tutoriel

DNSMasq : Utilisation du plug-in DNSMasq de NetworkManager

Supposons que vous vouliez tester quelque chose dans une installation de démonstration avec 5 machines. Dans votre environnement local, vous pouvez créer les machines virtuelles nécessaires, sans toutefois pouvoir les nommer correctement. Avec 5 machines, il vous faut aussi définir les adresses IP appropriées, ce qui n'est pas très pratique.

Le plugin dnsmasq est un joyau caché de NetworkManager. Avec ce plugin, au lieu d'utiliser le serveur de noms DNS fourni par DHCP, NetworkManager configure une copie locale de dnsmasq qui peut être personnalisée.

Deux cas d'utilisation :

1. Sur l'ordinateur portable, une installation complète d'OpenShift est en cours d'exécution. Pour que cela fonctionne, il faut pouvoir ajouter des entrées DNS.
    1. Pour cela, on pourrait faire tourner un serveur DNS dans une machine virtuelle ou localement, mais cela nécessiterait de modifier le fichier resolv.conf à chaque changement de réseau (et parfois même plus souvent).
2. À la maison, je voudrais continuer à avoir accès aux entrées DNS du réseau domestique pendant que je suis sur le VPN.
    1. Beaucoup de VPN sont configurés pour que seul le trafic lié au réseau VPN soit envoyé à travers le tunnel. L'accès à l'ensemble du réseau local reste possible et le trafic sort en grande partie par la passerelle par défaut.
    2. C'est très pratique, car il est possible d'accéder à l'imprimante réseau ou d'écouter de la musique à partir du serveur multimédia tout en travaillant.Par contre, la connexion VPN écrase le fichier resolv.conf par les serveurs DNS du réseau VPN. Le DNS du réseau domestique n'est donc plus accessible.

Le plugin dnsmasq pour NetworkManager résout ces deux problématiques :

1. **Concernant les entrées DNS locales**, le plugin peut configurer des domaines locaux disponibles quel que soit le réseau auquel on est connecté.
2. **Pour le scénario VPN**, il est possible de configurer dnsmasq pour qu'il transmette les requêtes destinées au domaine du domicile vers le serveur DNS du domicile. Le DNS du VPN est configuré comme une redirection dans dnsmasq plutôt que comme un serveur DNS primaire, ce qui permet de résoudre aussi bien les entrées DNS du réseau VPN que les miennes.

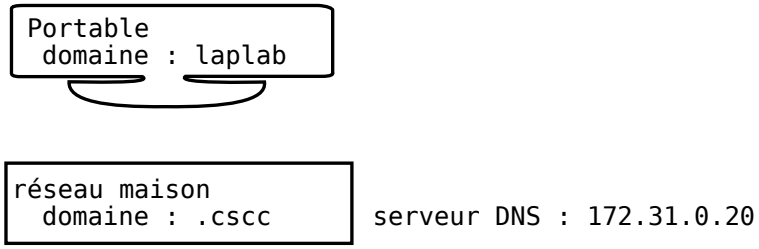
Voici la méthode à suivre pour le configurer dans Fedora 29 : Rappelons que le domaine de l'ordinateur portable se nomme laplab et que le domaine du domicile se nomme .cscc.

Le fichier resolv.conf pointe toujours vers localhost.

Dnsmasq résout correctement les enregistrements définis au niveau local (par exemple pour le cluster OpenShift).

Une configuration plus élaborée de dnsmasq permet de transférer sélectivement les requêtes de certains domaines vers des serveurs spécifiques (par exemple, pour toujours résoudre correctement les hôtes du réseau domestique).

Quant aux autres requêtes, dnsmasq les transmet aux serveurs DNS associés au réseau actuel ou au réseau privé virtuel (VPN).



1. **Domaines** :
   - de l'ordinateur portable : **laplab**
   - de l'ordinateur personnel : **.cscc**.
2. **serveur DNS du domicile** : 172.31.0.20

La plupart des entrées DNS de laplab sont définies dans /etc/hosts. Dnsmasq peut alors les récupérer.

Quelques entrées DNS supplémentaires sont également définies pour un DNS générique ainsi que quelques alias.

