



SERVICES RESEAUX

Armel KEUPONDJO

PLAN

- GESTION DES HOTES
- LES SERVICES
- LA SECURITE DES SERVICES

LES SERVICES RESEAUX

❑ DHCP

Un **serveur DHCP** permet de fournir automatiquement une configuration IP à une machine, par exemple à des ordinateurs, des smartphones, des imprimantes réseau, en gros tous ceux qui peut être connecté à un réseau. Cette configuration IP est composée :

- ❖ d'une adresse IP
- ❖ d'un masque de réseau
- ❖ d'une passerelle
- ❖ d'une adresse de DNS

LES SERVICES RESEAUX

❑ DHCP

Lorsqu'une machine sans configuration IP se connecte au réseau, celle-ci émet un **DHCP DISCOVER** en broadcast afin de demander si un serveur DHCP existe. Le serveur DHCP répond par un **DHCP OFFER** et commence à donner des premiers paramètres, la machine envoie une demande **DHCP REQUEST** puis le serveur DHCP envoie un **DHCP ACK** afin de fournir la configuration IP.

La configuration IP peut être fournie avec **un bail**. Ce bail indique que tel machine aura tel IP pour une durée de 7 jours par exemple. À la date limite, si la machine est toujours sur le réseau, elle garde la même IP pour un nouveau bail de 7 jours. Si au bout de 2 jours la machine quitte le réseau, l'IP qu'elle utilisait sera disponible pour les autres machines à la fin du bail.

LES SERVICES RESEAUX

❑ DHCP

TP MISE EN PLACE D'UN SERVEUR DHCP SOUS LINUX UBUNTU.

Pour mettre en place le service DHCP dans notre réseau local, nous allons utiliser le paquet *isc-dhcp-server*.

Commençons par l'installer :

```
apt install isc-dhcp-server -y
```

Ensuite, nous devons préciser sur quel interface du serveur, le “démon” (le “service”) va écouter et donc attendre les requêtes des clients.

Modifiez le fichier nécessaire avec la commande suivante :

```
nano /etc/default/isc-dhcp-server
```

LES SERVICES RESEAUX

❑ DHCP

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?  
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".  
INTERFACESv4="ens33"  
#INTERFACESv6=""
```

Nous n'utiliserons pas d'**IPv6**, la ligne peut donc être commentée. En revanche, pour l'interface en **IPv4**, il vous faudra ajouter le nom de l'interface réseau de votre serveur entre les **guillemets**.

Ensuite, il faut éditer le fichier **dhcpd.conf** pour configurer le service DHCP :

nano /etc/dhcp/dhcpd.conf

Chaque ligne de paramètre doit se terminer par le symbole **“;”**. Veillez également à bien **ouvrir/fermer** les blocs de paramètres pour les étendues avec les symboles **“{ }”**.

LES SERVICES RESEAUX

❑ DHCP

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
#option domain-name "example.org";
option domain-name-servers 8.8.8.8;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;
```

LES SERVICES RESEAUX

❑ DHCP

Regardons rapidement les options présentes.

Le premier paramètre est le “**domain-name**”. Comme son nom l’indique, on déclare ici le nom de notre domaine. Nous n’en avons pas, ce paramètre est donc à commenter.

Ensuite nous avons l’option “**domain-name-servers**”. Si les serveurs DNS sont les mêmes pour toutes les étendues, on peut déclarer cette option à partir de ce point. Pour le DNS, j’ai choisi tout bêtement celui de Google, le **8.8.8.8**. Peut importe les étendues, ce sera le même DNS partout, sinon, l’option peut être précisée dans chaque déclaration d’étendue. C’est donc ici une option dite “de serveur” et non pas “**d’étendue**”.

Les paramètres “**default-lease-time**” et “**max-lease-time**” sont les durées des baux pour les adresses attribuées avant une libération ou un renouvellement. Par défaut ces durées sont comprise entre **3600** secondes (1 heure) et **7200** secondes (2 heures).

LES SERVICES RESEAUX

❑ DHCP

Regardons rapidement les options présentes.

L'option “**ddns-update-style**” sert à définir le type de mise à jour du DNS. Nous n'en n'avons pas besoin dans notre contexte. Par défaut, ce paramètre est défini sur “**none**”.

Le paramètre “**authoritative**” peut être décommenté si ce serveur DHCP est le serveur officiel du réseau local, ce qui sera le cas ici.

