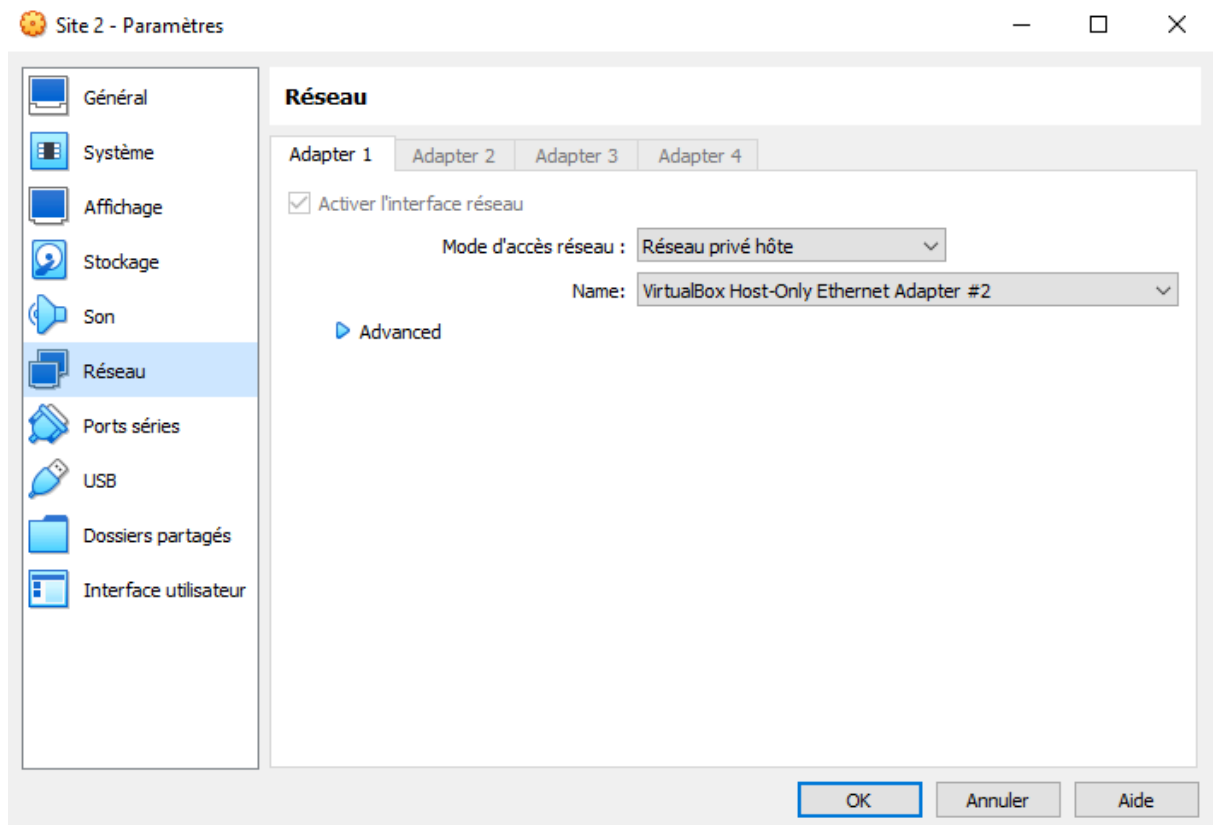
Nommer le clone “Site 2”, et sélectionner “Générer de nouvelles adresses MAC pour toutes les interfaces réseau”.



Sélectionner un Clone intégral puis cliquer sur Finish.



Aller dans les paramètres, réseau, et sélectionner “Réseau privé hôte” et cette fois le 2ème réseau.



Allumer la VM, et configurer le réseau comme suit:

