

logiciel, BROUILLON

unbound sur Raspberry Pi : un serveur DNS sur votre réseau local

Pré-requis

Disposer :

- d'un réseau local .
- et d'une connexion à Internet active pour les tests.

Première étape

Placez-vous sur le Raspberry Pi (connectez-vous au Raspberry Pi via ssh).

Installez les paquets unbound et dnsutils en ligne de commande :

```
...@...:~ $ sudo apt-get install -y unbound dnsutils
...@...:~ $ sudo systemctl enable unbound
```

Le paquet **dnsutils** comporte l'utilitaire **dig** avec lequel nous ferons les tests.

Autres étapes

Configuration

Pour ne pas toucher au fichier `/etc/unbound/unbound.conf`, les fichiers de configuration sont placés dans le répertoire **`/etc/unbound/unbound.d`**.

Créez avec les droits d'administration le fichier **`/etc/unbound/unbound.d/local.conf`** :

[`/etc/unbound/unbound.d/local.conf`](#)

```
server:
#   Adresse du serveur DNS Unbound : toutes les interfaces
  interface: 0.0.0.0

#   autoriser le réseau local
  access-control: 192.168.0.0/16 allow

  local-zone: "monlan." static
```

```
local-data: "box.monlan. IN A 192.168.0.254"  
local-data-ptr: "192.168.0.254 box.monlan"
```

- **interface: 0.0.0.0** : Accessible sur toutes les interfaces
- **access-control: 192.168.0.0/16 allow** : Accessible depuis le réseau local
- **local-zone: / local-data:** : Domaines et sous-domaines
Nous avons créé sur le réseau local un nouveau pseudo-domaine **monlan**, et la box est accessible par **box.monlan**.

2. Pour vérifier, utilisons l'utilitaire **dig** :

```
...@...:~ $ dig freebox.monlan
```

retourne l'adresse IP **192.168.0.254**

3. Pour vérifier l'adresse IP :

```
...@...:~ $ dig -x 192.168.0.254
```

retourne **freebox.monlan**

4. Un autre exemple de config :

<https://blog.zenithar.org/post/2015/08/11/installer-resolveur-dns-raspberry-pi/>

Création d'un pseudo-domaine statique

Pour créer le domaine **mondomaine**, ajouter la ligne :

```
local-zone: "mondomaine." static
```

et les lignes précisant les machines correspondantes :

```
local-data: "box.mondomaine. IN A 192.168.0.254"  
local-data-ptr: "192.168.0.254 box.mondomaine"  
local-data: "machine1.mondomaine. IN A 192.168.0.1"  
local-data-ptr: "192.168.0.1 machine1.mondomaine"  
local-data: "machine2.mondomaine. IN A 192.168.0.2"  
local-data-ptr: "192.168.0.2 machine2.mondomaine"  
etc.
```

Création d'un pseudo-domaine dynamique

Ici, nous créons un domaine **dyndom** et tous ses sous-domaines ***.dyndom** : **site1.dyndom**, **site2.dyndom**, etc.

Ajoutez les lignes :

```
local-zone: "dyndom." redirect
```

```
local-data: "dyndom. A 192.168.0.1"
local-data-ptr: "192.168.0.1 dyndom."
```

Dans cet exemple, le domaine **dyndom** et tous ses sous-domaines renvoient sur le serveur d'adresse IP **192.168.0.1**.

local-zone: <zone> <type>

Configure une zone locale.

Le type détermine la réponse à donner s'il n'y a pas de local-data correspondant. Les types utiles sont :

static

S'il y a correspondance avec des données locales, une réponse est donnée.

redirect

réponse à partir des local-data pour la zone. Cela répond à des requêtes pour la zone, et tous les sous-domaines de la zone ayant des local-data pour la zone.

Par défaut, les zones sont de classe IN

local-data: "<resource record string>"

Fournit une correspondance exacte sauf si la zone est configurée en redirect.

local-data-ptr: "IPaddr name"

