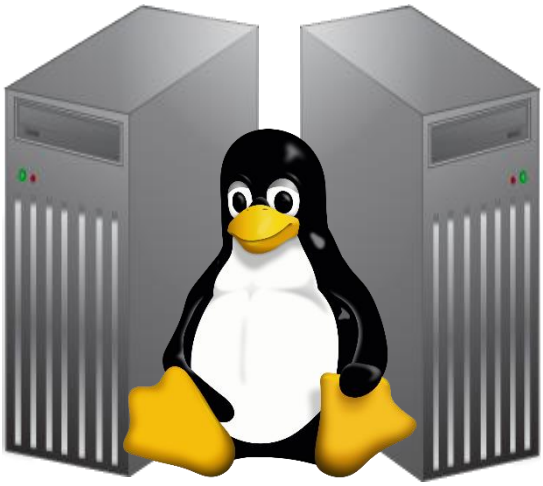
2021

Installation Server DNS en haute disponibilité sous Linux



Réalisé par :

Winness RAKOTOZAFY

AVANT - PROPOS

Qu'est-ce qu'un serveur DNS ? ou Domain Name System, est un service informatique généralement utilisé pour traduire les noms de domaine Internet en adresse IP. Un serveur DNS mal configuré peut amener à un problème de résolution de nom de domaine, mais aussi générer à une erreur DNS où la machine client ne serait plus en mesure de se connecter au réseau, nuisant à son fonctionnement dans le cadre d'une activité en entreprise.

Dans le cadre de ce compte-rendu, nous allons procéder à la mise en place de serveur DNS sous Linux, une famille de système d'exploitation open source de type Unix, fondé sur le noyau Linux créé en 1991. Plus spécifiquement, cette configuration et cette installation se basera sur une distribution du GNU/Linux, en Debian sans GUI (Interface graphique) pour les serveurs, et encore sous Debian avec GUI pour la machine client. Vous verrez plus bas les paramètres des matériels de nos machines, le tout sur des VMs (Virtual Machines) du cluster VSphere de Webforce3.

Qu'est-ce qu'un serveur DNS en haute disponibilité ? Les serveurs DNS en haute disponibilité permettent à l'architecture informatique de continuer à fonctionner si un serveur est en panne, afin que l'autre serveur prenne le relais pour maintenir la résolution DNS dans le domaine. En effet, dans une architecture informatique, il est toujours préférable d'avoir plus d'un serveur pour éviter cette « panne générale », qui troublerait le bon fonctionnement des services d'une entreprise, la requête des utilisateurs mais surtout rendre plus flexible la résolution de DNS.

Nous allons procéder à une installation de deux serveurs DNS pour communiquer entre eux, et de permettre la résolution DNS sur notre machine client Linux. Pour faire ci, nous allons mettre en place l'architecture réseau de notre système d'information, avec deux serveurs DNS 1 et DNS 2, et une machine client, le tout relié à un pfsense, qui sera notre firewall. Ensuite, dans le cadre de ce compte-rendu, nous allons déterminer les étapes d'installation des serveurs DNS sous GNU/Linux, détailler avec les commandes qui correspondent et enfin vérifier la configuration et le bon fonctionnement de notre installation et notre mise en place des serveurs DNS.

NB : Comme les machines sont tous en root, ils auront toutes les permissions nécessaires pour l'installation des paquets, changement de nom d'hôte, modification de fichier de configutation, etc…

INFRASTRUCTURE INFORMATIQUE

Comme dit précédemment donc, notre infrastructure sera composée de deux serveurs DNS sur GNU/Linux, que l'on nommera DNS 1 et DNS 2, qui assurera la résolution DNS pour notre machine client ; une machine client Linux, ; un switch pour relier les trois (3) machine et le firewall ; un Firewall qui sera notre passerelle par défaut pour communiquer au WAN (Wide Area Network). Cette infrastructure sera présentée comme le schéma ci-dessus, suivi des tableaux de configuration des matériels des machines de l'infrastructure, et de l'adressage IP de toute l'architecture :

