Paramètres de dnsmasq.conf

Cf Page de man en français : Dnsmasq - Un serveur DHCP et cache DNS poids-plume

- Dnsmasq - Un serveur DHCP et cache DNS poids-plume

Format

Une option par ligne.

Les options valides sont les mêmes que les options légales de la ligne de commande.

Détails sur

```
/usr/sbin/dnsmasq --help
```

ou

```
man 8 dnsmasq
```

Options

domain-needed
bogus-priv
: On évite de faire sortir les requêtes de domaines locaux!!

domain=mondomaine.local
: Domaine local qui sera ajouté aux noms DNS des machines assignées par le DHCP et fourni en nom de domaine local aux clients DHCP!!

expand-hosts
: Cette directive permet d'ajouter le domaine défini ci-dessus aux noms simples figurant dans /etc/hosts!!

resolv-file=/etc/resolv.dnsmasq
: Fichier à lire pour les serveurs DNS par défaut (au lieu de /etc/resolv.conf, utilisé par défaut)!!

On peut forcer la transmission des requêtes relatives à un domaine particulier vers un serveur précis :

server=/mondomaine.local/192.168.1.1
server=/1.168.192.in-addr.arpa/192.168.1.1
: ici 192.168.1.1!!

user=dnsmasq
group=nogroup
interface=eth0
: Le daemon doit fonctionner sous l'utilisateur dnsmasq et sur l'interface eth0!!

dhcp-range=eth0,192.168.1.100,192.168.1.199,12h
: plage d'adresses dynamiques fournies par le serveur DHCP

ici de 192.168.1.100 à 192.168.1.199 avec des baux de 12h!!

dhcp-option=option:router,192.168.1.1
: passerelle du réseau à fournir aux clients DHCP!!

dhcp-host=11:22:33:44:55:66,192.168.0.60
: Exemple pour assigner l'IP 192.168.0.60 à la MAC 11:22:33:44:55:66!!
dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m
: assigne également le nom «fred», pour 45 minutes!!
dhcp-host=00:50:7F:E9:A7:45,passerelle,192.168.1.1,12h
dhcp-host=00:21:B7:30:53:44,lexmark,192.168.1.6,12h
dhcp-host=00:1B:38:32:EA:31,frodo,192.168.1.11,12h
: adresses DHCP statiques

Les noms précisés seront servis par le service DNS de dnsmasq!!

cname=frodon,frodo
: On peut définir des alias pour les noms de machines définies par le DHCP

(exemple pour pouvoir utiliser le nom «frodon» tout comme «frodo»)!!

dhcp-authoritative
: Désactiver cette directive uniquement si votre serveur est le serveurs DHCP officiel du réseau!!

log-queries
: logger les requêtes DNS (dans syslog)!!
log-dhcp
: logs plus complets des requêtes DHCP (dans syslog)!!

address=/<domaine>/[domaine/]<adresse IP>
: adresse IP à retourner pour toute requête pour les domaines fournis en option.

Les requêtes pour ce(s) domaine(s) ne sont jamais transmises aux serveurs amont et reçoivent comme réponse l'adresse IP spécifiée qui peut être une adresse IPv4 ou IPv6. Pour donner à la fois une adresse IPv4 et une adresse IPv6 pour un domaine, utiliser plusieurs options address.
Il faut noter que le contenu du fichier /etc/hosts et de celui des baux DHCP supplante ceci pour des noms individuels.
Une utilisation courante de cette option est de rediriger la totalité du domaine doubleclick.net vers un serveur web local afin d'éviter les bannières publicitaires.
La spécification de domaine fonctionne de la même façon que server, avec la caractéristique

supplémentaire que **/#/** coïncide avec tout domaine.!!

address=/#/1.2.3.4
: retournera 1.2.3.4 pour toute requête n'ayant de réponse ni dans **/etc/hosts**, ni dans les baux DHCP, et n'étant pas transmise à un serveur spécifique par le biais d'une directive **server**!!

Fichier type

/etc/dnsmasq.d/local.conf

```
# On évite de faire sortir les requêtes de domaines locaux
domain-needed
bogus-priv

# Domaine local qui sera ajouté aux noms DNS des machines
assignées par le DHCP,
# et fourni en nom de domaine local aux clients DHCP
domain=mondomaine.local

# Cette directive permet d'ajouter le domaine défini ci-dessus aux
noms simples figurant dans /etc/hosts
expand-hosts

# Fichier à lire pour les serveurs DNS par défaut (au lieu de
/etc/resolv.conf, utilisé par défaut)
resolv-file=/etc/resolv.dnsmasq

# On peut forcer la transmission des requêtes relatives à un
domaine particulier
# vers un serveur précis, ici 192.168.1.1 :
server=/mondomaine.local/192.168.1.1
server=/1.168.192.in-addr.arpa/192.168.1.1

# Le daemon doit fonctionner sous l'utilisateur dnsmasq et sur
l'interface eth0
user=dnsmasq
group=nogroup
interface=eth0

# La plage d'adresses dynamiques fournies par le serveur DHCP
# C'est ici de 192.168.1.100 à 192.168.1.199 avec des leases de
12h
dhcp-range=eth0,192.168.1.100,192.168.1.199,12h

# La passerelle du réseau à fournir aux clients DHCP
dhcp-option=option:router,192.168.1.1

# Exemple pour assigner l'IP 192.168.0.60 à la MAC
```

```
11:22:33:44:55:66
# La seconde forme assigne également le nom « fred », pour 45
minutes
#dhcp-host=11:22:33:44:55:66,192.168.0.60
#dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m

# Nos adresses DHCP statiques
# Les noms précisés seront servis par le service DNS de dnsmasq
dhcp-host=00:50:7F:E9:A7:45,passerelle,192.168.1.1,12h
dhcp-host=00:21:B7:30:53:44,lexmark,192.168.1.6,12h
dhcp-host=00:1B:38:32:EA:31,frodo,192.168.1.11,12h

# On peut définir des alias pour les noms de machines définies par
le DHCP
# Exemple: pour pouvoir utiliser le nom « frodon » tout comme «
frodo », utilisez cette ligne:
#cname=frodon,frodo

# Désactiver cette directive uniquement si votre serveur est le
serveurs DHCP officiel du réseau
#dhcp-authoritative

# Pour logger les requêtes DNS (dans syslog)
log-queries
# Pour des logs plus complets des requêtes DHCP (dans syslog)
log-dhcp
```

Mise en place du cache DNS

- On évite de faire sortir les requêtes de domaines locaux

domain-needed
: Ne transmet pas les requêtes dont le nom de domaine n'est pas complet.
Par exemple une requête pour **machine** ne sera pas transmise aux serveurs DNS de votre FAI, alors qu'une requête pour **machine.domain.com** le sera.

bogus-priv
: Fausse résolution inverse pour les réseaux privés.
Toutes les requêtes DNS inverses pour des adresses IP privées (ie 192.168.x.x, etc…) qui ne sont pas trouvées dans /etc/hosts ou dans le fichier de baux DHCP se voient retourner une réponse "no such domain" [1] au lieu d'être transmises aux serveurs de nom amont ("upstream server")

- Domaine local qui sera ajouté aux noms DNS des machines assignées par le DHCP et fourni en nom de domaine local aux clients DHCP

domain=mondomaine.local
: définit le nom de domaine du réseau local. [2]

/etc/dnsmasq.d/local.conf

```
domain-needed
expand-hosts
bogus-priv

interface=eth0
domain=home.chezmoi.invalid
cache-size=256
```

bogus-priv
: Fausse résolution inverse pour les réseaux privés.
Toutes les requêtes DNS inverses pour des adresses IP privées (ie 192.168.x.x, etc...) qui ne sont pas trouvées dans /etc/hosts ou dans le fichier de baux DHCP se voient retourner une réponse “no such domain” [3] au lieu d’être transmises aux serveurs de nom amont (“upstream server”).

domain-needed
: Ne transmet pas les requêtes dont le nom de domaine n'est pas complet. Par exemple une requête pour **machine** ne sera pas transmise aux serveurs DNS de votre FAI, alors qu’une requête pour **machine.domain.com** le sera.

expand-hosts
: Ajoute automatiquement le domaine défini dans la configuration aux noms de machines présents dans **/etc/hosts**.

domain=home.chezmoi.invalid
: définit le nom de domaine du réseau local. [4]

interface=eth0
: Dans le cas d’un réseau local, DnsMasq n'est en écoute que sur l’interface locale et non sur le lien relié au Net.

cache-size=256
: Précise la taille du cache qui contient les dernières requêtes ainsi que leurs réponses. [5]

DHCP

/etc/dnsmasq.d/local.conf

```
domain-needed
expand-hosts
bogus-priv

interface=eth0
domain=home.chezmoi.invalid
cache-size=256

# plage dynamique 192.168.1.10-192.168.1.100
dhcp-range=192.168.1.10,192.168.1.100,24h
dhcp-option=23,64

# adresse IP fixe pour la machine 00:40:F4:6C:43:AF
```

```
dhcp-host=00:40:F4:6C:43:AF,machinefixe,192.168.1.10
```

dhcp-range=débutPlage,finPlage,duréeBail
: L'option la plus importante qui permet de définir l'espace IP dans lequel DnsMasq va fournir des adresses aux ordinateurs de votre réseau local.

dhcp-option=option,valeur
: Permet de définir les options envoyées aux clients avec l'adresse IP affectée. Les options suivantes sont déterminées automatiquement à partir de la configuration de la machine qui fait tourner DnsMasq :

1
: masque de sous-réseau

3
: passerelle par défaut

6
: serveur DNS

28
: adresse de broadcast Par exemple, l'option 23 utilisée ici, permet de définir le TTL envoyé aux clients.

dhcp-host=adresseMac,nomMachine,IP
: L'option dhcp-host permet d'envoyer des options ou une adresse IP fixe à un ordinateur particulier. Celui-ci peut être identifié à l'aide son adresse MAC ou bien par le nom envoyé comme identifiant.

DnsMasq et Resolvconf

Resolvconf est un programme présent dans la distibution GNU/Linux Debian qui facilite la maintenance du fichier "/etc/resolv.conf". Il agit comme une interface entre les programmes qui définissent l'adresse des serveurs de noms (client DCHP, démon PPP…) et les programmes qui utilisent ces informations (résolveur de noms, cache DNS…).

Ce programme évite l'édition directe du fichier "/etc/resolv.conf" par les programmes en question : les programmes qui ont à ajouter un serveur de nom fournissent son adresse à Resolvconf qui, selon la configuration de l'utilisateur, choisit de l'inclure ou non dans le fichier "/etc/resolv.conf".

Fonctionnement avec DnsMasq

Resolvconf offre un avantage particulier avec DnsMasq : il permet de ne fournir à DnsMasq que les DNS externes. Resolvconf va générer un fichier spécifique ("/var/run/dnsmasq/resolv.conf") qui contient l'ensemble des adresses des serveurs de noms sauf celle de DnsMasq. D'autre part, à chaque changement de DNS externes, l'information sera automatiquement mise à jour dans le fichier en question. DnsMasq va utiliser automatiquement ce fichier à la place du classique "/etc/resolv.conf" Pour installer Resolvconf :

```
sudo apt-get install resolvconf
```

Le fichier /etc/dnsmasq.conf

Il est entièrement mis en commentaires. Les modifications se feront de préférence dans le fichier **/etc/dnsmasq.d/local.conf**.

