Mise en place d'un serveur DNS dans le réseau de GSB

Correction

Le **système DNS** (Domain Name System) a en charge d'établir la correspondance entre un nom pleinement qualifié (FQDN) et une adresse IP. Le système DNS permet à des hôtes du réseau de soumettre des requêtes à un serveur DNS afin d'obtenir l'adresse IP d'un hôte connaissant le nom de cet hôte (par exemple www.google.com → 209.85.229.99). Cette traduction des noms en adresses IP doit toujours être réalisée puisque que seule l'adresse IP permet de communiquer sur le réseau.

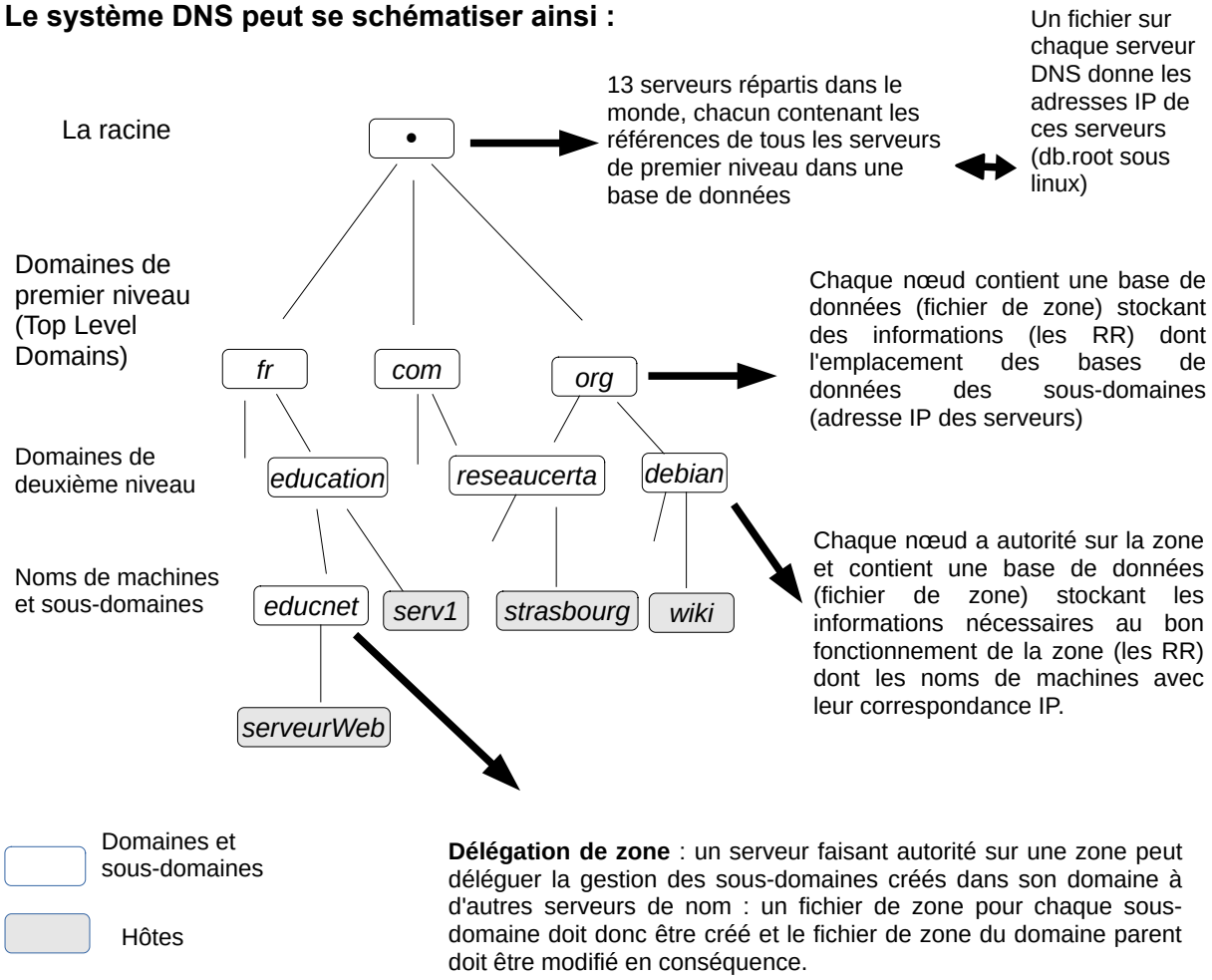
Il s'agit d'un modèle en **arborescence hiérarchique** avec une **gestion décentralisée des données** (chacun étant responsable des données de sa zone).

Le système de noms DNS se présente sous forme d'un arbre inversé avec pour sommet "la racine" et un ensemble de nœuds représentant des domaines identifiés par un label (fr, education.fr, org, com, etc.).

Un serveur de noms particulier s'occupe d'un nœud de l'arborescence ou d'un ensemble de nœuds sur lequel il aura **autorité**. On dit que le serveur gère une **zone d'autorité.** C'est à dire qu'il gérera l'attribution des noms et résoudra les noms via **une base de données** (matérialisée par ce qu'on appelle un **fichier de zone**) distincte pour chaque nœud. Chaque information élémentaire de la base de données DNS est un objet appelé "*resource record*" (RR).

