



Network-Manager : configuration du réseau

- Objet : Utilisation de Network-Manager
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Pour gérer vos connexions réseau avec Network-Manager.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-placer](#)
 - Création par  [captfnfab](#) le 03/02/2014
 - Testé par  [captfnfab](#) le 01/04/2014
- Commentaires sur le forum : [ici](#) ¹⁾

Introduction

Network Manager (NM) est un démon de gestion des connexions réseau. Il permet, en principe, de se connecter à internet via un câble ou par connexion WiFi et à tout type de réseau.

Alors que le démon NM tourne en tant que root, il est possible de le configurer en tant qu'utilisateur, par exemple via une icône dans la zone de notification (petit icône près de l'horloge).

Dans la plupart des cas, NM n'a besoin d'aucune configuration. Si votre carte réseau est détectée et son pilote activé, la liste des réseaux doit s'afficher, vous permettant de sélectionner celui sur lequel vous souhaitez vous connecter.

Parfois, certaines options peuvent être grisées, il vous faudra alors creuser du côté des permissions et de consolekit/systemd-logind.

Conflits potentiels

Network-Manager, tout comme [ifupdown](#), wicd, Connection-Manager et dhcpcd5, s'occupe de gérer les connexions réseau, il peut donc **entrer en conflit** avec ces outils s'ils sont présents !

Avec Connection-Manager

Pour le cas où Connection-Manager serait installé, nous vous proposons donc de [le supprimer](#) ainsi :

```
apt remove connman
```

Avec dhcpcd5

Pour le cas où dhcpcd5 serait installé, nous vous proposons donc de [le supprimer](#) ainsi :

```
apt remove dhcpcd5
```

Avec Wicd

Pour le cas où wicd serait installé, nous vous proposons donc de [le supprimer](#) ainsi :

```
apt remove wicd
```

Avec resolvconf

Je ne suis pas sûr du niveau d'incompatibilité entre ces deux là, mais [resolvconf](#) fixant les [DNS](#) et NM²⁾ aussi, je préfère l'enlever :

```
apt remove resolvconf
```

Avec ifupdown

Network-Manager et ifupdown s'entendent très bien entre eux, à condition de ne pas tenter de configurer une même interface avec les deux outils en même temps !

Par défaut, l'interface loopback³⁾ est gérée par ifupdown, ce qui est très bien.

Par défaut, Network-Manager s'occupe uniquement des **interfaces qui ne sont pas mentionnées** dans [le fichier Interfaces](#), ce qui est très bien. 😊

Ce comportement est réglé par le fichier `/etc/NetworkManager/NetworkManager.conf`, consultez-le avec par exemple la commande [less](#) ainsi :

```
less /etc/NetworkManager/NetworkManager.conf
```

Et assurez-vous bien d'y avoir ces lignes :

```
[ifupdown]
managed=false
```

Sinon éditer le fichier [en root](#) pour les ajouter :

```
nano /etc/NetworkManager/NetworkManager.conf
```

En Wifi, si votre connexion est détectée. Si vous avez entré le mot de passe du wifi et que la connexion ne s'établit pas avec la Box (routeur). Vous pouvez ajouter au fichier, *NetworkManager.conf*, les lignes suivantes:

[/etc/NetworkManager/NetworkManager.conf](#)

```
[device]
```

```
wifi.scan-rand-mac-address=no
```

Redémarrez et la connexion devrait s'établir.

Faire en sorte que [le fichier interfaces](#) ne s'occupe que de l'interface loopback⁴⁾.

