Domain Name System

Domain Name System

Le **Domain Name System (DNS)** a un objectif simple mais **crucial** : faire correspondre les **noms de domaine (DNS)** (www.google.com) à des **adresses IP (Internet Protocol)** (142.250.74.196). Et inversement !

Fonctionnement de DNS

Le fonctionnement du DNS est basé sur deux principes clés:

C'est Hiérarchique (en Arbre) : Comme un arbre généalogique, il y a une racine (le domaine racine, noté .) , puis des branches (les domaines de premier niveau ou TLD), puis des sous-branches (domaines de second niveau, etc.)

C'est une Base de données distribuée : Au lieu d'avoir une seule machine qui stocke toutes les informations du monde, les informations sont réparties sur des milliers de serveurs, chacun gérant sa propre **Zone d'autorité**.

Le Processus de Résolution

Quand vous tapez une adresse web, votre ordinateur lance un processus appelé la **résolution DNS** :

- Votre **résolveur DNS** (souvent géré par votre FAI/ISP) reçoit la requête.
- Si le résolveur ne connaît pas l'adresse IP (elle n'est pas dans son cache), il commence par interroger le **serveur racine** (.).
- Le serveur racine le dirige vers le serveur de noms qui fait autorité pour le **Top Level Domain (TLD)** (par exemple, le serveur .com ou .fr).
- Le serveur TLD le dirige vers le serveur de noms qui fait autorité pour le **domaine de second niveau** (par exemple, le serveur labo-linux.com).
- Enfin, le serveur de noms qui fait autorité pour le domaine fournit l'**adresse IP** demandée.
- L'adresse IP est renvoyée à votre ordinateur, qui peut alors se connecter au site web.

Structure et composants du nom de domaine

Pour comprendre comment le DNS fonctionne, il est crucial de bien décomposer un nom de domaine.

Le nom de domaine complet (FQDN)

Un nom de domaine complet, ou **FQDN** pour *Fully Qualified Domain Name*, est le nom qui permet d'identifier de manière **unique** un emplacement sur Internet.

Prenons l'exemple : **www.destination-internet.fr.**

Ce nom est composé de plusieurs parties, lues de droite à gauche, qui suivent la hiérarchie DNS :

- **Le point final (.)** : C'est le **domaine racine** (*root domain*). Bien que souvent omis dans les navigateurs, il est le point de départ de toute résolution. Les serveurs racines gèrent cette zone.
- **L'Extension ou TLD** : fr. C'est le **Top Level Domain** (TLD). Il est résolu depuis les serveurs racines. Les exemples incluent .com, .org, .gouv, .fr.
- **Le Domaine de second niveau** : destination-internet. C'est le nom que vous achetez. Il est résolu à partir des serveurs TLD.
- **Le Préfixe / Sous-domaine** : www. Il s'agit d'un **sous-domaine** ou d'un **nom d'hôte**. Dans l'exemple, www pourrait désigner la partie web du site.

La Hiérarchie DNS

Cette structure illustre la **hiérarchie de l'arbre DNS** :

- **Racine (.)** : Au sommet, géré par les **serveurs racines**.
- **TLD** (Top Level Domain) : Le premier niveau, comme .com ou .uk.
- **Domaines de second niveau** : L'entité (comme mycorp) sous le TLD.
- **Sous-domaines / Hôtes** : Les niveaux inférieurs (comme myhost ou mi.detroit).

Chaque point dans le nom de domaine représente un niveau de la hiérarchie et potentiellement une **délégation d'autorité** à un serveur différent.

Configuration des serveurs DNS avec Bind

Le DNS ne serait rien sans les serveurs qui le font tourner. C'est là que l'implémentation logicielle, comme **Bind** (Berkeley Internet Name Daemon), entre en jeu.

Les Rôles des Serveurs DNS

Un serveur DNS (qui est souvent un logiciel comme Bind) peut avoir plusieurs rôles:

Rôle du Serveur	Fonction Clé
Primaire (Maître)	Détient la **base de données complète et fait autorité** (c'est-à-dire que ses données font foi) pour au moins une zone. Il centralise les modifications.
Secondaire (Esclave)	Fournit de la **redondance** et se **synchronise** sur un serveur primaire. Il récupère une copie de la zone du maître par un processus appelé *Zone Transfer*.
Cache	Garde en mémoire (en cache) les requêtes déjà résolues pour **augmenter les performances**.
Récursif	Transmet les requêtes à d'autres DNS (il fait la démarche complète du processus de résolution) et fournit la réponse finale aux clients.
Hybride	Le plus courant : il combine les rôles, souvent **Cache + Primaire**.

