

TSSR - Compte rendu TP4 : Serveur Nginx Sous Debian

Jean-Paul MELISSE

09 Janvier 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 Nginx sous Debian. Ce dernier est un logiciel serveur web open-source qui permet de gérer un grand nombre de connexions simultanées. Il peut aussi être utilisé comme un reverse proxy, un équilibreur de charge (load balancer), et un serveur de cache. On utilisera l'hyperviseur VirtualBox pour créer 2 machines virtuelles ; une pour faire tourner notre serveur Nginx et l'autre pour faire tourner un client Windows. 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\tp4\", comme montre la Figure 1.

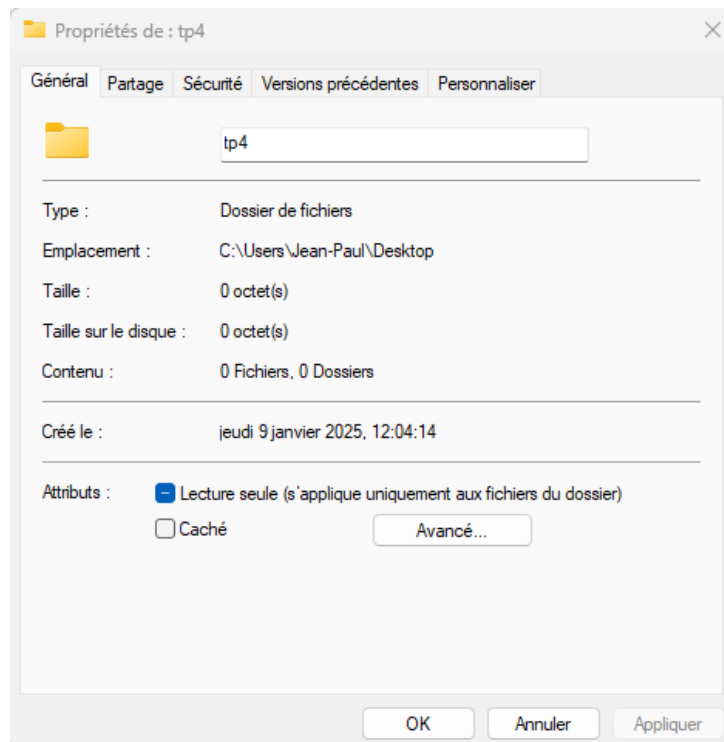


Figure 1

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 aussi dans le dossier tp4 sous le format .iso (Comme montre la Figure 2).

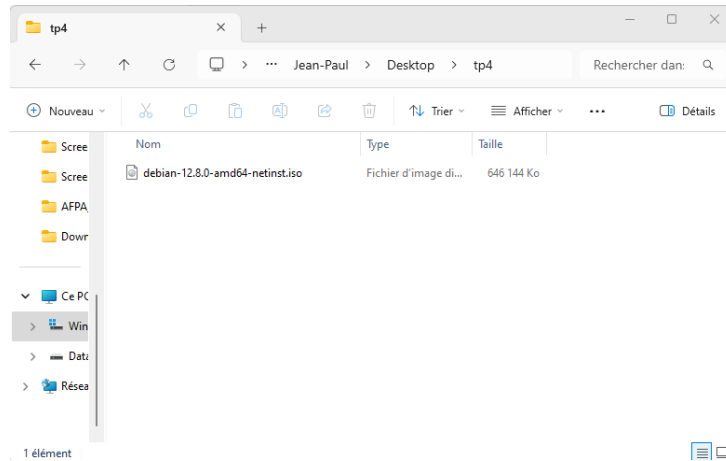


Figure 2

4 Télécharger Windows

Dans ce TP, on aura besoin aussi d'un système d'exploitation Windows. On va donc le télécharger via le lien <https://www.microsoft.com/en-us/software-download/windows11>. On choisit "Download Windows 11 Disk Image (ISO) for x64 devices" et l'option "Windows 11 (multi-edition ISO for x64 devices)" et on clique sur le bouton "Download Now". On choisit la langue "English (United States)" et on clique sur "Confirm". Enfin, on clique sur "64-bit Download" pour le télécharger. On le sauvegarde dans le dossier tp4 sous le format .iso (Voir la Figure 3).

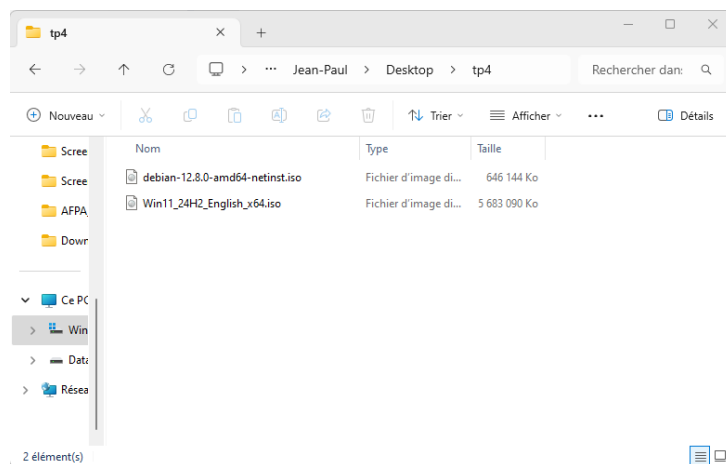


Figure 3

5 Créer la machine virtuelle tp4-debian

On ouvre VirtualBox. On clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp4-debian" 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 4).

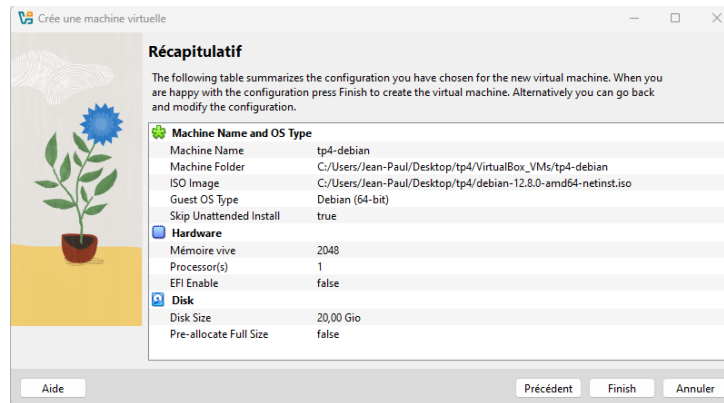


Figure 4

Maintenant que la nouvelle machine est créée, il faut la configurer. On clique sur "Configurations". On clique sur "Réseau" et 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. Maintenant qu'on a configuré la machine virtuelle, on clique sur "Démarrer" pour l'allumer.

