

# TSSR - Compte rendu TP11 : Serveur OpenVPN Sous Debian

Jean-Paul MELISSE

27 Février 2025

## 1 Introduction

L'objectif de ce TP est de nous familiariser un peu plus avec la distribution Linux. Pour ce faire, on va créer un serveur sous Debian qui contiendra le logiciel OpenVPN. OpenVPN est un logiciel libre qui permet de créer des VPNs (réseaux privés virtuels). Il permet de créer un tunnel sécurisé entre deux machines via Internet. Ce tunnel chiffre (encode) les données échangées, comme si les deux machines étaient sur le même réseau local, même si elles sont à distance. On utilisera l'hyperviseur VirtualBox pour créer plusieurs machines virtuelles. Ce petit document va nous décrire les étapes à suivre.

## 2 Créer un dossier de travail

On choisit un emplacement de travail où on va garder tous les fichiers installés et la configuration des machines virtuelles. On décide de sauvegarder dans le chemin : "D:\Users\Jean-Paul\Desktop\tp11\", comme montre la Figure 1.

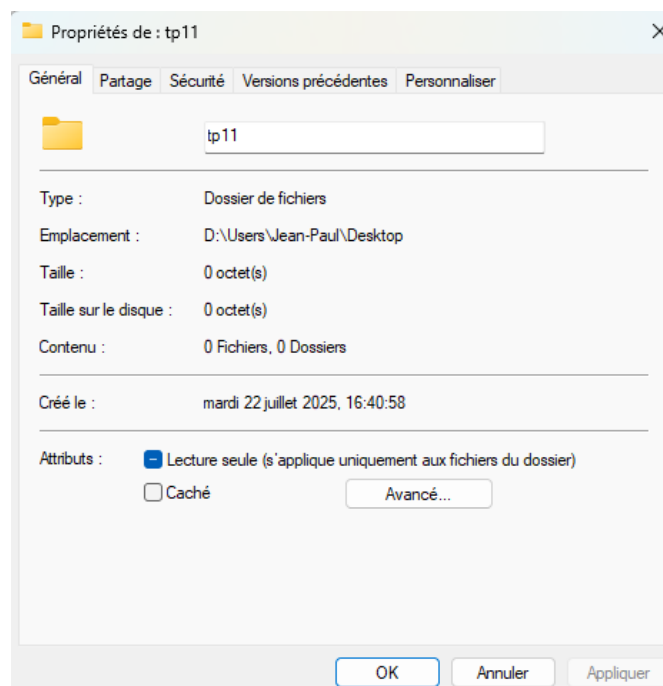


Figure 1

### 3 Télécharger Debian

La suite est de télécharger le système d'exploitation Debian. On y accède via le lien <https://www.debian.org/>. On clique sur "Téléchargement". On le sauvegarde dans le dossier tp11 sous le format .iso (Voir la Figure 2).

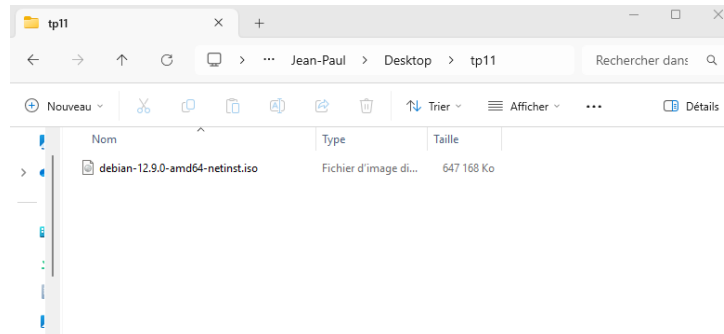


Figure 2

### 4 Créer la machine virtuelle tp11-router

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp11-router" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation de Debian. En détectant le CD, le type "Linux", le subtype "Debian" et la version "Debian (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 20 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 3).

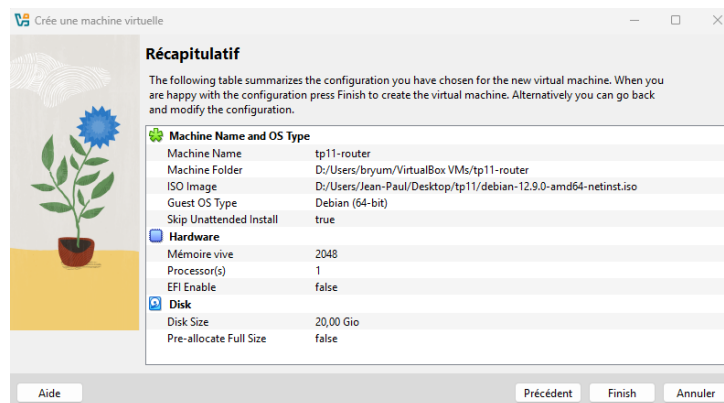


Figure 3

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et sur l'interface "Adapter 1". On change le mode d'accès réseau à "Accès par pont", puis on clique sur "OK". Le mode "Accès par pont" va permettre à la machine virtuelle de récupérer une adresse IP directement depuis un serveur DHCP (par exemple : notre box). Cela permet de connecter la machine virtuelle à l'Internet. On clique sur "Adapter 2" et on coche la case "Activer l'interface réseau". Pour cette interface, on change le mode d'accès réseau à "Réseau interne". Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

## 5 Installer Debian sur tp11-router

Après avoir démarré la machine virtuelle, on choisit l'option "Graphical Install". On choisit la langue "English" et on clique sur "Continue". Le fait qu'on ait choisi l'anglais comme langue, Debian va nous proposer des pays anglophones par défaut pour la géolocalisation. On choisit donc "other" et on clique sur "Continue". Il nous propose donc dans un premier temps, des zones continentales. On choisit "Europe" pour la localisation car on se trouve en Europe et on clique sur "Continue". Enfin, on choisit "France" pour la localisation et on clique sur "Continue". Debian comprend qu'il y a une incohérence entre la langue choisie et le pays. Il nous propose différentes versions de la langue anglaise en fonction des pays. On choisit "United States - en\_US.UTF-8" et on clique sur "Continue". Pour la configuration du clavier, on choisit "French" et on clique sur "Continue". Debian a détecté plusieurs interfaces réseau. On choisit l'interface "enp0s3" comme l'interface primaire et on clique sur "Continue". Après avoir reçu par le service DHCP une adresse IP, on renomme le nom de l'hôte (Hostname) en "tp11-router" et on clique sur "Continue" (Figure 4).



Figure 4

On lui donne un nom de domaine (Domain Name), qui sera "afpa" ici et on clique sur "Continue". On donne un mot de passe à l'administrateur root (dans ce cas ici, le mot de passe est "root" et on le met deux fois) et on clique sur "Continue". On crée un nouvel utilisateur qu'on nomme "user" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur user (dans ce cas ici, le mot de passe est "user" et on le met deux fois) et on clique sur "Continue".

On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique sur "Continue". Comme on a mis un seul disque dur pour cette machine virtuelle, on n'a pas besoin de scanner d'autres disques. On coche la case "No" et on clique sur "Continue".

La prochaine étape concerne le gestionnaire de paquets "Advanced Packaging Tool" (APT). Ce gestion-

naire contient tous les programmes et applications utilisés pour les machines Debian et Ubuntu. Il gère aussi les mises à jour de ces applications. On choisit le pays "France" et on clique sur "Continue". On choisit le miroir pour le gestionnaire de paquets : "deb.debian.org" et on clique sur "Continue" deux fois. Ce qui signifie que quand on va utiliser le gestionnaire APT, il va se connecter à ce miroir pour vérifier ou installer les paquets. On coche la case "No" car on ne veut pas participer aux études statistiques et on clique sur "Continue".

On décoche toutes les cases des logiciels et on clique sur "Continue". On coche la case "Yes" pour installer le boot GRUB et on clique sur "Continue". On choisit l'option `/dev/sda` pour installer GRUB dans cet emplacement et on clique sur "Continue". On clique sur "Continue" pour finaliser l'installation de Debian et redémarrer la machine. Note : Après le redémarrage de Debian, on peut faire un clone de notre machine virtuelle.

## 6 Installer les paquets nécessaires sur tp11-router

Le but ici est d'installer les paquets nécessaires pour le bon fonctionnement de notre routeur. Après le redémarrage de notre machine virtuelle, on se connecte avec les identifiants de l'utilisateur root. On tape la commande : `"apt install bind9 iptables iptables-persistent isc-dhcp-server man sudo"` pour installer le paquet manuel, la commande sudo, les paquets iptables et iptables-persistent qui sont essentiels pour configurer le pare-feu, et les paquets bind9 et isc-dhcp-server pour les services DHCP et DNS. Ici c'est notre routeur qui va gérer la distribution des adresses IP. (Voir la Figure 5).

```

tp11-router [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

Debian GNU/Linux 12 tp11-router tty1

tp11-router login: root
Password:
Linux tp11-router 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

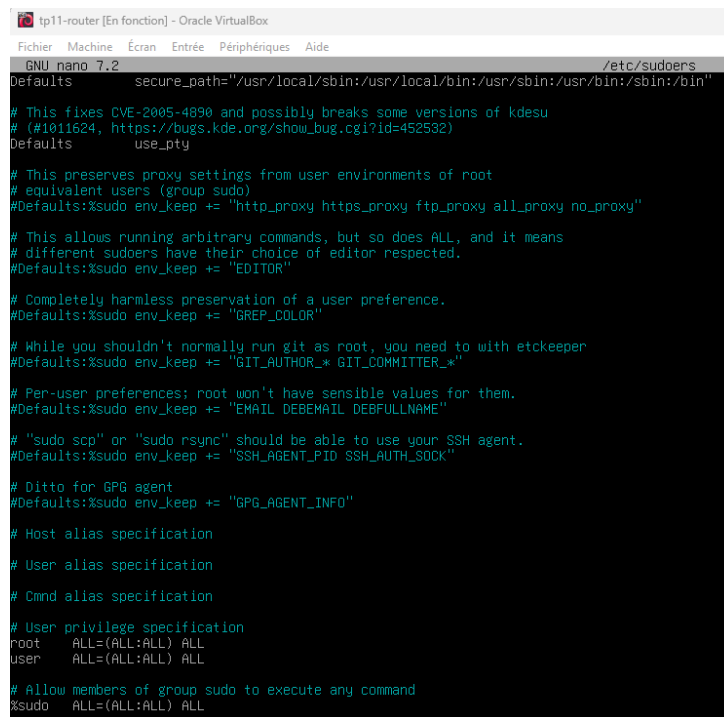
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@tp11-router:~# apt install bind9 iptables iptables-persistent isc-dhcp-server man sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'man-db' instead of 'man'
The following additional packages will be installed:
  bind9-libs bind9-utils bsdextrautils dns-root-data groff-base libfstrm0 libgdbm6 libip6tc2 libjemalloc2 libltdb0 libnsminddb0 libnetfilter-contrack3
  libnetlink0 libnftnl2-14 libpipelinelibprotobuf-c libuchardet0 libuv1 netfilter-persistent policycoreutils selinux-utils
Suggested packages:
  bind-doc dnstools resolvconf ufw groff firewallld policykit-1 isc-dhcp-server-ldap ieee-data gdbm-l10n mmdns-bin www-browser
  bind9-doc bind9-libs bind9-utils bsdextrautils dns-root-data groff-base iptables iptables-persistent isc-dhcp-server libfstrm0 libgdbm6 libip6tc2 libjemalloc2
  libltdb0 libnsminddb0 libnetfilter-contrack3 libnetlink0 libnftnl2-14 libpipelinelibprotobuf-c libuchardet0 libuv1 man-db netfilter-persistent
  policycoreutils selinux-utils sudo
0 upgraded, 27 newly installed, 0 to remove and 0 not upgraded.
Need to get 8,875 kB of archives.
After this operation, 31.3 MB of additional disk space will be used.
Do you want to continue? (Y/n) _

```

Figure 5

Le suite maintenant est de configurer la commande `sudo` afin qu'on puisse l'utiliser avec l'utilisateur `user`. Après avoir installé `sudo`, on va modifier le fichier `sudoers` pour ajouter l'utilisateur `user` dans les droits de permissions. On fait une copie du fichier `sudoers` en tapant la commande `"cp /etc/sudoers /etc/sudoers.ori"` pour éviter d'écraser le fichier original. Pour modifier ce fichier, on tape la commande : `"nano /etc/sudoers"`. Dans ce fichier, on ajoute après la ligne du `root` la commande : `"user ALL=(ALL:ALL) ALL"`. On sauvegarde en utilisant `"Ctrl+X"` et la touche `"y"` pour `"Yes"`, et la touche `"Entrée"` pour sauvegarder avec le même nom. On peut visualiser cette étape sur la Figure 6.