Un nœud peut contenir aussi bien des domaines que des noms de machines.

Le système DNS peut se schématiser ainsi :



Dans notre exemple, quel serveur va résoudre le nom d'hôte pleinement qualifié serveurWeb.educnet.education.fr ?

Le serveur racine ne sait pas où se trouve cet hôte, par contre il possède un enregistrement pour le **domaine "fr".**

La **zone "fr"** est aussi restreinte au nœud correspondant. Son fichier de zone inclut donc des informations sur les délégations de gestion du reste du domaine dont le domaine "education".

Le fichier de zone "education.fr" peut éventuellement posséder un enregistrement pour cet hôte car il est possible qu'une zone d'autorité comprenne un domaine et un sous-domaine. Mais nous supposerons ici que la gestion des noms a été déléguée. Le fichier de zone correspondant contient donc les informations nécessaires pour résoudre des noms dans cette zone (tel que serv1.education.fr) et les informations sur la délégation de la zone "educnet.education.fr".

Le **fichier de zone "educnet.education.fr"** possède l'information concernant l'hôte "serveurWeb" et pourra ainsi résoudre le nom **serveurWeb.educnet.education.fr**.

Il existe deux modes de résolution de noms : le mode récursif et le mode itératif :

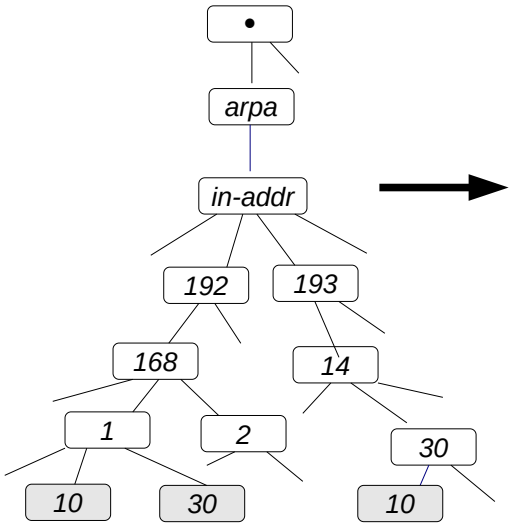
Dans le mode récursif, le client envoie une requête au serveur DNS qui renvoie la réponse complète au client après avoir lui-même éventuellement interrogé d'autres serveurs de noms (s'il n'a pas la réponse en cache et s'il n'est pas autoritaire pour la zone).

Dans le mode itératif, le client envoie une requête au serveur DNS qui renvoie la réponse complète s'il est autoritaire pour la zone concernée mais une réponse partielle dans le cas contraire qui redirige le demandeur vers un autre serveur DNS afin qu'il poursuivre lui-même la résolution et ainsi de suite jusqu'à l'obtention de la réponse complète.

Pour de raisons de performance et de sécurité, il est conseillé de configurer les serveurs de noms pour qu'ils n'acceptent les requêtes en mode récursif que pour les machines de la zone pour laquelle ils sont autoritaires.

La zone in-addr.arpa (zone reverse)

