

Serveur Linux : DHCP

Mise en place d'un service dhcp sous Linux

Bouron Dimitri

23/10/2013

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

Table des matières

I.	Machine virtuelle par défaut.....	2
A.	Identifiants de connexion.....	2
B.	Hostname	2
C.	Interfaces.....	2
1)	Première solution : /etc/udev/rules.d/70-persistent-net.rules.....	2
2)	Deuxième solution : /etc/network/interfaces	3
D.	Accès ssh.....	4
E.	Dhcp3.....	4
II.	Installation et configuration de base.....	5
A.	Apt-get update	5
B.	Hostname	5
C.	Interfaces.....	6
D.	Mise en place de openssh-server.....	7
1)	Installation du paquet	7
2)	Configuration de openssh-server	7
E.	Mise en place de dhcp3.....	8
1)	Installation du paquet	8
2)	Configuration de dhcp3-server.....	8
3)	Attribution d'adresses IP fixes.....	10
4)	Analyse des baux.....	10

I. Machine virtuelle par défaut

Ce document sert de base pour la mise en place d'un service dhcp, 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-DHCP-base.

C. Interfaces

La configuration réseau est la suivante :

Adresse IP statique : 192.168.1.2 /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 :

```
GNU nano 2.2.6 Fichier : /etc/udev/rules.d/70-persistent-net.rules
# This file was automatically generated by the /lib/udev/write_net_rules
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.
# PCI device 0x8086:/sys/devices/pci0000:00/0000:00:03.0 (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?* ", ATTR{address}=="08:00:27:c7:7f:$
```

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

23 octobre 2013

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 dhcp, 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. Dhcp3

La configuration de dhcp3-server est faite de telle sorte que :

Le serveur DHCP assignera aux clients une adresse IP comprise entre 192.168.1.50 et 192.168.1.100 pour une durée de 600 secondes. Le temps d'allocation maximale est de 7200 secondes. Les clients récupéreront les informations suivantes :

- Un masque de sous réseau : 255.255.255.0
- Une adresse de diffusion : 192.168.1.255
- Une adresse de passerelle : 192.168.1.254
- Un serveur DNS : 192.168.1.5
- Un suffixe DNS : mangetsu.fr

On a réservé 4 adresses IP :

- 192.168.1.2 : serveur-DHCP-base
- 192.168.1.5 : serveur-DNS-base
- 192.168.1.150 : serveur-WEB-base
- 192.168.1.155 : serveur-WEB-passif-base
- L'adresse 192.168.1.160 est prise pour l'interface virtuelle Heartbeat liée à nos deux serveurs web. On ne fait pas la réservation mais on ne doit pas la donner à une autre machine.

II. Installation et configuration de base

Une fois l'installation de Ubuntu 12.04 faite, il faut se connecter avec l'identifiant de connexion créé 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 :

```
root@dhcp-ubuntu:~# nano /etc/hostname_
```

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

```
serveur-DHCP-base_
```

```

root@dhcp-ubuntu:~# /etc/init.d/hostname restart
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service hostname restart

Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) and then start(8) utilities,
e.g. stop hostname ; start hostname. The restart(8) utility is also available.
hostname stop/waiting
root@dhcp-ubuntu:~# hostname
serveur-DHCP-base
    
```

Attention, on peut remarquer que la modification est bien prise en compte actuellement en utilisant la commande pour afficher l'hostname mais la valeur affichée dans « **root@dhcp-ubuntu:~#** » est toujours la même, elle n'a pas été modifiée malgré le changement pris 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-DHCP-base tty1

Hint: Num Lock on

serveur-DHCP-base login:
    
```

C. Interfaces

Nous utilisons un serveur, dhcp pour être exacte, ce qui implique de lui attribuer une adresse IP fixe, nous allons donc attribuer l'adresse fixe directement sur notre serveur dhcp. 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 primary network interface
auto eth0
iface eth0 inet static
    address 192.168.1.2
    network 192.168.1.0
    netmask 255.255.255.0
    broadcast 192.168.1.255
    gateway 192.168.1.254
    
```

Les adresses IP sont à modifier selon la situation naturellement.

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. Voir ci-dessous :

```
root@serveur-DHCP-base:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:43:08:c5
          inet adr:192.168.1.2  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe43:8c5/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:0 (0.0 B) Octets transmis:468 (468.0 B)

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:24 erreurs:0 :0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:1792 (1.7 KB) Octets transmis:1792 (1.7 KB)
```

Il faudra au préalable placer la machine virtuelle en réseau interne pour ne pas rentrer en conflit avec le service dhcp déjà présent sur le réseau utilisé. Avant de faire cette modification, j'ai installé le paquet de openssh-server avec la commande **# apt-get install openssh-server**, sinon il ne nous serait plus possible de l'installer en réseau interne.

On aura aussi installé le paquet dhcp3-server avec la commande **# apt-get install dhcp3-server**.

D. Mise en place de openssh-server

1) Installation du paquet

Nous avons préalablement installé ce paquet toute à l'heure, il ne nous reste plus qu'à le configurer.

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 simplement 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**. On est arrivé jusqu'ici sans problèmes, c'est que tout va bien on peut poursuivre, sinon il faut recommencer donc reprendre la sauvegarde précédente et réessayer.

E. Mise en place de dhcp3

1) Installation du paquet

Nous avons préalablement installé ce paquet toute à l'heure, il ne nous reste plus qu'à le configurer.

