

Créer son serveur PXE sous Linux (Branche Debian)

Prérequis

- Un système Linux (dans cet exemple, nous utiliserons Debian/Ubuntu)
- Droits d'administrateur (sudo)
- Une connexion réseau

Étape 1 : Installation de dnsmasq

1. Mettez à jour la liste des paquets :

```
sudo apt update
```

2. Installez dnsmasq :

```
sudo apt install dnsmasq
```

Étape 2 : Configuration de dnsmasq

1. Sauvegardez la configuration originale :

```
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

2. Créez un nouveau fichier de configuration :

```
sudo nano /etc/dnsmasq.conf
```

3. Ajoutez les lignes suivantes :

```
interface=eth0
dhcp-range=192.168.1.50,192.168.1.150,12h
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=/tftpboot
```

Assurez-vous de remplacer `eth0` par le nom de votre interface réseau et ajustez la plage DHCP selon votre configuration réseau.

Étape 3 : Préparation du répertoire TFTP

1. Créez le répertoire TFTP :

```
sudo mkdir -p /tftpboot
sudo chmod -R 777 /tftpboot
```

2. Installez syslinux pour obtenir les fichiers nécessaires au démarrage PXE :

```
sudo apt install syslinux pxelinux
```

3. Copiez les fichiers nécessaires :

```
sudo cp /usr/lib/PXELINUX/pxelinux.0 /tftpboot/  
sudo cp /usr/lib/syslinux/modules/bios/*.c32 /tftpboot/
```

Étape 4 : Configuration du menu PXE

1. Créez le répertoire de configuration PXE :

```
sudo mkdir -p /tftpboot/pxelinux.cfg
```

2. Créez un fichier de configuration par défaut :

```
sudo nano /tftpboot/pxelinux.cfg/default
```

3. Ajoutez un menu de base :

```
DEFAULT menu.c32  
PROMPT 0  
MENU TITLE PXE Boot Menu  
TIMEOUT 300  
  
LABEL local  
    MENU LABEL Boot from local drive  
    LOCALBOOT 0  
  
LABEL debian  
    MENU LABEL Install Debian  
    KERNEL debian-installer/amd64/linux  
    APPEND initrd=debian-installer/amd64/initrd.gz
```

Étape 5 : Ajout d'une image d'installation

Pour cet exemple, nous utiliserons Debian :

1. Téléchargez les fichiers d'installation netboot :

```
cd /tftpboot  
sudo wget http://ftp.debian.org/debian/dists/stable/main/installer-  
amd64/current/images/netboot/netboot.tar.gz  
sudo tar xzf netboot.tar.gz  
sudo rm netboot.tar.gz
```

Étape 6 : Redémarrage et vérification

1. Redémarrez le service dnsmasq :

```
sudo systemctl restart dnsmasq
```

2. Vérifiez que le service fonctionne correctement :

```
sudo systemctl status dnsmasq
```

Test du serveur PXE

1. Configurez un ordinateur client pour démarrer via PXE (généralement dans les paramètres BIOS/UEFI).
2. Démarrez l'ordinateur client. Il devrait recevoir une adresse IP de dnsmasq et afficher le menu PXE.
3. Sélectionnez l'option d'installation de Debian pour tester le démarrage PXE.

Ce tutoriel fournit une configuration de base pour un serveur PXE utilisant dnsmasq. N'oubliez pas d'ajuster les paramètres réseau et les chemins de fichiers selon votre environnement spécifique. Pour des configurations plus avancées ou l'ajout d'autres systèmes d'exploitation, vous devrez modifier le fichier de configuration PXE et ajouter les fichiers d'installation appropriés.

Étape 7 : Récupération d'un iso pour démarrer en live

1. Rendez-vous à l'adresse suivante : https://store.lacapsule.org/LACAPSULE/TOOLBOX/TOOLBOX_amd64.iso
2. Une fois le téléchargement terminé, monter l'iso. Dans notre cas, il se trouve dans les Téléchargements :
3. Créez un répertoire qui servira de point de montage pour l'image ISO :

```
sudo mkdir /mnt/iso
```

4. Utilisez la commande `mount` avec les options appropriées pour monter l'image ISO :

```
sudo mount -t iso9660 -o loop '/home/$USER/Téléchargements/TOOLBOX_amd64.iso' /mnt/iso
```

5. Une fois l'image ISO montée, vous pouvez accéder à son contenu en naviguant vers le point de montage :

```
cd /mnt/iso  
ls -la
```

6. Dans cette image iso, vous trouverez 5 dossiers dont un caché, symbolisé par un point devant son nom :

