جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

المدرسة العليا للأساتذة
ÉCOLE NORMALE SUPÉRIEURE

Serveur DNS

Sous Linux

Livrable

Réalisé par :

- **Rania Diyani**
- **Fatima Zahra Ouandouri**

Encadré par :

Dr. Mehdi Moukhafi

➢ C’est quoi le serveur Dns ?

Un serveur DNS, ou Domain Name System, est un élément essentiel de l'infrastructure d'Internet. Son rôle principal est de traduire les noms de domaine que les humains utilisent pour naviguer sur le web en adresses IP que les ordinateurs et les réseaux utilisent pour communiquer entre eux.

➢ La hiérarchie du serveur Dns :

- **Serveurs Racines (Root Servers) :**
  - Au sommet de la hiérarchie.
  - Gère les requêtes pour les domaines de premier niveau (TLDs).

  Domaines de Premier Niveau (TLD - Top-Level Domains) :
  - Juste en dessous des serveurs racines.
  - Comprend des extensions comme .com, .org, .net, etc.
- **Domaines Secondaires (Second-Level Domains) :**
  - Suivent les TLDs.
  - Incluent des noms de domaine spécifiques tels que google.com, wikipedia.org, etc.
- **Sous-Domaines :**
  - Ajoutent un niveau de spécificité sous les domaines secondaires.
  - Par exemple, mail.google.com, fr.wikipedia.org, etc.

➡ À chaque niveau, les serveurs DNS contiennent des informations spécifiques pour répondre aux requêtes de résolution de noms de domaine. La résolution commence par les serveurs racines et se déplace vers les serveurs appropriés pour trouver l'adresse IP associée à un nom de domaine donné. Cette structure hiérarchique permet une navigation efficace sur Internet en traduisant les noms de domaine en adresses IP.

Fonctionnement du serveur Dns détaillé :

1. Interrogation du Serveur DNS :

- **Requête DNS :** Lorsqu'un périphérique souhaite accéder à un site web via son nom de domaine (ex: www.example.com), il envoie une requête DNS à son serveur DNS local ou au serveur DNS configuré.

2. Recherche de l'Adresse IP :

- **Cache Local :** Le serveur DNS vérifie d'abord son cache local pour voir s'il a déjà enregistré l'association entre ce nom de domaine et son adresse IP. Si l'enregistrement est présent et valide, la résolution est terminée et l'adresse IP est renvoyée directement.
- **Interrogation des Serveurs Racines :** Si l'enregistrement n'est pas dans le cache, le serveur DNS envoie une requête aux serveurs racines pour obtenir l'adresse IP du serveur DNS responsable du TLD (.com, .org, etc.).
- **Serveurs TLD :** Avec cette information, le serveur DNS interroge alors le serveur DNS du TLD spécifique (.com, .org, etc.) pour obtenir l'adresse IP des serveurs DNS autoritaires du domaine recherché (example.com).
- **Serveurs Autoritaires :** Enfin, le serveur DNS contacte les serveurs DNS autoritaires du domaine (example.com) pour obtenir l'adresse IP associée au nom de domaine spécifique (www.example.com).

3. Réponse et Mise en Cache :

- **Réponse DNS :** Une fois que le serveur DNS a obtenu l'adresse IP correspondant au nom de domaine, il la renvoie au périphérique demandeur.
- **Mise en Cache :** Le résultat obtenu est mis en cache localement pour une durée déterminée (définie par le TTL - Time To Live) afin d'accélérer les futures requêtes pour le même nom de domaine.

4. Transmission de l'Adresse IP :

- **Périphérique Client :** Le serveur DNS transmet finalement l'adresse IP associée au nom de domaine au périphérique demandeur, lui permettant ainsi d'établir la connexion avec le serveur correspondant.

➡ Ce processus de résolution DNS est essentiel pour permettre aux utilisateurs d'accéder aux sites web, services et ressources en ligne en utilisant des noms de domaine conviviaux, en les traduisant en adresses IP utilisées pour l'acheminement du trafic sur Internet.

La cache Dns :

La cache DNS est une mémoire temporaire utilisée par un serveur DNS pour stocker les résultats des requêtes de résolution DNS précédentes. Elle conserve temporairement les associations entre les noms de domaine et leurs adresses IP correspondantes, ainsi que d'autres informations liées à la résolution DNS.

