Serveur DNS

Julien Danjou
jdanjou@linuxenrezo.org

Pour convertir les noms d'ordinateurs en adresses IP, la méthode la plus simple consiste à tenir à jour un fichier hosts contenant les adresses IP suivies des noms de chaque machine composant le réseau.
En voici un exemple:

127.0.0.1 localhost
192.168.1.1 quark
192.168.1.2 mulder

Cette méthode n'est valable que pour un petit réseau comportant un nombre réduit (une dizaine) de machines connectées, gérées par un seul administrateur et ayant aucun contact avec le monde extérieur, c'est-à-dire n'étant pas raccordé directement Internet.

Avec un grand nombre de machine, une solution alternative consiste à utiliser le programme bind pour la résolution des noms en adresses IP.

Avant de créer un serveur de nom, il faut attribuer un nom de domaine à votre réseau. Surtout ne choisissez pas un nom qui existe déjà sur l'Internet, sinon vous risqueriez d'avoir quelques problèmes. J'utilise le nom nunux.fr dans mes exemples.

1 Le fichier /etc/resolv.conf

Avant de configurer le DNS (Domain Name System - Système des noms de domaines), il faut vérifier le contenu du fichier /etc/resolv.conf. Il doit contenir un champ domain suivi du nom de votre domaine (nunux.fr dans mon exemple).

Il doit aussi contenir une ligne commençant par nameserver suivi de l'adresse IP du serveur DNS. Si ce dernier est votre PC personnel, alors l'adresse sera 127.0.0.1.

Voici un exemple de fichier **/etc/resolv.conf**:

domain nunux.fr
nameserver 127.0.0.1

Vous devez aussi installer le package bind qui contient le programme named qui est le serveur de nom.

2 Création du fichier /etc/named.conf

Ce fichier est généralement de très petite taille (quelques octets) et ne contient que des pointeurs vers les fichiers de référence.

Voici ici un exemple de fichier **/etc/named.conf**, que vous pourrez adapter selon votre réseau:

```
        zone 0.0.127.in-addr.arpa
{
type master;
     file /var/named/pz/127.0.0;
};
zone 0.168.192.in-addr.arpa
{
     type master;
     file /var/named/pz/192168.0;
};
zone nunux.fr
{
     type master;
     file /var/named/pz/nunux.fr;
};
```

La première partie sert à la résolution inversée des noms. Cela permet de connaître le nom d'un PC à partir de son adresse IP. La deuxième partie sert à la résolution de noms du domaine nunux.fr.

3 Création des fichiers de zone

Un fichier de zone est composé de RR. RR est l'abréviation de Ressource Record, et est la plus petite donnée du serveur DNS. Chaque RR a un type: A, par exemple, fait correspondre un nom avec une adresse IP.

Voici une liste de quelques-uns des principaux RR utilisés:

SOA: signifie Start Of Authority et signale que l'enregistrement qui suit contient les informations ayant autorité pour ce domaine. Chaque fichier de zone doit contenir un RR de type SOA. Il contient les champs suivants. Le numéro de révision du fichier. On peut aussi le remplacer par la date.
Le temps de rafraîchissement (en secondes) des informations lues à partir du serveur principal (ou primaire).
Le temps (en secondes) pendant lequel le serveur va essayer de joindre le serveur primaire si une requête ou le rafraîchissement échoue.
Le temps (en secondes) au bout duquel un serveur secondaire doit éliminer toutes les informations de zones s'il n'a pas pu contacter un serveur primaire.
Le temps (en secondes aussi :-) pendant lequel les autres serveurs doivent conserver le RR (Ressource Record) dans leur cache.
Le temps peut aussi être écris sous une autre forme: 8H = 8 heures, 1W = 1 semaine, 1D = 1 jour, etc...

A: cet enregistrement associe une adresse IP á un nom de machine. Il ne doit y avoir qu'un seul champ A par machine.