Correspondance inverse IP → nom



Utilisation du serveur DNS depuis une machine du réseau

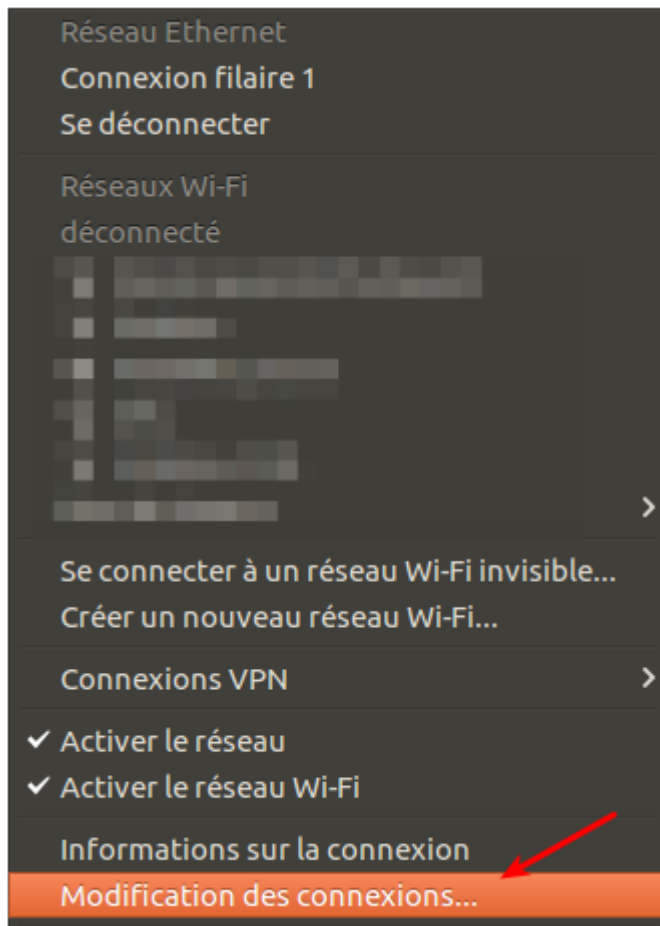
Placez-vous sur la machine à configurer. Nous supposons que l'adresse IP de notre Raspberry Pi est **192.168.0.31**.

Premier cas : sous Linux

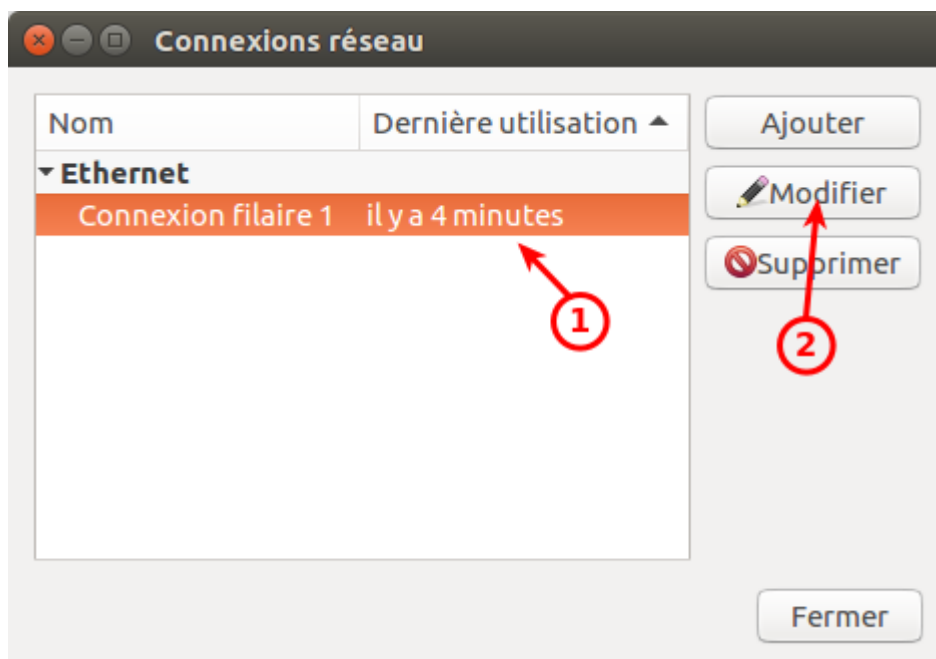
Cliquez en haut sur l'icône du réseau :



Le menu s'ouvre :



Choisissez **Modification des connexions** :



Sélectionnez la connexion à régler

Cliquez sur le bouton Modifier

Modification de Connexion filaire 1

Nom de la connexion : **Connexion filaire 1**

Général | Ethernet | Sécurité 802.1x | DCB | **Paramètres IPv4** | Paramètres IPv6

Méthode : Adresses automatiques uniquement (DHCP)

Adresses

Adresse	Masque de réseau	Passerelle

Ajouter
Supprimer

Serveurs DNS : 192.168.0.31

Domaines de recherche :

ID de client DHCP :

☐ Requier un adressage IPv4 pour que cette connexion fonctionne

Routes...

Annuler Enregistrer

Onglet **Paramètres IPv4**

Méthode : Adresses automatiques uniquement (DHCP)