Il existe deux types de caches DNS :

- **Cache du côté client :** Les systèmes d'exploitation et les navigateurs web peuvent également conserver des informations DNS en cache pour accélérer les futures requêtes DNS. Ces caches sont situés sur les appareils des utilisateurs.
- **Cache du côté serveur :** Chaque serveur DNS, qu'il soit autoritaire (qui stocke les informations sur des zones spécifiques) ou récursif (qui recherche des informations pour les clients), possède son propre cache DNS. Ce cache stocke temporairement les informations des requêtes DNS récentes.

Principaux Points à Retenir sur la Cache DNS :

- **Stockage Temporaire :** La cache DNS conserve temporairement les résultats des résolutions de noms de domaine. Cela permet de réduire le temps nécessaire pour résoudre les requêtes DNS ultérieures pour les mêmes domaines.
- **Amélioration des Performances :** En stockant localement les résultats des résolutions DNS récentes, la cache permet aux serveurs DNS de répondre plus rapidement aux futures requêtes pour les mêmes domaines.
- **Réduction du Trafic Réseau :** En évitant de refaire les mêmes requêtes pour des domaines fréquemment visités, la cache réduit la charge du réseau en limitant le nombre d'interrogations externes.
- **Gestion des Durées de Vie (TTL) :** Chaque enregistrement dans la cache DNS a un TTL (Time To Live) spécifié, indiquant pendant combien de temps l'information est considérée comme valide avant d'être rafraîchie.
- **Élimination des Données Périmées :** Les enregistrements obsolètes sont supprimés de la cache après l'expiration de leur TTL pour éviter la conservation de données périmées.
- **Optimisation de la Navigation Web :** La cache DNS contribue à rendre la navigation web plus rapide et plus fluide pour les utilisateurs en réduisant le délai nécessaire pour traduire les noms de domaine en adresses IP.

Les enregistrements Dns :

1. Enregistrement A (Address Record) :

- Associe un nom de domaine à une adresse IPv4. Exemple : example.com -> 192.0.2.1

2. Enregistrement AAAA (IPv6 Address Record) :

- Associe un nom de domaine à une adresse IPv6. Exemple : example.com -> 2001:0db8:85a3:0000:0000:8a2e:0370:7334

3. Enregistrement CNAME (Canonical Name Record) :

- Établit un alias pour un nom de domaine, redirigeant vers un autre nom de domaine. Exemple : www -> example.com

4. Enregistrement MX (Mail Exchange Record) :

- Indique les serveurs de messagerie gérant les e-mails pour un domaine. Exemple : example.com -> mail.example.com

5. Enregistrement TXT (Text Record) :

- Contient du texte libre associé à un nom de domaine, utilisé pour diverses informations comme les signatures SPF, DKIM, etc.

6. Enregistrement PTR (Pointer Record) :

- Associe une adresse IP à un nom de domaine. Principalement utilisé dans la résolution inverse. Exemple : 192.0.2.1 -> example.com

7. Enregistrement NS (Name Server Record) :

- Indique les serveurs DNS autoritaires pour un domaine. Exemple : example.com -> ns1.example.com

8. Enregistrement SOA (Start of Authority Record) :

- Contient des informations sur la zone DNS, y compris l'autorité pour le domaine, les délais de rafraîchissement, etc.

9. Enregistrement SRV (Service Record) :

- Utilisé pour identifier les services disponibles dans un réseau, spécifiant le service et le port. Exemple : _http._tcp.example.com -> 80

Le bind9 (Berkeley Internet Name Domain version 9) :

Bind9 (Berkeley Internet Name Domain version 9) est un logiciel open-source de serveur DNS largement utilisé sur Internet. Il est développé et maintenu par l'Internet Systems Consortium (ISC). Bind est l'un des logiciels les plus populaires pour gérer les services DNS sur de nombreux systèmes d'exploitation, en particulier sur les distributions Linux et les systèmes UNIX.

