

Sommaire :

Sommaire :	1
Introduction.....	1
Installer bind9 :	1
Serveur maître :	2
Serveur esclave :	5
Serveur DHCP :	7
Serveur DHCP et DNS dynamique :	8

Introduction

Nous devons mettre en place un service DNS.

Il faut vérifier que bind a les droits sur les répertoires ***/etc/bind*** et ***/var/cache/bind***, ou tout répertoire de destination ! On utilise la commande `chown -R bind.bind bind` par exemple.

Il faut également mettre l'adresse de son serveur DNS dans ***/etc/resolv.conf*** !

Il faut redémarrer le ***service bind9 restart*** et incrémenter le fichier de zone à chaque modification !

```
#domain sio.local
#search sio.local
#nameserver 192.168.1.49
#nameserver 192.168.1.50
#nameserver 8.8.8.8
#nameserver 81.253.149.6
#nameserver 80.10.246.136
#nameserver 192.168.1.254

domain sebdomain.local
search sebdomain.local
nameserver 192.168.1.116
```

Installer bind9 :

On utilise ***apt-get install bind9***. Des documents ont été ajouté à l'emplacement ***/etc/bind***.

named-checkconf fichierdeconf → Permet de vérifier les fichiers de configuration.

named-checkzone domaine fichier → Permet de vérifier les fichiers de zones.

Serveur maître :

Nous allons maintenant transformer notre serveur cache seulement en un serveur maître pour un domaine Intranet que nous allons baptiser **sebdomain.local**.

Le serveur possède maintenant les informations officielles sur le domaine et fait autorité sur la zone concernée. Il charge en mémoire cache ses informations à partir des fichiers disques que nous devons renseigner. Ce sera l'occasion de découvrir la syntaxe des enregistrements DNS.

Nous considérons toujours que notre serveur accède à l'Internet et qu'il va ainsi pouvoir réaliser des recherches sur le réseau mondial.

Nous accédons ensuite au fichier **nano /etc/bind/named.conf.local** pour commencer la configuration du serveur maître.

Il faut donc préciser à notre serveur que nous allons créer des fichiers de zones :

```
zone "sebdomain.local" IN {
    type master;
    file "/etc/bind/meszones/zone.sebdomain.local";
};
```

On déclare notre domaine, puis notre domaine en inversé :

```
zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/meszones/1.168.192.in-addr.arpa";
};
```

Par défaut, les fichiers de zones sont à placer dans **/var/cache/bind**, c'est pourquoi on peut directement écrire le chemin complet de l'emplacement de nos fichiers de zones, ou modifier le chemin par défaut de **named.conf.options**.

Dans le répertoire **/etc/bind/meszones**, on crée ensuite le fichier **zone.sebdomain.local** :

```
$TTL 86400
sebdomain.local. IN SOA sebdns.sebdomain.local. root.sebdomain.local. (
    2016092204      ; serial
    86400          ; refresh
    21600         ; retry
    3600000       ; expire
    3600          ; minimum
)
sebdomain.local. IN NS sebdns.
sebdns.sebdomain.local. IN A 192.168.1.116
pcimaginaire.sebdomain.local. IN A 192.168.1.117
```

Avec la commande ***named-checkzone domaine fichier***, on peut tester la validité du fichier :

```
root@sebdns:/etc/bind/meszones# named-checkzone sebdomain.local zone.sebdomain.local
zone sebdomain.local/IN: loaded serial 2016092205
OK
```