Pour cela, le fichier `/etc/network/interfaces` ne doit contenir que les lignes suivantes :

[/etc/network/interfaces](#)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

auto lo
iface lo inet loopback
```

Sinon, éditer ce fichier pour le modifier :

```
nano /etc/network/interfaces
```

Installation

L'installation se fait en deux parties.

Premièrement, le serveur Network-Manager :

```
apt update && apt install network-manager
```

Et ensuite l'applet qui sera utilisé par l'utilisateur pour faire la configuration.

En ligne de commande, les utilitaires sont déjà présents dans le paquet network-manager. 😊

Sous KDE

```
apt install network-manager-kde
```

Sous Gnome et autres

```
apt install network-manager-gnome
```

Graphique

Via l'applet GNOME

Si l'applet Network-Manager n'est pas lancé automatiquement, exécutez-le via la commande :

```
nm-applet
```

Pour plus de commodité, vous pouvez demander à votre gestionnaire de fenêtre ou votre environnement de bureau de le lancer automatiquement à chaque connexion.

Via l'applet KDE

Ajoutez l'applet dans KDE

Ligne de commande

En ligne de commande les utilitaires sont déjà présents dans le paquet network-manager. 😊

Voir le man des commandes NetworkManager, nm, nmcli, nm-online et nm-tool.

Interface nmcli

L'interface nmcli⁵⁾ est une commande permettant de créer, modifier, supprimer, activer ou désactiver les interfaces réseaux.

nmcli est très simple à utiliser grâce à l'[auto-complétion](#).

nmcli

Afficher la liste et l'état des connexions réseaux.

La commande **nmcli** peut remplacer les commandes **ip a** ou **ifconfig** :

```
nmcli
```

[retour de la commande nmcli](#)

```
virbr1: connecté to virbr1
    bridge, FE:54:00:78:67:DB, sw, mtu 1500
    inet4 192.168.2.1/24
    inet6 fe80::fc54:ff:fe78:67db/64

enp3s0: connecté to enp3s0
    "Realtek RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller
    (Onboard Ethernet)"
    ethernet (r8169), xx:xx:xx:xx:xx:xx, hw, mtu 1500
    ip4 default
```

```

    inet4 192.168.1.254/24
    inet6 fe80::xxxx:xxxx:xxxx:xxx/64

vnet0: connecté to vnet0
    tun, FE:54:00:78:67:DB, sw, mtu 1500
    master virbr

lo: non-géré
    loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

DNS configuration:
    servers: 212.27.40.241 212.27.40.240
    interface: enp3s0

```

nmcli help

Aide générale :

```
nmcli help
```

```
Usage: nmcli [OPTIONS] OBJECT { COMMAND | help }
```

OPTIONS

-t[erse]	terse output
-p[retty]	pretty output
-m[ode] tabular multiline	output mode
-f[ields] <field1,field2,...> all common	specify fields to output
-e[scape] yes no	escape columns separators in values
-n[ocheck]	don't check nmcli and NetworkManager versions
-a[sk]	ask for missing parameters
-w[ait] <seconds>	set timeout waiting for finishing operations
-v[ersion]	show program version
-h[elp]	print this help

OBJECT

g[eneral]	NetworkManager's general status and operations
n[etworking]	overall networking control
r[adio]	NetworkManager radio switches
c[onnection]	NetworkManager's connections
d[evice]	devices managed by NetworkManager
a[gent]	NetworkManager secret agent or polkit agent

Aide sur une sous-commande

Ici la sous-commande *general*

```
nmcli general help
```

```
Utilisation : nmcli general { COMMAND | help }
```

```
COMMAND := { status | hostname | permissions | logging }
```

```
status
```

```
hostname [<hostname>]
```

```
permissions
```

```
logging [level <log level>] [domains <log domains>]
```

nmcli connection

Dans Network Manager, toutes les interfaces réseaux (sous-commande *device*) sont gérées à travers des profils (sous-commande *connection*).

Afficher la liste des profils existants :

```
nmcli connection
```

NOM	UUID	TYPE	PÉRIPHÉRIQUE
enp3s0	2864a7ef-2c77-42ba-8875-6effe1b58a4a	802-3-ethernet	enp3s0
virbr1	e8bcd63-957a-48df-87c6-ca943dc0c285	bridge	virbr1
vnet0	201edef9-055d-4a3b-ad1e-9611626aee59	tun	vnet0



Le nom du profil peut être différent du nom de l'interface réseau associée.

nmcli connection show

Afficher la liste de tous les profils actifs :