/etc/dnsmasq.conf

```
# Configuration file for dnsmasq.
#
# Format is one option per line, legal options are the same
# as the long options legal on the command line. See
# "/usr/sbin/dnsmasq --help" or "man 8 dnsmasq" for details.

# The following two options make you a better netizen, since
they
# tell dnsmasq to filter out queries which the public DNS
cannot
# answer, and which load the servers (especially the root
servers)
# unnecessarily. If you have a dial-on-demand link they also
stop
# these requests from bringing up the link unnecessarily.

# Never forward plain names (without a dot or domain part)
#domain-needed
# Never forward addresses in the non-routed address spaces.
#bogus-priv


# Uncomment this to filter useless windows-originated DNS
requests
# which can trigger dial-on-demand links needlessly.
# Note that (amongst other things) this blocks all SRV
requests,
# so don't use it if you use eg Kerberos, SIP, XMMP or Google-
talk.
# This option only affects forwarding, SRV records originating
for
# dnsmasq (via srv-host= lines) are not suppressed by it.
#filterwin2k

# Change this line if you want dns to get its upstream servers
from
# somewhere other that /etc/resolv.conf
#resolv-file=

# By  default,  dnsmasq  will  send queries to any of the
upstream
# servers it knows about and tries to favour servers to are
known
```

```
# to  be  up.  Uncommenting this forces dnsmasq to try each
query
# with  each  server  strictly  in  the  order  they   appear
in
# /etc/resolv.conf
#strict-order

# If you don't want dnsmasq to read /etc/resolv.conf or any
other
# file, getting its servers from this file instead (see
below), then
# uncomment this.
#no-resolv

# If you don't want dnsmasq to poll /etc/resolv.conf or other
resolv
# files for changes and re-read them then uncomment this.
#no-poll

# Add other name servers here, with domain specs if they are
for
# non-public domains.
#server=/localnet/192.168.0.1

# Example of routing PTR queries to nameservers: this will
send all
# address->name queries for 192.168.3/24 to nameserver
10.1.2.3
#server=/3.168.192.in-addr.arpa/10.1.2.3

# Add local-only domains here, queries in these domains are
answered
# from /etc/hosts or DHCP only.
#local=/localnet/

# Add domains which you want to force to an IP address here.
# The example below send any host in double-click.net to a
local
# web-server.
#address=/double-click.net/127.0.0.1

# --address (and --server) work with IPv6 addresses too.
#address=/www.thekelleys.org.uk/fe80::20d:60ff:fe36:f83

# You can control how dnsmasq talks to a server: this forces
# queries to 10.1.2.3 to be routed via eth1
# server=10.1.2.3@eth1

# and this sets the source (ie local) address used to talk to
# 10.1.2.3 to 192.168.1.1 port 55 (there must be a interface
```

```
with that
# IP on the machine, obviously).
# server=10.1.2.3@192.168.1.1#55

# If you want dnsmasq to change uid and gid to something other
# than the default, edit the following lines.
#user=
#group=

# If you want dnsmasq to listen for DHCP and DNS requests only
on
# specified interfaces (and the loopback) give the name of the
# interface (eg eth0) here.
# Repeat the line for more than one interface.
#interface=
# Or you can specify which interface _not_ to listen on
#except-interface=
# Or which to listen on by address (remember to include
127.0.0.1 if
# you use this.)
#listen-address=
# If you want dnsmasq to provide only DNS service on an
interface,
# configure it as shown above, and then use the following line
to
# disable DHCP and TFTP on it.
#no-dhcp-interface=

# On systems which support it, dnsmasq binds the wildcard
address,
# even when it is listening on only some interfaces. It then
discards
# requests that it shouldn't reply to. This has the advantage
of
# working even when interfaces come and go and change address.
If you
# want dnsmasq to really bind only the interfaces it is
listening on,
# uncomment this option. About the only time you may need this
is when
# running another nameserver on the same machine.
#bind-interfaces

# If you don't want dnsmasq to read /etc/hosts, uncomment the
# following line.
#no-hosts
# or if you want it to read another file, as well as
/etc/hosts, use
# this.
#addn-hosts=/etc/banner_add_hosts
```

```
# Set this (and domain: see below) if you want to have a
domain
# automatically added to simple names in a hosts-file.
#expand-hosts

# Set the domain for dnsmasq. this is optional, but if it is
set, it
# does the following things.
# 1) Allows DHCP hosts to have fully qualified domain names,
as long
#     as the domain part matches this setting.
# 2) Sets the "domain" DHCP option thereby potentially setting
the
#    domain of all systems configured by DHCP
# 3) Provides the domain part for "expand-hosts"
#domain=thekelleys.org.uk

# Set a different domain for a particular subnet
#domain=wireless.thekelleys.org.uk,192.168.2.0/24

# Same idea, but range rather then subnet
#domain=reserved.thekelleys.org.uk,192.68.3.100,192.168.3.200

# Uncomment this to enable the integrated DHCP server, you
need
# to supply the range of addresses available for lease and
optionally
# a lease time. If you have more than one network, you will
need to
# repeat this for each network on which you want to supply
DHCP
# service.
#dhcp-range=192.168.0.50,192.168.0.150,12h

# This is an example of a DHCP range where the netmask is
given. This
# is needed for networks we reach the dnsmasq DHCP server via
a relay
# agent. If you don't know what a DHCP relay agent is, you
probably
# don't need to worry about this.
#dhcp-range=192.168.0.50,192.168.0.150,255.255.255.0,12h

# This is an example of a DHCP range which sets a tag, so that
# some DHCP options may be set only for this network.
#dhcp-range=set:red,192.168.0.50,192.168.0.150

# Use this DHCP range only when the tag "green" is set.
#dhcp-range=tag:green,192.168.0.50,192.168.0.150,12h
```

```
# Specify a subnet which can't be used for dynamic address
allocation,
# is available for hosts with matching --dhcp-host lines. Note
that
# dhcp-host declarations will be ignored unless there is a
dhcp-range
# of some type for the subnet in question.
# In this case the netmask is implied (it comes from the
network
# configuration on the machine running dnsmasq) it is possible
to give
# an explicit netmask instead.
#dhcp-range=192.168.0.0,static

# Supply parameters for specified hosts using DHCP. There are
lots
# of valid alternatives, so we will give examples of each.
Note that
# IP addresses DO NOT have to be in the range given above,
they just
# need to be on the same network. The order of the parameters
in these
# do not matter, it's permissible to give name, address and
MAC in any
# order.

# Always allocate the host with Ethernet address
11:22:33:44:55:66
# The IP address 192.168.0.60
#dhcp-host=11:22:33:44:55:66,192.168.0.60

# Always set the name of the host with hardware address
# 11:22:33:44:55:66 to be "fred"
#dhcp-host=11:22:33:44:55:66,fred

# Always give the host with Ethernet address 11:22:33:44:55:66
# the name fred and IP address 192.168.0.60 and lease time 45
minutes
#dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m

# Give a host with Ethernet address 11:22:33:44:55:66 or
# 12:34:56:78:90:12 the IP address 192.168.0.60. Dnsmasq will
assume
# that these two Ethernet interfaces will never be in use at
the same
# time, and give the IP address to the second, even if it is
already
# in use by the first. Useful for laptops with wired and
wireless
# addresses.
#dhcp-host=11:22:33:44:55:66,12:34:56:78:90:12,192.168.0.60
```

```
# Give the machine which says its name is "bert" IP address
# 192.168.0.70 and an infinite lease
#dhcp-host=bert,192.168.0.70,infinite

# Always give the host with client identifier 01:02:02:04
# the IP address 192.168.0.60
#dhcp-host=id:01:02:02:04,192.168.0.60

# Always give the host with client identifier "marjorie"
# the IP address 192.168.0.60
#dhcp-host=id:marjorie,192.168.0.60

# Enable the address given for "judge" in /etc/hosts
# to be given to a machine presenting the name "judge" when
# it asks for a DHCP lease.
#dhcp-host=judge

# Never offer DHCP service to a machine whose Ethernet
# address is 11:22:33:44:55:66
#dhcp-host=11:22:33:44:55:66,ignore

# Ignore any client-id presented by the machine with Ethernet
# address 11:22:33:44:55:66. This is useful to prevent a
machine
# being treated differently when running under different OS's
or
# between PXE boot and OS boot.
#dhcp-host=11:22:33:44:55:66,id:*

# Send extra options which are tagged as "red" to
# the machine with Ethernet address 11:22:33:44:55:66
#dhcp-host=11:22:33:44:55:66,set:red

# Send extra options which are tagged as "red" to
# any machine with Ethernet address starting 11:22:33:
#dhcp-host=11:22:33:*:*:*,set:red

# Ignore any clients which are specified in dhcp-host lines
# or /etc/ethers. Equivalent to ISC "deny unknown-clients".
# This relies on the special "known" tag which is set when
# a host is matched.
#dhcp-ignore=tag:!known

# Send extra options which are tagged as "red" to any machine
whose
# DHCP vendorclass string includes the substring "Linux"
#dhcp-vendorclass=set:red,Linux

# Send extra options which are tagged as "red" to any machine
one
```

```
# of whose DHCP userclass strings includes the substring
"accounts"
#dhcp-userclass=set:red,accounts

# Send extra options which are tagged as "red" to any machine
whose
# MAC address matches the pattern.
#dhcp-mac=set:red,00:60:8C:*:*:*

# If this line is uncommented, dnsmasq will read /etc/ethers
and act
# on the ethernet-address/IP pairs found there just as if they
had
# been given as --dhcp-host options. Useful if you keep
# MAC-address/host mappings there for other purposes.
#read-ethers

# Send options to hosts which ask for a DHCP lease.
# See RFC 2132 for details of available options.
# Common options can be given to dnsmasq by name:
# run "dnsmasq --help dhcp" to get a list.
# Note that all the common settings, such as netmask and
# broadcast address, DNS server and default route, are given
# sane defaults by dnsmasq. You very likely will not need
# any dhcp-options. If you use Windows clients and Samba,
there
# are some options which are recommended, they are detailed at
the
# end of this section.

# Override the default route supplied by dnsmasq, which
assumes the
# router is the same machine as the one running dnsmasq.
#dhcp-option=3,1.2.3.4

# Do the same thing, but using the option name
#dhcp-option=option:router,1.2.3.4

# Override the default route supplied by dnsmasq and send no
default
# route at all. Note that this only works for the options sent
by
# default (1, 3, 6, 12, 28) the same line will send a zero-
length option
# for all other option numbers.
#dhcp-option=3

# Set the NTP time server addresses to 192.168.0.4 and
10.10.0.5
#dhcp-option=option:ntp-server,192.168.0.4,10.10.0.5
```

```
# Set the NTP time server address to be the same machine as
# is running dnsmasq
#dhcp-option=42,0.0.0.0

# Set the NIS domain name to "welly"
#dhcp-option=40,welly

# Set the default time-to-live to 50
#dhcp-option=23,50

# Set the "all subnets are local" flag
#dhcp-option=27,1

# Send the etherboot magic flag and then etherboot options (a
string).
#dhcp-option=128,e4:45:74:68:00:00
#dhcp-option=129,NIC=eepro100