6 Installer Debian

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 "tp4-debian" et on clique sur "Continue" (Figure 5).

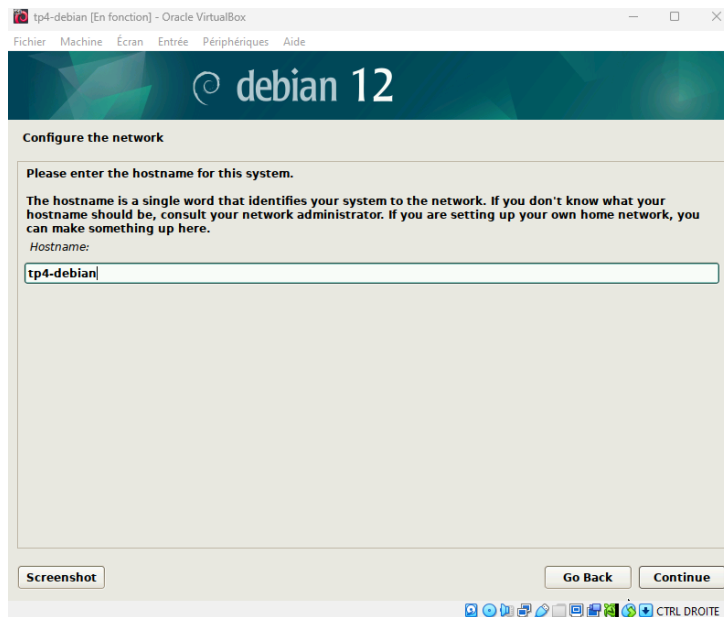


Figure 5

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 "afpa" et on clique sur "Continue" deux fois. Comme pour l'administrateur root, on donne un mot de passe à l'utilisateur afpa (dans ce cas ici, le mot de passe est "afpa" 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.

7 Installer les paquets nécessaires

Le but ici est d'installer les paquets nécessaires pour le bon fonctionnement de notre serveur Nginx. Après le redémarrage de notre machine virtuelle, on se connecte avec les identifiants de l'utilisateur root. On tape la commande "apt install nginx openssl man sudo" pour installer le paquet manuel, la commande sudo, le paquet nginx pour le serveur web et le paquet openssl pour intégrer dans le serveur les connexions HTTPs. (Voir la Figure 6).

```
Debian GNU/Linux 12 tp4-debian tty1
tp4-debian login: root
Password:
Linux tp4-debian 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) x86_64

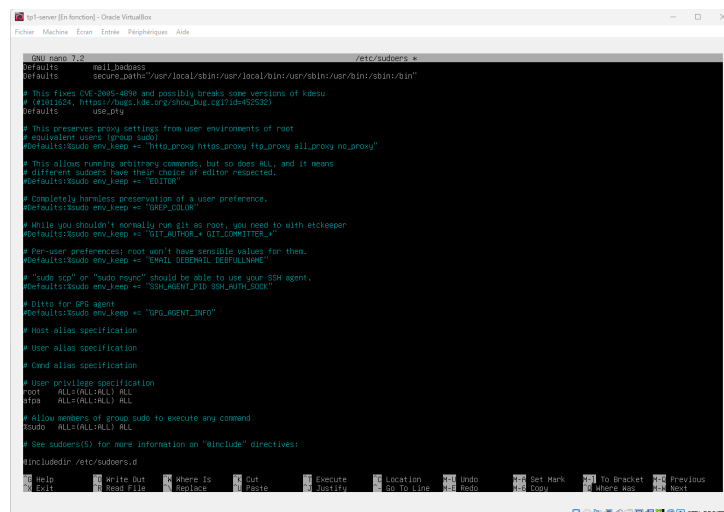
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@tp4-debian:~# apt install nginx openssl man sudo_
```

Figure 6

8 Configurer sudo

Le suite maintenant est de configurer la commande sudo afin qu'on puisse l'utiliser avec l'utilisateur afpa. Après avoir installé sudo, on va modifier le fichier sudoers pour ajouter l'utilisateur afpa 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 : "afpa 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 7.



```
# nano /etc/sudoers
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
# This line is REQUIRED and breaks a lot of old scripts
# including /etc/passwd and /etc/shadow.
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_AGENT_INFO"
# While you shouldn't normally run git as root, you need to with etckeeper
Defaults:sudo env_keep += "GIT_AUTHOR_EMAIL GIT_COMMITTER_EMAIL"
# For user preferences; root won't have sensible values for them.
Defaults:sudo env_keep += "EMAIL DESKTOP DESKTOPNAME"
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# GPG for GPG agent
Defaults:sudo env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Card alias specification
# User privilege specification
root ALL=(ALL:ALL) ALL
afpa ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
sudo ALL=(ALL:ALL) ALL
# See sudoers(8) for more information on "include" directives:
#includedir /etc/sudoers.d
```

Figure 7

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

9 Configurer le réseau de la machine tp4-debian

On retourne sur la machine physique et on va changer l'interface réseau de notre Debian. Sur VirtualBox on clique sur "tp4-debian" et sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Réseau interne", puis on clique sur "OK" (Voir la Figure 8).