L'option “**log-facility**” est le niveau de log à conserver. Par défaut, vous trouverez les logs du DHCP dans **/var/log/syslog**. Vous pouvez décommentez cette ligne pour déplacer les logs dans un autre endroit.

LES SERVICES RESEAUX

❑ DHCP

Maintenant, on va déclarer l'étendue du réseau que devra desservir le DHCP.

Ajoutez ces lignes à la fin du fichier :

```
# Etendue LAN
subnet 192.168.3.0 netmask 255.255.255.0 { # Adresse reseau et masque + ouverture bloc
option routers 192.168.3.1;                # Passerelle de l'etendue
range 192.168.3.100 192.168.3.200;        # Plage d'adresses a distribuer
}                                           # Fermeture bloc de l'etendue
```

Remarque: Info + : Sur **isc-dhcp-server**, il n'existe pas de paramètre permettant de créer des plages d'exclusions. Pour palier à ça, il vous suffit de mettre plusieurs lignes de "range" les unes à la suite des autres, toujours sans oublier le ";".

LES SERVICES RESEAUX

❑ DHCP

On redémarre le service pour prendre en compte les modifications :

service isc-dhcp-server restart

Pour voir les baux en cours d'utilisation sur le serveur DHCP, vous pouvez lire le contenu du fichier

/var/lib/dhcp/dhcpd.leases avec “cat”.

Nous avons également à mettre en place une réservation pour un PC bien spécifique. Pour cela, il faut commencer par récupérer son adresse physique (MAC).

LES SERVICES RESEAUX

❑ DHCP

Modifiez de nouveau le fichier **dhcpcd.conf** et dans l'étendue précédemment créée, juste avant de refermer le bloc de paramètres (sur la ligne précédant le symbole **"}**"), ajoutez les lignes suivantes en adaptant l'adresse MAC par celle que vous avez notée.

Ce qui vous donnera finalement une configuration de ce genre :

```
# Etendue LAN
subnet 192.168.3.0 netmask 255.255.255.0 {
option routers 192.168.3.1;
range 192.168.3.100 192.168.3.200;
    #Reservation
    host pc-boss {
        hardware ethernet 00:0c:29:EF:00:B1;
        fixed-address 192.168.3.150;
    }
}
```

Redémarrer le service.

LES SERVICES RESEAUX

❑ DNS

Définition du nom de la machine

La configuration du nom d'un ordinateur n'est pas à proprement parler une opération indispensable, mais elle permet de le nommer de manière plus conviviale qu'en utilisant son adresse IP. La commande de base permettant de manipuler le nom de la machine locale est la commande **hostname**. Appelée telle quelle, elle renvoie le nom actuel de la machine :

hostname

Cette commande permet également de modifier ce nom, simplement avec la syntaxe suivante :

hostname nom

où **nom** est le nom à utiliser. Il est d'usage de n'utiliser que le nom de la machine, sans son domaine. Le nom de domaine est déterminé automatiquement par le système à partir des informations issues de la configuration de la résolution des noms de domaine. Nous verrons cela dans le paragraphe suivant.

LES SERVICES RESEAUX

❑ DNS

Définition du nom de la machine

Cette commande est donc très simple à utiliser, et elle est en général appelée dans les scripts de démarrage du système. La plupart des distributions utilisent le fichier **/etc/HOSTNAME** pour stocker le nom de la machine. Vous êtes bien entendu libre de choisir le nom que vous voulez pour votre ordinateur, mais vous devez vous assurer que ce nom est unique dans votre domaine !

LES SERVICES RESEAUX

❑ DNS

Résolution des noms de domaine

La commande **hostname** ne permet de fixer que le nom de la machine locale. Pour les autres machines du réseau, il faut mettre en place les mécanismes de résolution de noms de domaine. Comme il l'a déjà été indiqué au début de ce chapitre, il existe deux solutions pour trouver l'adresse IP d'une machine à partir de son nom : la consultation d'une liste de noms stockée en local, soit l'interrogation d'un serveur de noms de domaine.