# Specify an option which will only be sent to the "red"
network
# (see dhcp-range for the declaration of the "red" network)
# Note that the tag: part must precede the option: part.
#dhcp-option = tag:red, option:ntp-server, 192.168.1.1

# The following DHCP options set up dnsmasq in the same way as
is specified
# for the ISC dhcpcd in
#
http://www.samba.org/samba/ftp/docs/textdocs/DHCP-Server-Confi
guration.txt
# adapted for a typical dnsmasq installation where the host
running
# dnsmasq is also the host running samba.
# you may want to uncomment some or all of them if you use
# Windows clients and Samba.
#dhcp-option=19,0           # option ip-forwarding off
#dhcp-option=44,0.0.0.0     # set netbios-over-TCP/IP
nameserver(s) aka WINS server(s)
#dhcp-option=45,0.0.0.0     # netbios datagram distribution
server
#dhcp-option=46,8           # netbios node type

# Send RFC-3397 DNS domain search DHCP option. WARNING: Your
DHCP client
# probably doesn't support this......
#dhcp-option=option:domain-
search,eng.apple.com,marketing.apple.com

# Send RFC-3442 classless static routes (note the netmask
encoding)
```

```
#dhcp-option=121,192.168.1.0/24,1.2.3.4,10.0.0.0/8,5.6.7.8

# Send vendor-class specific options encapsulated in DHCP
option 43.
# The meaning of the options is defined by the vendor-class so
# options are sent only when the client supplied vendor class
# matches the class given here. (A substring match is OK, so
"MSFT"
# matches "MSFT" and "MSFT 5.0"). This example sets the
# mtftp address to 0.0.0.0 for PXEClients.
#dhcp-option=vendor:PXEClient,1,0.0.0.0

# Send microsoft-specific option to tell windows to release
the DHCP lease
# when it shuts down. Note the "i" flag, to tell dnsmasq to
send the
# value as a four-byte integer - that's what microsoft wants.
See
#
http://technet2.microsoft.com/WindowsServer/en/library/a70f1bb
7-d2d4-49f0-96d6-4b7414ecfaae1033.mspx?mfr=true
#dhcp-option=vendor:MSFT,2,1i

# Send the Encapsulated-vendor-class ID needed by some
configurations of
# Etherboot to allow is to recognise the DHCP server.
#dhcp-option=vendor:Etherboot,60,"Etherboot"

# Send options to PXELinux. Note that we need to send the
options even
# though they don't appear in the parameter request list, so
we need
# to use dhcp-option-force here.
# See http://syslinux.zytor.com/pxe.php#special for details.
# Magic number - needed before anything else is recognised
#dhcp-option-force=208,f1:00:74:7e
# Configuration file name
#dhcp-option-force=209,configs/common
# Path prefix
#dhcp-option-force=210,/tftpboot/pxelinux/files/
# Reboot time. (Note 'i' to send 32-bit value)
#dhcp-option-force=211,30i

# Set the boot filename for netboot/PXE. You will only need
# this is you want to boot machines over the network and you
will need
# a TFTP server; either dnsmasq's built in TFTP server or an
# external one. (See below for how to enable the TFTP server.)
#dhcp-boot=pxelinux.0

# The same as above, but use custom tftp-server instead
```

```
machine running dnsmasq
#dhcp-boot=pxelinux,server.name,192.168.1.100

# Boot for Etherboot gPXE. The idea is to send two different
# filenames, the first loads gPXE, and the second tells gPXE
what to
# load. The dhcp-match sets the gpxe tag for requests from
gPXE.
#dhcp-match=set:gpxe,175 # gPXE sends a 175 option.
#dhcp-boot=tag:!gpxe,undionly.kpxe
#dhcp-boot=mybootimage

# Encapsulated options for Etherboot gPXE. All the options are
# encapsulated within option 175
#dhcp-option=encap:175, 1, 5b         # priority code
#dhcp-option=encap:175, 176, 1b       # no-proxydhcp
#dhcp-option=encap:175, 177, string   # bus-id
#dhcp-option=encap:175, 189, 1b       # BIOS drive code
#dhcp-option=encap:175, 190, user     # iSCSI username
#dhcp-option=encap:175, 191, pass     # iSCSI password

# Test for the architecture of a netboot client. PXE clients
are
# supposed to send their architecture as option 93. (See RFC
4578)
#dhcp-match=peecees, option:client-arch, 0 #x86-32
#dhcp-match=itanics, option:client-arch, 2 #IA64
#dhcp-match=hammers, option:client-arch, 6 #x86-64
#dhcp-match=mactels, option:client-arch, 7 #EFI x86-64

# Do real PXE, rather than just booting a single file, this is
an
# alternative to dhcp-boot.
#pxe-prompt="What system shall I netboot?"
# or with timeout before first available action is taken:
#pxe-prompt="Press F8 for menu.", 60

# Available boot services. for PXE.
#pxe-service=x86PC, "Boot from local disk"

# Loads <tftp-root>/pxelinux.0 from dnsmasq TFTP server.
#pxe-service=x86PC, "Install Linux", pxelinux

# Loads <tftp-root>/pxelinux.0 from TFTP server at 1.2.3.4.
# Beware this fails on old PXE ROMS.
#pxe-service=x86PC, "Install Linux", pxelinux, 1.2.3.4

# Use bootserver on network, found my multicast or broadcast.
#pxe-service=x86PC, "Install windows from RIS server", 1
```

```
# Use bootserver at a known IP address.
#pxe-service=x86PC, "Install windows from RIS server", 1,
1.2.3.4

# If you have multicast-FTP available,
# information for that can be passed in a similar way using
options 1
# to 5. See page 19 of
#
http://download.intel.com/design/archives/wfm/downloads/pxespe
c.pdf


# Enable dnsmasq's built-in TFTP server
#enable-tftp

# Set the root directory for files available via FTP.
#tftp-root=/var/ftpd

# Make the TFTP server more secure: with this set, only files
owned by
# the user dnsmasq is running as will be send over the net.
#tftp-secure

# This option stops dnsmasq from negotiating a larger
blocksize for TFTP
# transfers. It will slow things down, but may rescue some
broken TFTP
# clients.
#tftp-no-blocksize

# Set the boot file name only when the "red" tag is set.
#dhcp-boot=net:red,pxelinux.red-net

# An example of dhcp-boot with an external TFTP server: the
name and IP
# address of the server are given after the filename.
# Can fail with old PXE ROMS. Overridden by --pxe-service.
#dhcp-boot=/var/ftpd/pxelinux.0,boothost,192.168.0.3

# If there are multiple external tftp servers having a same
name
# (using /etc/hosts) then that name can be specified as the
# tftp_servername (the third option to dhcp-boot) and in that
# case dnsmasq resolves this name and returns the resultant IP
# addresses in round robin fasion. This facility can be used
to
# load balance the tftp load among a set of servers.
#dhcp-boot=/var/ftpd/pxelinux.0,boothost,tftp_server_name

# Set the limit on DHCP leases, the default is 150
```

```
#dhcp-lease-max=150

# The DHCP server needs somewhere on disk to keep its lease
database.
# This defaults to a sane location, but if you want to change
it, use
# the line below.
#dhcp-leasefile=/var/lib/misc/dnsmasq.leases

# Set the DHCP server to authoritative mode. In this mode it
will barge in
# and take over the lease for any client which broadcasts on
the network,
# whether it has a record of the lease or not. This avoids
long timeouts
# when a machine wakes up on a new network. DO NOT enable this
if there's
# the slightest chance that you might end up accidentally
configuring a DHCP
# server for your campus/company accidentally. The ISC server
uses
# the same option, and this URL provides more information:
# http://www.isc.org/files/auth.html
#dhcp-authoritative

# Run an executable when a DHCP lease is created or destroyed.
# The arguments sent to the script are "add" or "del",
# then the MAC address, the IP address and finally the
hostname
# if there is one.
#dhcp-script=/bin/echo

# Set the cachesize here.
#cache-size=150

# If you want to disable negative caching, uncomment this.
#no-negcache

# Normally responses which come form /etc/hosts and the DHCP
lease
# file have Time-To-Live set as zero, which conventionally
means
# do not cache further. If you are happy to trade lower load
on the
# server for potentially stale date, you can set a time-to-
live (in
# seconds) here.
#local-ttl=

# If you want dnsmasq to detect attempts by Verisign to send
```

```
queries
# to unregistered .com and .net hosts to its sitefinder
service and
# have dnsmasq instead return the correct NXDOMAIN response,
uncomment
# this line. You can add similar lines to do the same for
other
# registries which have implemented wildcard A records.
#bogus-nxdomain=64.94.110.11

# If you want to fix up DNS results from upstream servers, use
the
# alias option. This only works for IPv4.
# This alias makes a result of 1.2.3.4 appear as 5.6.7.8
#alias=1.2.3.4,5.6.7.8
# and this maps 1.2.3.x to 5.6.7.x
#alias=1.2.3.0,5.6.7.0,255.255.255.0
# and this maps 192.168.0.10->192.168.0.40 to
10.0.0.10->10.0.0.40
#alias=192.168.0.10-192.168.0.40,10.0.0.0,255.255.255.0

# Change these lines if you want dnsmasq to serve MX records.

# Return an MX record named "maildomain.com" with target
# servermachine.com and preference 50
#mx-host=maildomain.com,servermachine.com,50

# Set the default target for MX records created using the
localmx option.
#mx-target=servermachine.com

# Return an MX record pointing to the mx-target for all local
# machines.
#localmx

# Return an MX record pointing to itself for all local
machines.
#selfmx

# Change the following lines if you want dnsmasq to serve SRV
# records.  These are useful if you want to serve ldap
requests for
# Active Directory and other windows-originated DNS requests.
# See RFC 2782.
# You may add multiple srv-host lines.
# The fields are <name>,<target>,<port>,<priority>,<weight>
# If the domain part if missing from the name (so that is just
has the
# service and protocol sections) then the domain given by the
domain=
# config option is used. (Note that expand-hosts does not need
```

```
to be
# set for this to work.)

# A SRV record sending LDAP for the example.com domain to
# ldapserver.example.com port 389
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389

# A SRV record sending LDAP for the example.com domain to
# ldapserver.example.com port 389 (using domain=)
#domain=example.com
#srv-host=_ldap._tcp,ldapserver.example.com,389

# Two SRV records for LDAP, each with different priorities
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389,1
#srv-host=_ldap._tcp.example.com,ldapserver.example.com,389,2

# A SRV record indicating that there is no LDAP server for the
domain
# example.com
#srv-host=_ldap._tcp.example.com

# The following line shows how to make dnsmasq serve an
arbitrary PTR
# record. This is useful for DNS-SD. (Note that the
# domain-name expansion done for SRV records _does_not
# occur for PTR records.)
#ptr-record=_http._tcp.dns-sd-services,"New Employee
Page._http._tcp.dns-sd-services"

# Change the following lines to enable dnsmasq to serve TXT
records.
# These are used for things like SPF and zeroconf. (Note that
the
# domain-name expansion done for SRV records _does_not
# occur for TXT records.)

#Example SPF.
#txt-record=example.com,"v=spf1 a -all"

#Example zeroconf
#txt-record=_http._tcp.example.com,name=value,paper=A4

# Provide an alias for a "local" DNS name. Note that this
_only_ works
# for targets which are names from DHCP or /etc/hosts. Give
host
# "bert" another name, bertrand
#cname=bertand,bert

# For debugging purposes, log each DNS query as it passes
```

```
through
# dnsmasq.
#log-queries

# Log lots of extra information about DHCP transactions.
#log-dhcp

# Include a another lot of configuration options.
#conf-file=/etc/dnsmasq.more.conf
#conf-dir=/etc/dnsmasq.d
```

