

Configuration d'un serveur pour lancer des installations par PXE boot

Table of Contents

- Configuration d'un serveur pour lancer des installations par PXE boot
- Présentation de PXE
 - L'environnement PXE
 - mécanismes mis en jeu lors d'un démarrage sous environnement PXE.
- Configuration de l'environnement PXE
 - Configuration du client
 - Configuration du serveur TFTP
 - Installation
 - Démarrage et arrêt du serveur
 - Mise en place des fichiers à servir par TFTP
 - Mise en place des fichiers à servir pour des clients dotés d'un micrologiciel BIOS qui utilisent syslinux
 - Mise en place des fichiers à servir pour des clients qui utilisent le chargeur de démarrage GRUB2
 - Mise en place de la configuration de GRUB2
 - Mise en place des noyaux et des systèmes de fichiers initiaux
 - Configuration du serveur DHCP
 - Installation et configuration du serveur ****DHCP****
- Essais
- Réplication en local des dépôts pour accélérer les installations
 - Installation d'un serveur HTTP
 - Réplication d'une partie des dépôts
 - Adaptation du fichier grub.cfg
- Pour aller plus loin: l'outil Cobbler

Cette page explique comment procéder à une installation une distribution linux sans avoir recours à un quelconque média de démarrage. Une telle installation nécessite la mise en place d'un environnement particulier appelé environnement **PXE** — c'est-à-dire **P**reboot **E**xecution **E**nvironment ou Environnement d'exécution pré-démarrage — pour que l'ordinateur sur lequel on souhaite installer la distribution puisse trouver sur le réseau local un chargeur de démarrage ainsi que les éléments primordiaux de l'installateur que sont le noyau Linux et l'initramfs.

Ces éléments lui permettront ensuite d'accéder à l'Internet pour y retirer le reste du système d'installation, puis celui du système d'exploitation cible de son choix, grâce au menu que lui présentera le chargeur de démarrage.

Dans un premier temps nous expliquerons ce qu'est un tel environnement et comment le mettre en place. Dans un deuxième temps nous expliquerons où et comment accéder aux différentes versions du système à installer, soit sur les dépôts officiels de Fedora, soit sur un dépôt local pour diminuer les

temps de transfert.

Présentation de PXE

L'environnement PXE

Un environnement PXE est un environnement dans lequel un ordinateur dit « PXE capable » va chercher son chargeur de démarrage, non plus sur l'un de ses périphériques matériels mais sur le réseau. Cet environnement nécessite la configuration préalable de plusieurs parties, à savoir :

- un ordinateur qui soit « PXE capable » ;
- un serveur **DHCP**;
- un serveur **TFTP**.



Dans un tel environnement, le serveur **DHCP** et le serveur **TFTP** peuvent se trouver, soit sur un seul et même ordinateur, soit sur des ordinateurs séparés.

mécanismes mis en jeu lors d'un démarrage sous environnement PXE.

À la différence d'un démarrage classique durant lequel, après avoir exécuté son micrologiciel — **BIOS** ou **UEFI** selon le cas —, un ordinateur n'est plus dirigé sur son chargeur de démarrage habituel — situé selon le cas, dans le secteur de démarrage d'un de ses disques ou dans le dossier `/boot/efi/EFI` — mais sur une partie de code placée dans une mémoire morte spéciale liée à une de ses interfaces réseau. C'est cette partie de code qui va initier les premières requêtes DHCP qui vont permettre au serveur DHCP de répondre au client et de lui fournir les éléments d'initialisation fondamentaux suivants :

- une adresse IP ;
- l'adresse IP d'un serveur **TFTP** sur lequel télécharger son chargeur de démarrage et l'exécuter ;
- le nom complet du fichier désignant ce chargeur de démarrage.

L'ordinateur client est alors capable d'aller chercher ce chargeur de démarrage sur le serveur pour finalement l'exécuter.

La suite dépendra de ce que ce chargeur de démarrage prévoira, en particulier dans son menu, et nécessitera en général d'aller chercher le reste du système d'installation, puis d'exploitation, sur un serveur dont le type est assez libre (**FTP**, **HTTP**, **partage NFS** ou autre).

Configuration de l'environnement PXE

Configuration du client

La plupart des ordinateurs modernes sont « PXE capables » mais il faut configurer le démarrage PXE.