Le fichier **/etc/host.conf** permet de définir le comportement du système lors de la résolution d'un nom. Sa structure est très simple, puisqu'il y a une option de recherche par ligne. Dans la plupart des cas, les lignes suivantes sont suffisantes :

order hosts,bind

multi on

LES SERVICES RESEAUX

❑ DNS

Résolution des noms de domaine

Elles permettent d'indiquer que la recherche des noms pour leur résolution doit se faire d'abord localement, puis par appel aux DNS si la recherche précédente a échoué. C'est en général le comportement désiré. La deuxième ligne permet de faire en sorte que toutes les adresses correspondant à une machine soient renvoyées. Si l'on avait utilisé l'option multi off, seule la première adresse IP trouvée aurait été renvoyée.

La liste de noms locale est stockée dans le fichier `/etc/hosts` (cela explique le nom hosts utilisé pour l'option order dans le fichier `/etc/host.conf`). Votre ordinateur connaîtra directement l'adresse IP de toutes les machines référencées dans ce fichier. Il est bon de placer ici une entrée pour les ordinateurs les plus couramment utilisés sur le réseau.

LES SERVICES RESEAUX

❑ DNS

TP MISE EN PLACE D'UN SERVEUR DNS SOUS LINUX UBUNTU.

Installation

Telecharger bind9 avec la commande suivante :

```
apt-get install bind9
```

Fichier principal

Le fichier principal de configuration de notre serveur est **named.conf** . Il se situe dans le répertoire «/etc/bind/».

Ubuntu fait la répartition de la configuration entre plusieurs fichiers.

Le fichier **named.conf.options** contient l'ensemble des options de configuration du serveur DNS.

Le fichier **named.conf.local** contient la configuration local du serveur DNS, on y déclare les zones associées au domaine.

LES SERVICES RESEAUX

❑ DNS

On va configurer un forwarder, c'est à dire un serveur DNS qui saura répondre aux requêtes que votre serveur ne sait pas résoudre, c'est à dire en gros toutes sauf celles de la zone que l'on va héberger pour nos machines locales.

Pour cela on va modifier **named.conf.options** avec la commande

```
sudo nano /etc/bind/named.conf.options
```

Décommentez le bloc **forwarders** et indiquez par exemple l'ip du serveur dns qui est donné par le serveur dhcp. Vous pouvez aussi mettre l'ip d'un dns de google comme **8.8.8.8**.

```
forwarders {  
    X.X.X.X;  
    8.8.8.8;  
};
```

LES SERVICES RESEAUX

❑ DNS

Pour héberger notre propre zone en tant que serveur maître, on va modifier le fichier **named.conf.local** avec la commande

```
nano /etc/bind/named.conf.local
```

On va déclarer une zone nommée **local.lan**

```
zone "local.lan" IN {  
  
    type master ;  
  
    file "/etc/bind/local.lan";  
  
};
```

LES SERVICES RESEAUX

❑ DNS

Puis il faut créer le fichier `/etc/bind/local.lan`

qui va avec. Pour cela on va copier **db.local** qui sert de référence.

```
sudo cp /etc/bind/db.local /etc/bind/local.lan
```

Editer le fichier avec

```
nano /etc/bind/local.lan
```

LES SERVICES RESEAUX

❑ DNS

```
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA localhost. root.localhost. (
        2      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS localhost.
@ IN A 127.0.0.1
@ IN AAAA ::1
```

```
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA ns.local.lan root.local.lan. (
        2      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS ns.local.lan.
ns IN A 192.168.0.23 ← ip du serveur
```

LES SERVICES RESEAUX

❑ DNS

Pour éviter que vos tests soient faussés car ils passeront par le dns de l'ipv6, modifiez

/etc/bind/named.conf.options

nano /etc/bind/named.conf.options

```
GNU nano 3.2 /etc/bind/named.conf.options

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; }; → listen-on-v6 { ::1; };
};
```

LES SERVICES RESEAUX

❑ DNS

Vérification de la configuration

Avant de lancer le service et de commencer à utiliser notre serveur DNS, il convient de s'assurer que l'on n'a rien oublié et que le paramétrage ne contient aucune erreur :

named-checkconf

Si tout se passe bien, on ne devrait pas avoir de message d'erreur. Sinon, il faut scruter **les journaux de logs** et voir ce qui ne va pas.

