

---

## Serveur pxe linux

---

-Passer par l'étape 1 pour avoir tout les fichiers initialiser est préférable (pxelinux.cfg et pxelinux.0)

-Installer syslinux-common syslinux-efi tftpd-hpa pxelinux

**-Sur le dhcp next-server @ipPXE et filename pxelinux.0**

### Serveur TFTP

-Ensuite télécharger une image iso netboot de debian

A la racine du tftp copier les fichiers de /usr/lib/syslinux/modules/bios/\* dans /srv/tftp/

( au minimum ces fichiers ldlinux.c32, libcom32.c32, libutil.c32, mboot.c32, menu.c32, sanboot.c32 (optionnel), vesamenu.c32,chain.c32)

Et le fichier /usr/lib/PXELINUX/pxelinux.0 dans /srv/tftp

Et créer le dossier pxelinux.cfg dans /srv/tftp

Créer le fichier default dedans qui contiendra le menu que l'on définira plus bas

Après avoir installé tous les paquets plus haut et le paquet pxelinux

#Passe cette étape si tu veux des iso live etc

### NETBOOT

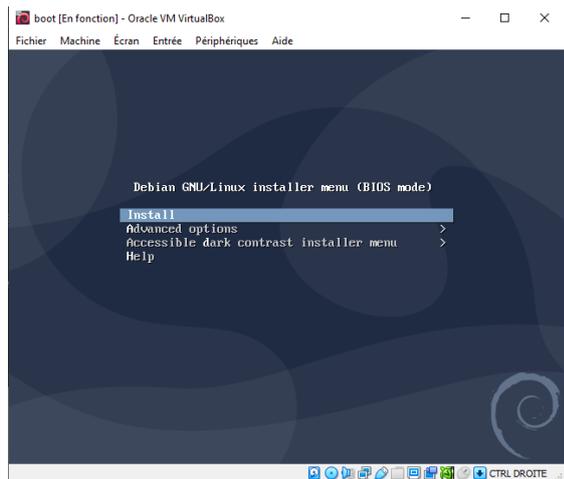
Aller à la racine du serveur tftp télécharger une image netboot

wget <http://ftp.debian.org/debian/dists/buster/main/installer-amd64/current/images/netboot/netboot.tar.gz>

Décompresser l'image et supprimer l'archive

booter sur le réseau avec une machine et sa installera automatiquement parceque on a que 1 os dans le serveur TFTP donc debian 10 sera installer automatiquement

Ici après boot sur le réseau



## Installation plus avancée (plusieurs image netboot) :

Pour booter plusieurs image il faut modifier le fichier defaut dans /pxelinux.cfg

Mais avant il faut executer ceci

```
Cp /usr/lib/syslinux/modules/bios/{libcom32.c32,menu.c32,libutil.c32}
```

Ces fichiers vont permettre de charger le menu attention bien choisir bios et pas efi64

Ensuite je me rend dans le fichier defaut

```
DEFAULT menu.c32
label Boot local disk
    kernel chain.c32
    append hd0

label Buster 64bit installer
    kernel debian-installer/amd64/boot-screens/vesamenu.c32
    append debian-installer/amd64/boot-screens/menu.cfg
```

Je précise que je vais utiliser un menu texte avec menu.c32

Ensuite je définis un titre avec label

Le kernel et append (pour ajouter le menu de la netinstall)

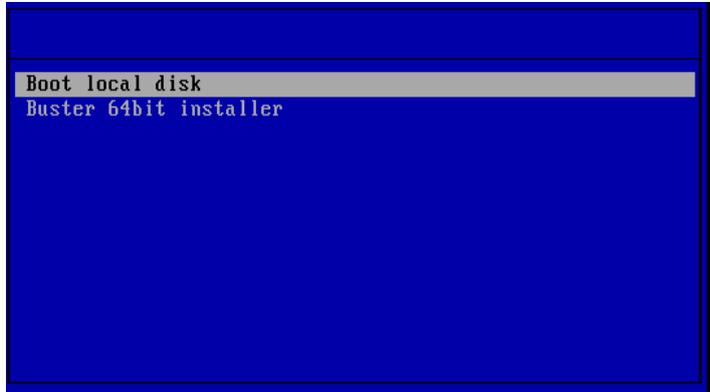
Ensuite viens ce qui nous interesse pour debian je definis un titre

Et le kernel sera le menu « vesamenu.c32 » pas besoin de preciser une image

Ensuite je rajoute le menu texte avec append

Je démarre via un boot pxe

Voila ce qui s'affiche



J'installe la nouvelle version de debian la 11

<http://ftp.debian.org/debian/dists/bullseye/main/installer-amd64/current/images/netboot/netboot.tar.gz>