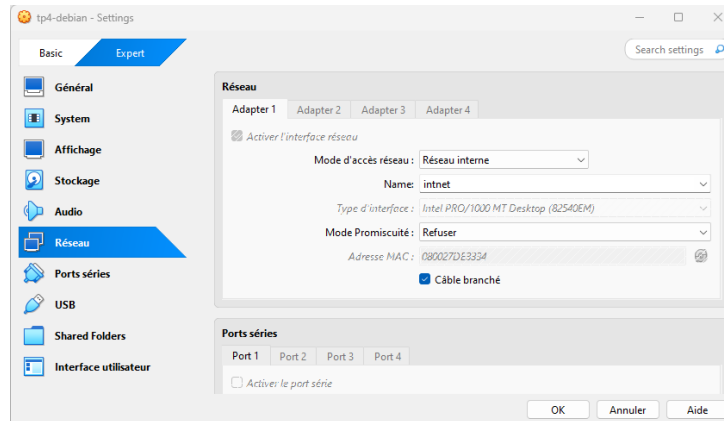


Figure 8

On retourne sur la machine Debian et on se connecte en tant qu'utilisateur afpa. 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 donne une IP fixe à notre serveur et on ajoute une passerelle (Voir la Figure 9)

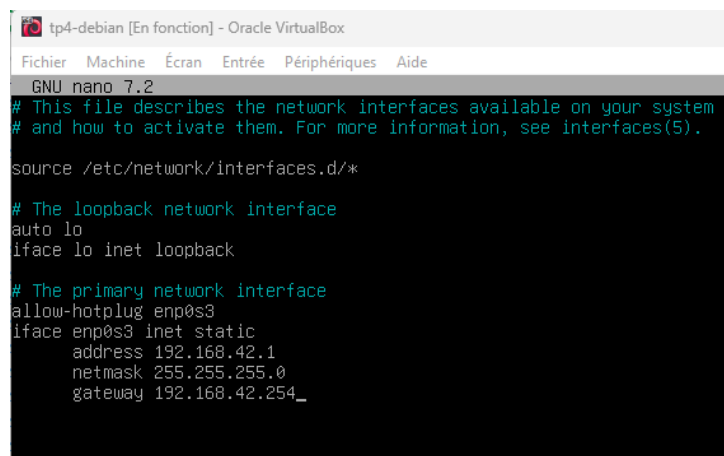


Figure 9

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.

10 Créer la machine virtuelle tp4-windows

Maintenant qu'on a fini de configurer le Debian, on va créer notre machine Client Windows pour qu'elle puisse interagir avec le serveur Nginx. Sur VirtualBox, on clique sur "Nouveau" pour créer une nouvelle machine virtuelle. On met "tp4-windows" comme nom de la machine virtuelle. On choisit son emplacement de sauvegarde et on importe le CD contenant le système d'exploitation Windows. En détectant le CD, le type "Microsoft Windows", le subtype "" et la version "Windows 11 (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 4096 Mo et 2 processeurs, et on clique sur "Suivant". On lui donne un disque dur de 80 Go et on clique sur "Suivant". On regarde bien le récapitulatif et on clique sur "Finish" (Voir la Figure 10).

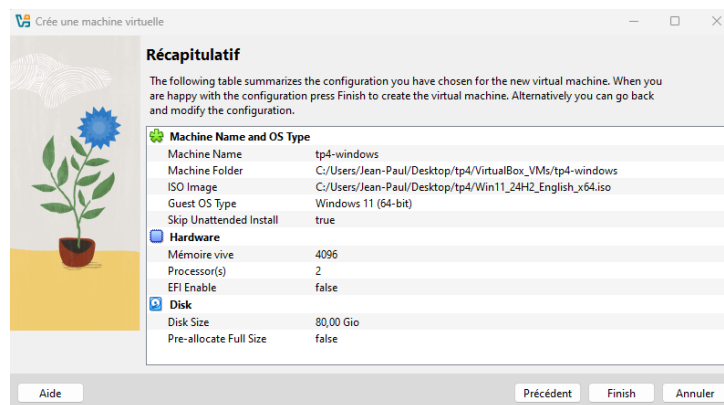


Figure 10

Après avoir créé notre machine virtuelle Windows, il faut la configurer afin qu'elle puisse être dans le même réseau que notre Debian. Sur VirtualBox on clique sur "tp4-windows" et sur "Configurations". On clique sur "Réseau" et on change le mode d'accès réseau à "Réseau interne", puis on clique sur "OK". Puis on clique sur "Démarrer" pour l'allumer.

11 Installer Windows

Après avoir démarré la machine virtuelle, on clique sur n'importe quelle touche pour passer sur le bootable du CD. On choisit pour le paramètre "Language to install: English (United States)" et on choisit pour l'option "Time and currency format: French (France)". On clique sur "Next". Pour le clavier (Keyboard or input method), on choisit "French (Legacy, AZERTY)" et on clique sur "Next". On coche les cases "Install Windows 11" et "I agree everything will be deleted", et on clique sur "Next". On clique sur "I don't have a product key" pour installer Windows sans clé de produit. On choisit d'installer "Windows 11 Pro" et on clique sur "Next" (cf Figure 11).