Ensuite, on peut passer à la vérification de la configuration, zone par zone, en exécutant les commandes suivantes :

named-checkzone local.lan /etc/bind/local.lan

LES SERVICES RESEAUX

❑ DNS

Vérification de la configuration

Lorsqu'il y a une erreur, généralement le texte en est suffisamment explicite pour corriger et relancer les commandes de vérifications, jusqu'à ce qu'il n'y ait plus aucun problème. On peut alors démarrer le service DNS et le rendre automatique lors des redémarrages du système :

```
service named start
```

```
chkconfig named on
```


LES SERVICES RESEAUX

❑ Serveur Web

Installez Apache

Apache est disponible dans les référentiels logiciels par défaut d'Ubuntu. Pour cette raison, nous allons commencer par mettre à jour l'index local des paquets pour avoir les dernières modifications:

```
sudo apt update
```

Maintenant nous pouvons installer le package apache2:

```
sudo apt install apache2
```

Après l'installation, nous pouvons vérifier quelle version d'Apache nous installons en tapant dans le même terminal:

```
sudo apache2ctl -v
```

LES SERVICES RESEAUX

❑ Serveur Web

Vérifiez le serveur Web

À la fin du processus d'installation, Ubuntu démarre Apache, donc le serveur Web devrait déjà être opérationnel. Nous pouvons le vérifier en écrivant:

```
sudo systemctl status apache2
```

La commande ci-dessus doit indiquer que le service a démarré avec succès. Cependant, la meilleure façon de tester cela est de demander une page à Apache. Nous pouvons y accéder via l'adresse IP pour confirmer que le logiciel fonctionne correctement.

Par défaut, Apache est configuré pour démarrer automatiquement avec l'ordinateur. Nous pouvons désactiver ceci dactylographie:

```
sudo systemctl disable apache2
```

pour les réactiver le service pour qu'il démarre au démarrage: *sudo systemctl enable apache2*

LES SERVICES RESEAUX

❑ Serveur Web

Fichiers et répertoires importants Apache

- ✓ `/Var/www/html` → Comprend le contenu Web. Cela peut être modifié dans les fichiers de configuration Apache.
- ✓ `/ etc /apache2` → Tout Fichiers de configuration Apache résider ici.
- ✓ `/etc/apache2/apache2.conf` → Il s'agit de Fichier de configuration principal d'Apache.
- ✓ `/etc/apache2/ports.conf` → Ce fichier spécifie les ports sur lesquels Apache écoutera.
- ✓ `/etc/apache2/sites-available` → Le répertoire où les hôtes virtuels peuvent être stockés par site. Apache n'utilisera pas les fichiers de configuration trouvés dans ce répertoire à moins qu'ils ne soient liés au répertoire activé pour le site. En règle générale, tous les paramètres de verrouillage du serveur sont définis dans ce répertoire.

LES SERVICES RESEAUX

❑ Serveur Web

Fichiers et répertoires importants Apache

- ✓ `/etc/apache2/sites-enabled/` → Le répertoire dans lequel les hôtes virtuels activés pour le site sont stockés. Ceux-ci sont généralement créés en établissant un lien vers les fichiers de configuration trouvés dans le répertoire des sites disponibles avec `a2ensite`. Apache lit les fichiers de configuration et les liens dans ce répertoire au démarrage ou se recharge pour compiler une configuration complète.
- ✓ `/etc/apache2/conf-available/` et `/etc/apache2/conf-enabled/` → Ces répertoires ont la même relation que les répertoires sites disponibles et sites activés, mais sont utilisés pour stocker des fragments de configuration qui n'appartiennent pas à un hôte virtuel.
- ✓ `/etc/apache2/mods-available/` et `/etc/apache2/mods-enabled/` → Ces répertoires contiennent les modules disponibles et activés Respectivement.

LES SERVICES RESEAUX

❑ Serveur Web

Les diverses configurations sont activées avec les commandes suivantes :

sudo a2ensite [configuration d'un site à activer]

sudo a2dissite [configuration d'un site à désactiver]

sudo a2enconf [configuration d'un service à activer]

sudo a2disconf [configuration d'un service à désactiver]

sudo a2enmod [configuration d'un module à activer]

sudo a2dismod [configuration d'un module à désactiver]

LES SERVICES RESEAUX

❑ Serveur Web

Cela aura pour effet de créer ou supprimer les liens symboliques correspondants dans les répertoires **xxx-enabled**.

Apache prendra alors en compte, ou pas, les fichiers de configuration concernés après rechargement :

```
sudo systemctl reload apache2
```