



debian

Machine Virtuelle

DOCUMENTATION TECHNIQUE

Table des matières

- 1. Définition**
- 2. Prérequis**
- 3. Installation**
- 4. Fonctionnement**
- 5. Fonctionnalités**

1. Définitions

VirtualBox est le logiciel de virtualisation gratuit, open source et multiplateforme d'Oracle. C'est un hyperviseur de type 2, celui-ci permet d'héberger une ou plusieurs machines virtuelles, avec des systèmes d'exploitation différents. (mettre le terme hyperviseur)

Debian a pour principal but de fournir un système d'exploitation composé uniquement de logiciels libres.

Une machine virtuelle ou VM est un environnement entièrement virtualisé qui fonctionne sur une machine physique. Elle exécute son propre système d'exploitation (OS) et bénéficie des mêmes équipements qu'une machine physique : CPU, mémoire RAM, disque dur et carte réseau.

2. Prérequis

Pour fonctionner, Debian requiert à minima :

- RAM : 1GB Minimum
- Disque dur : 10GB
- CPU : Pentium 4, 1GHz

3. Installation

Pour installer Debian, nous devrons d'abord installer la machine virtuelle VirtualBox.

Nous allons nous rendre sur le lien

[“https://www.oracle.com/fr/virtualization/technologies/vm/downloads/virtualbox-downloads.html”](https://www.oracle.com/fr/virtualization/technologies/vm/downloads/virtualbox-downloads.html)

Et sélectionner l'installateur Windows 64bits

Oracle VM VirtualBox Base Packages - 7.0.10

Freely available for Windows, Mac OS X, Linux and Solaris x86 platforms under GPLv3:

Platform	64-bit
Windows	 Windows Installer

Nous allons maintenant télécharger le fichier .iso de Debian. Rendez-vous sur “<https://www.debian.org/distrib/index.fr.html>”, puis cliquez sur “ISO netinst pour pc 64 bits”

Télécharger une image d'installation

- Une image d'installation de taille réduite peut être téléchargée rapidement et enregistrée sur disque amovible. Pour cela, une machine avec une connexion à Internet est nécessaire.

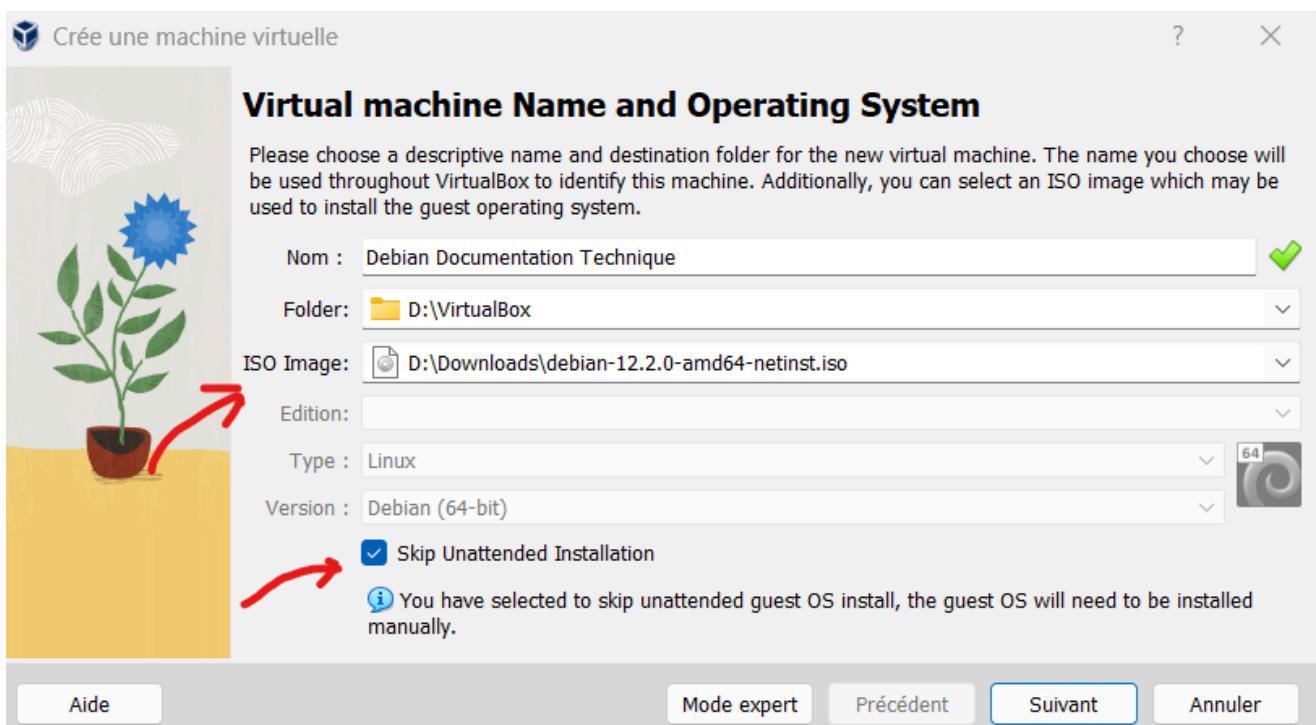


[ISO « netinst » pour PC 64 bits](#), [ISO « netinst » pour PC 32 bits](#), [Torrents « netinst » pour PC 64 bits](#),
[Torrents « netinst » pour PC 32 bits](#)

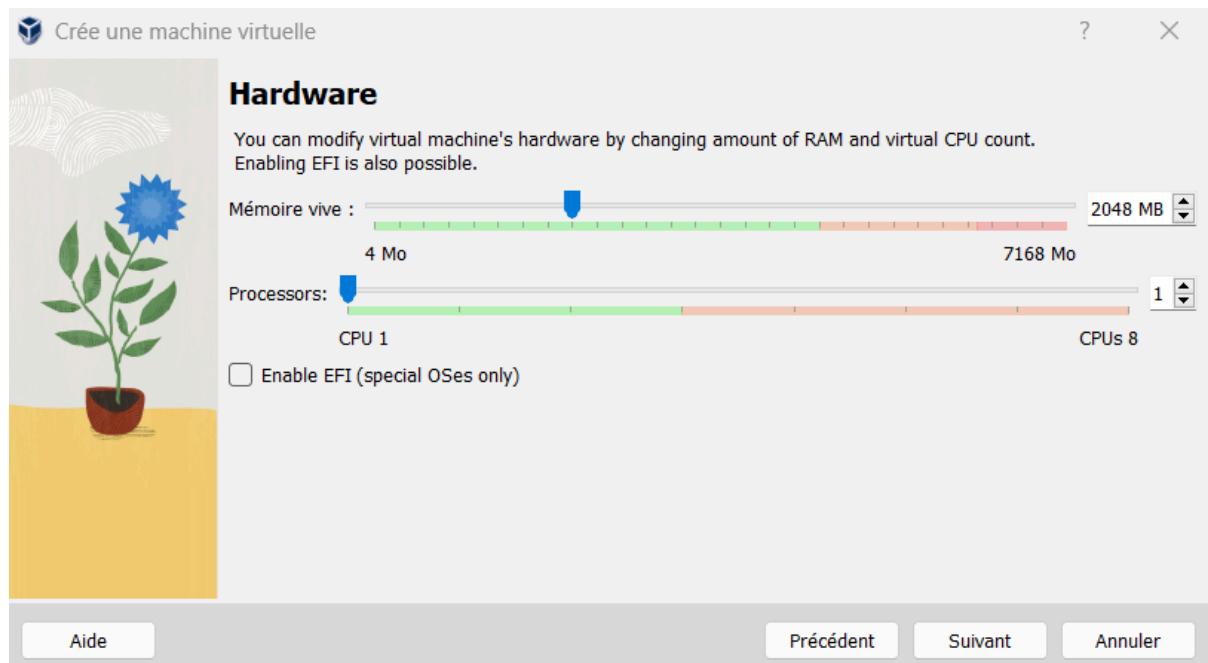
Lancer le logiciel VirtualBox, créer une nouvelle machine virtuelle en cliquant sur “Nouvelle” depuis la section “Outils”