Annuler

Filaire

Appliquer

Détails

Identité

IPv4

IPv6

Sécurité

Méthode IPv4

☐ Automatique (DHCP)
☒ Manuel
☐ Partagée avec d'autres ordinateurs

☐ Réseau local seulement
☐ Désactiver

Adresses

Adresse	Masque de réseau	Passerelle	
7.8.9.11	255.255.255.0	7.8.9.10	✕
			✕

DNS

Automatique ☒

Séparer les adresses IP avec des virgules

Routes

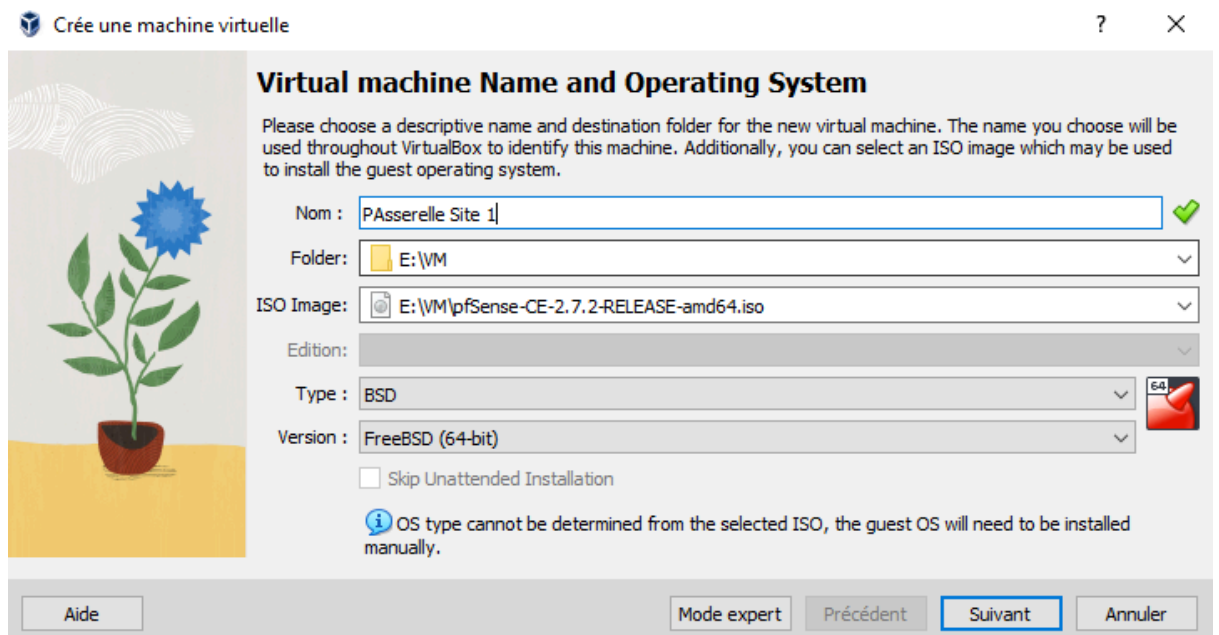
Automatique ☒

Adresse	Masque de réseau	Passerelle	Métrique	
				✕

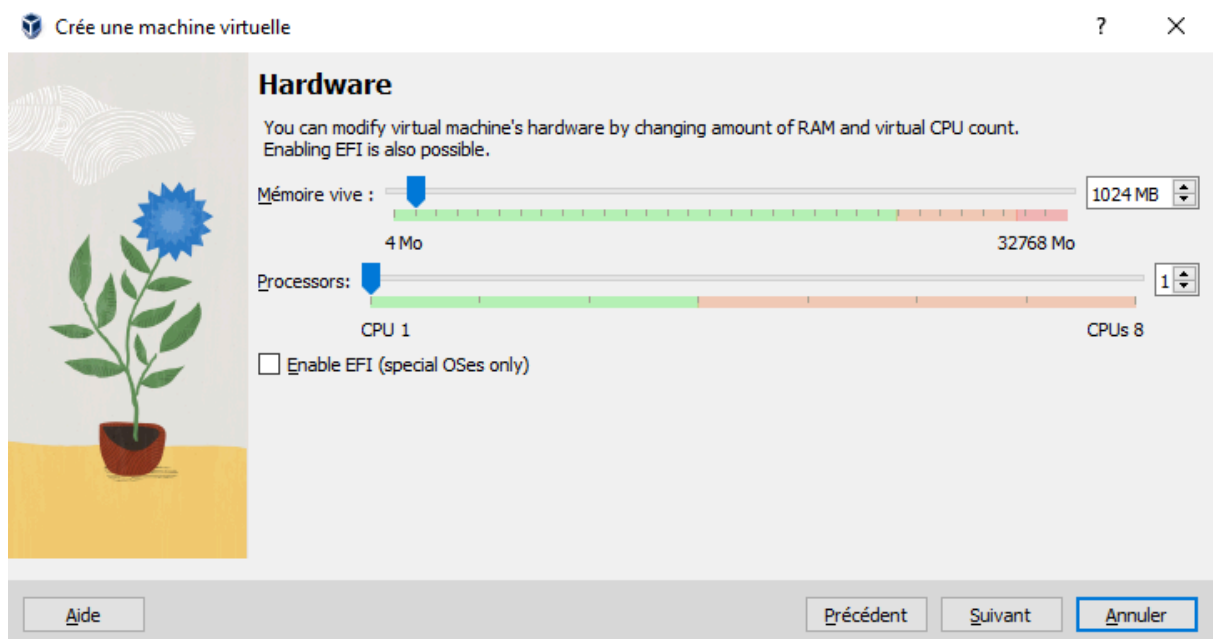
Faire de même pour les 2 serveurs VPN.  
Le 1er aura pour IP 1.2.3.12 et le 2ème 7.8.9.12.

#### 4) Installer et configurer la 1ère machine PfSense

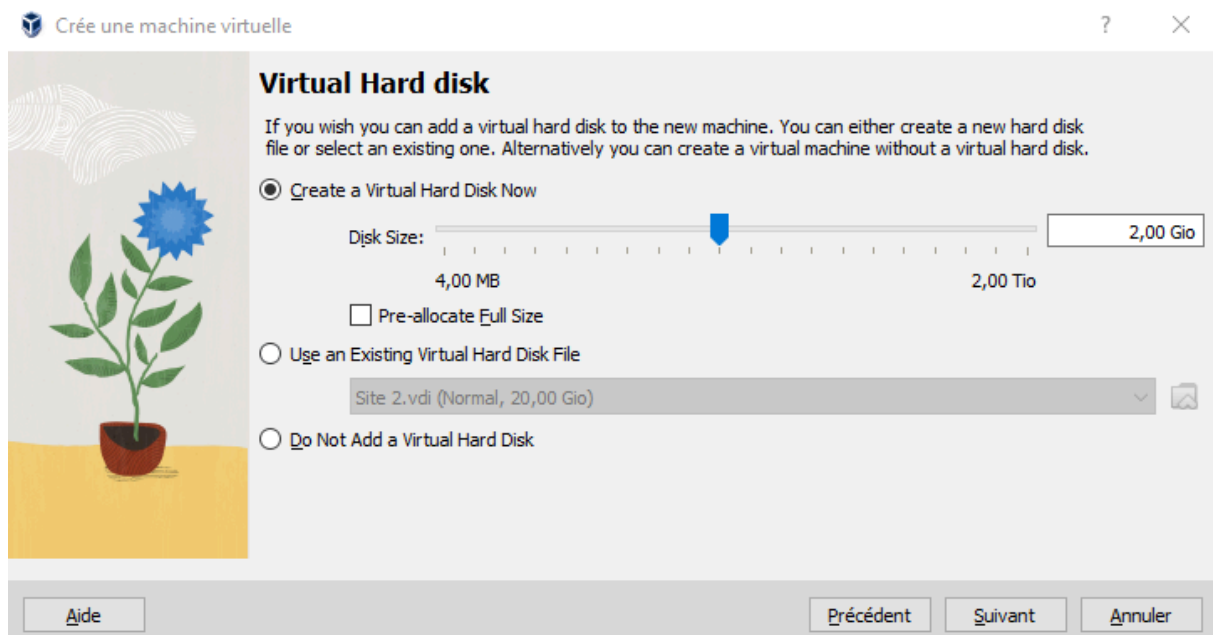
Créer la VM.  
Mettre en Type de système BSD et en version FreeBSD (64 bits).



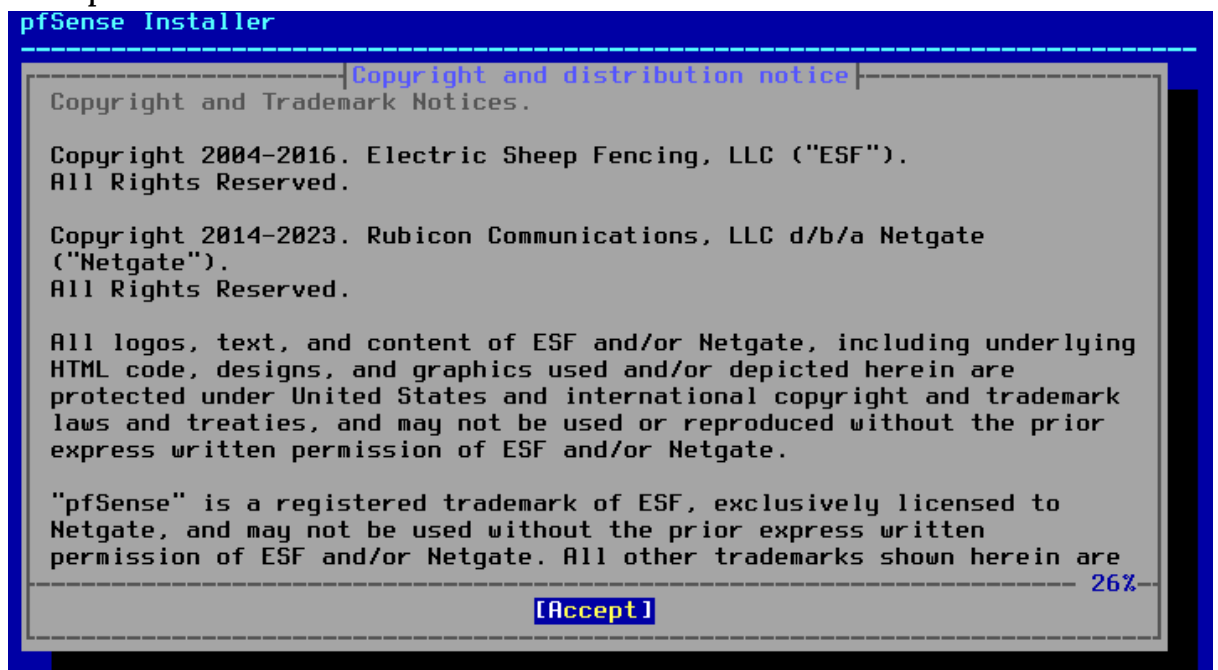
Mettre 1024 MB de ram et 1 CPU.