Principales Caractéristiques de Bind9 :

- **Résolution DNS :** Bind9 est capable de résoudre les requêtes DNS, traduisant les noms de domaine en adresses IP et inversement.
- **Serveur Autoritaire :** Il peut agir en tant que serveur DNS autoritaire, répondant aux requêtes DNS pour les domaines dont il a l'autorité.
- **Serveur de Cache :** Bind9 peut fonctionner en tant que serveur de cache DNS, stockant temporairement les informations pour accélérer les futures requêtes.
- **Support IPv6 :** Il est compatible avec IPv6, le protocole de nouvelle génération pour l'adressage IP.
- **Sécurité :** Bind9 propose des fonctionnalités de sécurité avancées pour protéger les serveurs DNS contre les attaques telles que le DNS spoofing, les attaques par déni de service, etc.
- **Zones DNS :** Il permet la configuration de différentes zones DNS pour gérer les enregistrements associés à des domaines spécifiques.
- **Flexibilité de Configuration :** Bind9 offre une grande flexibilité dans la configuration et la personnalisation des services DNS en fonction des besoins spécifiques.

Utilisation de Bind9 :

Bind9 est largement utilisé par les fournisseurs d'accès Internet, les entreprises, les administrateurs système et dans le fonctionnement global d'Internet pour assurer la résolution des noms de domaine. Il peut être installé sur des serveurs dédiés pour gérer les services DNS, qu'il s'agisse de gérer les zones DNS d'un réseau privé ou de fournir des services DNS publics sur Internet.

Installation des packets de Bind9 :

- Sudo dnf install bind

```
[root@localhost-live liveuser]# dnf install bind
Fedora 38 openh264 (From Cisco) - x86_64        137  B/s | 2.5 kB     00:18
Fedora Modular 38 - x86_64                      350 kB/s | 2.8 MB     00:08
Fedora 38 - x86_64 - Updates                    566 kB/s |  36 MB     01:04
Fedora Modular 38 - x86_64 - Updates             88 kB/s | 2.1 MB     00:24
Last metadata expiration check: 0:00:01 ago on Wed 13 Dec 2023 09:53:33 AM EST.
Dependencies resolved.
================================================================================
 Package                 Architecture   Version                    Repository       Size
================================================================================
Installing:
 bind                    x86_64         32:9.18.20-1.fc38          updates         528 k
Upgrading:
 bind-libs               x86_64         32:9.18.20-1.fc38          updates         1.3 M
 bind-license            noarch         32:9.18.20-1.fc38          updates          14 k
 bind-utils              x86_64         32:9.18.20-1.fc38          updates         224 k
Installing weak dependencies:
 bind-dnssec-utils       x86_64         32:9.18.20-1.fc38          updates         148 k

Transaction Summary
================================================================================
Install  2 Packages
Upgrade  3 Packages

Total download size: 2.2 M
Is this ok [y/N]: Y
Downloading Packages:
(1/5): bind-dnssec-utils-9.18.20-1.fc38.x86_64.rpm            76 kB/s | 148 kB     00:01
(2/5): bind-license-9.18.20-1.fc38.noarch.rpm                 48 kB/s |  14 kB     00:00
(3/5): bind-utils-9.18.20-1.fc38.x86_64.rpm                  194 kB/s | 224 kB     00:01
(4/5): bind-9.18.20-1.fc38.x86_64.rpm                        153 kB/s | 528 kB     00:03
(5/5): bind-libs-9.18.20-1.fc38.x86_64.rpm                   200 kB/s | 1.3 MB     00:06
--------------------------------------------------------------------------------
Total                                                        271 kB/s | 2.2 MB     00:08
Fedora 38 - x86_64 - Updates                                 115 kB/s | 1.6 kB     00:00
Importing GPG key 0xEB10B464:
 Userid     : "Fedora (38) <fedora-38-primary@fedoraproject.org>"
 Fingerprint: 6A51 BBAB BA3D 5467 B617 1221 809A 8D7C EB10 B464
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-38-x86_64
Is this ok [y/N]: Y
```