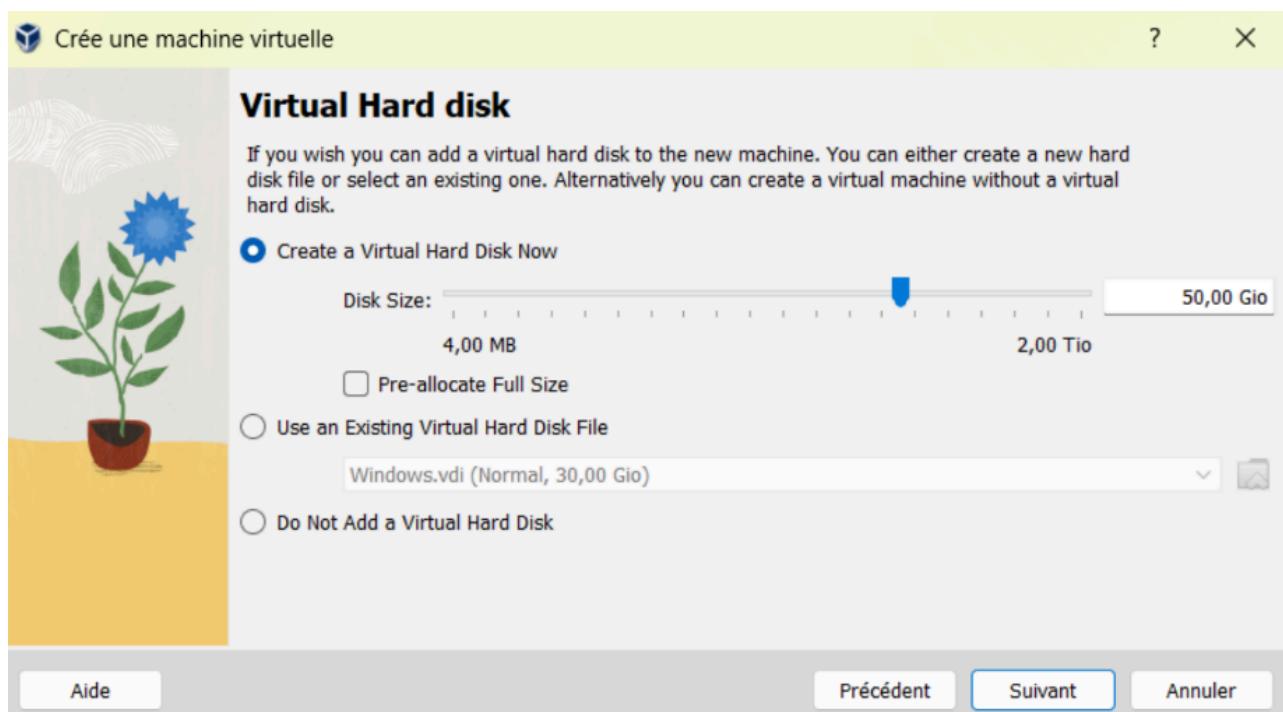
Nommer votre machine virtuelle puis sélectionner l'image iso de Debian dans la section “ISO Image”
Cocher la case “Skip Unattended Installation”



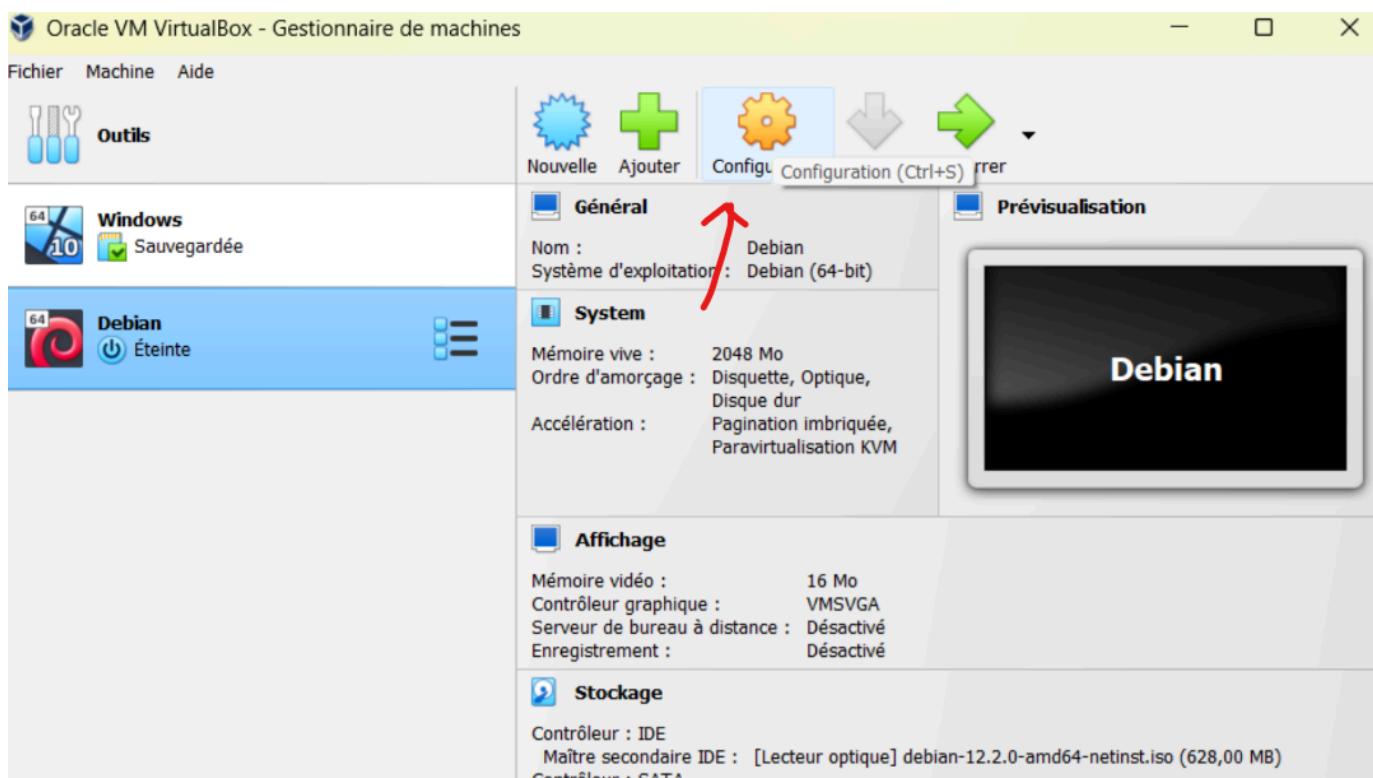
Sélectionner la mémoire vive et le nombre de coeurs accordés à la machine, on garde ceux par défaut



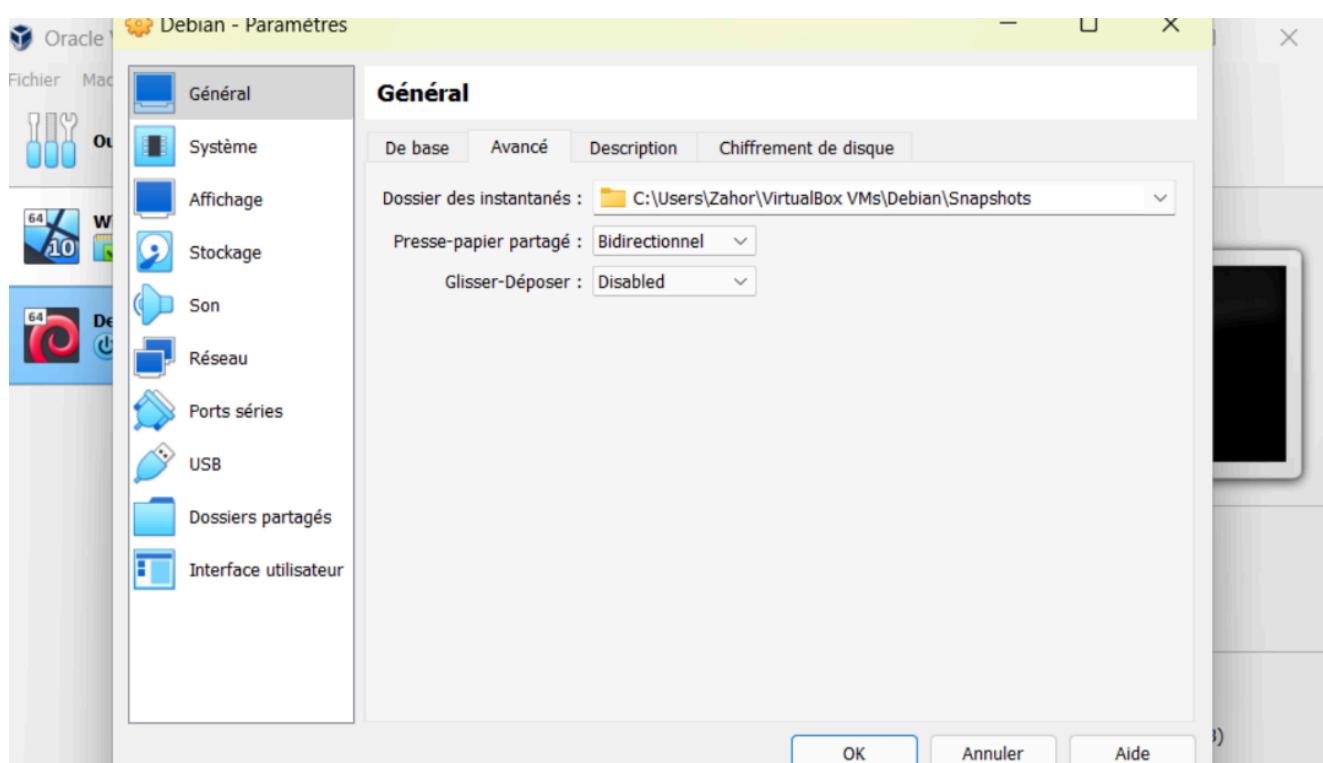
On alloue ensuite l'espace disque, on choisira 50 Go pour cette machine