Redémarrer et entrer dans la configuration du micrologiciel BIOS.

Une fois dans cette configuration, il faut :

- activer le démarrage par le réseau.

Voici quelques exemples qui montrent que la terminologie et l'emplacement dans le menu varient beaucoup d'un fabricant à l'autre :

```
Carte mère Gigabyte P55A-UD4 : Integrated Peripherals -> Onboard LAN
Boot ROM
Carte mère Asusrock B250 Pro4 : Advanced -> Boot -> Boot From Onboard
LAN
PC portable Gigabyte P27 GV : Boot -> UEFI Settings ->Net Work Stack
(doit être activé) -> IPv4 PXE Support
PC portable Asus X750JV : Advanced -> Network Stack (doit être activé)
-> IPv4 PXEsupport
```

- configurer la séquence de boot pour mettre PXE en premier dans l'ordre de démarrage

Configuration du serveur TFTP

Installation

Sur l'ordinateur qui accueillera le serveur **TFTP** — et cela peut-être le même ordinateur que celui qui accueillera le serveur DHCP ou un autre —, il faut en premier lieu installer le paquet **tftp-server**. Pour cela utiliser la commande suivante :

```
# dnf install tftp-server
```

Démarrage et arrêt du serveur

Pour démarrer et arrêter le serveur, il faut faire appel à **systemd** qui gère les services. Voici ces commandes :

Pour démarrer ponctuellement le service

```
# systemctl start tftp.service
```

Pour démarrer automatiquement au démarrage de l'ordinateur

```
# sytemctl enable tftp.service
```

Pour arrêter le service

```
# systemctl stop tftp.service
```

Pour vérifier l'état du service

```
# systemctl status tftp.service
```

Mise en place des fichiers à servir par TFTP

Il y a plusieurs fichiers qu'il faut rendre disponibles sur le serveur **TFTP**, à savoir :

- le chargeur de démarrage et ses fichiers auxiliaires dont le fichier de configuration de son menu ;
- les différents noyaux et **initramfs**.

Ces fichiers devront être rendus disponibles à la fois pour les ordinateurs dotés d'un micrologiciel BIOS et pour ceux dotés d'un micrologiciel **UEFI**.

Par ailleurs, il faudra aussi déterminer quel chargeur de démarrage devant être utilisés. On va traiter les cas suivants :

- utilisation du chargeur de démarrage **syslinux** dans le cas des clients dotés d'un micrologiciel BIOS ;
- utilisation du chargeur de démarrage **GRUB2**:
 - dans le cas de clients dotés du micrologiciel **BIOS**;
 - dans le cas clients dotés du micrologiciel **UEFI**.

Mise en place des fichiers à servir pour des clients dotés d'un micrologiciel BIOS qui utilisent syslinux

Cette section traite de la mise en place du chargeur de démarrage et fichiers associés pour une installation uniquement sur des ordinateurs clients dotés d'un micrologiciel **BIOS**.

Pour disposer du chargeur de démarrage **syslinux**, il vous faut installer le paquet **syslinux** à l'aide de la commande suivante :

```
# dnf install syslinux
```

Le répertoire racine du serveur **TFTP** est `/var/lib/tftpboot/`. C'est donc dans ce dossier que doivent se trouver les fichiers de **syslinux**:

Pour bénéficier de possibles mises à jour de **syslinux**, établir des liens symboliques vers ces derniers :

```
# ln -s  
/usr/share/syslinux{pxelinux.0,vesamenu.c32,ldlinux.c32,libcom32.c32,libutil  
.c32} /var/lib/tftpboot/
```

Enfin, créer un dossier `/var/lib/tftpboot/pxelinux.cfg` pour y accueillir le fichier de configuration du menu :

```
# mkdir -p /var/lib/tftpboot/pxelinux.cfg
```