Suite de l'installation :

```
Is this ok [y/N]: Y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : bind-license-32:9.18.20-1.fc38.noarch                  1/8
  Upgrading        : bind-libs-32:9.18.20-1.fc38.x86_64                     2/8
  Upgrading        : bind-utils-32:9.18.20-1.fc38.x86_64                    3/8
  Installing       : bind-dnssec-utils-32:9.18.20-1.fc38.x86_64             4/8
  Running scriptlet: bind-32:9.18.20-1.fc38.x86_64                          5/8
  Installing       : bind-32:9.18.20-1.fc38.x86_64                          5/8
  Running scriptlet: bind-32:9.18.20-1.fc38.x86_64                          5/8
  Cleanup          : bind-utils-32:9.18.13-1.fc38.x86_64                    6/8
  Cleanup          : bind-libs-32:9.18.13-1.fc38.x86_64                     7/8
  Cleanup          : bind-license-32:9.18.13-1.fc38.noarch                  8/8
  Running scriptlet: bind-license-32:9.18.13-1.fc38.noarch                  8/8
  Verifying        : bind-32:9.18.20-1.fc38.x86_64                          1/8
  Verifying        : bind-dnssec-utils-32:9.18.20-1.fc38.x86_64             2/8
  Verifying        : bind-libs-32:9.18.20-1.fc38.x86_64                     3/8
  Verifying        : bind-libs-32:9.18.13-1.fc38.x86_64                     4/8
  Verifying        : bind-license-32:9.18.20-1.fc38.noarch                  5/8
  Verifying        : bind-license-32:9.18.13-1.fc38.noarch                  6/8
  Verifying        : bind-utils-32:9.18.20-1.fc38.x86_64                    7/8
  Verifying        : bind-utils-32:9.18.13-1.fc38.x86_64                    8/8

Upgraded:
  bind-libs-32:9.18.20-1.fc38.x86_64          bind-license-32:9.18.20-1.fc38.noarch
  bind-utils-32:9.18.20-1.fc38.x86_64
Installed:
  bind-32:9.18.20-1.fc38.x86_64          bind-dnssec-utils-32:9.18.20-1.fc38.x86_64

Complete!
[root@localhost-live liveuser]#
```

Verifier si le Bind9 est installé :

- **rpm -qa | grep -i bind :** rechercher les paquets installés sur le système qui contiennent le mot "bind« pour verifier si bind9 est installé.

```
[root@localhost-live liveuser]# rpm -aq | grep -i bind
rpcbind-1.2.6-4.rc2.fc38.x86_64
keybinder3-0.3.2-15.fc38.x86_64
bind-license-9.18.20-1.fc38.noarch
bind-libs-9.18.20-1.fc38.x86_64
bind-utils-9.18.20-1.fc38.x86_64
bind-dnssec-utils-9.18.20-1.fc38.x86_64
bind-9.18.20-1.fc38.x86_64
[root@localhost-live liveuser]#
```

Modifier le hostname :

```
[root@localhost-live liveuser]# hostname
localhost-live
[root@localhost-live liveuser]# hostname serveurDNS
[root@localhost-live liveuser]# hostname
serveurDNS
[root@localhost-live liveuser]#
```

- **Ip a :** affiche des informations détaillées sur les interfaces réseau disponibles sur le système Cela peut aider à confirmer la configuration réseau de la machine où BIND9 est installé et à vérifier si elle a une adresse IP valide pour fonctionner en tant que serveur DNS

```
[root@localhost-live liveuser]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:46:bd:ea brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 81985sec preferred_lft 81985sec
    inet6 fe80::27eb:2660:c63b:ab95/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@localhost-live liveuser]#
```

Fichier named.conf :

Ouvrir le fichier named.conf par l ’éditeur de texte vi pour faire des modification

```
[root@localhost-live liveuser]# vi /etc/named.conf
```

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 127.0.0.1;10.0.2.15; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        secroots-file   "/var/named/data/named.secroots";
        recursing-file  "/var/named/data/named.recursing";
        allow-query     { localhost;10.0.2.15/24;any;  };

        /*
         - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
         - If you are building a RECURSIVE (caching) DNS server, you need to enable
           recursion.
         - If your recursive DNS server has a public IP address, you MUST enable access
           control to limit queries to your legitimate users. Failing to do so will
           cause your server to become part of large scale DNS amplification
           attacks. Implementing BCP38 within your network would greatly
           reduce such attack surface
        */
        recursion yes;

        dnssec-validation yes;

        managed-keys-directory "/var/named/dynamic";
        geoip-directory "/usr/share/GeoIP";

        pid-file "/run/named/named.pid";
        session-keyfile "/run/named/session.key";

-- INSERT --                                                    1,3           Top
```