Sa sera la bullseye

Je crée un dossier bullseye dedans je décompresse l'archive dedans et je rajoute ceci (je fais pareil pour kali)

```
DEFAULT menu.c32
label Boot local disk
    kernel chain.c32
    append hd0

label Buster 64bits Installer
    kernel debian-installer/amd64/boot-screens/vesamenu.c32
    append debian-installer/amd64/boot-screens/menu.cfg

label Bullseye 64 bits installer
    kernel bullseye/debian-installer/amd64/boot-screens/vesamenu.c32
    append bullseye/debian-installer/amd64/boot-screens/menu.cfg

label Kalilinux 64 bits installer
    kernel kali/debian-installer/amd64/boot-screens/vesamenu.c32
    append kali/debian-installer/amd64/boot-screens/menu.cfg
```

Pour l'installation il ne faut installer aucun paquet pendant l'installation tout sera installer après sinon cela bug

## Installation avec image ISO (la meilleur)

### Installation GPARTED

Il faut installer apache2 car un fichier de config qui contiendra la racine des fichiers de l'OS sera à mettre dans /var/www/html

Je suis ce tuto

Et j'installe GPARTED ici <https://gparted.org/download.php>

Ensuite je fais ceci après **avoir monté l'ISO dans tmp**

### Je créer un répertoire gparted dans le serveur tftp (mkdir /srv/tftp/gparted)

```
cp /tmp/live/{vmlinuz,initrd.img} /srv/tftp/gparted/
```

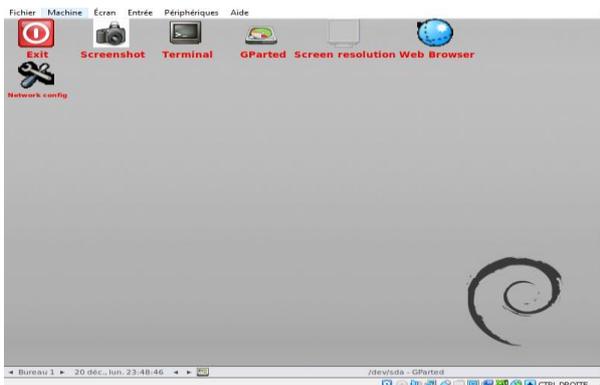
Ensuite je mets le système de fichier de l'image ISO à la racine du serveur web pour que le live puisse s'enclencher le client ira chercher les fichiers à la racine

```
cp /tmp/live/filesystem.squashfs /var/www/html
```

Dans le menu default ensuite je rajoute ceci

```
label GParted Live
    MENU LABEL GParted Live
    kernel gparted/vmlinuz
    append initrd=gparted/initrd.img boot=live config components
union=overlay username=user noswap noeject vga=788
fetch=http://$webserverIP/filesystem.squashfs
```

Donc filesystem permet de récupérer le système de fichier pour démarrer en live c'est pour ça que l'on a besoin d'un serveur web et de mettre à la racine le système de fichier



Si je veux mettre d'autre image live sa sera toujours le fichier filesystem.squashfs qui sera utiliser sauf qu'il est différent pour chaque ISO donc sur mon serveur web je crée un répertoire par iso live avec le nom de l'OS

Pour Kali j'ai mis sa (append peut-être écrit sous une forme plus condenser voir plus bas)

```
label Kali Live
    MENU LABEL Kali Live
    kernel livekali/vmlinuz
    append initrd=livekali/initrd.img boot=live config components union=overlay username=user no
swap noeject vga=788 fetch=http://192.168.1.21/kali/filesystem.squashfs
```

Ça fonctionne mais le système de fichier à vraiment beaucoup de mal à s'installer (fallait rajouter de la RAM)

Pour append init kali on peut se contenter de mettre que sa

```
label Kali Live
  MENU LABEL Kali Live
  kernel livekali/vmlinuz
  append initrd=livekali/initrd.img boot=live vga=788 fetch=http://192.168.1.21/kali/filesystem.squashfs
label Q4os Live
  MENU LABEL Q4os Live
  kernel q4os/vmlinuz
  append initrd=q4os/initrd.img boot=live vga=788 fetch=http://192.168.1.21/q4os/filesystem.squashfs
root@kaliboot:~#
```

Et il faut de 2 à 4GB de RAM pour que sa fonctionne

Q4os = Os linux GUI très léger

<https://sourceforge.net/projects/q4os/>

Même procédure on monte l'iso on créer répertoire sur serveur TFTP, on copie dedans vmlinuz et initrd .img

Et sur le serveur web dans html on crée un répertoire Q4os et on met le fichier filesystem.squashfs