

# TSSR - Compte rendu TP10 : Serveur Squid Sous Debian

Jean-Paul MELISSE

20 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 Squid sous Debian. Un serveur Squid est un logiciel de proxy open-source principalement utilisé pour la mise en cache de contenu web et la gestion de la bande passante. Il est souvent utilisé dans les réseaux d'entreprise ou les fournisseurs d'accès Internet (FAI) pour améliorer les performances d'accès à Internet et réduire la charge sur les connexions réseau. 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 : "C:\Users\Jean-Paul\Desktop\tp10\", comme montre la Figure 1.

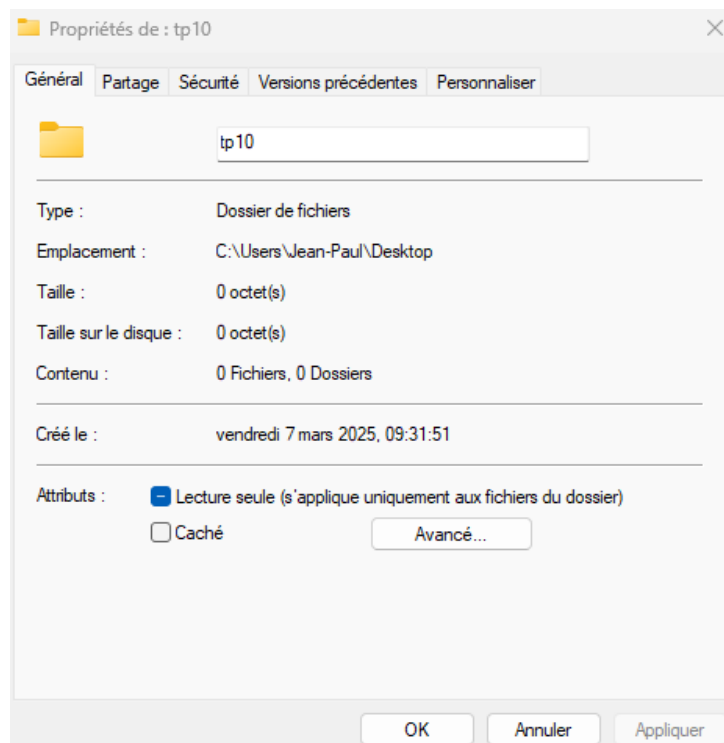


Figure 1

### 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 tp10 sous le format .iso (Voir la Figure 2).

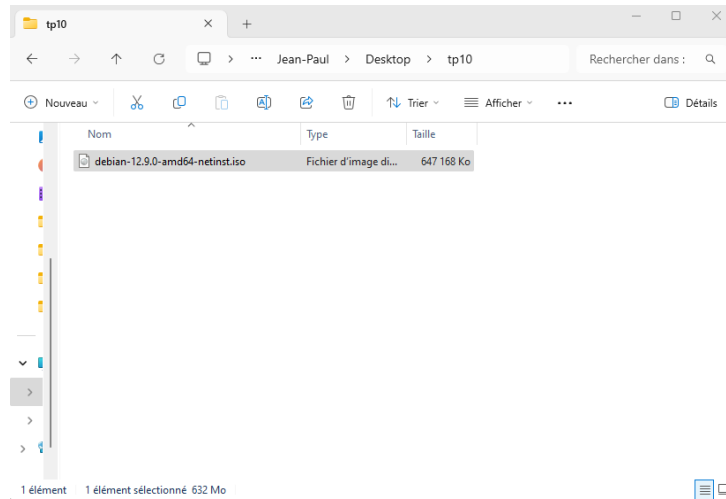


Figure 2

### 4 Créer la machine virtuelle tp10-router

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp10-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).

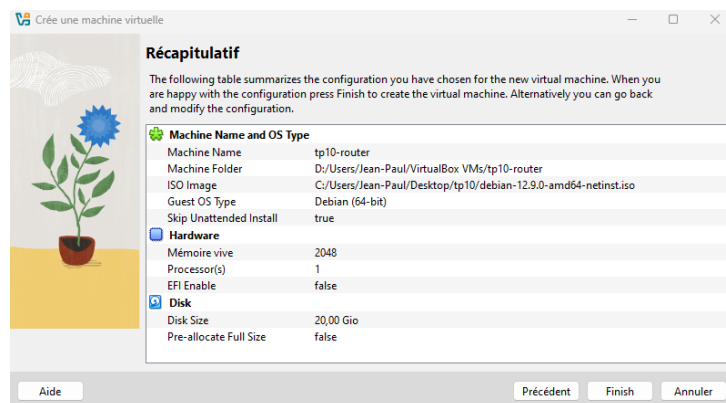


Figure 3

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 tp10-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 "tp10-router" et on clique sur "Continue" (Figure 4).

