

NSA POOL – YOU SHALL NOT PASS

PROJET NSA 501 DEVOPS

YOU SHALL NOT PASS



MARTIN DAIGNAN | EREN VARLI | 16 DÉCEMBRE 2024 |
EPITECH

CONTEXTE

Nous avons pour mission de mettre en place 4 machines virtuelles :

- Gateway (VM1)
- **Serveur web** (VM2)
- **Administration** (VM3)
- **Employé** (VM4)

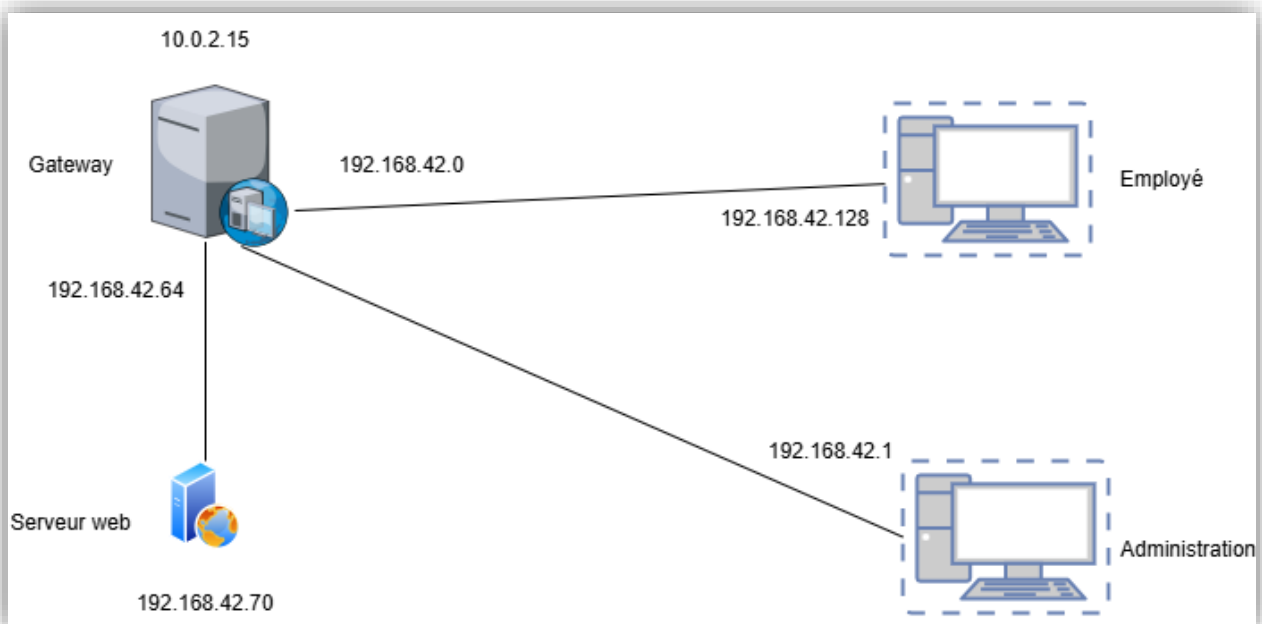
Dans cette documentation, nous allons expliquer en détail comment chaque VM a été créée ainsi que les configurations associées.

SCHÉMA

Le schéma ci-dessous montre la configuration de quatre machines virtuelles : **VM1 (Gateway)**, **VM2 (Serveur Web)**, **VM3 (Administration)** et **VM4 (Employé)**, reliées via quatre interfaces réseau : **NAT (em0)** pour l'accès à Internet, **em1** pour l'**administration**, **em2** pour le **serveur** et **em3** pour les **employés**.

Le **DHCP** est configuré sur la **Gateway** pour attribuer dynamiquement des adresses IP aux réseaux internes. Le filtrage de paquets (pf) est mis en place pour sécuriser les connexions, en autorisant uniquement certains types de trafic (comme HTTP, HTTPS pour les **employés**). Le **NAT** masque les adresses internes pour les connexions sortantes.

Les réseaux peuvent se connecter à Internet, récupérer des informations DHCP et DNS, et échanger des paquets en toute sécurité via des règles strictes de filtrage.



CRÉATION DES MACHINES

Il est **important** de créer premièrement la VM1 ; en effet, cette dernière sera la machine dans laquelle nous allons configurer nos paramètres **réseau** et le DHCP* (Dynamic Host Configuration Protocol).

Nous verrons ce qu'est le DHCP un peu plus tard.

VM1 – Installation

Les consignes de la VM1 sont les suivantes :

- Basée sur OpenBSD 7.6
- Comporte 4 réseaux (1 NAT, 3 réseaux privés)



Partie 1- VirtualBox

Nous commençons par configurer la VM1 sur **VirtualBox**, mais avant, il est impératif d'installer le fichier ISO d'OpenBSD. Un fichier ISO est, en termes simples, un format de fichier numérique reproduisant un CD, un DVD ou un BD physique. L'extension de fichier **ISO** ne se contente pas de stocker des fichiers et des dossiers : elle contient toutes les informations vitales du système de fichiers concernant la structure du disque. On peut

openbsd.org/faq/faq4.html#Download

- IP address and subnet masks for each NIC
- Gateway address

Downloading OpenBSD

The following installation images are available:

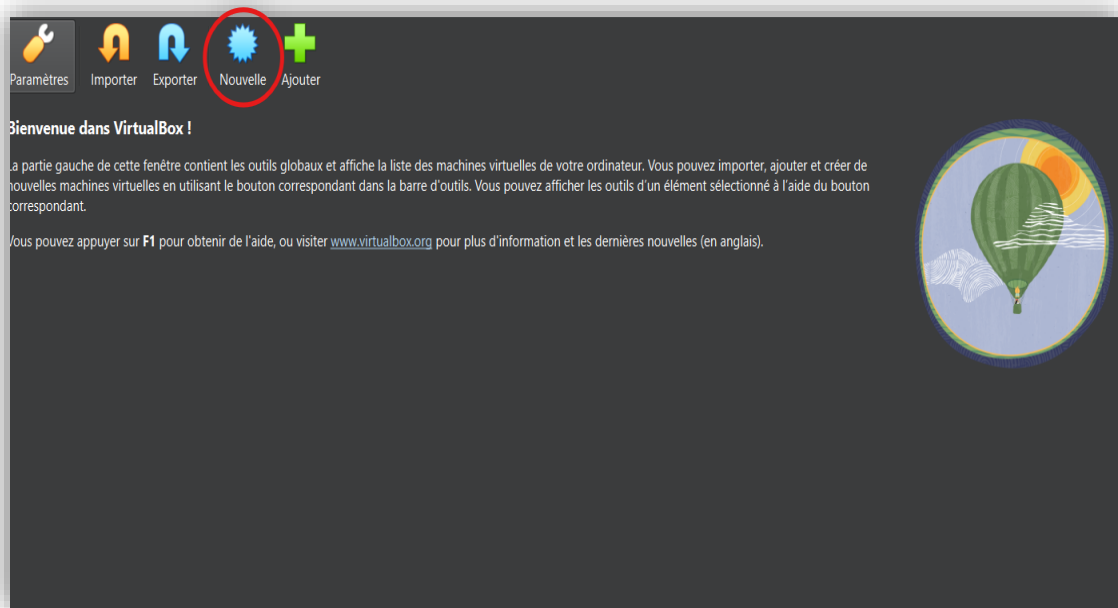
install76.img	A disk image that can be written to a USB flash drive or similar device. Includes the file sets . amd64 arm64 i386 octeon powerpc64 risev64 sparc64
miniroot76.img	The same as above, but file sets are not included. They can be pulled down from the internet or from a local disk. alpha amd64 arm64 armv7 i386 landisk loongson luna88k octeon powerpc64 risev64 sparc64
install76.iso	An ISO 9660 image that can be used to create an install CD/DVD. Includes the file sets. alpha amd64 arm64 hppa i386 macppc powerpc64 sparc64
cd76.iso	The same as above, but file sets are not included. alpha amd64 hppa i386 loongson macppc sparc64
floppy76.img	Supports some older machines that lack other booting options. amd64 i386 sparc64

Images can also be downloaded from a number of alternate [mirror sites](#).

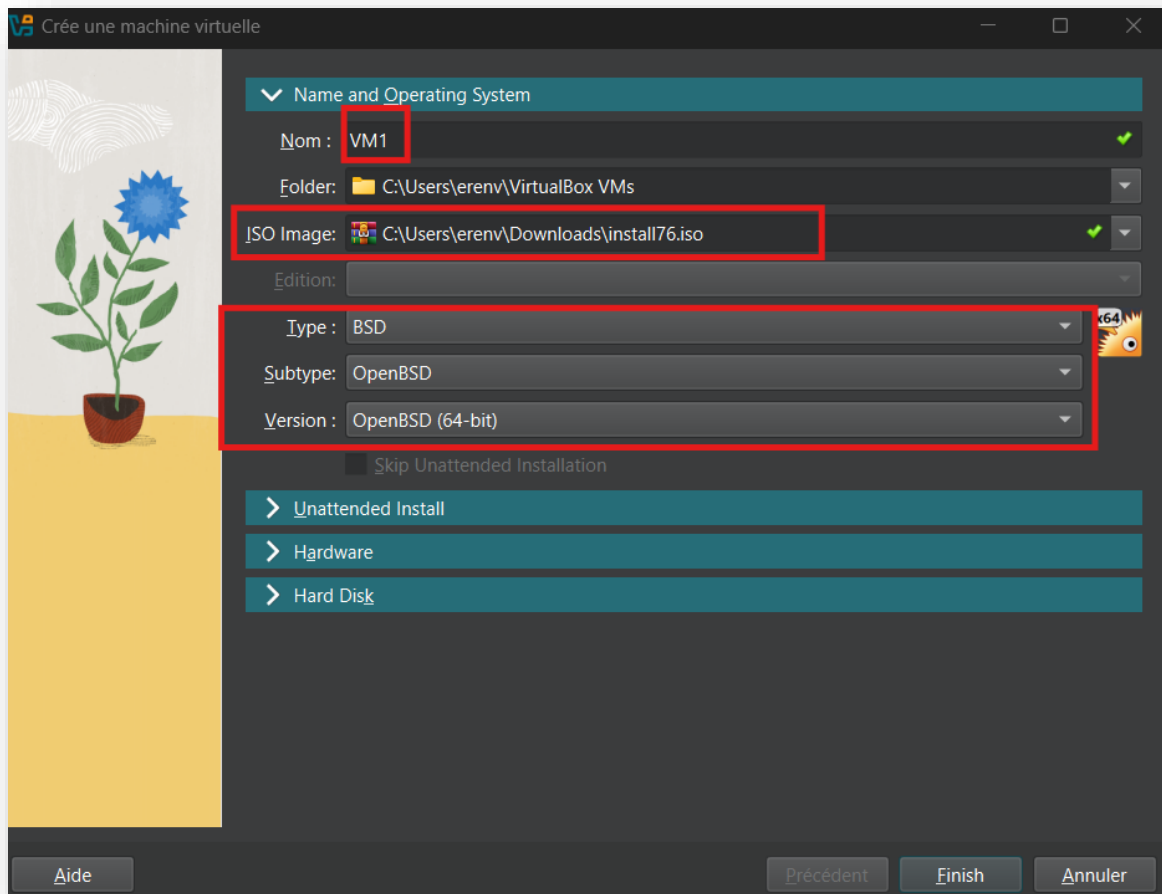
télécharger l'ISO sur le site officiel d'**OpenBSD** comme suit, ce qui va nous permettre d'installer OpenBSD et de le configurer.

