

Environnement serveur PXE Boot

Le présent article explique comment mettre en place un environnement PXE Boot (*Pre-eXecution Environment*) hébergé sous CentOS 6.x avec SELinux actif. Un tel environnement permet de démarrer des machines physiques (virtuelles) sans avoir besoin d'y installer de système d'exploitation et est donc particulièrement intéressant dans le cas de machines qui ne sont pas dotées d'une mémoire de masse locale encore appelées machine *diskless*. Bien que dans le cas de ces dernières il est courant de trouver un stockage réseau au travers de NFS, ce n'est cependant pas une obligation, la machine pouvant être directement raccordée à une baie de stockage au travers d'un réseau dédié (*Storage Area Network - SAN*) Fibre Channel¹⁾, Infiniband ou autre.

Comment fonctionne un environnement PXE Boot ?

Une légende urbaine veut que *PXE boot* soit un protocole réseau ou une implémentation d'une spécification. **En fait il n'en est absolument rien car PXE n'est rien d'autre qu'une spécification** reposant entre autres sur l'usage de protocoles réseau bien connus comme DHCP et TFTP (PXE boot repose donc la classique architecture client-serveur). Bien que la spécification PXE boot mentionne l'existence du protocole BOOTP, nous en ferons abstraction ici et DHCP couvrira, par un abus de langage impardonnable et totalement volontaire, les deux protocoles.

Ce qu'il se passe lors d'un amorçage PXE boot

Le dialogue entre une station cliente (C) et un serveur (S) configuré pour servir un environnement PXE boot est, en gros, le suivant:

1. C commence par envoyer sur le réseau un *DHCP Discover*²⁾. La requête envoyée est quasi-similaire à une requête *DHCP Discover* standard, excepté que C y glisse quelques informations supplémentaires à son sujet³⁾ ;
2. S voit passer le *DHCP Discover* émis par C et y répond en faisant une offre (*DHCP Offer*). Cette offre contient notamment l'adresse IP proposée à C ainsi qu'une liste de serveurs à partir desquels il pourra s'amorcer (liste de serveurs de démarrage). Si plusieurs serveurs sont à l'écoute et donc répondent C aura plusieurs offres ;
3. C collecte le *DHCP Offer* renvoyé par S. Si plusieurs offres DHCP ont été faites (cas où plus d'un serveur DHCP est présent), C va choisir une des offres faites et émettre un *DHCP Request* pour confirmer l'utilisation de l'adresse IP qui lui a été assignée ;
4. S va, à la suite de la réception de la *DHCP Request* de C, émettre un accusé de réception de la confirmation et envoyer à C un *DHCP Acknowledgement* (*DHCP Ack*). Dans le cas d'un serveur DHCP en environnement PXE boot, le *DHCP Ack* contient en plus le nom du fichier que C devra récupérer auprès d'un des serveurs d'amorçage dont il a obtenu la liste précédemment ;

A ce stade, C connaît ses paramètres réseau, le fichier qu'il doit utiliser pour la suite de son processus de démarrage et la liste des serveurs (TFTP) qu'il peut contacter pour l'obtenir... ce qu'il va s'empresser de faire. Une fois qu'un serveur TFTP a pu être contacté et le fichier de démarrage récupéré, C va exécuter le contenu dudit fichier qui va à son tour aboutir à faire charger à C un noyau du système d'exploitation et bien évidemment l'exécuter.

Tout cela ne ne vous rappelle-t'il pas quelque chose? Le chargeur (Grub, LILO, etc) situé sur le disque dur

de votre ordinateur ! Le processus est cependant un rien plus complexe car il implique quelques acteurs externes supplémentaires externes à la machine, reste que dans les grandes lignes la philosophie reste la même.

La seule différence entre un échange DHCP standard et PXE est l'utilisation de l'option DHCP 60 (Vendor class identifier⁴⁾). Une option DHCP est un paramètre envoyé par un serveur DHCP (il en existe plus d'une dizaine) à un client DHCP et permet de spécifier des paramètres tels que son adresse IP, sa passerelle, l'adresse des serveurs de nom, etc.