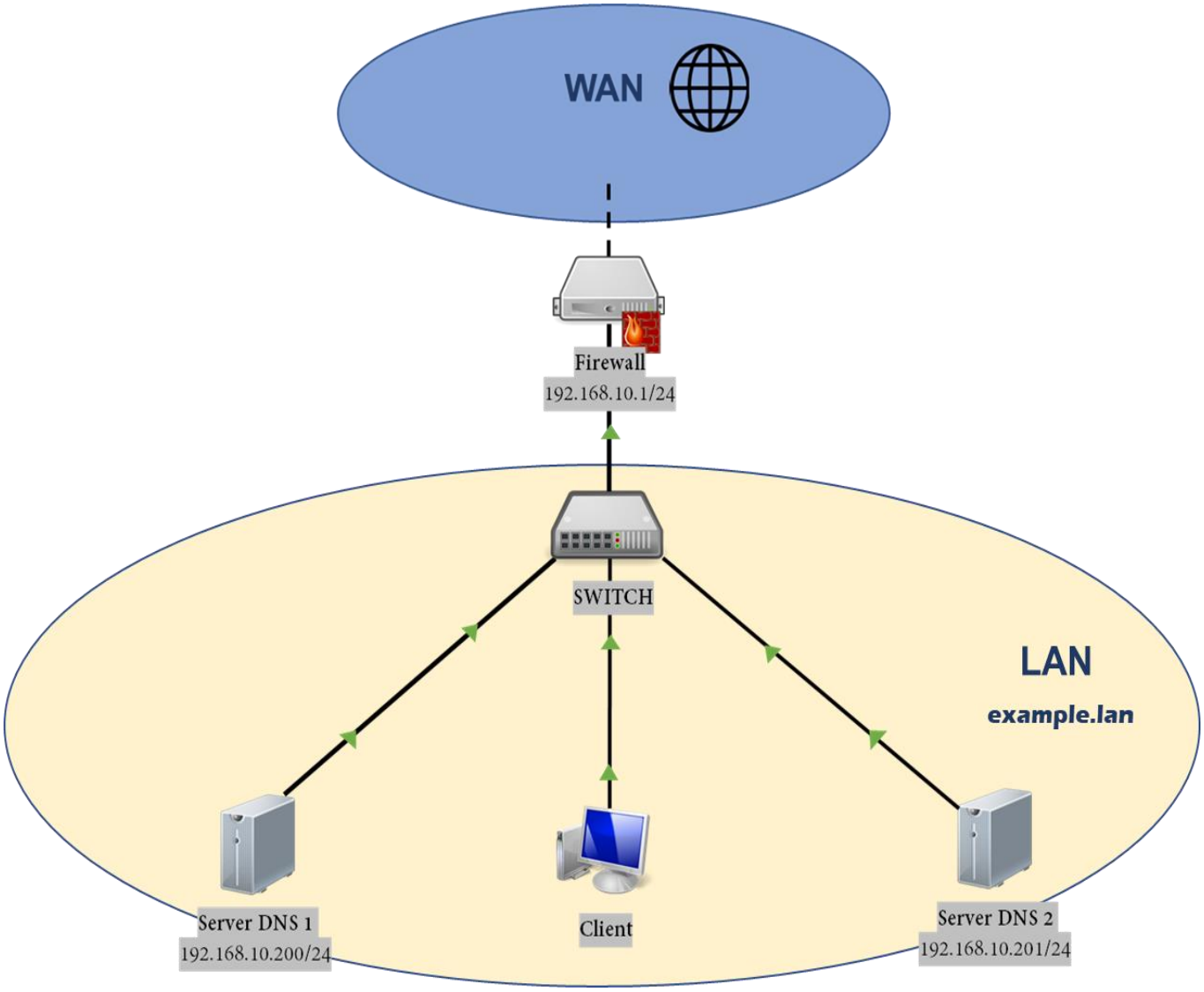


Schéma 1 : Infrastructure réseau de notre système d'information

Tableau 1 : Adressage IP de l'infrastructure

Matériels	Hostname	Adresse IP	Masque de sous-réseau	Passerelle	Rôle
Serveur 1	DNS1	192.168.10.200	255.255.255.0	192.168.10.1	Master DNS
Serveur 2	DNS2	192.168.10.201	255.255.255.0	192.168.10.1	Slave DNS
Client Linux	Client	En DHCP	255.255.255.0	192.168.10.1	Client
Firewall	pfsense	192.168.10.1	255.255.255.0	-	Firewall

Dans le cadre de notre installation de serveur DNS, on mettra en place une adressage IP statique pour les machines de notre infrastructure en /24, et comme passerelle par défaut, notre firewall, le pfsense avec l'adresse 192.168.10.1. Seul la machine client aura

