

[tutoriel](#)

DNSSMasq : Utilisation du plug-in DNSSMasq de NetworkManager

- Cf la **traduction** [Utilisation du plugin DNSSMasq du NetworkManager](https://devopstales.github.io/home/networkmanager-dnsmasq/) de l'article <https://devopstales.github.io/home/networkmanager-dnsmasq/> du 24 février 2020
- Voir aussi la **traduction** [clark](http://blog.deadvax.net/2019/01/08/network-manager-and-dnsmasq-plugin/) de l'article <http://blog.deadvax.net/2019/01/08/network-manager-and-dnsmasq-plugin/>

Supposons que vous voulez tester quelque chose dans une configuration de démonstration avec 5 machines. Vous créez les machines virtuelles nécessaires dans votre environnement local, mais vous ne pouvez pas les adresser correctement par leur nom. Avec 5 machines, vous devez également noter leurs adresses IP, ce qui n'est guère pratique.

Heureusement, il existe une solution élégante : le plugin dnsmasq est un joyau caché de NetworkManager.

Avec le plugin dnsmasq, NetworkManager exécute une copie locale et personnalisable de dnsmasq.

Portable domaine : laplab

réseau maison domaine : .homelab
serveur DNS : 172.31.0.1

Deux cas d'utilisation :

sur un ordinateur portable, pour pouvoir ajouter des enregistrements DNS.



à la maison, je veux utiliser le DNS de mon réseau domestique lorsque je suis sur VPN. De nombreux VPN sont configurés pour acheminer uniquement le trafic spécifique via le tunnel VPN et laisser en place ma route par défaut. Cela signifie que je peux accéder à mon imprimante réseau locale tout en me connectant aux ressources sur le VPN. C'est très agréable, car cela signifie que je peux toujours accéder à mon imprimante réseau ou écouter de la musique depuis mon serveur multimédia tout en travaillant. Cependant, la connexion VPN écrase mon resolv.conf avec les serveurs DNS du réseau VPN. Par conséquent, le DNS de mon réseau domestique n'est plus accessible.

Domaines :



- sur l'ordinateur portable : **laplab**
- domaine personnel : **.homelab.**

À la maison, mon serveur DNS est 172.31.0.1.

Mon resolv.conf pointe toujours vers localhost.

Pour les enregistrements définis localement (par exemple pour mon cluster OpenShift), dnsmasq les résout correctement.

En utilisant une configuration dnsmasq plus avancée, je peux transférer de manière sélective des demandes pour certains domaines vers des serveurs spécifiques (par exemple, pour toujours résoudre correctement les hôtes de mon réseau domestique).

Et pour toutes les autres demandes, dnsmasq les transmettra aux serveurs DNS associés à mon réseau actuel ou VPN.

Pré-requis

Première étape

Connaître le **serveur DNS** utilisé :

- sur un Raspberry :

```
...@...:~$ cat /etc/resolv.conf
# Generated by resolvconf
nameserver fd0f:ee:b0::1
```

C'est l'adresse IPV6 de la box

- sous Ubuntu :

```
...@...:~ $ nmcli dev show |grep DNS
IP4.DNS[1]:          192.168.0.254
IP6.DNS[1]:          fd0f:ee:b0::1
```

192.168.0.254 = adresse IPv4 de la box

Autres étapes

(Exemple pour le domaine localhost)

Activez le dnsmasq de NetworkManager en éditant avec les droits d'administration le fichier **/etc/NetworkManager/conf.d/00-use-dnsmasq.conf** pour y écrire :

[/etc/NetworkManager/conf.d/00-use-dnsmasq.conf](#)

```
# Activer le plugin dnsmasq.
[main]
dns=dnsmasq
```

Éditez avec les droits d'administration le fichier **/etc/NetworkManager/dnsmasq.d/localhost.conf** pour y écrire :

[/etc/NetworkManager/dnsmasq.d/localhost.conf](#)

```
# Ce fichier configure le domaine (local) localhost
# et définit des alias et un caractère générique.
address=/localhost/127.0.0.1

# une entrée DNS générique.
#address=/.localhost/127.0.0.1
```