Capacité d'amorçage en PXE boot

Bien évidemment un environnement PXE boot n'est pas la seule manière d'amorcer un ordinateur au travers du réseau, il est cependant désormais supporté par un grand nombre de machines, majoritairement IA-32 (x86), cette architecture étant de nos jours courante.⁵⁾ En passant, si vous avez utilisé des machines Sun/Oracle SPARC, vous aurez peut être remarqué que le fameux *netboot* est un mécanisme quasi-identique reposant également sur BOOTP/TFTP (ou DHCP pour les versions plus récentes d'OpenBoot). Dans le même style, citons également:

- [Remote Initial Program Load \(RIPL\)](#) qui émane originalement d'IBM et qui permettait de démarrer des machines sur des réseaux Novell NetWare ;
- [Etherboot](#) (spécifique au monde IA-32/x86) ;

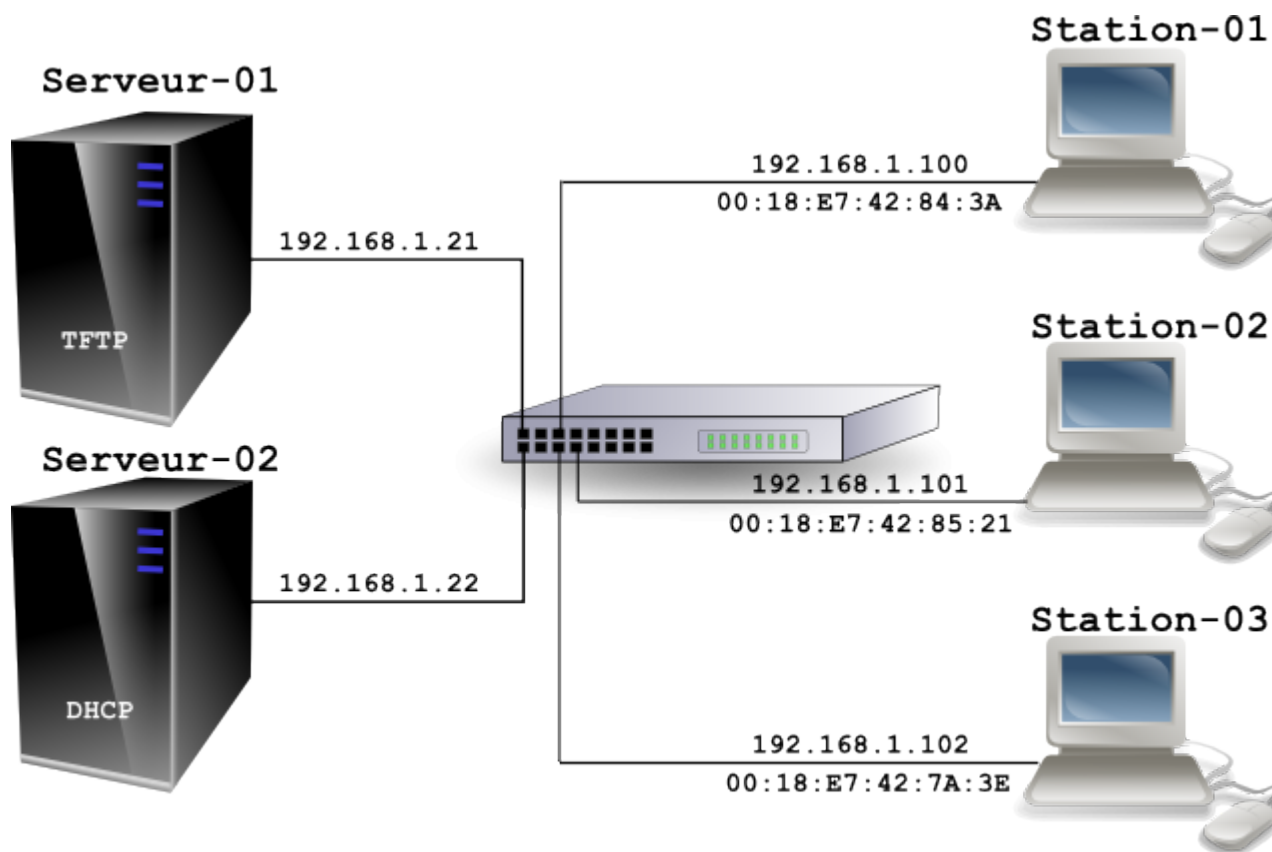
Rien de magique cependant, l'amorçage d'une machine dans un environnement *PXE Boot* nécessite la présence d'un micrologiciel permettant de dialoguer avec les différents acteurs. Ce micrologiciel est stocké dans une mémoire non volatile et est donc typiquement situé sur l'adaptateur réseau lui-même ou être partie intégrante du micrologiciel de la machine. Il est par ailleurs indispensable que la séquence de démarrage « donne la main » au micrologiciel d'amorçage PXE.