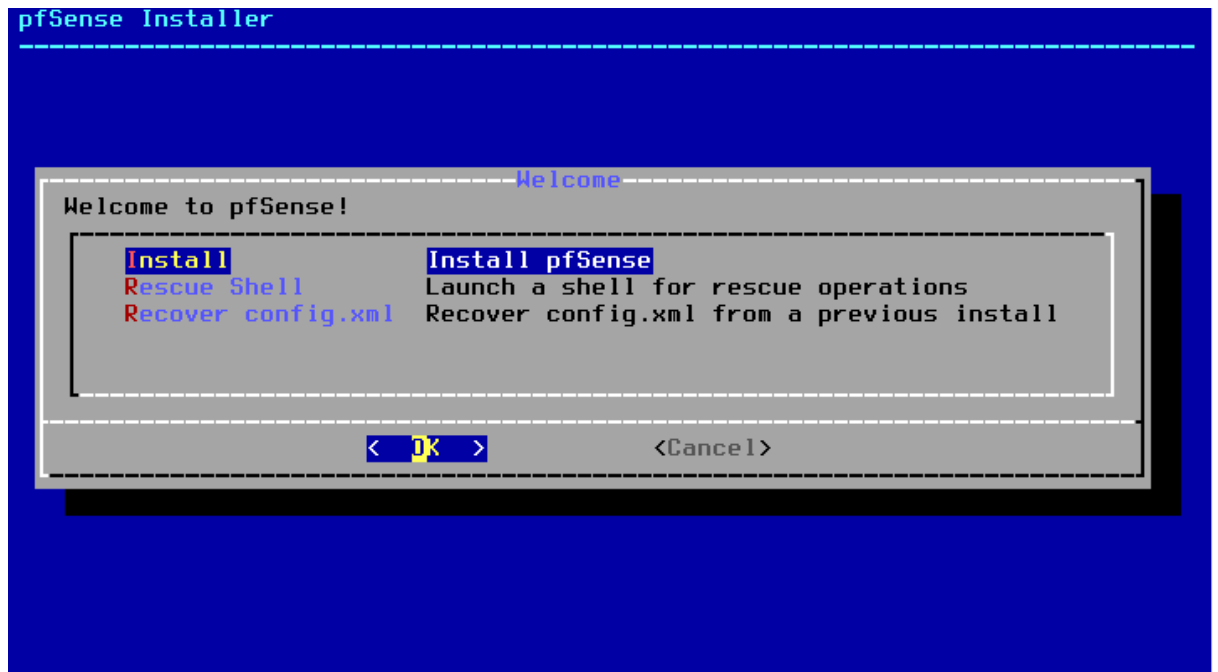
Mettre 2 Go de disque dur.



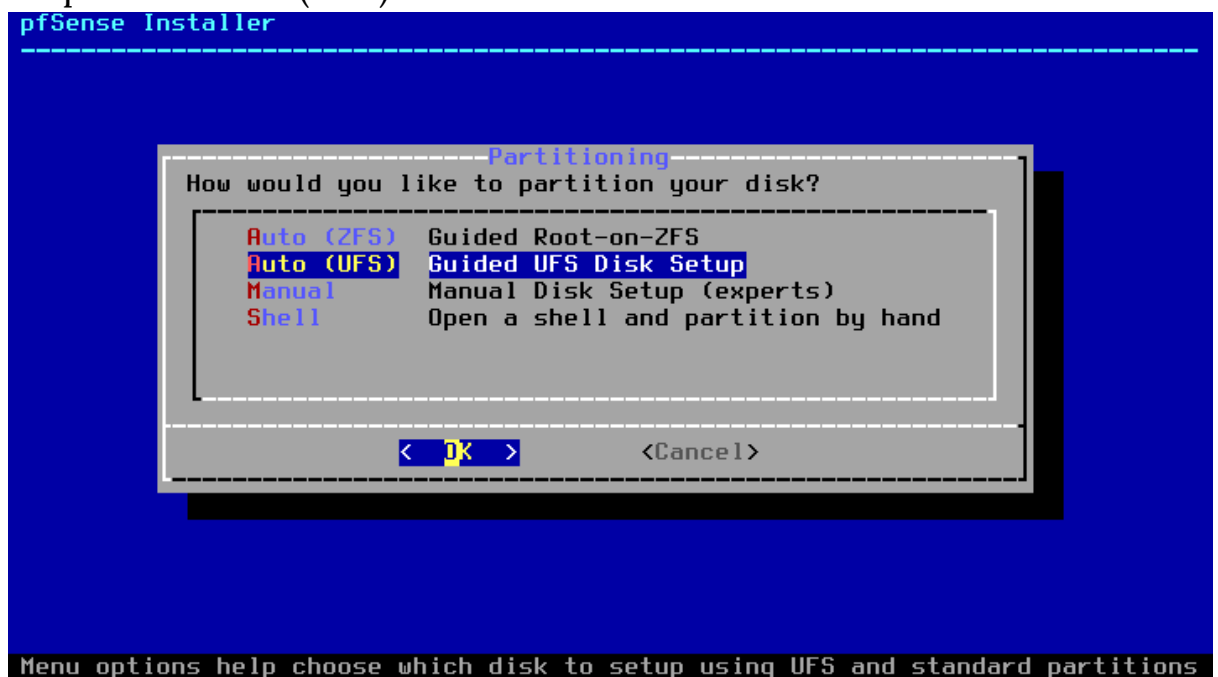
Accepter les conditions d'utilisation.



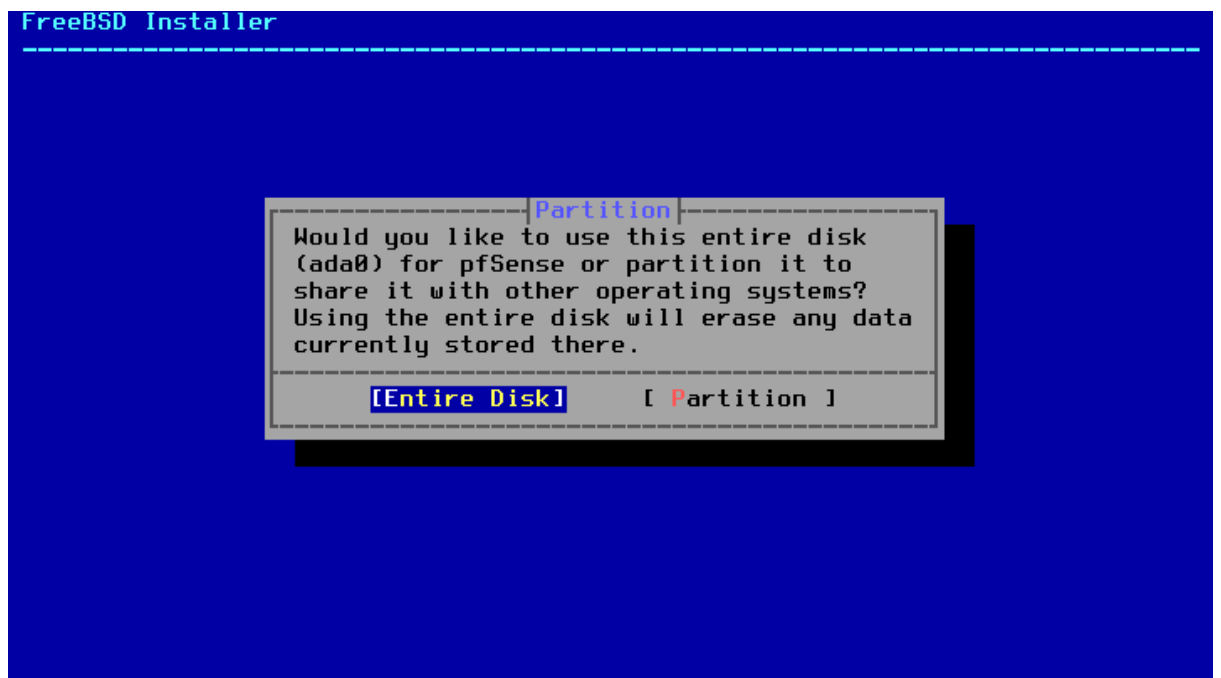
Cliquer sur "Install".