On va ensuite configurer notre VM en cliquant sur l'onglet "Configuration"



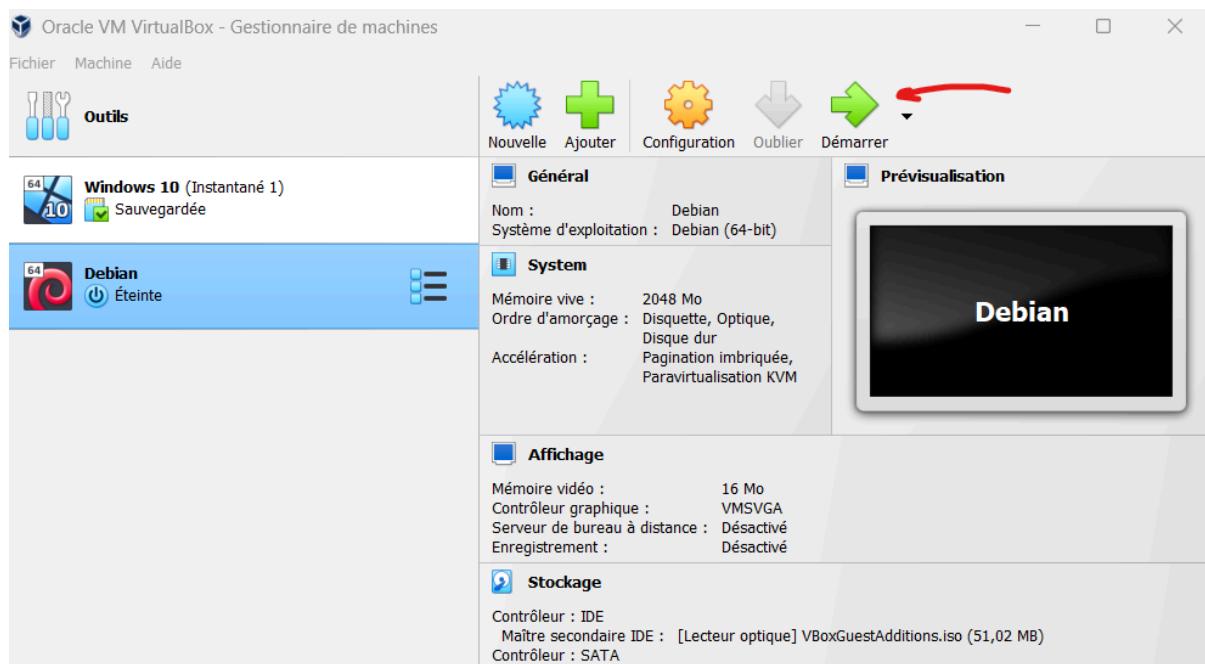
On se rend dans Général > Avancé puis on choisit "Bidirectionnel" dans la section "Presse-papier partagé"



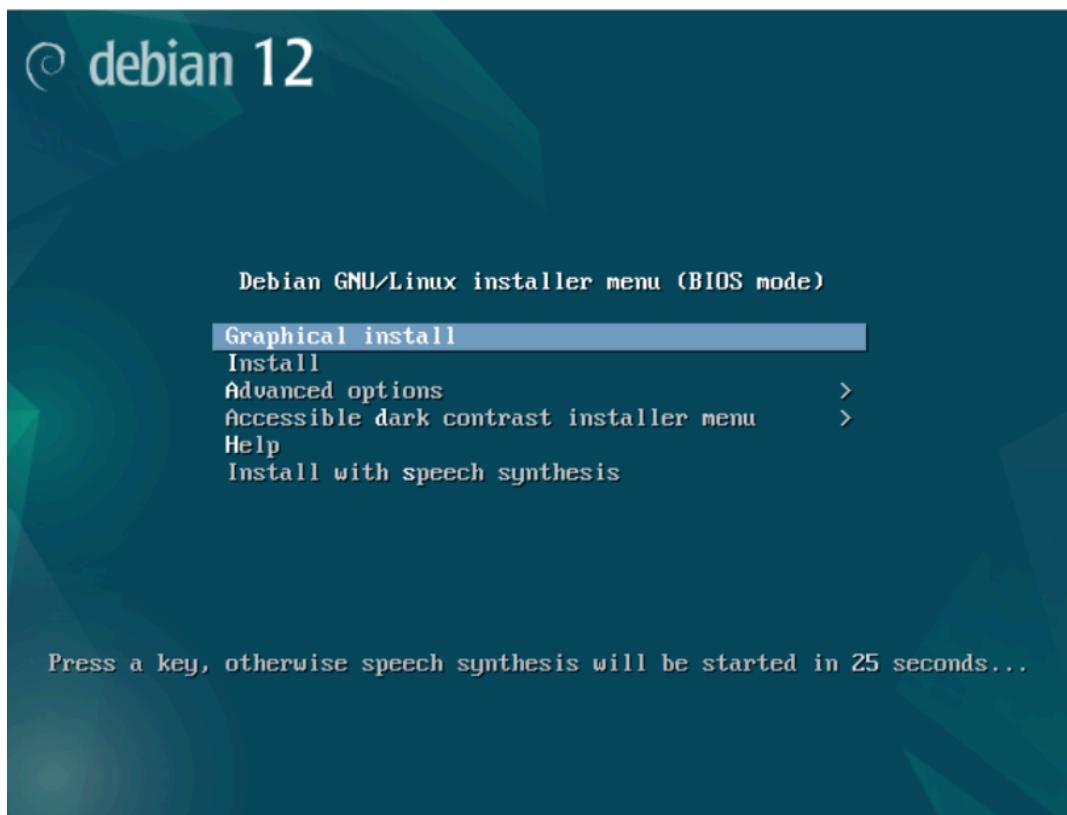
3. Fonctionnement

Nous allons maintenant démarrer notre machine virtuelle.

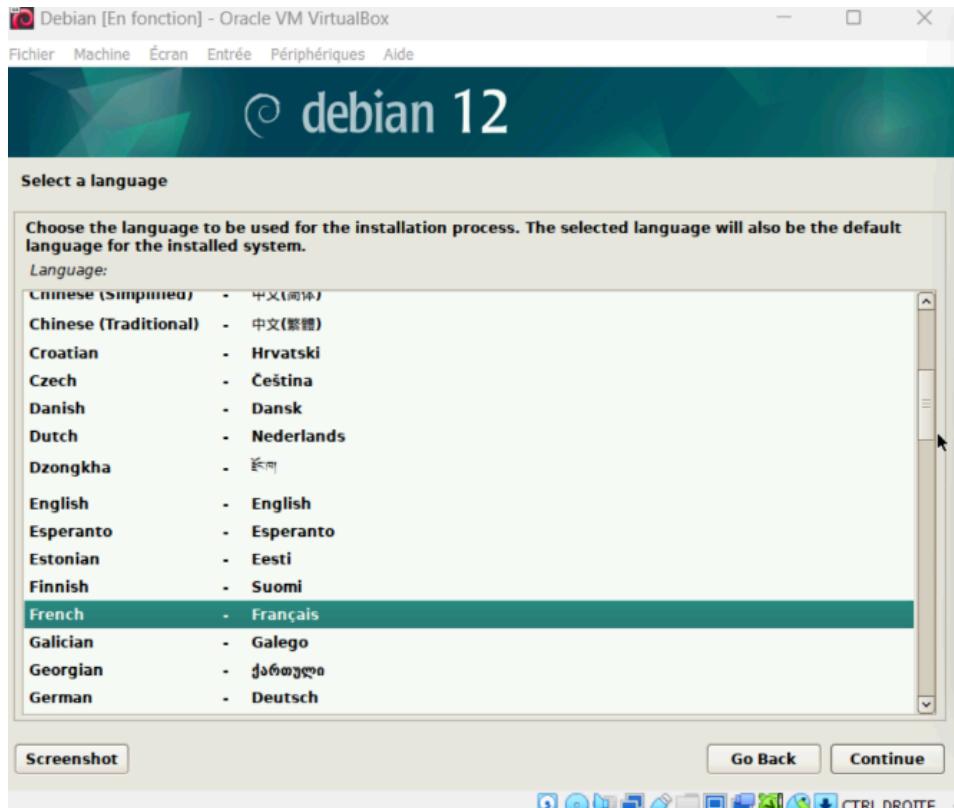
Selectionner la machine en cliquant dessus puis cliquer sur l'icône “Démarrer”