Les Zones d'Autorité

La **zone** est la base de données complète qui définit une sous-partie de l'espace des noms de domaine (comme un TLD, un sous-domaine, ou un nom d'hôte).

- Ces données **font autorité**.
- Un serveur DNS **primaire** est le maître de sa zone.
- Un serveur DNS **secondaire** récupère une copie de la zone via un transfert de zone.

Les Enregistrements (Resource Records – RR)

Une zone est constituée de plusieurs **enregistrements de ressources (RR)** qui définissent les informations de la zone. Ce sont les lignes qui font le *mapping* entre le nom et l'IP (ou d'autres infos).

Option	Explication
SOA	**Start Of Authority**. Les informations sur le domaine, y compris le numéro de série (*serial number*), le *refresh*, le *retry* et l'*expire*.
NS	**Name Server**. Définit les serveurs qui **font autorité** pour la zone (ou pour une délégation).
A	Relie un hôte ou sous-domaine à une adresse **IPv4**.
AAAA	Relie un hôte ou sous-domaine à une adresse **IPv6**.
CNAME	**Canonical Name**. Crée un **alias** ou un raccourci vers une autre ressource (un autre nom).
MX	**Mail Exchanger**. Spécifie l'hôte responsable de la messagerie (MTA) pour le domaine.
PTR	**Pointeur**. Utilisé pour le **DNS inverse** (*Reverse Lookup*), liant une adresse IP à un nom.
TXT	Enregistrement d'un bloc de texte, souvent pour la **validation** (comme SPF ou DKIM pour l'e-mail).

Utilitaire réseaux

NSLOOKUP

```
seed@laptop:~$ nslookup google.fr
Server:         9.9.9.9
Address:        9.9.9.9#53

Non-authoritative answer:
Name:   google.fr
Address: 172.217.20.163
Name:   google.fr
Address: 2a00:1450:4007:809::2003
```

Nslookup (Name Server Lookup) est un outil qui interroge les serveurs DNS pour trouver l'adresse IP correspondant à un nom de domaine, ou inversement.

HOST

```
seed@laptop:~$ host google.fr
google.fr has address 142.250.179.99
google.fr has IPv6 address 2a00:1450:4007:818::2003
google.fr mail is handled by 0 smtp.google.com.
```

Utilitaire simple et rapide en ligne de commande qui permet d'effectuer des recherches **DNS** pour convertir un nom de domaine en adresse IP (et inversement).

DIG

```
seed@laptop:~$ dig google.fr

; <<>> DiG 9.20.15-1~deb13u1-Debian <<>> google.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51485
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;google.fr.                     IN      A

;; ANSWER SECTION:
google.fr.              152     IN      A       172.217.18.195

;; Query time: 16 msec
;; SERVER: 9.9.9.9#53(9.9.9.9) (UDP)
;; WHEN: Mon Dec 08 17:42:03 CET 2025
;; MSG SIZE  rcvd: 54
```

+short elle supprime tout le "bavardage" technique pour ne retourner que l'adresse IP

-x dot-notation (shortcut for reverse lookups)

Dig (Domain Information Groper) est un outil puissant et flexible pour interroger les serveurs DNS. Contrairement à nslookup, il affiche la réponse brute du serveur et détaille le temps de réponse, ce qui est idéal pour le diagnostic approfondi.

Le fichier hosts

```
seed@laptop:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      laptop

192.168.139.254       livebox

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Le fichier /etc/hosts est un fichier texte local qui permet de faire la **correspondance statique** entre des adresses **IP** et des **noms de domaine** ou des noms d'hôtes.

La vérification du fichier /etc/hosts est l'**une des premières étapes** de la résolution d'un nom de domaine sur votre machine, **avant** même d'interroger les serveurs DNS externes.

- Si le nom que vous cherchez est trouvé dans /etc/hosts, l'adresse IP associée est utilisée immédiatement, et aucune requête n'est envoyée aux serveurs DNS.
- Si le nom n'est pas trouvé, le système procède à la résolution classique via les serveurs DNS que nous avons étudiés (en utilisant les configurations que vous avez vérifiées avec resolvconf).

Pourquoi est-il important ?

- **Accès local** : Il est souvent utilisé pour permettre à l'ordinateur d'accéder à ses propres services ou à d'autres machines sur le réseau local en utilisant des noms simples.
- **Tests** : Il est très pratique pour les développeurs ou les administrateurs pour tester un site web ou un service avec un nom de domaine spécifique **avant** que la résolution DNS ne soit mise à jour sur Internet.
- **Blocage** : Il peut être utilisé pour **bloquer** l'accès à certains sites en faisant pointer leur nom de domaine vers une adresse invalide ou la boucle locale (127.0.0.1).

Vérifier sa configuration DNS