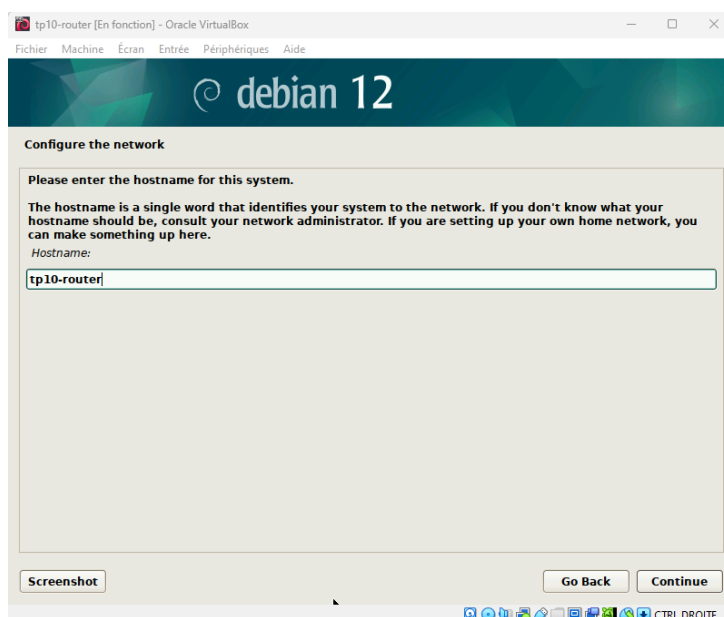


Figure 4

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

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".

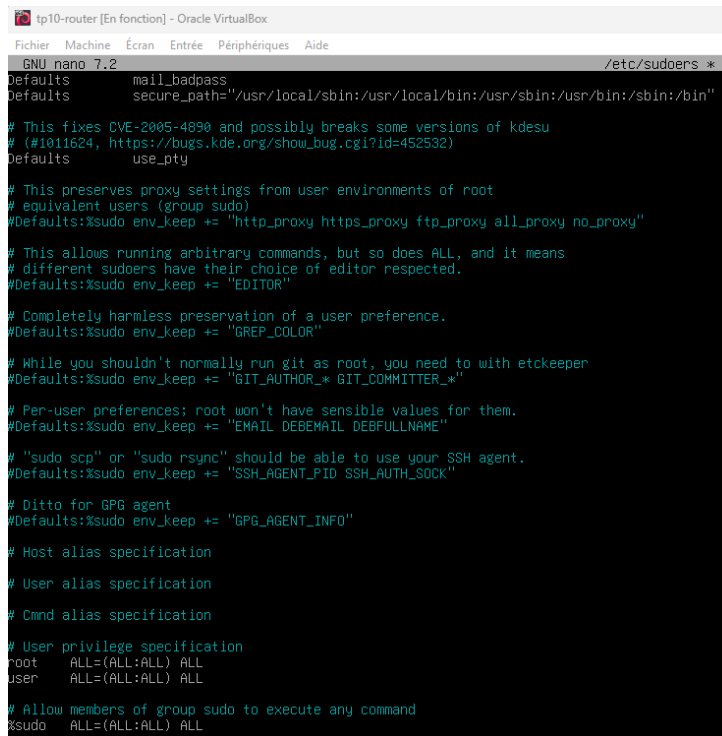
## 6 Installer les paquets nécessaires sur tp10-router

```

root@tp10-router:~# apt install bind9 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 bsdxtrautils dns-root-data groff-base libfstrm0 libgdbm6 libip6tc2 libjemalloc2 liblmdb0 libmaxminddb0 libnetfilter-contrack3
  libnetfilterlnk0 libnetfilter-t2-14 libopenl1n0 libprotobuf-c1 libuchardet0 libuv1 netfilter-persistent policycoreutils selinux-utils
Suggested packages:
  bind-doc dnsutils resolvconf ufw groff firewall0 policykit-1 isc-dhcp-server-ldap ieee-data gdm-l10n mmdb-bin www-browser
The following NEW packages will be installed:
  bind9 bind9-libs bind9-utils bsdxtrautils dns-root-data groff-base iptables-persistent isc-dhcp-server libfstrm0 libgdbm6 libip6tc2 libjemalloc2
  liblmdb0 libmaxminddb0 libnetfilter-contrack3 libnetfilterlnk0 libnetfilter-t2-14 libopenl1n0 libprotobuf-c1 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 8,872 kB of archives.
After this operation, 31.3 MB of additional disk space will be used.
Do you want to continue? [Y/n]

```

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.