Tableau 2 : Paramètres des matériels des machines

Machine	CPU	RAM	Stockage	Adapteur réseau
Serveur 1	1	1 Go	8 Go	Switch LAN
Serveur 2	1	1 Go	8 Go	Switch LAN
Client Linux	2	4 Go	32 Go	Switch LAN
Parefeu	1	1 Go	4 Go	- Internet WAN - Switch LAN

Les trois machines : le serveur 1, le serveur 2 et la machine client sont paramétrés comme tels, pour accéder au réseau via un switch dans le réseau LAN avec un câble Ethernet. Ce switch se connectera au pare-feu (pfsense) également avec un câble Ethernet, et c'est avec ce dernier que le reste se connectera à Internet WAN. Le pfsense jouera également le rôle de routeur, qui permettra la connexion entre les deux serveurs et la machine client au WAN qui sera Internet.

TABLE DES MATIERES

I- INSTALLATION DU SERVEUR DNS 1

Etape 0 : Prérequis pour une configuration optimale

- *Phase de login*

Pour commencer, nous allons déployer notre machine Master Debian, qui sera notre Master DNS et que l'on nommera DNS 1. Arrivé à l'écran de l'ouverture de session, nous allons renseigner comme utilisateur « root » qui aura toutes les permissions nécessaires pour la configuration. Les informations de login par défaut sont :

```
Debian GNU/Linux 10 debian tty1

Debian login: root
Password: toor
```

Il est à noter que lorsque l'on tape un mot de passe sous le terminal Linux, on ne verra pas des chaines de caractères « **** », mais il ne faut pas s'inquiéter, car le terminal a pris en compte la saisie des mots de passes 🤓.

- *Installation d'un éditeur de texte*

Il existe plusieurs éditeurs de texte sur les distributions Linux : Vim, Nano, Gedit, Kate, etc…. Mais pour notre cas pratique, nous allons installer Vim comme éditeur de texte, pour modifier nos fichiers de configuration, notamment avec les avantages qu'il procure en termes de maniabilité et de fonctionnalités. Pour faire ceci, voici les commandes :

```
root@debian:~# apt install vim -y
```

Le suffixe -y est utilisé pour confirmer les messages de confirmation lors de l'installation des paquets. Maintenant que nous avons un éditeur de texte, nous pouvons commencer la configuration de notre machine serveur DNS.

- *Installation du package bind9 et dnsutils*

Avant de commencer il nous faudrait installer le Bind9 (Berkley Internet Naming Daemon), le serveur DNS le plus utilisé sur Internet, spécialement sur les systèmes d'exploitation de type Unix. Pour procéder à l'installation on rentre la commande `apt-get install [nom du paquet].`

Ici ce sera bind9, et en plus de cela, nous installerons également le paquet dnsutils, qui nous fournira des outils très pratiques pour tester et débugger le service DNS.

```
root@debian:~# apt-get install bind9 -y
root@debian:~# apt-get install dnsutils -y
```

La commande est un peu longue non ? Sachons que nous pouvons installer deux paquets sur une seule une même commande.

```
root@debian:~# apt-get install bind9 dnsutils -y
```

Maintenant que l'on a terminé d'installer les pré-requis nous pouvons maintenant entrer dans la partie intéressante, celle de la configuration du serveur DNS.

Etape 1 : Configuration de base du serveur Debian Master

- *Changement du nom d'hôte de la machine serveur*

Le changement de nom d'hôte de la machine est nécessaire pour éviter toutes confusions de configuration avec les autres machines. Pour parvenir à cela, voici la commande à entrer dans le terminal vim `/etc/hostname`:

```
root@debian:~# vim /etc/hostname

debian
~
~
~
~
~
~

---VISUEL---
```

On entre donc dans le fichier de configuration, et dans Vim, notre éditeur de texte, on a deux modes : le mode `VISUEL` et le mode `INSERTION`. Pour entrer dans le mode `INSERTION`, il faut taper sur la touche I comme insertion :

```
root@debian:~# vim /etc/hostname
```

Arrivé dans le fichier de configuration, il faudrait modifier le nom de « debian » en « dns1 »

```
dns1
~
~
~
~
~
~

---INSERTION---
```