Voici les cinq fichiers à mettre en place. NetworkManager utilise un répertoire config. Vous pouvez tout à fait disposer ces fichiers différemment si vous le souhaitez :

- /etc/NetworkManager/conf.d/00-use-dnsmasq.conf
- /etc/NetworkManager/dnsmasq.d/02-add-

hosts.conf

- /etc/NetworkManager/dnsmasq.d/01-laplab.conf
- /etc/NetworkManager/dnsmasq.d/00-cscc.conf
- /etc/hosts

Pré-requis

Connaître le **serveur DNS** utilisé :

- **sur un Raspberry** :

```
pi@framboise4:~$ cat /etc/resolv.conf
# Generated by resolvconf
nameserver fd0f:ee:b0::1
```

fd0f:ee:b0::1 = adresse IPV6 de la box

- **sous Ubuntu** :

```
USER@MACHINE:~$ 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

Première étape : Activation du plugin dnsmasq de NetworkManager

/etc/NetworkManager/conf.d/00-use-dnsmasq.conf

```
#
/etc/NetworkManager/conf.d/00-use-dnsmasq.conf
#
# Ceci active le plugin dnsmasq.
[main]
```

```
dns=dnsmasq
```

Autres étapes

(Exemple pour le domaine localhost)

Les fichiers contenus dans dnsmasq.d pourraient être combinés, mais ils sont séparés pour une meilleure démonstration de l'exemple.

1. **Fichiers dans dnsmasq.d** :
    1. **réseau maison (domaine .cscc, serveur DNS 172.31.0.20)** :

       /etc/NetworkManager/dnsmasq.d/00-cscc.conf

```
#
/etc/NetworkManager
/dnsmasq.d/00-
cscc.conf
#
# DNSmasq
redirigera toute
demande de
résolution de noms
sous le domaine
.cscc
# vers le serveur
DNS domestique
172.31.0.20
server=/cscc/172.31
.0.20
```

    2. **Domaine laplab** :

       /etc/NetworkManager/dnsmasq.d/01-laplab.conf

```
#
/etc/NetworkManager
/dnsmasq.d/01-
```

```
laplab.conf
# Définition du
domaine local
laplab
# ainsi que de
quelques alias et
d'un joker.
local=/laplab/

# Définition d'une
entrée DNS de type
"joker".
address=/.ose.lapla
b/192.168.101.125
# Définition de
quelques noms
d'hôtes.
address=/openshift.
laplab/192.168.101.
120
address=/openshift-
int.laplab/192.168.
101.120
```

3. **Pour lire le fichier hosts** :

   /etc/NetworkManager/dnsmasq.d/02-add-hosts.conf

```
#
/etc/NetworkManager
/dnsmasq.d/02-add-
hosts.conf
# Par défaut, le
plugin ne lit pas
le fichier
/etc/hosts.
# Ces lignes
obligent le plugin
à l'insérer dans le
fichier
# Pour éviter
d'écrire dans le
fichier /etc/hosts,
# il peut pointer
vers un autre
fichier.
addn-
hosts=/etc/hosts
```

Un exemple de fichier 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
192.168.101.12
```

```
0  ose-lap-jumphost ose-lap-jumphost.laplab
192.168.101.121  ose-lap-master1 ose-lap-master1.laplab
192.168.101.122  ose-lap-master2 ose-lap-master2.laplab
192.168.101.123  ose-lap-master3 ose-lap-master3.laplab
192.168.101.125  ose-lap-infnode1 ose-lap-infnode1.laplab
192.168.101.126  ose-lap-infnode2 ose-
```

```
lap-
infnode
2.lapla
b
192.168
.101.12
7  ose-
lap-
infnode
3 ose-
lap-
infnode
3.lapla
b
192.168
.101.12
8  ose-
lap-
node1
ose-
lap-
node1.l
aplab
192.168
.101.12
9  ose-
lap-
node2
ose-
lap-
node2.l
aplab
192.168
.101.13
0  ose-
lap-
node3
ose-
lap-
node3.l
aplab

# Le
nom qui
ne
figure
pas
dans
.laplab
```

```
sera
égaleme
nt
récupér
é.
# Soyez
donc
prudent
en
définis
sant
les
élément
s ici.
172.31.
0.88
overwri
te.publ
ic.doma
in.com
```

2. **Redémarrez NetworkManager** :

```
USER@MACHINE:~$ sudo systemctl
restart NetworkManager
```

3. **Si tout se passe bien**, votre resolv.conf devrait pointer vers 127.0.0.1 et un nouveau processus dnsmasq devrait être actif :

```
USER@MACHINE:~$ 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/dnsm
asq.pid
--listen-address=127.0.0.1 --
cache-size=400 --clear-on-reload
--conf-file=/dev/null
--proxy-dnssec --enable-
dbus=org.freedesktop.NetworkManag
er.dnsmasq
--conf-
dir=/etc/NetworkManager/dnsmasq.d
USER@MACHINE:~$ cat
/etc/resolv.conf
```

```
# Generated by NetworkManager
nameserver 127.0.0.1
USER@MACHINE:~$ host ose-lap-jumphost.laplab
ose-lap-jumphost.laplab has address 192.168.101.120
```

Conclusion

Cette configuration survit aux redémarrages et, d'après ses tests, fonctionne avec presque tous les réseaux et VPN que Clark Hale a essayés.

