Serveur Linux : DNS

Mise en place d'un service dns sous Linux

Bouron Dimitri

27/10/2013

Ce document sert de démonstration concise pour l'installation, la configuration, d'un serveur dns sous Linux.

Table des matières

I. Machine virtuelle par défaut

Ce document sert de base pour la mise en place d'un service dns, mais aussi la configuration de base faite sur la machine virtuelle complète du serveur.

A. Identifiants de connexion

Les identifiants de connexion de la machine virtuelle par défaut sont les suivants :

Login : root / Password : P@ssword

Login : dimitri / Password : P@ssword

B. Hostname

L'hostname de la machine virtuelle par défaut est : serveur-DNS-base.

C. Interfaces

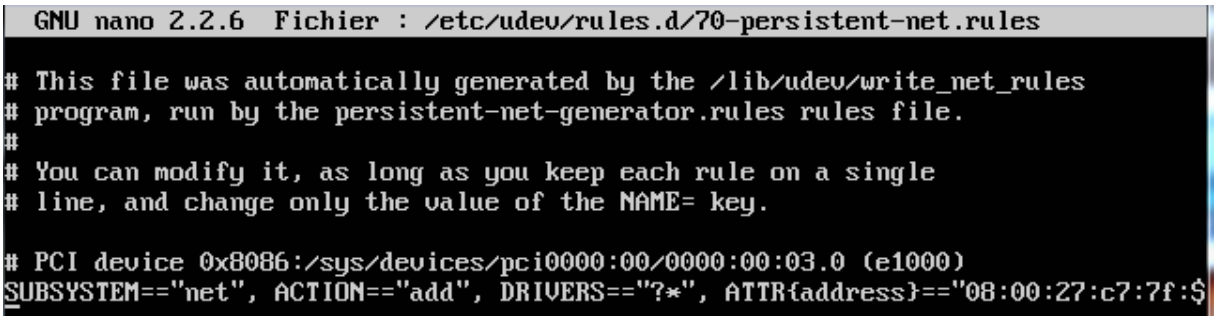
La configuration réseau est la suivante :

Adresse IP statique : 192.168.1.5 /24
Réseau : 192.168.1.0
Adresse de diffusion : 192.168.1.255
Passerelle : 192.168.1.254

Si on crée une nouvelle machine virtuelle à partir de celle-ci, un problème d'interface surviendra. C'est-à-dire que l'interface réseau va s'ajouter et devenir eth1 (si vous utilisez la machine par défaut, dans un autre cas il se peut que vous ayez plusieurs autres interfaces réseau). Le problème c'est que l'on rend la machine accessible uniquement par eth0, donc problème de connexion réseau. Pour y remédier deux solutions, soit on fait ça proprement en supprimant dans un premier temps l'interface eth0 actuelle et on change le nom de l'interface eth1 en eth0, soit on le fait salement en remplaçant la valeur eth0 par eth1 dans le fichier de conf. Voir ci-dessous les deux solutions :

1) Première solution : /etc/udev/rules.d/70-persistent-net.rules

Bon, pour le faire proprement, on va donc faire ce qui est expliqué plus haut, c'est-à-dire supprimer notre eth0 actuelle pour la remplacer par notre eth1. Pour ce faire on va aller dans le fichier avec la commande **# nano /etc/udev/rules.d/70-persistent-net.rules**. Voir ci-dessous le résultat :



Voilà ce que nous avons actuellement, naturellement je ne peux pas montrer les deux interfaces (eth0 et eth1) vu que je n'ai pas déplacé ma machine. Mais ce qui arrivera dans ce cas c'est que les deux dernières lignes visibles sur l'image ci-dessus seront en double exemplaire, ou plutôt 2 autres lignes semblables (les valeurs spécifiques seront différentes) seront visibles. Pour voir le nom de

l'interface il faut aller à la fin de la ligne (qui n'apparaît pas entièrement) qui commence par « SUBSYSTEM=="net" ». Voir ci-dessous :

```
# PCI device 0x8086:/sys/devices/pci0000:00/0000:00:03.0 (e1000)
$c7:7f:b5", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth0"_
```