Nous pouvons dès à présent configurer la machine et les paramètres demandés sur VirtualBox comme suit :

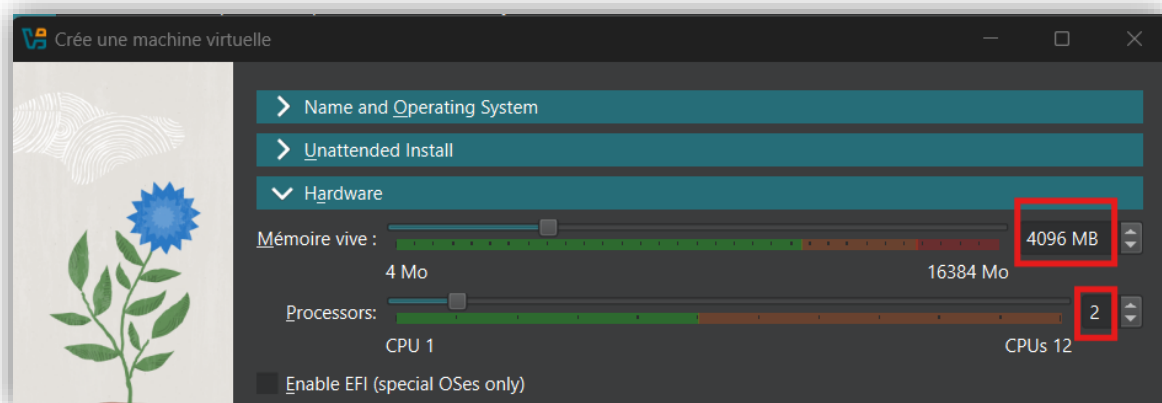
Étape 1 – Sélectionner « Nouvelle » pour créer la machine.



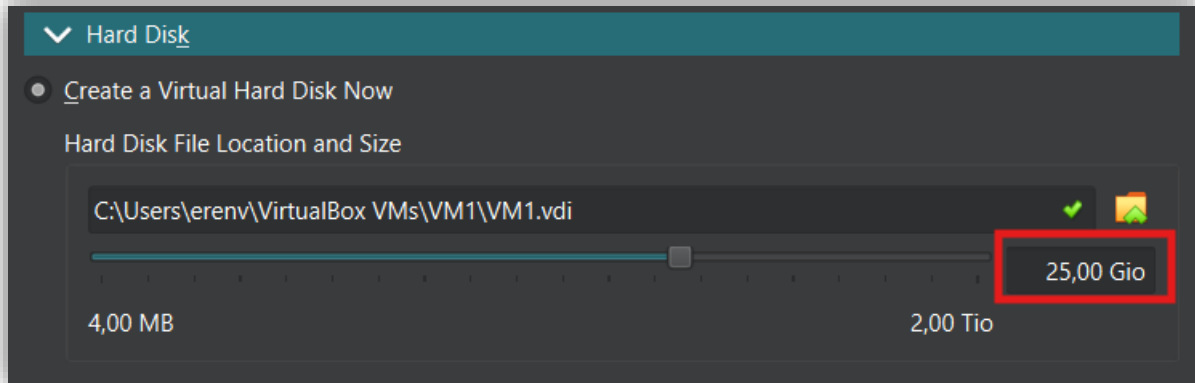
Étape 2 – Attribuer un nom à la VM, sélectionner l'ISO que nous venons d'installer et choisir l'OS OpenBSD.



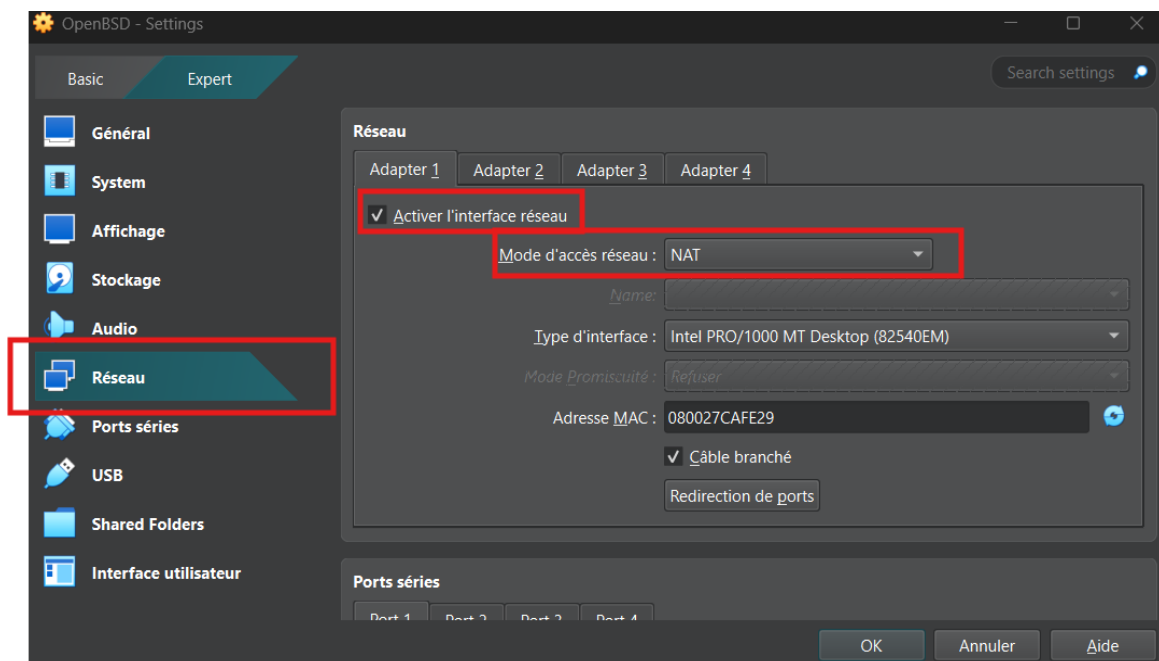
Étape 3 – Allouer de la **mémoire vive** à la VM, sachant qu'elle n'aura **pas d'interface graphique** et qu'il y aura peu d'installations à effectuer. On peut se permettre d'allouer une **petite quantité** de mémoire : 4096 Mo suffiront largement, ainsi que 2 cœurs pour le processeur. Ces paramètres sont largement suffisants pour assurer le bon fonctionnement de la machine.



Étape 4 – Vous devez ensuite attribuer du **stockage** sur le disque de la machine. Comme pour l'étape précédente, vous pouvez vous permettre de donner une petite taille, comme 25 Go. Vous avez le libre choix de l'emplacement, mais dans notre cas, nous laissons l'emplacement par défaut.



Étape 5 – Une étape **importante**, nous allons attribuer les paramètres de réseau donnés dans les consignes. Pour cela, cliquez droit sur la VM > Configuration > Réseau :

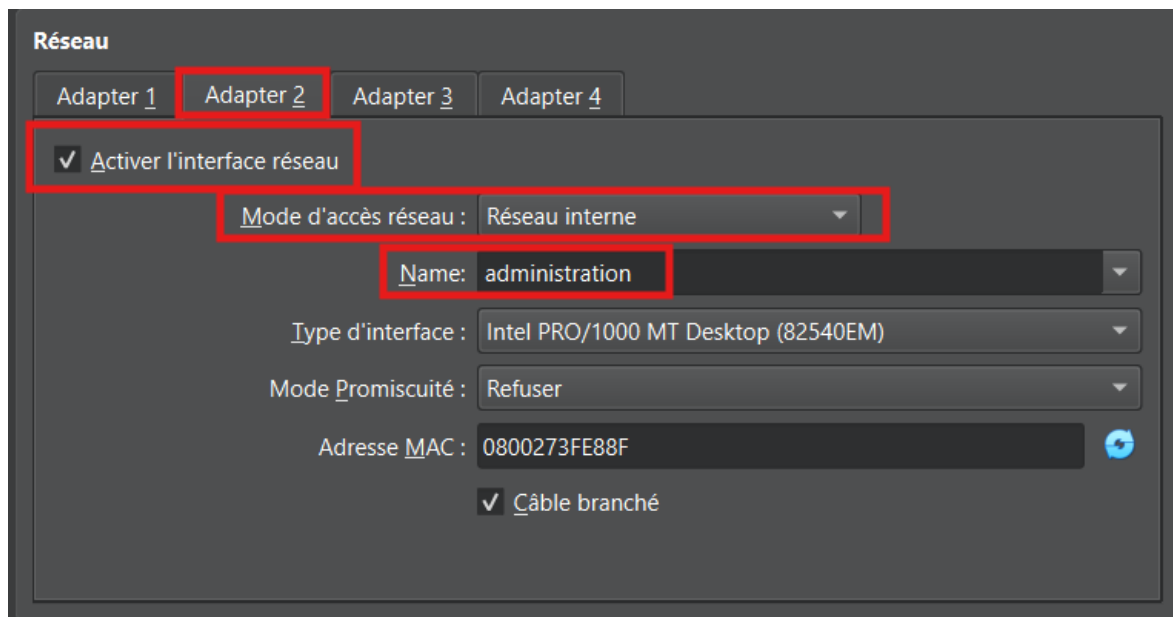


Rappelez-vous, nous devons configurer 4 réseaux.