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/sudoers
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
user    ALL=(ALL:ALL) ALL

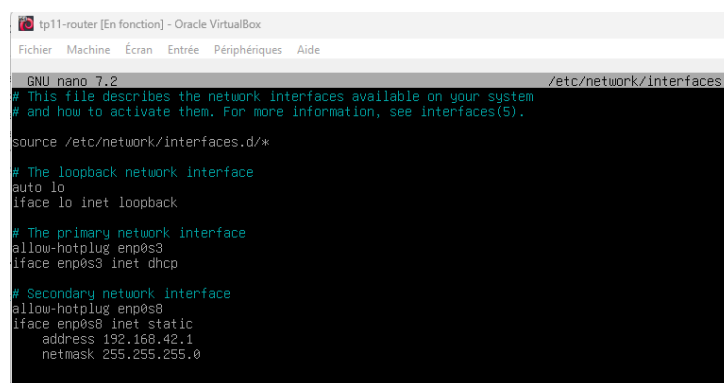
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

Figure 6

On a fini de configurer le fichier sudo. On peut se déconnecter du compte en utilisant la commande "logout".

## 7 Configurer le réseau de tp11-router

On retourne sur la machine "tp11-router" et on se connecte en tant qu'utilisateur user. On crée une copie du fichier "interfaces" qui se trouve dans le chemin "/etc/network" avec la commande : "sudo cp /etc/network/interfaces /etc/network/interfaces.ori" pour ne pas écraser le fichier original. Par la suite, on va travailler sur le fichier "interfaces" pour configurer le réseau de notre Debian. On modifie le fichier en tapant la commande : "sudo nano /etc/network/interfaces". Dans ce fichier, on laisse l'interface enp0s3 en dhcp IPV4 et on donne une IP fixe à notre routeur sur l'interface enp0s8 (Voir la Figure 7)



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

# Secondary network interface
allow-hotplug enp0s8
iface enp0s8 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

Figure 7

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On redémarre la configuration réseau de la machine en utilisant la commande "sudo systemctl restart networking". On

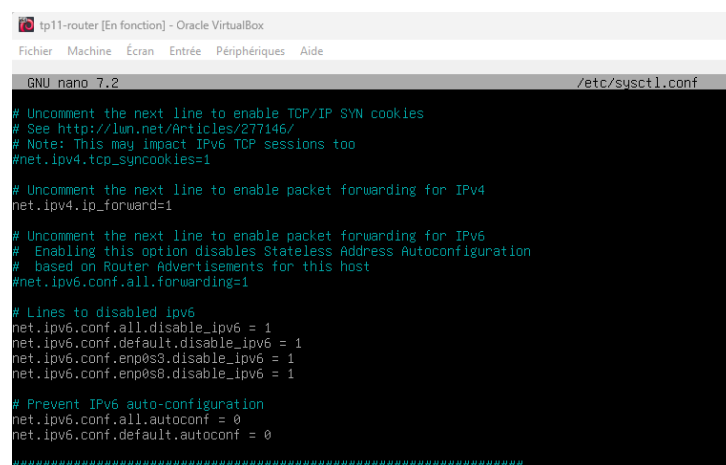
vérifie si le service de configuration réseau marche correctement en tapant la commande "sudo systemctl status networking" et c'est bien le cas.

## 8 Configurer le pare-feu de tp11-router

On vérifie avec la commande "sudo sysctl net.ipv4.ip\_forward" que l'IP forwarding est désactivé et c'est bien le cas. On décide de l'activer en modifiant le fichier "sysctl.conf" qui se trouve dans le chemin "/etc/". On le copie d'abord avec la commande : "sudo cp /etc/sysctl.conf /etc/sysctl.conf.ori" pour éviter d'écraser le fichier original. On tape donc : "sudo nano /etc/sysctl.conf". On enlève le caractère "#" devant la ligne "net.ipv4.ip\_foward=1" pour activer l'IP forwarding. On en profite pour désactiver l'IPv6 complètement. On ajoute les lignes suivantes :

- net.ipv6.conf.all.disable\_ipv6 = 1
- net.ipv6.conf.default.disable\_ipv6 = 1
- net.ipv6.conf.enp0s3.disable\_ipv6 = 1
- net.ipv6.conf.enp0s8.disable\_ipv6 = 1

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On tape la commande "sudo sysctl -p" afin de prendre en compte les changements. On peut voir les étapes précédentes sur la Figure 8.



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/sysctl.conf

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lun.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1

# Lines to disabled ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.enp0s3.disable_ipv6 = 1
net.ipv6.conf.enp0s8.disable_ipv6 = 1

# Prevent IPv6 auto-configuration
net.ipv6.conf.all.autoconf = 0
net.ipv6.conf.default.autoconf = 0

#####
```

Figure 8

On retape la commande "sudo sysctl net.ipv4.ip\_forward" pour vérifier que maintenant l'IP forwarding est activé et c'est bien le cas. On va créer un fichier où on va stocker des règles de pare-feu pour notre routeur. On tape la commande : "sudo nano iptables.sh" pour créer le fichier iptables.sh. Dans ce fichier, on ajoute les règles suivantes :

- 1 # delete rules and chains in the tables
- 2 iptables -F
- 3 iptables -X

```

4 iptables -t nat -F
5 iptables -t nat -X
6
7 # set the policy for the chains to the target
8 iptables -P INPUT DROP
9 iptables -P OUTPUT ACCEPT
10 iptables -P FORWARD ACCEPT
11
12 # add rules to the end of the chain
13 iptables -A INPUT -i lo -j ACCEPT
14 iptables -A INPUT -i enp0s8 -j ACCEPT
15 iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
16 iptables -t nat -A POSTROUTING -s 192.168.42.0/24 -j MASQUERADE

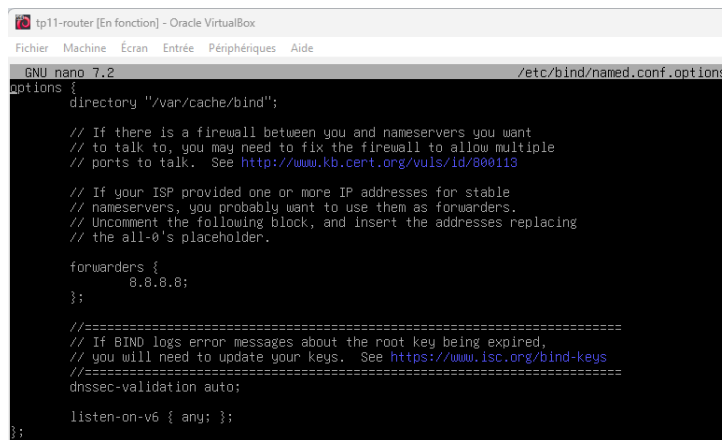
```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On applique les changements en tapant la commande : "sudo bash iptables.sh" [Attention : Si on redémarre le serveur tp11-router, il faut relancer cette commande]. On peut vérifier les règles et les chaînes du pare-feu dans la table "filter" en tapant la commande : "sudo iptables -L -v". On vérifie aussi la table "nat" en tapant la commande : "sudo iptables -t nat -L -v". On voit qu'il y a bien une entrée "POSTROUTING" venant des sources 192.168.42.0/24.

On va rendre les règles de pare-feu persistantes en tapant la commande : "sudo netfilter-persistent save". Cela va enregistrer les règles actuelles dans les fichiers /etc/iptables/rules.v4 pour IPv4. On redémarre la machine avec la commande "sudo reboot". Après le redémarrage de la machine, on se connecte avec les identifiants de l'utilisateur "user". On vérifie que les règles de pare-feu sont conservées en tapant les commandes : "sudo iptables -L -v" ou "sudo iptables -t nat -L -v" [Si ce n'est pas le cas, on recharge les règles avec la commande : "sudo iptables-restore -n < /etc/iptables/rules.v4"].

## 9 Configurer le serveur BIND de tp11-router

Maintenant on va modifier le fichier configuration du paquet bind afin d'ajouter l'adresse IP du serveur DNS de Google dans notre routeur. On modifie le fichier en tapant la commande : "sudo nano /etc/bind/named.conf.options". Dans ce fichier, on ajoute "8.8.8.8" dans la partie "forwarder" afin d'ajouter le serveur DNS de Google comme "forwarder". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir l'étape précédente sur la Figure 9.



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 7.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
    };

    //=====  

    // If BIND logs error messages about the root key being expired,  

    // you will need to update your keys.  See https://www.isc.org/bind-keys  

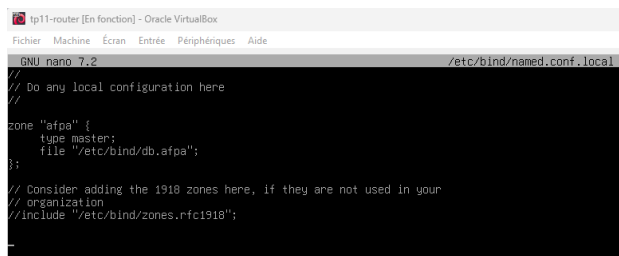
    //=====  

    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

Figure 9

Après avoir modifié le fichier de configuration du bind, il faut redémarrer le service DNS. Pour cela on tape : "sudo systemctl restart named". Puis, on va ajouter une nouvelle zone DNS dans notre serveur DNS. On tape la commande : "sudo nano /etc/bind/named.conf.local" pour éditer ce fichier. Dans ce fichier, on ajoute les commandes "zone" pour définir la zone qu'on veut créer. On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". Ensuite, on va créer la base de données pour la zone "afpa". On tape la commande : "sudo nano /etc/bind/db.afpa" pour créer le fichier de configuration. Dans ce fichier, on ajoute les commandes pour configurer la zone "afpa" et créer les enregistrements du serveur principal de noms (ns1), du tp11-router et du tp11-server. On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur les Figures 10 et 11.



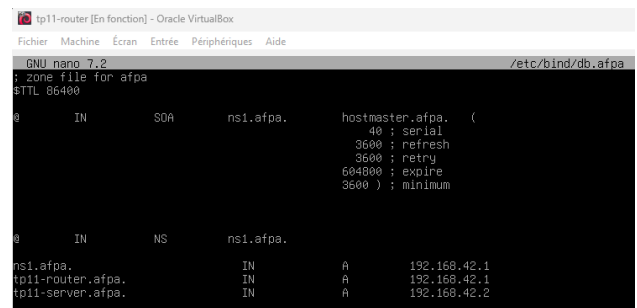
```
tp11-router [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 7.2 /etc/bind/named.conf.local
//
// Do any local configuration here
//

zone "afpa" {
    type master;
    file "/etc/bind/db.afpa";
};

