I – Le service DNS :

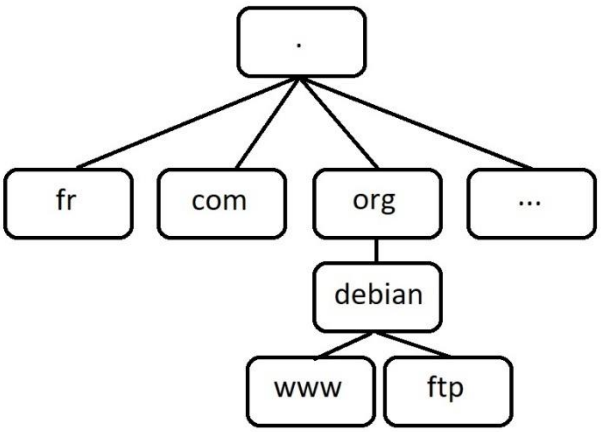
1 – Définition :

DNS (Domain Name Service) est un service de résolution de nom : il permet d'obtenir l'adresse IP correspondant au FQDN (Fully Qualified Domain Name) d'un hôte sur internet.

Par rapport aux autres systèmes de résolution de nom, DNS présente la particularité d'être distribué : il existe des millions de serveurs DNS sur internet.
Il est donc possible de mettre en place son propre serveur DNS.

2 – Notions de base du DNS :

Les noms de domaines sont organisés selon une hiérarchie dont le sommet, la racine, est représenté par un point :



Les domaines situés directement sous la racine sont appelés les TLD (Top Level Domains).

Chaque domaine peut avoir des sous-domaines.
Sur le schéma, le domaine *org* a un sous-domaine *debian* qui lui-même a deux sous-domaines *www* et *ftp*.

Un domaine peut gérer lui-même les résolutions de noms sur ses sous-domaines, mais il peut aussi confier cette gestion à d'autres serveurs : c'est ce qu'on appelle une délégation.
Dans ce cas, quand une requête arrive à un domaine pour un de ses sous-domaines délégués, la requête est transmise au serveur concerné.

L' ensemble formé d'un domaine avec ses sous-domaines non délégués, gérés par un serveur précis, s'appelle une zone.

Les TLD délèguent quasiment tous leurs sous-domaines : si une requête arrive pour résoudre www.debian.org, le TLD *org* va l'envoyer au serveur DNS de la zone *debian*.

On dit d'un serveur gérant une zone qu'il a autorité sur cette zone : c'est un authoritative server. On les différencie des recursive servers qui ne gèrent pas de zone, mais qui vont interroger des serveurs DNS pour résoudre les requêtes, et garder en cache les noms de domaines résolus.

La résolution DNS se fait en parcourant le nom de domaine de droite à gauche et en partant de la racine de la hiérarchie. Tout est basé sur la délégation : un nom de domaine doit avoir été correctement délégué dans le domaine de niveau supérieur pour pouvoir être résolu.

Exemple :

Pour résoudre ftp.debian.org sur un hôte :
Si le serveur DNS configuré pour l'hôte concerné n'a pas autorité sur la zone debian.org, alors il va interroger les serveurs DNS racine pour savoir qui contacter pour la zone org.
Puis, il va choisir un serveur qui lui dira quel serveur a autorité sur la zone debian.org.
Ce dernier lui retournera alors l'adresse IP de ftp.debian.org.

II – Un serveur DNS sous Linux :

1 – Logiciel :

Nous allons utiliser le serveur DNS le plus répandu sur Linux, BIND (Berkeley Internet Name Daemon) version 9.

Pour l'installer :

apt install bind9

Après installation, le répertoire /etc/bind contient tous les fichiers de configuration.

De plus, on installera le paquet dnsutils qui contient notamment l'outil de résolution de nom dig qui nous permettra de tester nos serveurs.

Le processus exécutant bind9 est appelé named.