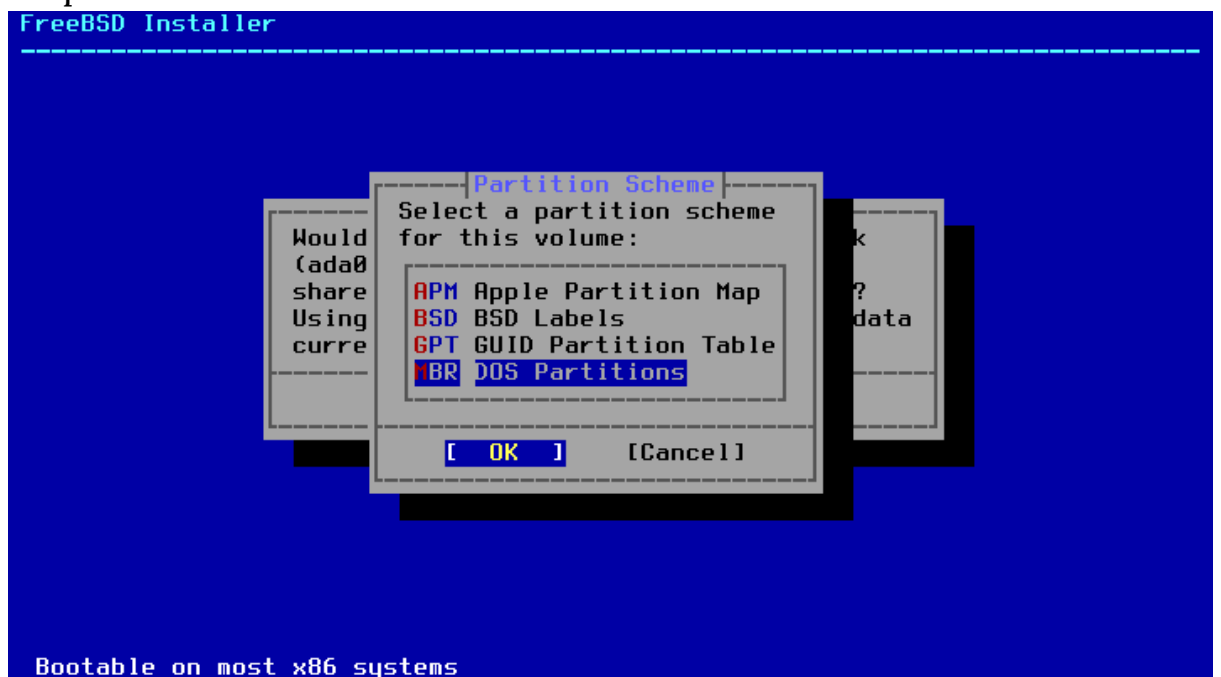
Cliquer sur Auto (UFS).



Cliquer sur "Entire disk".

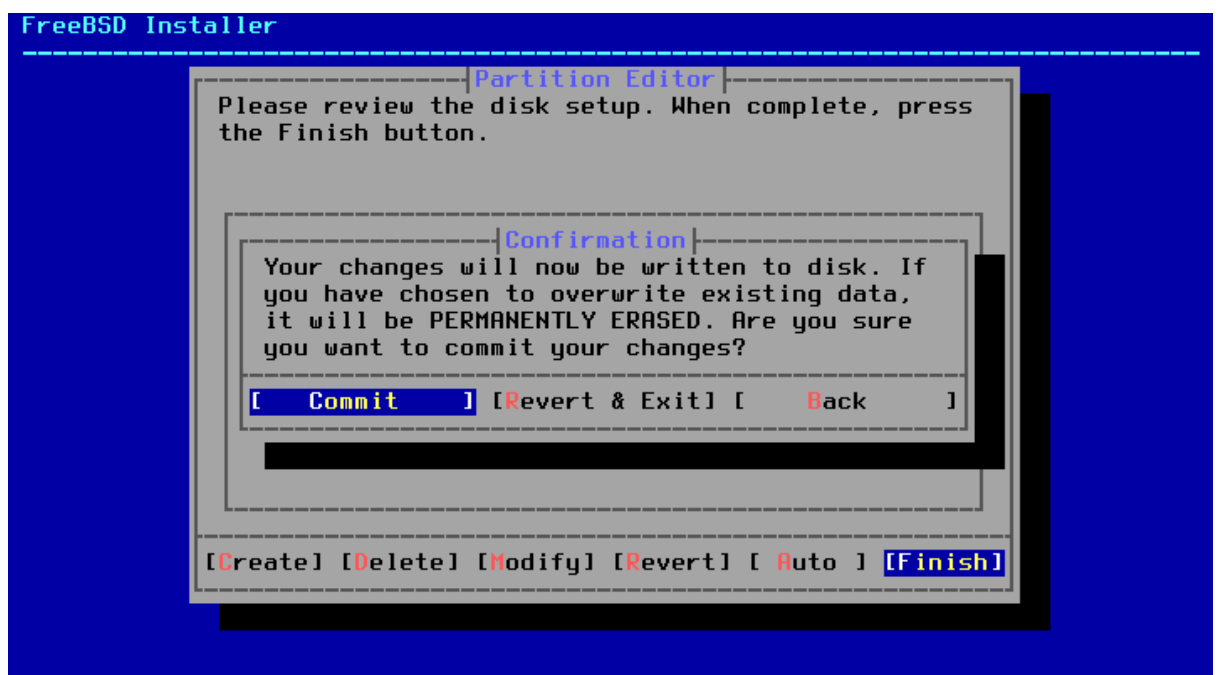
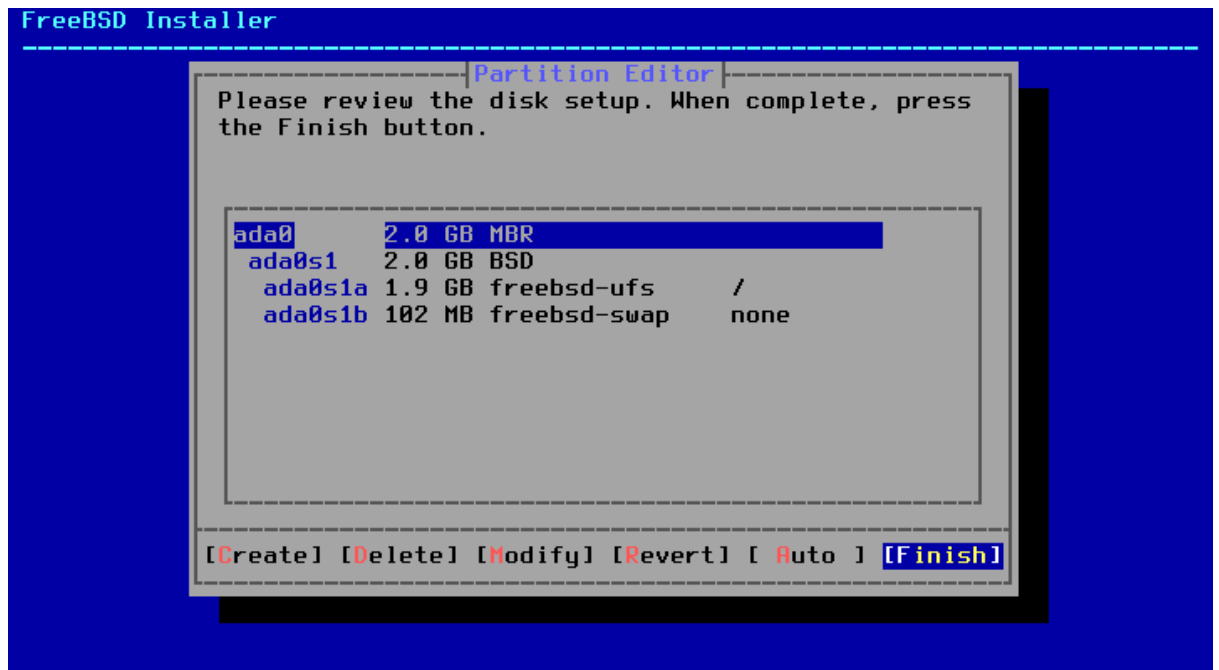