Le fichier contenant le menu doit se nommer `default` et être placé dans le dossier l'on viens de créer. Voici un exemple d'un tel menu :

```
default vesamenu.c32  
prompt 1  
timeout 600  
label linux  
menu label ^Install Fedora Workstation 27 64-bit  
menu default  
kernel f27/vmlinuz  
append initrd=f27/initrd.img  
inst.stage2=http://download.fedoraproject.org/pub/fedora/linux/releases/27/W  
orkstation/x86_64/os/ ip=dhcp  
  
label linux  
menu label ^Install Fedora Server 26 64-bit  
menu default  
kernel f26/vmlinuz  
append initrd=f26/initrd.img  
inst.stage2=http://download.fedoraproject.org/pub/fedora/linux/releases/26/S  
erver/x86_64/os/ ip=dhcp  
  
label rescue  
menu label ^Rescue installed system  
kernel f26/vmlinuz  
append initrd=f26initrd.img ip=dhcp  
root=live:http://download.fedoraproject.org/pub/fedora/linux/releases/26/Ser  
ver/x86_64/os/LiveOS/squashfs.img rescue  
  
label local  
menu label Boot from ^local drive  
localboot 0xffff
```

- **kernel** fait référence à un noyau linux sous la forme `kernel f27/vmlinuz`, dans laquelle `f27`

représente un sous-dossier f27 de la racine du serveur **TFTP** et **vmlinuz**, l'image du noyau F27 concerné.

- **append initrd** fait référence à un système de fichiers initial sous la forme `initrd=f27/initrd.img`, laquelle f27 représente le même dossier que dans le cas précédent et `initrd.img` l'image du système de fichiers initial.

Il faut donc créer, à la racine du serveur **TFTP**, un dossier pour chacune des versions de système et y placer les noyaux ainsi que les `initrd.img` correspondants.

Mise en place des fichiers à servir pour des clients qui utilisent le chargeur de démarrage GRUB2

Créer, à la racine du serveur **TFTP**, une structure de dossiers permettant à **GRUB2** de trouver tout ce dont il a besoin. **GRUB2** permet de le faire en un clin d'œil grâce à une commande dédiée. Néanmoins, avant de l'utiliser, il faut s'assurer que l'on dispose bien des éléments nécessaires pour l'architecture cible, car a priori après une installation standard ce n'est pas le cas.

Il faut installer le paquet `grub2-tools-efi.<architecture>`.

ARCHITECTURE CLIENT i386-pc

La commande générique précédente est :

```
# dnf install grub2-tools-efi.i386-pc
```

Cela devrait aboutir à la création et au peuplement d'un dossier `/usr/lib/grub/i386-pc` qui contient tout le nécessaire à la commande qu'on va utiliser maintenant.

Cette commande va créer l'arborescence des dossiers nécessaires à l'environnement **PXE** dans le serveur **TFTP**:

```
# grub2-mknetdir --net-directory=/var/lib/tftpboot --subdir=/boot/grub -d /usr/lib/grub/i386-pc
```



Dans cette commande, il faut bien sûr adapter le chemin vers la racine du serveur **TFTP**.

Noter le path dans la phrase qui apparaît dans le terminal à la fin du compte-rendu de la commande :

```
« Configurez le serveur DHCP pour pointer vers /var/lib/tftpboot/boot/grub/i386-pc/core.0 ».
```

qu'il faudra indiquer lors de la configuration du serveur DHCP.

ARCHITECTURE CLIENT x86_64-efi

On peut répéter les 2 commandes précédentes pour le cas de l'architecture x86_64-efi, soit :

```
# dnf install grub2-tools-efi.x86_64-efi
```

et

```
# grub2-mknetdir --net-directory=/var/lib/tftpboot --subdir=/boot/grub -d /usr/lib/grub/x86_64-efi
```

Noter le path dans la phrase qui apparaît dans le terminal à la fin du compte-rendu de la commande :

```
« Configurez le serveur `DHCP` pour pointer vers  
**/var/lib/tftpboot/boot/grub/x86_64-efi/**core.efi ».
```

qu'il faudra donc indiquer lors de la configuration du serveur DHCP.

Maintenant, si on explore l'arborescence des fichiers dans le serveur TFTP, dans `boot/grub/<architecture>`, on y trouvera un fichier `grub.cfg` dont le contenu est : `source /boot/grub/grub.cfg`. Cela invite donc à placer la configuration de GRUB2 dans `boot/grub/grub.cfg` qui est un dossier commun aux deux architectures. Le même fichier de configuration de GRUB2 peut donc être utilisé.