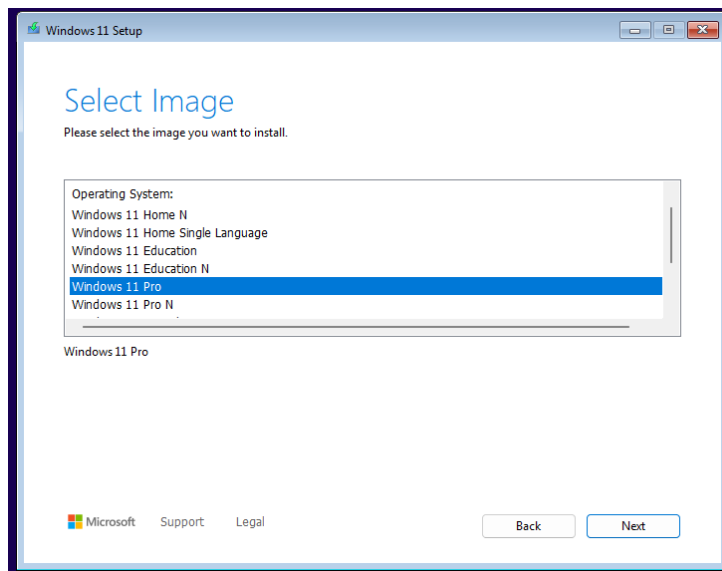


Figure 11

On clique sur "Accept". On choisit le disque "Disk 0 Unallocated Space" pour l'emplacement d'installation et on clique sur "Next". Enfin, on clique sur "Install" pour démarrer l'installation de Windows 11. Après que la machine ait démarré plusieurs fois et qu'on voit la page pour indiquer le pays, on va faire une manipulation afin de pouvoir configurer l'installation finale de Windows sans Internet. Pour ce faire, on appuie sur "Shift + F10". Cela va faire apparaître le prompt. Sur le prompt, on tape la commande : "oobe\bypassNRO", puis on tape sur "Entrée". La machine va de nouveau redémarrer et on retrouve la même page du choix du pays. On choisit "France" et on clique sur "Next". Pour le clavier, on choisit "French (Legacy, AZERTY)" et on clique sur "Yes". On n'a pas de second clavier, donc on clique sur "Skip". Là, la machine va vouloir se connecter à l'internet, mais elle n'est connectée qu'au réseau interne (pas Internet). On clique donc sur le bouton "I don't have Internet" (ce bouton est apparu grâce à la commande oobe tapée plus haut). On donne le nom de l'utilisateur "afpa" et on clique sur "Next". On ne donne pas de mot de passe (on laisse le champ vide) et on clique sur "Next". On clique sur "No" et "Accept", et on répète cette action 1 fois. On choisit l'option "Required only" et on clique sur "Accept". On choisit "No" et on clique sur "Accept", et on répète cette action 2 fois. On laisse Windows finaliser son installation.

12 Configurer le réseau de la machine tp4-windows

Maintenant que le Client Windows est installé, on va configurer sa carte réseau afin qu'elle soit dans le même réseau que le serveur Nginx. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "ncpa.cpl" et on clique sur "OK". Ce qui va ouvrir une nouvelle fenêtre avec toutes les cartes réseau. On fait un clic-droit sur la carte réseau qu'on utilise (ici "Ethernet") et on clique sur "Propriétés". On double-clique sur "Protocole Internet version 4 (TCP/IPv4)". On coche la case "Use the following IP address" et on lui donne une adresse IP fixe ainsi que la passerelle par défaut (Voir la Figure 12). On coche la case "Validate settings upon exit" et on clique sur "OK" (2 fois).

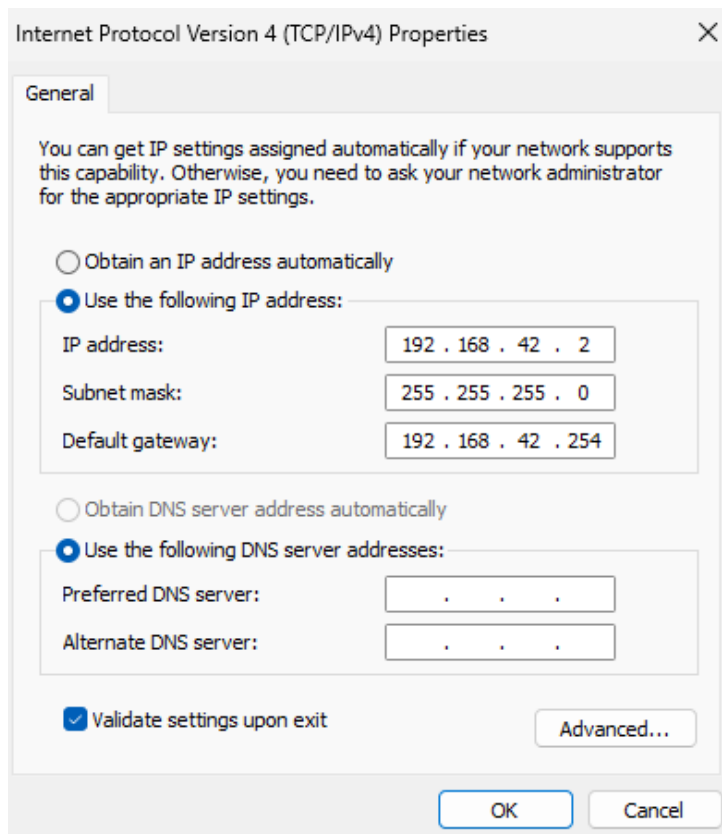


Figure 12

13 Tester le serveur nginx sous Edge

Maintenant que les 2 machines sont sur la même connexion, on va essayer de se connecter sur la page web de notre serveur HTTP. Sur la machine cliente Windows, on ouvre le navigateur "Edge". On tape sur la barre de recherche l'adresse IP du serveur : "https://192.168.1/" et on appuie sur "Entrée". On voit bien que la page web par défaut de Nginx s'affiche bien, donc on a bien accès à notre serveur Nginx depuis notre client Windows (Voir la Figure 13).

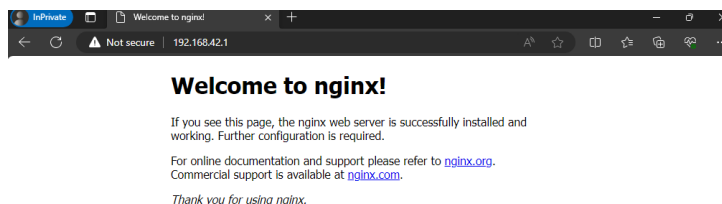
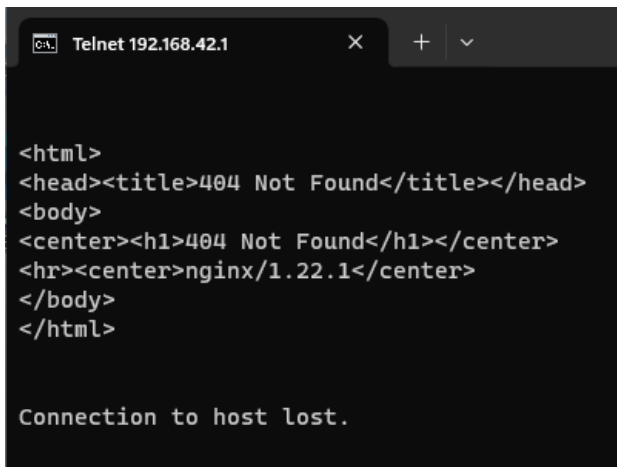


Figure 13

14 Tester le serveur nginx sous Telnet

On teste aussi la connexion avec Telnet. Pour cela, il faut activer Telnet sur le client Windows. On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "control" et on clique sur "OK". On clique sur "Programs", puis sur "Turn Windows features on or off". On coche la case "Telnet client" et on clique sur "OK". On ouvre une invite de commande : On appuie sur "Win + R", ce qui lance le programme "RUN". Dans le programme RUN, on tape "cmd" et on clique sur "OK" pour ouvrir l'invite de commande. Sur l'invite de commandes, on tape la commande : "telnet 192.168.42.1 80". Cela ouvre telnet qui est connecté sur notre serveur Nginx. On tape ensuite la commande : "GET /index.html" et on voit bien sur les Figures 14 et 15 qu'on retrouve la page html de notre serveur Nginx. [La figure 15 est la sortie de la console redirigée dans un fichier pour mieux voir le début de la réponse de la requête GET].

