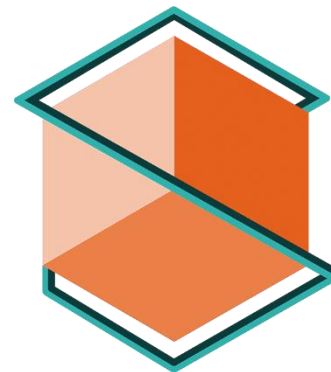


LINUX DHCP-DNS



Semifir

contact@semifir.com
13 Avenue du Président John F. Kennedy,
59000 Lille.

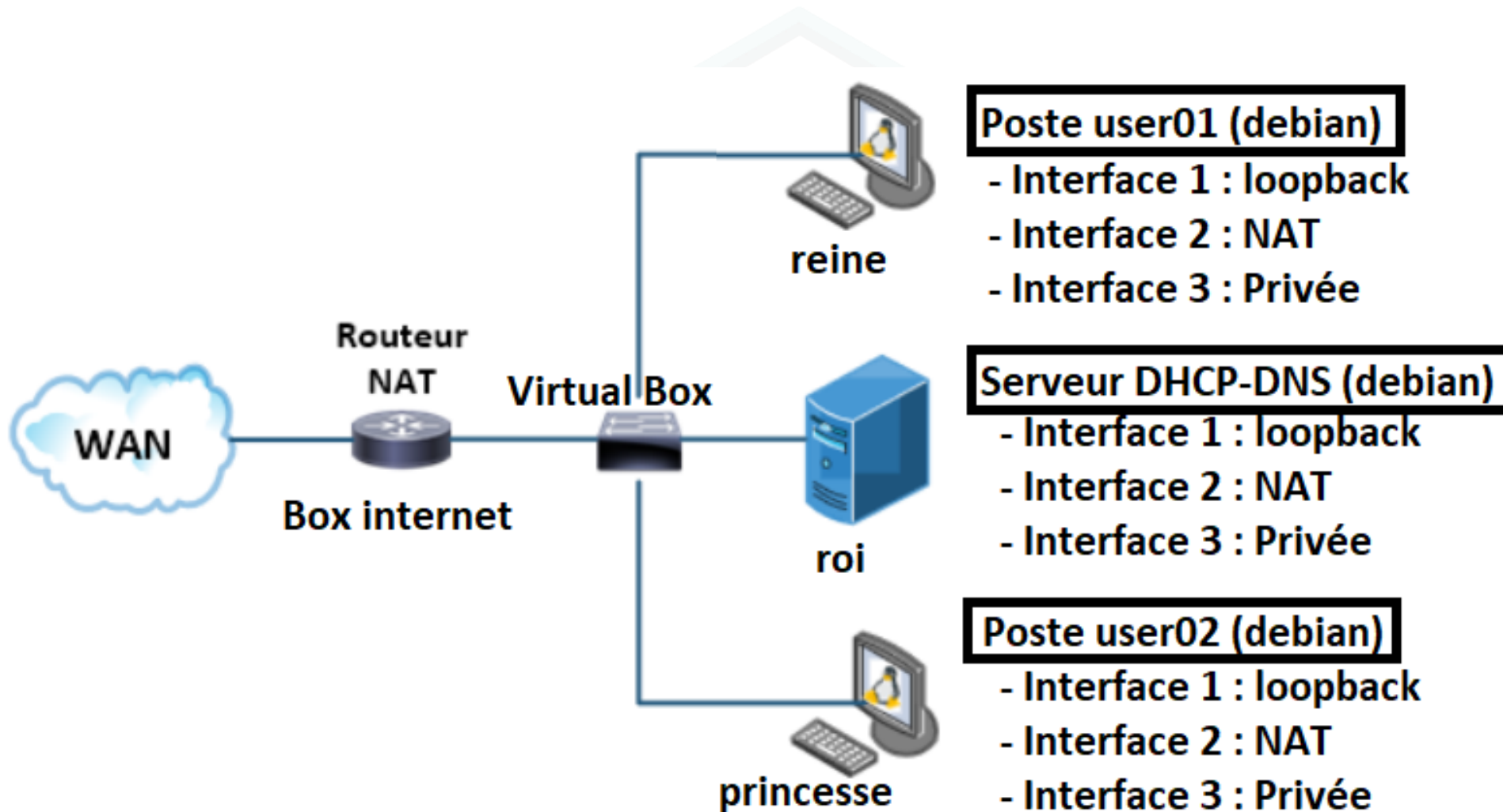
Objectifs de ce module

- ✓ Rappel des prérequis
- ✓ Savoir le vocabulaire et termes concernant DNS (complément du rappel)
- ✓ Savoir les composants de Bind9
- ✓ Savoir la syntaxe des fichiers de configuration DNS Bind
- ✓ Savoir implémenter et configurer un serveur DNS et postes clients
 - Savoir et configurer les prérequis nécessaires avant l'installation du DNS Maître
 - Savoir bloquer/débloquer les entrées des serveurs DNS dans `/etc/resolv.conf`
 - Savoir installer un serveur DNS Maître avec Bind9
 - Savoir configurer un serveur DNS Maître avec Bind9
 - Savoir configurer un serveur DNS Maître avec Bind9 pour les postes clients
 - Savoir configurer des postes clients pour obtenir un service DNS
 - Vérifier les relations serveur DNS / postes clients