Comme on peut le voir, pour cette ligne on peut voir à la fin le critère « NAME="eth0" » qui correspond bien au nom de l'interface réseau, si j'avais une seconde ligne ce serait : « NAME="eth1" ». Donc ce qu'il nous reste à faire est tout simplement de supprimer les deux lignes correspondants à eth0, pour aller plus vite il nous suffit de se mettre en début de ligne et de faire la combinaison de touche **[CTRL + K]** (qui correspond à un couper, cela coupera la ligne entièrement), il faut donc faire cette manipulation pour les deux lignes. Une fois qu'il ne reste que deux lignes (celles pour notre eth1), il suffit juste de modifier la valeur de « NAME="eth1" » en remplaçant eth1 par eth0. Une fois que cela est fait, on utilise la commande **# /etc/init.d/networking restart** pour prendre en compte les modifications.

Pour vérifier si oui ou non la manipulation a bien fonctionné, il suffit d'essayer de ping (même une machine locale). Si ça marche tant mieux, sinon il faut essayer un **# reboot** de la machine. Et si ça ne marche toujours pas, il faut essayer de faire en partie la seconde solution (qui sera sale mais fonctionnera peut-être mieux au final).

2) Deuxième solution : /etc/network/interfaces

Comme dit précédemment, cette solution est « sale ». La raison est simple, cela s'explique par la manière de résoudre le problème. Explication : Précédemment, la solution était de supprimer l'actuelle interface eth0 et de la remplacer par l'interface eth1 pour obtenir le même résultat que si nous n'avions pas pris une machine virtuelle déjà existante. Pour notre seconde solution, nous n'allons pas remplacer l'interface eth1 par eth0 mais définir dans notre fichier de configuration réseau que l'interface réseau à utiliser est l'interface eth1 à la place de l'interface eth0 (soit on va remplacer eth0 par eth1). Le fait d'agir ainsi fait que nos interfaces réseaux vont continuer à s'empiler au fur et à mesure qu'on les déplacera ou les réutilisera.

Voilà comment il faut procéder. On utilise la commande **# nano /etc/network/interfaces**, puis on va remplacer la valeur eth0 par eth1, une fois fait et sauvegardé on utilisera la commande **# /etc/init.d/networking restart**. Voir ci-dessous :

```
# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.150
        netmask 255.255.255.0

iface eth0 inet dhcp
```

On remplace eth0 par eth1 pour obtenir :

```
# The primary network interface
auto eth1
iface eth1 inet static
        address 192.168.1.150
        netmask 255.255.255.0

iface eth1_inet dhcp
```

Une fois fait, pour vérifier le bon fonctionnement on peut toujours essayer de ping et la commande **# ifconfig** (je précise que les captures d'écran sont faites à partir d'un serveur web et non du serveur dns, donc c'est normal si les valeurs ne correspondent pas).

D. Accès ssh

La configuration d'openssh-server est faite de telle sorte que :

- Le port d'écoute est 12543.
- Le PermitRootLogin n'est pas autorisé (valeur à no). On ne peut pas se connecter directement par le biais de l'utilisateur root.

E. Bind9

La configuration de bind9 est faite de telle sorte que :

La résolution principale est faite sur le réseau 192.168.1.0 sur le domaine mangetsu.fr.

Le DNS résout les noms suivant :

- serveur-DNS-base = 192.168.1.5
- serveur-DHCP-base = 192.168.1.2
- serveur-WEB-base = 192.168.1.160

Le DNS résout les adresses IP suivantes :

- 192.168.1.5 = serveur-DNS-base.mangetsu.fr
- 192.168.1.2 = serveur-DHCP-base.mangetsu.fr
- 192.168.1.160 = serveur-WEB-base.mangetsu.fr