On peut ensuite vérifier le fonctionnement du service DNS avec la commande ***nslookup nommachine.domaine*** ou la commande ***dig nommachine.domaine*** :

```
root@sebdns:/etc/bind/meszones# nslookup
> pcimaginaire.sebdomain.local
Server:          192.168.1.116
Address:         192.168.1.116#53

Name:   pcimaginaire.sebdomain.local
Address: 192.168.1.117
```

```
root@sebdns:/etc/bind/meszones# dig sebdns.sebdomain.local

;<<>> DiG 9.9.5-9+deb8u6-Debian <<>> sebdns.sebdomain.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40632
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;sebdns.sebdomain.local.          IN      A

;; ANSWER SECTION:
sebdns.sebdomain.local. 86400   IN      A      192.168.1.116

;; AUTHORITY SECTION:
sebdomain.local.        86400   IN      NS     sebdns.

;; Query time: 18 msec
;; SERVER: 192.168.1.116#53(192.168.1.116)
;; WHEN: Thu Sep 22 10:30:05 CEST 2016
;; MSG SIZE rcvd: 87
```

On peut ensuite créer le fichier de zone inversée :

```
$TTL 86400
1.168.192.in-addr.arpa. IN SOA sebdns.sebdomain.local. root.sebdomain.local. (
    2016092203      ; serial
    86400           ; refresh
    21600           ; retry
    3600000        ; expire
    3600           ; minimum
)
1.168.192.in-addr.arpa. IN NS sebdns.sebdomain.local.
116 IN PTR sebdns.sebdomain.local.
117 IN PTR pcimaginaire.sebdomain.local.
```

Et le vérifier avec la commande ***named-checkzone domaine fichier*** :

```
root@sebdns:/etc/bind/meszones# named-checkzone 1.168.192.in-addr.arpa 1.168.192
.in-addr.arpa
zone 1.168.192.in-addr.arpa/IN: loaded serial 2016092202
OK
```

Puis tester un ***nslookup ip*** ou un ***dig -x ip*** :

```
root@sebdns:/etc/bind/meszones# nslookup
> 192.168.1.117
Server:          192.168.1.116
Address:         192.168.1.116#53

117.1.168.192.in-addr.arpa      name = pcimaginaire.sebdomain.local.
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32849
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;117.1.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
117.1.168.192.in-addr.arpa. 86400 IN     PTR     pcimaginaire.sebdomain.local.

;; AUTHORITY SECTION:
1.168.192.in-addr.arpa. 86400 IN     NS      sebdns.sebdomain.local.

;; ADDITIONAL SECTION:
sebdns.sebdomain.local. 86400 IN     A       192.168.1.116

;; Query time: 25 msec
;; SERVER: 192.168.1.116#53(192.168.1.116)
;; WHEN: Thu Sep 22 11:36:31 CEST 2016
;; MSG SIZE rcvd: 134
```

Parmi les erreurs, si le nslookup renvoie la réponse « **serveur failed** », cela veut dire que le service n'est pas disponible. Si la réponse est « **server can't find ...** », le service fonctionne mais n'a pas l'enregistrement recherché.

Serveur esclave :

Pour un domaine donné, il est important de configurer au moins un serveur de secours au cas où le serveur maître connaisse un gros problème. Ce serveur de secours est qualifié d'esclave dans le schéma fonctionnel DNS.

En complément des trois fichiers, déjà étudiés, qui existent toujours quel que soit le type de serveur, un serveur esclave stockera des fichiers disques contenant les informations sur les machines du domaine. Cependant, il faut bien comprendre que ces fichiers seront des copies de ceux de même

rôle sur le serveur maître. Ils seront obtenus auprès de ce serveur maître lors du démarrage de l'esclave et ils ne seront en aucun cas constitués manuellement.

Le fichier de configuration général d'un serveur esclave fait apparaître le type slave dans les strophes des zones concernées. Une directive `masters` fait apparaître l'adresse IP du serveur maître qu'il faudra contacter pour obtenir une copie du fichier de même rôle. Nous décidons de donner à ce fichier, obtenu auprès du maître, le préfixe slave.

On utilise `apt-get install bind9`. Des documents ont été ajoutés à l'emplacement `/etc/bind`.