L'utilisation, ultérieure, de la commande `grub2-mknetdir` n'écrasera pas l'arborescence créée lors de sa première utilisation avec une architecture client différente. Elle ne fait que la compléter.



Avec cette méthode, il faut désactiver momentanément le **Secure boot** dans le micrologiciel **UEFI**.

ARCHITECTURE CLIENT x86_64-efi : méthode alternative

Créer un dossier à la racine du serveur **tftpboot**. Pour l'exemple on l'appellera **UEFI** :

```
# mkdir /var/lib/tftpboot/UEFI
```

Monter l'image iso d'installation:

```
# mkdir /mnt/iso

# mount -t iso9660 -o loop <chemin et nom de l'image iso> /mnt/iso
```

et copier les fichiers des chargeurs de démarrage nécessaires dans le dossier <racine du serveur tftp>/UEFI/:

```
# cp /mnt/iso/EFI/BOOT/{BOOTX64.EFI,grubx64.efi} /var/lib/tftpboot/UEFI/
```

Démonter l'image iso :

```
# umount /mnt/iso
```

Mise en place de la configuration de GRUB2

Le fichier de configuration, dont un exemple est fourni ci-dessous, doit être placé dans (en partant de la racine du serveur tftpboot) :

- le dossier boot/grub si on utilise la méthode de base ;
- le dossier **UEFI** si on utilise la méthode alternative.

Un certain nombre de modules de **GRUB2** sont déjà pré-embarqués dans les images essentielles (core image en anglais). Ces modules sont les suivants :



mincmd, configfile, normal, gzio, gettext, extcmd, crypto, terminal, tftp, net, bufio, datetime, boot, priority_queue

pour core.efi (x86_64-efi) : efinet,

pour core.0 (i386-pc) : pxe, net

Aucun module supplémentaire n'est strictement nécessaire au démarrage. Néanmoins, si on ne prévoit pas les modules nécessaires, le démarrage se fera complètement en aveugle et il faudra attendre un certain temps de téléchargement avant de voir l'image d'anaconda apparaître à l'écran.

On a vu que cette configuration est commune aux deux architectures. En voici un exemple :

```
function load_video {
insmod efi_gop
insmod efi_uga
insmod video_bochs
insmod video_cirrus
insmod all_video
}
load_video
```



```
set gfxpayload=keep
set net_default_server=192.168.1.19
menuentry 'Installer Fedora 26 Server 64-bits' --class fedora --class gnu-
linux --class gnu --class os {
    echo 'Message de grub.cfg : chargement du noyau vmlinuz'
    linux (tftp)/f26/vmlinuz ip=dhcp
    inst.repo=http://download.fedoraproject.org/pub/fedora/linux/releases/26/Ser
ver/x86_64/os/
    echo 'Message de grub.cfg : chargement du système de fichiers initial'
    initrd (tftp)/f26/initrd.img
}
menuentry 'Installer Fedora 26 Workstation 64-bit' --class fedora --class
gnu-linux --class gnu --class os {
    echo 'Message de grub.cfg : chargement du noyau vmlinuz'
    linux (tftp,192.168.1.19)/f26/vmlinuz ip=dhcp
    inst.repo=http://download.fedoraproject.org/pub/fedora/linux/releases/26/Wor
kstation/x86_64/os/
    echo 'Message de grub.cfg : chargement du système de fichiers initial'
    initrd (tftp,192.168.1.19)/f26/initrd.img
}
menuentry 'Installer Fedora 27 Server 64-bits' --class fedora --class gnu-
linux --class gnu --class os {
    echo 'Message de grub.cfg : chargement du noyau vmlinuz'
    linux (tftp)/f27/vmlinuz ip=dhcp
    inst.repo=http://download.fedoraproject.org/pub/fedora/linux/releases/27/Ser
ver/x86_64/os/
    echo 'Message de grub.cfg : chargement du système de fichiers initial'
    initrd (tftp)/f27/initrd.img
}
menuentry 'Installer Fedora 27 Workstation 64-bit' --class fedora --class
gnu-linux --class gnu --class os {
    echo 'Message de grub.cfg : chargement du noyau vmlinuz'
    linux (tftp)/f27/vmlinuz ip=dhcp
    inst.repo=http://download.fedoraproject.org/pub/fedora/linux/releases/27/Wor
kstation/x86_64/os/
    echo 'Message de grub.cfg : chargement du système de fichiers initial'
    initrd (tftp)/f27/initrd.img
}
menuentry 'Rescue installed system' --class fedora --class gnu-linux --
class gnu --class os {
    kernel f26/vmlinuz
    append f26/initrd=initrd.img
    root=live:http://download.fedoraproject.org/pub/fedora/linux/releases/26/Ser
ver/x86_64/os/LiveOS/squashfs.img rescue
```