Une fois la machine lancée, sélectionner “Graphics install”



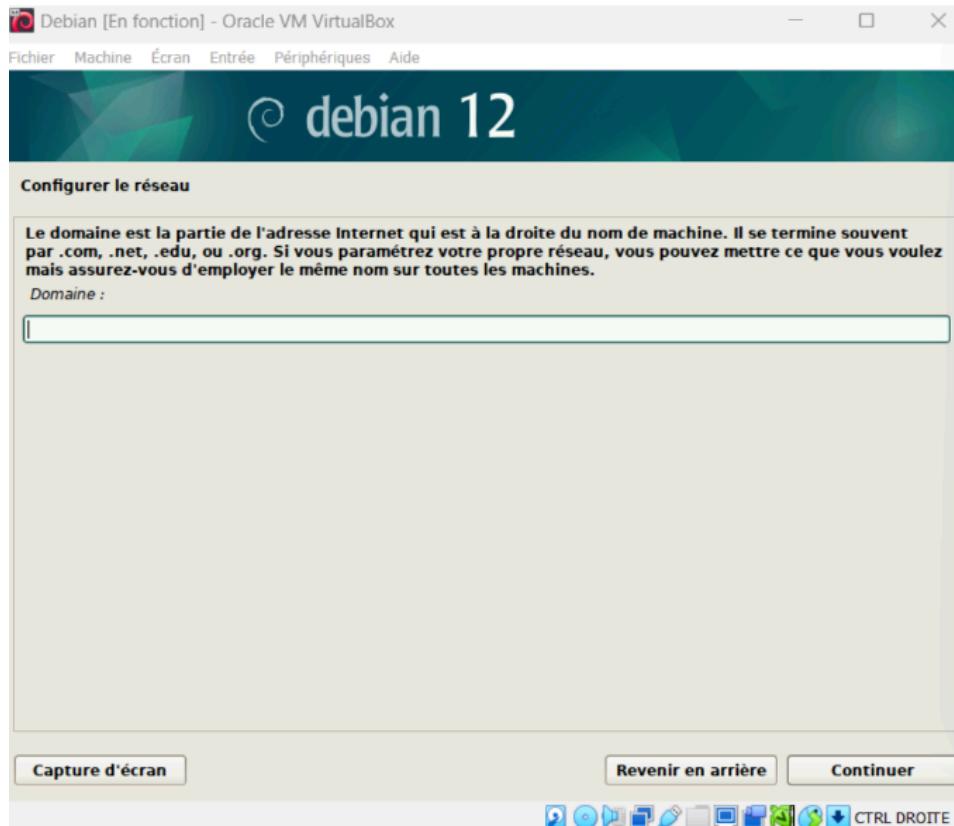
Dans la sélection de langage nous sélectionnons “Français”, de même pour la situation géographique et le clavier.



On choisit un nom d'utilisateur selon nos goûts



Dans la partie “Configurer le réseau” laisser la ligne vide et cliquer sur “continuer”



Dans la création de mot de passe, nous choisissons un mdp personnalisé.

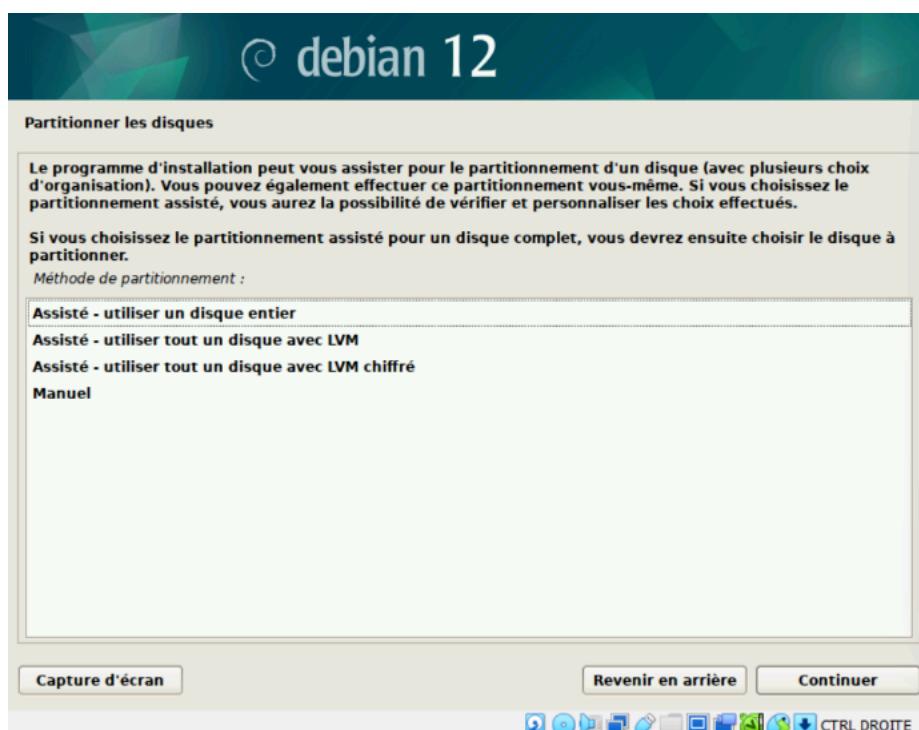


De même pour notre nom d'utilisateur

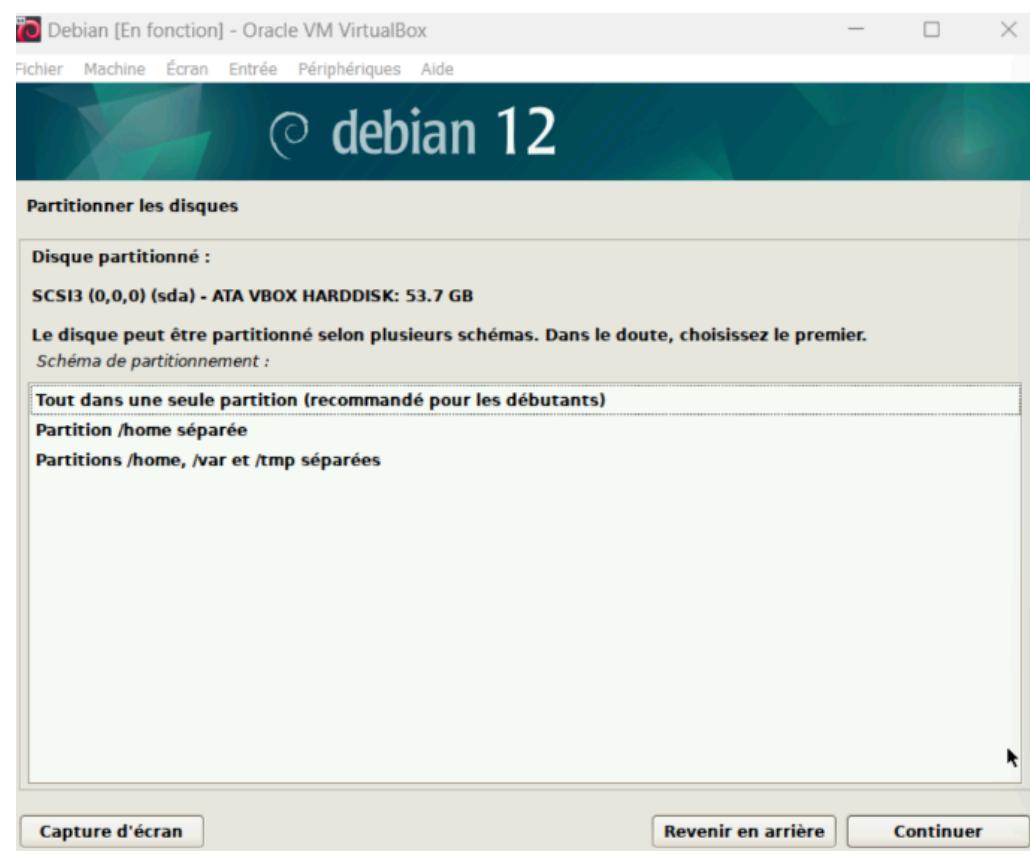


Puis nous réalisons la même chose pour les 2 prochaines étapes qui concernent aussi notre nom d'utilisateur et le mot de passe