Cliquer sur DOS Partitions

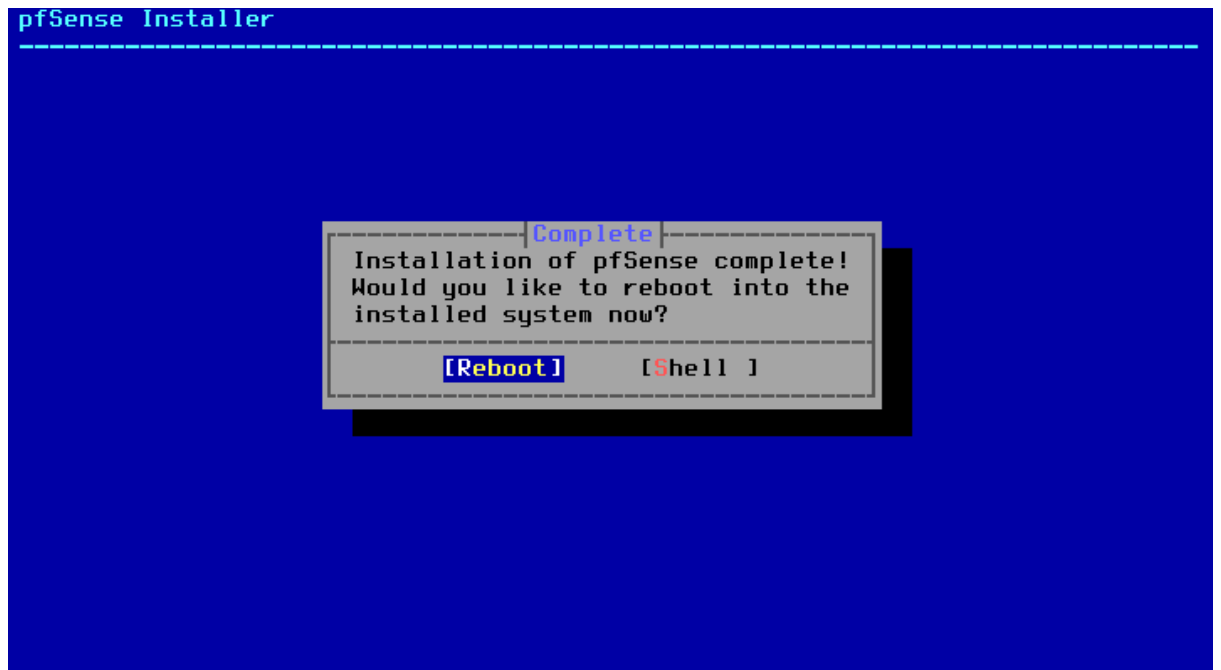


Cliquer sur Finish, puis sur Commit.

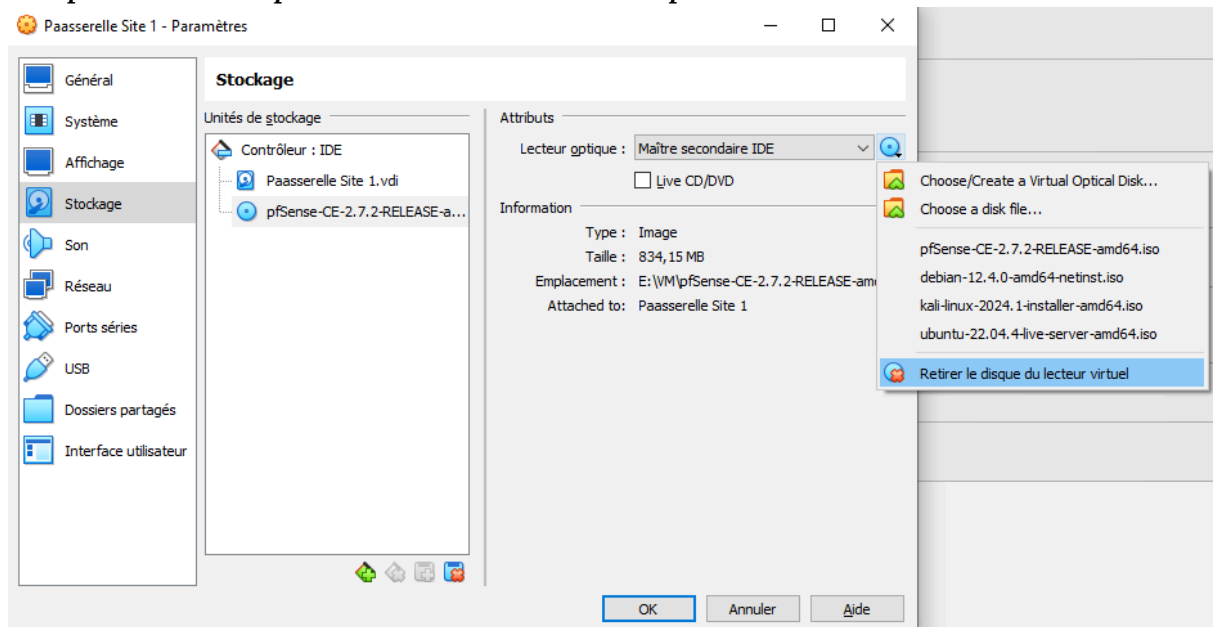




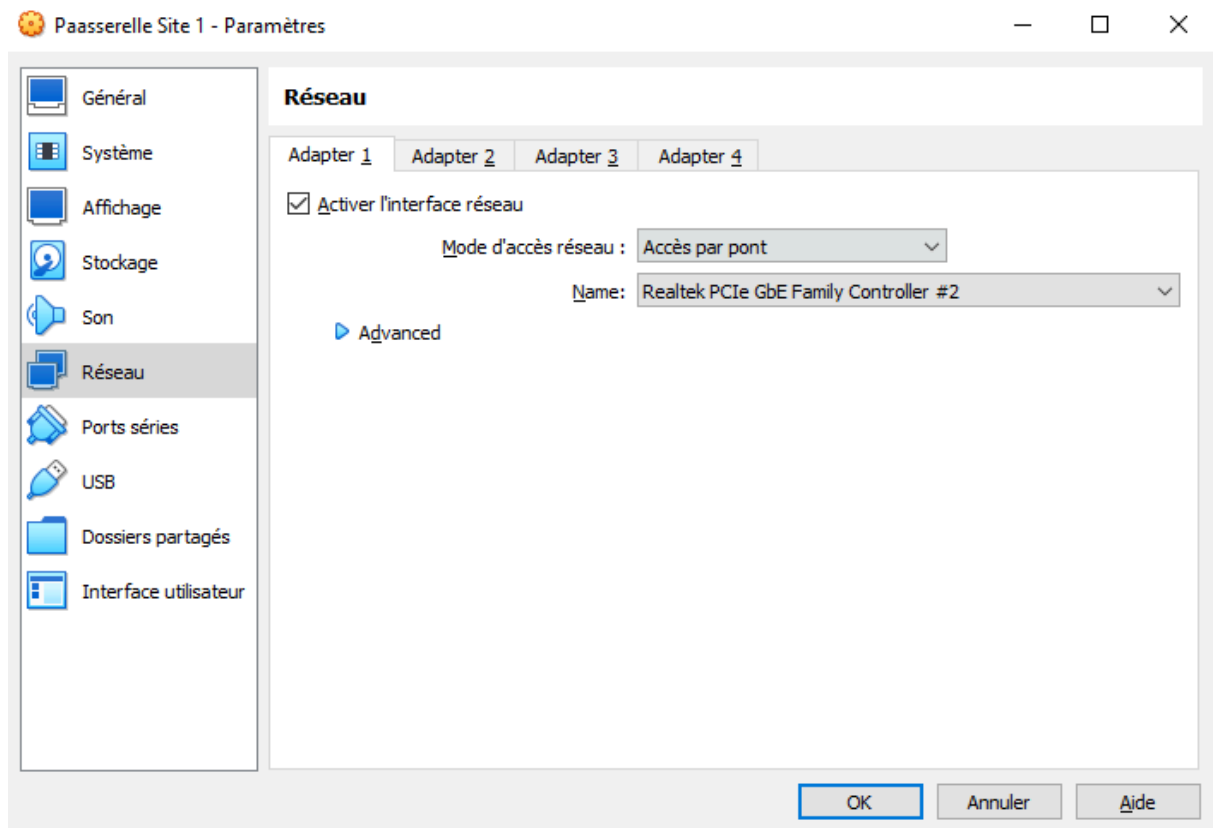
Une fois sur cet écran, éteindre la VM.