Bien entendu ceci n'est qu'un exemple et il faut prévoir des entrées pour d'autres distributions.

lorsqu'on utilise la méthode alternative pour les clients dotés d'un micrologiciel **UEFI**, il faut remplacer les commandes **linux** et **initrd** de **GRUB2** par **linuxefi** et **initrdefi** respectivement.

Par ailleurs les chemins vers les noyaux et les **initrd** doivent être modifiés pour commencer par (tftp)/.. (p. ex. (tftp)/../f26/initrd)

Mise en place des noyaux et des systèmes de fichiers initiaux

GRUB2 va chercher les noyaux et les systèmes de fichiers initiaux dans des dossiers du type <version>, nom dans lequel <version> est à remplacer par la version à installer.

Créer d'abord ces dossiers :

```
# mkdir -p /var/lib/tftpboot/{f26,f27}
```

Pour chacun de ces dossiers, il vous suffit de faire :

```
# wget
http://download.fedoraproject.org/pub/fedora/linux/releases/<version>/Server
/x86_64/os/images/pxeboot/vmlinuz -O /var/lib/tftpboot/f<version>/vmlinuz
```

commande dans laquelle <version> est à remplacer par le numéro de la version concernée, soit pour la version 26 :

```
# wget
http://download.fedoraproject.org/pub/fedora/linux/releases/26/Server/x86_64
/os/images/pxeboot/vmlinuz -O /var/lib/tftpboot/f26/vmlinuz
```

Cette commande est à répéter pour les autres distributions, ainsi que pour les **initrd** pour lesquels la commande ressemble à :

```
# wget
http://download.fedoraproject.org/pub/fedora/linux/releases/26/Server/x86_64
/os/images/pxeboot/initrd.img -O /var/lib/tftpboot/f26/initrd.img
```

Configuration du serveur DHCP



Pour connaître le nom de des interfaces exécuter la commande **ip address** sans paramètre.

Installation et configuration du serveur ****DHCP****

En premier lieu, installer le paquet **dhcp-server**:

```
# dnf install dhcp-server
```

Une fois installé, la configuration du serveur se fait dans le fichier `/etc/dhcp/dhcpd.conf`.

Voici un exemple de configuration

```
ddns-update-style none;
ignore client-updates;
authoritative;
# allow booting;
# allow bootp;
option arch code 93 = unsigned integer 16; # RFC4578

subnet 192.168.1.0 netmask 255.255.255.0 {
    next-server 192.168.1.19;          # adresse du serveur tftp
    if option arch =00:07
    {
        filename "boot/grub/x86_64-efi/core.efi";
    } else
    {
        filename "boot/grub/i386-pc/core.0";
    }
    option routers 192.168.1.1;        # passerelle par défaut
    option subnet-mask 255.255.255.0;  # masque de sous-réseau
    option domain-name "home.lan";     # nom de domaine
    option domain-name-servers 192.168.1.1; # serveurs DNS
    range 192.168.1.20 192.168.1.100;  # plage d'adresse
    default-lease-time 21600;          # durée du bail en secondes
    max-lease-time 43200 ;             # durée maxi du bail en sec.

#Ici on passe le serveur d'essai et le client en IP fixes
    host PC-JO { # le serveur DHCP et TFTP
        hardware ethernet 70:85:c2:49:b8:df;
        fixed-address 192.168.1.19;
    }
    host PC-CHAMBRE { # le client est en IP fixe mais ça n'a rien
d'obligatoire
        hardware ethernet 1c:6f:65:93:55:39;
        fixed-address 192.168.1.127;
    }
}
```