Explication des paramètres

Cf la page officielle de man en français : http://www.linuxcertif.com/man/8/dnsmasq/

Fichier /etc/hosts

no-hosts
: Ne pas charger les noms du fichier **/etc/hosts**.

addn-hosts=<fichier_ou_répertoire>
: Fichiers d'hôtes additionnels.

- Lire le fichier ou le répertoire spécifié (=les fichiers contenus dans ce répertoire) en plus de **/etc/hosts**.
- Cette option peut être répétée pour ajouter d'autres fichiers.

Nom de domaine

domain=<domaine>[,<gamme d'adresses>]
: domaine du serveur DHCP, valide pour les hôtes DHCP configurés.

- Le domaine peut être donné de manière inconditionnelle (sans spécifier de gamme d'adresses IP) ou pour des gammes d'adresses IP limitées.
- Si aucun nom de domaine n'est spécifié, les noms d'hôtes avec un nom de domaine (c-à-d un point dans le nom) seront interdits et enregistrés dans le journal (logs).
- Si un suffixe est fourni,
  - les noms d'hôtes possédant un domaine sont autorisés si le nom de domaine coïncide avec le nom fourni.
  - les noms d'hôtes ne possédant pas de nom de domaine se voient rajouter le suffixe fourni dans l'option domain.

4. Ainsi, sur mon réseau, je peux configurer **domain=thekelleys.org.uk** et avoir une machine dont le nom DHCP serait **laptop**. L'adresse IP de cette machine sera disponible à la fois pour **laptop** et **laptop.thekelleys.org.uk**.
5. Si la valeur fournie pour <domaine> est “#”, alors le nom de domaine est positionné à la première valeur de la directive “search” du fichier /etc/resolv.conf (ou équivalent).

6. <gamme d'adresses> : de la forme <adresse ip>,<adresse ip> ou <adresse ip>/<masque de réseau> voire une simple <adresse ip>.

expand-hosts
: Ajoute le nom de domaine aux noms simples (qui ne contiennent pas de point) :

- contenus dans le fichier **/etc/hosts**
- ou fournis par DHCP.

localise-queries
: Répond aux requêtes DNS de l'interface sur laquelle la requête a été reçue, à partir du fichier **/etc/hosts**.

- Si un serveur a de multiples adresses, une pour chaque interface dans **/etc/hosts**, dnsmasq retourne l'adresse du réseau de l'hôte appelant.

domain-needed
: empêche Dnsmasq de transmettre en amont les requêtes pour des noms simples (ne comprenant ni points ni nom de domaine).

- Si un nom n'est pas dans **/etc/hosts** ni dans la liste des baux DHCP, renvoie “non trouvé”

TTL (durée de vie)

local-ttl=<durée>
: durée de vie à retourner (en secondes) quand dnsmasq retourne une information provenant du fichier **/etc/hosts** ou d'un bail DHCP

- valeur par défaut = zéro → pas de mise en cache (généralement correct)

neg-ttl=<durée>
: durée de vie par défaut (en secondes) pour la mise en cache des réponses négatives des serveurs amont, quand ceux-ci ne donnent pas cette information dont dnsmasq se sert pour mettre la réponse en cache.
Sans cette option, dnsmasq ne met alors pas en cache la réponse.

max-ttl=<durée>
: valeur maximum de TTL retournée, si la vraie valeur de TTL est supérieure.

Fichiers de log

log-queries
: Tracer les résultats des requêtes DNS traitées par Dnsmasq dans un fichier de traces (“logs”).

pid-file=<chemin>
: fichier où stocker le numéro de processus (pid). Valeur par défaut : **/var/run/dnsmasq.pid**.

Utilisateur et groupe

user=<nom d'utilisateur>
: utilisateur sous lequel dnsmasq s'exécute.

- valeur par défaut = nobody

group=<nom de groupe>
: groupe sous lequel Dnsmasq s'exécute.

- valeur par défaut = “dip”

Port

port=<port>
: <port> numéro du port écouté

- valeur par défaut : port DNS standard (53).
- une valeur de zéro désactive la fonction DNS pour ne laisser actif que le DHCP

Interface et adresses d'écoute

L'ordre des options **listen-address**, **interface** et **except-interface** n'a pas d'importance

interface=<nom d'interface>
: N'écouter que sur l'interface réseau <nom d'interface>.

- l'interface locale (“loopback”) est automatiquement ajoutée à la liste des interfaces.
- En l'absence d'option **interface** ou **listen-address**, Dnsmasq écoutera sur toutes les interfaces disponibles sauf celle(s) spécifiée(s) par l'option **except-interface**.

except-interface=<interface name>
: Ne pas écouter sur l'interface réseau <interface name>

- l'option **except-interface** l'emporte sur les autres.

listen-address=<adresse IP>
: Ecouter sur l'adresse IP <adresse IP>

- Si les options **interface** et **listen-address** sont fournies simultanément, un jeu d'interfaces et d'adresses sera utilisé.

  si aucune option **interface** n'est donnée alors qu'une option **listen-address** l'est, Dnsmasq n'écoutera pas automatiquement sur l'interface locale (“loopback”). Pour activer l'écoute sur l'interface locale, il faut alors fournir explicitement son adresse IP, 127.0.0.1 via l'option **listen-address**.

Désactivation de DNS ou DHCP

no-dhcp-interface=<nom d'interface>
: désactive DHCP sur cette interface mais laisse actif le DNS.

port=0
: désactive la fonction DNS mais laisse actif le DHCP

Pour éviter de sortir

bogus-priv
: Fausse résolution inverse pour les réseaux privés.

- pour les requêtes DNS inverses pour des adresses IP privées (ie **192.168.x.x**, etc…) qui ne sont pas trouvées dans **/etc/hosts** ou dans les baux DHCP, dnsmasq retourne une réponse (“no such domain”) au lieu d'être transmises aux serveurs de nom amont

alias=[<ancienne IP>]|[<IP de début>-<IP de fin>],<nouvelle IP>[,<masque>]
: Modifie les adresses IPv4 retournées par les serveurs de nom amont; <ancienne IP> est remplacée par <nouvelle IP>.
Si le <masque> optionnel est fourni, alors toute adresse correspondant à l'adresse <ancienne IP>/<masque> sera réécrite.
par exemple **alias=1.2.3.0,6.7.8.0,255.255.255.0** modifiera **1.2.3.56** en **6.7.8.56** et **1.2.3.67** en **6.7.8.67**.
Si l'ancienne IP est donnée sous la forme d'une gamme d'adresses, alors seules les adresses dans cette gamme seront récrites, et non le sous-réseau dans son ensemble. Ainsi,
alias=192.168.0.10-192.168.0.40,10.0.0.0,255.255.255.0 fait correspondre **192.168.0.10** → **192.168.0.40** à **10.0.0.10** → **10.0.0.40**

Blocage d'adresses

bogus-nxdomain=<adresse IP>
: Pour l'adresse <adresse IP>, dnsmasq retourne “no such domain”.
Ceci bloque cette adresse IP. Permet de bloquer la modification sournoise mise en place par Verisign en septembre 2003, qui retourne l'adresse d'un serveur web publicitaire (64.94.110.11) en réponse aux requêtes pour les noms de domaines non enregistrés.

filterwin2k
: active le filtrage des requêtes DNS périodiques faites par certaines versions de windows auxquelles les serveurs DNS ne peuvent donner de réponse.

Fichier resolv.conf

resolv-file=<fichier>
: Lit les adresses des serveurs de nom amont dans le fichier <fichier>, au lieu

du fichier **/etc/resolv.conf**.

- Pour le format de ce fichier, voir dans le manuel de resolv.conf(5) les entrées correspondant aux serveurs de noms (nameserver).
- Dnsmasq peut lire plusieurs fichiers de type resolv.conf, le premier fichier spécifié remplace le fichier par défaut, le contenu des suivants est rajouté dans la liste des fichiers à consulter.

no-resolv
: Ne pas lire le contenu du fichier **/etc/resolv.conf**. N'obtenir l'adresse des serveurs de nom amont que depuis le fichier de configuration de Dnsmasq.

strict-order
: force Dnsmasq à interroger les serveurs DNS dans leur ordre d'apparition dans le fichier **/etc/resolv.conf**.

all-servers
: Cette option force dnsmasq à effectuer ses requêtes à tous les serveurs disponibles. En effet, par défaut, lorsque dnsmasq a plus d'un serveur amont disponible, il n'envoie les requêtes qu'à un seul serveur.

- résultat renvoyé : celui fourni par le premier serveur ayant répondu.

local, server=[/[<domaine>]/[domaine/]][<Adresse IP>[#<port>][@<Adresse IP source>|<interface>[#<port>]]]
: adresse IP d'un serveur de nom amont, éventuellement pour le ou les domaines indiqués
  - concerne le réseau interne : si le réseau comporte un serveur de nom d'adresse IP **192.168.1.1**, résolvant les des noms de la forme **xxx.internal.thekelleys.org.uk**, alors **server=/internal.thekelleys.org.uk/192.168.1.1** envoie les requêtes pour les machines internes vers ce serveur de nom, ; les autres requêtes sont adressées aux serveurs indiqués dans le fichier /etc/resolv.conf.
  - // : si le domaine spécifié est vide : uniquement pour les noms sans domaine, qui n'ont pas de point.
  - #<port> : on peut spécifier un port non standard à la suite des adresses IP.
  - on peut mettre plusieurs options **server**, avec la même syntaxe.
  - Le domaine le plus spécifique l'emporte sur le domaine le moins spécifique, ainsi :

    ```
    server=/google.com/1.2.3.4
    server=/www.google.com/2.3.4.5
    ```

    enverra ***.google.com** → **1.2.3.4**, sauf ***www.google.com** → **2.3.4.5**.

  - L'adresse spéciale **#** = serveurs standards, ainsi

    ```
    server=/google.com/1.2.3.4
    ```

```
server=/www.google.com/#
```

enverra les requêtes ***.google.com** → **1.2.3.4**, sauf ***www.google.com** → serveurs définis par défaut.

- nom de domaine sans adresse IP = domaine local ; n'utiliser que les entrées contenues dans le fichier **/etc/hosts** ou les baux DHCP, ne jamais transmettre les requêtes aux serveurs amont.

- **local** : synonyme de **server** pour clarifier l'utilisation de cette option pour cet usage particulier.
- chaîne de caractères optionnelle après le caractère @ : source à utiliser pour les réponses à ce serveur de nom. Ce peut être :
  - une adresse IP de la machine sur laquelle tourne dnsmasq. Dans ce cas, l'adresse source de la requête sera l'adresse en question
  - ou un nom d'interface. Dans ce cas, les requêtes vers le serveur de nom seront envoyées depuis cette interface

3. L'option query-port est ignorée pour tous les serveurs ayant une adresse source spécifiée, mais on peut la fournir dans la spécification de l'adresse source.

address=/<domaine>/[domaine/]<adresse IP>
: adresse IP à retourner pour les domaines indiqués [6]. Les requêtes pour ce ou ces domaines ne sont jamais transmises aux serveurs amont et retournent l'adresse IP spécifiée.

- Pour donner à la fois une adresse IPv4 et une adresse IPv6 pour un domaine, utiliser plusieurs options **address**.
- pour des noms individuels, le contenu du fichier **/etc/hosts** et de celui des baux DHCP prend le dessus sur ce mécanisme.
- Une utilisation courante de cette option : rediriger le domaine **doubleclick.net** vers un serveur web local pour éviter les bannières publicitaires.
- **<domaine>** : même fonctionnement que **server**, avec en plus que /#/ coïncide avec tout domaine. Ainsi, **address=/#/1.2.3.4** retournera **1.2.3.4** pour toute requête n'ayant de réponse ni dans **/etc/hosts**, ni dans les baux DHCP, et non transmise à un serveur spécifique par le biais d'une directive **server**.

cname=<cname>,<cible>
: enregistrement de type CNAME qui indique que <cname> est en réalité <cible>.

- <cible> doit être un nom DNS connu de dnsmasq via /etc/hosts (ou un fichier hôtes additionnel) ou via DHCP.
- Ignoré si une cible ne satisfait pas ces critères
- une cible peut avoir plusieurs CNAME

interface-name=<nom>,<interface>

entregistrement DNS associant <nom> avec l'adresse primaire sur <interface>.