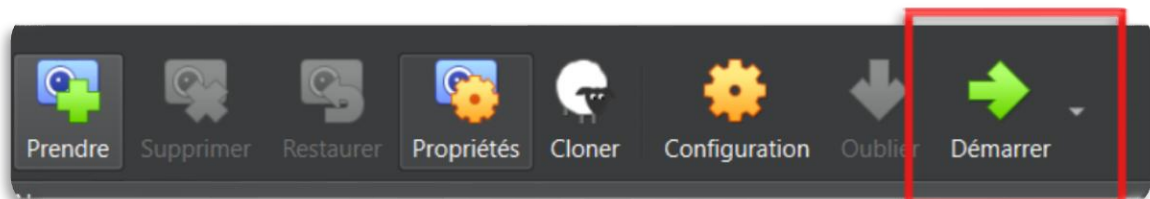
Pour cela, cochez « Activez l'interface réseau » et configurez les adaptateurs comme ceci :

- Adaptateur 1 = NAT
- Adaptateur 2 = Réseau interne sous le nom « **administration** »
- Adaptateur 3 = Réseau interne sous le nom « **server** »
- Adaptateur 4 = Réseau interne sous le nom « **employee** »

Exemple pour l'adaptateur 2 :



La machine est maintenant prête à être démarrée. Lancez-la et vous tomberez sur le processus d'installation.



Partie 2- Processus d'installation OpenBSD



Laissez la machine démarrer, et sélectionnez I pour choisir l'option « Install » .

```
em1 at pci0 dev 8 function 0 "Intel 82540EM" rev 0x02: apic 1 int 16, address 08:00:27:3f:e8:8f
em2 at pci0 dev 9 function 0 "Intel 82540EM" rev 0x02: apic 1 int 17, address 08:00:27:b7:32:fb
em3 at pci0 dev 10 function 0 "Intel 82540EM" rev 0x02: apic 1 int 18, address 08:00:27:8a:18:18
ehci0 at pci0 dev 11 function 0 "Intel 82801FB USB" rev 0x00: apic 1 int 19
usb0 at ehci0: USB revision 2.0
uhub0 at usb0 configuration 1 interface 0 "Intel EHCI root hub" rev 2.00/1.00 address 1
usb1 at ohci0: USB revision 1.0
uhub1 at usb1 configuration 1 interface 0 "Apple OHCI root hub" rev 1.00/1.00 address 1
isa0 at mainbus0
pckbc0 at isa0 port 0x60/5 irq 1 irq 12
pckbd0 at pckbc0 (kbd slot)
wskbd0 at pckbd0: console keyboard, using wsdplay0
softraid0 at root
scsibus1 at softraid0: 256 targets
root on rd0a swap on rd0b dump on rd0b
WARNING: CHECK AND RESET THE DATE!
erase ^?, werase ^W, kill ^U, intr ^C, status ^T

Welcome to the OpenBSD/amd64 7.6 installation program.
(I)nstall, (U)pgrade, (A)utoinstall or (S)hell? I_
```

Vous pouvez commencer à répondre aux options proposées lors du processus d'installation. La plupart des réponses sont prédéfinies par défaut, il vous suffira donc d'appuyer sur la touche ENTRÉE pour les valider.

Configuration du clavier en français (fr)

```
Choose your keyboard layout ('?' or 'L' for list) [default] fr_
```

Attribution du nom du système (nsapool)

```
System hostname? (short form, e.g. 'foo') nsapool_
```

Ajoutez un utilisateur pour éviter d'utiliser root (nom en minuscule)

```
Setup a user? (enter a lower-case loginname, or 'no') [no] eren_
```

L'installation complète d'OpenBSD est divisée en plusieurs ensembles de fichiers :

bsd	Le noyau (obligatoire)
bsd.mp	Le noyau multiprocesseur (uniquement sur certaines plateformes)
bsd.rd	Le noyau du disque virtuel
base76.tgz	Le système de base (obligatoire)
comp76.tgz	La collection de compilateurs, les en-têtes et les bibliothèques
man76.tgz	Pages du manuel
game76.tgz	Jeux basés sur du texte
xbase76.tgz	Bibliothèques et utilitaires de base pour X11 (nécessite xshare76.tgz)
xfont76.tgz	Polices utilisées par X11
xserv76.tgz	Les serveurs X de X11
xshare76.tgz	Pages de manuel, paramètres régionaux et inclusions de X11

Il est recommandée de tous les installer, donc nous l'avons fais.

ATTENTION

Pour notre cas, nous sommes sur VirtualBox, il n'y a rien à craindre. En revanche, **si vous êtes sur une machine physique, soyez prudent à ne pas effacer des données déjà stockées dans un emplacement du disque.** Ici, l'installation se fait automatiquement dans le disk3, un emplacement libre pour OpenBSD, donc tout ce qui est sur le disk3 sera effacé.

Il faut s'assurer que les racines sont bien créées dans les partitions (/tmp, /var, /usr, /home).

```

...for root account? (again)
Start sshd(8) by default? [yes]
Do you expect to run the X Window System? [yes]
Do you want the X Window System to be started by xenodm(1)? [no]
Setup a user? (enter a lower-case loginname, or 'no') [no] eren
Full name for user eren? [eren] eren
Password for user eren? (will not echo)
Password for user eren? (again)
WARNING: root is targeted by password guessing attacks, pubkeys are safer.
Allow root ssh login? (yes, no, prohibit-password) [no]
What timezone are you in? ('?' for list) [Europe/Paris]

Available disks are: wd0.
Which disk is the root disk? ('?' for details) [wd0]
Encrypt the root disk with a (p)assphrase or (k)eydisk? [no]
Disk: wd0      geometry: 3263/255/63 [52428800 Sectors]
Offset: 0      Signature: 0xAA55

#  id      Starting      Ending      LBA Info:
#  id      C  H  S  -    C  H  S  I    start:      size I
-----
0: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
1: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
2: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
3: A6      0  1  2  -    3263 138 11 I    64:      52428736 I OpenBSD
(W)hole disk MBR, whole disk (G)PT, (O)penBSD area or (E)dit? [OpenBSD]

```

```

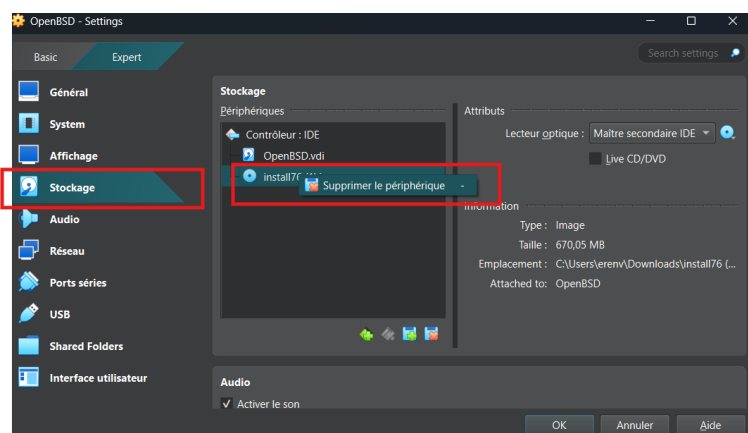
Encrypt the root disk with a (p)assphrase or (k)eydisk? [no]
Disk: wd0      geometry: 3263/255/63 [52428800 Sectors]
Offset: 0      Signature: 0xAA55

#  id      Starting      Ending      LBA Info:
#  id      C  H  S  -    C  H  S  I    start:      size I
-----
0: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
1: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
2: 00      0  0  0  -    0  0  0  I    0:          0 I Unused
3: A6      0  1  2  -    3263 138 11 I    64:      52428736 I OpenBSD
Use (W)hole disk MBR, whole disk (G)PT, (O)penBSD area or (E)dit? [OpenBSD]
The auto-allocated layout for wd0 is:

#      size      offset      fstype [fsize bsize cpgr]
a:      853.5M          64      4.2BSD      2048 16384      1 # /
b:     1487.0M     1748000      swap
c:     25600.0M          0      unused
d:     1245.6M     4793376      4.2BSD      2048 16384      1 # /tmp
e:     1909.1M     7344320      4.2BSD      2048 16384      1 # /var
f:     2907.0M     11254112      4.2BSD      2048 16384      1 # /usr
g:       806.1M     17207616      4.2BSD      2048 16384      1 # /usr/X11R6
h:     3134.5M     18858496      4.2BSD      2048 16384      1 # /usr/local
i:     2329.4M     25277920      4.2BSD      2048 16384      1 # /usr/src
j:     5682.8M     30048512      4.2BSD      2048 16384      1 # /usr/obj
k:     5245.1M     41686880      4.2BSD      2048 16384      1 # /home
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a]

```

Une fois que vous avez validé vos paramètres, vous pouvez voir que la machine demande la même chose qu'au début. Pour **éviter cette boucle** d'installation, vous devez éteindre la machine et **retirer l'ISO** depuis VirtualBox > Configuration > Stockage, puis cliquez droit sur l'ISO pour le supprimer.



La machine est désormais configurée. Nous allons effectuer quelques réglages pour la rendre plus simple à utiliser. L'une des premières étapes consiste à activer **SSH**, ce qui nous permettra de prendre le contrôle de la machine à distance et de réaliser les opérations nécessaires au projet.

Partie 3- Utiliser SSH

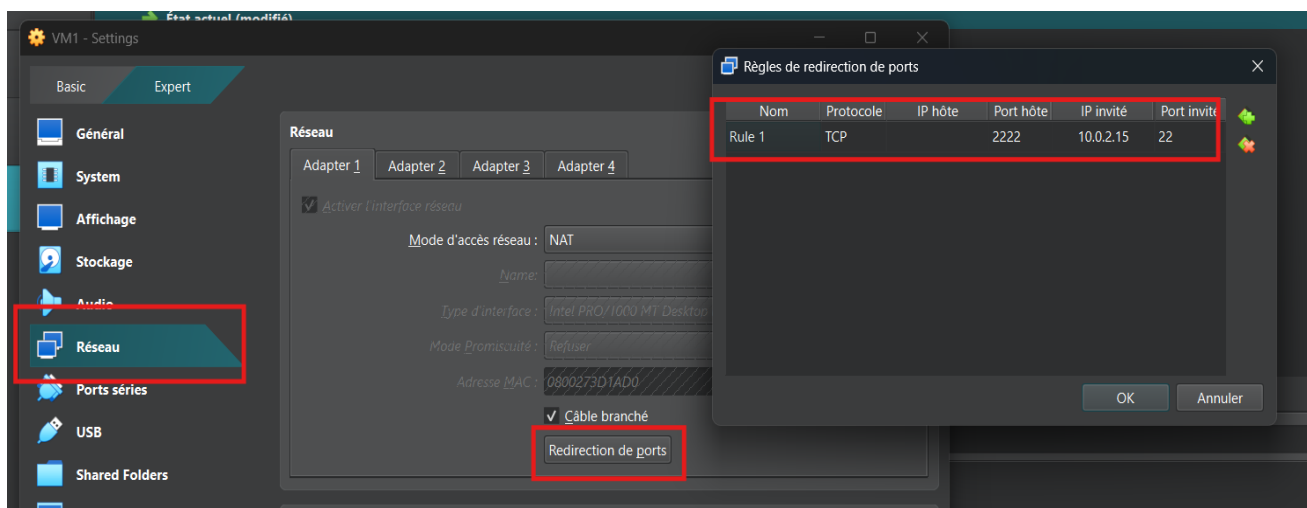


Qu'est-ce que le SSH ?