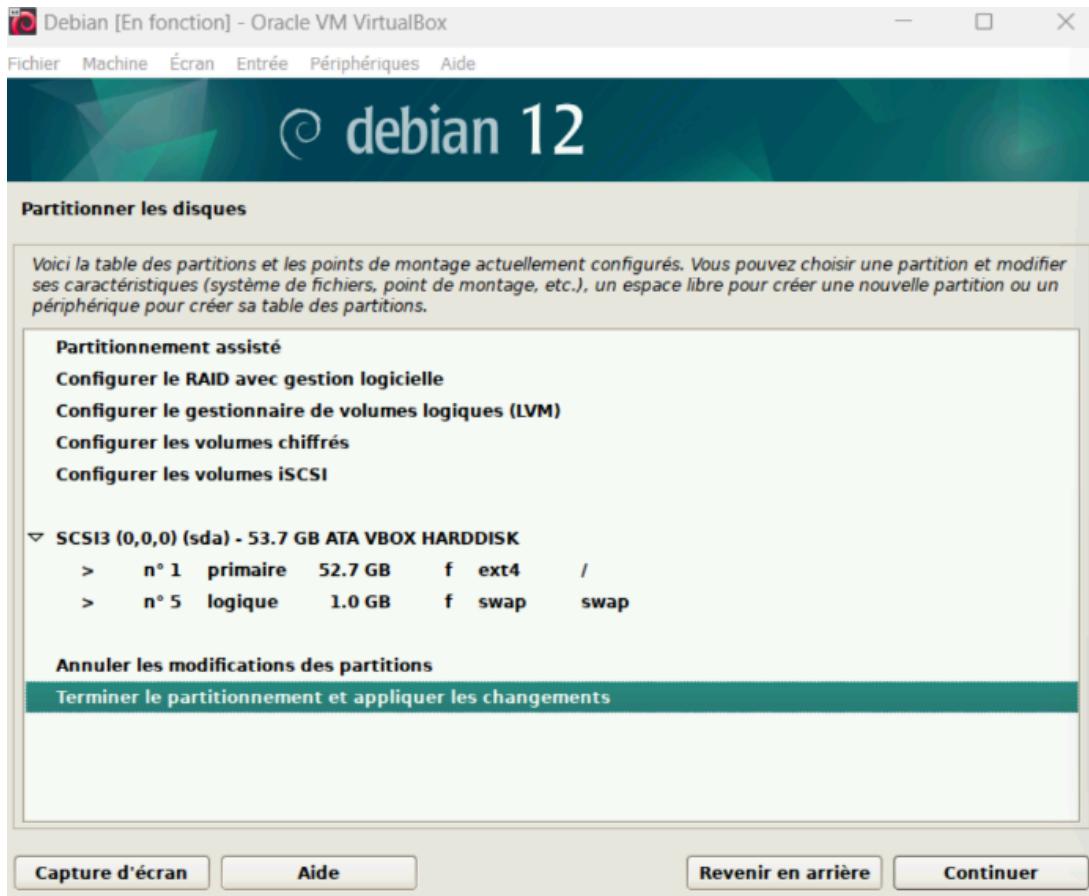
Dans la catégorie "partitionner les disques", sélectionner "utiliser un disque entier"



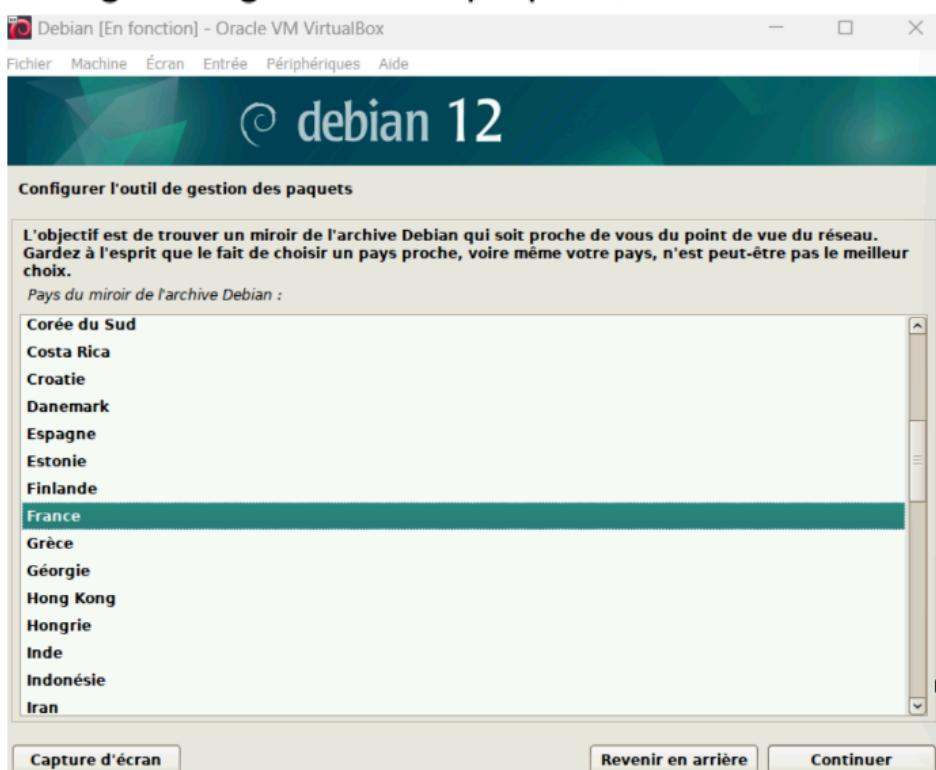
Nous sélectionnons notre disque puis à l'étape d'après nous choisissons "Tout dans une seule partition"



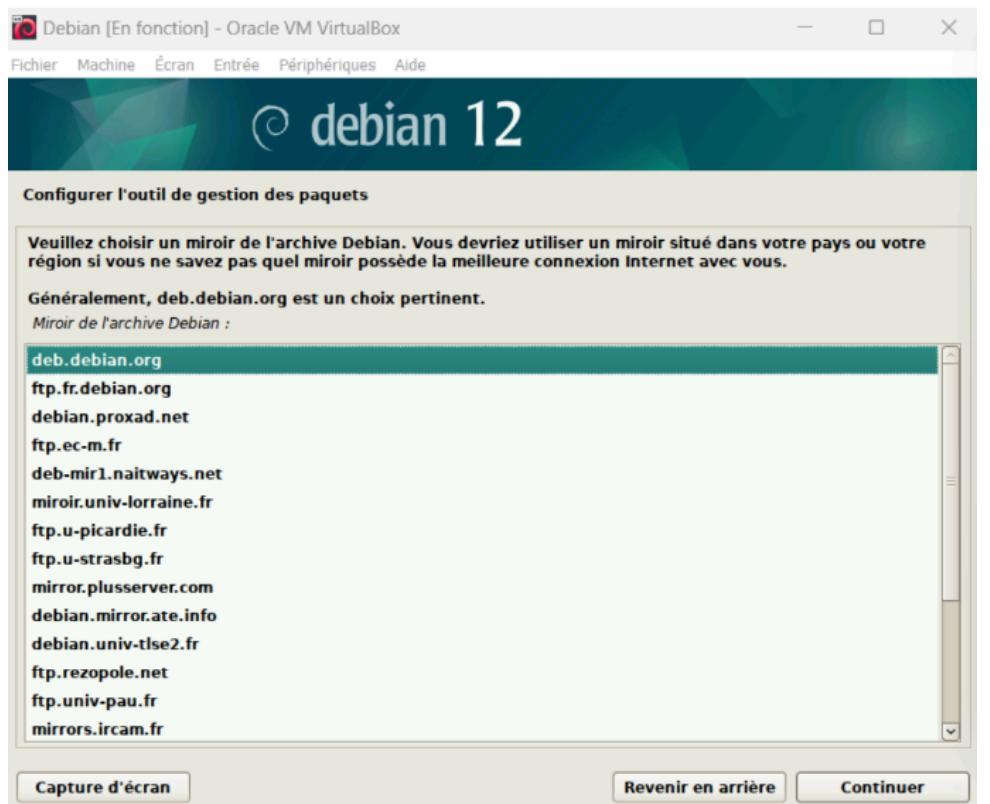
Puis sélectionner “Terminer le partitionnement et appliquer les changements, puis sur les pages suivantes, sélectionner “Oui”, puis “Non”