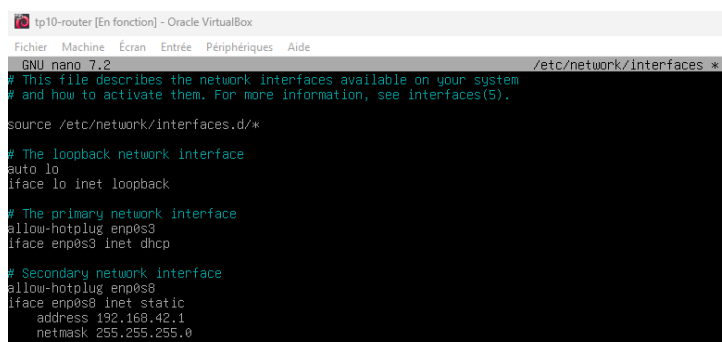
```
tp10-router [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 /etc/sudoers *
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#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 += "GPG_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 tp10-router

On retourne sur la machine "tp10-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)



```
tp10-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
```

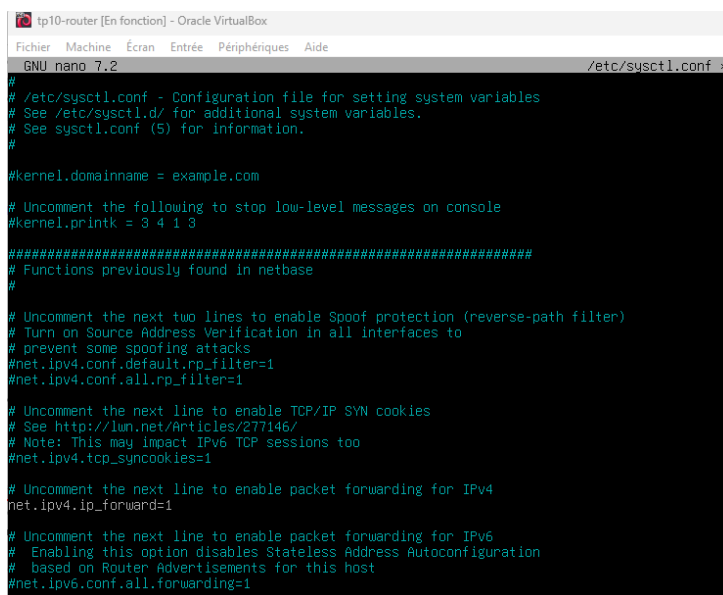
Figure 7

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 tp10-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 (Voir la Figure 8).



```
tp10-router [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 /etc/sysctl.conf *
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.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
```

Figure 8

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 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 tp10-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 tp10-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.

```

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 tp10-router et du tp10-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.

```

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

```

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

@ IN NS ns1.afpa.