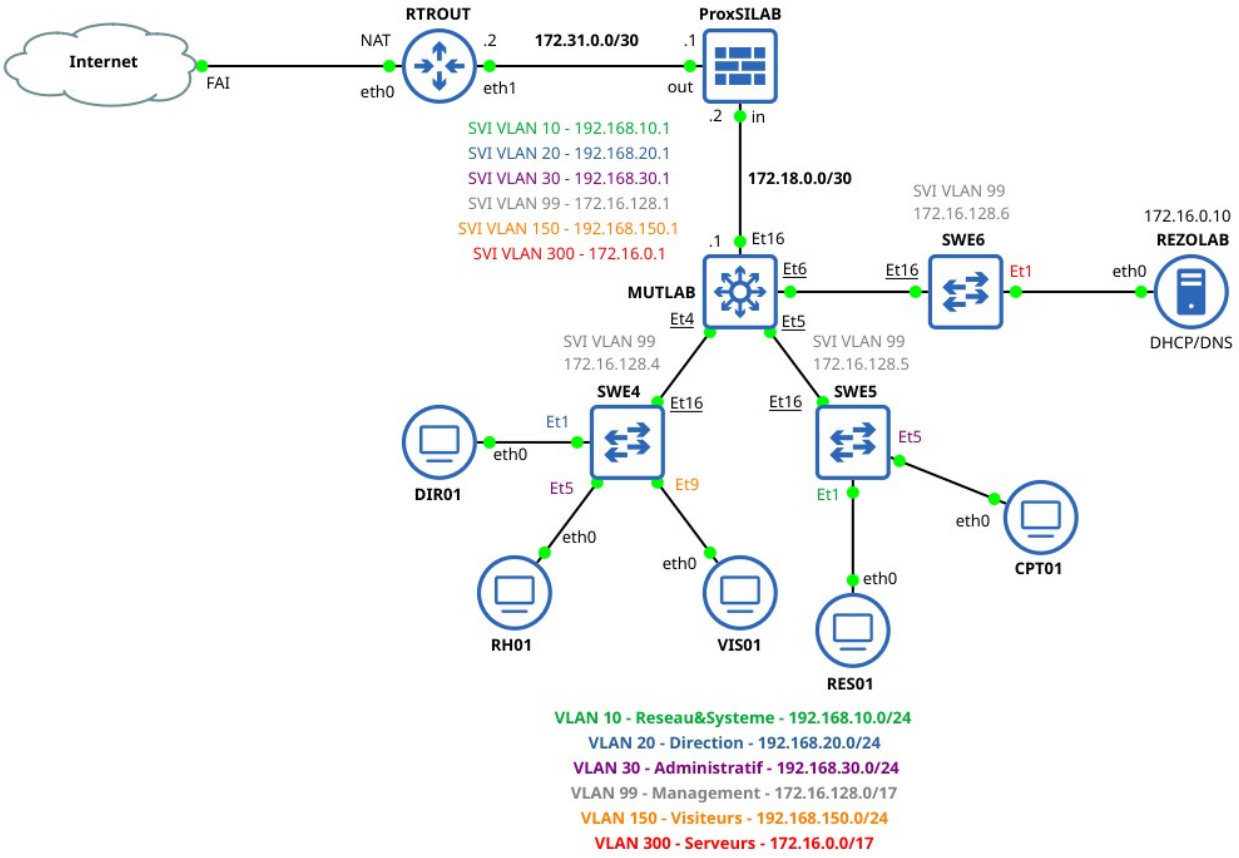
Le domaine in-addr.arpa est un domaine spécial chargé de réaliser les recherches inversées, c'est-à-dire retrouver un nom en connaissant l'adresse IP.



Zone inverse pour chaque réseau dans le domaine in-addr.arpa.
Par exemple, la zone de recherche inverse pour le reseau 192.168.1.0 dans le domaine sera 1.168.192.in-addr.arpa. Cette zone devra répondre pour toutes les adresses déclarées dans la tranche 192.168.1.0 à 192.168.1.254. On inscrira dans cette zone tous les noeuds du réseau pour lesquels on désire que la résolution inverse fonctionne.

Des exemples de fichiers de zone sont donnés dans le document 1.

Schéma réseau du TP



Le service DNS BIND sera installé sur **REZOLAB**. Les principaux fichiers de configuration sont décris dans le document 1.

Installation du serveur DNS sur REZOLAB

Après avoir pris connaissance du document 1, répondez aux questions suivantes :

Questions

1) Par défaut, le service DNS BIND est-il configuré pour répondre à des clients DNS distants ?
   Non. Il est en attente de requêtes DNS sur l'adresse IP 127.0.0.1 :
   `listen-on port 53 { 127.0.0.1; };`
   De plus seules les requêtes en provenance du localhost sont autorisées :
   `allow-query { localhost; };`
2) Que contient le fichier `named.ca` et quel est son utilité ?
   Le fichier `named.ca` contient la liste des 13 serveurs DNS racines. Ils sont interrogés, par défaut, lorsque le serveur DNS ne trouve pas de réponse à une requête DNS.

3) Donnez les chemins absolus des fichiers de zone `named.localhost` et `named.loopback`.
`/var/named/named.localhost` et `/var/named/named.loopback`

En vous aidant du document 2 :

➔ Installez sur **REZOLAB** le serveur DNS BIND ainsi que les outils qui lui sont liés.

```
[root@rezolab ~]# dnf install bind bind-utils
```

Remarque

Un nouveau groupe ainsi qu'un utilisateur système *named* sont créés.

Configuration de REZOLAB comme serveur DNS de mise en cache

En vous aidant des documents 1 et 2 :

➔ Configurez le service BIND de **REZOLAB** en respectant ces consignes :

- Il doit pouvoir répondre, non seulement à lui-même, mais aussi aux clients DNS du réseau local (hôtes des réseaux 192.168.0.0/16 et 172.16.0.0/17). Il acceptera les requêtes DNS récursives.
- Le serveur DNS du FAI 100.64.122.1 sera relais DNS (forwarder). Il se chargera de la résolution DNS itérative lorsque **REZOLAB** ne saura pas résoudre un nom. La validation DNSSEC devra être désactivée, le serveur DNS du FAI ne la prenant pas en charge.

```
[root@rezolab etc]# cat /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; 172.16.0.10; };
      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; 192.168.0.0/16; 172.16.0.0/17; };

      /*
```

```
       - 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;
      allow-recursion { localhost; 192.168.0.0/16; 172.16.0.0/17; };

      forwarders { 100.64.122.1 ;};

      dnssec-validation no;

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

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

      /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
      include "/etc/crypto-policies/back-ends/bind.config";
};

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";
```

➔ Vérifiez que votre serveur est correctement configuré syntaxiquement puis lancez et activez le démarrage automatique du service `named`.

```
[root@rezolab ~]# named-checkconf
[root@rezolab ~]#systemctl enable --now named
```