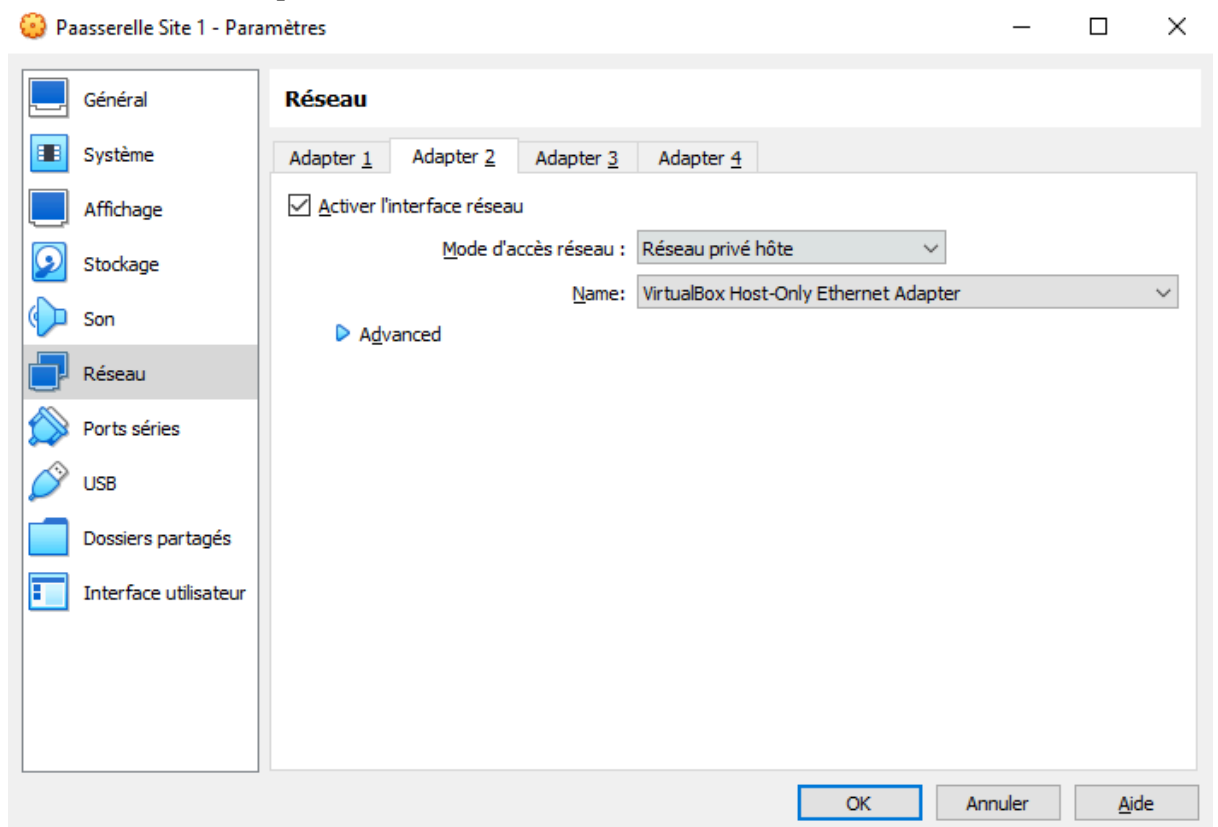
Aller dans les paramètres, puis dans Stockage. Cliquer sur le disque à droite puis sur “Retirer le disque du lecteur virtuel”.



Aller dans Réseau. Mettre le mode d'accès par pont pour la 1ère interface.



Aller dans “Adapter 2”. Activer l’interface, et mettre en type réseau “Réseau privé hôte”, en réseau le 1er réseau.



Relancer la VM.

```
FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)
VirtualBox Virtual Machine - Netgate Device ID: 2964c0434fa6ee1b340e
*** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

WAN (wan)          -> em0          -> v4/DHCP4: 192.168.1.27/24
                                   v6/DHCP6: 2a01:cb08:a83:c900:a00:27ff:fe67:20f
9/64
LAN (lan)          -> em1          -> v4: 192.168.1.1/24
                                   v6/t6: 2a01:cb08:a83:c9eb:a00:27ff:fef4:51ba/6
4

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) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █
```

Entrer 2, puis 2, puis n.

```
4
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) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: 2

Available interfaces:

1 - WAN (em0 - dhcp, dhcp6)
2 - LAN (em1 - static)

Enter the number of the interface you wish to configure: 2

Configure IPv4 address LAN interface via DHCP? (y/n) n

Enter the new LAN IPv4 address. Press <ENTER> for none:
> █
```

Entrer l'adresse IP 1.2.3.10 et 24 comme masque de sous-réseau.

```

Enter an option: 2

Available interfaces:

1 - WAN (em0 - dhcp, dhcp6)
2 - LAN (em1 - static)

Enter the number of the interface you wish to configure: 2

Configure IPv4 address LAN interface via DHCP? (y/n) n

Enter the new LAN IPv4 address. Press <ENTER> for none:
> 1.2.3.10

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
      255.255.0.0  = 16
      255.0.0.0    = 8

Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

```

Cliquer sur Enter, puis n, puis Enter, puis n, puis n, puis Enter.

```

For a LAN, press <ENTER> for none:
>

Configure IPv6 address LAN interface via DHCP6? (y/n) n

Enter the new LAN IPv6 address. Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) n
Disabling IPv4 DHCPD...
Disabling IPv6 DHCPD...

Do you want to revert to HTTP as the webConfigurator protocol? (y/n) n

Please wait while the changes are saved to LAN...
Reloading filter...
Reloading routing configuration...
DHCPD...

The IPv4 LAN address has been set to 1.2.3.10/24
You can now access the webConfigurator by opening the following URL in your web
browser:
      https://1.2.3.10/

Press <ENTER> to continue.

```

## 5) Installer et configurer la 2ème machine PfSense

Reprendre ce qui a été fait pour la 1ère passerelle. Modifier simplement l'adresse IP par 7.8.9.10

## 6) Vérifier la bonne installation de l'environnement

Pour vérifier la bonne installation de l'environnement, on va sur chacune des VMs Debian.

On tente de ping l'adresse de la passerelle PFSense sur le LAN, puis 8.8.8.8 pour voir si la passerelle fait bien son rôle.