```
seed@laptop:~$ nmcli dev show | grep DNS
IP4.DNS[1]:                             192.168.139.254
IP6.DNS[1]:                             2a01:cb19:900a:b700:1adf:26ff:fea7:9350
IP6.DNS[2]:                             fe80::1adf:26ff:fea7:9350
```

Installer resolvconf

```
seed@laptop:~$ sudo apt install resolvconf
[sudo] password for seed:
Installing:
  resolvconf

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 57.1 kB
  Space needed: 190 kB / 828 GB available

Get:1 http://deb.debian.org/debian trixie/main amd64 resolvconf all 1.94 [57.1 kB]
Fetched 57.1 kB in 0s (334 kB/s)
Preconfiguring packages ...
Selecting previously unselected package resolvconf.
(Reading database ... 154286 files and directories currently installed.)
Preparing to unpack .../resolvconf_1.94_all.deb ...
Unpacking resolvconf (1.94) ...
Setting up resolvconf (1.94) ...
Created symlink '/etc/systemd/system/sysinit.target.wants/resolvconf.service' →
'/usr/lib/systemd/system/resolvconf.service'.
Created symlink '/etc/systemd/system/systemd-resolved.service.wants/resolvconf-pull-resolved.path' →
'/usr/lib/systemd/system/resolvconf-pull-resolved
.path'.
Unit /usr/lib/systemd/system/resolvconf-pull-resolved.path is added as a dependency to a non-existent unit
systemd-resolved.service.
Created symlink '/etc/systemd/system/systemd-resolved.service.wants/resolvconf-pull-resolved.service' →
'/usr/lib/systemd/system/resolvconf-pull-resol
ved.service'.
Unit /usr/lib/systemd/system/resolvconf-pull-resolved.service is added as a dependency to a non-existent unit
systemd-resolved.service.
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for resolvconf (1.94) ...
```

Lancer / vérifier le lancement de resolvconf

```
seed@laptop:~$ sudo systemctl start resolvconf.service
seed@laptop:~$ sudo systemctl status resolvconf.service
● resolvconf.service - Nameserver information manager
     Loaded: loaded (/usr/lib/systemd/system/resolvconf.service; enabled; preset: enabled)
     Active: active (exited) since Tue 2025-12-02 17:23:34 CET; 43s ago
 Invocation: 837b4e8c63e448718aabc10eff0db914
       Docs: man:resolvconf(8)
   Main PID: 25898 (code=exited, status=0/SUCCESS)
   Mem peak: 2M
        CPU: 17ms

Dec 02 17:23:34 laptop systemd[1]: Started resolvconf.service - Nameserver information manager.
Dec 02 17:23:34 laptop resolvconf[25903]: /etc/resolvconf/update.d/libc: Warning: /etc/resolv.conf is not a
symbolic link to /run/resolvconf/resolv.c>
```

Modifier le fichier head

```
  GNU nano 8.4                                          /etc/resolvconf/resolv.conf.d/head *

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 9.9.9.9
nameserver 192.168.139.254
```

Le fichier de résolution actuel

```
seed@laptop:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 9.9.9.9
nameserver 192.168.139.254
nameserver 192.168.139.254
nameserver 2a01:cb19:900a:b700:1adf:26ff:fea7:9350
nameserver fe80::1adf:26ff:fea7:9350%enp5s0
search home
```

Ce fichier est celui que la librairie C de votre système utilise **réellement** pour les requêtes DNS. Avec resolvconf, ce fichier est souvent un **lien symbolique** géré par le système. C'est l'endroit où votre résolution 9.9.9.9 (le serveur Quad9 que vous avez ajouté, dans le fichier head) doit apparaître !