Configuration des résolveurs (clients) DNS

➜ Vérifiez que **REZOLAB** peut interroger son service DNS et résoudre le nom *www.google.fr* avec la commande `dig @locahost`

```
[root@rezolab etc]# dig @localhost www.google.fr

; <<>> DiG 9.16.23-RH <<>> @localhost www.google.fr
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9378
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 7cbd3a48f20c5139010000006419be6ada43dd3ce433ec03 (good)
;; QUESTION SECTION:
;www.google.fr.                 IN      A

;; ANSWER SECTION:
www.google.fr.          252     IN      A       142.251.37.227

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Tue Mar 21 14:25:46 UTC 2023
;; MSG SIZE  rcvd: 86
```

➜ Modifiez les paramètres IP de **REZOLAB** afin qu'il interroge systématiquement son propre service DNS pour la résolution de noms.

```
[root@rezolab ~]# nmtui
...
   ┌──────────────────────────────┤ Edit Connection ├──────────────────────────────┐
   │                                                                              ↑│
   │         Profile name System eth0_____________________________                ▮│
   │               Device eth0 (0C:B4:DC:83:00:00)_________________               ▒│
   │                                                                              ▒│
   │ = ETHERNET                                                    <Show>         ▒│
   │                                                                              ▒│
   │ ╤ IPv4 CONFIGURATION <Manual>                                 <Hide>         ▒│
   │ │          Addresses 172.16.0.10/17___________ <Remove>                      ▒│
   │ │                    <Add...>                                                ▒│
   │ │            Gateway 172.16.0.1_______________                               ▒│
   │ │        DNS servers 127.0.0.1________________ <Remove>                      ▒│
   │ │                    <Add...>                                                ▒│
   │ │     Search domains <Add...>                                                ▒│
   │ │                                                                            ▒│
   │ │            Routing (No custom routes) <Edit...>                            ▒│
   │ │ [ ] Never use this network for default route                               ▒│
  ...
```

➜ Modifiez votre serveur DHCP afin qu'il distribue comme serveur DNS à interroger par les clients l'IP de **REZOLAB**. N'oubliez pas de redémarrer le service `dhcpd`

```
[root@rezolab ~]# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp-server/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#

# Serveur DNS
option domain-name-servers 172.16.0.10;
…

[root@rezolab ~]#systemctl restart dhcpd
```

➜ Renouvelez les baux DHCP des clients et vérifiez avec la commande `dig` que la résolution du nom *www.google.fr* fonctionne. Vous vous assurerez, grâce au résultat de la commande, que le serveur interrogé est bien **REZOLAB**.

```
root@RES01:/# dig www.google.fr

; <<>> DiG 9.16.33-Debian <<>> www.google.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29800
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: d33d6033f89f135e010000006419c001aa0eed73c3ddf324 (good)
;; QUESTION SECTION:
;www.google.fr.                 IN      A

;; ANSWER SECTION:
www.google.fr.          118     IN      A       142.250.201.163

;; Query time: 15 msec
;; SERVER: 172.16.0.10#53(172.16.0.10)
;; WHEN: Tue Mar 21 14:32:33 UTC 2023
;; MSG SIZE  rcvd: 86
```

Configuration de REZOLAB comme serveur DNS autoritaire de la zone *gsb.intra*

En vous aidant des documents 1 et 3 :

➜ Configurez une zone nommée *gsb.intra*. Vous devez créer les enregistrements A des hôtes :
- rezolab → 172.16.0.10
- mutlab → 172.16.128.1

- swe4 → 172.16.128.4
- swe5 → 172.16.128.5
- swe6 → 172.16.128.6
- proxsilab → 172.18.0.2
- rtrout → 172.31.0.1

Ajout à `/etc/named.conf` de la zone *gsb.intra* :

```
zone "gsb.intra" {
    type master;
    file "gsb.intra.zone";
    allow-query { any; };
    allow-transfer { none; };
};
```

Création du fichier de zone `/var/named/gsb.intra.zone` :

```
[root@rezolab ~]# cat /var/named/gsb.intra.zone
$TTL 8h
@           IN SOA      rezolab.gsb.intra. hostmaster.gsb.intra. (
                              2023032101 ; serial number
                              1d         ; refresh period
                              3h         ; retry period
                              3d         ; expire time
                              3h )       ; minimum TTL

            IN NS   rezolab.gsb.intra.

rezolab           IN    A     172.16.0.10
mutlab            IN    A     172.16.128.1
swe4        IN    A     172.16.128.4
swe5        IN    A     172.16.128.5
swe6        IN    A     172.16.128.6
proxsilab   IN    A     172.18.0.2
rtrout            IN    A     172.31.0.1



[root@rezolab ~]# chown root:named /var/named/gsb.intra.zone
[root@rezolab ~]# chmod 640 /var/named/gsb.intra.zone
```