- spécifie un enregistrement de type A pour le nom donné en argument comme s'il était défini par une ligne de /etc/hosts, sauf que l'adresse n'est pas constante mais dépend de l'interface définie.
- Si l'interface est inactive, inexistante ou non configurée, retourne une réponse vide.
- Un enregistrement inverse (PTR) est également créé, associant l'adresse de l'interface avec le nom.
- cette option peut être répétée pour associer plus d'un nom à une interface donnée ; dans ce cas, l'enregistrement inverse pointe vers le nom fourni dans la première instance de cette option.

Cache

cache-size=<taille>
: taille du cache de Dnsmasq

- valeur par défaut = 150 noms.
- une valeur nulle désactive le cache.

no-negcache
: Désactive le “cache négatif”.

- Le “cache négatif” permet à Dnsmasq de se souvenir des réponses de type “no such domain” fournies par les serveurs DNS en amont et dene pas avoir à re-transmettre les requêtes aux serveurs amont.

dns-forward-max=<nombre de requêtes>
: nombre maximum de requêtes DNS simultanées.

- valeur par défaut = 150 (en général suffisant)

DHCP

dhcp-range=[interface:<interface>,][tag:<label>[,tag:<label>],][set:<label],]<adresse de début>,<adresse de fin>[,<masque de réseau>[,<broadcast>]][,<durée de bail>]
: Active le serveur DHCP.

- Les adresses seront données dans la plage comprise entre <adresse de début> et <adresse de fin> et à partir des adresses définies statiquement dans l'option dhcp-host.
- Si une durée de bail est donnée, les baux seront donnés pour cette durée.
- La durée de bail est donnée en secondes, en minutes (exemple : 45m), en heures (exemple : 1h) ou être la chaine de caractère “infinite” pour une durée indéterminée.

- Si aucune valeur n'est donnée, la durée de bail par défaut est de une heure.
- La valeur minimum pour un bail DHCP est de 2 minutes.
- Cette option peut être répétée, avec différentes adresses, pour activer le service DHCP sur plus d'un réseau.
- Pour des réseaux directement connectés (c'est-à-dire des réseaux dans lesquels la machine sur laquelle tourne Dnsmasq possède une interface), le masque de réseau est optionnel.
- Il est par contre requis pour les réseaux pour lesquels le service DHCP se fait via un relais DHCP (“relay agent”).
- L'adresse de broadcast est toujours optionnelle.
- un même sous-réseau peut toujours avoir plus d'une plage DHCP.
- L'identifiant de label optionnel set:<label> identifie ce réseau par une étiquette alphanumérique pour permettre la fourniture d'options DHCP spécifiques à chaque réseau.
  - Lorsque préfixé par 'tag:', la signification change, et au lieu de définir un label, il définit le label pour laquelle la règle s'applique.
  - Un seul label peut être défini mais plusieurs labels peuvent coïncider.

12. L'adresse de fin peut être remplacée par le mot-clef static (“statique”) qui indique à Dnsmasq d'activer le service DHCP pour le réseau spécifié, mais de ne pas activer l'allocation dynamique d'adresses IP : Seuls les hôtes possédant des adresses IP statiques fournies via dhcp-host ou présentes dans le fichier /etc/ethers seront alors servis par le DHCP.
13. L'adresse de fin peut-être remplacée par le mot-clef proxy, auquel cas Dnsmasq fournira un service de DHCP proxy pour le sous-réseau spécifié.
14. La section interface:<nom d'interface> n'est normalement pas utilisée.

dhcp-host=[<adresse matérielle>][,id:<identifiant client>|*][,set:<label>][,<adresse IP>][,<nom d'hôte>][,<durée de bail>][,ignore]

paramètres DHCP relatifs à un hôte.

- Cela permet à une machine possédant une adresse matérielle spécifique de se voir toujours allouée les mêmes nom d'hôte, adresse IP et durée de bail.
- Le nom d'hôte spécifié ainsi remplace le nom fourni par le client DHCP de la machine hôte.
- On peut aussi omettre l'adresse matérielle et d'inclure le nom d'hôte, auquel cas l'adresse IP et la durée de bail s'appliqueront à toute machine se réclamant de ce nom.
  - Par exemple **dhcp-host=00:20:e0:3b:13:af,wap,infinite** demande à Dnsmasq de fournir le nom “wap” à la machine d'adresse matérielle **00:20:e0:3b:13:af** et un bail de durée indéterminée.
  - **dhcp-host=lap,192.168.0.199** demande à Dnsmasq de toujours allouer à la machine portant le nom lap l'adresse IP 192.168.0.199.

4. Les adresses allouées ne sont pas contraintes à une plage d'adresse spécifiée par une option **dhcp-range**, mais doivent se trouver dans le même sous-réseau qu'une plage dhcp-range valide.
5. Pour les sous-réseaux qui n'ont pas besoin d'adresses dynamiques, utiliser le mot-clef “static” dans la déclaration de plage d'adresses dhcp-range.
6. On peut utiliser des identifiants clients plutôt que des adresses matérielles pour identifier les hôtes, en préfixant par ceux-ci par 'id:'.
    - Ainsi, **dhcp-host=id:01:02:03:04,.....** se réfère à l'hôte d'identifiant 01:02:03:04.
7. On peut aussi spécifier l'identifiant client sous la forme d'une chaîne de caractères, comme ceci : **dhcp-host=id:identifiantclientsousformedechaine,.....**
8. L'option spéciale id:* signifie : “ignorer tout identifiant client et n'utiliser que l'adresse matérielle” ; utile quand un client présente un identifiant client mais pas les autres.
9. Si un nom apparaît dans /etc/hosts, l'adresse associée peut être allouée à un bail DHCP mais seulement si une option **dhcp-host** spécifiant le nom existe par ailleurs.
10. Seul un nom d'hôte peut-être donné dans une option **dhcp-host** mais les alias sont possibles au travers de l'utilisation des CNAMEs.
11. mot clef **ignore** : ne jamais fournir de bail DHCP à une machine.
12. La machine peut être spécifiée par son adresse matérielle, son identifiant client ou son nom d'hôte. Par exemple **dhcp-host=00:20:e0:3b:13:af,ignore** ; utile lorsqu'un autre serveur DHCP sur le réseau doit être utilisé par certaines machines.
13. paramètre **set:<identifiant réseau>** : définit un identifiant réseau lorsque l'option dhcp-host est utilisée. Cela peut servir à sélectionner des options DHCP juste pour cet hôte.
14. On peut donner plus d'un label dans une directive dhcp-host (et dans cette seule directive).
15. Quand une machine coïncide avec une directive dhcp-host, le label réservé “known” (“connu”) est associé. Ainsi, Dnsmasq peut ignorer les requêtes issus de machines inconnues par le biais de **dhcp-ignore=tag:!known**.
16. Les adresses ethernet (mais pas les identifiants clients) peuvent être définies avec des jokers,
    - ainsi par exemple **dhcp-host=00:20:e0:3b:13:*,ignore** demande à Dnsmasq d'ignorer une gamme d'adresses matérielles.

dhcp-option=[tag:<label>,[tag:<label>]][encap:<option>,][vi-encap:<entreprise>,][vendor:[<classe_vendeur>],][<option>|option:<nom d'option>],[<valeur>[,<valeur>]]

options pour des clients DHCP.

- Par défaut, Dnsmasq envoie un ensemble standard d'options aux clients DHCP : le masque de réseau et l'adresse de broadcast sont

les mêmes que pour l'hôte sur lequel tourne Dnsmasq, et le serveur DNS ainsi que la route par défaut prennent comme valeur l'adresse de la machine sur laquelle tourne Dnsmasq.

- L'option DHCP à transmettre est fournie sous forme d'un nombre décimal ou sous la forme “option:<nom d'option>”.
- Les noms d'options connus par Dnsmasq peuvent être obtenus via “Dnsmasq –help dhcp”.
    - Par exemple, pour définir la route par défaut à 192.168.4.4, il est possible de faire **dhcp-option=3,192.168.4.4** ou **dhcp-option = option:router, 192.168.4.4**
    - pour positionner l'adresse du serveur de temps à 192.168.0.4, on peut faire **dhcp-option = 42,192.168.0.4** ou **dhcp-option = option:ntp-server, 192.168.0.4**
    - L'adresse 0.0.0.0 prend ici le sens “d'adresse de la machine sur laquelle tourne Dnsmasq”.

4. Les types de données autorisées sont des adresses IP sous la forme de 4 chiffres séparés par des points, un nombre décimal, une liste de caractères hexadécimaux séparés par des 2 points, ou une chaîne de caractères.
5. Si des labels optionnels sont fournis, alors cette option n'est envoyée qu'aux réseaux dont tous les labels coïncident avec ceux de la requête.

dhcp-option-force=[tag:<label>,[tag:<label>]][encap:<option>,][vi-encap:<entreprise>,][vendor:[<classe_vendeur>],][<option>|option:<nom d'option>],[<valeur>[,<valeur>]]
: même fonctionnement que **dhcp-option** sauf que cette option sera toujours envoyée, même si le client ne la demande pas dans la liste de paramètres requis.

dhcp-vendorclass=set:<label>,<classe de vendeur>
: Associe une chaîne de classe de vendeur à un label. La plupart des clients DHCP fournissent une “classe de vendeur” (“vendor class”) qui représente, d'une certaine façon, le type d'hôte.

- Cette option associe des classes de vendeur à des labels, de sorte que des options DHCP peuvent-être fournies de manière sélective aux différentes classes d'hôtes.
- Par exemple, **dhcp-vendorclass=set:printers,Hewlett-Packard JetDirect** ou **dhcp-vendorclass=printers,Hewlett-Packard JetDirect** permet de n'allouer des options qu'aux imprimantes HP de la manière suivante : **dhcp-option=tag:printers,3,192.168.4.4**
- La chaîne de caractères de la classe de vendeur founie en argument est cherchée en temps que sous-chaîne de

caractères au sein de la classe de vendeur fournie par le client, de façon à permettre la recherche d'un sous-ensemble de la chaîne de caractères (“fuzzy matching”).

- Le préfixe set: est optionnel mais autorisé afin de conserver une certaine homogénéité.

dhcp-userclass=set:<label>,<classe utilisateur>

Associe une chaîne de classe d'utilisateur à un label (effectue la recherche sur des sous-chaînes, comme pour les classes de vendeur). La plupart des clients permettent de configurer une “classe d'utilisateur”.

- Cette option associe une classe d'utilisateur à un label, de telle manière qu'il soit possible de fournir des options DHCP spécifiques à différentes classes d'hôtes.
- Il est possible, par exemple, d'utiliser ceci pour définir un serveur d'impression différent pour les hôtes de la classe “comptes” et ceux de la classe “ingénierie”.

dhcp-mac=set:<label>,<adresse MAC>

Associe une adresse matérielle (MAC) à un label. L'adresse matérielle peut inclure des jokers. Par exemple **dhcp-mac=set:3com,01:34:23:*:*:*** permet de définir le label “3com” pour n'importe quel hôte dont l'adresse matérielle coïncide avec les critères définis.

dhcp-fqdn

par défaut, dnsmasq insère les noms non-qualifiés des clients DHCP dans le DNS.

- Pour cette raison, les noms doivent être uniques, même si deux clients ayant le même nom sont dans deux domaines différents.
- Si un deuxième client DHCP apparaît avec le même nom qu'un client déjà existant, ce nom est transféré au nouveau client.
- Si **dhcp-fqdn** est spécifié, ce comportement change : les noms non qualifiés ne sont plus rajoutés dans le DNS, seuls les noms qualifiés le sont.
- Deux clients DHCP avec le même nom peuvent tous les deux garder le nom, pour peu que la partie relative au domaine soit différente (c-à-d que les noms pleinements qualifiés diffèrent).
- Pour s'assurer que tous les noms ont une partie domaine, il doit y avoir au moins un **domain** sans gamme d'adresses lorsque l'option **dhcp-fqdn** est configurée.