NS: les enregistrements NS servent à spécifier un serveur primaire de zone, et tous ses serveurs secondaires. Le champ de données contient le nom du serveur de noms primaire de cette zone.

CNAME: cet enregistrement associe un alias au nom canonique d'un hôte. Le nom canonique est celui indiqué par l'enregistrement A.

PTR: ce type d'enregistrement est utilisé pour associer les noms dans le domaine in-addr.arpa avec les noms d'hôtes. Il sert à la résolution inversé. Le nom indiqué doit être le nom canonique.

MX: cet enregistrement annonce un échangeur de courrier (Mail eXchanger) pour un domaine. il faut faire précéder le nom de l'échangeur par un nombre entier, qui correspond á la préférence. Un agent de transport de courrier désirant délivrer un message à un domaine (par exemple sendmail) essaiera tous les hôtes qui ont un enregistrement MX en commençant par celui qui a le plus petit numéro de préférence.

HINFO: cet enregistrement donne des informations sur le matériel et le système d'exploitation utilisé. Tout d'abord il faut créer le fichier /var/named/pz/127.0.0 qui contiendra le nom des PC du réseau 127.0.0.0. En fait, ceci sera fait sur n'importe quel réseau, puisque l'adresse IP 127.0.0.1 est celle de l'interface loopback (votre machine). Il est donc fortement recommandée, voir obligatoire :-) d'avoir ce fichier. Il sert à la résolution de nom inverse, ce qui permet de demander le nom d'un PC à partir de son adresse IP.

Voici un fichier **/var/named/pz/127.0.0**:

```
        @               IN      SOA     quark.nunux.fr surfman.nunux.fr (
                                        1
                                        8H
                                        2H
                                        1W
                                        1D)
                        NS      quark.nunux.fr
1               PTR     localhost.
```

L'enregistrement PTR a pour premier champ le numéro de l'hôte, ici 1, ce qui donne 127.0.0.1. Il est suivi du nom correspondant à l'adresse IP, le nom se finissant par un point. Tous les autres types de RR sont indiqués plus haut, lisez leur définition pour en savoir plus.

On a vu dans le fichier /etc/named.conf que les adresses IP des machines de nunux.fr se trouvaient dans **/var/named/pz/nunux.fr**.

Nous allons donc crer ce fichier, dont voici un exemple:

```
        @               IN      SOA     quark.nunux.fr surfman.nunux.fr (
                                        199907241
                                        8H
                                        2H
                                        1W
                                        1D)
                        NS      quark
                        MX      10 mail
localhost       A       127.0.0.1
quark           A       192.168.1.1
mail            CNAME   192.168.1.1
quark           HINFO   Pentium 150 MHz Linux 2.2.9
ns1             CNAME   quark
```

```
www             CNAME   quark
ftp             CNAME   quark
pop             CNAME   quark
mulder          A       192.168.1.2
mulder          HINFO   Celeron 500 MHz Windows 98
quake           CNAME   mulder
```

On voit d'abord que nunux.fr est composé de deux machines:

quark.nunux.fr avec 192.168.1.1 comme adresse
mulder.nunux.fr avec 192.168.1.2 comme adresse

quark.nunux.fr a aussi d'autres nom (CNAME):

mail.nunux.fr
www.nunux.fr
ftp.nunux.fr
pop.nunux.fr

mulder.nunux.fr a un seul autre nom, qui est quake.nunux.fr

Le champ de l'enregistrement MX est mail, ce qui signifie mail.nunux.fr. Si on veut indiquer le nom en entier, il faudra crire mail.nunux.fr.. Notez bien le point la fin !

Il reste un problème: si on veut connaitre le nom d'une machine à partir de son adresse IP. Cela s'appele la recherche de noms inverse: il faut créer un fichier **/var/named/pz/192.168.1**.

