TP1 - DNS

Authors : Gabin Chognot & Julien Da Costa - P2025

Table of Contents

Configuration du serveur DNS pour la zone `fr`

> Sur le serveur *srvFR*, créer la zone `.fr` en tant que master. Il faut créer :
>
> - Un enregistrement SOA
> - Un enregistrement NS pour *srvFR1*

Configuration

On modifie `/etc/bind/named.conf.local`, qui contient la configuration locale du serveur DNS, pour y déclarer les zones associées au domaine :

```
zone ".fr" {
        type master;
        file "/var/cache/bind/db.fr";
};
```

Puis on crée le fichier `/var/cache/bind/db.fr` et on y ajoute les RR nécessaires :

- Un SOA (start of authority) pour indiquer le serveur primaire (`.fr`), le contact technique (`.root.fr`), et les paramètres d'expiration
- Un NS (nameserver) pour lier le sous domaine `srv1` à la zone `.fr`
- Un A pour lier le serveur `srv1.fr` à `195.25.25.1` (l'IP de la machine srvFR)
- Le `TTL` indique la duree de validité (secondes), des informations contenues dans les RRs, et le délai à partir duquel il faut les revérifier.

```
$TTL    3600
@       IN      SOA     .fr. root.fr. (
                     2023040701           ; Serial
                           3600           ; Refresh [1h]
                            600           ; Retry   [10m]
                          86400           ; Expire  [1d]
                            600 )         ; Negative Cache TTL [1h]
;
@        IN      NS      srv1.fr.
srv1     IN      A       195.25.25.1
```

Tests

1 - Checkconf

`named-checkconf -z` :

- Vérifie la validité syntaxique des fichiers de configuration de Bind9
- Le paramètre `-z` réalise un *test load* sur toutes les *master zones* trouvées dans *named.conf*

- Renvoie `zone fr/IN: loaded serial 2023040701` → le fichier a été *load*

2 - Checkzone

`named-checkzone fr /var/cache/bind/db.fr` :

- Vérifie la validité des fichiers de zones avant de recharger la configuration
- Renvoie une validation : `zone srv1.fr/IN: loaded serial 2023040701 OK`

3 - Dig

Interroge directement le serveur DNS demandé et renvoie beaucoup informations, en plus de la résolution de noms et de la résolution inverse. On réalise deux `dig`, sur `srv1.fr` et `fr`

`dig fr` :

```
; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59191
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 35a75e6068a5fb1857998511642fda8bdc4e9dfcd0a9f805 (good)
;; QUESTION SECTION:
;fr. IN A

;; AUTHORITY SECTION:
fr. 600 IN SOA srv1.fr. root.example.com. 2023040701 3600 600 86400 600

;; Query time: 2 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Fri Apr 07 08:55:39 UTC 2023
;; MSG SIZE rcvd: 116
```

On a bien une réponse de l'*Authority Section*, et le domaine `fr` est bien configuré.

Délégation de la zone DNS `irc.fr`

> Sur *SrvFR* : Créer une délégation pour la zone irc.fr, vers la machine `srvIRC1.irc.fr`.
>
> Sur *srvIRC1* : Créer la zone irc.fr en tant que master, avec :
>
> - Un enregistrement A pour `www.irc.fr`
> - Un enregistrement CNAME `web.irc.fr` pointant sur `www.irc.fr`

Sur la machine srvFR

On ajoute les RRs NS et A à `/var/cache/bind/db.fr` :

```
irc IN NS srvIRC1.irc.fr.
srvIRC1.irc.fr. IN A 195.25.25.2
```

On modifie aussi `/etc/bind/named.conf.options` pour autoriser les requêtes récursives :

```
options {
        ...
        recursion yes;
        allow-recursion { any; };
        ...
};
```

Sur la machine *svrIRC1*

On modifie `/etc/bind/named.conf.local` pour définir *srvIRC1* comme serveur maitre sur `irc.fr`

```
zone "irc.fr" {
        type master;
        file "/var/cache/bind/db.irc.fr";
};
```

Et de même avec `/var/cache/bind/db.irc.fr`

```
$TTL    3600
@       IN      SOA     srvIRC1.irc.fr. root.irc.fr. (
                     2023040701         ; Serial
                        3600            ; Refresh [1h]
                         600            ; Retry   [10m]
                       86400            ; Expire  [1d]
                         600 )          ; Negative Cache TTL [1h]
;
```

```
@           IN      NS      srvIRC1.irc.fr.
srvIRC1     IN      A       195.25.25.2
```