NB : A chaque modification des fichiers de configuration de bind, il faudra forcer leur relecture en relançant le service bind9 : `systemctl restart bind9`

2 – Configuration réseau du serveur DNS :

Il faut configurer une adresse IP fixe pour le serveur afin que les clients puissent lui envoyer des requêtes.

Pour cela, on doit d'abord noter le nom donné par le système à l'interface réseau sur laquelle écoutera le serveur DNS.

On utilise la commande :

ip -4 -o addr

pour afficher les interfaces réseau de la machine.

On obtient alors la liste voulue avec les éventuelles adresses IP associées aux interfaces. L'interface lo étant la boucle locale (127.0.0.1), vous devez donc noter le nom de l'une des autres interfaces réseau.

Exemple :

```
esgi@debian7:~$ ip -4 -o addr
1: lo     inet 127.0.0.1/8 scope host lo\         valid_lft forever
preferred_lft forever
2: enp0s3     inet 100.0.2.6/24 brd 100.0.2.255 scope global dynamic
enp0s3\       valid_lft 809sec preferred_lft 809sec
esgi@debian7:~$
```

Ici, l'interface réseau sur laquelle écoutera le serveur DNS sera donc enp0s3.

Il nous faut maintenant configurer cette interface avec une adresse IP statique (et éventuellement configurer le serveur DHCP du même réseau pour qu'il ne distribue plus cette adresse).

Pour cela, on édite le fichier /etc/network/interfaces et on ajoute :

auto *nom_interface_réseau*
iface *nom_interface_réseau* inet static
address *adresse_IP_machine*
netmask *masque_sous_réseau*
gateway *adresse_IP_passerelle*
dns-nameservers *adresse_IP_DNS*

Exemple :

```
auto enp0s3
iface enp0s3 inet static
      address 100.0.2.6
      netmask 255.255.255.0
      dns-nameservers 127.0.0.1
```

3 – Configuration de la résolution de noms :

Il y a plusieurs façons de résoudre un nom :

- Grâce au fichier /etc/hostname qui contient le nom de la machine locale.
- Grâce au fichier /etc/hosts qui contient des noms de machines suivis de leur adresse IP. On y trouve notamment la ligne "127.0.0.1 localhost". On y place les machines importantes du réseau local.
- Grâce au service DNS.

C'est le fichier /etc/nsswitch.conf qui va déterminer l'ordre dans lequel ces méthodes vont être utilisées, et notamment la ligne hosts :

```
hosts:      files mdns4_minimal [NOT_FOUND=return] dns   myhostname
```

Elle indique que la recherche se fera d'abord dans /etc/hosts, puis si l'adresse se termine en .local, une recherche DNS multicast sera tentée, puis une recherche DNS classique sera tentée, et enfin le fichier /etc/hostname sera lu.

Dans le cas de la recherche DNS, pour les machines n'ayant pas d'IP fixe, le système utilisera le fichier /etc/resolv.conf pour récupérer l'adresse du ou des serveurs DNS à contacter.

Ce fichier est modifié par de nombreux services réseaux, notamment DHCP. Pour que ce soit votre serveur DNS qui soit entré dans /etc/resolv.conf, il faut modifier la configuration du client DHCP : celui-ci ne doit plus utiliser les adresses des serveurs DNS donnés par le serveur DHCP qui lui a répondu.

Dans le fichier /etc/dhcp/dhclient.conf, avant la ligne commençant par request, il faut donc ajouter la ligne suivante :

```
supersede domain-name-servers adresse_de_votre_DNS;
```

4 – Mise en place d'un serveur cache :

Ce type de serveur n'a autorité sur aucune zone : il interroge d'autres serveurs qui vont lui donner, s'il existe, le résultat de la résolution. Le serveur va alors mettre en cache la solution de la requête afin d'accélérer les recherches futures.

Tout ce dont a besoin un serveur cache, c'est du fichier *db.root* contenu dans /etc/bind.
Ce fichier contient les adresses des 13 serveurs DNS racines ayant autorité sur les TLD. Ils sont obligatoires pour que le serveur cache commence sa résolution.

C'est le plus simple à mettre en place des serveurs DNS.