Nous accédons ensuite au fichier `nano /etc/bind/named.conf.local` pour commencer la configuration du serveur esclave :

```
zone "sebdomain.local" IN {
    type slave;
    masters { 192.168.1.116; } ;
    file "/etc/bind/meszones/slave.zone.sebdomain.local";
};

zone "1.168.192.in-addr.arpa" IN {
    type slave;
    masters { 192.168.1.116; } ;
    file "/etc/bind/meszones/slave.1.168.192.in-addr.arpa";
};
```

Sur le serveur maître, on ajoute ensuite un enregistrement de type NS et de type A qui mentionnera le nouveau serveur esclave, à la fois dans le fichier de `zone.sebdomain.local` :

```
sebdomain.local. IN NS sebdns.
sebdomain.local. IN NS esclavedns.

sebdns.sebdomain.local. IN A 192.168.1.116
esclavedns.sebdomain.local. IN A 192.168.1.117
```

Et dans le fichier de zone inverse `1.168.192.in-addr.arpa` :

```
1.168.192.in-addr.arpa. IN NS sebdns.sebdomain.local.
1.168.192.in-addr.arpa. IN NS esclavedns.sebdomain.local.

116 IN PTR sebdns.sebdomain.local.
117 IN PTR esclavedns.sebdomain.local.
```

Toujours sur le serveur maître, dans le fichier `named.conf.local`, on autorise le serveur esclave à recopier les informations de zone à l'aide des directives `notify` et `allow-transfer` :

```
zone "sebdomain.local" IN {
    type master;
    file "/etc/bind/meszones/zone.sebdomain.local";
    notify yes;
    allow-transfer { 192.168.1.117 ; } ;
};
```

Après avoir redémarré les deux services, les fichiers de zones sont bien présents sur le serveur esclave :

```
sept. 29 09:17:14 esclavedns named[1713]: zone 1.168.192.in-addr.arpa/IN: se...)
sept. 29 09:17:14 esclavedns named[1713]: zone sebdomain.local/IN: sending n...)
```

```
root@esclavedns:/etc/bind/meszones# ls -l
total 8
-rw-r--r-- 1 bind bind 388 sept. 29 09:17 slave.1.168.192.in-addr.arpa
-rw-r--r-- 1 bind bind 361 sept. 29 09:17 slave.zone.sebdomain.local
```

Serveur DHCP :

Avant de le mettre en place, il faut vérifier que l'on possède une IP fixe et un serveur DNS. Il faut ensuite installer le paquet **apt-get install isc-dhcp-server**.

Pour démarrer, il faut au moins une étendue sur le service DHCP. Il faut donc configurer le fichier de conf en conséquence. Pour cela il faut accéder au répertoire **/etc/dhcp** et au fichier **dhcpd.conf**.

Pour créer une étendue basique, il faut faire :

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.118 192.168.1.119 ;
    option domain-name « sebdns.local » ;
    option domain-name-servers 192.168.1.116, 192.168.1.117 ;
    option routers 192.168.1.254 ;
    default-lease-time 600 ;
    max-lease-time 7200 ;
}
```

Ensuite, il faut démarrer le service : **service isc-dhcp-server start** et démarrer le client DHCP sur un Windows 7.

Avec des **ipconfig /release** et **ipconfig /renew** sur notre client Windows 7, on peut tester les différents services DHCP :

```

C:\Windows\system32\cmd.exe
C:\Users\seb>ipconfig /renew

Configuration IP de Windows

Une erreur s'est produite lors de la libération de l'interface Loopback Pseudo-Interface 1 : Le fichier spécifié est introuvable.