Au niveau client, il existe plusieurs implémentations libres de la spécification PXE, parmi lesquelles:

- gPXE (intégré à Qemu/KVM)
- iPXE (permet d'accéder à des volumes SAN notamment au travers de iSCSI et FCoE)

Attention, [PXELINUX](#) (dérivé de SYSLINUX) n'est pas une implémentation cliente PXE mais un des multiples chargeurs de noyau Linux qui peut être chargé au travers d'un réseau par un client PXE. Nous aurons l'occasion d'y revenir un peu plus loin, PXELINUX est précisément le chargeur Linux mis en oeuvre ici.

Mise en place de l'environnement PXE



La configuration réseau à laquelle le reste de l'article fait référence est la suivante:

- Les trois machines situées sur la droite de l'image et identifiées « Station » sont des machines ayant une mémoire de masse locale⁶⁾ mais aucun système d'exploitation n'y est installé. Elles devront donc charger le programme d'installation de leur système d'installation au travers du réseau. Le système installé sera, pour les besoins de la cause, CentOS 6.x.
- Les deux machines situées sur la gauche de l'image et identifiées « Serveur » sont les serveurs TFTP et DHCP (il est tout à fait possible de regrouper les services DHCP et TFTP sur une machine unique). **Il est assumé dans le reste de l'article que ces deux machines fonctionnent sous Linux CentOS 6.x.**

Configuration du serveur DHCP

Pour commencer il faut installer le paquet **dhcp** et ses éventuelles dépendances sur la machine **serveur-02**:

```
invite=root@serveur-02 ~ |commande=yum install dhcp
```

Il faut ensuite éditer le fichier **/etc/dhcpd.conf** qui est, dans notre cas de figure, relativement simple:

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
  
    # Par défaut le serveur DHCP n'a pas autorité  
    authoritative ;  
}
```

```
option subnet-mask 255.255.255.0
option domain-name test.lan

# Si non précisé est par défaut le serveur DHCP lui-même
next-server 192.168.1.21;

# Fichier (SYSLINUX/PXELINUX) que les stations vont récupérer sur le
serveur TFTP
filename pxelinux.0;

host station-01 {
    hardware ethernet 00:18:E7:42:84:3A;
    fixed-address 192.168.1.101;
    option host-name station-01;
}

host station-02 {
    hardware ethernet 00:18:E7:42:85:21;
    fixed-address 192.168.1.102;
    option host-name station-02;
}

host station-03 {
    hardware ethernet 00:18:E7:42:7A:3E;
    fixed-address 192.168.1.102;
    option host-name ;
}
}
```

Il n'y a pas grand chose à dire sur cette configuration relativement minimaliste hormis que chaque machine à une configuration d'adressage « fixe » par rapport à son adresse MAC (vous aurez peut être remarqué que nous ne précisons aucune passerelle, ce n'est pas nécessaire car l'environnement est cloisonné). Noter cependant que:

- A la ligne 4, la directive *authoritative* :
- A la ligne 10, la directive *next-server* : cette directive spécifie à un client PXE le prochain serveur qu'il doit contacter pour continuer son processus d'amorçage (le serveur TFTP donc). Pour l'exemple, les serveurs DHCP et TFTP sont sur des machines différentes et la directive est ici nécessaire. Dans le cas où ils sont hébergés sur une même machine il est possible d'omettre la directive *next-server* (designera alors le serveur DHCP lui-même);
- A la ligne 13:

Plusieurs pages ailleurs sur le web font mention des options 128 et 129 et les définissent au début du fichier **/etc/dhcpd.conf**⁷⁾. Dans notre cas nous n'en avons pas besoin, d'une part car ces options sont déjà connues de la version de dhcpd utilisée

Configuration du serveur TFTP

Mise en places des fichiers nécessaires aux stations

Voir aussi

[HowTo: Setup your own PXE Boot Server using Ubuntu Server](#)

[HowTo: Get an Ubuntu Live CD to boot off a PXE server](#)

1)

et pas Fiber Channel...

2)

Etant donné que C ne connaît pas l'existence d'un serveur DHCP à ce stade, la requête est émise en diffusion broadcast au niveau 2.

3)

Entre autres une identifiant unique (UUID) et l'architecture système client.

4)

Voir [RFC 2132](#)

5)

La spécification PXE ne fait cependant pas référence à une architecture matérielle particulière, des architectures non IA-32 comme Itanium (IA-64) la supportent.

6)

mémoire de grande capacité, non volatile et qui peut être lue et écrite (exemple:disque dur

7)

La [RFC 4578](#) stipule que « les clients conformes à la spécification PXE doivent faire une requête pour les options DHCP 128 à 135 ».

From:
<https://linuq.org/> - **LinuQ: Logiciels libres à Québec**

Permanent link:
https://linuq.org/logiciels/environnement_serveur_pxe_boot

Last update: **2018/09/28 18:27**