On tente également de ping l'autre passerelle, afin de vérifier qu'il n'y a pas de connexion entre les deux réseaux privés.

```
root@debian:~# ping 1.2.3.10
PING 1.2.3.10 (1.2.3.10) 56(84) bytes of data.
64 bytes from 1.2.3.10: icmp_seq=1 ttl=64 time=0.415 ms
64 bytes from 1.2.3.10: icmp_seq=2 ttl=64 time=0.813 ms
64 bytes from 1.2.3.10: icmp_seq=3 ttl=64 time=1.19 ms
^C
--- 1.2.3.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.415/0.804/1.186/0.314 ms
root@debian:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=16.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=4.33 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=4.38 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 4.329/8.442/16.614/5.778 ms
root@debian:~# ping 7.8.9.10
PING 7.8.9.10 (7.8.9.10) 56(84) bytes of data.
^C
--- 7.8.9.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2039ms
```

```
root@debian:~# ping 7.8.9.10
PING 7.8.9.10 (7.8.9.10) 56(84) bytes of data.
64 bytes from 7.8.9.10: icmp_seq=1 ttl=64 time=0.698 ms
64 bytes from 7.8.9.10: icmp_seq=2 ttl=64 time=1.50 ms
64 bytes from 7.8.9.10: icmp_seq=3 ttl=64 time=0.552 ms
^C
--- 7.8.9.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.552/0.918/1.504/0.418 ms
root@debian:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=4.11 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=5.68 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=4.42 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 4.114/4.735/5.678/0.677 ms
root@debian:~# ping 1.2.3.10
PING 1.2.3.10 (1.2.3.10) 56(84) bytes of data.
^C
--- 1.2.3.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2027ms
```

### ***III. Mise en place du VPN IPsec***

1)



# Mise en place d'un VPN IPsec Site-à-Site avec StrongSwan

(musique d'ambiance :

<https://users.content.ytmnd.com/e/2/e/e2e7d0baf4799ab52ad89f89c9e84e4f.mp3>)

Notre infrastructure se compose de deux sites reliés par un réseau WAN. Voici la configuration détaillée de chaque site :

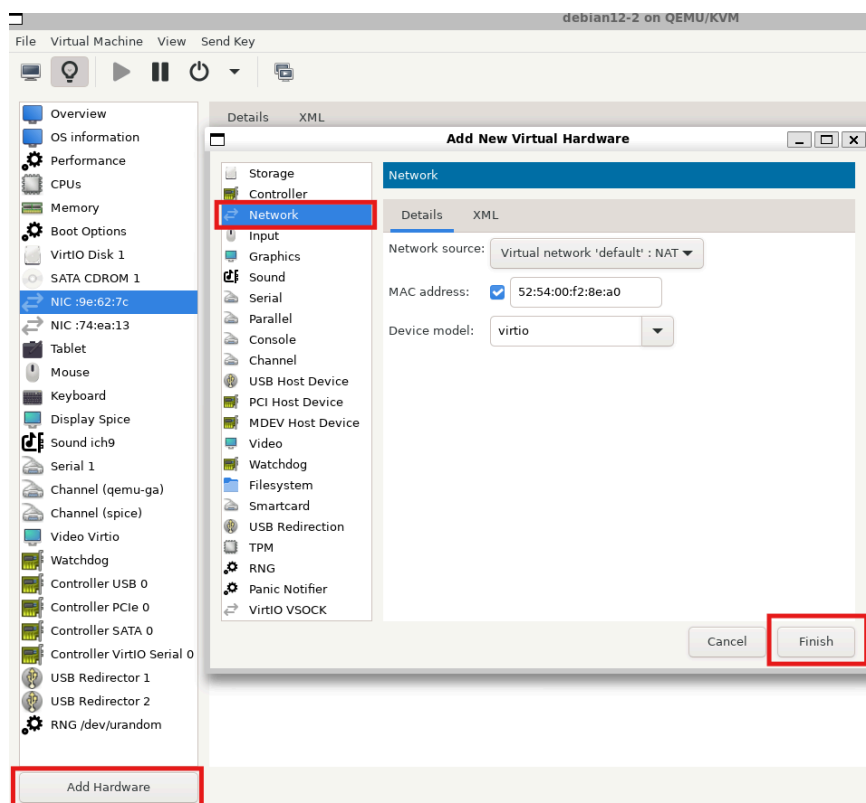
Site A (machine debian) :

- Interface WAN (enp1s0) : 192.168.122.87/24
- Interface LAN (eth0) : 10.0.2.1/24

Site B (machine debian2) :

- Interface WAN (enp1s0) : 192.168.122.86/24
- Interface LAN (enp7s0) : 10.0.1.1/24

Sous virt-manager (mais vous pouvez utiliser l'émulateur de votre choix), comment on ajoute des cartes réseaux :



## Phase 1 : Préparation de l'environnement (sur les deux machines)

(pour rappel, pour configurer les routes et changer l'adressage)

```
sudo ip addr flush dev [INTERFACE]
sudo ip addr add 10.0.2.1/24 dev [INTERFACE]
sudo ip link set [INTERFACE] up

sudo ip route add 10.0.2.0/24 via 192.168.122.87
```

```
sudo apt install strongswan strongswan-pki
libcharon-extra-plugins -y
# Activation du forwarding IP
sudo sysctl -w net.ipv4.ip_forward=1
```

## Phase 2 : Configuration IPsec

Pour le Site A, voici la configuration IPsec mise (/etc/ipsec.conf)

```
# Configuration générale de base
config setup
    charondebug="ike 2, knl 2, cfg 2, net 2, esp 2" #
Définit les niveaux de débogage pour différents composants
    uniqueids=yes # Garantit que chaque identifiant de
connexion est unique

conn site-a-to-b # Définit une nouvelle connexion nommée
"site-a-to-b"
    # Type de connexion
    type=tunnel # Spécifie qu'il s'agit d'un tunnel
(mode tunnel vs transport)
    auto=start # Démarre automatiquement la connexion
au lancement d'IPsec
    keyexchange=ikev2 # Utilise la version 2 du protocole
Internet Key Exchange

    # Configuration côté local
    left=%defaultroute # Utilise l'interface par défaut
pour la connexion
    leftauth=psk # Authentification par clé
pré-partagée
```

```

    leftid=@site-a      # Identifiant unique pour ce côté
du tunnel
    leftsubnet=10.0.2.0/24 # Réseau local à protéger

    # Configuration côté distant
    right=192.168.122.86  # Adresse IP du pair distant
    rightauth=psk         # Méthode d'authentification
du pair distant
    rightid=@site-b      # Identifiant du pair distant
    rightsubnet=10.0.1.0/24 # Réseau distant à atteindre

    # Paramètres de sécurité
    ike=aes256-sha256-modp2048! # Algorithmes pour la
phase 1 (négociation)
    esp=aes256-sha256!         # Algorithmes pour la
phase 2 (données)

    # Paramètres de durée de vie
    ikelifetime=3h          # Durée de vie de la phase 1
    keylife=1h             # Durée de vie des clés de la phase
2

    # Dead Peer Detection (détection de pair mort)
    dpddelay=30s           # Intervalle entre les vérifications
    dpdtimeout=120s       # Temps avant de considérer le pair
comme mort
    dpdaction=restart      # Action à prendre si le pair ne
répond pas

```

Cette configuration établit un tunnel IPsec où tout le trafic entre les deux réseaux (10.0.2.0/24 et 10.0.1.0/24) sera automatiquement chiffré et authentifié. La négociation des clés se fait via IKEv2, avec une renégociation automatique toutes les heures pour les clés de données et toutes les 3 heures pour les clés IKE. La détection de pair mort permet de relancer automatiquement le tunnel en cas de perte de connexion.

Sur le Site B (debian2), créez une configuration miroir (/etc/ipsec.conf)

```

GNU nano 7.2 /etc/ipsec.conf
config setup
    charondebug="ike 2, knl 2, cfg 2, net 2, esp 2"
    uniqueids=yes

conn site-b-to-a
    type=tunnel
    auto=start
    keyexchange=ikev2

    left=%defaulttroute
    leftauth=psk
    leftid=@site-b
    leftsubnet=10.0.1.0/24

    right=192.168.122.87
    rightauth=psk
    rightid=@site-a
    rightsubnet=10.0.2.0/24

    ike=aes256-sha256-modp2048!
    esp=aes256-sha256!

    ikelifetime=3h
    keylife=1h

    dpddelay=30s
    dpdtimeout=120s
    dpdaction=restart

```

Puis il faut faire la configuration des clés pré-partagées sur les deux machines (/etc/ipsec.secrets):

```
@site-a @site-b : PSK "VotreCleSecrete123!"
```

Pour plus de sécurité, on peut très bien faire une génération aléatoire de celui-ci via la commande :

```
head -c 32 /dev/urandom | base64
```