ns1.afpa. IN A 192.168.42.1
tp10-router.afpa. IN A 192.168.42.1
tp10-server.afpa. IN A 192.168.42.2

```

Figure 11

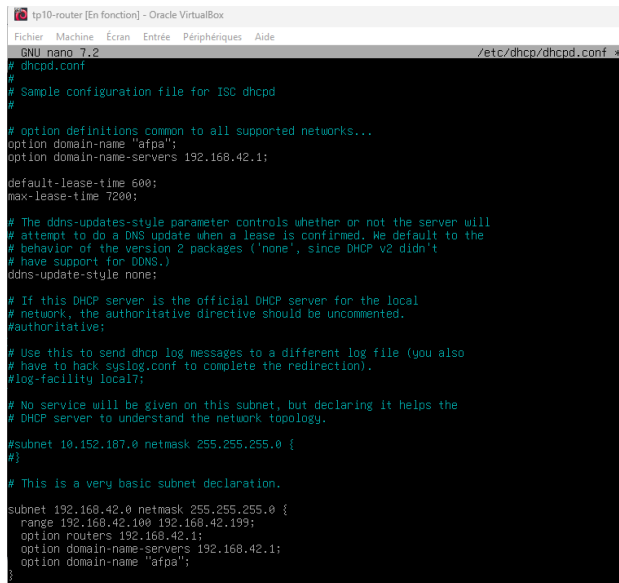
On peut vérifier si le fichier de configuration est bon en tapant la commande : "sudo named-checkconf" et "sudo named-checkzone afpa /etc/bind/db.afpa". Si tout est OK, on redémarre le service bind en tapant : "sudo service bind9 restart". On redémarre la configuration réseau de la machine en utilisant 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 tp10-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.)
ddns-update-style none;

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

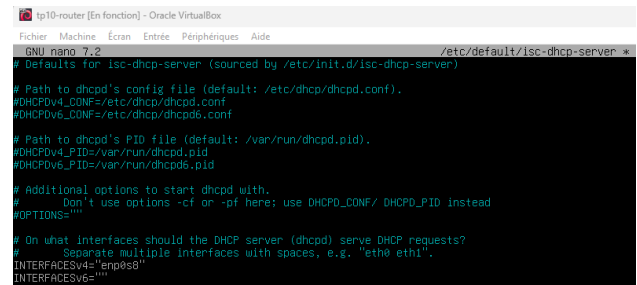
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
#subnet 10.152.187.0 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).
DHCPDv4_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 tp10-server

On retourne sur VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp10-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).

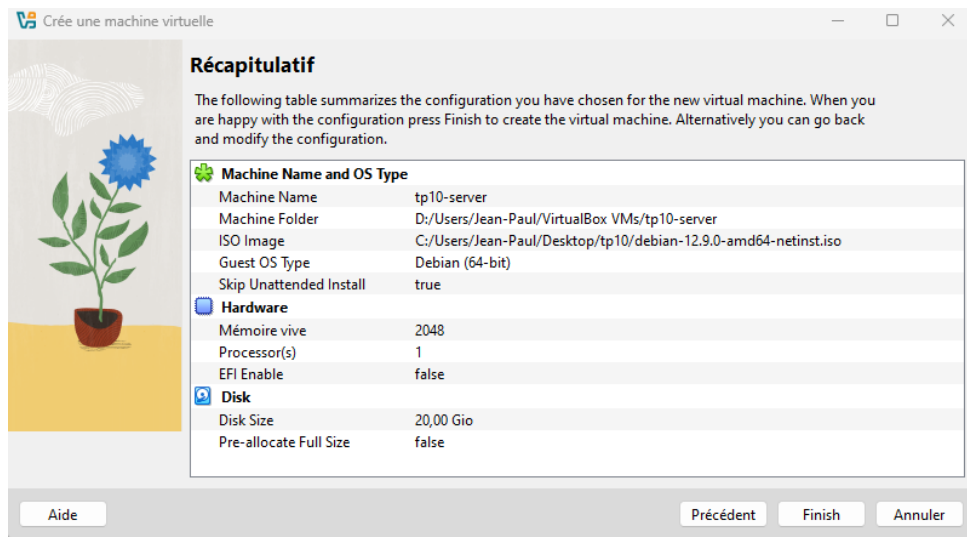


Figure 14

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 tp10-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 "tp10-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.

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 squid man sudo" pour installer le paquet manuel, la commande sudo et le paquet squid qui est essentiel pour le proxy. (Voir la Figure 16).



```

tp10-server [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 /etc/sudoers *
Defaults mail_badpass
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

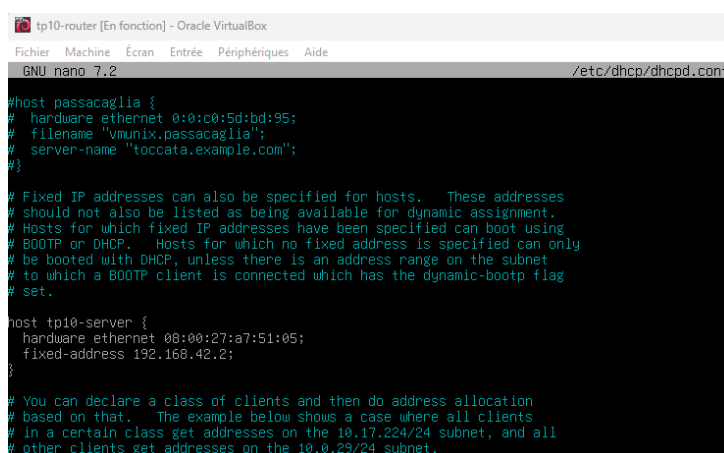
```

12

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

## 14 Configurer le réseau de tp10-server

On retourne sur la machine "tp10-router". On va faire une réservation d'adresse IP pour la machine "tp10-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 "tp10-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 "tp10-server". Pour cela, sur la machine "tp10-server", on tape la commande : "ip a".



```
tp10-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.