Le SSH (Secure Shell) est un protocole réseau qui permet d'établir une connexion sécurisée entre deux ordinateurs, même sur un réseau non sécurisé comme Internet. Il est utilisé pour se connecter à distance à une machine, exécuter des commandes et transférer des fichiers de manière protégée.

Comment le configurer ?

Sur VirtualBox, faites un clic droit sur la VM > Configuration > Réseau > Redirection de port et appliquez les règles comme suit :



Explications

Protocole TCP

Cette règle utilise le protocole TCP. Cela signifie que seules les communications TCP (et non UDP) seront redirigées.

TCP garantit une transmission sans faille des données, même si les paquets perdus ou

endommagés sont retransmis. UDP est un protocole « tire et oublie » qui ne vérifie pas les erreurs et ne renvoie pas les paquets de données perdus. UDP est plus adapté à la diffusion et au streaming en direct.

IP hôte

Cette colonne est vide car la redirection est configurée pour toutes les adresses IP de la machine hôte (localhost = 127.0.0.1). Cela permet à tout périphérique se connectant au port spécifié de l'hôte d'accéder à la machine virtuelle.

Port hôte 2222

C'est le port sur lequel la machine hôte écoute pour rediriger le trafic vers la machine virtuelle. Par exemple, si vous vous connectez à localhost:2222 sur l'hôte, cette connexion sera redirigée vers notre VM OpenBSD.

IP invité 10.0.2.15

C'est l'adresse IP de la machine virtuelle sur laquelle le trafic sera envoyé.

Port 22

C'est le port de la machine virtuelle vers lequel le trafic est redirigé. Ici, il s'agit du port 22, utilisé pour le SSH.

Enregistrez vos paramètres, démarrez votre machine sur VirtualBox et connectez-vous avec le mot de passe que vous avez attribué lors de l'installation de votre machine.

Ensuite appliquez la commande suivante pour modifier le fichier de configuration du ssh :



```
VM1 (sysctl) [En fonction] - Oracle VirtualBox
saproject# nano /etc/ssh/sshd_config _
```

Nano	Outil	Editeur de texte en ligne de commande simple et facile à utiliser.
/etc	Répertoire	Répertoire système sur les systèmes d'exploitation de type Unix qui contient les fichiers de configuration du système et des applications.
/ssh	Répertoire	Sous-répertoire utilisé pour stocker des fichiers liés à la configuration et au fonctionnement du service SSH, comme les clés, les hôtes connus ou les fichiers de configuration. Il est souvent présent sous /etc ou d'autres chemins spécifiques
/sshd_config	Fichier	Fichier principal de configuration du service SSH. Il définit les paramètres permettant de contrôler le comportement du serveur SSH, comme les options d'authentification, les ports utilisés, ou les protocoles autorisés.

Maintenant, vous pouvez éditer le fichier /sshd_config

```
GNU nano 8.1 /etc/ssh/sshd_config
# $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Voici les objectifs pour activer le SSH :

- Autoriser la connexion par root
- Autoriser la connexion avec mot de passe
- Autoriser la connexion par clé publique (facultatif si l'autorisation par mot de passe est activée)

Important : Pensez à bien retirer le commentaire en supprimant #.

Autoriser la connexion par root :

Raison : Avoir tous les droits

#PermitRootLogin no -> PermitRootLogin yes

Autoriser la connexion avec mot de passe :

Raison : Les utilisateurs peuvent s'authentifier en utilisant un mot de passe.

#PasswordAuthentication no -> PasswordAuthentication yes

Autoriser la connexion par clé publique :

Raison : Les utilisateurs peuvent s'authentifier en utilisant une clé publique au lieu d'un mot de passe, ce qui améliore la sécurité des connexions SSH.

#PubkeyAuthentication no -> PubkeyAuthentication yes

Test

Ouvrez un terminal > tapez la commande suivante :

ssh utilisateur@hote -p port

En fonction de notre paramètre sur VirtualBox :

```
C:\Users\erenv>ssh root@127.0.0.1 -p 2222|
```

Bravo ! Le ssh fonctionne maintenant, vous pouvez le voir qu'on a nsaproject (notre machine).

```
C:\Users\erenv>ssh root@127.0.0.1 -p 2222
root@127.0.0.1's password:
Last login: Fri Dec 13 16:15:07 2024
OpenBSD 7.6 (GENERIC.MP) #338: Mon Sep 30 08:55:35 MDT 2024

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code. With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

nsaproject#
```

Partie 4 – Configuration du DHCP.



Maintenant que notre SSH est prêt, nous pouvons configurer les fichiers nécessaires pour les fonctions demandées. Avant toute chose, vérifions que nous avons bien nos 4 interfaces réseaux avec la commande suivante : `ifconfig` (ou `ip a` sur Debian). Comme vous pouvez le voir, nous avons bien nos 4 interfaces réseaux :

- em0 -> interface réseau (NAT)
- em1 -> interface **administrateur**
- em2 -> interface **server**
- em3 -> interface **employeee**

```
lo0: flags=2008049<UP,LOOPBACK,RUNNING,MULTICAST,LR0> mtu 32768
    index 6 priority 0 llprio 3
    groups: lo
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x6
    inet 127.0.0.1 netmask 0xff000000
em0: flags=808843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,AUTOCONF4> mtu 1500
    lladdr 08:00:27:3d:1a:d0
    index 1 priority 0 llprio 3
    groups: egress
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 10.0.2.15 netmask 0xfffff00 broadcast 10.0.2.255
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:a6:88:82
    index 2 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.1 netmask 0xfffff00 broadcast 192.168.42.63
em2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:5f:e3:28
    index 3 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.65 netmask 0xfffff00 broadcast 192.168.42.127
em3: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:ef:b1:5f
    index 4 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.129 netmask 0xfffff00 broadcast 192.168.42.191
enc0: flags=0<>
    index 5 priority 0 llprio 3
    groups: enc
    status: active
pflog0: flags=141<UP,RUNNING,PROMISC> mtu 33136
    index 7 priority 0 llprio 3
    groups: pflog
```

C'est quoi le DHCP ?

Dynamic Host Configuration Protocol (DHCP, **protocole de configuration dynamique** des hôtes) est un protocole **réseau** dont le rôle est d'assurer la **configuration automatique** des paramètres IP d'une station ou d'une machine, notamment en lui **attribuant automatiquement** une adresse **IP** et un **masque de sous-réseau**.

Consignes :

Souvenez-vous, nous avons eu pour consigne d'avoir 3 sous-réseaux : **administration**, **server**, et **employee**. Ces 3 sous-réseaux vont être configurés en DHCP. Nous avons pour consigne de mettre en place les DHCP suivants :

Create 3 lan with the following configurations:

✓ **lan-1:** **administration**

- network : 192.168.42.0
- broadcast : 192.168.42.63
- range DHCP : 192.168.42.40 - 192.168.42.60

✓ **lan-2** **server**

- network : 192.168.42.64
- range DHCP : 192.168.42.70 - 192.168.42.110
- broadcast : 192.168.42.127

✓ **lan-3:** **employee**

- network : 192.168.42.128
- range DHCP : 192.168.42.140 - 192.168.42.180
- broadcast : 192.168.42.191

Le DHCP se configure comme ceci :

```
subnet Réseau netmask NetMaskACalculer {
    option routers Router;
    range Intervalle DHCP;
    option broadcast-address le Broadcast;
}
```

Calcul du Netmask**Données :**

- Network : 192.168.42.0
- Broadcast : 192.168.42.63
- Range DHCP : 192.168.42.40 - 192.168.42.60

La plage d'adresses va de 192.168.42.0 à 192.168.42.63, ce qui donne :
64 adresses (de 0 à 63) = 2^{62} adresses.

- Le réseau utilise 6 bits pour les hôtes, donc $32-6=26$ bits pour le préfixe réseau.
- CIDR : /26. (méthode de notation des adresses IP qui permet une gestion plus flexible et efficace des sous-réseaux en utilisant un préfixe, comme dans 192.168.1.0/24, au lieu de la notation basée sur les classes d'adresses.)

Netmask = 255.255.255.192

Donc la LAN 1 (Idem pour la LAN3, on changera juste les valeurs) va être configuré comme

Adresse IP Réseau → subnet 192.168.42.0

Netmask calculé → netmask 255.255.255.192

Intervalle d'attribution → range 192.168.42.40 192.168.42.60

Spécifie une liste d'adresses IP de routeurs qui sont sur le sous-réseau du client. → option routers 192.168.42.1

Adresse réseau utilisée pour transmettre à tous les appareils connectés à un réseau de communication à accès multiple. Un message envoyé à une adresse de diffusion peut être reçu par tous les hôtes connectés au réseau. → option broadcast-address 192.168.42.63

(il n'y aura pas de 192.168.42.61 et + ou 192.168.42.39 et moins pour le réseau administrateur)

ceci :

La LAN **serveur** (LAN 2) va être différente des deux autres. En effet, nous avons pour consigne de lui attribuer une adresse IP **statique** (qui ne sera pas donnée automatiquement lors de la distribution du DHCP) : l'adresse IP sera **192.168.42.70**.

Adresse MAC de la machine, pour pouvoir l'identifier et attribuer l'IP Statique → hardware ethernet 08:00:27:4C:67:3C

(Vous pouvez le trouver dans le menu Réseau sur VirtualBox)

IP Statique → fixed-address 192.168.42.70

Une fois le DHCP configuré, on peut l'activer avec les commandes suivantes :

Démarrer → nsaproject# rcctl enable dhcpd

Activer → nsaproject# rcctl start dhcpd

Vérifier → nsaproject# rcctl check dhcpd

Status → dhcpd(ok)

L'équivalent de cette commande sur Debian pourrait être :

```
systemctl start isc-dhcp-server
systemctl (qui remplace rcctl) start (qui remplace enable et start) isc-dhcp-server (dhcpd)
systemctl status isc-dhcp-server
```

Pour finir, il faudrait mettre net.inet.ip.forwarding sur 1 pour autoriser la transmission de paquets entre nos interfaces réseaux. En effet, cette variable contrôle si le noyau de l'OS

peut transférer des paquets d'une interface réseau à une autre. On le fait ici car notre machine agit comme un routeur.

Pour activer ceci, voici la commande à taper :