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

Figure 10



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

GNU nano 7.2 /etc/bind/db.afpa
; zone file for afpa
$TTL 86400
@ IN SOA ns1.afpa. hostmaster.afpa. (
    40 ; serial
    3600 ; refresh
    3600 ; retry
    604800 ; expire
    3600 ) ; minimum

@ IN NS ns1.afpa.

ns1.afpa. IN A 192.168.42.1
tp11-router.afpa. IN A 192.168.42.1
tp11-server.afpa. IN A 192.168.42.2
```

Figure 11

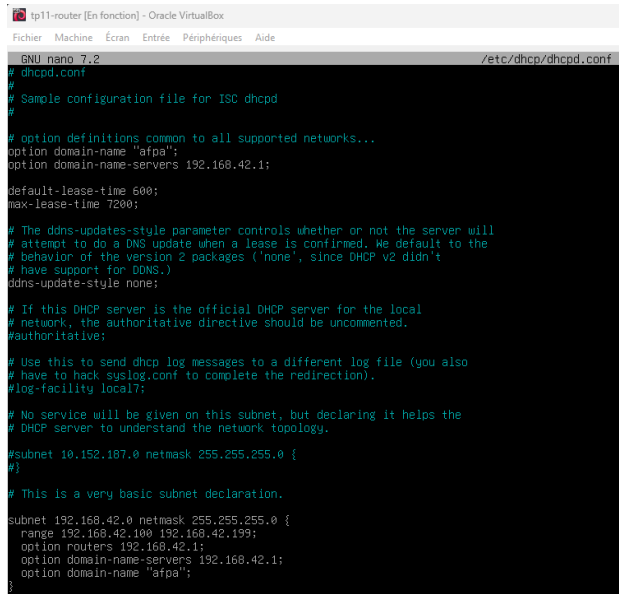
On peut vérifier si le fichier de configuration est bon en tapant la commande : "sudo named-checkconf" et "sudo named-checkzone afpa /etc/bind/db.afpa". Si tout est OK, on redémarre le service bind en tapant : "sudo service bind9 restart". On redémarre la configuration réseau de la machine en utilisant la commande "sudo systemctl restart networking". On vérifie si le service de configuration réseau marche correctement en tapant la commande "sudo systemctl status networking" et c'est bien le cas.

## 10 Configurer le serveur DHCP de tp11-router

Le but ici est de configurer le serveur DHCP sur notre routeur. On copie le fichier dhcpd.conf afin de pouvoir modifier une copie pour ne pas écraser le fichier original en faisant la commande : "sudo cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.ori". On édite le fichier dhcpd.conf avec la commande : "sudo



nano /etc/dhcp/dhcpd.conf". Dans ce fichier, on va ajouter le réseau 192.168.42.0, une nouvelle plage d'adresses (192.168.42.100 - 192.168.42.199) qui sera accessible pour les machines. On rajoute aussi une passerelle par défaut qui sera 192.168.42.1, un serveur DNS qui sera à l'adresse IP : 192.168.42.1 et un nom de domaine "afpa". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". Puis on va modifier le fichier isc-dhcp-server. On tape la commande : "sudo nano /etc/default/isc-dhcp-server". Dans ce fichier, on va spécifier notre interface réseau (ici c'est "enp0s8"). On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur les Figures 12 et 13.



```
GNU nano 7.2 /etc/dhcp/dhcpd.conf
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
option domain-name "afpa";
option domain-name-servers 192.168.42.1;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

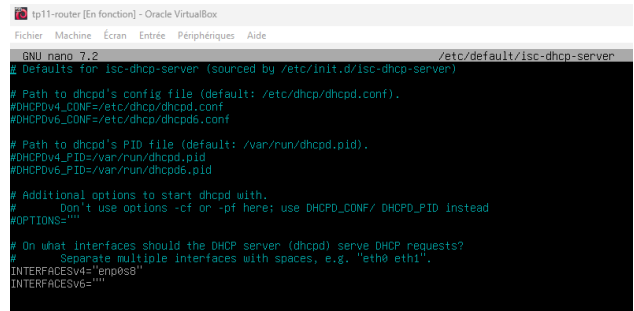
# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.107.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 192.168.42.0 netmask 255.255.255.0 {
    range 192.168.42.100 192.168.42.199;
    option routers 192.168.42.1;
    option domain-name-servers 192.168.42.1;
    option domain-name "afpa";
}
```

Figure 12