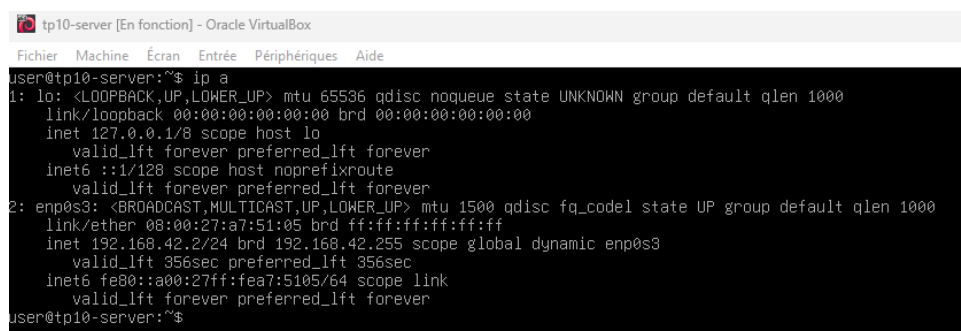
host tp10-server {
    hardware ethernet 08:00:27:a7:51:05;
    fixed-address 192.168.42.2;
}

# 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 "tp10-server" et on se connecte en tant qu'utilisateur user. 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.



```
tp10-server [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
user@tp10-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
    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:a7:51:05 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.2/24 brd 192.168.42.255 scope global dynamic enp0s3
        valid_lft 356sec preferred_lft 356sec
    inet6 fe80::a00:27ff:fea7:5105/64 scope link
        valid_lft forever preferred_lft forever
user@tp10-server:~$
```

Figure 19

## 15 Créer la machine virtuelle tp10-client

On retourne sur VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp10-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 20).

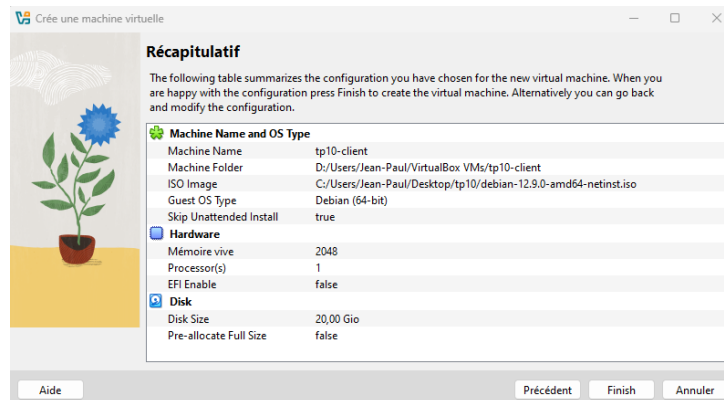


Figure 20

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.

## 16 Installer Debian sur tp10-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 "tp10-client" et on clique sur "Continue" (Figure 21).



Figure 21

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".

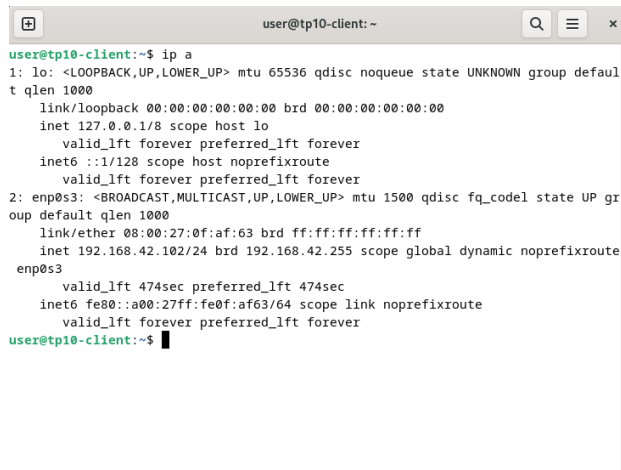
On choisit l'option "Guide - use entire disk" pour la partition du disque et on clique sur "Continue" deux fois. Pour la partition, on choisit "All files in one partition" et on clique sur "Continue" deux fois. On coche la case "Yes" pour appliquer les changements sur la partition du disque et on clique 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.

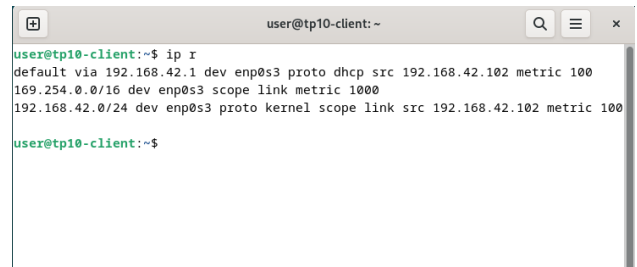
## 17 Tester le serveur DHCP

Après le redémarrage de notre machine "tp10-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 22). On tape ensuite la commande : "ip r" pour voir l'adresse IP de la passerelle par défaut et on voit, comme dans la Figure 23, qu'on a bien récupéré la bonne adresse IP de notre routeur "tp10-router" comme passerelle.



```
user@tp10-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 08:00:27:0f:af:63 brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.102/24 brd 192.168.42.255 scope global dynamic noprefixroute
        valid_lft 474sec preferred_lft 474sec
    inet6 fe80::a00:27ff:fe0f:af63/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
user@tp10-client:~$
```