On peut ensuite tester notre serveur avec une requête dig.

Exemple :

```
esgi@debian7:~$ dig www.debian.org

; <<>> DiG 9.10.3-P4-Debian <<>> www.debian.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53498
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
Question posée : Quel est l'adresse(A) IPv4(IN) de www.debian.org. ?
;www.debian.org.                IN     A

;; ANSWER SECTION:
Réponses : 130.89.148.14 et 5.153.231.4
www.debian.org.           257    IN     A      130.89.148.14
www.debian.org.           257    IN     A      5.153.231.4

;; AUTHORITY SECTION:
Serveurs DNS ayant autorité sur la zone debian.org (NS=NameServer) :
www.debian.org.           26921  IN     NS     geo1.debian.org.
www.debian.org.           26921  IN     NS     geo3.debian.org.
www.debian.org.           26921  IN     NS     geo2.debian.org.

Statistiques sur la recherche :
;; Query time: 1551 msec Temps de la requête
;; SERVER: 127.0.0.1#53(127.0.0.1) Adresse du serveur DNS : la machine locale
;; WHEN: Tue Apr 03 14:41:33 CEST 2018
;; MSG SIZE  rcvd: 132

esgi@debian7:~$
```

Le temps de résolution est de 1551 ms : notre serveur ne connaissait ni l'adresse de debian.org, ni celle de www.debian.org. Par contre, il a mis en cache toutes ces résolutions, ce qui permet de gagner du temps si elles se reproduisent.

Si on relance la même requête :

```
esgi@debian7:~$ dig www.debian.org

; <<>> DiG 9.10.3-P4-Debian <<>> www.debian.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28160
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.debian.org.                IN     A

;; ANSWER SECTION:
www.debian.org.           179    IN     A      130.89.148.14
www.debian.org.           179    IN     A      5.153.231.4

;; AUTHORITY SECTION:
www.debian.org.           26182  IN     NS     geo1.debian.org.
www.debian.org.           26182  IN     NS     geo3.debian.org.
www.debian.org.           26182  IN     NS     geo2.debian.org.

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Apr 03 14:53:52 CEST 2018
;; MSG SIZE  rcvd: 132

esgi@debian7:~$
```

On a un temps de résolution nul : toutes les informations nécessaires étaient déjà en cache.

Si on lance maintenant une requête sur ftp.debian.org :

```
esgi@debian7:~$ dig ftp.debian.org

; <<>> DiG 9.10.3-P4-Debian <<>> ftp.debian.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 16019
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ftp.debian.org.                IN     A

;; ANSWER SECTION:
ftp.debian.org.           22     IN     A      130.89.148.12

;; AUTHORITY SECTION:
debian.org.         22350  IN     NS     sec1.rcode0.net.
debian.org.         22350  IN     NS     dnsnode.debian.org.
debian.org.         22350  IN     NS     sec2.rcode0.net.

;; Query time: 18 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue Apr 03 14:56:53 CEST 2018
;; MSG SIZE  rcvd: 129

esgi@debian7:~$
```

On a un temps de 18 ms, nécessaire pour interroger le serveur ayant autorité sur debian.org pour obtenir l'adresse de l'hôte nommé ftp.

5 – Mise en place d'un serveur primaire :

Un serveur primaire (ou maître) est un serveur ayant autorité sur une zone. Il existe des serveurs secondaires, qui dépendent du serveur primaire et peuvent le remplacer en cas de panne.

Le serveur que nous allons configurer par la suite aura autorité sur la zone nation.esgi.

a – Configuration de la zone primaire :

Il va falloir définir un fichier pour la zone nation.esgi sur laquelle ce serveur aura autorité. Ce fichier contiendra notamment les correspondances entre noms de machines et adresses IP. Généralement, il se trouve dans /etc/bind et se nomme db.*<nom_de_la_zone>*, mais ce ne sont que des conventions, pas des obligations.
Nous nommerons notre fichier db.nation.esgi.