```
GNU nano 7.2 /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)
#
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
DHCPD_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf
#
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid
#
# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""
#
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. 'eth0 eth1'.
INTERFACESv4="enp0s8"
INTERFACESv6=""
```

Figure 13

On redémarre le service DHCP en utilisant la commande "sudo systemctl restart isc-dhcp-server". On vérifie si le DHCP marche correctement en tapant la commande "sudo systemctl status isc-dhcp-server" et c'est bien le cas.

## 11 Créer la machine virtuelle tp11-server

On retourne sur VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp11-server" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation de Debian. En détectant le CD, le type "Linux", le subtype "Debian" et la version "Debian (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 20 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 14).

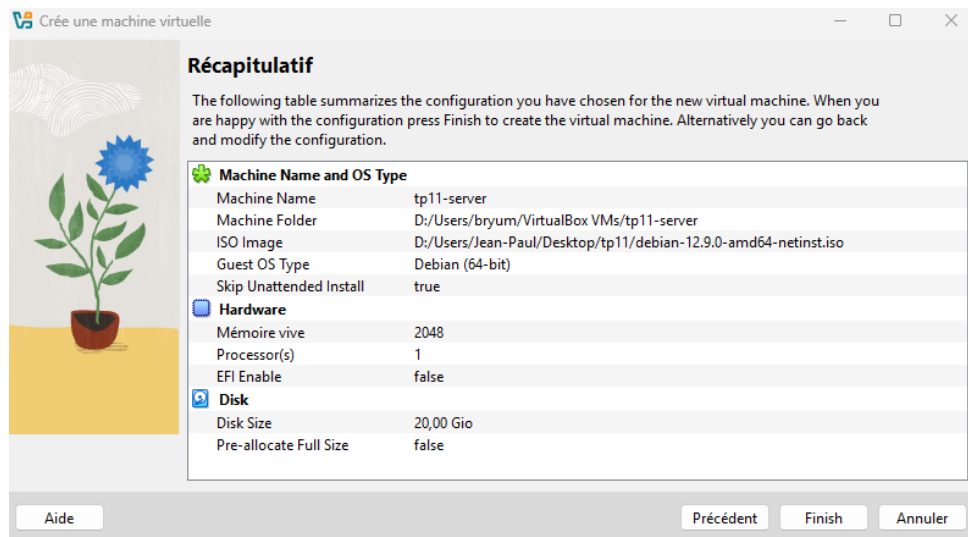


Figure 14

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et sur l'interface "Adapter 1". On change le mode d'accès réseau à "Réseau interne". Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

## 12 Installer Debian sur tp11-server

Après avoir démarré la machine virtuelle, on choisit l'option "Graphical Install". On choisit la langue "English" et on clique sur "Continue". Le fait qu'on ait choisi l'anglais comme langue, Debian va nous proposer des pays anglophones par défaut pour la géolocalisation. On choisit donc "other" et on clique sur "Continue". Il nous propose donc dans un premier temps, des zones continentales. On choisit "Europe" pour la localisation car on se trouve en Europe et on clique sur "Continue". Enfin, on choisit "France" pour la localisation et on clique sur "Continue". Debian comprend qu'il y a une incohérence entre la langue choisie et le pays. Il nous propose différentes versions de la langue anglaise en fonction des pays. On choisit "United States - en\_US.UTF-8" et on clique sur "Continue". Pour la configuration du clavier, on choisit "French" et on clique sur "Continue". Après avoir reçu par le service DHCP une adresse IP, on renomme le nom de l'hôte (Hostname) en "tp11-server" et on clique sur "Continue" (Figure 15).

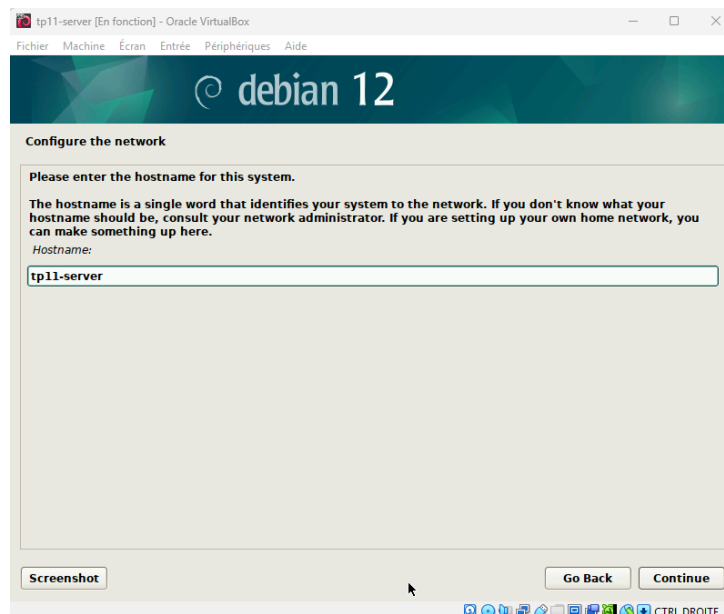


Figure 15

On lui donne un nom de domaine (Domain Name), qui sera "afpa" ici et on clique sur "Continue". On donne un mot de passe à l'administrateur root (dans ce cas ici, le mot de passe est "root" et on le met deux fois) et on clique sur "Continue". On crée un nouvel utilisateur qu'on nomme "user" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur user (dans ce cas ici, le mot de passe est "user" et on le met deux fois) et on clique sur "Continue".

On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique sur "Continue". Comme on a mis un seul disque dur pour cette machine virtuelle, on n'a pas besoin de scanner d'autres disques. On coche la case "No" et on clique sur "Continue".

La prochaine étape concerne le gestionnaire de paquets "Advanced Packaging Tool" (APT). Ce gestionnaire contient tous les programmes et applications utilisés pour les machines Debian et Ubuntu. Il gère aussi les mises à jour de ces applications. On choisit le pays "France" et on clique sur "Continue". On choisit le miroir pour le gestionnaire de paquets : "deb.debian.org" et on clique sur "Continue" deux fois. Ce qui signifie que quand on va utiliser le gestionnaire APT, il va se connecter à ce miroir pour vérifier ou installer les paquets. On coche la case "No" car on ne veut pas participer aux études statistiques et on clique sur "Continue".

On décoche toutes les cases des logiciels et on clique sur "Continue". On coche la case "Yes" pour installer le boot GRUB et on clique sur "Continue". On choisit l'option "/dev/sda" pour installer GRUB dans cet emplacement et on clique sur "Continue". On clique sur "Continue" pour finaliser l'installation de Debian et redémarrer la machine. Note : Après le redémarrage de Debian, on peut faire un clone de notre machine virtuelle.

### 13 Installer les paquets nécessaires sur tp11-server

Le but ici est d'installer les paquets nécessaires pour le bon fonctionnement de notre serveur. Après le redémarrage de notre machine virtuelle, on se connecte avec les identifiants de l'utilisateur root. On tape la commande "apt install openvpn ssh man sudo" pour installer le paquet manuel, la commande sudo, le paquet openvpn qui est essentiel pour créer des VPNs (réseaux privés virtuels) et le paquet ssh (Secure Shell) qui permet de se connecter à distance à une autre machine (Voir la Figure 16).

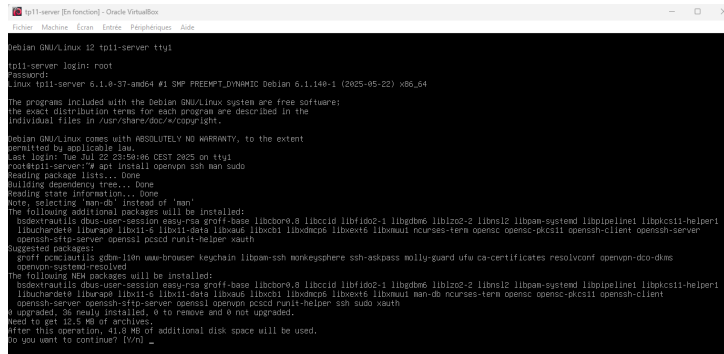


Figure 16

La suite maintenant est de configurer la commande `sudo` afin qu'on puisse l'utiliser avec l'utilisateur `user`. Après avoir installé `sudo`, on va modifier le fichier `sudoers` pour ajouter l'utilisateur `user` dans les droits de permissions. On fait une copie du fichier `sudoers` en tapant la commande `"cp /etc/sudoers /etc/sudoers.ori"` pour éviter d'écraser le fichier original. Pour modifier ce fichier, on tape la commande : `"nano /etc/sudoers"`. Dans ce fichier, on ajoute après la ligne du `root` la commande : `"user ALL=(ALL:ALL) ALL"`. On sauvegarde en utilisant `"Ctrl+X"` et la touche `"y"` pour `"Yes"`, et la touche `"Entrée"` pour sauvegarder avec le même nom. On peut visualiser cette étape sur la Figure 17.

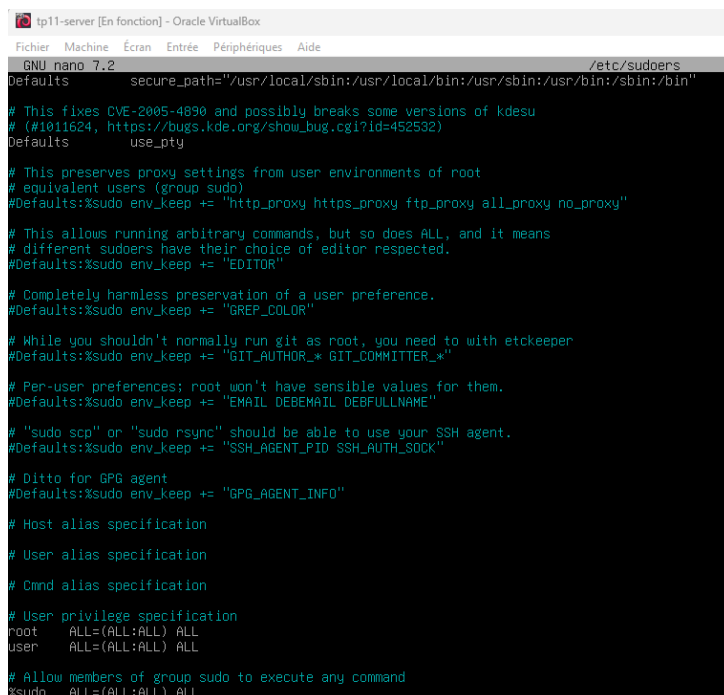
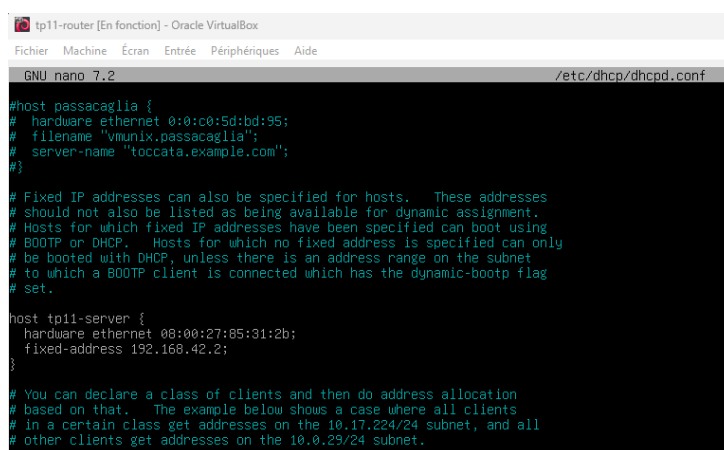


Figure 17

On a fini de configurer le fichier sudo. On peut se déconnecter du compte en utilisant la commande "logout".

## 14 Configurer le réseau de tp11-server

On retourne sur la machine "tp11-router". On va faire une réservation d'adresse IP pour la machine "tp11-server". On va rouvrir le fichier dhcp.conf en tapant : "sudo nano /etc/dhcp/dhcpd.conf". Dans ce fichier, on va attribuer à la machine "tp11-server" l'adresse IP suivante : 192.168.42.2 (cf : Figure 18). Pour cela, il faut connaître l'adresse physique (ou mac) de la machine "tp11-server". Pour cela, sur la machine "tp11-server", on tape la commande : "ip a".



```
tp11-router [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/dhcp/dhcpd.conf

#host passacaglia {
#   hardware ethernet 0:0:c0:5d:bd:95;
#   filename "vmunix.passacaglia";
#   server-name "toccata.example.com";
#}

# Fixed IP addresses can also be specified for hosts.  These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP.  Hosts for which no fixed address is specified can only
# be booted with DHCP, unless there is an address range on the subnet
# to which a BOOTP client is connected which has the dynamic-bootp flag
# set.

host tp11-server {
    hardware ethernet 08:00:27:85:31:2b;
    fixed-address 192.168.42.2;
}

# You can declare a class of clients and then do address allocation
# based on that.  The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet, and all
# other clients get addresses on the 10.0.29/24 subnet.
```

Figure 18

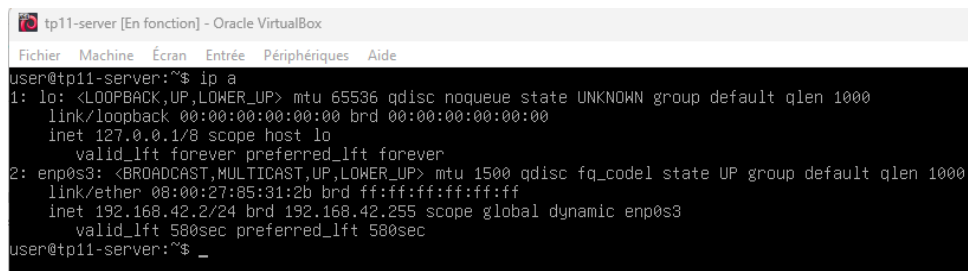
On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On redémarre le service DHCP en utilisant la commande "sudo systemctl restart isc-dhcp-server". On vérifie si le DHCP marche correctement en tapant la commande "sudo systemctl status isc-dhcp-server" et c'est bien le cas.

On retourne sur la machine "tp11-server" et on se connecte en tant qu'utilisateur user. Avant de redémarrer le réseau, on va d'abord désactiver IPv6. On copie le fichier "sysctl.conf" avec la commande : "sudo cp /etc/sysctl.conf /etc/sysctl.conf.ori" pour éviter d'écraser le fichier original. Puis on tape : "sudo nano /etc/sysctl.conf" pour éditer le fichier. On ajoute les lignes suivantes :

- net.ipv6.conf.all.disable\_ipv6 = 1
- net.ipv6.conf.default.disable\_ipv6 = 1
- net.ipv6.conf.enp0s3.disable\_ipv6 = 1

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On tape la commande "sudo sysctl -p" afin de prendre en compte les changements. On redémarre la configuration réseau de la machine en utilisant la commande "sudo systemctl restart networking". On vérifie si le service de configuration réseau marche correctement en tapant la commande "sudo systemctl status networking" et c'est bien le cas. On tape la commande : "ip a" et on voit, comme sur la Figure 19, qu'on a bien récupéré la bonne adresse IP réservée et que IPv6 est bien désactivé. On tape ensuite la commande : "ip r" pour

voir l'adresse IP de la passerelle par défaut et on voit qu'on a bien récupéré la bonne adresse IP de notre routeur "tp11-router" comme passerelle.

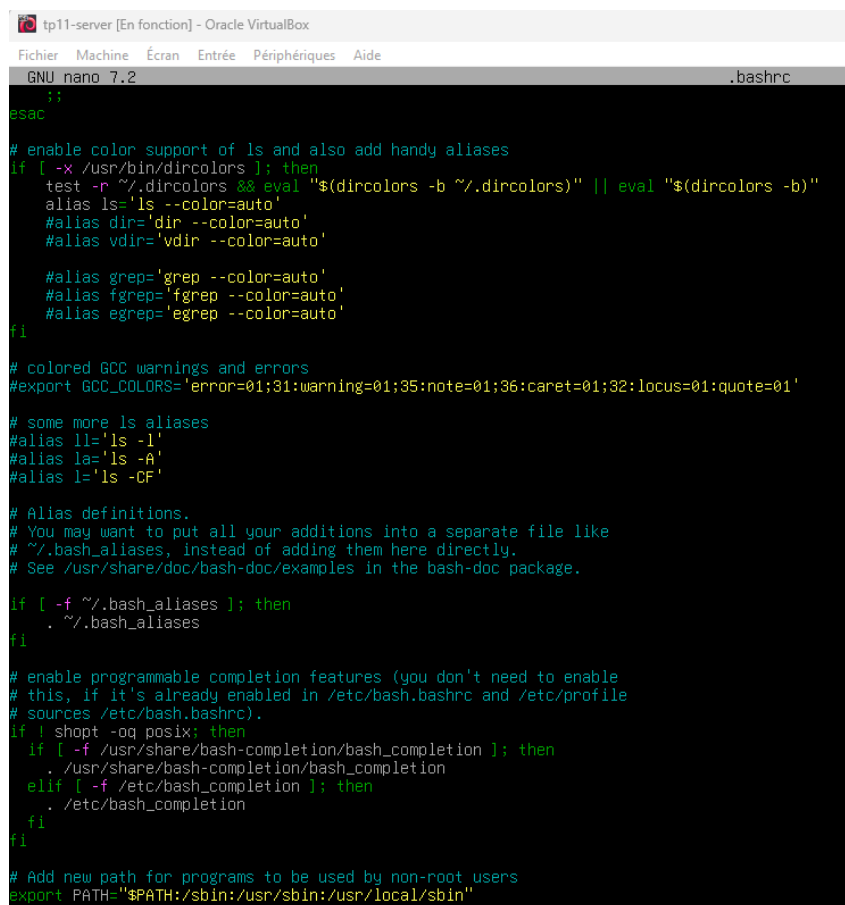