```
total 16
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 18:07 ./
drwxr-x---+ 3 root      root      4096 déc.  31 09:14 ../
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 17:57 boot/
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 18:07 .disk/
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 17:57 efi/
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 17:58 isolinux/
drwxr-xr-x  1 lacapsule lacapsule 2048 déc.  30 18:01 live/
```

7. Créer un répertoire **TOOLBOX** à la racine du répertoire **tftpboot** qui contiendra les fichiers nécessaires au démarrage du système

```
sudo mkdir /tftpboot/TOOLBOX
```

8. Nous allons copier l'intégralité des répertoires contenus dans notre **ISO** vers notre répertoire **TOOLBOX**

```
sudo cp -r /mnt/iso/* /tftpboot/TOOLBOX
```

Nous pouvons à présent ajouter une entrée de menu dans notre fichier default afin de permettre un boot sur le système contenu dans l'iso.

Lorsque vous avez terminé d'utiliser l'image **ISO**, vous pouvez la démonter :

```
sudo umount /mnt/iso
```

Étape 8 : Configuration du partage NFS

1. Installez le serveur NFS :

```
sudo apt install nfs-kernel-server
```

2. Configurez le partage NFS :

```
sudo nano /etc/exports
```

Ajoutez la ligne suivante :

```
/tftpboot *(ro,sync,no_subtree_check)
```

3. Redémarrez le service NFS :

```
sudo systemctl restart nfs-kernel-server && sudo exportfs -a
```

Vous pouvez à présent envoyer de gros fichiers tel qu'un filesystem.squashfs via le réseau afin de démarrer dessus en live.

Étape 9 : Modification du menu PXE

1. Ouvrir le fichier de configuration du menu :

```
sudo nano /tftpboot/pxelinux.cfg/default
```

2. Modifier les options pour inclure le partage NFS vers notre nouvelle distribution :

```
DEFAULT menu.c32
PROMPT 0
MENU TITLE PXE Boot Menu
TIMEOUT 300

LABEL local
    MENU LABEL Boot from local drive
    LOCALBOOT 0

LABEL debian
    MENU LABEL Install Debian
    KERNEL debian-installer/amd64/linux
    APPEND initrd=debian-installer/amd64/initrd.gz

# Modification du menu de Debian à notre distribution TOOLBOX
LABEL TOOLBOX      : Reconditionnement
    KERNEL TOOLBOX/live/vmlinuz-5.4.0-72-generic
    APPEND boot=live ksdevice=bootif rootfstype=nfs netboot=nfs
nfsroot=192.168.1.127:/tftpboot/TOOLBOX initrd=TOOLBOX/live/initrd.img-5.4.0-72-
generic ip=dhcp nosplash --
    TEXT HELP
    Une distribution faite pour le reconditionnement informatique
    ENDTEXT
```

Il sera nécessaire de modifier l'adresse IP **192.168.1.127** par celle de votre choix.

3. Pour savoir quelle IP devriez-vous utiliser, vous pouvez utiliser celle qui vous est retournée par la commande :

```
ip -4 addr show $(ip route show default | awk '/default/ {print $5}') | grep -oP
'(?<=inet\s)\d+(\.\d+){3}'
```

Étape 10 : Configuration du routage

Pour permettre aux clients ayant des adresses IP en 10.x.x.x d'accéder au réseau 192.168.1.x, vous devrez configurer le routage sur votre serveur PXE :

1. Activez le forwarding IP :

```
bash
sudo sysctl -w net.ipv4.ip_forward=1
```

2. Ajoutez une règle de NAT (Network Address Translation) :

```
bash
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

3. Assurez-vous que ces modifications persistent après un redémarrage en les ajoutant à `/etc/sysctl.conf` et en sauvegardant les règles iptables.

Étape 11 : Configuration de l'interface réseau

Configurez l'interface réseau de votre serveur PXE pour qu'elle ait une adresse IP dans les deux plages :

1. Éditez le fichier `/etc/network/interfaces` :

```
sudo nano /etc/network/interfaces
```

1. Ajoutez la configuration suivante :

```
auto eth0
iface eth0 inet static
    address 192.168.1.x # Adresse dans le réseau 192.168.1.x
    netmask 255.255.255.0
    gateway 192.168.1.1

iface eth0 inet static
    address 10.0.0.1 # Adresse dans le réseau 10.x.x.x
    netmask 255.255.255.0
```

Étape 12 : Redémarrage des services

Après avoir effectué ces modifications, redémarrez les services concernés :

```
sudo systemctl restart networking
sudo systemctl restart dnsmasq
sudo systemctl restart nfs-kernel-server
```

Cette configuration permettra à votre serveur PXE de distribuer des adresses IP dans la plage 10.x.x.x tout en permettant l'accès au réseau 192.168.1.x. Assurez-vous d'adapter les adresses IP et les interfaces réseau en fonction de votre configuration spécifique