Après avoir remplacé le nom d'hôte de la machine, on tape `Echap` du clavier pour sortir du mode INSERTION, et dans le mode `VISUEL` entrer `:x` pour quitter et sauvegarder les modifications du fichier de configuration. Ensuite, pour appliquer le nouveau fichier de configuration sur le nom d'hôte, il faudrait saisir cette commande sur le terminal, puis se déconnecter `(exit)` et se reconnecter :

```
root@debian:~# hostname -F /etc/hostname
```

- *Adressage IP du serveur DNS 1*

On configure le fichier de configuration de l'interface réseau avec l'éditeur de texte Vim, et le fichier de configuration des interfaces réseaux se trouve dans le dossier `/etc/network/interfaces.` Donc la commande à saisir est comme ceci :

```
root@dns1:~# vim /etc/network/interfaces
```

Arrivé dans le fichier de configuration de l'interface réseau, voici les modifications à faire : changer l'adresse IP (DHCP par défaut) en IP statique, et renseigner le masque de sous-réseau `/24` et la passerelle par défaut, qui est notre pfsense `192.168.10.1.` (A ne surtout pas oublier !)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
iface ens192 inet static
        address 192.168.10.200/24
        gateway 192.168.10.1
```

Après avoir changé les configurations dans Vim, on quitte et on sauvegarde le fichier de configuration, qui est la même commande, et sera toujours notre commande pour quitter et sauvegarder nos fichiers de configuration `:x`. Ensuite il faudrait redémarrer le service network, et réactiver l'interface `ens192`, l'interface réseau de notre serveur DNS1, comme ci-dessous :

```
root@dns1:~# systemctl restart networking
root@dns1:~# ifup ens192
```

Pour vérifier que l'adressage IP a été bien réalisée, on fait une vérfication avec la commande `ip a` sur le terminal. Le résultat devrait être comme-ci-dessous :

```
root@dns1:~# ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever

2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 00:50:56:9e:bf:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.200/24 brd 192.168.10.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9e:bfd4/64 scope link
       valid_lft forever preferred_lft forever
```

Etape 2 : Configuration du service DNS sous Linux

- *Redirection de dossier et modification de notre ficher database `db`*

Pour configurer notre serveur DNS, nous nous redirigerons vers notre bind de notre serveur. Pour parcourir un dossier, la commande à entrer est `cd [chemin du dossier]`.

```
root@dns1:~# cd /etc/bind
```

Arrivé dans le dossier bind, on copiera une configuration vierge le `db.empty`, que l'on nommera en rapport au nom de notre domaine, donc `db.example.lan`. Cette copie nous sera nécessaire car nous aurons besoin du fichier db.empty si jamais nous voudrions restaurer par défaut les paramètres, en raison d'un bug ou problème trop difficile à régler. Donc, pour faire cette copie, voici la commande :

```
root@dns1:~# cp db.empty db.example.lan
```

Cette commande est facile à comprendre, on essaie jsute de dupliquer le fichier du 1er paramètre `db.empty` que l'on nommera au 2nd paramètre `db.example.lan`.

A présent, notre fichier database créé, il est temps de le modifier, et comment modifier un fichier ? En utilisant Vim ! Notre éditeur de texte

```
root@dns1:~# vim db.example.lan
```

- *Configuration du fichier `db.example.lan`*

A présent, dans le fichier de configuration `db.example.lan` , on supprime toutes les occurrences localhost en notre nom de domaine `example.lan`. Sur la première ligne qui suit, on y entre l'information de notre premier server Dns au lieu de localhost donc dns1.example.lan. **!! N'OUBLIONS SURTOUT PAS LE POINT A LA FIN DU NOM DE DOMAINE !!**, car cela peut engendrer des erreurs lors de la lecture du fichier `db`, notamment sur le fait que `lan` est le top level domain, et que sans le point après le `lan,` le DNS comprendra dns1.example.lan.example.lan mais non dns1.example.lan. En occurrence, nous pouvons ajouter à la deuxième ligne après le Dns1, le nom du domaine Dns2 c'est-à-dire dns2.example.lan