Dans l'onglet de gestion des paquets, sélectionner “France”

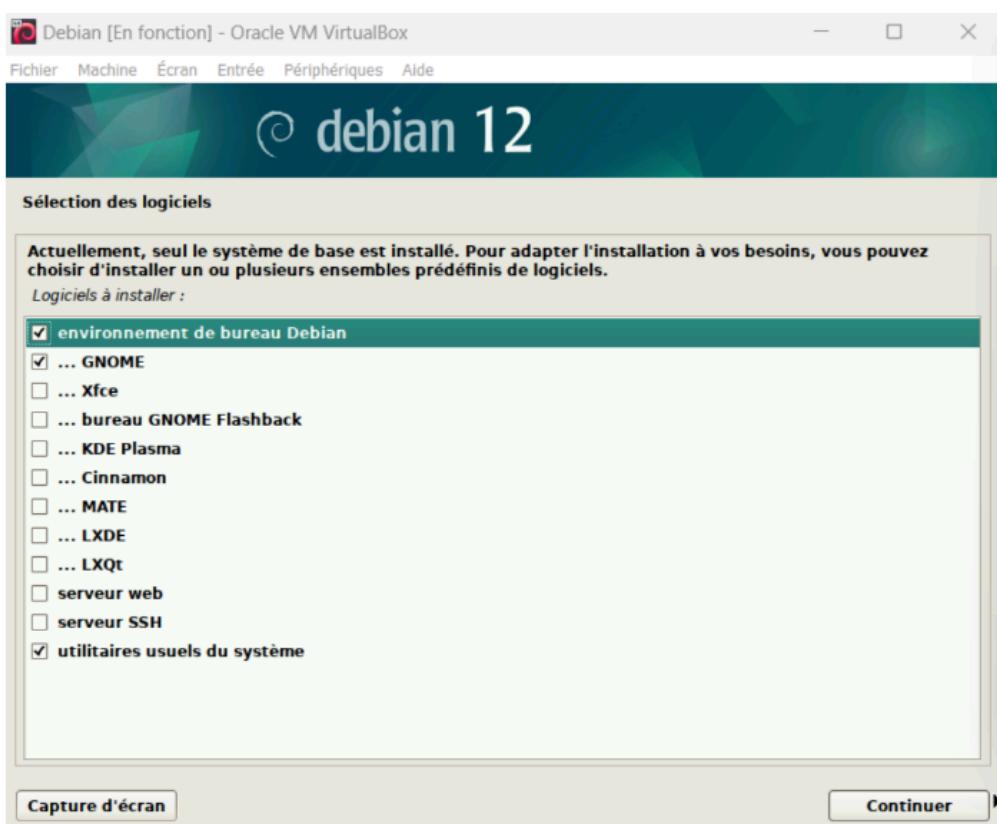


Sélectionner deb.debian.org comme miroir



Cliquer directement sur continuer dans l'outil de gestion des paquets

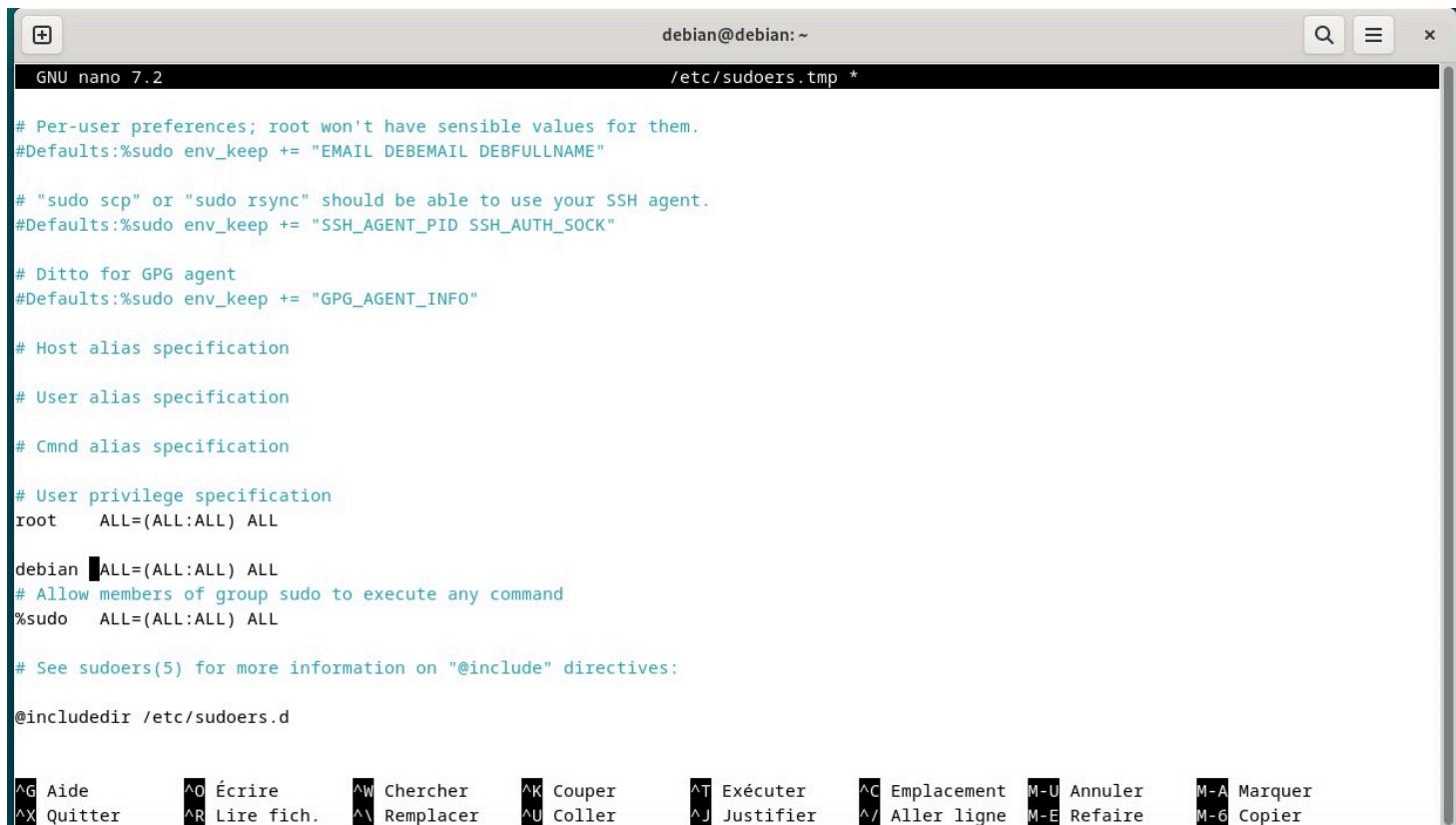
Puis dans la sélection des logiciels, laisser les sélections par défaut



PARTIE 2 : Création d'un serveur web

Depuis le terminal on accède à cette page pour donner tous les droits à notre utilisateur en écrivant “/sbin/visudo”