`dig irc.fr` nous retourne des informations valides sur la zone déléguée :

```
root@client:~# dig irc.fr

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> irc.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35070
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 8e734948f41d3ad8f0eab989642fda9987111314143f852f (good)
;; QUESTION SECTION:
;irc.fr. IN A

;; AUTHORITY SECTION:
irc.fr. 455 IN SOA srvIRC1.irc.fr. root.example.com. 2023040701 3600 600 86400 600

;; Query time: 1 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Fri Apr 07 08:55:53 UTC 2023
;; MSG SIZE rcvd: 123
```

`www` section

On ajoute dans `/var/cache/bind/db.irc.fr` le A record pour `www.` (qui pointe vers la même IP), et le CNAME pour `web.`

```
www IN A 195.25.25.2
web IN CNAME www
```

On peut ping `www.irc.fr` et `web.irc.fr`, mais les requêtes prennent du temps car le reverse DNS n'a pas encore été configuré.

On peut également dig `www.irc.fr` et `web.irc.fr` pour vérifier la validité de notre manipulation :

```
root@client:~# dig www.irc.fr

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> www.irc.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30004
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 6bbaca3c9cc9f0aab0825bd2642fdc2d56441443ac507096 (good)
;; QUESTION SECTION:
;www.irc.fr. IN A

;; ANSWER SECTION:
www.irc.fr. 3475 IN A 195.25.25.2

;; AUTHORITY SECTION:
irc.fr. 3600 IN NS srvIRC1.irc.fr.

;; Query time: 2 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Fri Apr 07 09:02:37 UTC 2023
;; MSG SIZE rcvd: 105
```

Et :

```
root@client:~# dig web.irc.fr

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> web.irc.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49395
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d404ba3a9a3ab6ad264948ee642fdc2fa41bafe375a3013b (good)
;; QUESTION SECTION:
;web.irc.fr. IN A

;; ANSWER SECTION:
web.irc.fr. 3499 IN CNAME www.irc.fr.
www.irc.fr. 3473 IN A 195.25.25.2

;; AUTHORITY SECTION:
irc.fr. 3600 IN NS srvIRC1.irc.fr.

;; Query time: 4 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Fri Apr 07 09:02:39 UTC 2023
;; MSG SIZE rcvd: 123
```

On a bien une réponse valide pour `www.irc.fr` et `web.irc.fr` -> la configuration est bonne.

Redondance de la zone `irc.fr`

> Sur *srvFr* : Ajouter un enregistrement NS pour la zone `irc.fr` vers `srvIRC2.irc.fr.`
>
> Sur *srvIRC2* : Configurer la zone slave `irc.fr`.
>
> Sur *srvIRC1* : Configurer la zone `irc.fr` pour ajouter le nouveau serveur secondaire.

Sur la machine *svrIRC2*

On modifie le fichier `/etc/bind/named.conf.local` pour définir *srvIRC2* comme serveur secondaire sur `irc.fr`

```
zone "irc.fr" {
        type slave;
        file "/var/cache/bind/db.irc.fr";
        masters { 195.25.25.2; };
};
```

On modifie le fichier `/var/cache/bind/db.irc.fr` pour définir *srvIRC2* comme serveur secondaire sur `irc.fr`

```
@ IN NS srvIRC2.irc.fr.
```

Sur *svrFR*

On ajoute le serveur secondaire dans `/var/cache/bind/db.fr` :

```
irc IN NS srvIRC2.irc.fr.
srvIRC2.irc.fr. IN A 195.25.25.3
```

Sur *svrIRC1*

On modifie les options dans `/etc/bind/named.conf.options` pour autoriser les requêtes de *srvIRC2* et notifier le serveur primaire à chaque transfert de zone `irc.fr`:

```
options {
  ...
  allow-transfer { any; };
  notify yes;
  ...
};
```

NB : En conditions réelles, on évite de mettre `any` dans `allow-transfer` et on utilise une liste d'adresses IP autorisées.