➔ Vérifiez que votre serveur est correctement configuré syntaxiquement et n'oubliez pas de redémarrer le daemon named afin de prendre en compte les modifications.

```
[root@rezolab ~]# named-checkzone gsb.intra /var/named/gsb.intra.zone
zone gsb.intra/IN: loaded serial 2023032101
OK
[root@rezolab ~]# systemctl reload named
```

➔ Testez la résolution de noms dans le domaine *gsb.intra* avec la commande `host`

```
root@CPT01:/# host rezolab.gsb.intra
rezolab.gsb.intra has address 172.16.0.10
root@CPT01:/# host mutlab.gsb.intra
mutlab.gsb.intra has address 172.16.128.1
root@CPT01:/# host swe4.gsb.intra
swe4.gsb.intra has address 172.16.128.4
root@CPT01:/# host swe5.gsb.intra
swe5.gsb.intra has address 172.16.128.5
root@CPT01:/# host swe6.gsb.intra
swe6.gsb.intra has address 172.16.128.6
root@CPT01:/# host proxsilab.gsb.intra
proxsilab.gsb.intra has address 172.18.0.2
root@CPT01:/# host rtrout.gsb.intra
rtrout.gsb.intra has address 172.31.0.1
```

Configuration de REZOLAB comme serveur DNS autoritaire de la zone inversée 172.in-addr.arpa

En vous aidant des documents 1 et 4 :

➔ Configurez la zone inversée *172.in-addr.arpa*. Vous devez créer les enregistrements PTR des hôtes :

- 10.0.16.172.in-addr.arpa → rezolab
- 1.128.16.172.in-addr.arpa → mutlab
- 4.128.16.172.in-addr.arpa → swe4
- 5.128.16.172.in-addr.arpa → swe5
- 6.128.16.172.in-addr.arpa → swe6
- 2.0.18.172.in-addr.arpa → proxsilab
- 1.0.31.172.in-addr.arpa → rtrout

Par défaut B IND charge automatiquement un certain nombre de zones vides et notamment *16.172.int-addr.arpa*, *18.172.int-addr.arpa* et *31.172.int-addr.arpa*. Pour que notre zone *172.in-addr.arpa* fonctionne, il faut désactiver ses zones vides avec le paramètre `empty-zones-enable no;`

Ajout à `/etc/named.conf` de la zone *172.in-addr.arpa* :

```
options {
…
      forwarders { 100.64.122.1 ;};
      empty-zones-enable no;
…
} ;

zone "172.in-addr.arpa" {
    type master;
    file "172.in-addr.arpa.zone";
    allow-query { any; };
    allow-transfer { none; };
```

```
};
```

Et création du fichier de zone */var/named/172.in-addr.arpa.zone* :

```
[root@rezolab ~]# cat /var/named/172.in-addr.arpa.zone
$TTL 8h
@           IN SOA      rezolab.gsb.intra. hostmaster.gsb.intra. (
                              2023032101 ; serial number
                              1d         ; refresh period
                              3h         ; retry period
                              3d         ; expire time
                              3h )       ; minimum TTL

            IN NS   rezolab.gsb.intra.

10.0.16     IN PTR      rezolab.gsb.intra.
1.128.16    IN PTR      mutlab.gsb.intra.
4.128.16    IN PTR      swe4.gsb.intra.
5.128.16    IN PTR      swe5.gsb.intra.
6.128.16    IN PTR      swe6.gsb.intra.
2.0.18      IN PTR      proxsilab.gsb.intra.
1.0.31      IN PTR      rtrout.gsb.intra.

[root@rezolab ~]# chown root:named /var/named/172.in-addr.arpa.zone
[root@rezolab ~]# chmod 640 /var/named/172.in-addr.arpa.zone
```

➔ Vérifiez que votre serveur est correctement configuré syntaxiquement et n'oubliez pas de redémarrer le daemon named afin de prendre en compte les modifications.

```
[root@rezolab ~]# named-checkzone 172.in-addr.arpa /var/named/172.in-
addr.arpa.zone
zone 172.in-addr.arpa/IN: loaded serial 2023032101
OK
[root@rezolab ~]# systemctl restart named
```

➔ Testez la résolution inverse dans la zone *172.in-addr.arpa*.

```
root@CPT01:/# host 172.16.0.10
10.0.16.172.in-addr.arpa domain name pointer rezolab.gsb.intra.
root@CPT01:/# host 172.16.128.1
1.128.16.172.in-addr.arpa domain name pointer mutlab.gsb.intra.
root@CPT01:/# host 172.16.128.4
4.128.16.172.in-addr.arpa domain name pointer swe4.gsb.intra.
root@CPT01:/# host 172.16.128.5
5.128.16.172.in-addr.arpa domain name pointer swe5.gsb.intra.
root@CPT01:/# host 172.16.128.6
6.128.16.172.in-addr.arpa domain name pointer swe6.gsb.intra.
root@CPT01:/# host 172.18.0.2
2.0.18.172.in-addr.arpa domain name pointer proxsilab.gsb.intra.
root@CPT01:/# host 172.31.0.1
1.0.31.172.in-addr.arpa domain name pointer rtrout.gsb.intra.
```