- ✓ Savoir faire un schéma de l'infrastructure mise en place

Rappel des prérequis



Prérequis : Définir la topologie de notre réseau



Prérequis : Définir le plan d'adressage (MAJ)

Réseau : 192.168.4.0/24		Réservations			
		Poste Serveurs	OUI NON	ADRESSE MAC	ADRESSE IP
Adresse début	192.168.4.100	user01	OUI	08:00:27:49:BB:C6	192.168.4.150
Adresse de fin	192.168.4.200	user02	NON	08:00:27:CC:CA:8E	IP DHCP
Masque	255.255.255.0				
Durée du bail	3600 secondes				
Options DHCP d'étendue					
Nom	Valeur				
Serveur DHCP	192.168.4.10			08:00:27:16:FC:C4	
Passerelle (PC)	192.168.4.1				
Routeur NAT	Box internet				
Options DHCP de serveur					
Nom	Valeur	Idem			
Serveur DNS	192.168.4.10				
Domaine	formation.local				

Prérequis : Définir les identifiants et mots de passe

Serveur DHCP-DNS :

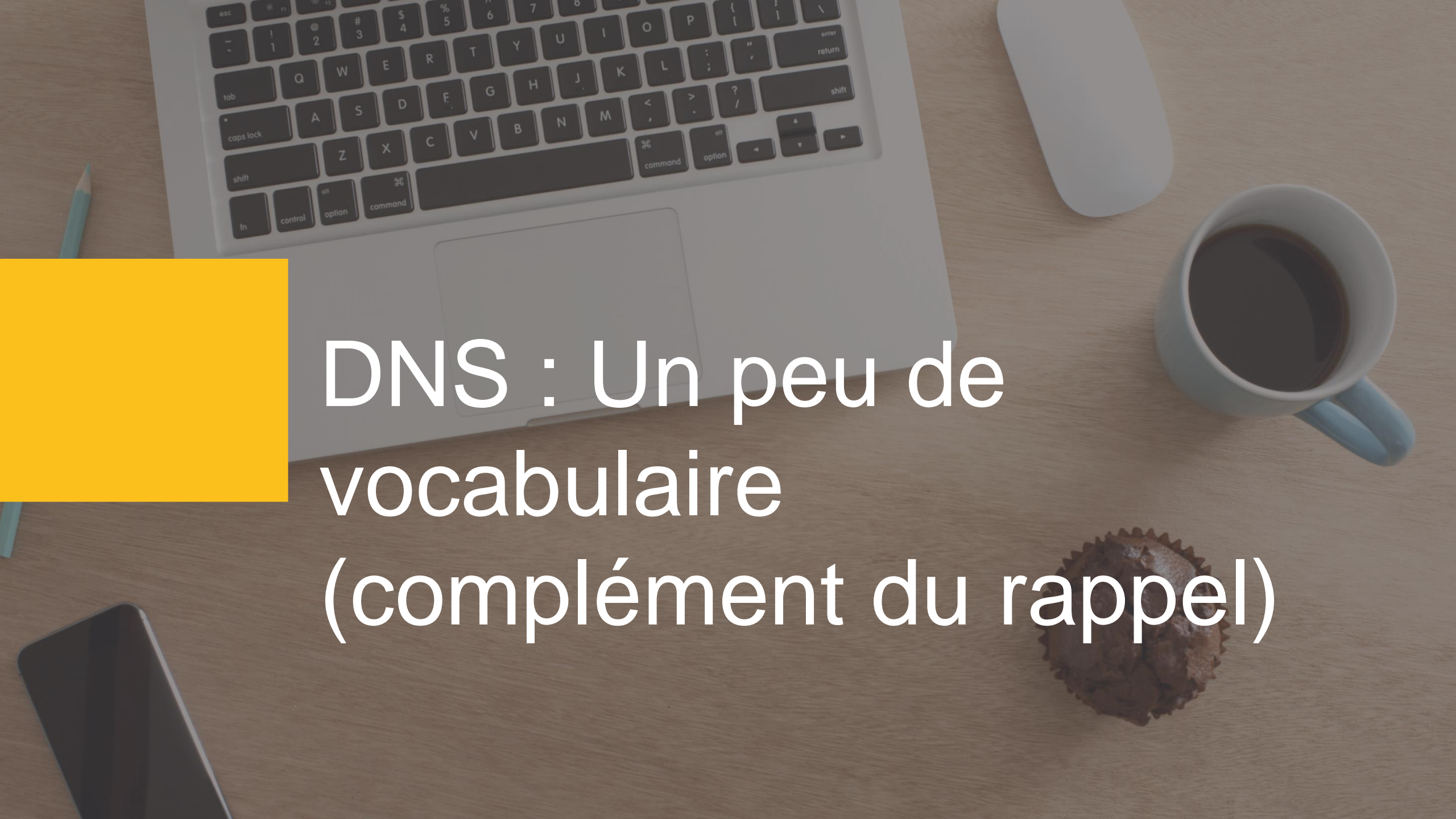
- Hostname : **srv-dhcp-dns**
- Compte root (login / password) : **root / root**
- Compte utilisateur (login / password) : **roi / matthieu** (votre prénom)