```
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL    86400
@       IN      SOA     example.lan. root.example.lan. (
                         202102253      ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      dns1.example.lan.
@       IN      NS      dns2.example.lan.
```

On va détailler les modifications faites : comme dit précédement, il fallait changer les localhost en notre nom de domaine, c'est-à-dire `example.lan`, ne jamais oublier le point après lan, sinon confusion du DNS lors de la résolution du nom. Et enfin changer le serial par défaut (1) en des digits reconnaissables. Pour notre cas, et dont on conseille le plus, c'est de changer le Serial en format de date de la modification + un chiffre entre 0-9. Par exemple si la modification a été effectué

le 25 Février 2021, le nouveau Sérial sera donc `20210225x` où x sera un chiffre compris entre 0-9. Dans cet exemple, on a choisi le chiffre 3. Cette incrémentation est plus que nécessaire pour pouvoir appliquer les modifications du fichier `db.example.lan.`

- *Ajout des entrées DNS dans le fichier db.example.lan*

Dans le même fichier donc, on y ajoutera deux entrées DNS, le DNS1 avec l'adresse IP `192.168.10.200` et le DNS2 `192.168.10.201.` Il faudrait les ajouter avec le format comme ci-dessous

```
dns1        IN      A       192.168.10.200
dns2        IN      A       192.168.10.201
```

Pour détailler cette modification donc, `dns1` et `dns2` sont des noms d'hôtes, `IN` signifie Internet, `A` est la correspondance, et la 4è colonne est l'adresse IP correspondante. Comme résultat final du fichier donc, on peut ajouter d'autres entrées pour les tests de résolution à effectuer en finalité, mais avec des adresses IP toutes différentes et donc, le fichier db.example.lan final devrait se présenter comme ceci :

```
; BIND reverse data file for empty rfc1918 zone
;
; DO NOT EDIT THIS FILE - it is used for multiple zones.
; Instead, copy it, edit named.conf, and use that copy.
;
$TTL    86400
@       IN      SOA     example.lan. root.example.lan. (
                         202102253      ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      dns1.example.lan.
@       IN      NS      dns2.example.lan.

dns1    IN      A       192.168.10.200
dns2    IN      A       192.168.10.201
www     IN      A       192.168.10.150
mail    IN      A       192.168.10.101
r2d2    IN      A       192.168.10.142
jcvd    IN      A       192.168.10.145
```

Après avoir quitté et enregistrer le fichier `db.example.lan` avec un `:x`, il faudrait recharger le service bind9 avec la commande `systemctl reload bind9` mais non `restart` afin de ne pas arrêter le service, et pour vérifier que le service est bien actif, un petit `systemctl status bind9` sera adéquat pour vérification, et nous devrions avoir le résultat comme tel

```
root@dns1:~# systemctl reload bind9
root@dns1:~# systemctl status bind9
```

```
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset:
   Active: active (running) since Thu 2021-02-25 13:31:32 CET; 1 day 23h ago
     Docs: man:named(8)
  Process: 497 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCC
  Process: 540 ExecReload=/usr/sbin/rndc reload (code=exited, status=0/SUCCES
 Main PID: 500 (named)
    Tasks: 4 (limit: 1146)
   Memory: 33.7M
   CGroup: /system.slice/bind9.service
           └─500 /usr/sbin/named -u bind
```

Etape 3 : Configuration du fichier local du DNS

Cette configuration est requise afin d'indiquer le fichier `named.conf.local` où se trouvent les informations que l'on a configuré précédemment, et d'indiquer au domaine que ce serveur est le serveur Master. Ce fichier se situe dans le répertoire `/etc/bind/named.conf.local.` On peut directement le modifier avec vim en indiquant la destination du fichier comme ceci.

```
root@dns1:~# vim /etc/bind/named.conf.local
```

- *Attribution des rôles du DNS1*

Arrivé dans le fichier de configuration les lignes à ajouter sont le nom de la zone, la destination du fichier où on récupère les informations, le type de serveur (master pour DNS1), une ligne pour permettre la notification entre les autres serveurs, et de permettre les transferts d'informations vers un autre DNS. Dans notre cas, on l'aura deviné que l'autre DNS est bien le DNS2 avec adresse IP `192.168.10.201`.

Point important ! La syntaxe est importante dans ce fichier de configuration, une erreur peut nuire à la résolution DNS, donc faudrait bien les appliquer et faire attention dessus. Le fichier de configuration devrait ressembler comme ceci :

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "example.lan" {
        file "/etc/bind/db.example.lan";
        type master;
        notify yes;
        allow-transfer { 192.168.10.201; };
};
```