```
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

- **Vi /etc/named.rfc1912.zones :** Accéder au fichier zone rfc1912 avec l 'éditeur de texte vi pour copier les zones. Ce fichier contient les configurations pour les zones DNS par défaut gérées par le serveur BIND.

```
[root@localhost-live liveuser]# vi /etc/named.conf
[root@localhost-live liveuser]# vi /etc/named.rfc1912.zones
```

```
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
//
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// and https://tools.ietf.org/html/rfc6303
// (c)2007 R W Franks
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// Note: empty-zones-enable yes; option is default.
// If private ranges should be forwarded, add
// disable-empty-zone "."; into options
//

zone "localhost.localdomain" IN {
        type primary;
        file "named.localhost";
        allow-update { none; };
};

zone "localhost" IN {
        type primary;
        file "named.localhost";
        allow-update { none; };
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {
        type primary;
        file "named.loopback";
        allow-update { none; };
};

zone "1.0.0.127.in-addr.arpa" IN {
        type primary;
        file "named.loopback";
        allow-update { none; };
};
```

On se servit des zones qu'on a copié de named.conf pour déclarer notre zone direct

```
zone "." IN {
        type hint;
        file "named.ca";
};
zone "unv.ens.info" IN {
        type primary;
        file "f.unv.ens.info.dir";
        allow-update { none; };
};
```

Créer le fichier f.unv.ens.info.dir puis ouvrir le fichier named.localhost el le copier en f.unv.ens.info.dir

```
[root@localhost-live liveuser]# cd /var/named/
[root@localhost-live named]# touch f.unv.ens.info.dir
[root@localhost-live named]# ls
data     f.unv.ens.info.dir  named.empty      named.loopback
dynamic  named.ca            named.localhost  slaves
[root@localhost-live named]# cat named.localhost
$TTL 1D
@       IN SOA  @ rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      @
        A       127.0.0.1
        AAAA    ::1
[root@localhost-live named]# vi f.unv.ens.info.dir
```

Contenu du fichier qu'on a créée (on a copié ce contenu de named.localhost)

```
[root@localhost-live named]# vi f.unv.ens.info.dir
```

```
$TTL 1D
@       IN SOA  serveurDNS.unv.ens.info.        admin.unv.ens.info. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS              serveurDNS
serveurDNS      IN      A       192.168.56.103
serveurDNS      IN      AAAA    fe80::26ba:1994:b7b0:ea5d
client          IN      A       192.168.56.101
client          IN      AAAA    fe80::c53:1c82:4eb8:6650
client1         IN      A       192.168.56.102
~
```

Ll :Voir les droits d'accès au fichier puis modifier le groupe du fichier qu'on a créé de root en named

```
[root@localhost-live named]# vi f.unv.ens.info.dir
[root@localhost-live named]# vi f.unv.ens.info.dir
[root@localhost-live named]# ll
total 32
drwxrwx---. 2 named named 4096 Nov 15 19:00 data
drwxrwx---. 2 named named 4096 Nov 15 19:00 dynamic
-rw-r--r--. 1 root  root   312 Dec 13 11:01 f.unv.ens.info.dir
-rw-r-----. 1 root  named 3312 Nov 15 19:00 named.ca
-rw-r-----. 1 root  named  152 Nov 15 19:00 named.empty
-rw-r-----. 1 root  named  152 Nov 15 19:00 named.localhost
-rw-r-----. 1 root  named  168 Nov 15 19:00 named.loopback
drwxrwx---. 2 named named 4096 Nov 15 19:00 slaves
[root@localhost-live named]# chgrp named f.unv.ens.info.dir
[root@localhost-live named]# ll
total 32
drwxrwx---. 2 named named 4096 Nov 15 19:00 data
drwxrwx---. 2 named named 4096 Nov 15 19:00 dynamic
-rw-r--r--. 1 root  named  312 Dec 13 11:01 f.unv.ens.info.dir
-rw-r-----. 1 root  named 3312 Nov 15 19:00 named.ca
-rw-r-----. 1 root  named  152 Nov 15 19:00 named.empty
-rw-r-----. 1 root  named  152 Nov 15 19:00 named.localhost
-rw-r-----. 1 root  named  168 Nov 15 19:00 named.loopback
drwxrwx---. 2 named named 4096 Nov 15 19:00 slaves
[root@localhost-live named]#
```

Named-chekconf : pour vérifier la validité et la syntaxe du fichier de configuration named.conf .