Dnsmasq est un logiciel de transmission de requêtes DNS : il n'est pas capable d'effectuer une résolution de nom récursive en partant des serveurs DNS racine, mais transmet de telles

requêtes à un serveur DNS amont capable de telles recherches récursives, ce qui est typiquement le cas d'un serveur DNS de FAI.

Par défaut, Dnsmasq lit /etc/resolv.conf pour découvrir les adresses IP des serveurs DNS amonts à utiliser, puisque cette information est en général stockée à cet endroit.

Pour que Dnsmasq se comporte comme un cache pour la machine sur laquelle il tourne, mettre “nameserver 127.0.0.1” dans le fichier **/etc/resolv.conf** pour forcer les processus locaux à envoyer leurs requêtes à Dnsmasq. Ensuite, spécifier les serveurs DNS amont via l'option **server**.e des addresses de serveurs DNS amont par le biais de PPP ou DHCP.

Les adresses dans /etc/hosts prennent le dessus sur celles fournies par le serveur DNS amont, ainsi “macompagnie.com 1.2.3.4” dans /etc/hosts assure que les requêtes pour “macompagnie.com” retourneront toujours 1.2.3.4, même si une requête au serveur DNS amont retournerait une adresse différente.

le système de label fonctionne comme suit : pour chaque requête DHCP, dnsmasq associe un ensemble de labels obtenus à partir des lignes de la configuration incluant set:<label>, y compris un pour la plage d'adresse (dhcp-range) utilisée pour allouer l'adresse, un pour chaque entrée dhcp-host associée (auquel est rajouté le mot-clef “known” si une entrée dhcp-host coïncide).

Pour les lignes de configuration comportant des éléments tag:<label>, seules seront valides celles pour lesquels tous les labels correspondants seront présents.

C'est typiquement le cas des lignes dhcp-options.

Un dhcp-option possédant des labels sera utilisé de préférence à un dhcp-option sans label, pour peu que _tous_ les labels positionnés correspondent à l'ensemble de labels décrit plus haut.

Le préfixe '!' sur un label est un indicateur de négation, ainsi **dhcp=option=tag:!purple,3,1.2.3.4** n'envoie l'option que lorsque le label “purple” n'est pas dans la liste de labels définis pour l'hôte considéré.

Veuillez noter que pour dhcp-range, les éléments tag:<label> et set:<label> sont tous les deux autorisés pour sélectionner la plage à utiliser selon, par exemple, le dhcp-host, et pour affecter l'option envoyée, sur la base de la plage sélectionnée.

Il est possible de spécifier un nom d'interface à dhcp-range sous la forme “interface:<nom d'interface>”. La sémantique est comme suit : Pour le DHCP, s'il existe une autre valeur de dhcp-

range pour laquelle _aucun_ nom d'interface n'est donné, alors le nom d'interface est ignoré et dnsmasq se comporte comme si la partie spécifiant l'interface n'existait pas, sinon le service DHCP n'est fourni qu'aux interfaces mentionnées dans les déclarations dhcp-range. Pour le DNS, si il n'y a pas d'option **interface** ou **listen-address**, alors le comportement n'est pas impacté par la spécification d'interface.

Si l'une ou l'autre de ces options est présente, alors les interfaces mentionnées dans les plages d'adresses dhcp-range sont rajoutées à la liste de celles où le service DNS est assuré.

Il est possible d'utiliser Dnsmasq pour bloquer la publicité sur la toile en associant des serveurs de publicité bien connus à l'adresse 127.0.0.1 ou 0.0.0.0 par le biais du fichier /etc/hosts ou d'un fichier d'hôte additionnel. Cette liste peut-être très longue, Dnsmasq ayant été testé avec succès avec un million de noms. Cette taille de fichier nécessite un processeur à 1 Ghz et environ 60 Mo de RAM.

On évite de faire sortir les requêtes de domaines locaux

domain-needed
: ne jamais transmettre en amont les requêtes pour des noms simples (sans point ni nom de domaine). Si un nom n'est pas dans /etc/hosts ni dans la liste des baux DHCP, une réponse “non trouvé” est renvoyée.

bogus-priv
: Fausse résolution inverse pour les réseaux privés. Toutes les requêtes DNS inverses pour des adresses IP privées (ie 192.168.x.x, etc...) qui ne sont pas trouvées dans /etc/hosts ou dans le fichier de baux DHCP se voient retourner une réponse “pas de tel domaine” (“no such domain”) au lieu d'être transmises aux serveurs de nom amont (“upstream server”).

Ajout de nom de domaine :

domain=<domaine>[,<gamme d'adresses>]
: Domaine local qui sera ajouté aux noms DNS des machines assignées par le DHCP et fourni en nom de domaine local aux clients DHCP.
domaine du serveur DHCP. Le domaine peut être donné de manière inconditionnelle (sans spécifier de gamme d'adresses IP) ou pour des gammes d'adresses IP limitées.

cela contraint les noms d'hôte afin qu'aucun hôte sur le LAN ne puisse fournir via DHCP un nom tel que par exemple “microsoft.com” et capturer du trafic de manière illégitime.
Si aucun nom de domaine n'est spécifié, les noms d'hôtes avec un nom de domaine (c-à-d un point dans le nom) seront interdits et enregistrés dans le journal (logs).
Si un suffixe est fourni,

- les noms d'hôtes possédant un domaine sont autorisés, pour peu que ce nom de domaine coïncide avec le nom fourni.
- les noms d'hôtes ne possédant pas de nom de domaine se voient rajouter le suffixe fourni dans l'option **domain**.

Par exemple,

domain=thekelleys.org.uk
: si le nom DHCP d'une machine est “laptop”, l'adresse IP de cette machine sera disponible à la fois pour “laptop” et “laptop.thekelleys.org.uk”. Fournit le nom de domaine pour “expand-hosts”

domain=wireless.thekelleys.org.uk,192.168.2.0/24
: Définit un domaine différent pour un sous-réseau particulier

domain=reserved.thekelleys.org.uk,192.68.3.100,192.168.3.200
: Même idée, mais avec une plage plutôt qu'un sous-réseau

La gamme d'adresses peut être de la forme <adresse ip>,<adresse ip> ou <adresse ip>/<masque de réseau> voire une simple <adresse ip>.

expand-hosts
: Ajoute le nom de domaine aux noms simples (ne contenant pas de point dans le nom) contenus dans le fichier /etc/hosts, de la même façon que pour le

service DHCP. Notez que cela ne s'applique pas au nom de domaine dans les CNAME, les enregistrements PTR, TXT, etc...

Fichier à lire pour les serveurs DNS par défaut (au lieu de /etc/resolv.conf, utilisé par défaut)

resolv-file=<fichier>

Lire les adresses des serveurs de nom amont dans le fichier de nom <fichier>, au lieu du fichier /etc/resolv.conf. Pour le format de ce fichier, voir dans le manuel pour resolv.conf(5) les entrées correspondant aux serveurs de noms (nameserver). Dnsmasq peut lire plusieurs fichiers de type resolv.conf, le premier fichier spécifié remplace le fichier par défaut, le contenu des suivants est rajouté dans la liste des fichiers à consulter. Seul le fichier ayant la dernière date de modification sera chargé en mémoire.

Par exemple,

```
resolv-file=/etc/resolv.dnsmasq
```

transmission des requêtes relatives à un domaine particulier vers un serveur précis

local, server=[/[<domaine>]/[domaine/]][<Adresse IP>[#<port>][@<Adresse IP source>|<interface>[#<port>]]]

Spécifie directement l'adresse IP d'un serveur de nom amont. Cette option ne supprime pas la lecture du fichier /etc/resolv.conf : utiliser pour cela l'option **-R**. Si un ou plusieurs nom(s) de domaine(s) optionnel(s) sont fournis, ce serveur sera uniquement utilisé pour ce(s) domaine(s) et toute requête concernant ce(s) domaine(s) sera adressée uniquement à ce serveur. Cette option est destinée aux serveurs de nom privés : si vous avez un serveur de nom sur votre réseau ayant pour adresse IP 192.168.1.1 et effectuant la résolution des noms de la forme xxx.internal.thekelleys.org.uk, alors **-S /internal.thekelleys.org.uk/192.168.1.1** enverra toutes les requêtes pour les machines internes vers

ce serveur de nom, alors que toutes les autres requêtes seront adressées aux serveurs indiqués dans le fichier /etc/resolv.conf. Une spécification de nom de domaine vide, // possède le sens particulier de “pour les noms non qualifiés uniquement”, c'est-à-dire les noms ne possédant pas de points. Un port non standard peut être rajouté à la suite des adresses IP en utilisant le caractère #. Plus d'une option **-S** est autorisée, en répétant les domaines et adresses IP comme requis.
Le domaine le plus spécifique l'emporte sur le domaine le moins spécifique, ainsi :

```
server=/google.com/1.2.3.4
server=/www.google.com/2.3.4.5
```

enverra les requêtes pour *.google.com à 1.2.3.4, à l'exception des requêtes *www.google.com, qui seront envoyées à 2.3.4.5.

L'adresse spéciale '#' signifie “utiliser les serveurs standards”, ainsi

```
server=/google.com/1.2.3.4
server=/www.google.com/#
```

enverra les requêtes pour *.google.com à 1.2.3.4, à l'exception des requêtes pour *www.google.com qui seront envoyées comme d'habitude (c-à-d aux serveurs définis par défaut).

Il est également permis de donner une option **-S** avec un nom de domaine mais sans adresse IP ; Cela informe Dnsmasq que le domaine est local et qu'il doit répondre aux requêtes le concernant depuis les entrées contenues dans le fichier /etc/hosts ou les baux DHCP, et ne doit en aucun cas transmettre les requêtes aux serveurs amonts. **local** est synonyme de **server** (“serveur”) afin de rendre plus claire l'utilisation de cette option pour cet usage particulier.
La chaîne de caractères optionnelle suivant le caractère @ permet de définir la source que Dnsmasq doit utiliser pour les réponses à ce serveur de nom. Il doit s'agir d'une des adresses IP appartenant à la machine sur laquelle tourne Dnsmasq ou sinon la ligne sera ignorée et une erreur sera consignée dans le journal des événements, ou alors d'un nom d'interface. Si un nom d'interface est donné, alors les requêtes vers le serveur de nom seront envoyées depuis cette interface; si une adresse ip est donnée, alors l'adresse source de la

requête sera l'adresse en question. L'option query-port est ignorée pour tous les serveurs ayant une adresse source spécifiée, mais il est possible de la donner directement dans la spécification de l'adresse source. Forcer les requêtes à être émises depuis une interface spécifique n'est pas possible sur toutes les plateformes supportées par dnsmasq.