Figure 22



```
user@tp10-client:~$ ip r
default via 192.168.42.1 dev enp0s3 proto dhcp src 192.168.42.102 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 192.168.42.102 metric 100
user@tp10-client:~$
```

Figure 23

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 24).



```
user@tp10-client:~$ cat /etc/resolv.conf
# Generated by NetworkManager
search afpa
nameserver 192.168.42.1
user@tp10-client:~$
```

Figure 24

## 18 Tester le serveur DNS

La prochaine étape est de vérifier si le serveur DNS fonctionne. Sur le terminal de la machine "tp10-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 25.



```
user@tp10-client:~  
user@tp10-client:~$ ping -c4 google.fr  
PING google.fr (216.58.214.163) 56(84) bytes of data.  
64 bytes from parl0s42-in-f3.1e100.net (216.58.214.163): icmp_seq=1 ttl=113 time=16.1 ms  
64 bytes from mad01s26-in-f163.1e100.net (216.58.214.163): icmp_seq=2 ttl=113 time=17.7 ms  
64 bytes from mad01s26-in-f3.1e100.net (216.58.214.163): icmp_seq=3 ttl=113 time=61.0 ms  
64 bytes from mad01s26-in-f3.1e100.net (216.58.214.163): icmp_seq=4 ttl=113 time=34.7 ms  
  
--- google.fr ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3004ms  
rtt min/avg/max/mdev = 16.086/32.379/61.042/18.090 ms  
user@tp10-client:~$ ping -c4 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=1.25 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=2 ttl=64 time=0.833 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=3 ttl=64 time=1.88 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=4 ttl=64 time=1.02 ms  
  
--- ns1.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3005ms  
rtt min/avg/max/mdev = 0.833/1.245/1.879/0.395 ms  
user@tp10-client:~$
```

Figure 25

On va vérifier maintenant les enregistrements des adresses IP du serveur et du routeur par le serveur DNS. On tape la commande : "ping tp10-router" pour voir si on peut interagir avec la machine "tp10-router" et c'est bien le cas. On fait de même avec la machine "tp10-server" en tapant la commande : "ping tp10-server" et on constate la même chose. On peut voir les étapes précédentes sur la Figure 26.

```
user@tp10-client:~  
user@tp10-client:~$ ping -c4 tp10-router  
PING tp10-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=0.819 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=2 ttl=64 time=1.14 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=3 ttl=64 time=0.913 ms  
64 bytes from _gateway (192.168.42.1): icmp_seq=4 ttl=64 time=3.62 ms  
  
--- tp10-router.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3010ms  
rtt min/avg/max/mdev = 0.819/1.622/3.619/1.158 ms  
user@tp10-client:~$ ping -c4 tp10-server  
PING tp10-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=1.79 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=2 ttl=64 time=0.851 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=3 ttl=64 time=0.800 ms  
64 bytes from 192.168.42.2 (192.168.42.2): icmp_seq=4 ttl=64 time=0.919 ms  
  
--- tp10-server.afpa ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3006ms  
rtt min/avg/max/mdev = 0.800/1.090/1.791/0.406 ms  
user@tp10-client:~$
```

Figure 26

## 19 Tester le serveur Squid

L'étape suivante est de tester le serveur Squid. Pour cela, on va ajouter un proxy à notre navigateur "Firefox". Sur la machine "tp10-client", on ouvre le navigateur Firefox: on clique sur "Activities" et on clique sur "Firefox". Sur Firefox, on clique sur "Open application menu", puis sur "Settings". Sur l'onglet "General", on descend jusqu'au chapitre "Network Settings" et on clique sur "Settings...". On coche la case "Manual proxy configuration..." et on entre "tp10-server" dans HTTP Proxy et le port 3128 (Attention : par défaut, Squid écoute sur le port 3128, faut vérifier dans le fichier squid.conf). On coche aussi la case "Also use this proxy for HTTPS", puis on clique sur "OK" (Voir l'étape précédente sur la Figure 27).