- *Finalisation de la configuration*

Et voilà ! Nous avons terminé la configuration du serveur DNS1 `Master`, il nous reste qu'à démarrer le service bind9 avec `systemctl start bind9` pour s'assurer qu'il est bien activé et il est prêt pour le déploiement. . . .

```
root@dns1:~# systemctl start bind9
```

Quoique, comme nous n'avons pas encore configurer dans la machine client, ce serveur, nous ne pouvons pas procéder à la phase de test. Ainsi, passons donc à la configuration du deuxième serveur DNS et c'est à la fin que nous ajouterons les deux serveurs dans la configuration de la machine cliente, pour tester la résolution DNS ;

II- INSTALLATION DU SERVEUR DNS 2

Etape 0 : Prérequis pour la configuration

Avant de commencer la configuration du deuxième serveur DNS, les informations clés sont que, le deuxième serveur DNS, est le serveur `Slave` du premier serveur DNS, et donc, on a plus besoin de configurer le service DNS sous le fichier `db.example.lan` dans le répertoire `bind`, car il héritera des configurations du serveur `Master`

Donc cette étape sur les prérequis est exactement la même que le serveur DNS1, et donc référons-nous au *Etape 0 : Prérequis pour* **une configuration optimale** (Cf.page 5).

Ensuite, songer à **INSTALLER VIM !** Etape très importante pour modifier les fichiers de configuration, plus besoin de réexpliquer l'importance que Vim a apporté par rapport aux autres éditeurs de texte. Mais comme on dit, à chacun ses goûts donc vous pouvez installer l'éditeur de texte qui vous convient le mieux, mais nous conseillons à travers de ce manuel, l'utilisation de Vim

Enfin, les commandes d'installation restent toujours la même « `apt-install {nom du paquet}` » donc n'oubliez pas d'installer les paquets bind9 et dnsutils pour la configuration du serveur.

Etape 1 : Configuration de base du serveur Debian Slave

La méthode de configuration sur le serveur reste similaire au serveur 1. Mais attention les informations qui changent sont le nom d'hôte, pour éviter la confusion sur la manipulation des deux serveurs. Comme nom d'hôte donc, mettons `dns2` pour le deuxième serveur. Ensuite pour l'adresse I'IP, nous prendrons la référence au *Tableau 1 : Adressage IP de l'infrastructure*, pour connaitre quelle adresse IP mettre pour le deuxième serveur, ainsi que sa passerelle par défaut, et son masque de sous-réseau. Si jamais nous ne nous souvenons pas des commandes à saisir pour le changement de `hostname` et d'`adresse IP`, référons-nous à l'*Etape 1 : Configuration de base du serveur Debian Master*, mais en saisissant les bons paramètres.

Etape 2 : Configuration du fichier local du DNS

On remarquera que l'on n'a pas configuré un fichier db.example.lan sur le serveur DNS2 ; en effet, car le DNS2 étant le serveur `Slave` de notre infrastructure, il héritera du fichier de configuration du DNS1 qui est le serveur `Master`. Par contre, on aura des modifications à faire sur le fichier local du serveur DNS2, car c'est dans ce fichier que l'on attribuera le rôle au serveur, et que l'on paramétrera, quel domaine il devrait communiquer qui est `example.lan`.

- *Attribution du rôle du DNS2*

Tout comme le serveur DNS1, on renseigne de nouveau le nom de domaine ou zone, le chemin dans lequel le fichier de configuration du DNS1, le serveur Master `(db.example.lan)` dont il utilisera et en appliquera les paramètres. Ce fichier se trouvera cette fois-ci dans le répertoire `/var/cache/bind/db.example.lan.` Pour le rôle du serveur, sous type, on y affecte le rôle de Slave pour le serveur DNS2. Ensuite, on autorisera la notification de la part du DNS1 tout en ajoutant le serveur `Master` de notre domaine dans notre fichier local de configuration. Enfin, tout comme dans le premier serveur DNS, **ATTENTION A LA SYNTAXE !** et bien évidemment nous allons faire recours à notre éditeur de texte Vim pour la configuration. Et donc, notre fichier local de configuration DNS du serveur DNS2 devrait ressembler comme ceci

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "example.lan" {
        file "/var/cache/bind/db.example.lan";
        type slave;

        allow-notify { 192.168.10.200; };

        masters { 192.168.10.200; };
};
```

A présent, tout comme pour le premier serveur, nous devons faire une actualisation du service bind9, pour que ce dernier charge les changements effectués au niveau de la configuration du service DNS du serveur DNS2.