Voici quelques exemples :

server=/localnet/192.168.0.1
: Ajouter d'autres serveurs de noms, avec les spécifications de domaine si elles concernent des domaines non-publics

server=/3.168.192.in-addr.arpa/10.1.2.3
: Exemple de routage de requêtes PTR vers des serveurs de noms : ceci enverra toutes les requêtes adresse→nom pour 192.168.3/24 au serveur de noms 10.1.2.3

local=/localnet/
: Ajouter ici les domaines exclusivement locaux, les requêtes dans ces domaines ne sont traitées que par /etc/hosts ou DHCP

server=10.1.2.3@eth1
: oblige les requêtes vers 10.1.2.3 à être acheminées via eth1

server=10.1.2.3@192.168.1.1#55
: l'adresse source (locale) utilisée

dialogue avec le port 55 de 10.1.2.3 à 192.168.1.1 (il doit évidemment y avoir une interface avec cette IP sur la machine).

```
server=/mondomaine.local/192.168.1.1
server=/1.168.192.in-addr.arpa/192.168.1.1
```

transmission des requêtes relatives à mondomaine.local vers le serveur 192.168.1.1

Interface sur laquelle doit fonctionner le daemon

interface=<nom d'interface>
: N'écouter que sur l'interface réseau spécifiée (par exemple eth0).
Dnsmasq aujoute automatiquement l'interface locale (“loopback”) à la liste des interfaces lorsque cette option est utilisée.
Si aucune option **interface** ou **listen-address** n'est donnée, Dnsmasq écoutera sur toutes les interfaces disponibles sauf celle(s) spécifiée(s) par l'option **except-interface**.
Répétez cette ligne si plus d'une interface.

except-interface=<interface name>
: Ne pas écouter sur l'interface spécifiée.

Peu importe l'ordre dans lequel les options **interface** et **except-interface** ; l'option **except-interface** l'emporte toujours sur les autres.

no-dhcp-interface=<nom d'interface>
: Ne pas fournir de service DHCP sur l'interface spécifiée, mais fournir tout de même le service DNS.

address=/<domaine>/[domaine/]<adresse IP>

Spécifie une adresse IP à retourner pour toute requête pour les domaines fournis en option. Les requêtes pour ce(s) domaine(s) ne sont jamais transmises aux serveurs amonts et reçoivent comme réponse l'adresse IP spécifiée qui peut être une adresse IPv4 ou IPv6. Pour donner à la fois une adresse IPv4 et une adresse IPv6 pour un domaine, utiliser plusieurs options **-A**. Il faut noter que le contenu du fichier /etc/hosts et de celui des baux DHCP supplante ceci pour des noms individuels. Une utilisation courante de cette option est de rediriger la totalité du domaine doubleclick.net vers un serveur web local afin d'éviter les bannières publicitaires. La spécification de domaine fonctionne de la même façon que **server**, avec la caractéristique supplémentaire que **/#/** coïncide avec tout domaine. Ainsi, **address=/#/1.2.3.4** retournera 1.2.3.4 pour toute requête n'ayant de réponse ni dans /etc/hosts, ni dans les baux DHCP, et n'étant pas transmise à un serveur spécifique par le biais d'une directive **server**.

Quelques exemples :

address=/double-click.net/127.0.0.1
: domaines pour lesquels vous voulez forcer à une adresse IP ; cet exemple envoie n'importe quel hôte de double-click.net sur un serveur web local

address=/www.thekelleys.org.uk/fe80::20d:60ff:fe36:f83
: **address=** (et **server=**) fonctionnent aussi avec des adresses IPv6

cache-size=<taille>

Définit la taille du cache de Dnsmasq. La

valeur par défaut est de 150 noms. Définir une valeur de zéro désactive le cache.

dhcp-range=[interface:<interface>,][tag:<label>[,tag:<label>],][set:<label],]<adresse de début>,<adresse de fin>[,<masque de réseau>[,<broadcast>]][,<durée de bail>]

Active le serveur DHCP. Les adresses seront données dans la plage comprise entre <adresse de début> et <adresse de fin> et à partir des adresses définies statiquement dans l'option **dhcp-host**. Si une durée de bail est donnée, alors les baux seront donnés pour cette durée. La durée de bail est donnée en secondes, en minutes (exemple : 45m), en heures (exemple : 1h) ou la chaine de caractère “infinite” pour une durée indéterminée. Si aucune valeur n'est donnée, une durée de bail par défaut de une heure est appliquée. La valeur minimum pour un bail DHCP est de 2 minutes. Cette option peut être répétée, avec différentes adresses, pour activer le service DHCP sur plus d'un réseau. Pour des réseaux directement connectés (c'est-à-dire des réseaux dans lesquels la machine sur laquelle tourne Dnsmasq possède une interface), le masque de réseau est optionnel. Il est par contre requis pour les réseaux pour lesquels le service DHCP se fait via un relais DHCP (“relay agent”). L'adresse de broadcast est toujours optionnelle.
Il est toujours possible d'avoir plus d'une plage DHCP pour un même sous-réseau. L'identifiant de label optionnel **set:<label>** fournit une étiquette alphanumérique qui identifie ce réseau, afin de permettre la fourniture d'options DHCP spécifiques à chaque réseau. Lorsque préfixé par 'tag:', la signification change, et au lieu de définir un label, il définit le label pour laquelle la règle s'applique. Un seul label peut être défini mais plusieurs labels peuvent coïncider. L'adresse de fin peut être remplacée par le mot-clef **static** (“statique”) qui indique à Dnsmasq d'activer le service

DHCP pour le réseau spécifié, mais de ne pas activer l'allocation dynamique d'adresses IP : Seuls les hôtes possédant des adresses IP statiques fournies via **dhcp-host** ou présentes dans le fichier /etc/ethers seront alors servis par le DHCP.
L'adresse de fin peut-être remplacée par le mot-clef **proxy** , auquel cas Dnsmasq fournira un service de DHCP proxy pour le sous-réseau spécifié. (voir **pxe-prompt** et **pxe-service** pour plus de détails).
La section interface:<nom d'interface> n'est normalement pas utilisée. Se référer aux indications de la section NOTES pour plus de détail à ce sujet.

dhcp-range=192.168.0.50,192.168.0.150,12h
: active le serveur DHCP
vous devez fournir la plage d'adresses disponibles pour le bail et éventuellement une durée de bail
Si vous avez plus d'un réseau, répétez cette opération pour chaque réseau sur lequel vous souhaitez fournir le service DHCP

dhcp-range=192.168.0.50,192.168.0.150,255.255.255.0,12h
: exemple d'une plage DHCP avec un masque
Nécessaire pour les réseaux qui atteignent le serveur DHCP dnsmasq via un relais
Si vous ne savez

pas ce qu'est un relais DHCP, pas d'inquiétude

dhcp-range=set:rouge,192.168.0.50,192.168.0.150

exemple d'une plage DHCP avec une étiquette, de sorte que certaines options DHCP ne peuvent être fixées que pour ce réseau

dhcp-range=tag:vert,192.168.0.50,192.168.0.150,12h

N'utiliser cette gamme de DHCP que quand l'étiquette «vert» est mise

dhcp-range=192.168.0.0,static

sous-réseau qui ne peut pas être utilisé pour l'allocation d'adresse dynamique, disponible pour les hôtes avec des lignes dhcp-host
Notez que les déclarations dhcp-host sont ignorées sauf s'il y a un dhcp-range d'un certain type pour le sous-réseau en question
Dans ce cas, le masque est implicite (d'après la configuration réseau de la machine sur laquelle tourne dnsmasq)
On peut aussi préciser un masque explicite

dhcp-range=1234::2, 1234::500, 64, 12h
: Activer DHCPv6. Notez que le préfixe de longueur n'a pas besoin d'être spécifié et vaut par défaut 64 si manquant

dhcp-range=1234::, ra-only
: Faire les annonces de routeur, MAIS PAS DHCP pour ce sous-réseau

dhcp-range=1234::, ra-names
: Faire les annonces de routeur, MAIS PAS DHCP pour ce sous-réseau Essayer aussi d'ajouter des noms à la DNS pour l'adresse IPv6 d'hôtes SLAAC-configured dual-stack. Utilisez le bail DHCPv4 pour obtenir le nom, segment de réseau et l'adresse MAC. On suppose que l'hôte aura également une adresse IPv6 calculée en utilisant l'alogrithme SLAAC

dhcp-range=1234::, ra-only, 48h
: Faire les annonces de routeur, MAIS PAS DHCP pour ce sous-réseau Définir la durée de vie à 46 heures (Remarque: la durée de vie minimum est de 2 heures.)

dhcp-range=1234::2, 1234::500, slaac
: Faire les annonces de routeur et DHCP pour ce sous-réseau. Définissez le bit A dans la RA afin que les clients puissent utiliser des adresses SLAAC aussi bien que DHCP.

dhcp-range=1234::, ra-stateless
: Faire des annonces de routeur et DHCP stateless pour ce sous-réseau. Les clients n'obtiendront pas d'adresse du DHCP, mais ils vont obtenir d'autres informations de configuration. Ils vont utiliser SLAAC pour les adresses

dhcp-range=1234::, ra-stateless, ra-names
: Faire du DHCP stateless, du DHCP et du SLAAC et générer des noms DNS pour les adresses SLAAC de baux DHCPv4

dhcp-host=[<adresse matérielle>][,id:<identifiant client>|*][,set:<label>][,<adresse IP>][,<nom d'hôte>][,<durée de bail>][,ignore]
: Spécifie les paramètres DHCP relatifs à un hôte. Cela permet à une machine possédant une adresse matérielle spécifique de se voir toujours allouée les mêmes nom d'hôte, adresse IP et durée de bail. Un nom d'hôte spécifié comme ceci remplace le nom fourni par le client DHCP de la machine hôte. Il est également possible d'omettre l'adresse

matérielle et d'inclure le nom d'hôte, auquel cas l'adresse IP et la durée de bail s'appliqueront à toute machine se réclamant de ce nom. Par exemple **dhcp-host=00:20:e0:3b:13:af,wap,infinite** spécifie à Dnsmasq de fournir à la machine d'adresse matérielle 00:20:e0:3b:13:af le nom, et un bail de durée indéterminée.
dhcp-host=lap,192.168.0.199 spécifie à Dnsmasq d'allouer toujours à la machine portant le nom lap l'adresse IP 192.168.0.199.
Les adresses allouées de la sorte ne sont pas contraintes à une plage d'adresse spécifiée par une option dhcp-range, mais elles se trouver dans le même sous-réseau qu'une plage dhcp-range valide. Pour les sous-réseaux qui n'ont pas besoin d'adresses dynamiquement allouées, utiliser le mot-clef “static” dans la déclaration de plage d'adresses dhcp-range.
Il est possible d'utiliser des identifiants clients plutôt que des adresses matérielles pour identifier les hôtes, en préfixant par ceux-ci par 'id:'. Ainsi, **dhcp-host=id:01:02:03:04,.....** réfère à l'hôte d'identifiant 01:02:03:04. Il est également possible de spécifier l'identifiant client sous la forme d'une chaîne de caractères, comme ceci : **dhcp-host=id:identifiantclientsousformedechaine,.....**
L'option spéciale id:* signifie : “ignorer tout identifiant client et n'utiliser que l'adresse matérielle”. Cela est utile lorsqu'un client présente un identifiant client mais pas les autres.
Si un nom apparaît dans /etc/hosts, l'adresse associée peut être allouée à un bail DHCP mais seulement si une option **dhcp-host** spécifiant le nom existe par ailleurs. Seul un nom d'hôte peut-être donné dans une option **dhcp-host** mais les alias sont possibles au travers de l'utilisation des CNAMEs. (Voir **cname**).
Le mot clef “ignore” (“ignorer”) indique à Dnsmasq de ne jamais fournir de bail