II. Installation et configuration de base

Une fois l'installation de Ubuntu 12.04 faite, il faut se connecter avec l'identifiant de connexion crée lors de l'installation de l'OS. Dans notre exemple nous avons créé l'utilisateur « dimitri:P@ssword ». Une fois que nous avons réussi à nous connecter nous allons devoir activer l'utilisateur root pour les configurations à venir en utilisant la commande **$ sudo passwd root**. Le mot de passe que nous donnerons à root sera : P@ssword (faites ce que vous voulez). Puis on se connectera avec les identifiants de ce nouvel utilisateur avec la commande **$ su root**. Voir ci-dessous :

```
mangetsu@serveur-WEB-base:~$ sudo passwd root
[sudo] password for mangetsu:
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd: password updated successfully
mangetsu@serveur-WEB-base:~$ su root
Mot de passe :
root@serveur-WEB-base:/home/mangetsu#
```

(L'image ci-dessus a été prise à partir d'un serveur web, donc il est normal que le nom d'utilisateur et le nom d'hôte ne correspondent pas à notre situation).

À partir de maintenant, nous considérerons que toutes les commandes qui suivront sont faites à partir du compte root, même si l'utilisateur simple peut faire la modification pour des raisons de facilités.

A. Apt-get update

On met rapidement la base à jour avec la commande **# apt-get update**.

B. Hostname

L'hostname correspond au nom d'hôte de la machine, que l'on renseigne une première fois lors de l'installation de l'OS. La commande qui permet de visualiser le nom d'hôte est **# hostname**. Pour modifier le nom d'hôte de la machine, on utilise la commande **# nano /etc/hostname**, puis nous modifierons la première ligne en la remplaçant par le nom d'hôte souhaité (attention, il faut éviter de mettre un ou plusieurs underscore, il se peut que cela ne fonctionne pas correctement). Une fois la ligne modifiée et le fichier hostname sauvegardé, nous allons devoir faire relire le fichier hostname par le serveur, le plus bourrin serait d'utiliser **# reboot** pour redémarrer le serveur, mais une solution simple est de forcer la relecture du fichier hostname en utilisant la commande suivante : **# /etc/init.d/hostname restart**. Pour vérifier si la modification a bien été lu après le redémarrage, il suffit de réutiliser la commande **# hostname**. Voir ci-dessous :

```
 GNU nano 2.2.6          Fichier : /etc/hostname                    Modifié

serveur-DNS-base.mangetsu.fr_
```

```
root@serveur-DNS-base:~# hostname
serveur-DNS-base.mangetsu.fr
```

Attention, on peut remarquer que la modification est bien prise en compte actuellement en utilisant la commande pour afficher la valeur affichée dans « **root@serveur-DNS-base:~#** » est toujours la même, elle n'a pas été modifié malgré le changement prit en compte. Toutefois, il suffit de taper la

commande **# exit** pour revenir à l'écran d'identification et nous pouvons constater que la modification apparaît bien désormais. Voir ci-dessous :

```
Ubuntu 12.04 LTS serveur-DNS-base.mangetsu.fr tty1

Hint: Num Lock on

serveur-DNS-base login: _
```

En réalité on peut voir sur la première ligne la modification qui est bien prise en compte, mais on a simplement ajouté un domaine à notre nom d'hôte.

C. Interfaces

Nous utilisons un serveur, dns pour être exacte, ce qui implique de lui attribuer une adresse IP fixe, nous allons donc attribuer l'adresse fixe directement sur notre serveur dns. Pour cela nous allons utiliser la commande **# nano /etc/network/interfaces** et nous modifierons le fichier en remplaçant la ligne « iface eth0 inet dhcp » comme suit :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.5
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
        dns-nameservers 192.168.1.5_
```

Les adresses IP sont à modifier selon la situation naturellement.

Renseigner toutes les valeurs peut rapidement devenir fastidieux. Du coup on peut utiliser cette méthode tout en conservant l'utilisation du DHCP du réseau (encore faut-il en avoir un). Comme je suis un feignant je n'ai pas envie d'avoir à renseigner toutes les valeurs, qui risque d'être faux si je me trompe alors je vais utiliser la seconde méthode qui est de rendre statique les valeurs souhaitées et de faire appel au DHCP pour renseigner le reste automatiquement et donc s'adapter au mieux à ma situation. Voir ci-dessous la configuration à faire :

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
        address 192.168.1.5
        netmask 255.255.255.0

iface eth0 inet dhcp
```