```
user@tp11-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:85:31:2b brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.2/24 brd 192.168.42.255 scope global dynamic enp0s3
        valid_lft 580sec preferred_lft 580sec
user@tp11-server:~$ _
```

Figure 19

## 15 Configurer le serveur OpenVPN de tp11-server

Maintenant, on va configurer openvpn sur la machine tp11-server. On va modifier le fichier ".bashrc" qui est un fichier de configuration du shell bash, exécuté à chaque fois qu'on ouvre un terminal interactif (non-login). On va d'abord copier le fichier ".bashrc" avec la commande : "cp .bashrc .bashrc.ori" pour éviter d'écraser le fichier original. Puis on tape : "nano .bashrc" pour l'éditer. Dans le fichier, on va ajouter les chemins "/sbin", "/usr/sbin" et "/usr/local/sbin" à la variable d'environnement PATH (Voir la Figure 20). PATH est une variable d'environnement qui indique au shell où chercher les programmes exécutables quand tu tapes une commande.



```
GNU nano 7.2 .bashrc
;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    #alias grep='grep --color=auto'
    #alias fgrep='fgrep --color=auto'
    #alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

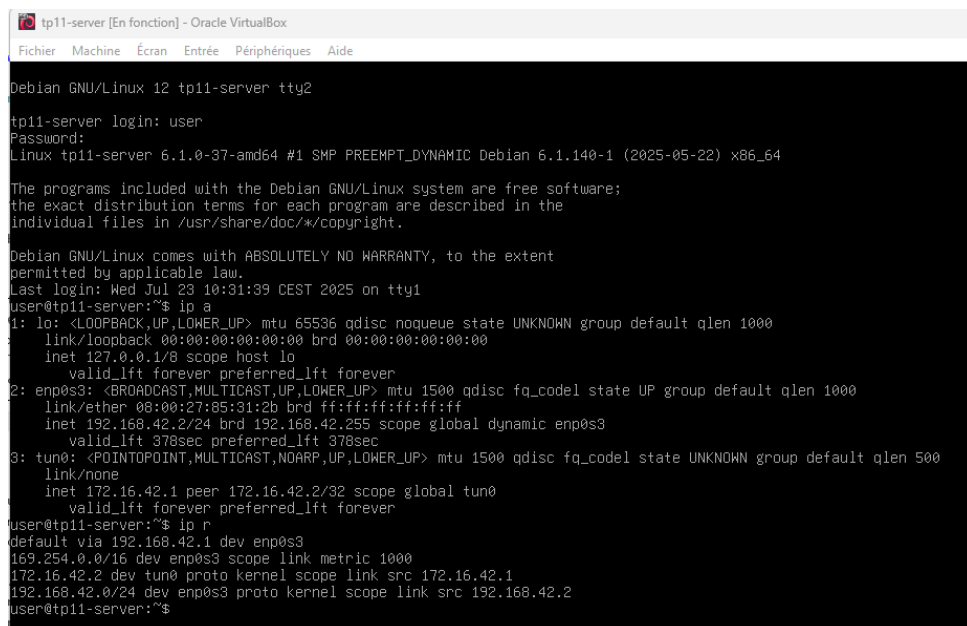
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

# Add new path for programs to be used by non-root users
export PATH="$PATH:/sbin:/usr/sbin:/usr/local/sbin"
```

Figure 20

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". Puis on tape la commande : "source .bashrc" pour appliquer les changements faits. On peut vérifier si la variable PATH a bien sauvegardé les chemins en tapant la commande : "echo \$PATH" et on voit que c'est bien le cas. La prochaine étape est de créer un tunnel avec openvpn. On tape donc la commande : "sudo openvpn --dev tun0 --ifconfig 172.16.42.1 172.16.42.2". Cette commande crée un tunnel (sans chiffrement) qui permet à deux machines de communiquer comme si elles étaient sur le même réseau privé virtuel. Ici, l'IP 172.16.42.1 est l'IP du côté serveur tandis que l'IP 172.16.42.2 est celle du côté client. Puis, on bascule sur un nouveau terminal texte (ici tty2) en appuyant "Ctrl + Alt + F2". Sur le nouveau terminal, on se connecte en tant qu'utilisateur user. On tape la commande : "ip a" pour afficher les adresses de toutes les cartes réseaux, puis on tape ensuite la commande : "ip r" pour voir les routes. On voit, comme dans la Figure 21, que le tunnel a bien été créé.



```

tp11-server login: user
Password:
Linux tp11-server 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 23 10:31:39 CEST 2025 on tty1
user@tp11-server:~$ 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
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:85:31:2b brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.2/24 brd 192.168.42.255 scope global dynamic enp0s3
        valid_lft 378sec preferred_lft 378sec
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 172.16.42.1 peer 172.16.42.2/32 scope global tun0
        valid_lft forever preferred_lft forever
user@tp11-server:~$ ip r
default via 192.168.42.1 dev enp0s3
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.42.2 dev tun0 proto kernel scope link src 172.16.42.1
192.168.42.0/24 dev enp0s3 proto kernel scope link src 192.168.42.2
user@tp11-server:~$

```

Figure 21

On essaie de joindre la machine côté client en utilisant la commande ping 172.16.42.2, mais aucune réponse n'est reçue. Nous verrons dans les prochaines sections comment établir la connexion avec la machine cliente du VPN.

## 16 Créer la machine virtuelle tp11-client

On retourne sur VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp11-client" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation de Debian. En détectant le CD, le type "Linux", le subtype "Debian" et la version "Debian (64 bits)" sont automatiquement mis. On coche la case "Skip Unattended Installation" et on clique sur "Suivant". On lui donne une mémoire vive (RAM) de 2048 Mo et un processeur, et on clique sur "Suivant". On lui donne un disque dur de 20 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 22).

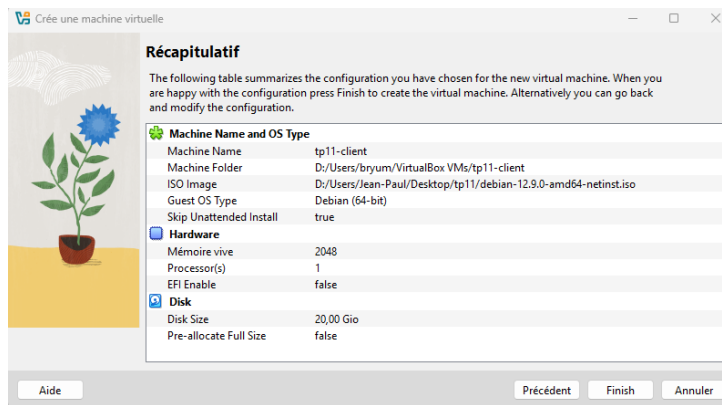


Figure 22

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et sur l'interface "Adapter 1". On change le mode d'accès réseau à "Réseau interne". Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

## 17 Installer Debian sur tp11-client

Après avoir démarré la machine virtuelle, on choisit l'option "Graphical Install". On choisit la langue "English" et on clique sur "Continue". Le fait qu'on ait choisi l'anglais comme langue, Debian va nous proposer des pays anglophones par défaut pour la géolocalisation. On choisit donc "other" et on clique sur "Continue". Il nous propose donc dans un premier temps, des zones continentales. On choisit "Europe" pour la localisation car on se trouve en Europe et on clique sur "Continue". Enfin, on choisit "France" pour la localisation et on clique sur "Continue". Debian comprend qu'il y a une incohérence entre la langue choisie et le pays. Il nous propose différentes versions de la langue anglaise en fonction des pays. On choisit "United States - en\_US.UTF-8" et on clique sur "Continue". Pour la configuration du clavier, on choisit "French" et on clique sur "Continue". Après avoir reçu par le service DHCP une adresse IP, on renomme le nom de l'hôte (Hostname) en "tp11-client" et on clique sur "Continue" (Figure 23).



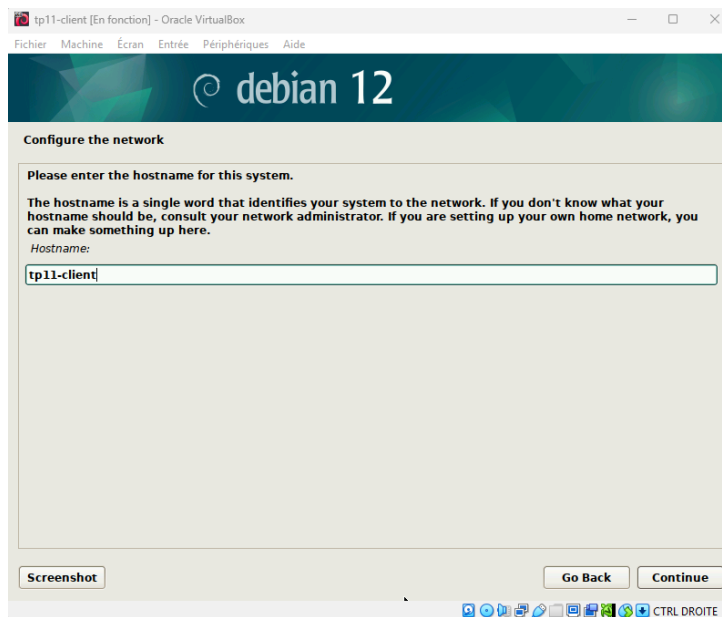


Figure 23

On lui donne un nom de domaine (Domain Name), qui sera "afpa" ici et on clique sur "Continue". On donne un mot de passe à l'administrateur root (dans ce cas ici, le mot de passe est "root" et on le met deux fois) et on clique sur "Continue". On crée un nouvel utilisateur qu'on nomme "user" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur user (dans ce cas ici, le mot de passe est "user" et on le met deux fois) et on clique sur "Continue".

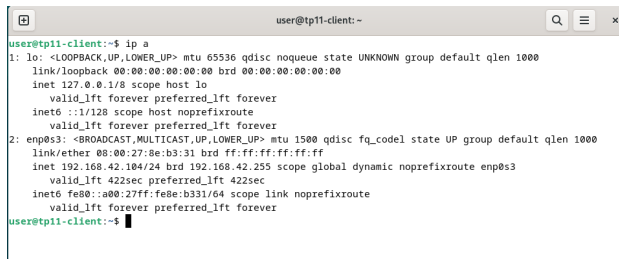
On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique sur "Continue". Comme on a mis un seul disque dur pour cette machine virtuelle, on n'a pas besoin de scanner d'autres disques. On coche la case "No" et on clique sur "Continue".

La prochaine étape concerne le gestionnaire de paquets "Advanced Packaging Tool" (APT). Ce gestionnaire contient tous les programmes et applications utilisés pour les machines Debian et Ubuntu. Il gère aussi les mises à jour de ces applications. On choisit le pays "France" et on clique sur "Continue". On choisit le miroir pour le gestionnaire de paquets : "deb.debian.org" et on clique sur "Continue" deux fois. Ce qui signifie que quand on va utiliser le gestionnaire APT, il va se connecter à ce miroir pour vérifier ou installer les paquets. On coche la case "No" car on ne veut pas participer aux études statistiques et on clique sur "Continue".

On coche les logiciels "Debian desktop environment, ... GNOME et standard system utilities" et on clique sur "Continue". On coche la case "Yes" pour installer le boot GRUB et on clique sur "Continue". On choisit l'option "/dev/sda" pour installer GRUB dans cet emplacement et on clique sur "Continue". On clique sur "Continue" pour finaliser l'installation de Debian et redémarrer la machine. Note : Après le redémarrage de Debian, on peut faire un clone de notre machine virtuelle.

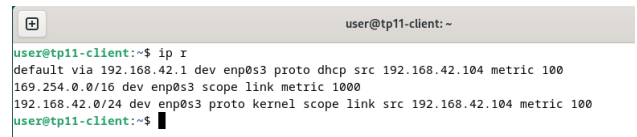
## 18 Tester le serveur DHCP

Après le redémarrage de notre machine "tp11-client", on se connecte avec les identifiants de l'utilisateur "user". On choisit "English" pour la langue et "French" pour le clavier. On décoche la case "Location Services" et on clique sur "Next", puis "Skip". Enfin, on clique sur "Start Using Debian GNU/Linux". On ouvre un terminal : on clique sur "Activities" et on tape "Terminal" sur la barre de recherche. On tape la commande : "ip a" pour vérifier si on a bien eu une adresse IP venant de notre serveur DHCP, et c'est bien le cas (Voir la Figure 24). On tape ensuite la commande : "ip r" pour voir l'adresse IP de la passerelle par défaut et on voit, comme dans la Figure 25, qu'on a bien récupéré la bonne adresse IP de notre routeur "tp10-router" comme passerelle.