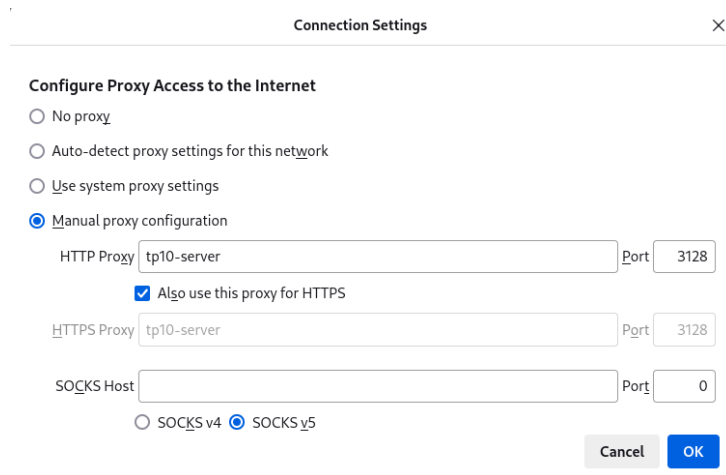


Figure 27

On ouvre un nouvel onglet sur Firefox et sur la barre de recherche, on entre l'adresse de Google, soit `https://google.fr/` et on appuie sur "Entrée". On constate, comme sur la Figure 28, que le proxy Squid refuse bien la connexion vers la page web Google.

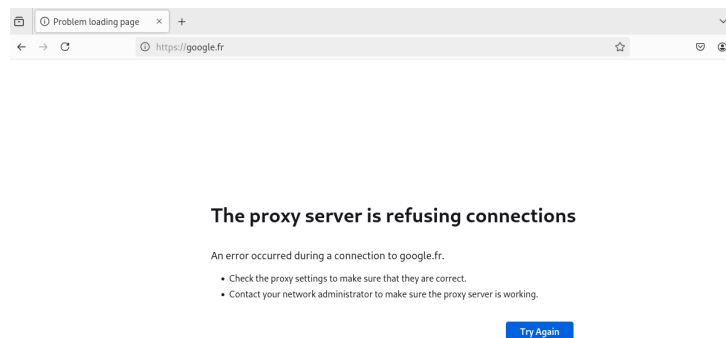
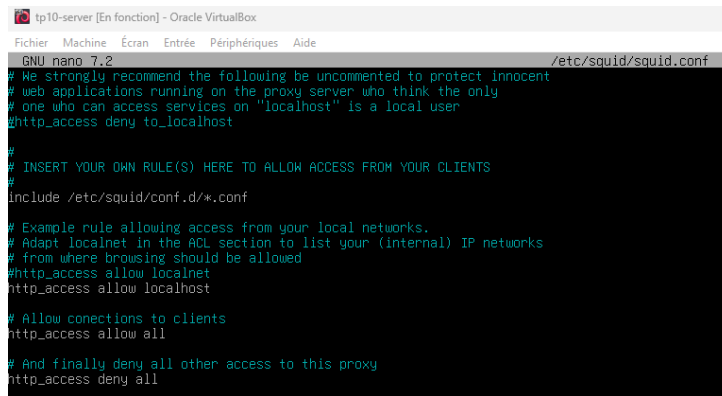


Figure 28

## 20 Accepter des clients avec Squid

On retourne sur la machine "tp10-server". Le but est de modifier la configuration de Squid afin de pouvoir autoriser l'accès HTTP à tous les clients. On copie le fichier `squid.conf` afin de pouvoir modifier une copie pour ne pas écraser le fichier original en faisant la commande : `sudo cp /etc/squid/squid.conf /etc/squid/squid.ori`. Puis, on édite le fichier `squid.conf` avec la commande : `sudo nano /etc/squid/squid.conf`. Dans ce fichier, on va chercher les lignes `"http_access deny all"` et `"http_access allow localnet"` (localnet ou localhost). Puis on ajoute la ligne `"http_access allow all"` avant la ligne `"http_access deny all"` pour autoriser l'accès à tout le monde. 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 29.



```
tp10-server [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

GNU nano 7.2 /etc/squid/squid.conf
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*.conf

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# Allow connections to clients
http_access allow all

# And finally deny all other access to this proxy
http_access deny all
```

Figure 29

On tape la commande : "sudo systemctl restart squid" pour redémarrer le service squid. On vérifie si le service squid marche correctement en tapant la commande "sudo systemctl status squid" et c'est bien le cas.

## 21 Tester l'acceptation de clients

On retourne sur la machine "tp10-client" et on ouvre Firefox. Sur la barre de recherche, on entre l'adresse de Google, soit <https://google.com/> et on appuie sur "Entrée". On constate, comme sur la Figure 30, que le proxy Squid accepte bien la connexion vers la page web Google.