- Les directives **Allow booting** et **Allow bootp** sont mises en commentaire, car bien qu'elles apparaissent dans tous les exemples, elles ne sont pas indispensables puisqu'il s'agit du comportement par défaut ;
- La ligne option **arch code 93 = unsigned integer 16**; permet de nommer l'option 93 et de donner son format ; elle correspond à l'architecture du client que ce dernier présente au serveur lors de sa requête **DHCP Discovery**;
- L'option **next-server** permet d'indiquer au client quelle adresse IP utiliser en tant que serveur

TFTP;

- Le test de la valeur **arch** (architecture du client) permet de l'aiguiller sur le bon chargeur de démarrage ;
 - Pour utiliser le chargeur de démarrage **syslinux** sur les clients dotés d'un micrologiciel **BIOS**, remplacer la ligne de la branche **else** du test d'architecture, par `filename = "pxelinux.0;"`
- l'option **filename** permet d'indiquer au client le nom du fichier à requérir (chargeur de démarrage) du serveur **TFTP**;
 - Pour utiliser la méthode alternative pour la mise en place de **GRUB2** pour les clients dotés d'un micrologiciel **UEFI**, remplacer la ligne relative au fichier de boot à charger en **UEFI** par `filename = "UEFI/B00TX64.EFI;"`
- pour finir les déclarations `host` permettent d'affecter à certains clients une adresse IP fixe.

Essais

Tout ce qu'il vous faut pour installer Fedora ou d'autres systèmes d'exploitation est maintenant en place.

Il faut relancer le service **dhcpcd** à chaque modification sa configuration, puis vérifier que les deux serveurs, **DHCP** et **TFTP** sont actifs.

Démarrer le serveur en ayant préalablement choisi l'option de démarrage PXE. En quelques secondes le menu de GRUB (ou de syslinux) devrait s'afficher.

Réplication en local des dépôts pour accélérer les installations

Lorsqu'on a nombreuses installations à effectuer, cela vaut peut valoir la peine de répliquer les dépôts sur un serveur **HTTP** local.

Installation d'un serveur HTTP

L'installation d'un serveur **HTTP** n'est pas très compliquée.

Une fois le serveur installé, créer les dossiers d'accueil suivants, extensibles par ramification à d'autres versions :

```
# mkdir -p /var/www/html/fedora/linux/releases/27/Everything/x86_64/os
```

Réplication d'une partie des dépôts

Pour répliquer seulement la partie du serveur correspondant à Workstation, utiliser une commande qui ressemble à celle-ci :

```
$ rsync -avrt  
rsync://rsync.mirrorservice.org/dl.fedoraproject.org/pub/fedora/linux/releases/27/Workstation/x86_64/os/  
/html/fedora/linux/releases/27/Everything/x86_64/os
```



seul ce dépôt est nécessaire à une première installation, pour que les mises à jour se fassent sur un dépôt updates local, il faudra également le répliquer et ensuite modifier le dossier `/etc/yum.conf.d/` en conséquence.

Pour télécharger les dépôts, les dossiers doivent être accessibles au démon `httpd`, en supposant qu'on est dans le dossier `/var/www/html/` et que `selinux` est en mode enforcing :

```
# usermod -aG apache <votre nom d'utilisateur>  
# chown -R <votre nom d'utilisateur>:apache fedora  
# chmod -R 750 fedora  
# chcon -R -t httpd_sys_content_rw_t fedora
```

Adaptation du fichier `grub.cfg`

Maintenant, il faut que les entrées dans le menu de GRUB2 pointent sur ce dépôt local :

```
menuentry 'Installer Fedora 27 Workstation 64-bit' --class fedora --class  
gnu-linux --class gnu --class os {  
    echo 'Message de grub.cfg : chargement du noyau vmlinuz'  
    linuxefi (tftp)/../f27/vmlinuz ip=dhcp  
    inst.repo=http://192.168.1.19/fedora/linux/releases/27/Everything/x86_64/os/  
    echo 'Message de grub.cfg : chargement du système de fichiers initial'  
    initrdefi (tftp)/../f27/initrd.img  
}
```

La partie à adapter est l'adresse du serveur **HTTP**, ou son nom d'hôte si la configuration DNS est au point.

Pour aller plus loin: l'outil Cobbler

Cobbler est un serveur d'installation **PXE** comportant en standard de nombreuses fonctions pratiques pour les administrateurs réseau et difficilement réalisables manuellement..

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=centos:pxe-install>

Last update: **2025/02/19 10:59**