```
user@tp11-client:~$ 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 08:00:27:8e:b3:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.104/24 brd 192.168.42.255 scope global dynamic noprefixroute enp0s3
        valid_lft 422sec preferred_lft 422sec
    inet6 fe80::a00:27ff:fe8e:b331/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
user@tp11-client:~$
```

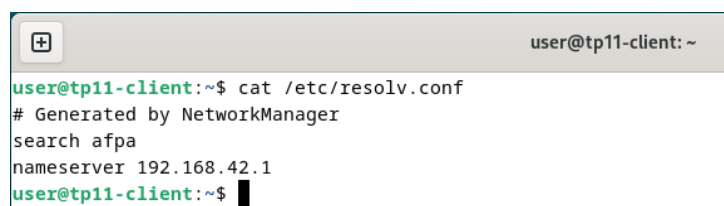
Figure 24



```
user@tp11-client:~$ ip r
default via 192.168.42.1 dev enp0s3 proto dhcp src 192.168.42.104 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
192.168.42.0/24 dev enp0s3 proto kernel scope link src 192.168.42.104 metric 100
user@tp11-client:~$
```

Figure 25

La suite est de vérifier si on a récupéré la bonne adresse IP du serveur DNS via le serveur DHCP. Pour cela, on tape la commande : "cat /etc/resolv.conf" et on constate que c'est bien le cas (cf : Figure 26).



```
user@tp11-client:~$ cat /etc/resolv.conf
# Generated by NetworkManager
search afpa
nameserver 192.168.42.1
user@tp11-client:~$
```

Figure 26

## 19 Tester le serveur DNS

La prochaine étape est de vérifier si le serveur DNS fonctionne. Sur le terminal de la machine "tp11-client", on tape la commande : "ping google.fr". On voit bien que le forwarder DNS fonctionne correctement. On fait de même avec l'enregistrement en tapant la commande : "ping ns1" et on constate que ça marche aussi. On peut voir les étapes précédentes sur la Figure 27.

```
user@tp11-client:~  
user@tp11-client:~$ ping -c 4 google.fr  
PING google.fr (142.250.179.67) 56(84) bytes of data.  
64 bytes from par21s19-in-f3.1e100.net (142.250.179.67): icmp_seq=1 ttl=114 time=21.4 ms  
64 bytes from par21s19-in-f3.1e100.net (142.250.179.67): icmp_seq=2 ttl=114 time=21.4 ms  
64 bytes from par21s19-in-f3.1e100.net (142.250.179.67): icmp_seq=3 ttl=114 time=22.8 ms  
64 bytes from par21s19-in-f3.1e100.net (142.250.179.67): icmp_seq=4 ttl=114 time=22.8 ms  
  
--- google.fr ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 21.366/22.090/22.803/0.710 ms  
user@tp11-client:~$ ping -c 4 ns1  
PING ns1.afpa (192.168.42.1) 56(84) bytes of data.  
64 bytes from _gateway (192.168.42.1): icmp_seq=1 ttl=64 time=0.841 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=2 ttl=64 time=2.70 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=3 ttl=64 time=1.73 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=4 ttl=64 time=3.12 ms  
  
--- ns1.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 0.841/2.098/3.123/0.883 ms  
user@tp11-client:~$
```

Figure 27

On va vérifier maintenant les enregistrements des adresses IP du serveur et du routeur par le serveur DNS. On tape la commande : "ping tp11-router" pour voir si on peut interagir avec la machine "tp11-router" et c'est bien le cas. On fait de même avec la machine "tp11-server" en tapant la commande : "ping tp11-server" et on constate la même chose. On peut voir les étapes précédentes sur la Figure 28.

```
user@tp11-client:~  
user@tp11-client:~$ ping -c 4 tp11-router  
PING tp11-router.afpa (192.168.42.1) 56(84) bytes of data.  
64 bytes from _gateway (192.168.42.1): icmp_seq=1 ttl=64 time=4.31 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=2 ttl=64 time=3.16 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=3 ttl=64 time=1.51 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=4 ttl=64 time=2.13 ms  
  
--- tp11-router.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3007ms  
rtt min/avg/max/mdev = 1.511/2.778/4.309/1.061 ms  
user@tp11-client:~$ ping -c 4 tp11-server  
PING tp11-server.afpa (192.168.42.2) 56(84) bytes of data.  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=1 ttl=64 time=6.39 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=2 ttl=64 time=4.38 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=3 ttl=64 time=1.97 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=4 ttl=64 time=2.13 ms  
  
--- tp11-server.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3007ms  
rtt min/avg/max/mdev = 1.971/3.717/6.390/1.813 ms  
user@tp11-client:~$
```

Figure 28

## 20 Tester le serveur OpenVPN

L'étape suivante est de tester le serveur OpenVPN. Pour commencer, on va donner des droits à l'utilisateur user sur la machine "tp11-client" pour qu'on puisse installer le logiciel openvpn. Sur la machine "tp11-client", on bascule sur un nouveau terminal texte (ici tty3) en appuyant "Ctrl + Alt + F3". Sur le nouveau terminal, on se connecte en tant qu'utilisateur root. Puis on va modifier le fichier sudoers pour ajouter l'utilisateur user dans les droits de permissions. On fait une copie du fichier sudoers en tapant la commande "cp /etc/sudoers /etc/sudoers.ori" pour éviter d'écraser le fichier original. Pour modifier ce fichier, on tape la commande : "nano /etc/sudoers". Dans ce fichier, on ajoute après la ligne du root la commande : "user ALL=(ALL:ALL) ALL". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder avec le même nom. On peut visualiser cette étape sur la Figure 29.

```

tp11-client [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/sudoers
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
user    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

```

Figure 29

On a fini de configurer le fichier sudo. On peut se déconnecter du compte en utilisant la commande "logout". On rebascule sur l'environnement graphique en appuyant "Ctrl + Alt + F1". On ouvre un terminal : on clique sur "Activities" et on tape "Terminal" dans la barre de recherche. Sur le terminal, on tape la commande : "sudo apt install openvpn" pour installer le logiciel openvpn. Ensuite, on va modifier le fichier ".bashrc" qui est un fichier de configuration du shell bash, exécuté à chaque fois qu'on ouvre un terminal interactif (non-login). On va d'abord copier le fichier ".bashrc" avec la commande : "cp .bashrc .bashrc.ori" pour éviter d'écraser le fichier original. Puis on tape : "nano .bashrc" pour l'éditer. Dans le fichier, on va ajouter les chemins "/sbin", "/usr/sbin" et "/usr/local/sbin" à la variable d'environnement PATH (Voir la Figure 30).

```

user@tp11-client: ~
GNU nano 7.2 .bashrc
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

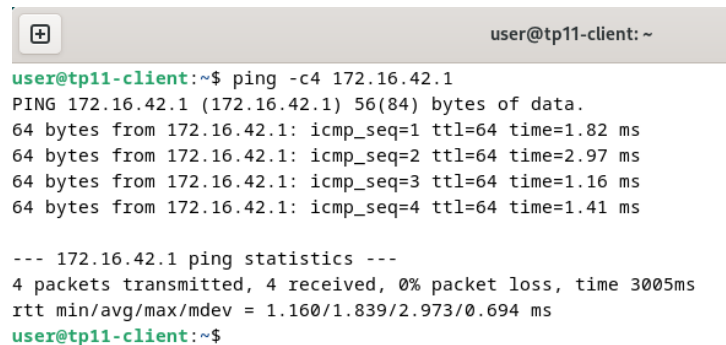
# Add new path for programs to be used by non-root user
export PATH="$PATH:/sbin:/usr/local/sbin"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^N Replace   ^U Paste     ^J Justify  ^_ Go To Line

```

Figure 30

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". Puis on tape la commande : "source .bashrc" pour appliquer les changements faits. On peut vérifier si la variable PATH a bien sauvegardé les chemins en tapant la commande : "echo \$PATH" et on voit que c'est bien le cas. Enfin, on tape donc la commande : "sudo openvpn --remote tp11-server --dev tun0 --ifconfig 172.16.42.2 172.16.42.1". Cette commande lance OpenVPN en mode client, qui va tenter de se connecter au serveur VPN nommé tp11-server, créer une interface tunnel tun0, et configurer le tunnel avec l'adresse IP locale 172.16.42.2 et l'adresse IP distante 172.16.42.1. On ouvre un deuxième terminal et on essaie de joindre la machine côté serveur en utilisant la commande ping 172.16.42.1 et on voit, comme sur la Figure 31, que ça marche. On tape la commande "exit" pour fermer le terminal.