```
nmcli connection show --active
```

NOM	UUID	TYPE	PÉRIPHÉRIQUE
enp3s0	2864a7ef-2c77-42ba-8875-6effe1b58a4a	802-3-ethernet	enp3s0
virbr1	e8bcd63-957a-48df-87c6-ca943dc0c285	bridge	virbr1
vnet0	201edef9-055d-4a3b-ad1e-9611626aee59	tun	vnet0

Obtenir les propriétés détaillées d'un profil particulier :

```
nmcli connection show enp3s0
```

[retour de la commande nmcli connection show enp3s0](#)

```
connection.id: enp3s0
connection.uuid: 2864a7ef-2c77-42ba-8875-6effe1b58a4a
connection.stable-id: --
connection.interface-name: enp3s0
connection.type: 802-3-ethernet
connection.autoconnect: yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (par défaut)
connection.timestamp: 1533040596
connection.read-only: no
connection.permissions:
connection.zone: external
connection.master: --
connection.slave-type: --
connection.autoconnect-slaves: -1 (par défaut)
connection.secondaries:
connection.gateway-ping-timeout: 0
connection.metered: inconnu
connection.lldp: -1 (default)
802-3-ethernet.port: --
802-3-ethernet.speed: 0
802-3-ethernet.duplex: --
802-3-ethernet.auto-negotiate: no
802-3-ethernet.mac-address: --
802-3-ethernet.cloned-mac-address: --
802-3-ethernet.generate-mac-address-mask: --
802-3-ethernet.mac-address-blacklist:
802-3-ethernet.mtu: auto
802-3-ethernet.s390-subchannels:
802-3-ethernet.s390-nettype: --
802-3-ethernet.s390-options:
802-3-ethernet.wake-on-lan: 1 (default)
802-3-ethernet.wake-on-lan-password: --
ipv4.method: manual
ipv4.dns: 212.27.40.241,212.27.40.240
ipv4.dns-search:
ipv4.dns-options: (par défaut)
ipv4.dns-priority: 0
ipv4.addresses: 192.168.1.254/24
ipv4.gateway: 192.168.1.1
ipv4.routes:
ipv4.route-metric: -1
ipv4.ignore-auto-routes: no
ipv4.ignore-auto-dns: no
ipv4.dhcp-client-id: --
ipv4.dhcp-timeout: 0
ipv4.dhcp-send-hostname: yes
ipv4.dhcp-hostname: --
```

```

ipv4.dhcp-fqdn: --
ipv4.never-default: no
ipv4.may-fail: yes
ipv4.dad-timeout: -1 (par défaut)
ipv6.method: auto
ipv6.dns:
ipv6.dns-search:
ipv6.dns-options: (par défaut)
ipv6.dns-priority: 0
ipv6.addresses:
ipv6.gateway: --
ipv6.routes:
ipv6.route-metric: -1
ipv6.ignore-auto-routes: no
ipv6.ignore-auto-dns: no
ipv6.never-default: no
ipv6.may-fail: yes
ipv6.ip6-privacy: -1 (inconnu)
ipv6.addr-gen-mode: stable-privacy
ipv6.dhcp-send-hostname: yes
ipv6.dhcp-hostname: --
ipv6.token: --
proxy.method: none
proxy.browser-only: no
proxy.pac-url: --
proxy.pac-script: --
GENERAL.NOM: enp3s0
GENERAL.UUID:
2864a7ef-2c77-42ba-8875-6effe1b58a4a
GENERAL.PÉRIPHÉRIQUES: enp3s0
GENERAL.ÉTAT: activ  
GENERAL.PAR D  FAUT: oui
GENERAL.DEFAULT6: non
GENERAL.VPN: non
GENERAL.ZONE: external
GENERAL.CHEMIN DBUS:
/org/freedesktop/NetworkManager/ActiveConnection/2
GENERAL.CON-PATH:
/org/freedesktop/NetworkManager/Settings/1
GENERAL.SPEC-OBJECT: --
GENERAL.CHEMIN-MA  TRE: --
IP4.ADRESSE[1]: 192.168.1.254/24
IP4.PASSERELLE: 192.168.1.1
IP4.DNS[1]: 212.27.40.241
IP4.DNS[2]: 212.27.40.240
IP6.ADRESSE[1]: fe80::xxxx:xxxx:xxxx:xxxx/64
IP6.PASSERELLE: --

```



On peut appeler un *profil* par son *nom* ou son *uuid*

nmcli device

Afficher la liste des interfaces réseaux existantes :