2) Configuration de dhcp3-server

a) Pour un seul sous-réseau

Nous allons maintenant configurer dhcp3 correctement. Pour commencer nous allons attribuer les adresses. Pour ce faire, nous utilisons la commande **# nano /etc/dhcp/dhcpd.conf**.

```

root@serveur-DHCP-base:~# nano /etc/dhcp/dhcpd.conf _
#Notre configuration :
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.100;
    #Si on souhaite ajouter une autre plage voila ce que l'on aurait :
    #range 192.168.1.101 192.168.1.120

    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.254;
    option domain-name-servers 192.168.1.5;
    option domain-name "mangetsu.fr";
}

```

Avec cette configuration, notre serveur DHCP assignera aux clients une adresse IP comprise entre 192.168.1.50 et 192.168.1.100 pour une durée de 600 secondes. Le client peut spécifier une période de temps spécifique, dans ce cas, le temps d'allocation est de 7200 secondes. Le serveur va également informer le client qu'il doit utiliser :

- Un masque de sous réseau : 255.255.255.0
- Une adresse de diffusion : 192.168.1.255
- Une adresse de passerelle : 192.168.1.254
- Un serveur DNS : 192.168.1.5
- Un suffixe DNS : mangetsu.fr

Ensuite on utilise la commande **# /etc/init.d/isc-dhcp-server restart**.

```
root@serveur-DHCP-base:~# /etc/init.d/isc-dhcp-server restart
```

b) Pour plusieurs sous-réseaux

Ce qui vient d'être fait est bien et fonctionne, mais uniquement si on ne gère qu'un seul sous-réseau « 192.168.1.0/24 », mais dans le cas où on aurait un autre (ou plusieurs autres) sous-réseau(x) alors ce ne serait pas correcte. Il faut que notre DHCP donne une configuration IP adaptée au sous-réseau auquel appartiennent chaque poste qui font appel au DHCP pour leur configuration IP. Un poste appartenant au sous-réseau 172.16.0.0/16 par exemple ne doit en aucun cas avoir la configuration précédemment définie, il lui faut une IP en 172.16.x.x/16 sinon on aura n'importe quoi.

La solution au niveau du DHCP est très simple, il suffit juste de rajouter un subnet dans /etc/dhcp/dhcpd.conf en l'adaptant (l'adresse du DNS peut-être la même car si on autorise les échanges pour les requêtes du DNS entre 192.168.1.0/24 et 172.16.0.0/16 alors il n'est pas nécessaire d'avoir un second DNS. Voilà ci-dessous la modification apportée, je l'ai mis en commentaire pour ne pas en tenir compte vu que je n'ai pas besoin de ça :

```
#Notre configuration :
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.100;
    #Si on souhaite ajouter une autre plage voila ce que l'on aurait :
    #range 192.168.1.101 192.168.1.120

    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.254;
    option domain-name-servers 192.168.1.5;
    option domain-name "mangetsu.fr";
}

#subnet 172.16.0.0 netmask 255.255.0.0 {
#    range 172.16.0.11 172.16.255.250;
#
#    option subnet-mask 255.255.0.0;
#    option broadcast-address 172.16.255.255;
#    option routers 172.16.255.254;
#    option domain-name-servers 192.168.1.5;
#    option domain-name "mangetsu.fr";
#}
```

J'ai bien parlé de la solution au niveau du DHCP ce qui veut dire que ce n'est pas la seule chose à faire, il faut bien comprendre que le principe du DHCP se fait par diffusion (broadcast) et donc les requêtes se limitent par défaut au domaine de diffusion hors si vous avez deux réseaux différents il y a deux domaines de diffusion différents donc il ne sera pas possible de faire des requêtes par diffusion au DHCP depuis 172.16.0.0/16 mais il suffit juste d'utiliser l'agent relais sur le routeur (en l'activant sur l'interface du réseau ne disposant pas de DHCP c'est-à-dire 172.16.0.0/16). Cette documentation est là pour parler uniquement du DHCP donc je n'en expliquerais pas plus mais il suffira d'aller voir sur une documentation sur les routeurs pour voir ce qu'il manque.

3) Attribution d'adresses IP fixes

Pour attribuer des adresses IP fixes, il faut retourner dans `/etc/dhcp/dhcpd.conf` :

```
#Allocation d'adresses IP fixes :
#host server1 {
#    hardware ethernet 00:00:00:00:00:00;
#    fixed-address 192.168.1.x;
#}
```

Pour attribuer une adresse, on modifiera la valeur de `hardware ethernet 00:00:00:00:00:00` par celui du poste voulu et l'adresse IP avec `fixed-address 192.168.1.x` par l'adresse IP voulu.

Je vais préparer quelques réservations, pour notre serveur DHCP, nos serveurs WEB et notre serveur DNS :

```
#Allocation d'adresses IP fixes :
host server-dns {
    hardware ethernet 08:00:27:67:75:10;
    fixed-address 192.168.1.5;
}

host server-dhcp {
    hardware ethernet 08:00:27:43:08:c5;
    fixed-address 192.168.1.2;
}

host server-web-actif {
    hardware ethernet 08:00:27:c7:7f:b5;
    fixed-address 192.168.1.150;
}

host server-web-passif {
    hardware ethernet 08:00:27:81:83:97;
    fixed-address 192.168.1.155;
}
```

Techniquement, si on règle nos machines pour qu'elles prennent toutes les informations réseaux par le DHCP. Alors les machines seront réglées automatiquement.

4) Analyse des baux

Il est possible d'observer les baux des adresses attribuées, mais nous utiliserons une commande qui nous permet d'observer le fichier en continu `# tail -f /var/lib/dhcp/dhcpd.leases`.

```
root@serveur-DHCP-base:~# tail -f /var/lib/dhcp/dhcpd.leases_
```