Carte Ethernet Connexion au réseau local :

    Suffixe DNS propre à la connexion. . . : sebdomain.local
    Adresse IPv6. . . . . : fd23:6507:b29b:1:88f0:609d:1a55:e4f
    Adresse IPv6 temporaire . . . . . : fd23:6507:b29b:1:a01c:7c19:5bf9:d577
    Adresse IPv6 de liaison locale. . . . . : fe80::88f0:609d:1a55:e4f%11
    Adresse IPv4. . . . . : 192.168.1.119
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.1.254

Carte Tunnel isatap.<A562146F-E883-4C16-AAA0-2486966B2639> :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . :

C:\Users\seb>
    
```

Serveur DHCP et DNS dynamique :

Pour Windows :

Nous devons ajouter le serveur DHCP dans nos zones DNS :

```

sebdns.sebdomain.local. IN A 192.168.1.116
esclavedns.sebdomain.local. IN A 192.168.1.117
sebdhcp.sebdomain.local. IN A 192.168.1.118_
    
```

```

116 IN PTR sebdns.sebdomain.local.
117 IN PTR esclavedns.sebdomain.local.
118 IN PTR sebdhcp.sebdomain.local.
    
```

Il faut ensuite modifier le fichier **/etc/bind/named.conf.local** pour autoriser la mise à jour du DNS par le DHCP :

```
zone "sebdomain.local" IN {
    type master;
    file "/etc/bind/meszones/zone.sebdomain.local";
    notify yes ;
    allow-transfer { 192.168.1.117; } ;
    allow-update { 192.168.1.118; } ;
};
```

```
zone "1.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/meszones/1.168.192.in-addr.arpa";
    notify yes ;
    allow-transfer { 192.168.1.117; } ;
    allow-update { 192.168.1.118; } ;
```

Il faut activer la mise à jour dans */etc/dhcp/dhcpd.conf* :

```
GNU nano 2.2.6      Fichier : dhcpd.conf
# Méthode dynamique pour la mise à jour :
ddns-update-style interim;

# Autorisation de la mise à jour :
ddns-updates on;

# La mise à jour est faite par le serveur DHCP :
ignore client-updates;

# Mise à jour même en cas d'IP statiques :
update-static-leases on;

# Admettre aussi les clients inconnus au niveau de l'adresse MAC :
allow-unknown-clients;
```

Puis, dans le même fichier, on ajoute le serveur DNS à mettre à jour :

```
# Donne précisément quel DNS à mettre à jour :
zone sebdomain.local. { primary 192.168.1.116; }
zone 1.168.192.in-addr.arpa. { primary 192.168.1.116; }
```

Nous relançons ensuite nos services DNS et DHCP (**dans cet ordre**) : ***service bind9 restart*** et ***service isc-dhcp-server restart***.

Il faut ensuite lancer des ***ipconfig /release*** et ***ipconfig /renew*** sur un client Windows. Lorsqu'un utilisateur reçoit une adresse IP de notre service DNS, ce dernier ajoute l'utilisateur dans ses fichiers de zones grâce au DHCP :


```
Oct 13 10:25:46 sebdns named[948]: client 192.168.1.118#16257: updating zone 'se  
bdomain.local/IN': adding an RR at 'sebwindows.sebdomain.local' A  
Oct 13 10:25:46 sebdns named[948]: client 192.168.1.118#16257: updating zone 'se  
bdomain.local/IN': adding an RR at 'sebwindows.sebdomain.local' TXT
```

Sur Windows, les nouveaux fichiers de zones modifiés sont créés avec l'extension **.jnl** :

```
root@sebdns:/etc/bind/meszones# ls  
1.168.192.in-addr.arpa      zone.sebdomain.local  
1.168.192.in-addr.arpa.jnl zone.sebdomain.local.jnl
```

Sur le serveur DHCP, les logs expriment bien l'ajout d'un client :

```
Oct 13 10:25:47 sebdhcp dhcpd: Added new forward map from sebwindows.sebdomain.l  
ocal to 192.168.1.119  
Oct 13 10:25:47 sebdhcp dhcpd: Added reverse map from 119.1.168.192.in-addr.arpa  
. to sebwindows.sebdomain.local
```

Un **nslookup clientwindows** permet de vérifier que le client a bien été ajouté, et que l'on peut maintenant faire une résolution DNS sur son nom ou son adresse IP :