```
nmcli device
```

PÉRIPHÉRIQUE	TYPE	ÉTAT	CONNEXION
virbr1	bridge	connecté	virbr1
enp3s0	ethernet	connecté	enp3s0
vnet0	tun	connecté	vnet0
lo	loopback	non-géré	--

nmcli device disconnect

Arrêter et redémarrer une interface réseau.



On arrête l'interface, plutôt que son profil, car ce dernier peut démarrer, tout seul, en cas d'événement particulier

Arrêt d'une interface réseau :

```
nmcli device disconnect enp3s0
```

Lancement d'une interface réseau, on relance plutôt le profil :

```
nmcli connection up enp3s0
```

nmcli connection add

Ajouter un nouveau profil.

On veut, par exemple ici, créer un bridge nommé **virbr1** :

```
nmcli connection add type bridge con-name virbr1 ifname virbr1
```

Dans cet exemple, la commande va créer un nouveau profil nommé **virbr1** (directive *con-name*) associé à l'interface **virbr1** (directive *ifname*).



Dans ce cas particulier, la commande va aussi créer une interface virtuelle, nommée **virbr1**, car elle n'existe pas physiquement.

Ajouter l'interface réseau.



Avant d'ajouter une interface réseau à un bridge, il est conseillé d'arrêter cette



interface sinon, l'interface peut se retrouver associée à plusieurs profils.

On veut, ensuite, ajouter l'interface réseau `enp3s0` à ce bridge :

```
nmcli connection add type bridge-slave con-name enp3s0 ifname enp3s0 master virbr1
```

nmcli connection modify

Modifier un profil.

On veut, par exemple, modifier le paramètre `ipv4.addresses` du profil **virbr1** :

```
nmcli connection modify virbr1 ipv4.addresses 192.168.3.254/24
```



On redémarrera l'interface réseau pour que la modification soit prise en compte.

nmcli connexion edit

Éditer un profil.

On veut éditer le profil **virbr1** :

```
nmcli connexion edit virbr1
```



A partir d'ici, on se retrouve dans l'**interface commande de nmcli**.
Le prompt `nmcli>` doit être affiché !

nmcli prompt

nmcli> print

Afficher les propriétés du profil :

```
nmcli> print
```

```
=====
===
                                Informations de profil de connexion (virbr1)
=====
===
connection.id:                                virbr1
connection.uuid:                             e8bcd63-957a-48df-87c6-ca943dc0c285
```

```
connection.stable-id:      --
connection.interface-name: virbr1
connection.type:           bridge
connection.autoconnect:    yes
connection.autoconnect-priority: 0
connection.autoconnect-retries: -1 (par défaut)
connection.timestamp:      1533041496
connection.read-only:      no
connection.permissions:
connection.zone:           trusted
connection.master:         --
connection.slave-type:     --
connection.autoconnect-slaves: -1 (par défaut)
connection.secondaries:
connection.gateway-ping-timeout: 0
connection.metered:        inconnu
connection.lldp:           -1 (default)
```