DHCP à une machine. La machine peut être spécifiée par son adresse matérielle, son identifiant client ou son nom d'hôte. Par exemple **dhcp-host=00:20:e0:3b:13:af,ignore** Cela est utile lorsqu'un autre serveur DHCP sur le réseau doit être utilisé par certaines machines.
Le paramètre set:<identifiant réseau> permet de définir un identifiant de réseau lorsque l'option dhcp-host est utilisée. Cela peut servir à sélectionner des options DHCP juste pour cet hôte.
Plus d'un label peut être fourni dans une directive dhcp-host (et dans cette seule directive). Lorsqu'une machine coïncide avec une directive dhcp-host (ou une impliquée par /etc/ethers), alors le label réservé “known” (“connu”) est associé. Cela permet à Dnsmasq d'être configuré pour ignorer les requêtes issus de machines inconnue par le biais de **dhcp-ignore=tag:!known**.
Les adresses ethernet (mais pas les identifiants clients) peuvent être définies avec des jokers, ainsi par exemple **dhcp-host=00:20:e0:3b:13:*,ignore** demande à Dnsmasq d'ignorer une gamme d'adresses matérielles.
Les adresses matérielles coïncident en principe avec n'importe quel type de réseau (ARP), mais il est possible de les limiter à un seul type ARP en les précédant du type ARP (en Hexadécimal) et de “-”. Ainsi **dhcp-host=06-00:20:e0:3b:13:af,1.2.3.4** coïncidera uniquement avec des adresses matérielles Token-Ring, puisque le type ARP pour une adresse Token-Ring est 6.
Un cas spécial correspond à l'inclusion d'une ou plusieurs adresses matérielles, c-à-d : **dhcp-host=11:22:33:44:55:66,12:34:56:78:90:12,192.168.0.2**. Cela permet à une adresse IP d'être associée à plusieurs adresses matérielles, et donne à dnsmasq la permission d'abandonner un bail DHCP attribué à l'une de ces adresses lorsqu'une autre adresse dans

la liste demande un bail. Ceci est une opération dangereuse qui ne fonctionnera de manière fiable que si une adresse matérielle est active à un moment donné et dnsmasq n'a aucun moyen de s'assurer de cela. Cela est utile, par exemple, pour allouer une adresse IP stable à un laptop qui aurait à la fois une connexion filaire et sans-fil.

Fournir les paramètres pour les hôtes spécifiés à l'aide de DHCP. Il ya beaucoup de possibilités, donc nous allons donner des exemples de chaque. Notez que les adresses IP n'ont pas à être dans la plage indiquée ci-dessus, elles doivent seulement être sur le même réseau. L'ordre des paramètres n'a pas d'importance, il est permis de donner le nom, l'adresse et MAC dans n'importe quel ordre.

dhcp-host=11:22:33:44:55:66,192.168.0.60
: Allouer toujours l'hôte avec l'adresse Ethernet 11:22:33:44:55:66 et l'adresse IP 192.168.0.60

dhcp-host=11:22:33:44:55:66,fred
: Toujours définir le nom de l'hôte “fred” avec l'adresse matérielle 11:22:33:44:55:66

dhcp-host=11:22:33:44:55:66,fred,192.168.0.60,45m
: Toujours donner à l'hôte d'adresse Ethernet 11:22:33:44:55:66 le nom fred et

l'adresse IP 192.168.0.60 et la durée du bail 45 minutes

dhcp-host=11:22:33:44:55:66,12:34:56:78:90:12,192.168.0.60
Donnez à un hôte dont l'adresse Ethernet est 11:22:33:44:55:66 ou 12:34:56:78:90:12 l'adresse IP 192.168.0.60. Dnsmasq suppose que ces deux interfaces Ethernet ne seront jamais en usage en même temps, et donne l'adresse IP au second, même si elle est déjà en cours d'utilisation par le premier. Utile pour les ordinateurs portables avec des adresses filaires et sans fil.

dhcp-host=bert,192.168.0.70,infinite
Donner à la machine qui se nomme “bert” l'adresse IP 192.168.0.70 et un bail infini

dhcp-host=id:01:02:02:04,192.168.0.60
Toujours donner à l'hôte d'identifiant client 01:02:02:04 l'adresse IP de 192.168.0.60

dhcp-host=id:marjorie,192.168.0.60
Toujours donner à

l'hôte d'identifiant “marjorie” l'adresse IP 192.168.0.60

dhcp-host=judge
Permettre de donner l'adresse donnée pour «juge» dans /etc/hosts à une machine présentant le nom de “juge” quand elle demande un bail DHCP.

dhcp-host=11:22:33:44:55:66,ignore
Ne jamais offrir un service DHCP à une machine dont l'adresse Ethernet est 11:22:33:44:55:66

dhcp-host=11:22:33:44:55:66,id:*
Ignorer tout identifiant client présenté par la machine avec l'adresse Ethernet 11:22:33:44:55:66. Utile pour empêcher qu'une machine soit traitée différemment lors de l'exécution sous différents OS ou entre le démarrage PXE et le boot de l'OS

dhcp-host=11:22:33:44:55:66,set:red
Envoyer des options supplémentaires étiquetées comme «red» à la machine avec l'adresse Ethernet 11:22:33:44:55:66

dhcp-host=11:22:33:*:*:*,set:red
: Envoyer des options supplémentaires étiquetées comme «red» à n'importe quelle machine avec une adresse Ethernet commençant par 11:22:33:

dhcp-host=id:00:01:00:01:16:d2:83:fc:92:d4:19:e2:d8:b2, fred, [1234::5]
: Donner une adresse IPv6 fixe et un nom au client de DUID 00:01:00:01:16:d2:83:fc:92:d4:19:e2:d8:b2. Notez que les adresses MAC peuvent pas être utilisés pour identifier les clients DHCPv6. Notez également que les [] autour de l'adresse IPv6 sont obligatoires.

dhcp-option=[tag:<label>,[tag:<label>]][encap:<option>,][vi-encap:<entreprise>,][vendor:[<classe_vendeur>],][<option>|option:<nom d'option>],[<valeur>[,<valeur>]]

: Spécifie des options différentes ou supplémentaires pour des clients DHCP. Par défaut, Dnsmasq envoie un ensemble standard d'options aux clients DHCP : le masque de réseau et l'adresse de broadcast sont les mêmes que pour l'hôte sur lequel tourne Dnsmasq, et le serveur DNS ainsi que la route par défaut prennent comme valeur l'adresse

de la machine sur laquelle tourne Dnsmasq. Si une option de nom de domaine a été définie, son contenu est transmis. Cette option de configuration permet de changer toutes ces valeurs par défaut, ou de spécifier d'autres options. L'option DHCP à transmettre peut être fournie sous forme d'un nombre décimal ou sous la forme “option:<nom d'option>”. Les nombres correspondants aux options sont définis dans la RFC2132 et suivants. Les noms d'options connus par Dnsmasq peuvent être obtenus via “Dnsmasq –help dhcp”. Par exemple, pour définir la route par défaut à 192.168.4.4, il est possible de faire –dhcp-option=3,192.168.4.4 ou –dhcp-option = option:router, 192.168.4.4 ou encore, pour positionner l'adresse du serveur de temps à 192.168.0.4, on peut faire –dhcp-option = 42,192.168.0.4 ou –dhcp-option = option:ntp-server, 192.168.0.4 L'adresse 0.0.0.0 prends ici le sens “d'adresse de la machine sur laquelle tourne Dnsmasq”. Les types de données autorisées sont des adresses IP sous la forme de 4 chiffres séparés par des points, un nombre décimal, une liste de caractères hexadécimaux séparés par des 2 points, ou une chaîne de caractères. Si des labels optionnels sont fournis, alors cette option n'est envoyée qu'aux réseaux dont tous les labels coïncident avec ceux de la requête.

Un traitement spécial est effectué sur les chaînes de caractères fournies pour l'option 119, conformément à la RFC 3397. Les chaînes de caractères ou les adresses IP sous forme de 4 chiffres séparés par des points donnés en arguments de l'option 120 sont traités conformément à la RFC 3361. Les adresses IP sous forme de 4 chiffres séparés par des points suivies par une barre montante “/”, puis une taille de masque sont encodés conformément à la RFC 3442.
Attention : aucun test n'étant fait pour vérifier que des données d'un type adéquat sont envoyées pour un numéro

d'option donné, il est tout à fait possible de persuader Dnsmasq de générer des paquets DHCP illégaux par une utilisation incorrecte de cette option. Lorsque la valeur est un nombre décimal, Dnsmasq doit déterminer la taille des données. Cela est fait en examinant le numéro de l'option et/ou la valeur, mais peut-être évité en rajoutant un suffixe d'une lettre comme suit : b = un octet, s = 2 octets, i = 4 octets. Cela sert essentiellement pour des options encapsulées de classes de vendeurs (voir plus bas), pour lesquelles Dnsmasq ne peut déterminer la taille de la valeur. Les données d'options consistant uniquement de points et de décimaux sont interprétées par Dnsmasq comme des adresses IP, et envoyées comme telles. Pour forcer l'envoi sous forme de chaîne de caractère, il est nécessaire d'utiliser des guillemets doubles. Par exemple, l'utilisation de l'option 66 pour fournir une adresse IP sous la forme d'une chaîne de caractères comme nom de serveur TFTP, il est nécessaire de faire comme suit : –dhcp-option=66,1.2.3.4
Les options encapsulées de classes de vendeurs peuvent-être aussi spécifiées en utilisant dhcp-option : par exemple dhcp-option=vendor:PXEClient,1,0.0.0.0 envoie l'option encapsulée de classe de vendeur “mftp-address=0.0.0.0” à n'importe quel client dont la classe de vendeur correspond à “PXEClient”. La correspondance pour les classes de vendeur s'effectue sur des sous-chaînes de caractères (voir dhcp-vendorclass pour plus de détails). Si une option de classe de vendeur (numéro 60) est envoyée par Dnsmasq, alors cela est utilisé pour sélectionner les options encapsulées, de préférence à toute option envoyée par le client. Il est possible d'omettre complètement une classe de vendeur : –dhcp-option=vendor:,1,0.0.0.0 Dans ce cas l'option encapsulée est toujours envoyée.

Les options peuvent-être encapsulées au sein d'autres options : par exemple –dhcp-option=encap:175, 190, iscsi-client0 enverra l'option 175, au sein de laquelle se trouve l'option 190. Plusieurs options encapsulées avec le même numéro d'option seront correctement combinées au sein d'une seule option encapsulée. Il n'est pas possible de spécifier encap: et vendor: au sein d'une même option dhcp.
La dernière variante pour les options encapsulées est “l'option de Vendeur identifiant le vendeur” (“Vendor-Identifying Vendor Options”) telle que décrite dans le RFC3925. Celles-ci sont spécifiées comme suit : –dhcp-option=vi-encap:2, 10, text Le numéro dans la section vi-encap: est le numéro IANA de l'entreprise servant à identifier cette option.
L'adresse 0.0.0.0 n'est pas traitée de manière particulière lorsque fournie dans une option encapsulée.

strict-order

force dnsmasq à tenter chaque requête avec les serveurs DNS dans l'ordre où ils apparaissent dans le fichier /etc/resolv.conf
(par défaut, dnsmasq envoie les requêtes aux serveurs amont qu'il connaît et tente de favoriser les serveurs qu'il sait actifs)

expand-hosts

pour ajouter automatiquement un nom de domaine aun noms simples dans un fichier hosts (voir aussi domain= ci-dessous)

[1] [3]
,
=“ce domaine n'existe pas”

[2] [4]
,
ne pas choisir un domaine existant, sinon les requêtes pour ce domaine ne seront pas transmises aux serveurs de votre FAI.

[5]
Une valeur de zéro désactive le cache.

[6]
IPv4 ou IPv6

From:
https://nfrappe.fr/doc-0/ - **Documentation du Dr Nicolas Frappé**

Permanent link:
https://nfrappe.fr/doc-0/doku.php?id=logiciel:internet:dnsmasq:config:start4

Last update: **2022/08/13 22:14**