Test de la redondance

1 - Test d'indisponibilité du serveur primaire

On éteint srvIRC1 :

```
root@srvIRC1:~# systemctl stop bind9
```

On vérifie que srvIRC2 répond toujours aux requêtes DNS :

```
root@client:~# dig irc.fr

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> irc.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10554
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: ba55375190404f736994b064643fe832215604d7eb318e4a (good)
;; QUESTION SECTION:
;irc.fr.                                IN      A

;; AUTHORITY SECTION:
irc.fr.                 206     IN      SOA     srvIRC1.irc.fr. root.example.com.
2023040701 3600 600 86400 600

;; Query time: 1 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Wed Apr 19 13:10:10 UTC 2023
;; MSG SIZE  rcvd: 123
```

Note : on a la même réponse avec une requête spécifique sur srvIRC2 (`dig irc.fr @195.25.25.3`).

Et on s'assure aussi que une requête vers srvIRC1 échoue (`dig irc.fr @195.25.25.2`).

2 - Test de la mise à jour de la zone

Pour tester notre configuration :

- On augmente de 2 le serial number de la zone `irc.fr` sur *svrIRC1* (2023040701 -> 2023040703).
- On redémarre *srvICR1* avec `systemctl start bind9`
- On effectue une requête sur *srvIRC1* pour vérifier que le serial number a bien été mis à jour : `dig irc.fr @195.25.25.2` nous renvoie bien le nouveau sérial number (2023040703).

- On redémarre le serveur secondaire *srvIRC2* et on vérifie qu'il a fait bien la mise à jour, avec `dig irc.fr @195.25.25.3` qui renvoie aussi le bon serial number.

Configuration d'un résolveur avec cache

> Le résolveur de la machine client doit être configuré pour qu'il utilise le serveur *srvCache* :
>
> - *srvCache* doit forward les requêtes DNS vers srvFr.
> - *srvCache* doit être configuré pour que seule la machine client puisse l'utiliser comme résolveur.

Sur *client*

La syntaxe générale de `/etc/resolv.conf` est la suivante :

```
nameserver <serveur DNS Primaire>
nameserver <serveur DNS Secondaire>
search <domaine de recherche>
```

On modifie le fichier `/etc/resolv.conf` pour utiliser *srvCache* comme serveur DNS, en ajoutant son IP :

```
nameserver 192.168.1.1
```

Ici, on ne précise qu'un seul serveur DNS car il n'y a pas de resolvers secondaires, et pas de domaine de recherche car on n'a qu'un seul TLD (`fr`).

Sur *srvCache*

On modifie le fichier `/etc/bind/named.conf.options` pour autoriser les requêtes de *client* et notifier le serveur primaire à chaque transfert de zone `fr`:

```
acl clients {
        192.168.1.10;
};

options {
  ...
  forwarders {
        195.25.25.1;
  };
  allow-query { clients; };
  dnssec-validation no;
  allow-recursion { any; };
  ...
};
```

Explications :

- `ACL clients` - liste d'adresses IP autorisées à faire des requêtes sur le serveur de cache.
- `forwarders` - liste de serveurs DNS qui vont être utilisés pour forwarder les requêtes DNS.
- `allow-query` - liste d'adresses IP autorisées à faire des requêtes sur le serveur de cache. Ici c'est notre liste `clients`.
- `allow-recursion` - autorise les requêtes récursives (nécessaires pour un resolver).
- `dnssec-validation no` - désactive la validation DNSSEC, car on n'a pas de clés DNSSEC sur nos serveurs.

Test du serveur de cache

On vérifie que le serveur de cache répond bien aux requêtes DNS :

```
root@client:~# dig fr

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51105
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 7f0bcf23c14161b2fe712cb36440093d81f920c7b7902e05 (good)
;; QUESTION SECTION:
;fr.                            IN      A

;; AUTHORITY SECTION:
fr.                     10800   IN      SOA     srv1.fr. root.example.com.
2023040701 3600 600 86400 600

;; Query time: 6 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Apr 19 15:31:10 UTC 2023
;; MSG SIZE  rcvd: 116
```

On a bien une réponse, le serveur de cache a bien forwardé la requête vers *srvFr* et a renvoyé la réponse.

NB : `dig irc.fr` fonctionne aussi, notre resolver est bien récursif.