```
---
ipv4.method:              shared
ipv4.dns:
ipv4.dns-search:
ipv4.dns-options:         (par défaut)
ipv4.dns-priority:        0
ipv4.addresses:           192.168.2.1/24
ipv4.gateway:             --
ipv4.routes:
ipv4.route-metric:        -1
ipv4.ignore-auto-routes:  no
ipv4.ignore-auto-dns:     no
ipv4.dhcp-client-id:      --
ipv4.dhcp-timeout:         0
ipv4.dhcp-send-hostname:  yes
ipv4.dhcp-hostname:       --
ipv4.dhcp-fqdn:           --
ipv4.never-default:       no
ipv4.may-fail:            yes
ipv4.dad-timeout:         -1 (par défaut)
```

```
---
ipv6.method:              ignore
ipv6.dns:
ipv6.dns-search:
ipv6.dns-options:         (par défaut)
ipv6.dns-priority:        0
ipv6.addresses:
ipv6.gateway:             --
ipv6.routes:
ipv6.route-metric:        -1
ipv6.ignore-auto-routes:  no
ipv6.ignore-auto-dns:     no
```

```

ipv6.never-default:      no
ipv6.may-fail:           yes
ipv6.ip6-privacy:        0 (désactivé)
ipv6.addr-gen-mode:      stable-privacy
ipv6.dhcp-send-hostname: yes
ipv6.dhcp-hostname:      --
ipv6.token:              --
-----

```

```

---
bridge.mac-address:      --
bridge.stp:              no
bridge.priority:          32768
bridge.forward-delay:    15
bridge.hello-time:        2
bridge.max-age:           20
bridge.ageing-time:       300
bridge.multicast-snooping: no
-----

```

```

---
proxy.method:            none
proxy.browser-only:      no
proxy.pac-url:            --
proxy.pac-script:        --
-----

```

describe ipv4.method

Obtenir maintenant une description de la propriété `ipv4.method` :

```
nmcli> describe ipv4.method
```

```

=== [method] ===
[description de la propriété NM]
IP configuration method. NMSettingIP4Config and NMSettingIP6Config both
support "auto", "manual", and "link-local".
See the subclass-specific documentation for other values.
In general, for the "auto" method, properties such as "dns" and "routes"
specify information that is added on to the information returned from
automatic configuration.
The "ignore-auto-routes" and "ignore-auto-dns" properties modify this
behavior.
For methods that imply no upstream network, such as "shared" or "link-
local", these properties must be empty.
For IPv4 method "shared", the IP subnet can be configured by adding one
manual IPv4 address or otherwise 10.42.x.0/24 is chosen.

```

ipv4.method auto

Modifier la propriété ipv4.method :

```
nmcli> set ipv4.method auto
```

nmcli> verify

Vérifier la cohérence des modifications effectuées précédemment :

```
nmcli> verify
```

nmcli> activate

Appliquer immédiatement les modifications :

```
nmcli> activate
```

nmcli> save

Sauver les modifications :

```
nmcli> save
```

nmcli> quit

Quitter l'interface commande nmcli :

```
nmcli> quit
```

Exemples d'utilisation de nmcli










Configuration d'une connexion wifi

Faire un scan manuel des points d'accès disponibles :

```
nmcli device wifi rescan
```

Afficher la liste des points d'accès trouvés :

```
nmcli device wifi list
```

IN-USE	SSID	MODE	CHAN	RATE	SIGNAL	BARS	SECURITY
	FreeWifi_secure	Infra	10	270 Mo/s	100		WPA1 802.1X
	FreeWifi	Infra	3	195 Mo/s	62		--
	Freebox-59C501	Infra	3	195 Mo/s	57		WPA1
	FreeWifi_secure	Infra	3	195 Mo/s	52		WPA2 802.1X
	SFR_8368	Infra	6	130 Mo/s	19		WPA1
	Bbox-78EFB3	Infra	1	54 Mo/s	17		WPA1 WPA2
	Livebox-9a79	Infra	6	130 Mo/s	17		WPA1 WPA2
	orange	Infra	6	130 Mo/s	17		--
	FreeWifi_secure	Infra	11	195 Mo/s	12		WPA2 802.1X

Se connecter à un réseau wifi :