Puis, dans la partie “User privilege specification” on écrit “debian ALL=(ALL:ALL)”, “debian” est notre utilisateur par défaut. Puis quitter en appuyant sur CTRL+X



```
GNU nano 7.2          debian@debian: ~          /etc/sudoers.tmp *
```

```
# 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

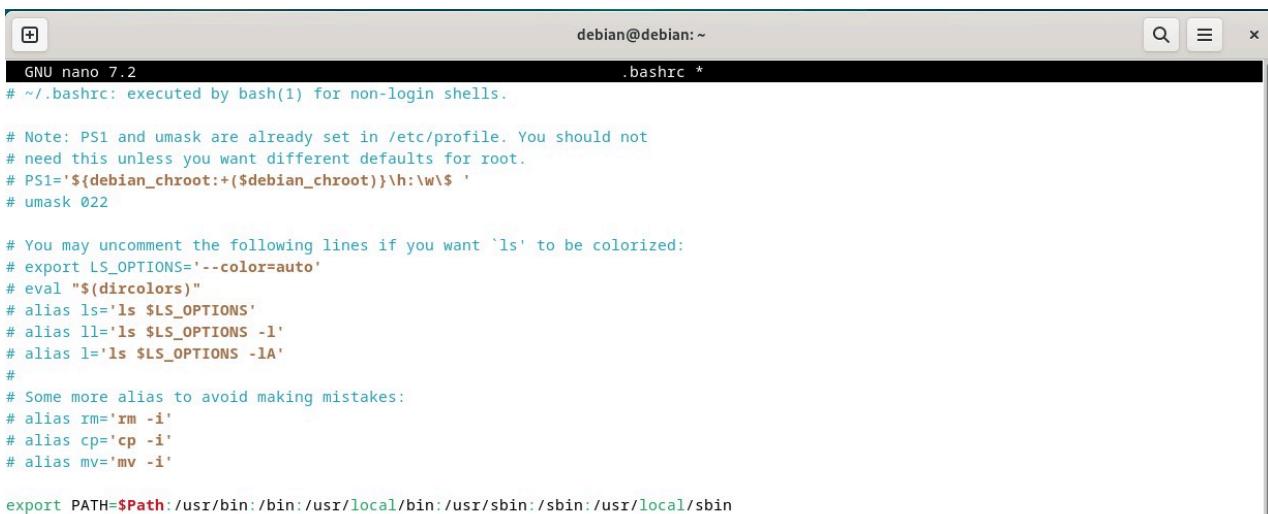
debian  ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

^G Aide      ^O Écrire      ^W Chercher      ^K Couper      ^T Exécuter      ^C Emplacement      M-U Annuler      M-A Marquer
^X Quitter   ^R Lire fich.  ^\ Remplacer   ^U Coller       ^J Justifier   ^V Aller ligne   M-E Refaire   M-G Copier
```

Puis dans le root, on écrit “nano .bashrc” et on écrit la commande “export PATH=\$PATH:/usr/bin:/bin:/usr/local/bin:/usr/sbin:/sbin:/usr/local/sbin”



```
GNU nano 7.2          debian@debian: ~          .bashrc *
```

```
# ~/.bashrc: executed by bash(1) for non-login shells.

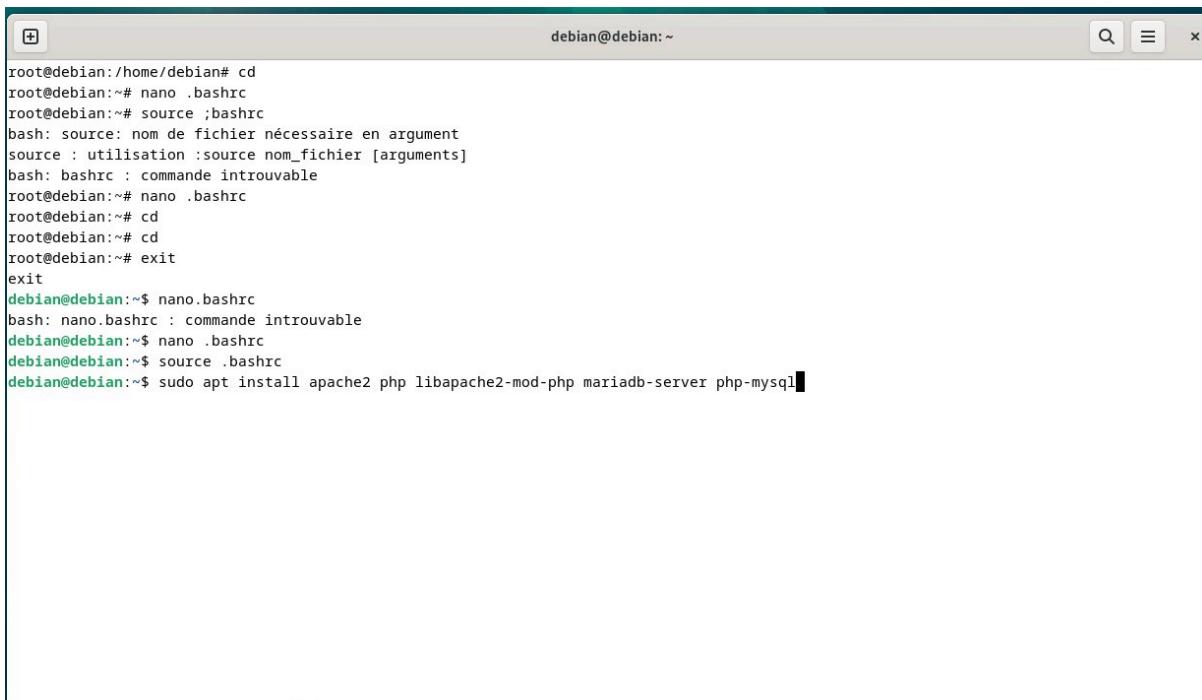
# Note: PS1 and umask are already set in /etc/profile. You should not
# need this unless you want different defaults for root.
# PS1='${debian_chroot:+($debian_chroot)}\h:\w\$ '
# umask 022

# You may uncomment the following lines if you want 'ls' to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -la'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

export PATH=$PATH:/usr/bin:/bin:/usr/local/bin:/usr/sbin:/sbin:/usr/local/sbin
```

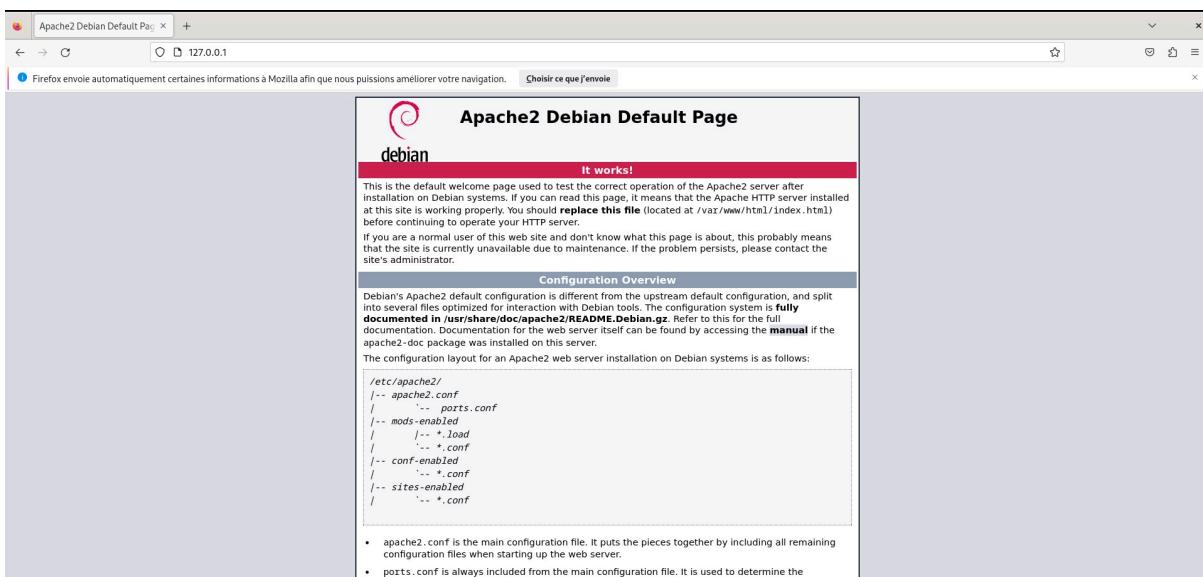
On fait la même chose depuis l'utilisateur

Puis, on installe les serveurs web avec cette commande : sudo apt install apache2 php libapache2-mod-php mariadb-server php-mysql



```
root@debian:/home/debian# cd
root@debian:~# nano .bashrc
root@debian:~# source .bashrc
bash: source: nom de fichier nécessaire en argument
source : utilisation :source nom_fichier [arguments]
bash: bashrc : commande introuvable
root@debian:~# nano .bashrc
root@debian:~# cd
root@debian:~# cd
root@debian:~# exit
exit
debian@debian:~$ nano.bashrc
bash: nano.bashrc : commande introuvable
debian@debian:~$ nano .bashrc
debian@debian:~$ source .bashrc
debian@debian:~$ sudo apt install apache2 php libapache2-mod-php mariadb-server php-mysql
```

Puis, se rendre sur firefox, et dans la barre de recherche, écrire "127.0.0.1" pour accéder au localhost



Nous allons maintenant, rediriger cette page à Netflix, Youtube et Google. Pour se faire nous devons modifier le index.html de notre serveur web Apache.