```

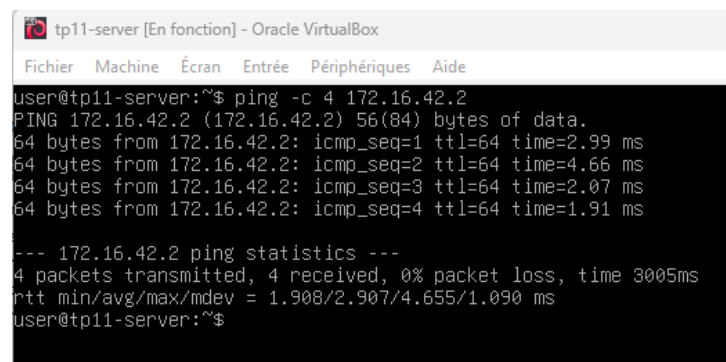
user@tp11-client: ~
user@tp11-client:~$ ping -c 4 172.16.42.1
PING 172.16.42.1 (172.16.42.1) 56(84) bytes of data.
64 bytes from 172.16.42.1: icmp_seq=1 ttl=64 time=1.82 ms
64 bytes from 172.16.42.1: icmp_seq=2 ttl=64 time=2.97 ms
64 bytes from 172.16.42.1: icmp_seq=3 ttl=64 time=1.16 ms
64 bytes from 172.16.42.1: icmp_seq=4 ttl=64 time=1.41 ms

--- 172.16.42.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.160/1.839/2.973/0.694 ms
user@tp11-client:~$

```

Figure 31

On retourne sur la machine "tp11-server", sur le terminal texte tty2 et on essaie de joindre la machine côté client en utilisant la commande ping 172.16.42.2 et on voit, comme sur la Figure 32, que ça marche. On tape la commande "exit" pour fermer le terminal.



```

tp11-server [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
user@tp11-server:~$ ping -c 4 172.16.42.2
PING 172.16.42.2 (172.16.42.2) 56(84) bytes of data.
64 bytes from 172.16.42.2: icmp_seq=1 ttl=64 time=2.99 ms
64 bytes from 172.16.42.2: icmp_seq=2 ttl=64 time=4.66 ms
64 bytes from 172.16.42.2: icmp_seq=3 ttl=64 time=2.07 ms
64 bytes from 172.16.42.2: icmp_seq=4 ttl=64 time=1.91 ms

--- 172.16.42.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.908/2.907/4.655/1.090 ms
user@tp11-server:~$

```

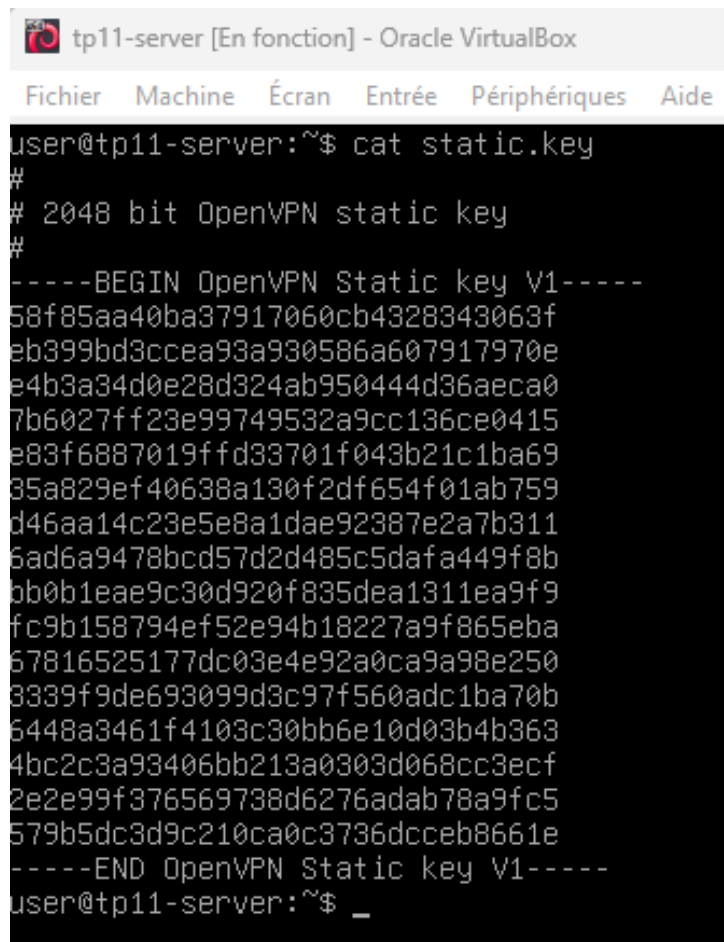
Figure 32

On retourne sur la machine "tp11-client", sur le terminal et on appuie sur "Ctrl + C" pour arrêter le client OpenVPN, puis on tape la commande "exit" pour fermer le terminal.

## 21 Créer une clé statique

Après avoir vérifié que la connexion fonctionne correctement entre le serveur et le client via OpenVPN, l'étape suivante consiste à générer et configurer une clé statique, qui permettra de sécuriser le tunnel VPN de manière simple et symétrique. Sur la machine "tp11-server", on va sur le terminal texte tty2 et on se connecte en tant qu'utilisateur user. Puis on tape la commande : "openvpn --genkey secret static.key"

pour générer une clé statique. On affiche la clé en tapant la commande : "cat static.key" (Voir la Figure 33).



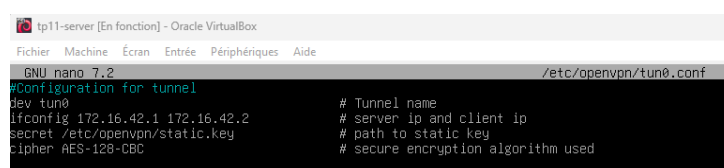
```
tp11-server [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
user@tp11-server:~$ cat static.key
#
# 2048 bit OpenVPN static key
#
-----BEGIN OpenVPN Static key V1-----
58f85aa40ba37917060cb4328343063f
eb399bd3ccea93a930586a607917970e
e4b3a34d0e28d324ab95044d36aeca0
7b6027ff23e99749532a9cc136ce0415
e83f6887019ffd33701f043b21c1ba69
35a829ef40638a130f2df654f01ab759
d46aa14c23e5e8a1dae92387e2a7b311
6ad6a9478bcd57d2d485c5dafa449f8b
bb0b1eae9c30d920f835dea1311ea9f9
fc9b158794ef52e94b18227a9f865eba
67816525177dc03e4e92a0ca9a98e250
3339f9de693099d3c97f560adc1ba70b
6448a3461f4103c30bb6e10d03b4b363
4bc2c3a93406bb213a0303d068cc3ecf
2e2e99f376569738d6276adab78a9fc5
579b5dc3d9c210ca0c3736dcceb8661e
-----END OpenVPN Static key V1-----
user@tp11-server:~$ _
```

Figure 33

On déplace d'abord la clé statique vers le répertoire d'OpenVPN à l'aide de la commande : "sudo mv static.key /etc/openvpn". Ensuite, nous créons le fichier de configuration pour le tunnel tun0 d'OpenVPN en tapant : "sudo nano /etc/openvpn/tun0.conf". Dans ce fichier, on va mettre les lignes suivantes :

- dev tun0
- ifconfig 172.16.42.1 172.16.42.2
- secret /etc/openvpn/static.key
- cipher AES-128-CBC

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur la Figure 34.



```
tp11-server [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 7.2 /etc/openvpn/tun0.conf
#Configuration for tunnel
dev tun0                                # Tunnel name
ifconfig 172.16.42.1 172.16.42.2        # server ip and client ip
secret /etc/openvpn/static.key          # path to static key
cipher AES-128-CBC                      # secure encryption algorithm used
```

Figure 34

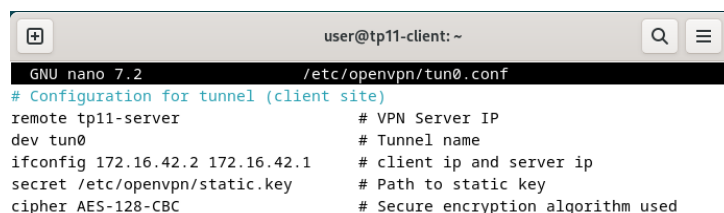
On tape la commande "exit" pour quitter la session, puis on appuie sur "Ctrl + Alt + F1" afin de revenir au terminal texte tty1. Ensuite, on utilise "Ctrl + C" pour arrêter le serveur OpenVPN en cours. Enfin, on lance à nouveau OpenVPN avec la clé statique en tapant : "sudo openvpn --config /etc/openvpn/tun0.conf".

## 22 Tester la clé statique

On retourne sur la machine "tp11-client" et on ouvre un terminal. Sur le terminal, on va récupérer la clé statique depuis la machine "tp11-server" en utilisant la commande scp. On tape donc : "scp user@tp11-server:/etc/openvpn/static.key ." afin de copier la clé statique. Puis, on déplace d'abord la clé statique vers le répertoire d'OpenVPN à l'aide de la commande : "sudo mv static.key /etc/openvpn". Ensuite, comme pour le serveur OpenVPN sur la machine "tp11-server", nous créons le fichier de configuration (côté client) pour le tunnel tun0 d'OpenVPN en tapant : "sudo nano /etc/openvpn/tun0.conf" Dans ce fichier, on va mettre les lignes suivantes :

- dev tun0
- ifconfig 172.16.42.1 172.16.42.2
- secret /etc/openvpn/static.key
- cipher AES-128-CBC

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur la Figure 35.



The screenshot shows a terminal window titled "user@tp11-client: ~". Inside, the GNU nano 7.2 editor is open, editing the file /etc/openvpn/tun0.conf. The content of the file is as follows:

```
# Configuration for tunnel (client site)
remote tp11-server          # VPN Server IP
dev tun0                   # Tunnel name
ifconfig 172.16.42.2 172.16.42.1 # client ip and server ip
secret /etc/openvpn/static.key # Path to static key
cipher AES-128-CBC          # Secure encryption algorithm used
```

Figure 35

Enfin, on lance à nouveau OpenVPN (côté client) avec la clé statique en tapant : "sudo openvpn --config /etc/openvpn/tun0.conf". On appuie sur "Ctrl + C" pour arrêter le client OpenVPN, puis on tape la commande "exit" pour fermer le terminal. On fait de même sur la machine "tp11-server" pour arrêter le serveur OpenVPN.

## 23 Activer TLS

Après avoir sécurisé le tunnel VPN à l'aide d'une clé statique, nous allons mettre en place un chiffrement plus robuste basé sur le protocole **TLS**. Pour cela, nous utiliserons **Easy-RSA**, un outil en ligne de commande permettant de gérer une infrastructure à clé publique (PKI) et de générer les certificats nécessaires. Contrairement à la méthode avec clé statique, le mode TLS permet :

- de mieux authentifier le client et le serveur à l'aide de certificats numériques ;

- d'assurer une meilleure sécurité des échanges grâce à des clés éphémères ;
- de gérer plusieurs clients, chacun avec son propre certificat.

L'objectif est donc de :

1. créer une autorité de certification (CA) locale ;
2. générer les certificats et clés pour le serveur OpenVPN ;
3. configurer le serveur pour qu'il utilise ces certificats avec le protocole TLS.

L'outil Easy-RSA facilitera la création de tous ces éléments, en particulier :

- la clé privée et le certificat du serveur ;
- le fichier `ca.crt` (certificat de l'autorité) ;
- le fichier `ta.key` (clé de protection TLS supplémentaire, optionnelle mais recommandée).

Une fois cette configuration terminée, le serveur OpenVPN pourra établir des connexions TLS sécurisées avec les clients disposant de certificats valides. On retourne sur la machine "tp11-server" et on installe l'easy-rsa en tapant la commande : "sudo apt install easy-rsa". On va ensuite créer un dossier PKI (Public Key Infrastructure) en tapant la commande : "make-cadir /easy-rsa" et on se place dans le dossier avec la commande "cd /easy-rsa". Puis, nous passons aux étapes de génération des certificats serveur :

1. Initialiser la PKI :