```
nmcli device wifi connect 'SSID' password 'CLÉ_WIFI' ifname
'NOM_DE_L_INTERFACE_WIFI' name 'NOM_DU_PROFIL'
```

Vous pouvez omettre la directive name 'NOM_DU_PROFIL'. Dans ce cas [Network Manager](#) va attribuer un nom arbitraire au profil.

Vous pouvez omettre la directive ifname 'NOM_DE_L_INTERFACE_WIFI' si vous n'avez qu'une interface wifi.



Il semblerait plus logique de créer un nouveau profil wifi avec la commande **nmcli connection**.

Toutefois, la commande dédiée **nmcli device wifi** permet de faire cela plus facilement.

Exemple :

```
nmcli device wifi connect 'FreeWifi_secure' password 'XXXXXXXXXXXXXXXXXX'
ifname 'wlan0' name 'wlan0_FreeWifi_secure'
```

Ou bien, plus simplement :

```
nmcli device wifi connect 'FreeWifi_secure' password 'XXXXXXXXXXXXXXXXXX'
```

Donner un accès au réseau local et à internet à des machines virtuelles

On veut créer un bridge partagé, nommé lxcbr0, sur l'hôte, qui aura comme [adresse ip](#) : 192.168.2.1, sur le réseau 192.168.2.0/24 :

```
nmcli connection add type 'bridge' con-name 'lxcbr0' ifname 'virbr0'
ipv4.method 'shared' ipv4.addresses '192.168.2.1/24'
```

Explication

La commande précédente a créé un nouveau profil de type bridge nommé lxcbr0 (**con-name**)

associé à l'interface réseau `virbr0` (**ifname**).

L'interface réseau `virbr0` n'existe pas physiquement, elle n'est créée que pour l'occasion et donc elle est une *interface virtuelle*.



En pratique, on essaiera de donner le même nom au profil et à son interface réseau.

Les directives `ipv4.method` et `ipv4.addresses` sont optionnelles.

La directive **ipv4.method shared** signifie que le [NAT](#) va être activé pour les invités connectés au bridge.

On peut le vérifier en interrogeant [le pare-feu](#).

Par exemple :

```
iptables -t nat -L
```

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE all  --  192.168.2.0/24          !192.168.2.0/24
```

Par ailleurs, la directive **ipv4.method shared** active aussi le routage `ipv4` :

```
cat /proc/sys/net/ipv4/ip_forward
```

```
1
```

De plus, elle lance `dnsmasq` pour fournir du service [dhcp](#) et [dns](#) aux invités.

Enfin, si on n'utilise pas la directive **ipv4.addresses**, [Network Manager](#) va utiliser le réseau `10.42.x.0/24` par défaut.

Assigner une interface réseau à un vlan

On veut, par exemple, ajouter l'interface nommée `ens3` au `vlan 1` :

```
nmcli connection add type vlan con-name vlan1 id 1 dev ens3
```

Explication

La commande précédente a créé un nouveau profil de type `vlan`, nommé `vlan1` (`con-name`) et utilisant le `vlan id 1` :

```
nmcli connection
```

NOM	UUID	TYPE	PÉRIPHÉRIQUE
<code>ens3</code>	<code>ceaf1df2-113c-46d2-b8b4-e7aa0a2d6b5f</code>	<code>802-3-ethernet</code>	<code>ens3</code>
<code>vlan1</code>	<code>6c2063ea-d3e6-442b-a7c6-5e515c000fd7</code>	<code>vlan</code>	<code>ens3.1</code>

De plus, une interface virtuelle nommée `ens3.1` a été créée pour l'occasion :

```
nmcli device
```

PÉRIPHÉRIQUE	TYPE	ÉTAT	CONNEXION
<code>ens3</code>	ethernet	connecté	<code>ens3</code>
<code>ens3.1</code>	vlan	connecté	<code>vlan1</code>

Lien externe : [Réseau local virtuel](#)

Obtenir un IP fixe et garder l'itinérance

Il est possible de définir un IP fixe avec NetworkManager.