Cache :

- On exporte le contenu du cache du serveur de cache avec `rndc dumpdb -cache`
- On affiche dans le fichier `named_dump.db` les éléments du cache qui correspondent à la requête `dig fr` :

```
root@srvCache:~# cat /var/cache/bind/named_dump.db

; Start view _default
;
```

```
; Cache dump of view '_default' (cache _default)
;
$DATE 20230419153454
; answer
fr.                     10575   IN \-A  ;-$NXRRSET
; fr. SOA srv1.fr. root.example.com. 2023040701 3600 600 86400 600
;
; Address database dump
;
; [edns success/4096 timeout/1432 timeout/1232 timeout/512 timeout]
; [plain success/timeout]
;
;
; Unassociated entries
;
;       2001:500:9f::42 [srtt 80280] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0]
[ttl 1575]
...
;       195.25.25.1 [srtt 31210] [flags 00004000] [edns 1/1/1/1/1] [plain 0/0]
[udpsize 512] [cookie=2c2a211d366eb9862aab00756440093dcaf17e2cac0f42f4] [ttl 1575]
...
;       192.36.148.17 [srtt 1920] [flags 00000000] [edns 0/1/1/1/1] [plain 0/0]
[ttl 1575]
;
; Bad cache
;
; SERVFAIL cache
...
; Dump complete
```

NB : dump tronqué, les . . . symbolysent un ensemble de lignes non-pertinentes ici.

Configuration zone reverse

> On veut créer la zone reverse pour le réseau `195.25.25.0/24`, avec toutes les machines du réseau.

Sur *srvFr*

On modifie le fichier `/etc/bind/named.conf.local` pour ajouter la zone `25.25.195.in-addr.arpa` :

```
zone "25.25.195.in-addr.arpa" {
        type master;
        file "/var/cache/bind/db.195.25.25";
};
```

On crée le fichier `/var/cache/bind/db.195.25.25` avec la configuration suivante :

```
$TTL 3600
@       IN      SOA     srv1.fr.        root.example.com. (
                     2023041901         ; Serial
                           3600         ; Refresh [1h]
                            600         ; Retry   [10m]
                          86400         ; Expire  [1d]
                            600 )       ; Negative Cache TTL [1h]
;
@ IN NS srv1.fr.

1 IN PTR srv1.fr.
2 IN PTR srvIRC1.irc.fr.
3 IN PTR srvIRC2.irc.fr.
```

On a donc 3 entrées dans la zone reverse :

- `1` qui pointe vers `srv1.fr`
- `2` qui pointe vers `srvIRC1.irc.fr`
- `3` qui pointe vers `srvIRC2.irc.fr`

Test de la zone reverse

1 - Tests avec `dig`

On vérifie que le serveur de cache répond bien aux requêtes DNS :

```
root@client:~# dig -x 195.25.25.1

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> -x 195.25.25.1
;; global options: +cmd
```

```
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54223
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: a3bded65870fb9a1cfa223d6644103c5b56a492a564149f9 (good)
;; QUESTION SECTION:
;1.25.25.195.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
1.25.25.195.in-addr.arpa. 3600  IN      PTR     srv1.fr.

;; AUTHORITY SECTION:
25.25.195.in-addr.arpa. 3600    IN      NS      srv1.fr.

;; ADDITIONAL SECTION:
srv1.fr.                3600    IN      A       195.25.25.1

;; Query time: 2 msec
;; SERVER: 195.25.25.1#53(195.25.25.1)
;; WHEN: Thu Apr 20 09:20:05 UTC 2023
;; MSG SIZE  rcvd: 132
```

L'option `-x` permet de faire une requête inverse, c'est-à-dire de demander l'adresse d'un nom de domaine.

On effectue des requêtes inverses similaires sur *srvIRC1* et *srvIRC2* avec `dig -x 195.25.25.2` et `dig -x 195.25.25.3`, on obtient bien également les bonnes réponses.

2 - Tests avec `ping`

Additionnellement, on peut tester la zone reverse avec `ping` :

- Si le revere DNS n'est pas configuré, `ping 195.25.25.1` est instantané, mais `ping srv1.fr` prend plusieurs secondes pour résoudre le nom de domaine.
- Si le reverse DNS est correctement configuré, `ping srv1.fr` est instantané.
    - C'est notre cas ici, notre reverse DNS fonctionne.