## Phase 3 : Configuration du pare-feu

Sur les deux machines, configurez les règles iptables:

```

sudo iptables -A INPUT -p udp --dport 500 -j ACCEPT
sudo iptables -A INPUT -p udp --dport 4500 -j ACCEPT
sudo iptables -A INPUT -p esp -j ACCEPT
sudo iptables -A INPUT -p ah -j ACCEPT

sudo iptables -A FORWARD -i enp1s0 -o enp7s0 -j ACCEPT
sudo iptables -A FORWARD -i enp7s0 -o enp1s0 -j ACCEPT

```

## Phase 4 : Démarrage et vérification

1. Démarrez le service IPsec sur les deux machines

```
sudo systemctl restart ipsec
```

```
sudo systemctl enable ipsec
```

On va vérifier l'état du tunnel

```
debian@debian:~$ sudo ipsec status
Security Associations (1 up, 0 connecting):
  site-a-to-b[1]: ESTABLISHED 44 minutes ago, 192.168.122.87[site-a]...192.168.122.86[site-b]
  site-a-to-b{2}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c1863ae2_i cb7cf220_o
  site-a-to-b{2}:  10.0.2.0/24 === 10.0.1.0/24
```

```
debian2@debian2:~$ sudo ipsec status
Security Associations (1 up, 0 connecting):
  site-b-to-a[3]: ESTABLISHED 45 minutes ago, 192.168.122.86[site-b]...192.168.122.87[site-a]
  site-b-to-a{4}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: cb7cf220_i c1863ae2_o
  site-b-to-a{4}:  10.0.1.0/24 === 10.0.2.0/24
```

```
debian@debian:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.8, Linux 6.1.0-30-amd64, x86_64):
  uptime: 45 minutes, since Jan 27 22:49:18 2025
  malloc: sbrk 2990080, mmap 0, used 1220512, free 1769568
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 3
  loaded plugins: charon aesni aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs12 pgp dnskey ssh
  key pem openssl pkcs8 fips-prf gmp agent xcbc hmac kdf gcm drbg attr kernel-netlink resolve socket-default connmark forecast farp stroke up
  down eap-identity eap-aka eap-md5 eap-gtc eap-mschapv2 eap-radius eap-tls eap-ttls eap-tnc xauth-generic xauth-eap xauth-pam tnc-tncs dhcp
  lookup error-notify certexpire led addrblock unity counters
Listening IP addresses:
  192.168.122.87
  192.168.122.84
  10.0.2.1
  192.168.122.228
Connections:
  site-a-to-b: %any...192.168.122.86 IKEv2, dpddelay=30s
  site-a-to-b:  local: [site-a] uses pre-shared key authentication
  site-a-to-b:  remote: [site-b] uses pre-shared key authentication
  site-a-to-b:  child: 10.0.2.0/24 === 10.0.1.0/24 TUNNEL, dpdaction=start
Security Associations (1 up, 0 connecting):
  site-a-to-b[1]: ESTABLISHED 45 minutes ago, 192.168.122.87[site-a]...192.168.122.86[site-b]
  site-a-to-b[1]: IKEv2 SPIs: f9e20alcfae49431_i* 9116d6862e5a0881_r, pre-shared key reauthentication in 2 hours
  site-a-to-b[1]: IKE proposal: AES_CBC_256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_2048
  site-a-to-b{2}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c1863ae2_i cb7cf220_o
  site-a-to-b{2}:  AES_CBC_256/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 41 minutes
  site-a-to-b{2}:  10.0.2.0/24 === 10.0.1.0/24
```

```

debian2@debian2:~$ sudo ipsec statusall
Status of IKE charon daemon (strongSwan 5.9.8, Linux 6.1.0-30-amd64, x86_64):
  uptime: 46 minutes, since Jan 27 22:49:11 2025
  malloc: sbrk 2850816, mmap 0, used 1232592, free 1618224
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 5
  loaded plugins: charon aesni aes rc2 sha2 sha1 md5 mgf1 random nonce x509 revocation constraints pubkey pkcs1
pkcs7 pkcs12 pgp dnskey sshkey pem openssl pkcs8 fips-prf gmp agent xcbc hmac kdf gcm drbg attr kernel-netlink r
esolve socket-default connmark forecast farp stroke updown eap-identity eap-aka eap-md5 eap-gtc eap-mschapv2 eap
-radius eap-tls eap-ttls eap-tnc xauth-generic xauth-eap xauth-pam tnc-tncs dhcp lookup error-notify certexpire
led addrblock unity counters
Listening IP addresses:
  192.168.122.86
  192.168.122.123
Connections:
  site-b-to-a: %any...192.168.122.87 IKEv2, dpddelay=30s
  site-b-to-a: local: [site-b] uses pre-shared key authentication
  site-b-to-a: remote: [site-a] uses pre-shared key authentication
  site-b-to-a: child: 10.0.1.0/24 === 10.0.2.0/24 TUNNEL, dpdaction=start
Security Associations (1 up, 0 connecting):
  site-b-to-a[3]: ESTABLISHED 46 minutes ago, 192.168.122.86[site-b]...192.168.122.87[site-a]
  site-b-to-a[3]: IKEv2 SPIs: f9e20a1cfae49431_i 9116d6862e5a0881_r*, pre-shared key reauthentication in 119 minu
tes
  site-b-to-a[3]: IKE proposal: AES_CBC_256/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_2048
  site-b-to-a[4]: INSTALLED, TUNNEL, reqid 1, ESP SPIs: cb7cf220_i c1863ae2_o
  site-b-to-a[4]: AES_CBC_256/HMAC_SHA2_256_128, 0 bytes_i, 0 bytes_o, rekeying in 40 minutes
  site-b-to-a[4]: 10.0.1.0/24 === 10.0.2.0/24

```

On va tester si le trafic est bien chiffré

```

debian@debian:~$ ping 10.0.1.1
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=64 time=2.87 ms
64 bytes from 10.0.1.1: icmp_seq=3 ttl=64 time=1.54 ms
64 bytes from 10.0.1.1: icmp_seq=4 ttl=64 time=2.13 ms

```

```

debian@debian:~$ sudo tcpdump -n -v -i enp1s0 esp or udp port 500 or udp port 4500
tcpdump: listening on enp1s0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:38:49.060829 IP (tos 0x0, ttl 64, id 29490, offset 0, flags [DF], proto UDP (17), length 112)
  192.168.122.87.4500 > 192.168.122.86.4500: NONESP-encap: isakmp 2.0 msgid 00000058: child_sa inf2[I]:
  (v2e: len=48)
23:38:49.062959 IP (tos 0x0, ttl 64, id 32331, offset 0, flags [DF], proto UDP (17), length 112)
  192.168.122.86.4500 > 192.168.122.87.4500: NONESP-encap: isakmp 2.0 msgid 00000050: child_sa inf2:
  (v2e: len=48)
23:38:49.062974 IP (tos 0x0, ttl 64, id 32332, offset 0, flags [DF], proto UDP (17), length 112)
  192.168.122.86.4500 > 192.168.122.87.4500: NONESP-encap: isakmp 2.0 msgid 00000058: child_sa inf2[R]:
  (v2e: len=48)
23:38:49.064158 IP (tos 0x0, ttl 64, id 29491, offset 0, flags [DF], proto UDP (17), length 112)
  192.168.122.87.4500 > 192.168.122.86.4500: NONESP-encap: isakmp 2.0 msgid 00000050: child_sa inf2[IR]:
  (v2e: len=48)
23:38:50.610265 IP (tos 0x0, ttl 64, id 27966, offset 0, flags [DF], proto ESP (50), length 156)
  192.168.122.87 > 192.168.122.86: ESP(spi=0xcb7cf220,seq=0x9), length 136
23:38:50.611291 IP (tos 0x0, ttl 64, id 58374, offset 0, flags [none], proto ESP (50), length 156)
  192.168.122.86 > 192.168.122.87: ESP(spi=0xc1863ae2,seq=0x4), length 136

```