Poste client user01 :

- Hostname : **poste-user01**
- Compte root (login / password) : **root / root**
- Compte utilisateur (login / password) : **reine / matthieu** (votre prénom)

Poste client user02 :

- Hostname : **poste-user02**
- Compte root (login / password) : **root / root**
- Compte utilisateur (login / password) : **princesse / matthieu** (votre prénom)

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a small, round chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Un peu de vocabulaire (complément du rappel)

DNS : Un peu de vocabulaire (complément du rappel)

Zone : Ensemble des directives correspondantes à un domaine. À chaque zone correspond un fichier mais une zone n'est pas forcément un domaine.

DNS récursif : DNS capable d'interroger d'autres servers DNS, lorsqu'il ne parvient à trouver un serveur faisant autorité sur le nom de domaine recherché.

Serveur « primaire » ou « maître » d'une zone (en anglais serveur « **authoritative** ») : serveur qui a la configuration de sa zone grâce à un fichier. C'est le serveur principal d'un domaine.

Serveur secondaire : serveur qui des informations sur une zone à partir d'un serveur primaire et non grâce à sa configuration.

DNS : Un peu de vocabulaire (complément du rappel)

Faire autorité sur un domaine : C'est le fait pour un serveur DNS de répondre directement aux requêtes d'un domaine, sans passer par un autre serveur ou un cache.

Un cache : C'est le fichier dans lequel le serveur DNS récursif conserve l'information qu'il a obtenu d'un autre serveur à la suite d'une requête qui lui a été faite par un client.

Donc les serveurs qui font autorités sur un domaine sont, soit des serveurs primaires, soit des serveurs secondaires s'ils ont une copie de ces informations.

Semifir

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. A small portion of a smartphone is visible in the bottom left corner. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Composants de Bind9

DNS : Composants de Bind9

Plusieurs serveur DNS sont disponibles dans la distribution Debian. Nous allons installer le **serveur DNS de référence à savoir BIND** (Berkeley Internet Name Daemon) de l'Internet Software Consortium **dans sa version 9**.