Pour **lire le fichier hosts**, éditez avec les droits d'administration le fichier **/etc/NetworkManager/dnsmasq.d/02-add-hosts.conf** pour y écrire :

[/etc/NetworkManager/dnsmasq.d/02-add-hosts.conf](#)

```
# Par défaut, le plugin ne lit pas à partir de /etc/hosts.
# Cela force le plugin à lire le fichier.
# Cela pourrait pointer vers un autre fichier si vous ne voulez
pas écrire dans le fichier /etc/hosts
addn-hosts=/etc/hosts
```

Redémarrez NetworkManager :

```
...@...:~$ sudo systemctl restart NetworkManager
```

Si tout fonctionne correctement, vous devriez voir que votre resolv.conf pointe vers 127.0.0.1 et qu'un nouveau processus dnsmasq est apparu :

```
...@...:~$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 127.0.0.1
```

la plupart des entrées DNS sont définies dans /etc/hosts. dnsmasq peut alors les récupérer.

J'ai également quelques entrées DNS supplémentaires pour un DNS générique et quelques alias. Les

fichiers dans `dnsmasq.d` pourraient être combinés, mais sont divisés pour mieux montrer l'exemple.

Configurer `.homelab` :

[/etc/NetworkManager/dnsmasq.d/00-homelab.conf](#)

```
# /etc/NetworkManager/dnsmasq.d/00-homelab.conf
#
# transmettre toute demande pour le domaine .homelab
# vers le serveur DNS domestique (172.31.0.1)
server=/homelab/172.31.0.1
```

fichier `/etc/hosts` :

[/etc/hosts](#)

```
# Les noms d'hôte définis ici seront importés grâce au fichier 02-
add-hosts.conf.
127.0.0.1    localhost localhost.localdomain
::1         localhost localhost.localdomain

# Les hôtes sont dans le domaine .laplab,
# comme configuré dans le fichier 01-laplab.conf
192.168.101.120 ose-lap-jumphost.laplab
192.168.101.128 ose-lap-node1.laplab

# Le nom qui ne figure pas dans .laplab sera également récupéré.
# Soyez donc prudent en définissant les éléments ici.
172.31.0.88     overwrite.public.domain.com
```

Redémarrez NetworkManager :

```
...@...:~$ sudo systemctl restart NetworkManager
```

Si tout fonctionne correctement, votre `resolv.conf` pointe vers `127.0.0.1` et un nouveau processus `dnsmasq` est apparu :

```
...@...:~$ ps -ef | grep dnsmasq
dnsmasq  1835  1188  0 08:01 ?          00:00:00 /usr/sbin/dnsmasq --no-
resolv
--keep-in-foreground --no-hosts --bind-interfaces --pid-
file=/var/run/NetworkManager/dnsmasq.pid
--listen-address=127.0.0.1 --cache-size=400 --clear-on-reload --conf-
file=/dev/null
--proxy-dnssec --enable-dbus=org.freedesktop.NetworkManager.dnsmasq
--conf-dir=/etc/NetworkManager/dnsmasq.d
...@...:~$ cat /etc/resolv.conf
# Generated by NetworkManager
```

```
nameserver 127.0.0.1
...@...:~$ host ose-lap-jumphost.laplab
ose-lap-jumphost.laplab has address 192.168.101.120
```

Conclusion

Cette configuration survivra aux redémarrages.

Problèmes connus

- [Comment éviter les conflits entre dnsmasq et systemd-resolved ?](#)

Voir aussi

- **(fr)** <https://www.cedric-augustin.eu/index.php?post/2018/11/29/Connaitre-le-serveur-DNS-utilise-sous-Ubuntu>
- **(en)** <https://devopstales.github.io/home/networkmanager-dnsmasq/>
- **(en)** <https://fedoramagazine.org/using-the-networkmanagers-dnsmasq-plugin/>

Basé sur « *Using the NetworkManager's DNSSMasq plugin* » par Clark Hale.

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

Permanent link:
<https://www.nfrappe.fr/doc-0/doku.php?id=tutoriel:internet:dnsmasq:networkmanager:start>

Last update: **2022/08/13 22:15**