Pour se faire, d'abord se placer dans le dossier en se rendant dans le terminal et en écrivant “cd /var/www/html”

```
debian@debian:~$ cd /var/www/html
```

```
[sudo] password for debian:
```

Puis on supprime le fichier index.html et en créons un nouveau pour avoir un fichier vierge en tapant ces commandes : “sudo rm index.html”, puis “sudo nano index.html”.

```
debian@debian:/var/www/html$ sudo rm index.html
debian@debian:/var/www/html$ sudo nano index.html
debian@debian:/var/www/html$
```

Notre nouveau fichier vide s'affiche, maintenant rentrer le code html pour rediriger vers Netflix, Youtube et Google. Dans mon cas j'ai fait

```
<a href="https://www.netflix.com/browse">Netflix</a>
<cd>
<a href="https://www.youtube.com/">Youtube</a>
<cd>
<a href="https://www.google.fr/">Google</a>"
```

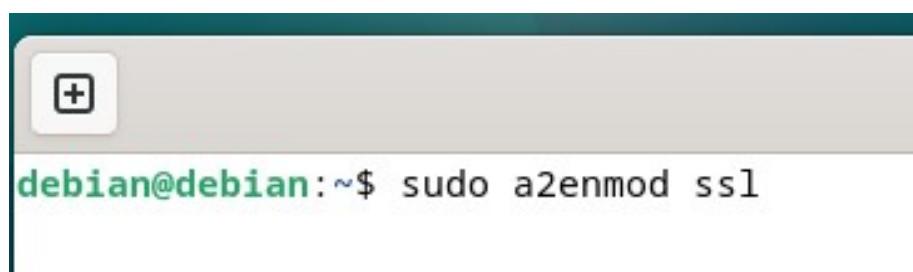
Puis refresh la page avec l'URL “127.0.0.1” et notre nouvelle page devrait s'afficher.



Maintenant, nous allons rendre notre page sécurisée grâce au protocole HTTPS.

Pour ce faire, il faut d'abord activer le module ssl grâce à la commande “sudo a2enmod ssl”

Puis taper la commande “sudo systemctl reload2” pour activer le module



```
debian@debian:~$ sudo a2enmod ssl
```

Puis nous allons créer un certificat TLS avec cette commande :
“sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048
-keyout /etc/ssl/private/apache-selfsigned.key -out
/etc/ssl/certs/apache-selfsigned.crt”



```
debian@debian:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Remplir les infos du certificat puis nous allons configurer Apache pour utiliser le certificat TLS. Rentrer la commande “sudo nano /etc/apache2/sites-available/your_domain_or_ip.conf”

On y rentre ces infos en remplaçant “your_domain_or_ip” par le lien de notre site :

```
<VirtualHost *:443>
    ServerName your_domain_or_ip
    DocumentRoot /var/www/your_domain_or_ip

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

```
GNU nano 7.2                                     debian@debian: ~
VirtualHost *:443>                               /etc/apache2/sites-available/your_domain_or_ip.conf
ServerName 127.0.0.1
DocumentRoot /var/www/127.0.0.1

SSLEngine on
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

```

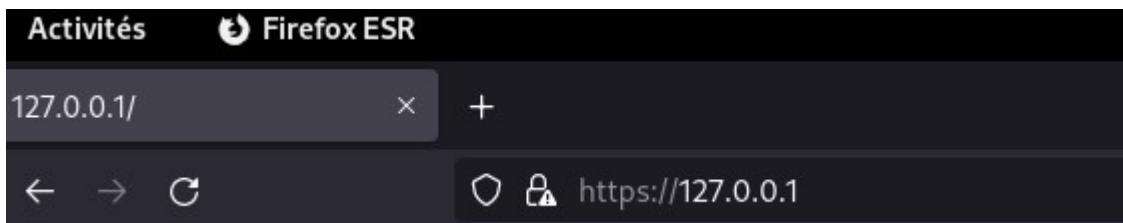
On crée le document pour notre site internet avec “sudo mkdir /var/www/your_domain_or_ip” puis “sudo nano /var/www/your_domain_or_ip/index.html” / Remplacer “your_domain_or_ip” par notre URL

```
debian@debian:~$ sudo mkdir /var/www/your_domain_or_ip
debian@debian:~$ sudo nano /var/www/your_domain_or_ip/index.html
```

Puis y rentrer le code HTML de notre site

```
GNU nano 7.2                                     /var/www/your_domain_or_ip/index.html
<a href="https://www.netflix.com/browse">Netflix</a>
<cd>
<a href="https://www.youtube.com/">Youtube</a>
<cd>
<a href="https://www.google.fr/">Google</a>
```

Nous finissons notre configuration avec cette commande :
“sudo a2ensite your_domain_or_ip.conf”
Puis nous pourrons accéder à notre site de façon sécurisée



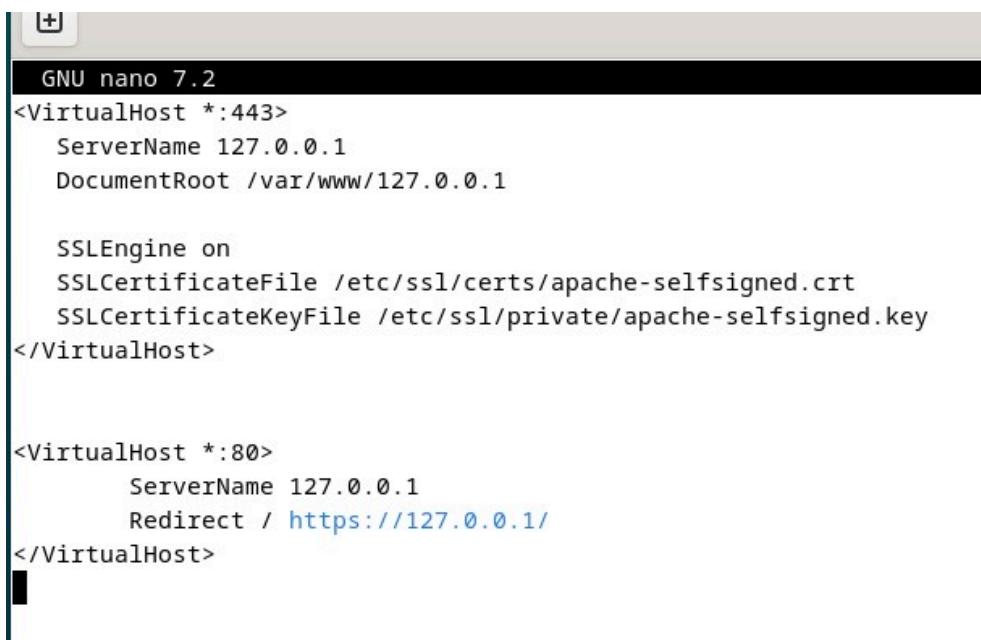
[Netflix](https://www.netflix.com/browse) [Youtube](https://www.youtube.com/) [Google](https://www.google.fr/)

Pour terminer, nous allons automatiser la redirection de notre site en HTTPS au lieu de l'écrire manuellement à chaque fois.

Rentrer la commande : “sudo nano /etc/apache2/sites-available/your_domain_or_ip.conf” avec notre nom personnalisé de fichier

Nous rajouterons ces lignes de code, toujours avec notre nom personnalisé pour le ServerName et le Redirect :

```
<VirtualHost *:80>
    ServerName your_domain_or_ip
    Redirect / https://your_domain_or_ip/
</VirtualHost>
```



The screenshot shows a terminal window titled "GNU nano 7.2". The text inside the window is the Apache configuration file content. It includes sections for port 443 (SSL) and port 80 (HTTP), with specific directives for server name and redirecting to HTTPS.

```
GNU nano 7.2
<VirtualHost *:443>
    ServerName 127.0.0.1
    DocumentRoot /var/www/127.0.0.1

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>

<VirtualHost *:80>
    ServerName 127.0.0.1
    Redirect / https://127.0.0.1/
</VirtualHost>
```