```
nsaproject# sysctl net.inet.ip.forwarding=1
net.inet.ip.forwarding: 0 -> 1
```

Bravo, notre DHCP est prêt maintenant. Mais il reste une dernière chose à faire, configurer le paquet filter (pf).

Utilisation de netstat

La commande netstat permet de lister les ports ouverts et les interfaces qui écoutent les connexions. Pour afficher spécifiquement les ports sur lesquels dhcpcd (le serveur DHCP) écoute :

```
nsaproject# netstat -an | grep 67
tcp      0      0  10.0.2.15.22          10.0.2.2.28670        ESTABLISHED
udp      0      0  10.0.2.15.2678        95.179.212.126.123
udp      0      0  10.0.2.15.6744        162.159.200.1.123
you have mail in /var/mail/root
```

-a : Affiche toutes les connexions et les ports d'écoute.

-n : Affiche les adresses et ports sous forme numérique.

grep 67 : Filtre pour ne montrer que les lignes contenant le port 67, qui est le port utilisé par **DHCP serveur** (serveur écoute sur le port 67 en UDP).

Partie 5 – Configuration du Paquet Filter.



Voyons les paquets filtrés comme un night-club : tout le monde ne peut pas entrer. Des restrictions sont imposées, par exemple sur les ports, les transmissions ou l'attribution du réseau aux interfaces.

Voici les consignes qu'on devra appliquer :

- Tous les sous-réseaux doivent pouvoir communiquer entre eux par l'intermédiaire de la passerelle.
- Le réseau local de l'**administration** peut accéder à n'importe quel **serveur** via le réseau de **serveurs**, sur tous les ports.
- Le réseau local des **employés** ne peut atteindre que le **serveur** sur les protocoles HTTP et HTTPS.
- Les réseaux locaux d'**employés**, d'**administration** et de **serveurs** peuvent se connecter à Internet, envoyer des signaux ping à des appareils situés dans un autre sous-réseau et récupérer des informations DHCP et DNS auprès de la passerelle.

Rappel des interfaces réseaux :

```

lo0: flags=2008049<UP,LOOPBACK,RUNNING,MULTICAST,LR0> mtu 32768
    index 6 priority 0 llprio 3
    groups: lo
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x6
    inet 127.0.0.1 netmask 0xff000000
em0: flags=808843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,AUTOCONF4> mtu 1500
    lladdr 08:00:27:3d:1a:d0
    index 1 priority 0 llprio 3
    groups: egress
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 10.0.2.15 netmask 0xffffffff0 broadcast 10.0.2.255
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:a6:88:82
    index 2 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.1 netmask 0xfffffc0 broadcast 192.168.42.63
em2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:5f:e3:28
    index 3 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.65 netmask 0xfffffc0 broadcast 192.168.42.127
em3: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 08:00:27:ef:b1:5f
    index 4 priority 0 llprio 3
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.42.129 netmask 0xfffffc0 broadcast 192.168.42.191
  
```

*Un rebouclage, loopback ou loop-back (de l'anglais signifiant "boucle arrière") est un système matériel ou logiciel en informatique, réseaux ou télécommunications, destiné à renvoyer un signal reçu vers son envoyeur sans modification ni traitement, et qui peut par exemple être utilisé à des fins de tests.

Comme pour le DHCP, il y a aussi un fichier pour configurer le packet filter, ce fichier est **pf.conf**.

Utilisons cat pour afficher notre fichier **pf.conf** :

```

nsaproject# cat /etc/pf.conf
# Interfaces
ext_if = "em0"
admin_if = "em1"
server_if = "em2"
emp_if = "em3"

# Subnet
admin_net = "192.168.42.0/26"
server_net = "192.168.42.64/26"
emp_net = "192.168.42.128/26"

set skip on lo
block return
pass
block return in on ! lo0 proto tcp to port 6000:6010

# Internet access to all networks
match out on $ext_if from { $admin_net, $server_net, $emp_net } to any nat-to ($ext_if)

# 1 - Administration LAN can reach any server through the server network, on all ports.
pass in on $admin_if from $admin_net to $server_net keep state
pass out on $admin_if from $admin_net to $server_net keep state

# 2 - Employee LAN can reach only reach the server on HTTP and HTTPS protocols.
pass in on $emp_if proto tcp from $emp_net to $server_net port { 80, 443 } keep state
pass out on $emp_if proto tcp from $emp_net to $server_net port { 80, 443 } keep state

# 3 - Employee, administration and server LANs can go out on the internet,
# ping devices on another subnet,
# and retrieve DHCP and DNS information from the gateway

# DHCP and DNS rules
pass in on $admin_if proto udp from $admin_net to any port { 67, 68, 53 } keep state
pass in on $server_if proto udp from $server_net to any port { 67, 68, 53 } keep state
pass in on $emp_if proto udp from $emp_net to any port { 67, 68, 53 } keep state

# Ping between devices
pass in on { $admin_if, $server_if, $emp_if } proto icmp from { $admin_net, $server_net, $emp_net } to any keep state
pass out on { $admin_if, $server_if, $emp_if } proto icmp from { $admin_net, $server_net, $emp_net } to any keep state

# Inter network communication - with the gateway
# Admin with server and employee
pass in on $admin_if from $admin_net to { $server_net, $emp_net } keep state
pass out on $admin_if from $admin_net to { $server_net, $emp_net } keep state

# Server with admin and employee
pass in on $server_if from $server_net to { $admin_net, $emp_net } keep state
pass out on $server_if from $server_net to { $admin_net, $emp_net } keep state

# Employee with admin and server
pass in on $emp_if from $emp_net to { $admin_net, $server_net } keep state
pass out on $emp_if from $emp_net to { $admin_net, $server_net } keep state

```

Décomposons maintenant le script.

Déclarations des variables d'interfaces

L'interface réseau NAT est em0

L'interface **admin** est em1

L'interface **server** est em2

L'interface **employee** est em3

```

# Interfaces
ext_if = "em0"
admin_if = "em1"
server_if = "em2"
emp_if = "em3"

```

Même chose pour la déclaration des variables pour les sous réseaux

Sous réseau **admin** : 192.168.42.0/26

Sous réseau **server** : 192.168.42.64/26

Sous réseau **employee** : 192.168.42.128/26

```

# Subnet
admin_net = "192.168.42.0/26"
server_net = "192.168.42.64/26"
emp_net = "192.168.42.128/26"

```

Maintenant que les variables sont prêtes, on peut attribuer les règles qui ont été données précédemment.

```
set skip on lo
block return
pass
block return in on ! lo0 proto tcp to port 6000:6010
```

- **Set skip on lo** -> Indique au pare-feu de ne pas filtrer les paquets sur l'interface locale (correspond à lo0 dans le ifconfig). Cela est souvent utilisé pour améliorer les performances, car il est inutile de filtrer le trafic local.
- **Block return** -> Tous les paquets sont bloqués par défaut, et donc un message de retour est envoyé à l'expéditeur pour lui signaler que le paquet a été rejeté. Cela garantit qu'aucun paquet non autorisé ne passe (comme un physio qui rejette une personne complètement ivre).
- **Pass** -> Ce simple mot permet de laisser passer tout le trafic (entrant et sortant) par défaut. Les règles seront mises en place plus tard.
- **Block return in on ! lo0 proto tcp to port 6000:6010** -> Bloque le trafic entrant sur toutes les interfaces sauf lo0 (!lo0) à destination des ports TCP 6000 à 6010. Cela permet de bloquer les connexions qui pourraient exploiter des vulnérabilités X11*.

* Faille de sécurité dans le système graphique X11 qui peut permettre à un attaquant de lire, injecter ou manipuler des données graphiques, voire d'exécuter du code malveillant, souvent à cause d'une absence d'authentification ou d'un accès réseau mal sécurisé.

Donner du réseau Internet à tous les réseaux, il est important de remarquer qu'on fait appel aux variables pour appliquer les règles (exemple : `$admin_net` pour le réseau `admin`).

```
# Internet access to all networks
match out on $ext_if from { $admin_net, $server_net, $emp_net } to any nat-to ($ext_if)
```

Match out -> S'applique uniquement au trafic sortant sur l'interface réseau externe (`ext_if`, qui est la variable de `em0`).

To any -> La règle s'applique pour des paquets à destination de n'importe quelle adresse IP.

Nat-to -> Masque les adresses internes (comme celles des réseaux locaux spécifiés) derrière une adresse publique, rendant les machines internes invisibles pour les hôtes externes.

Le réseau local de l'`administration` peut accéder à n'importe quel `serveur` via le réseau de `serveurs`, sur tous les ports :

```
# 1 - Administration LAN can reach any server through the server net
pass in on $admin_if from $admin_net to $server_net keep state
pass out on $admin_if from $admin_net to $server_net keep state
```

Pass in -> Permet le trafic entrant sur l'interface spécifiée, comme em1 pour cette situation (même chose pour les autres interfaces pour la suite).

Pass out -> Permet le trafic sortant (même chose pour les lignes suivantes).

to \$server_net -> Le trafic doit être destiné au réseau des serveurs, défini par **\$server_net**.

keep state -> PF suit l'état des connexions pour permettre les réponses correspondantes sans avoir à écrire une règle explicite pour le trafic retour.

Le réseau local des employés ne peut atteindre que le serveur sur les protocoles HTTP et HTTPS :

```
# 2 - Employee LAN can reach only reach the server on HTTP and HTTPS protocols.
pass in on $emp_if proto tcp from $emp_net to $server_net port { 80, 443 } keep state
pass out on $emp_if proto tcp from $emp_net to $server_net port { 80, 443 } keep state
```

proto tcp -> Limite la règle aux connexions utilisant le protocole TCP (nécessaire pour HTTP/HTTPS).

from \$emp_net -> Le trafic doit provenir du réseau des employés (\$emp_net).

to \$server_net -> Le trafic doit être destiné au réseau des serveurs (\$server_net).

port { 80, 443 } -> Restreint l'accès aux ports 80 et 443 (HTTP et HTTPS).

Les réseaux locaux des employés, de l'administration et des serveurs peuvent se connecter à Internet, envoyer des signaux ping à des appareils situés dans un autre sous-réseau et récupérer des informations DHCP et DNS auprès de la passerelle :

```
# 3 - Employee, administration and server LANs can go out on the internet,
# ping devices on another subnet,
# and retrieve DHCP and DNS information from the gateway

# DHCP and DNS rules
pass in on $admin_if proto udp from $admin_net to any port { 67, 68, 53 } keep state
pass in on $server_if proto udp from $server_net to any port { 67, 68, 53 } keep state
pass in on $emp_if proto udp from $emp_net to any port { 67, 68, 53 } keep state