```
root@dns2:~# systemctl reload bind9
```

Et voilà ! On a terminé les configurations de nos deux serveurs DNS, il nous faut à présent vérifier la réplication entre les deux serveurs DNS, et tester si la machine cliente reçoit la résolution DNS provenant des deux serveurs, qui est vraisemblablement dit l'intérêt du serveur DNS en haute disponibilité.

III- TESTS DE LA RESOLUTION DNS

Etape 1 : Ajout des deux serveurs dans la machine cliente

Pour ajouter les deux serveurs que la machine cliente devra contacter, il nous faut modifier le fichier `resolv.conf` qui se situe dans le répertoire `/etc/resolv.conf`, « `resolv` » comme résolution du nom de domaine. Pour procéder ainsi à la modification, il faudrait avoir `vim` également sur la machine cliente, donc pensez à l'installer sur la machine cliente également.

Cependant, comme la machine cliente n'est pas en `root` par défaut, on ne peut pas installer les paquets sans être l'utilisateur `root`. Pour se connecter en root donc, il vous faut entrer la commande `su -` et saisir ensuite votre mot de passe de l'utilisateur `root`.

```
webforce3@wr-linux:~$ su -
Mot de passe :
```

NB : On remarque que si après le nom de votre le symbole est « $ » et non « # », c'est que nous n'avons pas les droits et permissions d'administrateurs.

Donc maintenant, après avoir installé Vim sur notre machine cliente, il est temps de modifier le fichier de résolution de domaine, la commande et les modifications sur le fichier de configuration devrait être comme ci-dessous :

```
root@wr-linux:~# vim /etc/resolv.conf

# Generated by NetworkManager
search localdomain
nameserver 192.168.10.200
nameserver 192.168.10.201
~
~
~
~
---VISUEL---
```

Donc pour expliquer, on a ajouté les deux serveurs comme étant les serveurs à contacter pour la résolution DNS, et que nous avons supprimé le pfsense `192.168.10.1` qui était celui par défaut.

Toutefois, il est à savoir que si NetworkManager est installé sur la machine cliente, la configuration faite pourrait ne pas être persistante, c'est-à-dire qu'elle redeviendra ce qu'elle était par défaut au redémarrage, et dans des cas comme ceci, il faudrait configurer cela sous le GUI (Interface graphique) du pfsense.

A présent, nous allons configurer l'envoie des requête DNS à des serveurs externes puis commencer nos phases de test de résolution de nom DNS sur notre machine cliente, à commencer sur la résolution DNS sur le server master.

Etape 2 : Envoie de requête DNS à des serveurs externes

Cette étape n'est pas une simple vérification, mais une configuration à faire en final sur les deux serveurs. En effet, si l'on voulait que nos serveurs DNS envoient la requête DNS à un autre serveur, dans l'optique où les deux serveurs ne sauront pas résoudre, il faudrait modifier le fichier de configuration situé dans le répertoire `/etc/bind/named.conf.options`. Cette configuration est indispensable pour naviguer sur le web car nos serveurs DNS actuels ne savent résoudre que des DNS locaux.

```
root@dns1:~# vim /etc/bind/named.conf.options

options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        //  Uncomment  the  following  block,  and  insert  the  addresses
replacing
        // the all-0's placeholder.

        // forwarders {
        //      8.8.8.8;
        // };
```

Ici donc, on a remplacé le forwarders par un autre DNS, celui de google `8.8.8.8` afin de renvoyer la requête vers le serveur DNS de google si notre DNS ne sait pas la résoudre. **!! ATTENTION CETTE MODIFICATION EST A FAIRE SUR LES DEUX SERVEUR DNS !!**

Ensuite, comme on n'aime bien que tout se passe bien, on n'oublie pas de faire une actualisation du service bind9 pour que le serveur prenne en compte les modifications effectuées :

```
root@dns1:~# systemctl reload bind9
```

Etape 3 : Test de la résolution DNS sur le serveur master DNS1

Il est temps de passer au test de résolution DNS, pour faire ceci la commande clé est le `dig`, comme creuser / chercher en français. Ainsi, la commande à saisir devrait prendre forme comme ceci :

`dig @192.168.10.200 + short {nom à tester}`, pour tester notre DNS1. Donc effectivement pour le test du DNS2, l'IP à `dig` sera celle du serveur DNS2 c'est-à-dire `192.168.10.201`.