Connaître le routeur

1. Savoir où notre ordinateur se trouve. L'adresse IP fixe choisi par NM en fonction de l'adresse MAC du routeur. Donc il faut que vous connaissiez et preniez note de l'adresse MAC du routeur. Nombreux routeurs se trouvent à l'adresse 192.168.0.1. Accédez à la page dont le titre peut ressembler à ``MAC address`` ou ``Clone MAC``.

Déterminer votre IP fixe

2. Pour faciliter les choses, établissons une connexion filaire `eth0`. Cliquez (bouton de droite) sur l'icône de NM. Cliquez sur "Modification des connexions".



2a. Dans le taquet "Filaire", faites "Ajouter" ou "Modifier" (après avoir choisi la bonne connexion) selon votre désir. Une nouvelle fenêtre s'ouvrira, comportant 4 taquets: "Filaire, Sécurité 802.1x, Paramètres IPv4, Paramètres IPv6". * D'abord dans Filaire: faites dérouler les options de "Adresse MAC du périphérique" (on identifie ici votre carte réseau) et choisissez la votre connexion (`eth0`). Dans la case Adresse MAC clônée, copiez l'adresse MAC de votre routeur (étape 1).



2b. Dans le taquet "Paramètres IPv4", choisissez d'abord le mode "manuel" dans la première case, puis ajoutez votre adresse, son masque de sous-réseau et la passerelle de votre choix. C'est tout. Les autres cases peuvent rester vierges ou vous les complèterez à votre convenance.



Reconnecter

3. Étape essentielle : passer au nouvel IP, le fixe que vous souhaitiez tant. Pour cela, assurez-vous que `/etc/network/interfaces` ne contienne aucune information relativement à l'interface que vous souhaitez utiliser pour obtenir votre IP fixe.

3a. Coupez le contact avec le réseau: cliquez sur “déconnecter” ou déconnectez le fil. NM ne s'en offusquera pas.



3b. Rebrancher le fil. Si vous avez cliqué sur “déconnecter”, retournez avec votre souris sur l'icône NM, vous y verrez vos connections filaires, choisissez la bonne et cliquez dessus.



Résultat final

Le fichier généré se trouvera dans /etc/NetworkManager/system-connections sous le nom que vous avez donné à votre connexion et ressemblera à ceci :

```
[802-3-ethernet]
duplex=full
mac-address=B4:99:BA:58:A4:76
cloned-mac-address=00:17:3F:DD:5D:50

[connection]
id=Bind9 DNS residence Igloolik
uuid=e9e3650d-37d4-4cf6-ade4-438e248446dd
type=802-3-ethernet

[ipv6]
method=auto

[ipv4]
method=manual
dns=192.168.132.240;192.168.132.1;8.8.4.4;8.8.8.8;
addresses1=192.168.132.240;24;192.168.132.1;
```

Configuration manuelle via interfaces

Si vous éprouvez des problèmes avec NM ou bien si votre réseau n'est pas détecté automatiquement, ne paniquez pas. Il reste possible de créer manuellement un réseau, voir :

- [Configuration réseau](#)

Référence :

- Documentation Debian : [Configuration réseau pour l'ordinateur de bureau](#)

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

Network-Manager

³⁾

donnant l'adresse 127.0.0.1

⁴⁾

interface locale seulement

⁵⁾

Network Manager Command Line Interface

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:network-manager>



Last update: **19/06/2023 22:07**