# Ping between devices
pass in on { $admin_if, $server_if, $emp_if } proto icmp from { $admin_net, $server_net, $emp_net } to any keep state
pass out on { $admin_if, $server_if, $emp_if } proto icmp from { $admin_net, $server_net, $emp_net } to any keep state
```

proto udp -> Le trafic doit utiliser le protocole UDP (le mieux pour DHCP et DNS).

port { 67, 68, 53 } :

- **67/68** : Ports utilisés pour les requêtes et réponses DHCP.
- **53** : Port utilisé pour le service DNS.
- **proto icmp** -> Autorise le protocole ICMP, utilisé par les commandes de diagnostic, ici **ping**.

Tous les sous-réseaux doivent pouvoir communiquer entre eux par l'intermédiaire de la passerelle :

```
# Inter network communication - with the gateway
# Admin with server and employee
pass in on $admin_if from $admin_net to { $server_net, $emp_net } keep state
pass out on $admin_if from $admin_net to { $server_net, $emp_net } keep state

# Server with admin and employee
pass in on $server_if from $server_net to { $admin_net, $emp_net } keep state
pass out on $server_if from $server_net to { $admin_net, $emp_net } keep state

# Employee with admin and server
pass in on $emp_if from $emp_net to { $admin_net, $server_net } keep state
pass out on $emp_if from $emp_net to { $admin_net, $server_net } keep state
```

Admin à serveur et employé - Serveur à admin et employé - Employé à admin et serveur

Une fois que les règles ont été configurées, on pourra les vérifier avec cette syntaxe :

pfctl -nf /etc/pf.conf

n -> analyse uniquement (ne charge pas les règles)*

f -> spécifie le fichier à analyser ici /etc/pf.conf

Si la syntaxe est correcte, la commande ne renvoie aucune erreur :

```
nsaproject# pfctl -nf /etc/pf.conf
nsaproject# |
```

Ensuite, activer pf.conf avec :

pfctl -e (censé être déjà actif par défaut)

e -> activer (enable, donc -d pour disable)

Charger maintenant les règles avec :

pfctl -f /etc/pf.conf

Vérifier l'état du pf :

pfctl -sr

-s -> Affiche l'état ou la configuration.

-r -> Liste toutes les règles de filtrage actuellement en vigueur.


```

nsaproject# pfctl -nt /etc/pf.conf
nsaproject# pfctl -sr
block return all
pass all flags S/SA
block return in on ! lo0 proto tcp from any to any port 6000:6010
match out on em0 inet from 192.168.42.0/26 to any nat-to (em0) round-robin
match out on em0 inet from 192.168.42.64/26 to any nat-to (em0) round-robin
match out on em0 inet from 192.168.42.128/26 to any nat-to (em0) round-robin
pass in on em1 inet proto udp from 192.168.42.0/26 to any port = 67
pass in on em1 inet proto udp from 192.168.42.0/26 to any port = 68
pass in on em1 inet proto udp from 192.168.42.0/26 to any port = 53
pass in on em2 inet proto udp from 192.168.42.64/26 to any port = 67
pass in on em2 inet proto udp from 192.168.42.64/26 to any port = 68
pass in on em2 inet proto udp from 192.168.42.64/26 to any port = 53
pass in on em3 inet proto udp from 192.168.42.128/26 to any port = 67
pass in on em3 inet proto udp from 192.168.42.128/26 to any port = 68
pass in on em3 inet proto udp from 192.168.42.128/26 to any port = 53
pass in on em1 inet proto icmp from 192.168.42.0/26 to any
pass in on em1 inet proto icmp from 192.168.42.64/26 to any
pass in on em1 inet proto icmp from 192.168.42.128/26 to any
pass in on em2 inet proto icmp from 192.168.42.0/26 to any
pass in on em2 inet proto icmp from 192.168.42.64/26 to any
pass in on em2 inet proto icmp from 192.168.42.128/26 to any
pass in on em3 inet proto icmp from 192.168.42.0/26 to any
pass in on em3 inet proto icmp from 192.168.42.64/26 to any
pass in on em3 inet proto icmp from 192.168.42.128/26 to any
pass in on em3 inet from 192.168.42.128/26 to 192.168.42.0/26 flags S/SA
pass in on em3 inet from 192.168.42.128/26 to 192.168.42.64/26 flags S/SA
pass in on em2 inet from 192.168.42.64/26 to 192.168.42.0/26 flags S/SA
pass in on em2 inet from 192.168.42.64/26 to 192.168.42.128/26 flags S/SA
pass in on em1 inet from 192.168.42.0/26 to 192.168.42.64/26 flags S/SA
pass in on em1 inet from 192.168.42.0/26 to 192.168.42.128/26 flags S/SA
pass out on em1 inet proto icmp from 192.168.42.0/26 to any
pass out on em1 inet proto icmp from 192.168.42.64/26 to any
pass out on em1 inet proto icmp from 192.168.42.128/26 to any
pass out on em2 inet proto icmp from 192.168.42.0/26 to any
pass out on em2 inet proto icmp from 192.168.42.64/26 to any
pass out on em2 inet proto icmp from 192.168.42.128/26 to any
pass out on em3 inet proto icmp from 192.168.42.0/26 to any
pass out on em3 inet proto icmp from 192.168.42.64/26 to any
pass out on em3 inet proto icmp from 192.168.42.128/26 to any
pass out on em3 inet from 192.168.42.128/26 to 192.168.42.0/26 flags S/SA
pass out on em3 inet from 192.168.42.128/26 to 192.168.42.64/26 flags S/SA
pass out on em2 inet from 192.168.42.64/26 to 192.168.42.0/26 flags S/SA
pass out on em2 inet from 192.168.42.64/26 to 192.168.42.128/26 flags S/SA
pass out on em1 inet from 192.168.42.0/26 to 192.168.42.64/26 flags S/SA
pass out on em1 inet from 192.168.42.0/26 to 192.168.42.128/26 flags S/SA
nsaproject#

```

Et voilà ! La machine 1 est complètement réglée. Nous avons vu ici comment configurer le DHCP, le PF, et comment activer SSH. Pour l'installation des machines suivantes, la procédure est exactement la même, la VM 2 va demander des outils comme MySql et PHP.

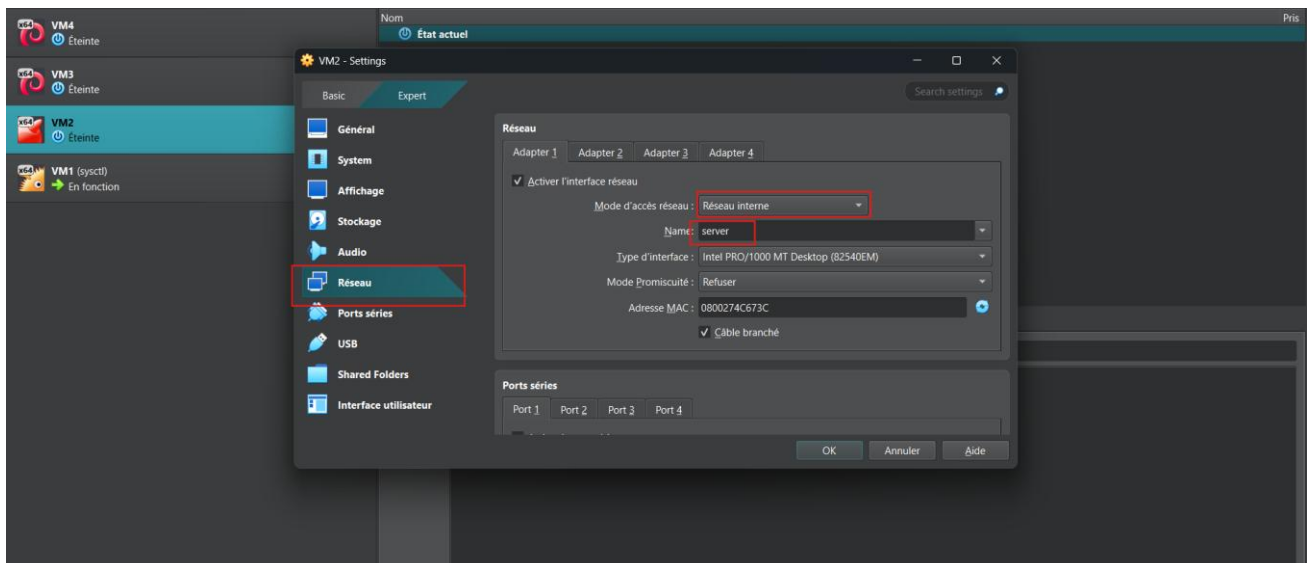
IMPORTANT :

Créons les 4 machines sur VirtualBox, attribuons les tailles de mémoire et stockage nécessaire comme on l'a fait au début. Ensuite, allez dans *Configuration > Réseau > Sélectionnez Réseau Interne* et :

Pour la VM 2 attribuez le nom de réseau « server »

Pour la VM 3 attribuez le nom de réseau « administration »

Pour la VM 4 attribuez le nom de réseau « employee »

Exemple VM2 :

VM2 – installation

Tout d'abord, récupérer le fichier iso de FreeBSD. Une fois cela effectué, l'installation de FreeBSD peut débuter.

Installation de FreeBSD

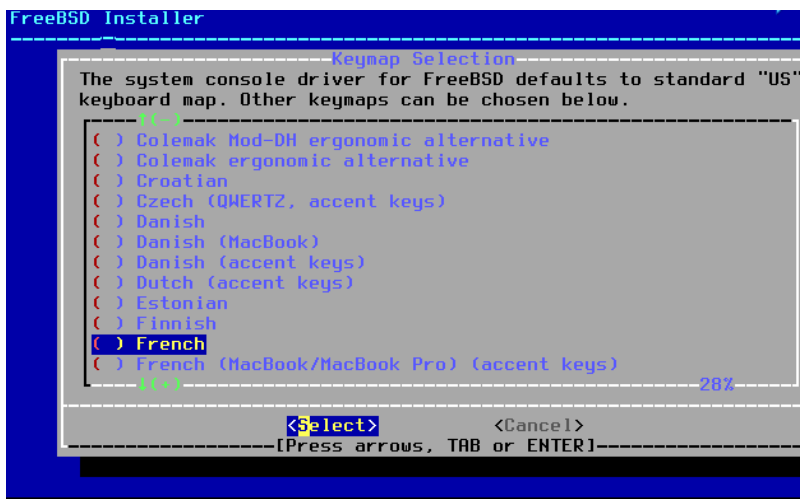
Une fois la VM lancée, cliquer sur la touche entrée afin de procéder à l'installation



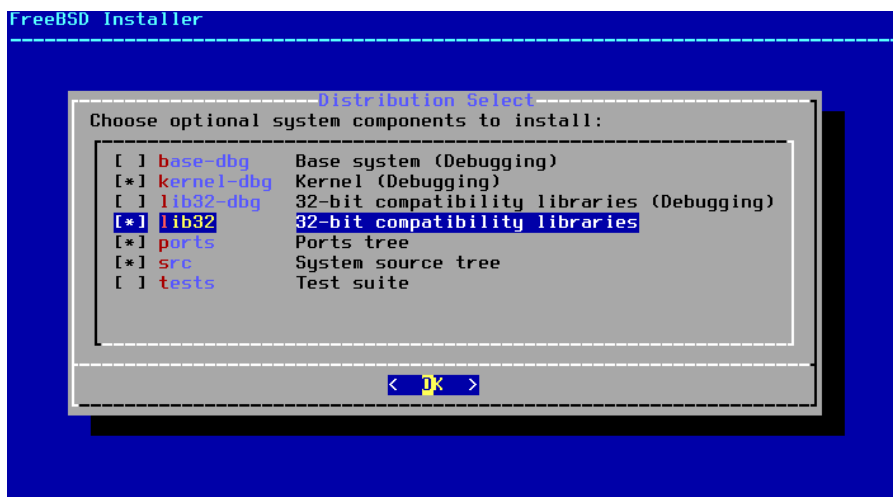
Cliquer sur "Install" pour continuer l'installation



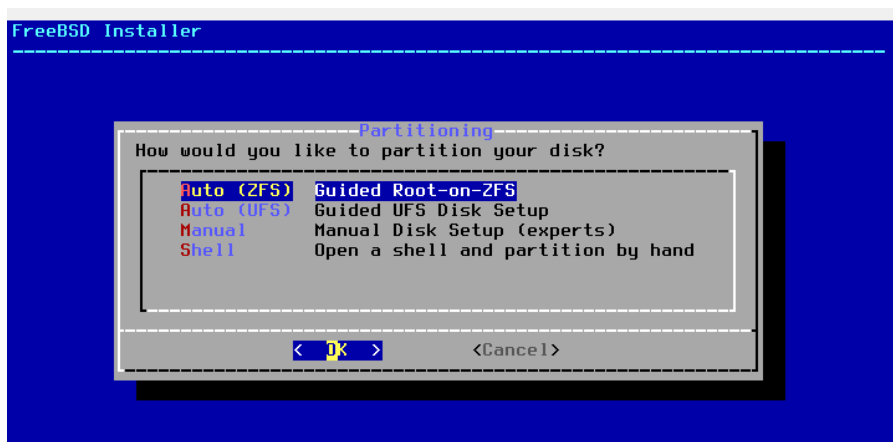
Choisir la langue du clavier:



Sélectionner les distributions nécessaires:



Allouer ensuite la taille de chaque partitions du disque comme le /var ou le / par exemple. Cela permet d'organiser les données de manière optimale.



Après l'installation, nous pouvons voir que l'adresse IP statique **192.168.42.70** a bien été attribuée à la VM2 (server).