```
webforce3@wr-linux:~$ dig @192.168.10.200 + short dns2.example.lan
>192.168.10.201
webforce3@wr-linux:~$ dig @192.168.10.200 + short dns1.example.lan
>192.168.10.200
webforce3@wr-linux:~$ dig @192.168.10.200 + short jcvd.example.lan
>192.168.10.145
```

Si la machine reçoit une adresse IP, identique aux IP d'entrée DNS que l'on a mis dans notre fichier de configuration `db.example.lan` du serveur DNS1 (Voir Ajout des entrées DNS dans le fichier db.example.lan), ce qui est notre cas, c'est que nous avions réussi la configuration du serveur DNS 1, bravo 🎉 !

Etape 4 : Test de la résolution DNS sur les serveur slave DNS2

Donc tout comme le serveur Master DNS1, on test la résolution avec la commande dig, donc commençons :

```
webforce3@wr-linux:~$ dig @192.168.10.201 + short dns1.example.lan
>192.168.10.200
webforce3@wr-linux:~$ dig @192.168.10.201 + short dns2.example.lan
>192.168.10.201
webforce3@wr-linux:~$ dig @192.168.10.201 + short www.example.lan
>192.168.10.150
```

Tout comme le test précédent, on a testé en cherchant dans le domaine du server slave, les noms à tester, et oui, si on a bien suivi la configuration, on a reçu les adresses IP de chacun, notre server DNS2 est fonctionnel, encore bravo 🎉 !

Etape 5 : Vérifications supplémentaires

Après avoir réussi à configurer à bien les deux serveurs DNS, on peut également vérifier les logs sur nos serveur DNS. Les logs se situent dans le répertoire `/var/log /daemon.log`, mais pour éviter de voir tous les logs du pc, on peut juste vérifier les derniers logs de notre machine, avec la commande `tail`, pour afficher les dernières lignes du fichier `daemon.log`.

On peut également vérifier les logs en affichant les nouvelles lignes du fichier log avec le paramètre `-f` sur `tail`, donc la commande à saisir au final serait :

```
root@dns1:~# tail -f /var/log/daemon.log
```

Et voilà ! On vous félicite d'avoir atteint cette dernière étape, suivez bien ces instructions, et vos serveurs DNS sous Linux devrait bien fonctionner sans problèmes.

CONCLUSION

En guise de conclusion donc, l'intérêt de cette infrastructure est lié aux besoins actuels des entreprises, car actuellement, sans serveur en haute disponibilité, lorsqu'une panne surgirait de nulle part, non seulement cela pénalisera toute l'activité de l'entreprise par la perte des données en cours, mais elle augmentera également ses coûts en matière de réparations, et pire encore, perdra en crédibilité par rapport à ses clients.

Pourquoi installer un serveur DNS sous Linux ? Cette question peut surgir dans nos réflexions, notamment car selon une étude de l'INSEE, Windows domine le marché du poste de travail depuis plus de 20 ans, car il serait plus user-friendly en comparaison de Linux, mais question Cloud et Web, c'est bien Linux qui est le standard de fait.

Du côté serveur, les configurations étant moins exposées, moins visibles et les parts de marché bien plus difficiles à mesurer, mais après avoir essayé d'installer un serveur DNS en haute disponibilité sous Windows et sur Linux, mon opinion personnelle sur la question est qu'il serait préférable de faire recours à Linux que Windows.

Tout d'abord, ne serait-ce le prix de la licence d'un serveur Windows, le point fort de Linux est dans sa liberté d'exploitation en tant qu'Open Source, (donc bien évidemment moins de coût que par rapport à Windows). Aussi, lors de nos expériences, il nous a été plus facile d'installer un serveur Linux que Windows, malgré l'absence d'un GUI du côté Linux, sa stabilité domine clairement les erreurs fréquentes que proposait Windows lors de sa mise en œuvre.

Cependant, nous tenons à dire que chacun a sa préférence en matière de système d'exploitation, ce débat pourrait continuer encore des heures et des heures, mais je tiens personnellement clore ce rapport sur ce point et je vous remercie d'avoir pris la peine de lire les étapes de configuration de serveur Linux en haute disponibilité pour mon travail personnel.