```
root@sebdns:/etc/bind# nslookup sebwindows  
Server:          192.168.1.116  
Address:         192.168.1.116#53  
  
Name:   sebwindows.sebdomain.local  
Address: 192.168.1.119
```

Dans le fichier **/var/lib/dhcp/dhcpd.leases**, on peut savoir quels baux notre service DHCP distribue :

```
GNU nano 2.2.6      Fichier : dhcpd.leases
lease 192.168.1.119 {
  starts 4 2016/10/13 08:30:48;
  ends 4 2016/10/13 08:40:48;
  cltt 4 2016/10/13 08:30:48;
  binding state active;
  next binding state free;
  rewind binding state free;
  hardware ethernet 08:00:27:82:c9:31;
  uid "\001\010\000'\202\3111";
  set ddns-rev-name = "119.1.168.192.in-addr.arpa.";
  set ddns-txt = "31fcd404e9e26354b6b7794fb52498c64";
  set ddns-fwd-name = "sebwindows.sebdomain.local";
  client-hostname "sebwindows";
}
```

Pour Linux :

Un ajout suffit sur le client (**avec les modifications du serveur**) au niveau du fichier de configuration DHCP client, au chemin `/etc/dhcp/dhclient.conf` :

```
send host-name "debianseb.sebdomain.local";
#send host-name = gethostname();
#send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
```

Nous relançons ensuite nos services DNS et DHCP (**dans cet ordre**) : **service bind9 restart** et **service isc-dhcp-server restart**. Puis, sur le client, nous pouvons redémarrer le service réseau jusqu'à obtenir une adresse IP de notre pool d'adresse :

```
root@debianseb:~# /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
root@debianseb:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b0:6a:0a
          inet adr:192.168.1.119  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fd23:6507:b29b:1:a00:27ff:feb0:6a0a/64  Scope:Global
```

Au final, les fichiers de zones sont directement modifiés sur notre serveur DNS avec l'écriture abrégée :

```

GNU nano 2.2.6      Fichier : zone.sebdomain.local
$ORIGIN .
$TTL 86400          ; 1 day
sebdomain.local    IN SOA  sebdns.sebdomain.local. root.sebdomain.local. (
                    2016092915 ; serial
                    86400    ; refresh (1 day)
                    21600    ; retry (6 hours)
                    3600000   ; expire (5 weeks 6 days 16 hours)
                    3600     ; minimum (1 hour)
                    )
                    NS      sebdns.
                    NS      esclavedns.
$ORIGIN sebdomain.local.
esclavedns         A       192.168.1.117
sebdhcp            A       192.168.1.118
sebdns             A       192.168.1.116
$TTL 300           ; 5 minutes
sebwindows         A       192.168.1.119
                   TXT      "31fcdf404e9e26354b6b7794fb52498c64"
$TTL 86400         ; 1 day
www                CNAME  sebdns
  
```

Les fichiers de logs des deux services attestent bien une nouvelle fois l'enregistrement d'une nouvelle entrée, sur le serveur DNS :

```

Oct 13 11:09:17 sebdns named[1093]: client 192.168.1.118#52126: updating zone 's
ebdomain.local/IN': adding an RR at 'debianseb.sebdomain.local.sebdomain.local'
A
Oct 13 11:09:17 sebdns named[1093]: client 192.168.1.118#52126: updating zone 's
ebdomain.local/IN': adding an RR at 'debianseb.sebdomain.local.sebdomain.local'
TXT
Oct 13 11:09:17 sebdns named[1093]: zone sebdomain.local/IN: sending notifies (s
erial 2016092918)
  
```

Et sur le serveur DHCP :

```

Oct 13 11:18:16 sebdhcp dhcpd: DHCPREQUEST for 192.168.1.119 from 08:00:27:b0:6$
$:0a (debianseb.sebdomain.local) via eth0_
  
```