Se contenter de ça ne nous suffira pas, il nous faut faire comme pour l'hostname forcer la reconfiguration réseau avec la commande **# /etc/init.d/networking restart** et utiliser la commande **# ifconfig** pour vérifier que le changement de valeur est pris en compte puis on peut toujours vérifier l'accès internet avec un ping. Voir ci-dessous :

```
root@serveur-DNS-base:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:67:75:10
          inet adr:192.168.1.5  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe67:7510/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:3206 erreurs:0 :0 overruns:0 frame:0
          TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:270084 (270.0 KB) Octets transmis:8886 (8.8 KB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:77 erreurs:0 :0 overruns:0 frame:0
          TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:6140 (6.1 KB) Octets transmis:6140 (6.1 KB)
```

Si on utilise une machine virtuelle qui existait déjà mais que l'on a déplacé d'un poste à l'autre, ou même lorsque l'on en fait une copie directement depuis VirtualBox (ne pas oublier de réinitialiser l'adresse MAC en cochant l'option lors de la copie de la machine). Il se peut qu'il y ait des problèmes de réseau. Certainement qu'il y a une interface ethx de trop et la bonne n'est pas prise en compte. Pour remédier à ce problème il y a deux solutions possibles, une propre et une sale. Ces 2 solutions sont expliquées plus haut dans le document.

D. Mise en place de openssh-server

1) Installation du paquet

Installons notre paquet openssh-server avec la commande **# apt-get install openssh-server**.

```
root@serveur-DNS-base:~# apt-get install openssh-server
```

On peut désormais vérifier le fonctionnement du ssh avec la commande **# service ssh status** et en essayant de se connecter avec putty au serveur.

```
root@serveur-DNS-base:~# service ssh status
ssh start/running, process 2284
```

S'il on voit start/running, on peut essayer putty. Sinon il faut essayer de le lancer avec la commande **# /etc/init.d/ssh start**.

2) Configuration de openssh-server

Nous allons maintenant configurer openssh-server correctement, ou plutôt sécuriser l'accès ssh un minimum pour l'instant en modifiant le port d'écoute ainsi que la restriction de l'utilisation de root pour se connecter en ssh. Pour cela on utilise la commande **# nano /etc/ssh/sshd_config**, puis on va remplacer le numéro de port (22) à la ligne 5 par le port souhaité mais en faisant attention à utiliser un port correcte et non utilisé. Il faut donc un port inférieur à 65535 et éviter les ports utilisés (pour en savoir plus sur les ports utilisés le plus souvent par les autres services et donc les éviter il faut

suivre ce lien). Ce n'est pas tout, on va modifier la valeur du PermitRootLogin à la ligne 27 en remplaçant la valeur actuelle (yes) par la valeur no. Voir ci-dessous :

```
# What ports, IPs and protocols we listen for
Port 12543_
```

```
# Authentication:
LoginGraceTime 120
PermitRootLogin no
StrictModes yes_
```

Une fois fait et sauvegardé, on utilisera la commande **# /etc/init.d/ssh restart** pour forcer la prise en compte des modifications.

Maintenant pour vérifier que cela fonctionne, il suffit d'utiliser putty et d'essayer de se connecter sur le port 22, normalement aucune réponse possible. Ensuite on se connecte au bon port (12543 pour l'exemple), cette fois on va essayer de se connecter en tant que root, si on rentre les bons identifiants le serveur répondra d'un « access denied ». Donc si on est là tout va bien, on va encore se connecter au serveur mais avec l'utilisateur simple cette fois-ci (dimitri pour l'exemple), et normalement on est censé pouvoir se connecter. Bien entendu il est possible ensuite de se connecter au root avec la commande **$ su root**.