systemctl status bind9 : vérification du statut du bind9 et puis il affiche failed alors on doit l'activer .

```
[root@localhost-live named]# named-checkconf /etc/named.conf
[root@localhost-live named]# systemctl status named.service
× named.service - Berkeley Internet Name Domain (DNS)
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: di>
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: failed (Result: exit-code) since Wed 2023-12-13 11:43:14 EST; 18mi>
        CPU: 36ms

Dec 13 11:43:14 localhost-live systemd[1]: Starting named.service - Berkeley In>
Dec 13 11:43:14 localhost-live bash[35843]: /etc/named.conf:19: '10.0.2.15/24':>
Dec 13 11:43:14 localhost-live systemd[1]: named.service: Control process exite>
Dec 13 11:43:14 localhost-live systemd[1]: named.service: Failed with result 'e>
Dec 13 11:43:14 localhost-live systemd[1]: Failed to start named.service - Berk>

[root@localhost-live named]# named-checkzone unv.ens.info f.unv.ens.info.dir
zone unv.ens.info/IN: loaded serial 0
OK
```

Activer le status du bind9 :

```
[root@localhost-live named]# systemctl enable named.service
[root@localhost-live named]# systemctl status named.service
```

systemctl status bind9 : revérification du statut du bind9.

```
[root@localhost-live named]# systemctl status named.service
● named.service - Berkeley Internet Name Domain (DNS)
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: active (running) since Wed 2023-12-13 12:05:49 EST; 7h ago
   Main PID: 36559 (named)
      Tasks: 4 (limit: 2257)
     Memory: 5.7M
        CPU: 1.770s
     CGroup: /system.slice/named.service
             └─36559 /usr/sbin/named -u named -c /etc/named.conf

Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 198>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 198.97.>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 199>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 199.7.9>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 192>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 192.36.>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './DNSKEY/IN': 192>
Dec 13 19:05:50 localhost-live named[36559]: network unreachable resolving './NS/IN': 192.5.5>
Dec 13 19:05:50 localhost-live named[36559]: managed-keys-zone: Unable to fetch DNSKEY set '.>
Dec 13 19:05:50 localhost-live named[36559]: resolver priming query complete: failure
```

Named-chekzone :pour vérifier la syntaxe et la validité d'une zone de domaine spécifique dans un fichier de zone.

```
[root@localhost-live named]# vi /etc/named.conf
[root@localhost-live named]# named-checkconf /etc/named.conf
[root@localhost-live named]# named-checkzone unv.ens.info f.unv.ens.info.dir
zone unv.ens.info/IN: loaded serial 0
OK
[root@localhost-live named]# nslookup
> exit

[root@localhost-live named]# systemctl start named
[root@localhost-live named]# nslookup
> client.unv.ens.info.
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   client.unv.ens.info
Address: 13.248.169.48
Name:   client.unv.ens.info
Address: 76.223.54.146
Name:   client.unv.ens.info
Address: 2c0f:fa18:0:10::df8:a930
Name:   client.unv.ens.info
Address: 2c0f:fa18:0:10::4cdf:3692
> exit

[root@localhost-live named]# systemctl start named
[root@localhost-live named]# nslookup
> client.unv.ens.info.
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   client.unv.ens.info
Address: 192.168.56.101
Name:   client.unv.ens.info
Address: fe80::c53:1c82:4eb8:6650
>
> exit
```

- **Vi /etc/namd.rfc1912.zones :**Accéder au fichier zone **rfc1912** avec l 'éditeur de texte vi pour copier la zone inverse . Ce fichier contient les configurations pour les zones DNS par défaut gérées par le serveur BIND.

```
[root@localhost-live named]# systemctl restart named
[root@localhost-live named]# vi /etc/named.conf
[root@localhost-live named]# cat /etc/named.rfc1912.zones
```

```
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
//
// ISC BIND named zone configuration for zones recommended by
// RFC 1912 section 4.1 : localhost TLDs and address zones
// and https://tools.ietf.org/html/rfc6303
// (c)2007 R W Franks
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// Note: empty-zones-enable yes; option is default.
// If private ranges should be forwarded, add
// disable-empty-zone "."; into options
//

zone "localhost.localdomain" IN {
        type primary;
        file "named.localhost";
        allow-update { none; };
};

zone "localhost" IN {
        type primary;
        file "named.localhost";
        allow-update { none; };
};

zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" IN {
        type primary;
        file "named.loopback";
        allow-update { none; };
};

zone "1.0.0.127.in-addr.arpa" IN {
        type primary;
        file "named.loopback";
        allow-update { none; };
};
```