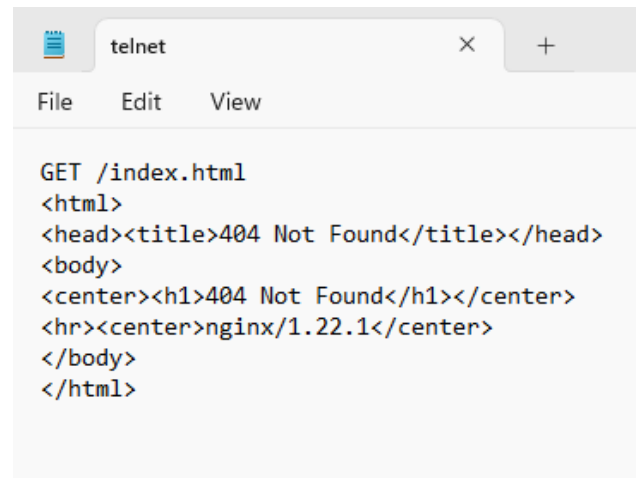


```
Telnet 192.168.42.1

<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.22.1</center>
</body>
</html>

Connection to host lost.
```

Figure 14



```
telnet

File Edit View

GET /index.html
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.22.1</center>
</body>
</html>
```

Figure 15

15 Vérifier les journaux

Avant d'aller plus loin, on va regarder les journaux des événements. On est sur la machine tp4-debian, connecté au terminal tty1 qui est le premier terminal virtuel. On peut changer de terminal virtuel ttyN en appuyant sur "Ctrl+Alt+Fn" où Fn correspond à une des touches F1 à F12. On bascule sur le terminal tty2 en appuyant sur "Ctrl+Alt+F2". Sur ce nouveau terminal, on se connecte avec les identifiants de l'utilisateur "afpa". Après s'être connecté, on va regarder le fichier log du serveur Nginx en tapant la commande : "sudo tail -f /var/log/nginx/access.log". La commande tail donne les 10 dernières lignes d'un fichier et l'option "-f" permet de suivre l'évolution du fichier en temps réel. Ainsi on peut voir, comme sur la Figure 16, qu'on a une entrée à chaque fois on a essayé de se connecter sur notre serveur Nginx via telnet depuis la machine tp4-windows.

```
Debian GNU/Linux 12 tp4-debian tty2
tp4-debian login: afpa
Password:
Linux tp4-debian 6.1.0-28-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.119-1 (2024-11-22) 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: Thu Jan  9 13:31:02 CET 2025 on tty1
afpa@tp4-debian:~$ ps
  PID TTY          TIME CMD
  1075 tty2      00:00:00 bash
  1078 tty2      00:00:00 ps
afpa@tp4-debian:~$ sudo tail -f /var/log/nginx/access.log
[sudo] password for afpa:
192.168.42.2 - - [09/Jan/2025:15:23:40 +0100] "GET /index.html" 404 153 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:24:39 +0100] "GET /index.html" 404 153 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:26:27 +0100] "GET /index.html" 404 153 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:27:34 +0100] "s" 400 157 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:28:57 +0100] "GET /index.html" 404 153 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:30:34 +0100] "a" 400 157 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:31:26 +0100] "GET " 400 157 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:32:06 +0100] "GET" 400 157 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:33:00 +0100] "GET /index.html" 404 153 "-" "-"
192.168.42.2 - - [09/Jan/2025:15:38:43 +0100] "GET /index.html" 404 153 "-" "-"
```

Figure 16

On retourne sur le premier terminal virtuel tty1 en appuyant sur "Ctrl+Alt+F1".

16 Activer des sites

Dans cette partie, on va créer des pages web pour notre serveur Nginx. Pour chaque page, on va donner un port différent. Pour cela on va faire un fichier de configuration pour chaque page. On a 3 pages qui correspondent à 3 personnes : Arthur, Jules et Louis. On va créer les fichiers arthur, jules et louis à partir du fichier default qui se trouve dans le chemin "/etc/nginx/sites-available/". Sur la machine tp4-debian, on tape donc les commandes suivantes :

Pour Arthur : "sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/arthur"

Pour Jules : "sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/jules"

Pour Louis : "sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/louis"

Puis, pour chaque fichier de configurations, on va modifier le port d'écoute ainsi que la racine pour les dossiers. On tape donc :

- "sudo nano /etc/nginx/sites-available/arthur" pour modifier le fichier d'Arthur. Dans ce fichier on change le port 80 en 8080 et la racine "/var/www/html" en "/var/www/arthur". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder.
- "sudo nano /etc/nginx/sites-available/jules" pour modifier le fichier de Jules. Dans ce fichier on change le port 80 en 8081 et la racine "/var/www/html" en "/var/www/jules". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder.
- "sudo nano /etc/nginx/sites-available/louis" pour modifier le fichier de Louis. Dans ce fichier on change le port 80 en 8082 et la racine "/var/www/html" en "/var/www/louis". On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder.