En voici un exemple:

```
        @       IN  SOA quark.nunux.fr surfman.nunux.fr (
                1
                8H
                2H
                1W
                1D)
            NS  quark.nunux.fr
1   PTR quark.nunux.fr.
2   PTR mulder.nunux.fr.
```

Pensez à mettre un point après le nom, sinon vous aures des surprises ! Tout est fini, vous pouvez lancez le serveur en tapant ndc start et l'arrêter en tapant ndc stop. Il est bien sur conseiller de le mettre au démarrage :-)

4 Utilisation de nslookup

nslookup est un programme permettant l'interrogation d'un serveur de noms. Il lit les adresses des serveurs de noms dans le fichier /etc/resolv.conf.

Voici un exemple de session avec nslookup:

```
        [root@quark /]# nslookup
Default Server:  localhost
Address:  127.0.0.1

[root@quark /]# nslookup quark.nunux.fr
Server:  localhost
Address:  127.0.0.1
Name:    quark.nunux.fr
Address:  192.168.1.1

[root@quark /]# nslookup www.nunux.fr
Server:  localhost
Address:  127.0.0.1
Name:    quark.nunux.fr
Address:  192.168.1.1
Aliases:  www.nunux.fr

[root@quark /]# nslookup
set q=ns
nunux.fr.
Server:  localhost
Address:  127.0.0.1
nunux.fr nameserver = quark.nunux.fr
quark.nunux.fr internet address = 192.168.1.1
set q=any
mulder.nunux.fr
Server:  localhost
Address:  127.0.0.1
mulder.nunux.fr CPU = Celeron 500 MHz OS = Windows 98
mulder.nunux.fr internet address = 192.168.1.2
nunux.fr nameserver = quark.nunux.fr
quark.nunux.fr internet address = 192.168.1.1
```

5 Serveur DNS pour le rseau local et l'Internet

Mon réseau étant parfois connecté l'Internet (je dis souvent l'Internet, car un internet est un réseau, tout comme l'intranet ou l'extranet) j'en avais marre de changer le fichier resolv.conf pour qu'il utilise mon serveur DNS quand je ne suis pas connecté, et le serveur DNS de mon FAI lorsque je me connecte á l'Internet.

Alors j'ai transformé mon serveur DNS. Je lui ai ajouté une option afin qu'il aille chercher tous les autres domaines sur d'autres serveurs. Il faut rajouter ceci dans votre /etc/named.conf:

```
zone .
{
     type hint;
file /var/named/root.hints;
};
```

Il faut aussi créer le fichier /var/named/root.hints qui contiendra les serveurs de noms de l'Internic. Pour obtenir un tel fichier il faut utiliser le programme dig et lancer:

```
dig @rs.internic.net . ns &gt;/var/named/root.hints
```

Le problème est que lorsque vous demanderez, par exemple, une image dans un fichier HTML sur un serveur qui n'est pas sur le rseau, named va interroger un serveur dont le nom apparat dans le fichier root.hints et cela ne va jamais, s'arrter (il ne vous reste plus qu' killer Netscape quand il se met faire)

Pour éviter cela, il suffit de créer un fichier root.hints vide lorsque l'on n'est pas connecté à l'Internet. A chaque connexion, on lance un script qui remplace temporairement le fichier root.hints vide par un fichier root.hints créé avec la commande dig ci-dessus.

De cette manière, lorsque vous serez déconnecté, le fichier root.hints sera vide et aucun autre serveur ne sera interrogé !

En pratique, créez un fichier root.hints vide et un fichier root.hints.full qui contient le résultat de la commande dig, et vous pourrez ajouter à vos scripts ip-up et ip-down:

```
ip-up
  mv /var/named/root.hints /var/named/root.hints.empty
  mv /var/named/root.hints.full /var/named/root.hints
ip-down
  mv /var/named/root.hints /var/named/root.hints.full
  mv /var/named/root.hints.empty /var/named/root.hints
```