On se servit des zones qu'on a copié de named.rfc1912.zones pour déclarer notre zone inverse

```
zone "unv.ens.info" IN {
        type primary;
        file "f.unv.ens.info.dir";
        allow-update { none; };
};
zone "2.0.10.in-addr.arpa" IN {
        type primary;
        file "f.unv.ens.info.inv";
};
```

Créer le fichier f.unv.ens.info.inv puis ouvrir le fichier named.localhost el le copier en f.unv.ens.info.inv et lui ajouter les enregistrements

```
[root@localhost-live named]# cp f.unv.ens.info.dir f.unv.ens.info.inv
[root@localhost-live named]# vi f.unv.ens.info.inv
```

```
$TTL 1D
@       IN SOA  serveurDNS.unv.ens.info.        admin.unv.ens.info. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
@       IN      NS              serveurDNS
serveurDNS      IN      A       192.168.56.103
103             IN      PTR     serveurDNS
101             IN      PTR     client
102             IN      PTR     client
```

Voir les droits d'accès au fichier et vérifier la syntaxe et la validité de la zone

```
[root@localhost-live named]# ll f.unv.ens.info.inv
-rw-r--r--. 1 named named 359 Dec 20 19:12 f.unv.ens.info.inv
[root@localhost-live named]# named-checkzone unv.ens.info f.unv.ens.info.inv
zone unv.ens.info/IN: loaded serial 0
OK
```

firewall-cmd --add-service=dns --permanent :ajouter un service spécifique, "dns", à la configuration du pare-feu avec firewall-cmd

firewall-cmd –reload: Cela permettra à notre pare-feu d'autoriser le trafic DNS en fonction des règles prédéfinies ou personnalisées pour le service DNS.

```
[root@localhost-live named]# firewall-cmd --add-service=dns --permanent
success
[root@localhost-live named]# firewall-cmd --reload
success
```

Une fois BIND9 installé, on peut utiliser **nslookup** pour vérifier si notre serveur DNS fonctionne correctement.

```
[root@localhost-live named]# nslookup
> client.unv.ens.info.
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   client.unv.ens.info
Address: 192.168.56.101
Name:   client.unv.ens.info
Address: fe80::c53:1c82:4eb8:6650
> serveurDNS.unv.ens.info.
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   serveurDNS.unv.ens.info
Address: 192.168.56.103
Name:   serveurDNS.unv.ens.info
Address: fe80::26ba:1994:b7b0:ea5d
```

Conclusion

La réalisation de ce projet visant à déployer un serveur DNS sous Linux, avec BIND9 comme pilier central, a permis une plongée profonde dans les arcanes cruciales du fonctionnement des systèmes DNS.

On a débuté notre exposé par une exploration des fondements du réseau informatique, mettant en lumière l'essence même du DNS dans la traduction des noms de domaine en adresses IP, permettant ainsi une navigation fluide sur Internet. La présentation de la hiérarchie du serveur DNS, des enregistrements essentiels et du processus de résolution DNS a jeté les bases théoriques nécessaires.

L'installation et la configuration de BIND9 ont été des étapes clés, offrant une vision pratique du déploiement d'un serveur DNS fonctionnel. Chaque commande exécutée, chaque fichier de configuration modifié a contribué à solidifier les connaissances acquises.

Ce projet ne constitue pas seulement une étape achevée, mais plutôt une porte ouverte vers une expertise accrue dans la gestion des infrastructures réseau. La connaissance du fonctionnement du DNS s'avère être un atout précieux dans un monde où la connectivité est essentielle.

Et cet exposé n'a pas simplement apporté des connaissances techniques, mais a également souligné l'importance cruciale du DNS dans le fonctionnement transparent d'Internet. Comprendre son rôle, sa structure et sa gestion s'avère être une compétence indispensable dans le domaine de l'administration système et réseau.

En conclusion, ce projet a été une immersion enrichissante dans le monde complexe mais fascinant du DNS, offrant une base solide pour des aventures futures dans le domaine des réseaux informatiques.

Les sources

- https://www.cloudflare.com/fr-fr/learning/dns/what-is-dns/#:~:text=Comment%20fonctionne%20le%20DNS%20%3F,voit%20attribuer%20une%20telle%20adresse.
- https://www.malekal.com/la-hierarchie-dns-serveurs-dns-racines-autorite-dns-recursifs-iteratives/
- https://www.youtube.com/watch?v=Y1OHtmNZfIM&t=3150s