On peut voir les étapes précédentes sur les Figures 17, 18 et 19.

```

GNU nano 7.2 /etc/nginx/sites-available/arthur
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8080 default_server;
    listen [::]:8080 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/arthur;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
}

```

Figure 17

```

GNU nano 7.2 /etc/nginx/sites-available/jules
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/NginxDirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8081 default_server;
    listen [::]:8081 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/jules;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
}

```

Figure 18

```

GNU nano 7.2 /etc/nginx/sites-available/louis
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/NginxDirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8082 default_server;
    listen [::]:8082 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/louis;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
}

```

Figure 19

Ensuite, on va activer les sites qu'on a créés pour chaque page. Pour cela on va utiliser la commande `ln -s`. Cette dernière permet de créer un lien symbolique dans un système Unix/Linux. Un lien symbolique est un fichier distinct qui pointe vers le chemin du fichier cible. Si le fichier cible est supprimé, le lien symbolique devient invalide. Pour activer les sites, on va créer les liens symbolique dans le dossier "sites-enabled" du dossier nginx. On tape donc :

- "`sudo ln -s /etc/nginx/sites-available/arthur /etc/nginx/sites-enabled`" pour activer le site d'Arthur
- "`sudo ln -s /etc/nginx/sites-available/jules /etc/nginx/sites-enabled`" pour activer le site de Jules
- "`sudo ln -s /etc/nginx/sites-available/louis /etc/nginx/sites-enabled`" pour activer le site de Louis

Enfin, on va créer les dossiers racines dans lesquels on placera un fichier index.html pour chaque page. On va donc faire les commandes suivantes :

- "sudo mkdir /var/www/(USR)"
- "sudo nano /var/www/(USR)/index.html"
- Dans le fichier index.html on va écrire :

```
1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <title>(USR)'s page</title>
5     </head>
6     <body>
7         <h1>(USR)</h1>
8         <p>Hello I am (USR)!</p>
9     </body>
10 </html>
```

Où (USR) est remplacé par Arthur, Jules ou Louis en fonction du port utilisé. Pour finir, on redémarre le serveur Nginx en tapant la commande : "sudo systemctl restart nginx". On vérifie si le serveur fonctionne correctement en tapant la commande : "sudo systemctl status nginx" et on voit bien qu'il n'y a pas d'erreur. Les 3 sites étant activés, on peut se connecter avec le client Windows.

17 Tester les sites activés

Sur la machine tp4-windows, on va tester les 3 sites en utilisant le navigateur Edge. On lance Edge et on tape sur la barre de recherche le lien `http://192.168.42.1:8080`. On voit bien sur la Figure 20 qu'on tombe sur la page web d'Arthur. Ensuite, en saisissant le lien `http://192.168.42.1:8081`, on accède à la page web de Jules, et en entrant le lien `http://192.168.42.1:8082`, on arrive bien sur la page web de Louis (Voir les Figures 21 et 22).

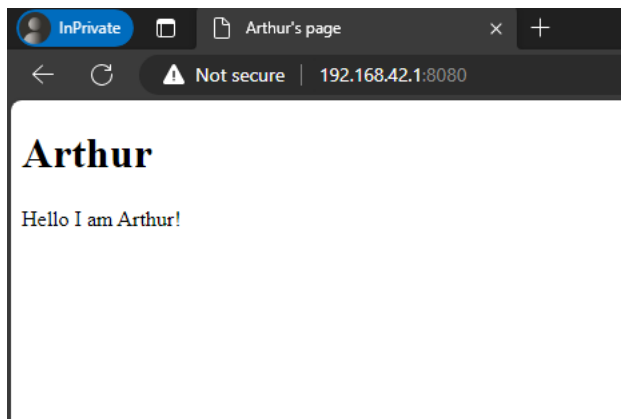


Figure 20

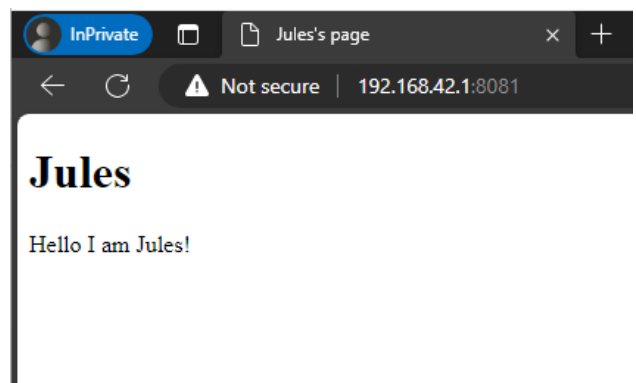


Figure 21

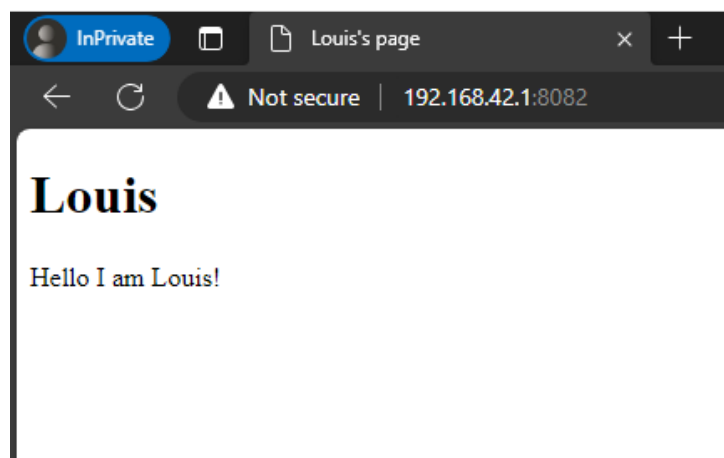


Figure 22

18 Désactiver les sites

Maintenant, on va désactiver les 3 sites. Pour cela, on va supprimer les fichiers liens symbolique présentes dans le dossier "sites-enabled". Sur la machine tp4-debian, on tape :

- "sudo rm /etc/nginx/sites-enabled/arthur" pour désactiver le site d'Arthur
- "sudo rm /etc/nginx/sites-enabled/jules" pour désactiver le site de Jules
- "sudo rm /etc/nginx/sites-enabled/louis" pour désactiver le site de Louis

On peut voir les étapes précédentes sur les Figures 23, 24 et 25.

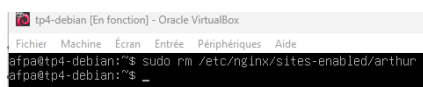


Figure 23

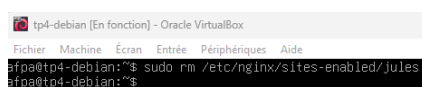


Figure 24

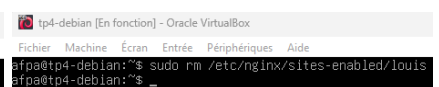


Figure 25

On redémarre le serveur Nginx en tapant la commande : "sudo systemctl restart nginx". On vérifie si le serveur fonctionne correctement en tapant la commande : "sudo systemctl status nginx".