Il faut préciser dans /etc/bind/named.conf.local : la nature du serveur, le nom de notre zone et la localisation du fichier correspondant :

```
zone "nation.esgi" { nom de la zone
     type master;    serveur primaire
     file "/etc/bind/db.nation.esgi";nom et localisation du fichier
};
```

Il faut maintenant construire ce fichier de zone.

Il se compose tout d'abord d'une entête SOA (Start Of Authority) qui donne diverses informations sur le serveur et qui définit le comportement d'éventuels serveurs secondaires :

```
@       IN      SOA     ns1.nation.esgi. admin.nation.esgi. (
        2018040301      ; Serial
        3h              ; Refresh
        1h              ; Retry
        1w              ; Expire
        1h              ; Minimum
)
```

@ : remplace le nom de la zone nation.esgi
ns1.nation.esgi. : serveur primaire de la zone
admin.nation.esgi. : adresse mail de l'administrateur (l'@ est remplacée par un .)
Serial : indique le numéro de version du serveur : il doit être incrémenté à chaque modification, ce qui indique aux autres serveurs que leurs données doivent être mises à jour.
Refresh : indique le temps entre deux mises à jour des serveurs secondaires par rapport au serveur primaire.
Retry : indique le temps entre 2 tentatives de Refresh s'il y a échec.
Expire : indique le temps au bout duquel un serveur secondaire ne fait plus autorité s'il n'a pas réussi à contacter le serveur primaire.
Minimum : indique le temps durant lequel une réponse négative doit être conservée en cache.

Puis, on définit le nom du(des) serveur(s)de nom de la zone avec un(des) enregistrement(s) NS (Name Server):

@ IN NS nom du serveur DNS

Exemple :

```
@    IN    NS    ns1.nation.esgi.
```

NB : quand on donne un nom de machine, si on ne le termine pas par un point (.), alors il lui sera automatiquement ajouté le nom de la zone (ici nation.esgi).
On aurait pu écrire ici :

```
@    IN    NS    ns1
```

On va ensuite définir les correspondances noms de machines/adresses IP avec les enregistrements A (Address) :

nom_machine IN A adresse_IP

Exemple :

```
ns1   IN    A     10.0.2.15
www   IN    A     10.0.2.77
ftp   IN    A     10.0.2.100
```

Il est aussi possible de définir des alias pour certaines machines qui peuvent être connues sous différents noms, avec les enregistrements CNAME (Canonical Name)

autre_nom IN CNAME nom_initial

Exemple :

```
main  IN    CNAME      ns1
```

Ainsi, la résolution de main.nation.esgi ou de ns1.nation.esgi donnera le même résultat, 10.0.2.15.

Il existe d'autres types d'enregistrements utiles, deux sont souvent utilisés :

MX (Mail eXchanger) : permet de définir un ou des serveurs(s) mail de la zone. Le champ priorité indique l'ordre dans lequel les contacter s'il y en a plusieurs :

@ MX priorité nom_serveur_mail

AAAA : permet de définir des correspondances noms/adresse IPv6

b – Configuration de la zone reverse :

Pour des raisons de sécurité ou des impératifs de certaines applications (MySQL, Postfix, OpenSSH,…), les serveurs DNS définissent souvent une zone de résolution inverse qui va permettre de déterminer, à partir d'une adresse IP, le nom de la machine correspondante (rDNS). Cette résolution inverse est quasiment obligatoire pour les adresses privées, non routables sur internet.

Pour cela, il existe un TLD spécial nommé arpa qui contient la base de données reverse d'internet.

Nous devons définir une nouvelle zone dont le nom sera constitué du suffixe in-addr.arpa précédé par la partie réseau de l'adresse IP du réseau (par octets entiers) écrite de droite à gauche.

Exemples :

Pour le réseau 10.0.2.0/24, la zone reverse sera :
2.0.10.in-addr.arpa

Pour le réseau 12.0.0.0/8, la zone reverse sera :
12.in-addr.arpa

Pour le réseau 178.34.0.0/16, la zone reverse sera :
34.178.in-addr.arpa

On crée un fichier de zone pour cette zone reverse et on le déclare dans /etc/bind/named.conf.local, de la même façon que la zone primaire précédente.