Problèmes connus

Comment éviter les conflits entre dnsmasq et systemd-resolved ?

Si on installe dnsmasq comme serveur DNS pour un réseau local, dnsmasq écoute sur le port 53 qui est déjà utilisé par systemd-resolved.

Arrêter simplement systemd-resolved puis le redémarrer après l'exécution de dnsmasq, résout ce problème mais il revient après un redémarrage : systemd-resolved est démarré d'abord et dnsmasq ne démarre pas car le port 53 est déjà utilisé.

Comment faire comprendre à systemd-resolved qu'il ne doit pas démarrer l'écoute et donc conserver le port 53 pour une utilisation par dnsmasq ?

Il est plus intéressant de savoir comment les deux services peuvent fonctionner ensemble. Peuvent-ils travailler côte à côte ou ne sont-ils résolus que par systemd si l'on utilise dnsmasq ?

Voici la solution pour (X)Ubuntu 18.04 Bionic

:

1. **Installez dnsmasq**

```
USER@MACHINE:~$ sudo apt
install {dnsmasq,}
```

2. **Désactivez l'écoute sur le port 53** pour systemd-resolved (ne touchez pas à /etc/systemd/resolved.conf, car il peut être écrasé lors de la mise à niveau) :
   1. C**réez le répertoire** :

```
USER@MACHINE:~$ sudo
mkdir
/etc/systemd/resolved.co
nf.d
```

   2. **Créez avec les droits d'administration le fichier** :

      /etc/systemd/resolved.conf.d/noresolved.conf

```
[Resolve]
DNSStubListene
r=no
```

3. **Redémarrez systemd-resolved** :

```
USER@MACHINE:~$ sudo
systemctl restart systemd-
resolved.service
```

4. **Supprimez et recréez /etc/resolv.conf** [1] :

```
USER@MACHINE:~$ sudo rm
/etc/resolv.conf
USER@MACHINE:~$ sudo touch
/etc/resolv.conf
```

5. Pour **désactiver l'écrasement de /etc/resolv.conf par NM, éditez avec les droits d'administration le**

fichier :

/etc/NetworkManager/conf.d/disableresolv.conf

```
[main]
dns=none
```

6. **Redémarrez NetworkManager** :

```
USER@MACHINE:~$ sudo systemctl restart NetworkManager.service
```

7. **Pour que dnsmasq utilise resolv.conf de NM**, éditez avec les droits d'administration le fichier :

/etc/dnsmasq.d/nmresolv.conf

```
resolv-file=/var/run/NetworkManager/resolv.conf
```

8. Créez avec les droits d'administration le fichier **/etc/dnsmasq.d/mondns.conf** pour y écrire vos réglages dnsmasq, par exemple :

/etc/dnsmasq.d/mondomaine.conf

```
address=/pc1.mondomaine/192.168.0.1
address=/framboise.mondomaine/192.168.0.31
address=/framboise4.mondomaine/192.168.0.32
```

9. **redémarrez dnsmasq** :

```
USER@MACHINE:~$ sudo systemctl restart dnsmasq
```

10. **Pour utiliser dnsmasq pour la résolution**, éditez avec les droits d'administration le fichier :

/etc/resolv.conf

```
# Use local dnsmasq for resolving
nameserver 127.0.0.1
```

Voir aussi

- **(fr)** https://www.cedric-augustin.eu/index.php?post/2018/11/29/Connaitre-le-serveur-DNS-utilise-sous-Ubuntu
- **(en)** https://xlark.sdf.org/blog/linuxunix/2019/01/08/network-manager-and-dnsmasq-plug-in.html
- **(en)** Using the NetworkManager’s DNSMasq plugin (27 fev 2020)
- **(en)** https://fedoramagazine.org/using-the-networkmanagers-dnsmasq-plugin/
- **(en)** http://blog.deadvax.net/2019/01/08/network-manager-and-dnsmasq-plug-in/
- **(en)** https://unix.stackexchange.com/questions/304050/how-to-avoid-conflicts-between-dnsmasq-and-systemd-resolved

---

Basé sur « Using the NetworkManager’s DNSMasq plugin|Network Manager and DNSmasq plug-in » par Clark Hale's Blog.

1)

C'est important, car resolv.conf est par défaut un lien symbolique vers /run/systemd/resolve/stub-resolv.conf. Si vous ne supprimez pas le lien symbolique, le fichier sera écrasé par systemd au redémarrage. NetworkManager (NM) vérifie également s'il s'agit d'un lien symbolique pour détecter la configuration de systemd-resolved.