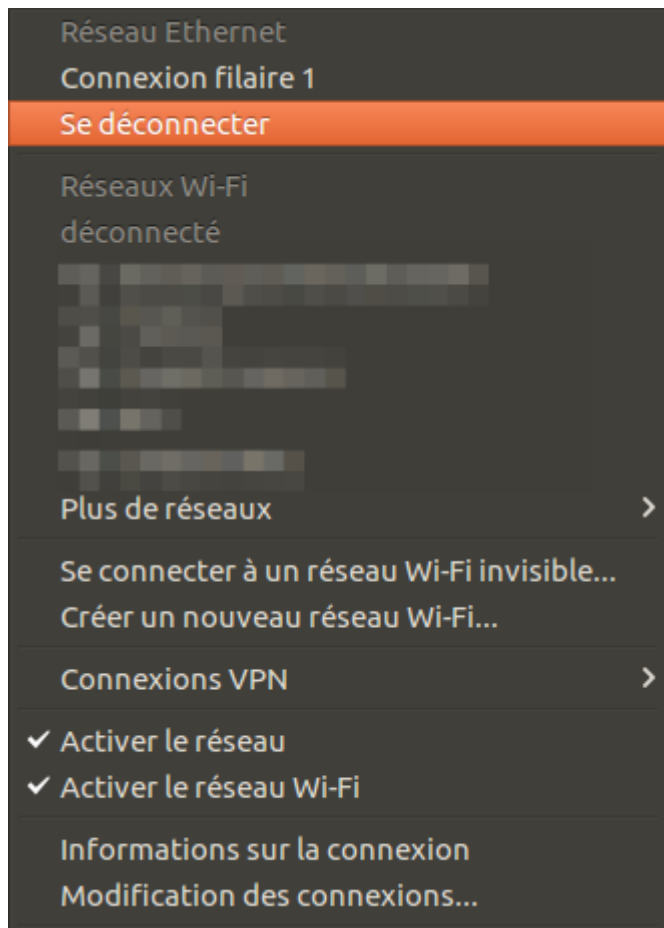
Serveurs DNS : **192.168.0.31** (l'adresse de notre Raspberry Pi)

et bouton Enregistrer

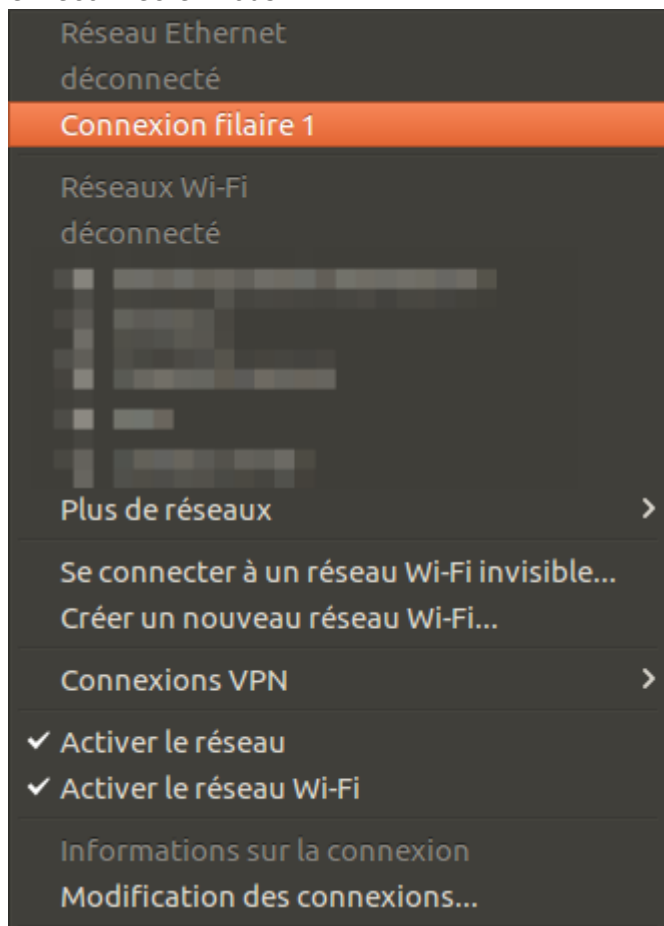
Redémarrez votre réseau :

Cliquez en haut sur l'icône du réseau :

Déconnectez-vous :



et reconnectez-vous :



Sous Windows

Si vous êtes sous Windows, allez dans **Panneau de configuration / Configuration Réseau / Réseaux TCP/IP** et paramétrez les résolveurs DNS primaire et secondaire.

Test du serveur DNS depuis une machine du réseau

Placez-vous sur la machine à configurer.

Installez le paquet  **dnsutils**, ou en ligne de commande :

- `sudo apt-get install -y dnsutils`

Nous supposons que l'adresse IP de notre Raspberry Pi est **192.168.0.31**.

Pour tester la résolution de nom, lancez :

- `dig machine1.mondomaine`

qui retourne l'adresse IP **192.168.0.1** et montre que le serveur DNS utilisé est bien **192.168.0.31**

Pour tester la résolution d'IP, lancez :

- `dig -x 192.168.0.1`

qui retourne le nom **machine1.mondomaine** et montre que le serveur DNS utilisé est bien **192.168.0.31**

Test complet avec namebench

Installez le(s) paquet(s)  **namebench**, ou en ligne de commande :

- `sudo apt-get install -y namebench`

Lancez-le :

- `namebench`

Renseignez les DNS à tester et voyez le résultat.

Conclusion

Problèmes connus

Voir aussi

- (en) [Créer un serveur DNS, HHTP, FTP sur un Raspberry Pi](#)
- (fr) [Créer un serveur DNS, HHTP, FTP sur un Raspberry Pi](#)

Contributeurs principaux : [Jamaïque](#).

Basé sur « [Titre original de l'article](#) » par [Auteur Original].

From:

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

Permanent link:

<https://nfrappe.fr/doc-0/doku.php?id=tutoriel:internet:serveur:rpi:unbound>

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