19 Tester les sites désactivés

On retourne sur la machine tp4-windows et on reteste les 3 sites sur Edge. Comme le montrent les Figures 26, 27 et 28, lorsqu'on tente d'accéder aux liens `http://192.168.42.1:8080`, `http://192.168.42.1:8081` ou `http://192.168.42.1:8082`, on n'arrive pas à accéder aux pages. Les 3 sites sont bien désactivés.

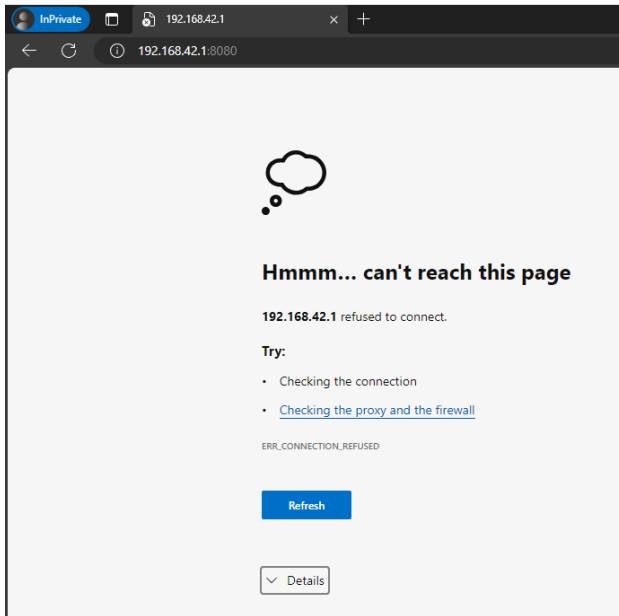


Figure 26

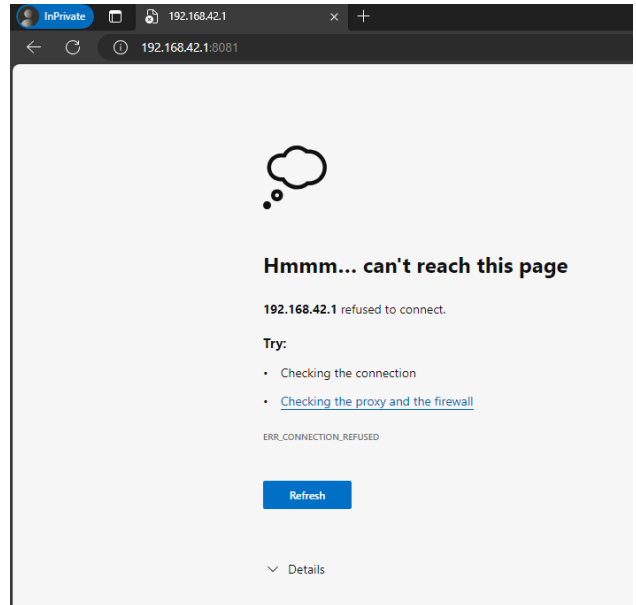


Figure 27

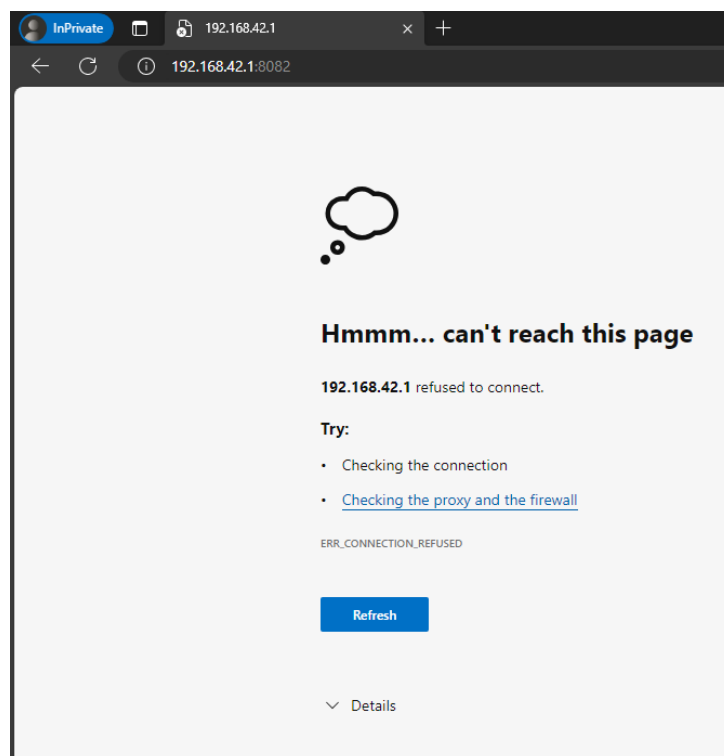
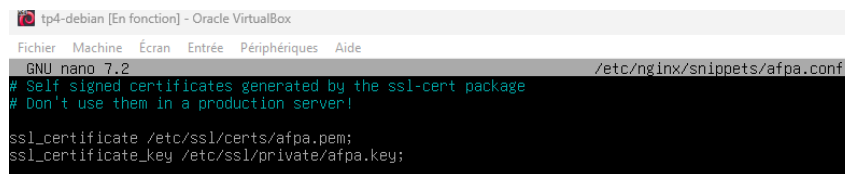


Figure 28

20 Activer SSL

Dans la dernière partie de ce TP, on va passer les sites de notre serveur Nginx en HTTPS. Pour cela, on va utiliser les certificats SSL qui vont permettre de sécuriser la connexion vers le serveur Debian. Sur la machine tp4-debian, on va générer un certificat SSL et sa clé. Pour cela, on tape la commande : "sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/afpa.key -out /etc/ssl/certs/afpa.pem". On va appuyer sur "Entrée" jusqu'à la fin du processus. On va ensuite copier le fichier snakeoil.conf qui se trouve dans le chemin "/etc/nginx/snippets/" afin de pouvoir configurer le SSL sans écraser le fichier original. On tape : "sudo cp /etc/nginx/snippets/snakeoil.conf /etc/nginx/snippets/afpa.conf". On modifie le fichier afpa.conf en tapant la commande : "sudo nano /etc/nginx/snippets/afpa.conf". Dans ce fichier, on indique le chemin vers le certificat SSL ainsi que sa clé (Voir la Figure 29). On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder.

