Recommandations DNS lorsqu'on change d'adresse IP

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 janvier 2008

https://www.bortzmeyer.org/changement-adresse-et-dns.html

Aujourd'hui, avec les mécanismes d'allocations d'adresses IP existants (cf. RFC 7020 [1]), il est relativement fréquent d'avoir à changer son adresse IP. Quelles sont les précautions particulières liées au DNS lors de ce changement ?

Supposons donc qu'un utilisateur aie une machine à lui chez Slicehost <https://www.bortzmeyer.org/slicehost-debut.html>, Gandi <http://www.gandi.net/hebergement/>, OVH <http://www.kimsufi.org/> ou un autre. L'adresse IP de cette machine est « fixe » au sens où elle ne change pas toutes les 24 heures, mais rien ne garantit qu'elle ne changera jamais. Une nouvelle configuration du réseau, un changement de fournisseur (pour les hébergeurs qui n'ont pas leurs propres adresses IP) et on doit changer. Il existe plusieurs précautions à prendre lors d'un tel changement mais on ne parle ici que de celles liées au DNS.

D'abord et avant tout, il faut penser que les informations distribuées par le DNS sont gardées dans des caches. La durée de vie maximale dans le cache est gouvernée par un paramètre nommé le TTL (RFC 1034, section 3.6), que l'administrateur de la zone fixe lui-même. Il est recommandé de réduire ce TTL avant le changement.

dig peut afficher le TTL, ce qui permet de vérifier la configuration serveur :

```
% dig @mon-serveur A www.venezvoirchezmoi.fr.
...
www.venezvoirchezmoi.fr.        86400   IN      A       192.0.2.249
```

Le TTL indiqué par le serveur faisant autorité est donc de 86400 secondes. Sur un serveur ne faisant pas autorité, le TTL diminue petit à petit et indique donc combien de temps il reste avant le renouvellement du cache :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc7020.txt

```
% dig A www.venezvoirchezmoi.fr.
www.venezvoirchezmoi.fr.        33982   IN      A       192.0.2.249
```

Cette deuxième requête a interrogé le résolveur local, qui ne fait pas autorité, et a donné l'information à partir de son cache. L'enregistrement était déjà dans ce cache depuis 86400 - 33982 = 52418 secondes.

Cas le plus simple ; soit un bête enregistrement ordinaire qui stocke une adresse IP d'un serveur Web (ici, en IPv6) :

```
www   IN  AAAA   2001:DB8::DEAD:BABE
```

Son TTL va être donné (si on utilise BIND ou NSD) par la directive `$TTL`. Mais on peut la changer pour chaque enregistrement :

```
www  300 IN  AAAA   2001:DB8::DEAD:BABE   ; Trois cents seconds, soit cinq minutes
```

Ainsi, lors du changement d'adresse, il n'y aura que cinq minutes de problème au maximum.

On peut descendre le TTL à 60 si on est pressé, mais il vaut mieux éviter de l'abaisser à zéro, pas mal de FAI l'ignorent (car il y a trop d'abus) et le forcent à une valeur plus élevée (ce qui, formellement, viole le RFC).

Évidemment, il faut faire ce changement de durée de vie à l'avance (si le TTL actuel est de 86 400 secondes, il faut faire le changement au moins une journée avant).

À l'heure H, on peut alors changer l'enregistrement (ici le AAAA, l'heure H est celle où on indique la nouvelle adresse IP). Il ne faut **pas** remonter le TTL tout de suite, il faut tester d'abord que tout marche bien.

Le TTL ne gouverne que la durée de vie dans les caches des résolveurs ne faisant pas autorité. Pour les serveurs faisant autorité sur la zone (le maître, autrefois nommé « primaire » et les esclaves, autrefois nommés « secondaires »), il faut aussi surveiller leur synchronisation. Si, par exemple, les NOTIFY (messages DNS normalisés dans le RFC 1996 et permettant de prévenir immédiatement les esclaves) sont ignorés (il peut y avoir des tas de raisons pour cela), le temps de synchronisation va s'ajouter au TTL. Les esclaves ayant raté le NOTIFY ne seront synchronisés qu'au prochain test. Les tests sont effectués toutes les `Refresh` secondes, avec réessai toutes les `Retry` secondes en cas d'échec. Les paramètres `Refresh` et `Retry` sont fixés dans l'enregistrement SOA. On voit donc l'importance de tester la synchronisation des serveurs de la zone **avant** l'heure H.

Maintenant, cas compliqué, l'adresse IP peut se trouver à d'autres endroits que la zone :

— règles d'un coupe-feu,
— documentations,
— chez le registre et/ou le "*registrar*".

Un coup de grep dans `/etc/**` est donc recommandé pour trouver toutes les occurrences de l'ancienne adresse IP.

Pour le cas registre / "*registrar*", ça dépend de leur réactivité combinée. Là encore, il vaut mieux étudier avant. Il est donc prudent de prévoir un recouvrement (ancien serveur qui marche toujours) si on change la **colle** (enregistrements d'adresses qui sont stockés dans la zone parente).

Cette question étant souvent celle qui pose le plus de problèmes, il faut donc faire doublement attention lorsqu'on change l'adresse IP d'une machine qui est serveur de noms, surtout lorsqu'elle a un nom dans la zone servie (obligeant ainsi le registre à garder la colle).

Prenons l'exemple de la machine `192.0.2.53`, qui est connue sous le nom de `ns1.example.com` et qui est serveur de noms de la zone `example.com`. Le fait qu'elle soit nommée dans la zone qu'elle sert oblige le registre de `.com` à distribuer la colle, et donc à mémoriser l'adresse IP (certains registres gardent l'adresse IP des serveurs même lorsqu'elle n'est pas nécessaire, ce qui est mal vu, mais arrive). Son changement d'adresse va donc nécessiter de vérifier l'information distribuée par le registre (et mise à jour via le "*registrar*" puisque `.com` impose le passage par un intermédiaire). Voyons avec dig comment vérifier l'adresse distribuée par le registre, en interrogeant les serveurs de noms de Verisign :

```
% dig @a.gtld-servers.net A ns1.example.com.
ns1.example.com.       172800  IN      A       192.0.2.3
```

(Notez que cette réponse ne fait pas autorité, le "*flag*" `aa` sera donc absent.) Le DNS ne reflète pas toujours l'état actuel de la base de données du registre. Selon les cas, il peut être prudent de vérifier avec whois :

```
% whois -h whois.internic.net ns1.example.com
   Server Name: NS1.EXAMPLE.COM
   IP Address: 192.0.2.3
```

Ces deux commandes permettent de voir si les changements ont bien été transmis au registre, et exécutés.