```
root@nsaproject:~ # ifconfig
em0: flags=1008843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST,LOWER_UP> metric 0 mtu
1500
    options=48505bb<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, JUMBO_MTU, VLAN_HWC
SUM, TS04, LRO, VLAN_HWFILTER, VLAN_HWTSO, HWSTATS, MEXTPG>
    ether 08:00:27:4c:67:3c
    inet 192.168.42.70 netmask 0xfffffc0 broadcast 192.168.42.127
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD, IFDISABLED, AUTO_LINKLOCAL>
lo0: flags=1008049<UP,LOOPBACK,RUNNING,MULTICAST,LOWER_UP> metric 0 mtu 16384
    options=680003<RXCSUM, TXCSUM, LINKSTATE, RXCSUM_IPV6, TXCSUM_IPV6>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
    groups: lo
    nd6 options=21<PERFORMNUD, AUTO_LINKLOCAL>
root@nsaproject:~ #
```

Le serveur web

Le serveur web utilisé est nginx. Nginx est un serveur web performant, open-source, qui joue un rôle important dans l'hébergement et la gestion du trafic sur des sites web. Initialement conçu pour être un serveur HTTP haute performance.

```
pkg install nginx
```

Activer le serveur web

```
sysrc nginx_enable="YES"
```

Configurer le serveur web

```
nano /usr/local/etc/nginx/nginx.conf
```

Vérifier les droits du fichier

```
ls -l /usr/local/www/nginx/index.html
```

Pour permettre l'accès à l'URL du serveur web, il est essentiel d'ouvrir le port 80. Le port 80 est le port par défaut pour les communications HTTP (HyperText Transfer Protocol), ce

qui permettra au serveur web d'écouter sur ce port est ainsi recevoir des requêtes HTTP depuis Internet ou un réseau local

La capture ci-dessous montre comment ouvrir le port 80 sur VirtualBox.

Rule 2	TCP		80	10.0.2.15	80
--------	-----	--	----	-----------	----

PHP

Installer php-7.4.33

```
pkg install openssl111
```

Pour récupérer le bon package

```
fetch https://www.php.net/distributions/php-7.4.33.tar.gz
```

unzip le dossier:

```
tar -xvf php-7.4.33.tar.gz
```

```
cd php-7.4.33
```

Pour installer les dépendances nécessaires:

```
./configure --prefix=/tmp/makeinstall --with-apxs2=/usr/local/sbin/apxs --with-libdir=lib64 --with-openssl --with-curl --with-mysqli --
```

```
with-pdo-mysql --enable-gd --with-curl --enable-mbstring
```

Pour compiler

```
gmake
```

```
gmake install clean
```

Installer les dépendances manquantes

```
pkg install graphics/png
```

```
nano ~/.profile
```

et rajouter cette ligne

```
export PATH=$PATH:/tmp/makeinstall/bin
```

MySQL

Pour Installer la dernière version de MySQL via le système de port, il faut tout d'abord se déplacer dans le dossier associé à mysql80-server. Une fois dans le dossier, il faut compiler les paquets de MySQL présents dans le système de port, comme le montre l'image ci-dessous.

```
root@nsaproject:~ # cd /usr/ports/databases/mysql80-server
root@nsaproject:/usr/ports/databases/mysql80-server # make
```

Créer le user backend

```
adduser
```

Pour activer le démarrage automatique du service MySQL au démarrage du système utilisé cette commande:

```
root@nsaproject:~ # sysrc mysql_enable="YES"
mysql_enable: YES -> YES
```

La configuration de MySQL s'effectue en plusieurs étapes:

La première étape consiste à se connecter à MySQL. La commande suivante permet de se connecter en tant qu'utilisateur backend via l'option -u, et demande un mot de passe avec l'option -p

```
mysql -u backend -p
```


Une fois connecté, créer une base de données qui a pour nom nsa501, à l'aide la commande:

```
CREATE DATABASE nsa501
```

Importer le fichier donnée dans la VM

Création du fichier nsa501.sql avec un éditeur de texte comme nano, et y coller les information contenue dans le fichier php fournie.

```
nano nsa501.sql
```

Ensuite pour importer ce fichier dans la base de données précédemment créée, effectuer cette commande:

```
mysql -u backend -p nsa501 < nsa501.sql
```

Donner tous les droits à l'utilisateur backend sur la base de données

```
GRANT ALL PRIVILEGES ON nsa501.* TO 'backend'@'%';
```

```
FLUSH PRIVILEGES;
```

Afin de vérifier que l'utilisateur backend possède bien tous les droits sur la base de donnée, effectuer cette commande:

```
SHOW GRANTS FOR 'backend'@'%';
```

vérifier que la base de données est fonctionnelle

```
mysql -u backend -pBit8Q6a6G -e "show databases"
```

Test avec la commande wget localhost :80 (fonctionnel)

```
root@nsaproject:~ # wget localhost:80
Prepended http:// to 'localhost:80'
--2024-12-15 04:11:10-- http://localhost/
Resolving localhost (localhost)... 127.0.0.1, ::1
Connecting to localhost (localhost)|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 615 [text/html]
Saving to: `index.html.9'

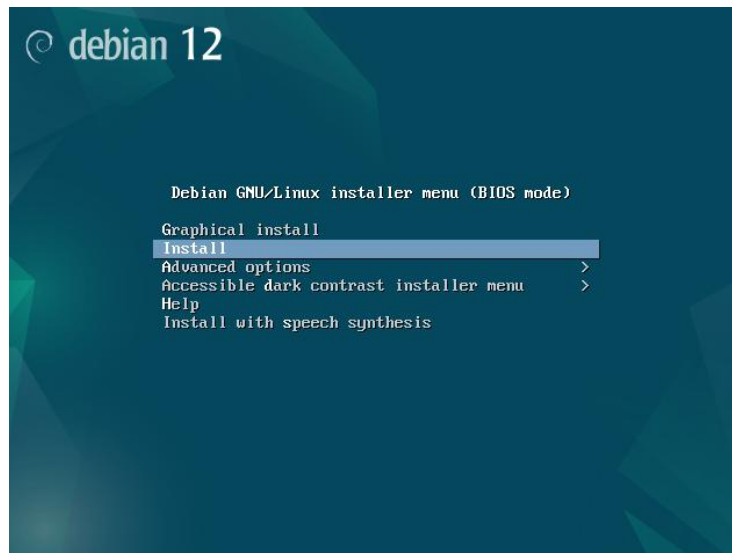
index.html.9      100%[=====>]      615  --.-KB/s   in 0s
2024-12-15 04:11:10 (16.1 MB/s) - `index.html.9' saved [615/615]
```

VM3 et 4 – installation

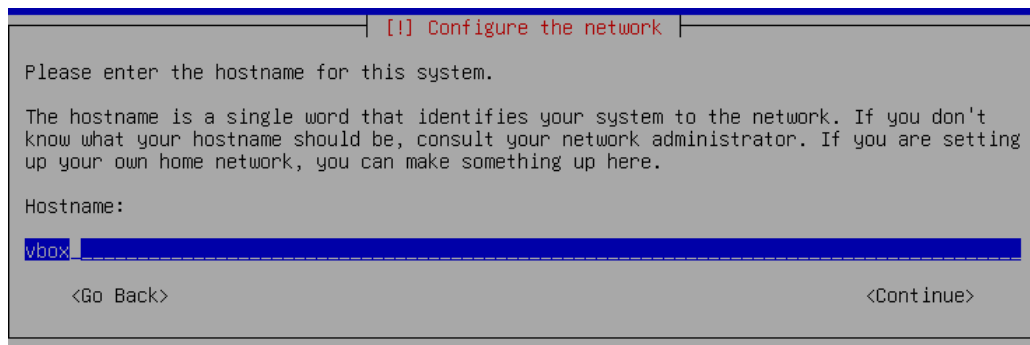
Installer debian 12:

Tout d'abord, récupérer le fichier iso de debian 12. Une fois cela effectué, l'installation de debian peut débuter.

Ensuite, cliquer sur install afin de démarrer l'installation:



Sélectionner ensuite le nom du système sur le réseau:



Définir un mot de passe pour le compte administrateur:

Création du compte utilisateur et de son mot de passe afin de pouvoir utiliser l'interface graphique de Debian.