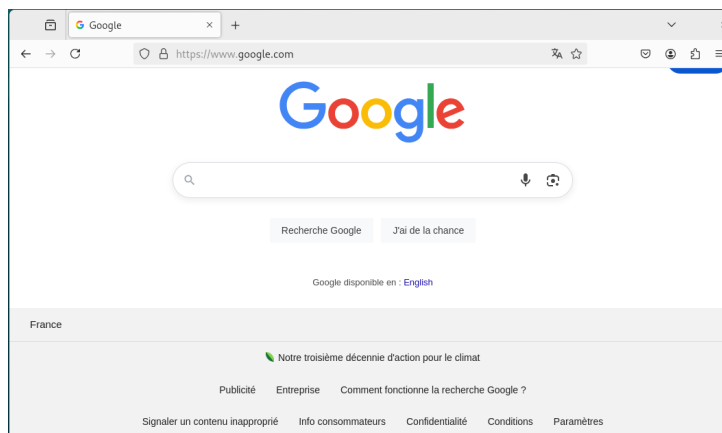
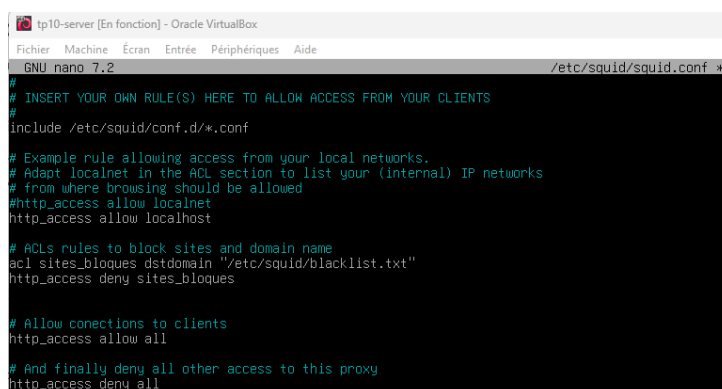


Figure 30

## 22 Refuser un nom de domaine avec Squid

La prochaine étape maintenant est de refuser un nom de domaine. On retourne sur la machine "tp10-server" et on va créer un nouveau fichier texte qui contiendra la liste des sites qu'on veut bloquer. On crée le fichier "blacklist.txt" en tapant la commande : "sudo nano /etc/squid/blacklist.txt". Dans ce fichier, on met la ligne ".google.com" pour bloquer le nom de domaine Google. On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". Puis on modifie le fichier squid.conf en tapant la commande : "sudo nano /etc/squid/squid.conf". Dans ce fichier, on va ajouter la ligne "acl sites\_bloques dstdomain "/etc/squid/blacklist.txt" " et la ligne "http\_access deny sites\_bloques" juste avant la ligne "http\_access deny all" (Squid lit les instructions d'ordre - haut vers le bas, les premières

lues seront prioritaires). On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée". On peut voir cette étape sur la Figure 31.



```
tp10-server [En fonction] - Oracle VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide
GNU nano 7.2 /etc/squid/squid.conf
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
include /etc/squid/conf.d/*.conf

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localhost

# ACLs rules to block sites and domain name
acl sites_bloques dstdomain "/etc/squid/blacklist.txt"
http_access deny sites_bloques

# Allow connections to clients
http_access allow all

# And finally deny all other access to this proxy
http_access deny all
```

Figure 31

## 23 Tester le refus du nom de domaine

On retourne sur la machine "tp10-client" et sur Firefox. On voit, comme sur les Figures 32 et 33, que le nom de domaine google.com est bloqué alors que les autres ne le sont pas (wikipedia.org par exemple).

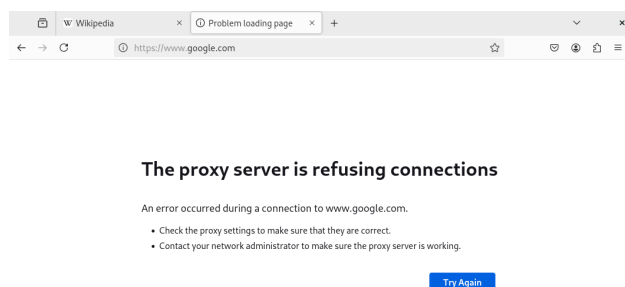


Figure 32

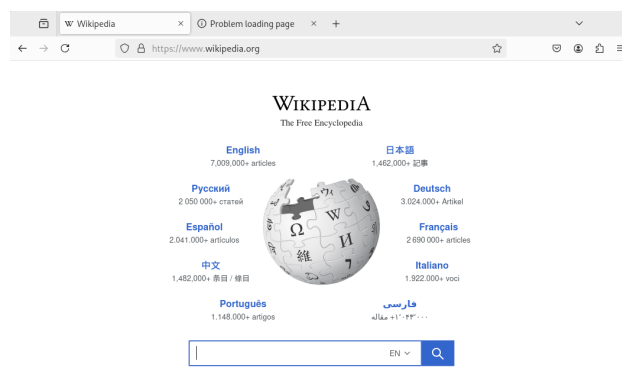


Figure 33

Ainsi avec Squid, on a pu voir comment utiliser un proxy pour bloquer ou autoriser les sites.