```
./easyrsa init-pki
```

2. Créer une autorité de certification (CA) locale :

```
./easyrsa build-ca
```

Cette commande génère une clé privée pour la CA ainsi que le certificat (ici CA key et PEM phrase : `azerty`) `ca.crt`.

3. Générer une requête de certificat (CSR) pour le serveur :

```
./easyrsa gen-req server nopass
```

Cela crée une clé privée `server.key` et une requête `server.req`.

4. Signer la requête du serveur avec la CA :

```
./easyrsa sign-req server server
```



Le fichier `server.crt` est généré, c'est le certificat signé du serveur (ici pass phrase : `azerty`).

#### 5. Générer un fichier de paramètres Diffie-Hellman :

```
./easyrsa gen-dh
```

Cela produit le fichier `dh.pem`, utilisé pour établir une clé de session sécurisée.

Les fichiers importants à conserver côté serveur dans `/etc/openvpn` sont :

- `ca.crt` : le certificat de l'autorité de certification,
- `server.crt` : le certificat du serveur,
- `server.key` : la clé privée du serveur,
- `dh.pem` : les paramètres Diffie-Hellman.

On va créer le fichier de configuration côté serveur "`server.conf`". On tape la commande : "`sudo nano /etc/openvpn/server.conf`". Dans ce fichier, on met les lignes :

- `port 1194`
- `proto udp`
- `dev tun`
- `tls-server`
- `server 172.16.42.0 255.255.255.0`
- 
- `ca ca.crt`
- `cert server.crt`
- `key server.key`
- `dh dh.pem`
- 
- `cipher AES-256-CBC`
- `auth SHA256`
- `data-ciphers AES-256-CBC`

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur la Figure 36.

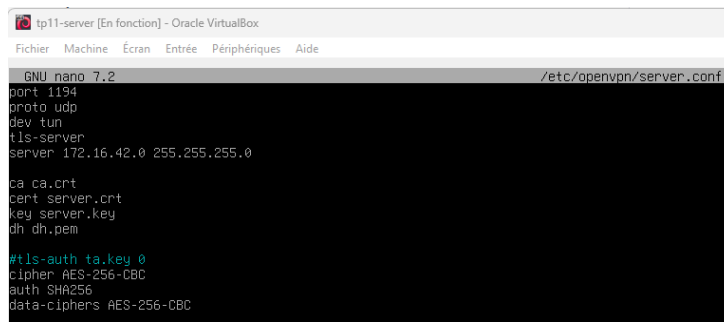


Figure 36

La prochaine étape est de lancer le service openvpn. On tape la commande : "sudo systemctl start openvpn@server". On vérifie si le serveur Openvpn marche correctement en tapant la commande "sudo systemctl status openvpn@server" et c'est bien le cas.

## 24 Tester TLS

Maintenant on va faire de même côté client. Avant de retourner sur la machine "tp11-client", on va générer la clé et le certificat sur le serveur. On est sur la machine "tp11-server" et on tape la commande "cd /easy-rsa" pour aller dans le dossier "easy-rsa". Puis on tape les commandes :

- Générer une requête de certificat (CSR) pour le client :

```
./easyrsa gen-req client1 nopass
```

Cela crée une clé privée `client.key` et une requête `client.req`.

- Signer la requête du client avec la CA :

```
./easyrsa sign-req client client1
```

Le fichier `client.crt` est généré, c'est le certificat signé du client (ici pass phrase : azerty).

On retourne sur la machine "tp11-client" et on ouvre un terminal. On crée le fichier de configuration côté client "client.conf". On tape la commande : "sudo nano /etc/openvpn/client.conf". Dans ce fichier, on met les lignes :

- client
- dev tun
- proto udp
- remote tp11-server 1194

- tls-client
- 
- ca /etc/openvpn/ca.crt
- cert /etc/openvpn/client.crt
- key /etc/openvpn/client.key
- 
- cipher AES-256-CBC
- auth SHA256
- data-ciphers AES-256-CBC
- resolv-retry infinite
- nobind
- persist-key
- persist-tun
- verb 3

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir les étapes précédentes sur la Figure 37.



```

user@tp11-client: ~
GNU nano 7.2 /etc/openvpn/client.conf
client
dev tun
proto udp
remote tp11-server 1194
tls-client

ca /etc/openvpn/ca.crt
cert /etc/openvpn/client.crt
key /etc/openvpn/client.key

cipher AES-256-CBC
data-ciphers AES-256-CBC
auth SHA256
resolv-retry infinite
nobind
persist-key
persist-tun
verb 3

```

Figure 37

Puis, on va récupérer les fichiers "ca.crt", "client.crt" et "client.key". Pour cela, on va utiliser la commande "scp". On récupère les fichiers depuis la machine "tp11-server" et on les sauvegarde dans le dossier openvpn sur la machine "tp11-client" :

- scp user@tp11-server:/home/user/easy-rsa/pki/ca.crt /etc/openvpn/
- scp user@tp11-server:/home/user/easy-rsa/pki/issued/client1.crt /etc/openvpn/client.crt

- `scp user@tp11-server:/home/user/easy-rsa/pki/private/client1.key /etc/openvpn/client.key`

Puis, on tape la commande : `"sudo chmod 600 /etc/openvpn/client.key"` pour que seul l'utilisateur root ait accès à ce fichier. Enfin, on lance le service openvpn. On tape la commande : `"sudo systemctl start openvpn@client"`. On vérifie si le serveur Openvpn marche correctement en tapant la commande `"sudo systemctl status openvpn@client"` et c'est bien le cas (voir la Figure 38).

```

user@tp11-client:~$ sudo systemctl status openvpn@client
● openvpn@client.service - OpenVPN connection to client
   Loaded: loaded (/lib/systemd/system/openvpn@.service; enabled-runtime; preset: enabled)
   Active: active (running) since Thu 2025-07-31 18:32:28 CEST; 22min ago
     Docs: man:openvpn(8)
           https://community.openvpn.net/openvpn/wiki/Openvpn24ManPage
           https://community.openvpn.net/openvpn/wiki/HOWTO
   Main PID: 8070 (openvpn)
   Status: "Initialization Sequence Completed"
     Tasks: 1 (limit: 10)
    Memory: 2.3M
       CPU: 110ms
    CGroup: /system.slice/system-openvpn.slice/openvpn@client.service
           └─8070 /usr/sbin/openvpn --daemon ovpn-client --status /run/openvpn/client.status 10 --cd /etc/op

Jul 31 18:54:08 tp11-client ovpn-client[8070]: TLS: tls_multi_process: initial untrusted session promoted to t
Jul 31 18:54:08 tp11-client ovpn-client[8070]: PUSH: Received control message: 'PUSH_REPLY,route 172.16.42.1,t
Jul 31 18:54:08 tp11-client ovpn-client[8070]: OPTIONS IMPORT: --ifconfig/up options modified
Jul 31 18:54:08 tp11-client ovpn-client[8070]: OPTIONS IMPORT: route options modified
Jul 31 18:54:08 tp11-client ovpn-client[8070]: OPTIONS IMPORT: tun-mtu set to 1500
Jul 31 18:54:08 tp11-client ovpn-client[8070]: Preserving previous TUN/TAP instance: tun0
Jul 31 18:54:08 tp11-client ovpn-client[8070]: Initialization Sequence Completed
Jul 31 18:54:08 tp11-client ovpn-client[8070]: Data Channel: cipher 'AES-256-CBC', auth 'SHA256', peer-id: 0
Jul 31 18:54:08 tp11-client ovpn-client[8070]: Timers: ping-restart 120
Jul 31 18:54:08 tp11-client ovpn-client[8070]: Protocol options: protocol-flags cc-exit tls-ekm dyn-tls-crypt
lines 1-24/24 (END)

```

Figure 38

Maintenant que la connexion est faite, on peut essayer de faire communiquer le client et le serveur entre eux. Sur le client, on tape la commande : `"ping -c4 172.16.42.1"` et on voit comme sur la Figure 39 que le serveur répond. On fait de même sur le serveur en tapant la commande : `"ping -c4 172.16.42.6"` et on voit comme sur la Figure 40 que le client répond bien.

```

user@tp11-client:~$ ping -c4 172.16.42.1
PING 172.16.42.1 (172.16.42.1) 56(84) bytes of data:
64 bytes from 172.16.42.1: icmp_seq=1 ttl=64 time=1.16 ms
64 bytes from 172.16.42.1: icmp_seq=2 ttl=64 time=1.33 ms
64 bytes from 172.16.42.1: icmp_seq=3 ttl=64 time=1.26 ms
64 bytes from 172.16.42.1: icmp_seq=4 ttl=64 time=1.77 ms

--- 172.16.42.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.161/1.381/1.772/0.233 ms
user@tp11-client:~$

```

Figure 39

```

tp11-server [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

user@tp11-server:~$ ping -c4 172.16.42.6
PING 172.16.42.6 (172.16.42.6) 56(84) bytes of data:
64 bytes from 172.16.42.6: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 172.16.42.6: icmp_seq=2 ttl=64 time=1.42 ms
64 bytes from 172.16.42.6: icmp_seq=3 ttl=64 time=2.20 ms
64 bytes from 172.16.42.6: icmp_seq=4 ttl=64 time=1.41 ms

--- 172.16.42.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.413/1.694/2.198/0.320 ms
user@tp11-server:~$ _

```

Figure 40

Remarque : Quand on regarde les adresses sur le serveur pour le tun0 on voit : `"inet 172.16.42.1 peer 172.16.42.2/32 scope global tun0"` et quand on regarde chez le client pour le tun0 on voit : `"inet 172.16.42.6 peer 172.16.42.5/32 scope global tun0"`. Le serveur a l'IP 172.16.42.1, et s'attend à ce que son pair direct (point-to-point) soit 172.16.42.2. Le client, lui, a une adresse différente : 172.16.42.6 / peer 172.16.42.5. Or, dans une config de type "subnet", le serveur assigne les IPs à ses clients dans une plage, mais n'utilise pas de pair point-to-point explicite. C'est OpenVPN qui gère ça en interne. D'où la raison que le client peut ne pas avoir l'adresse attendue.

Ainsi, on a créé un tunnel avec chiffrement robuste et asymétrique pour permettre à 2 machines de se communiquer comme si elles étaient dans le même réseau.