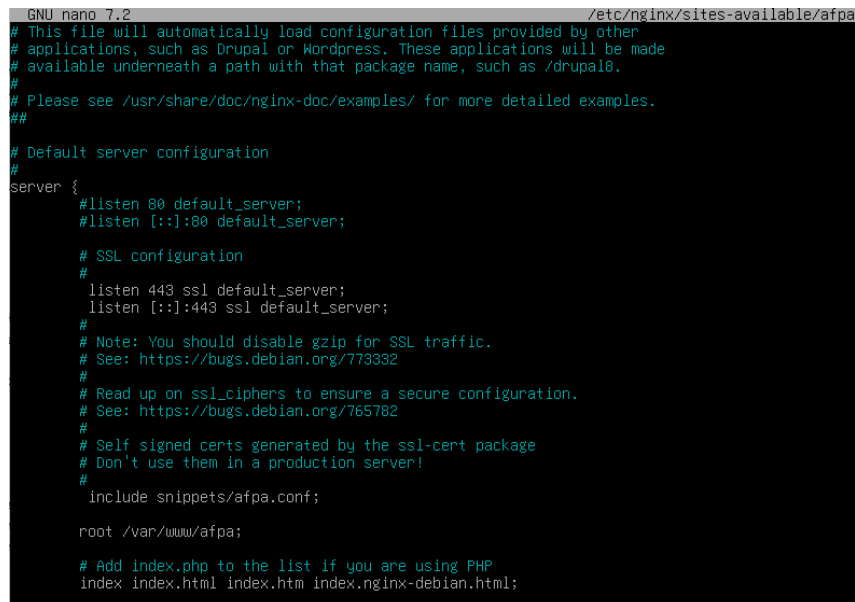


```
tp4-debian [En fonction] - Oracle VirtualBox
Fichier Machine Écran Entrée Périphériques Aide
GNU nano 7.2 /etc/nginx/snippets/afpa.conf
# Self signed certificates generated by the ssl-cert package
# Don't use them in a production server!

ssl_certificate /etc/ssl/certs/afpa.pem;
ssl_certificate_key /etc/ssl/private/afpa.key;
```

Figure 29

Ensuite on crée un nouveau fichier de configuration à partir du fichier default. On tape : "sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/afpa". On modifie ce fichier avec la commande nano : "sudo nano /etc/nginx/sites-available/afpa". Dans ce fichier, on change le port d'écoute 80 à 443, le dossier racine et on indique le chemin vers le fichier de configuration (Voir la Figure 30).



```
GNU nano 7.2 /etc/nginx/sites-available/afpa
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    #listen 80 default_server;
    #listen [::]:80 default_server;

    # SSL configuration
    #
    listen 443 ssl default_server;
    listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    include snippets/afpa.conf;

    root /var/www/afpa;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;
```

Figure 30

On va créer le dossier afpa dans le chemin /var/www avec la commande : "sudo mkdir /var/www/afpa". On va ensuite créer le fichier index.html à l'intérieur de ce dossier. On tape la commande : "sudo nano /var/www/afpa/index.html". Dans ce fichier, on écrit les commandes suivantes :


```

1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <title>Afpa's page</title>
5     </head>
6     <body>
7         <h1>Afpa</h1>
8         <p>Hello I am Afpa!</p>
9         <p>This site supports SSL!</p>
10    </body>
11 </html>

```

On sauvegarde en utilisant "Ctrl+X" et la touche "y" pour "Yes", et la touche "Entrée" pour sauvegarder. On active le site d'Afpa en créant un lien symbolique dans le dossier "sites-enabled". On tape : "sudo ln -s /etc/nginx/sites-available/afpa /etc/nginx/sites-enabled". Enfin on redémarre le serveur Nginx en tapant la commande : "sudo systemctl restart nginx". On vérifie si le serveur fonctionne correctement en tapant la commande : "sudo systemctl status nginx".

21 Tester SSL

On retourne sur le client Windows 11 et on ouvre le navigateur Edge. On tape le lien `https://192.168.42.1/` sur la barre de recherche et on clique sur "Entrée". On voit qu'on arrive sur une page d'alerte de sécurité SSL (Voir la Figure 31). Cette alerte est due au certificat SSL qui est autosigné. On clique sur "Advanced" et sur "Continue to 192.168.42.1 (unsafe)" pour accéder au site et on tombe sur le site `afpa.conf` (Voir la Figure 32).

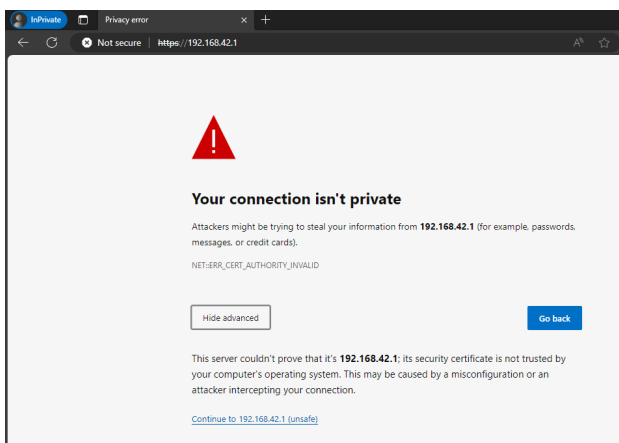


Figure 31

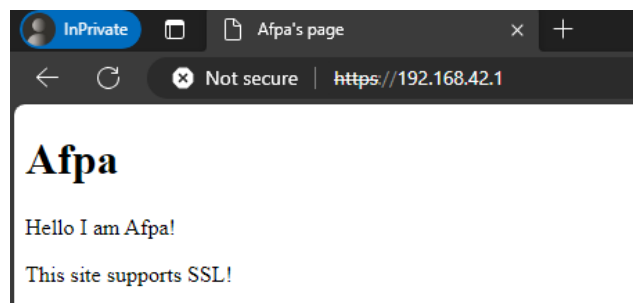


Figure 32