Exemple :

```
zone "2.0.10.in-adddr.arpa" { nom de la zone
      type master;     serveur primaire
      file "/etc/bind/db.2.0.10.in-addr.arpa"; fichier de zone
};
```

Dans le fichier de zone reverse, on a un SOA de la même façon que dans la zone primaire :

```
@       IN      SOA     ns1.nation.esgi. admin.nation.esgi. (
        2018040301      ; Serial
        3h              ; Refresh
        1h              ; Retry
        1w              ; Expire
        1h              ; Minimum
)
```

De même, on déclare le serveur DNS de la zone avec un enregistrement NS.

Enfin, pour effectuer la résolution inverse, on doit déclarer des enregistrements PTR (PoinTeR)

partie_machine_de_l'IP IN PTR nom_machine

Exemples :

```
15    IN    PTR   ns1.nation.esgi.
77    IN    PTR   www.nation.esgi.
100   IN    PTR   ftp.nation.esgi.
```

On peut alors tester la résolution inverse avec l'option -x de dig (option +short pour un affichage court).

Exemple :

```
esgi@debian7:~$ dig -x 10.0.2.15 +short
ns1.nation.esgi.
esgi@debian7:~$
```

6 – Mise en place d'un serveur secondaire :

Il est prudent de définir un serveur DNS secondaire afin de palier l'éventuelle panne du serveur de nom. Il faudra alors le déclarer dans le fichier de configuration du client DHCP donc l'ajouter dans la directive supersede en le séparant avec une virgule de l'adresse du serveur primaire.

Un serveur secondaire télécharge les fichiers de zone du serveur primaire quand celui-ci les modifie.

La première étape consiste à déclarer ce serveur secondaire dans le fichier /etc/bind/named.conf.local de la machine qui l'héberge :

```
zone "nom_de_la_zone" {
        type slave;
        masters { adresse_IP_du_serveur_primaire; };
        file "nom_du_fichier_où_sera_sauvée_la_zone";
};
```

Déclarer un fichier de zone n'est pas obligatoire mais recommandé. Si ce n'est pas le cas, le serveur ne conserve les fichiers de zone qu'en mémoire et si on le redémarre, il ne fonctionnera pas tant qu'il n'aura pas retéléchargé les fichiers de zone.

Le fichier de zone que les serveurs secondaires récupèrent du serveur secondaire est par défaut en format RAW (binaire). Pour qu'il soit en format texte, il faut ajouter dans la déclaration de la zone, la directive :

masterfile-format text;

Vous aurez ainsi un fichier de zone qui sera une copie du fichier de zone du serveur primaire.

NB : Attention, le fichier de zone doit être dans un répertoire où le pseudo-utilisateur named a le droit d'écrire, donc pas /etc/bind. Le répertoire conseillé sous Debian est /var/lib/bind.

Il faut ensuite configurer le serveur primaire pour qu'il interagisse avec le serveur secondaire.

Tout d'abord, il faut lui dire de notifier le secondaire quand un des fichiers de zone est modifié. Pour cela, on ajoute la directive *notify yes* pour chaque zone dont on veut qu'elle soit reproduite.

Exemple :

```
zone "nation.esgi" {
          type master;
          file "/etc/bind/db.nation.esgi";
          notify yes;
};
```

On ajoute ensuite la liste des serveurs secondaires autorisés à effectuer un transfert de zone avec la directive allow-transfer.

Exemple :

```
zone "nation.esgi" {
        type master;
        notify yes;
        allow-transfer { 100.0.2.4; };
        file "/etc/bind/db.nation.esgi";
};
```

Pour tester tout cela, il faut modifier le numéro de série dans le SOA du serveur primaire et voir si le serveur secondaire le télécharge bien.
On peut ensuite mettre hors-ligne le serveur primaire et vérifier que les requêtes DNS sur les machines du réseau local sont bien résolues par le serveur secondaire.