```

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for
non-administrative activities.

Please enter the real name of this user. This information will be used for instance as
default origin for emails sent by this user as well as any program which displays or uses
the user's real name. Your full name is a reasonable choice.

Full name for the new user:
martin

```

```
[!!] Set up users and passwords
```

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:

[] Show Password in Clear

<Go Back> <Continue>

Allouez ensuite la taille de chaque partition du disque, comme le /home ou le / par exemple. Cela permet d'organiser les données de manière optimale.

```

[!!!] Partition disks

The installer can guide you through partitioning a disk (using different standard
schemes) or, if you prefer, you can do it manually. With guided partitioning you will
still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk
should be used.

Partitioning method:

Guided - use entire disk
Guided - use entire disk and set up LVM
Guided - use entire disk and set up encrypted LVM
Manual

<Go Back>

```

Vérifier que la machine possède l'IP généré par le DHCP de la VM. Pour effectuer cela, on utilise cette commande:

Sur cette image, on voit bien que l'IP est fournie dynamiquement grâce au mot-clé **dynamic**. Cela signifie que le serveur DHCP attribue automatiquement une adresse IP au poste, selon les plages définies dans la configuration.

Ensuite, pour vérifier que le poste a bien accès à Internet, vous pouvez effectuer un **ping** vers une adresse IP externe, comme le **DNS de Google (8.8.8.8)**. En exécutant la commande suivante :

ping 8.8.8.8

Cela enverra des paquets de test à l'adresse IP de Google, et si le réseau est correctement configuré, vous recevrez des réponses indiquant que le poste peut accéder à Internet. Si aucune réponse n'est reçue, cela pourrait indiquer un problème de connectivité, comme un mauvais routage ou une configuration incorrecte du DHCP.

TESTS DES MACHINES

TESTS : VM 1

Tests Réseau

Ping vers une adresse IP pour vérifier si le réseau est disponible.

```
nsaproject# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=255 time=17.583 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=255 time=17.544 ms
^C
```

Traceroute vers google.com

```
nsaproject# traceroute google.com
traceroute to google.com (172.217.20.174), 64 hops max, 40 byte packets
1 10.0.2.2 (10.0.2.2) 2.132 ms 2.353 ms 2.869 ms
^C
```

Tests Ping

Ping depuis la VM1 (gateway) à la VM2 (server).

```
nsaproject# ping 192.168.42.70
PING 192.168.42.70 (192.168.42.70): 56 data bytes
64 bytes from 192.168.42.70: icmp_seq=0 ttl=64 time=3.403 ms
64 bytes from 192.168.42.70: icmp_seq=1 ttl=64 time=3.643 ms
^C
```

Ping depuis la VM1 (gateway) à la VM3 (admin).

```
nsaproject# ping 192.168.42.140
PING 192.168.42.140 (192.168.42.140): 56 data bytes
64 bytes from 192.168.42.140: icmp_seq=0 ttl=64 time=1.957 ms
64 bytes from 192.168.42.140: icmp_seq=1 ttl=64 time=2.603 ms
^C
```

Ping depuis la VM1 (gateway) à la VM4 (employee).

```
nsaproject# ping 192.168.42.40
PING 192.168.42.40 (192.168.42.40): 56 data bytes
64 bytes from 192.168.42.40: icmp_seq=0 ttl=64 time=2.733 ms
```

TESTS : VM 2

Tests Réseau

Ping vers une adresse IP pour vérifier si le réseau est disponible.

```
root@nsaproject:~ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=254 time=21.196 ms
```

traceroute vers google.com

```
root@nsaproject:~ # traceroute google.com
traceroute to google.com (216.58.215.46), 64 hops max, 40 byte packets
 1  192.168.42.65 (192.168.42.65)  2.131 ms  3.020 ms  1.806 ms
 2  10.0.2.2 (10.0.2.2)  2.669 ms  2.941 ms  3.544 ms
^C
```

Tests Ping

Ping depuis la VM2 (server) à la VM1 (gateway).

```
root@nsaproject:~ # ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15): 56 data bytes
64 bytes from 10.0.2.15: icmp_seq=0 ttl=255 time=4.354 ms
64 bytes from 10.0.2.15: icmp_seq=1 ttl=255 time=1.584 ms
^C
```

Ping depuis la VM2 (server) à la VM3 (admin).

```
root@nsaproject:~ # ping 192.168.42.140
PING 192.168.42.140 (192.168.42.140): 56 data bytes
64 bytes from 192.168.42.140: icmp_seq=0 ttl=63 time=3.764 ms
64 bytes from 192.168.42.140: icmp_seq=1 ttl=63 time=3.615 ms
```

Ping depuis la VM2 (server) à la VM4 (employee).

```
root@nsaproject:~ # ping 192.168.42.40
PING 192.168.42.40 (192.168.42.40): 56 data bytes
64 bytes from 192.168.42.40: icmp_seq=0 ttl=63 time=4.738 ms
64 bytes from 192.168.42.40: icmp_seq=1 ttl=63 time=5.334 ms
```

TESTS : VM 3

Tests Réseau

Affichages des informations réseau avec la commande « ip a » (différent de ifconfig, en effet nous sommes sur debian)

```

martin@nsaproject: ~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:43:f2:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.140/26 brd 192.168.42.191 scope global dynamic noprefixroute enp0s3
        valid_lft 42337sec preferred_lft 42337sec
    inet6 fe80::a00:27ff:fe43:f23e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
martin@nsaproject: ~$

```

Ping 8.8.8.8 et google.com pour voir si nous avons du réseau

```

martin@nsaproject: ~$ 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=254 time=27.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=254 time=18.3 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 18.340/22.760/27.181/4.420 ms
martin@nsaproject: ~$ ping google.com
PING google.com (172.217.20.174) 56(84) bytes of data:
64 bytes from par10s49-in-f14.1e100.net (172.217.20.174): icmp_seq=1 ttl=254 time=17.4 ms
^C
--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 17.352/17.352/17.352/0.000 ms

```

Tests Ping

Ping depuis la VM3 (administration) à la VM1 (gateway).

```
martin@nsaproject:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=255 time=1.79 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=255 time=1.13 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=255 time=2.03 ms
```

Ping depuis la VM3 (administration) à la VM2 (server).

```
martin@nsaproject:~$ ping 192.168.42.70
PING 192.168.42.70 (192.168.42.70) 56(84) bytes of data.
64 bytes from 192.168.42.70: icmp_seq=1 ttl=63 time=4.03 ms
64 bytes from 192.168.42.70: icmp_seq=2 ttl=63 time=4.05 ms
```

Ping depuis la VM3 (administration) à la VM4 (employee).

```
martin@nsaproject:~$ ping 192.168.42.40
PING 192.168.42.40 (192.168.42.40) 56(84) bytes of data.
64 bytes from 192.168.42.40: icmp_seq=1 ttl=63 time=3.21 ms
64 bytes from 192.168.42.40: icmp_seq=2 ttl=63 time=2.84 ms
```

Se connecter à **ssh root@192.168.42.70** depuis la VM3 (administration)

```
martin@nsaproject:~$ ssh root@192.168.42.70
(root@192.168.42.70) Password for root@nsaproject:
Last login: Sun Dec 15 02:53:51 2024
FreeBSD 14.1-RELEASE (GENERIC) releng/14.1-n267679-10e31f0946d8

Welcome to FreeBSD!

Release Notes, Errata: https://www.FreeBSD.org/releases/
Security Advisories:  https://www.FreeBSD.org/security/
FreeBSD Handbook:     https://www.FreeBSD.org/handbook/
FreeBSD FAQ:          https://www.FreeBSD.org/faq/
Questions List:       https://www.FreeBSD.org/lists/questions/
FreeBSD Forums:       https://forums.FreeBSD.org/

Documents installed with the system are in the /usr/local/share/doc/freebsd/
directory, or can be installed later with: pkg install en-freebsd-doc
For other languages, replace "en" with a language code like de or fr.
```


TESTS : VM 4

Tests Réseau

Affichages des informations réseau avec la commande « ip a ». On peut bien voir que le DHCP s'est bien appliqué en respectant le range employées

```

eren@nsaproject: ~
eren@nsaproject:~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9a:db:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.40/26 brd 192.168.42.63 scope global dynamic noprefixroute enp0s3
        valid_lft 42888sec preferred_lft 42888sec
    inet6 fe80::a00:27ff:fe9a:db0f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Ping 8.8.8.8 et google.com pour voir si nous avons du réseau.

```

eren@nsaproject:~$ 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=254 time=18.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=254 time=22.6 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 18.075/20.342/22.610/2.267 ms
eren@nsaproject:~$ ping google.com
PING google.com (142.250.179.78) 56(84) bytes of data:
64 bytes from par21s19-in-f14.1e100.net (142.250.179.78): icmp_seq=1 ttl=254 time=17.8 ms
64 bytes from par21s19-in-f14.1e100.net (142.250.179.78): icmp_seq=2 ttl=254 time=19.4 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1058ms
rtt min/avg/max/mdev = 17.805/18.593/19.382/0.788 ms

```

Tests Ping

Ping depuis la VM4 (employee) à la VM1.

```

eren@nsaproject:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data:
64 bytes from 10.0.2.15: icmp_seq=1 ttl=255 time=1.54 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=255 time=1.39 ms

```

Ping depuis la VM4 (employee) à la VM2.

```











eren@nsaproject:~$ ping 192.168.42.70
PING 192.168.42.70 (192.168.42.70) 56(84) bytes of data:
64 bytes from 192.168.42.70: icmp_seq=1 ttl=63 time=4.74 ms
64 bytes from 192.168.42.70: icmp_seq=2 ttl=63 time=4.38 ms

```

Ping depuis la VM4 (employee) à la VM3.

```
eren@nsaproject:~$ ping 192.168.42.140
PING 192.168.42.140 (192.168.42.140) 56(84) bytes of data.
64 bytes from 192.168.42.140: icmp_seq=1 ttl=63 time=2.77 ms
```

Tableau de test.

VM	Ping vers toute les VM	Ping vers google.com	Configuré complètement
VM1			
VM2			
VM3			
VM4			

Toutes les VM sont fonctionnelles

FIN -