Documentation

Document 1 - Les fichiers de configuration de BIND

Fichier `/etc/named.conf` :

```
[root@rezolab named]# cat /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; };
       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; };

       /*
        - 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";

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

Par défaut le service bind est en attente de requêtes sur le port 53 de l'**adresse locale uniquement**

Répertoire de travail où se trouvent les fichiers de zone

Seules les requêtes provenant du serveur lui-même sont autorisées

Le DNS est autorisé à faire de la résolution récursive. Cette option peut être limitée en portée par l'option allow-recursion

```
        /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
        include "/etc/crypto-policies/back-ends/bind.config";
};


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";
```

"hint" est le type pour le fichier des serveurs root, qui ont autorité pour la zone point ("."). Le fichier /var/named/named.ca contient la liste des différents serveurs root avec leurs adresses IP.

Il est possible d'utiliser des serveur DNS relais plutôt qu'interroger directement des serveurs racines en utlisant l'option forwarders

Fichier utilisé pour la déclaration des zones imposées par la RFC1912

Extrait du fichier `/etc/named.rfc1912.zones` :

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

Nom de la zone primaire (« master »)

Chemin donné en relatif car ce fichier est dans le répertoire de stockage défini par la directive "directory" du fichier named.conf

Déclaration d'une zone inversée

Fichier `named.localhost` :

```
$TTL 3H
@       IN SOA  @ rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
        NS      @
        A       127.0.0.1
        AAAA    ::1
```

Durée de vie par défaut d'un enregistrement de ressource (RR) : ici 3H

informations utiles pour des serveurs secondaires

Outre la variable **$TTL** (obligatoire pour bind9), on peut spécifier au début du fichier les variables suivantes :

- **$ORIGIN** : Pour définir le nom de domaine à ajouter pour reconstituer le FQDN (est positionné par défaut au nom du domaine que le fichier de zone décrit).
- **$INCLUDE** : Pour indiquer le chemin d'accès d'un fichier à utiliser .

Chaque information élémentaire de la base de données DNS est un objet appelé "*resource record*" (RR) qui partage le format commun suivant (les éléments entre crochet sont facultatifs) :

[domaine] [ttl] [classe] type données

- **domaine** : nom de la zone auquel s'appliquent les entrées. S'il est omis, le RR s'applique au domaine du précédent RR. On peut aussi comme dans cet exemple utiliser le symbole @ qui remplace le nom de la zone telle que défini dans "named.conf" ou éventuellement dans la variable $ORIGIN
- **ttl** : définit le "time to live" ou durée de vie, c'est à dire le temps pendant lequel cette information peut rester en cache. C'est un nombre décimal sur 8 chiffres, qui indique des secondes. S'il est omis, sa valeur sera égale à la valeur par défaut écrite dans la variable $TTL.
- **classe** : il s'agit d'une classe d'adresses. Toujours IN pour les adresses IP, s'il n'y a aucun champ classe, c'est la classe du précédent RR qui s'applique.
- **type** : décrit le type du RR (les plus courants sont A, SOA, PTR et NS)
- **données** : contient les données associées au RR, les données dépendront du type du RR.

Les types de l'exemple :

- **L'enregistrement de type SOA** (**S**tart **O**f **A**uthority – en français : responsable de la zone) est obligatoirement le premier : il donne les caractéristiques techniques générales de la zone. Il est suivi de l'adresse mél de l'administrateur qui ne comporte pas de signe « @ » (remplacé par un point).
- **L'enregistrement de type NS** définit un serveur de nom pour le nom de domaine.
- **L'enregistrement de type A** met en correspondance un nom de machine et une adresse IP.
- **L'enregistrement de type AAAA** est utilisé dans les réseaux IPv6.

Fichier `named.loopback` :

```
$TTL 1D
@       IN SOA  @ rname.invalid. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
```

```
NS      @
A       127.0.0.1
AAAA    ::1
PTR     localhost.
```

L'enregistrement PTR permet de faire de la résolution inverse c'est-à-dire associer à une adresse IP un nom FQDN.

Document 2 - Configuring BIND as a caching DNS server[1]

By default, the BIND DNS server resolves and caches successful and failed lookups. The service then answers requests to the same records from its cache. This significantly improves the speed of DNS lookups.

Prerequisites

- The IP address of the server is static.

Procedure

1. Install the `bind` and `bind-utils` packages:

```
# dnf install bind bind-utils
```

2. If you want to run BIND in a change-root environment install the `bind-chroot` package:

```
# dnf install bind-chroot
```

   Note that running BIND on a host with SELinux in `enforcing` mode, which is default, is more secure.

3. Edit the `/etc/named.conf` file, and make the following changes in the `options` statement:

   i. Update the `listen-on` and `listen-on-v6` statements to specify on which IPv4 and IPv6 interfaces BIND should listen:

```
listen-on port 53 { 127.0.0.1; 192.0.2.1; };
listen-on-v6 port 53 { ::1; 2001:db8:1::1; };
```

   ii. Update the `allow-query` statement to configure from which IP addresses and ranges clients can query this DNS server:

```
allow-query { localhost; 192.0.2.0/24; 2001:db8:1::/64; };
```

   iii. Add an `allow-recursion` statement to define from which IP addresses and ranges BIND accepts recursive queries:

```
allow-recursion { localhost; 192.0.2.0/24; 2001:db8:1::/64; };
```

1 https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/9/html/managing_networking_infrastructure_services/assembly_setting-up-and-configuring-a-bind-dns-server_networking-infrastructure-services#proc_configuring-bind-as-a-caching-dns-server_assembly_setting-up-and-configuring-a-bind-dns-server

Warning

Do not allow recursion on public IP addresses of the server. Otherwise, the server can become part of large-scale DNS amplification attacks.

iv. By default, BIND resolves queries by recursively querying from the root servers to an authoritative DNS server. Alternatively, you can configure BIND to forward queries to other DNS servers, such as the ones of your provider. In this case, add a `forwarders` statement with the list of IP addresses of the DNS servers that BIND should forward queries to:

```
forwarders { 198.51.100.1; 203.0.113.5; };
```

As a fall-back behavior, BIND resolves queries recursively if the forwarder servers do not respond. To disable this behavior, add a `forward only;` statement.

4. Verify the syntax of the `/etc/named.conf` file:

```
# named-checkconf
```

If the command displays no output, the syntax is correct.

5. Update the `firewalld` rules to allow incoming DNS traffic:

```
# firewall-cmd --permanent --add-service=dns
# firewall-cmd --reload
```

6. Start and enable BIND:

```
# systemctl enable --now named
```

If you want to run BIND in a change-root environment, use the `systemctl enable --now named-chroot` command to enable and start the service.

Verification

1. Use the newly set up DNS server to resolve a domain:

```
# dig @localhost www.example.org
...
www.example.org.    86400    IN    A    198.51.100.34

;; Query time: 917 msec
...
```

This example assumes that BIND runs on the same host and responds to queries on the `localhost` interface.

After querying a record for the first time, BIND adds the entry to its cache.

2. Repeat the previous query:

```
# dig @localhost www.example.org
...
www.example.org.    85332    IN    A    198.51.100.34
```

```
;; Query time: 1 msec
...
```

Because of the cached entry, further requests for the same record are significantly faster until the entry expires.

Next steps

- Configure the clients in your network to use this DNS server. If a DHCP server provides the DNS server setting to the clients, update the DHCP server's configuration accordingly.

Document 3 - Setting up a forward zone on a BIND primary server[2]

Forward zones map names to IP addresses and other information. For example, if you are responsible for the domain `example.com`, you can set up a forward zone in BIND to resolve names, such as `www.example.com`.

Prerequisites

- BIND is already configured, for example, as a caching name server.
- The `named` or `named-chroot` service is running.

Procedure

1. Add a zone definition to the `/etc/named.conf` file:

```
zone "example.com" {
    type master;
    file "example.com.zone";
    allow-query { any; };
    allow-transfer { none; };
};
```

These settings define:

- This server as the primary server (`type master`) for the `example.com` zone.
- The `/var/named/example.com.zone` file is the zone file. If you set a relative path, as in this example, this path is relative to the directory you set in `directory` in the `options` statement.
- Any host can query this zone. Alternatively, specify IP ranges or BIND access control list (ACL) nicknames to limit the access.
- No host can transfer the zone. Allow zone transfers only when you set up secondary servers and only for the IP addresses of the secondary servers.

2 https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/9/html/managing_networking_infrastructure_services/assembly_setting-up-and-configuring-a-bind-dns-server_networking-infrastructure-services#proc_setting-up-a-forward-zone-on-a-bind-primary-server_assembly_configuring-zones-on-a-bind-dns-server

2. Verify the syntax of the `/etc/named.conf` file:

```
# named-checkconf
```

   If the command displays no output, the syntax is correct.

3. Create the `/var/named/example.com.zone` file, for example, with the following content:

```
$TTL 8h
@ IN SOA ns1.example.com. hostmaster.example.com. (
                         2022070601 ; serial number
                         1d         ; refresh period
                         3h         ; retry period
                         3d         ; expire time
                         3h )       ; minimum TTL

                  IN NS   ns1.example.com.
                  IN MX   10 mail.example.com.

www               IN A    192.0.2.30
www               IN AAAA 2001:db8:1::30
ns1               IN A    192.0.2.1
ns1               IN AAAA 2001:db8:1::1
mail              IN A    192.0.2.20
mail              IN AAAA 2001:db8:1::20
```

   This zone file:

   - Sets the default time-to-live (TTL) value for resource records to 8 hours. Without a time suffix, such as `h` for hour, BIND interprets the value as seconds.
   - Contains the required SOA resource record with details about the zone.
   - Sets `ns1.example.com` as an authoritative DNS server for this zone. To be functional, a zone requires at least one name server (`NS`) record. However, to be compliant with RFC 1912, you require at least two name servers.
   - Sets `mail.example.com` as the mail exchanger (`MX`) of the `example.com` domain. The numeric value in front of the host name is the priority of the record. Entries with a lower value have a higher priority.
   - Sets the IPv4 and IPv6 addresses of `www.example.com`, `mail.example.com`, and `ns1.example.com`.

4. Set secure permissions on the zone file that allow only the `named` group to read it:

```
# chown root:named /var/named/example.com.zone
# chmod 640 /var/named/example.com.zone
```

5. Verify the syntax of the `/var/named/example.com.zone` file:

```
# named-checkzone example.com /var/named/example.com.zone
zone example.com/IN: loaded serial 2022070601
OK
```

6. Reload BIND:

```
# systemctl reload named
```

   If you run BIND in a change-root environment, use the `systemctl reload named-chroot` command to reload the service.

Verification

- Query different records from the `example.com` zone, and verify that the output matches the records you have configured in the zone file:

```
# dig +short @localhost AAAA www.example.com
2001:db8:1::30

# dig +short @localhost NS example.com
ns1.example.com.

# dig +short @localhost A ns1.example.com
192.0.2.1
```

  This example assumes that BIND runs on the same host and responds to queries on the `localhost` interface.

Document 4 - Setting up a reverse zone on a BIND primary server[3]

Reverse zones map IP addresses to names. For example, if you are responsible for IP range `192.0.2.0/24`, you can set up a reverse zone in BIND to resolve IP addresses from this range to hostnames.

Note

If you create a reverse zone for whole classful networks, name the zone accordingly. For example, for the class C network `192.0.2.0/24`, the name of the zone is `2.0.192.in-addr.arpa`. If you want to create a reverse zone for a different network size, for example `190.0.2.0/28`, the name of the zone is `28-2.0.192.in-addr.arpa`.

Prerequisites

- BIND is already configured, for example, as a caching name server.
- The `named` or `named-chroot` service is running.

Procedure

1. Add a zone definition to the `/etc/named.conf` file:

```
zone "2.0.192.in-addr.arpa" {
    type master;
    file "2.0.192.in-addr.arpa.zone";
```

3 https://access.redhat.com/documentation/fr-fr/red_hat_enterprise_linux/9/html/managing_networking_infrastructure_services/assembly_setting-up-and-configuring-a-bind-dns-server_networking-infrastructure-services#proc_setting-up-a-reverse-zone-on-a-bind-primary-server_assembly_configuring-zones-on-a-bind-dns-server

```
    allow-query { any; };
    allow-transfer { none; };
};
```

These settings define:

- This server as the primary server (`type master`) for the `2.0.192.in-addr.arpa` reverse zone.
- The `/var/named/2.0.192.in-addr.arpa.zone` file is the zone file. If you set a relative path, as in this example, this path is relative to the directory you set in `directory` in the `options` statement.
- Any host can query this zone. Alternatively, specify IP ranges or BIND access control list (ACL) nicknames to limit the access.
- No host can transfer the zone. Allow zone transfers only when you set up secondary servers and only for the IP addresses of the secondary servers.

2. Verify the syntax of the `/etc/named.conf` file:

```
# named-checkconf
```

If the command displays no output, the syntax is correct.

3. Create the `/var/named/2.0.192.in-addr.arpa.zone` file, for example, with the following content:

```
$TTL 8h
@ IN SOA ns1.example.com. hostmaster.example.com. (
                         2022070601 ; serial number
                         1d         ; refresh period
                         3h         ; retry period
                         3d         ; expire time
                         3h )       ; minimum TTL

                  IN NS   ns1.example.com.

1                 IN PTR  ns1.example.com.
30                IN PTR  www.example.com.
```

This zone file:

- Sets the default time-to-live (TTL) value for resource records to 8 hours. Without a time suffix, such as `h` for hour, BIND interprets the value as seconds.
- Contains the required SOA resource record with details about the zone.
- Sets `ns1.example.com` as an authoritative DNS server for this reverse zone. To be functional, a zone requires at least one name server (`NS`) record. However, to be compliant with RFC 1912, you require at least two name servers.
- Sets the pointer (`PTR`) record for the `192.0.2.1` and `192.0.2.30` addresses.

4. Set secure permissions on the zone file that only allow the `named` group to read it:

```
# chown root:named /var/named/2.0.192.in-addr.arpa.zone
# chmod 640 /var/named/2.0.192.in-addr.arpa.zone
```

5. Verify the syntax of the `/var/named/2.0.192.in-addr.arpa.zone` file:

```
# named-checkzone 2.0.192.in-addr.arpa /var/named/2.0.192.in-
addr.arpa.zone
zone 2.0.192.in-addr.arpa/IN: loaded serial 2022070601
OK
```

6. Reload BIND:

```
# systemctl reload named
```

   If you run BIND in a change-root environment, use the `systemctl reload named-chroot` command to reload the service.

Verification

- Query different records from the reverse zone, and verify that the output matches the records you have configured in the zone file:

```
# dig +short @localhost -x 192.0.2.1
ns1.example.com.

# dig +short @localhost -x 192.0.2.30
www.example.com.
```

  This example assumes that BIND runs on the same host and responds to queries on the `localhost` interface.