E. Mise en place de bind9

1) Installation du paquet

Pour installer le paquet nous utiliserons la commande **# apt-get install bind9**.

```
root@serveur-DNS-base:~# apt-get install bind9
```

2) Configuration de bind9

On va modifier notre fichier /etc/host.conf avec la commande **# nano /etc/host.conf** comme suit :

```
  GNU nano 2.2.6          Fichier : /etc/host.conf

# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

Maintenant on va modifier le fichier /etc/hosts avec la commande **# nano /etc/hosts** comme suit :

```
  GNU nano 2.2.6              Fichier : /etc/hosts

127.0.0.1       localhost.localdomain   localhost
127.0.1.1       serveur-DNS-base.mangetsu.fr    serveur-DNS-base
192.168.1.5     serveur-DNS-base.mangetsu.fr    serveur-DNS-base
```

On va ensuite modifier le fichier /etc/resolv.conf avec la commande **# nano /etc/resolv.conf** comme suit :

```
 GNU nano 2.2.6          Fichier : /etc/resolv.conf

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.5
search mangetsu.fr
```

On va maintenant copier le fichier db.local pour créer 2 nouveaux fichiers que l'on appellera « db.mangetsu.fr » et « db.mangetsu.fr.inv ». avec les 2 commandes suivantes : **# cp /etc/bind/db.local /etc/bind/db.mangetsu.fr** et **# cp /etc/bind/db.local /etc/bind/db.mangetsu.fr.inv**.

```
root@serveur-DNS-base:~# cp /etc/bind/db.local /etc/bind/db.mangetsu.fr
root@serveur-DNS-base:~# cp /etc/bind/db.local /etc/bind/db.mangetsu.fr.inv
```

On va maintenant modifier le premier fichier « db.mangetsu.fr » avec **# nano /etc/bind/db.mangetsu.fr** pour avoir ce qui suit :

```
 GNU nano 2.2.6        Fichier : /etc/bind/db.mangetsu.fr          Modifié

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     serveur-DNS-base. root.serveur-DNS-base. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      serveur-DNS-base.
serveur-DNS-base        A       192.168.1.5
serveur-DHCP-base       A       192.168.1.2
serveur-WEB-base        A       192.168.1.160_
```

Puis on va modifier le second fichier avec la commande **# nano /etc/bind/db.mangetsu.fr.inv** pour avoir ce qui suit :

```
 GNU nano 2.2.6       Fichier : /etc/bind/db.mangetsu.fr.inv

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     serveur-DNS-base. root.serveur-DNS-base. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      serveur-DNS-base.
5       IN      PTR     serveur-DNS-base.mangetsu.fr.
2       IN      PTR     serveur-DHCP-base.mangetsu.fr.
160     IN      PTR     serveur-WEB-base.mangetsu.fr.
```

On continue en créant nos deux zones (la zone principale et la zone inversée), avec la commande **# nano /etc/bind/named.conf.local** et faire comme suit :

```
 GNU nano 2.2.6       Fichier : /etc/bind/named.conf.local

//
// Do any local configuration here
//
zone "mangetsu.fr" {
        type master;
        file "/etc/bind/db.mangetsu.fr";
        forwarders {};
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.mangetsu.fr.inv";
        forwarders {};
};
```

On va maintenant forcer la prise en compte des modifications des fichiers de configuration avec la commande **# /etc/init.d/bind9 restart**.

On peut désormais essayer le bon fonctionnement du service avec quelques tests :

- **# nslookup serveur-DNS-base**
- **# nslookup 192.168.1.5**
- **# dig serveur-DNS-base**
- **# dig –x 192.168.1.5**

```
root@serveur-DNS-base:~# nslookup serveur-DHCP-base
Server:         192.168.1.5
Address:        192.168.1.5#53

Name:   serveur-DHCP-base.mangetsu.fr
Address: 192.168.1.2

root@serveur-DNS-base:~# nslookup 192.168.1.2
Server:         192.168.1.5
Address:        192.168.1.5#53

2.1.168.192.in-addr.arpa        name = serveur-DHCP-base.mangetsu.fr.
```