Version 9 : stable et sécurisée

Version 10 depuis 2013 intègre le DHCP

/usr/sbin/named → C'est le programme qui **lance le serveur**

/etc/init.d/bind9 → Permet de **gérer et de redémarrer le service bind**

En root :

- **/etc/init.d/bind9 stop** : pour **arrêter**
- **/etc/init.d/bind9 start** : pour **démarrer**
- **/etc/init.d/bind9 restart** : pour **redémarrer** (si il était démarré, avec restart, il est éteint, puis redémarrer avec un nouveau processus)
- **/etc/init.d/bind9 reload** : pour **recharger la configuration** (ne stoppe pas avant de recharger)

On peut aussi utiliser « **service** » avec chacune des commandes décrites pour « **init.d** » :

- Par exemple : « **service bind9 restart** »

DNS : Composants de Bind9

L'utilitaire rndc

« **/usr/sbin/rndc** » est le fichier binaire de l'**utilitaire de contrôle rndc**.
Il **permet de gérer Bind9**

```
rndc [b source-adress] [-c config-file] [k key-file] [-s serveur] [-p port] [-V] [-y key-id] {commande}
```

Après l'installation de bind9, on peut utiliser les commandes rndc suivantes :

- **reload** : pour **recharger**
- **stop** : **arrêter** le serveur
- **flush** : **vider le cache**
- **status** : **afficher l'état du serveur**
- **aucune** : **liste des commandes** utilisables

DNS : Composants de Bind9

« **/etc/bind/named.conf** »

C'est le **fichier de configuration centrale** de bind.

Il peut se trouver dans différents dossiers (sécurité, chroot)

➤ par exemple dans « **/etc/named.conf** » ou « **/etc/** »

On peut externaliser certains points de configuration de ce fichier central dans des fichiers:

➤ **/etc/bind/named.conf.local**

➤ **/etc/bind/named.conf.options**

« **/var/named/** »

Il s'agit d'un **répertoire de travail**.

Semifir

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Syntaxe des fichiers de configuration

DNS : Syntaxe des fichiers de configuration

Exemples de fichier : named.conf, named.conf.local, named.conf.options, etc...

Il faut **toujours un point virgule pour finir une instruction.**

Exemple d'**instruction** (statements) entre **accolades** :

```
mot-clé {  
    ...  
};
```

Une **instruction simple** est entre **guillemets double**.

Par exemple dans le fichier « **/etc/bind/named.conf** », nous avons :

```
include "/etc/bind/name.conf.options";  
include "/etc/bind/name.conf.local";  
include "/etc/bind/name.conf.default-zones";
```

DNS : Syntaxe des fichiers de configuration

Options de configuration du DNS

Ils sont souvent situés dans le fichier « **named.conf.options** ».

Dans l'**instruction "option"** du fichier named.conf.options, on peut donner les instructions suivantes:

OPTIONS	SIGNIFICATIONS	EXEMPLES
directory	répertoire de travail	directory "/var/named";
forwarders	serveurs de référence (aucun par défaut)	forwarders { adresses.IP.de.serveurs.de.référence; } (sinon il interroge récursivement les autres serveurs DNS)
forward	comportement avec les forwarders (first : en priorité only : uniquement)	forward only ;
version	version du serveur à afficher quand le serveur est interrogé	version none ;


DNS : Syntaxe des fichiers de configuration

L'instruction zones

Permet de définir les **paramètres généraux d'une zone**.

```
zone "nom-de-notre-zone" { #1
    type master;             #2
    file "/etc/bind/db.xxx"; #3
}                             #4
```

- #1 : **Nom de la zone** dans l'entête ;
- #2 : type (**master** pour primaire ou **slave** pour secondaire ou **int** pour Le programme qui lance le server : /usr/sbin/nrachine) ;
- #3 : fichier **chemin du fichier** de configuration de zone
- #4 : éventuellement des **options**

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text. The text is white and centered within the yellow box.

DNS : Prérequis nécessaire avant installation DNS Maître

DNS : Prérequis nécessaire avant installation DNS Maître

Il va s'agir de configurer un serveur DNS **qui servira de serveur cache pour le système sur lequel Bind va être installé**, et qui **sera le serveur DNS maître** pour les systèmes clients du réseau local.

- Soit notre **serveur DHCP-DNS** nommé : « **srv-dhcp-dns** »
- Adresse IP pour l'interface « **enp0s8** » du serveur « **srv-dhcp-dns** » : **192.168.4.10**
- Soit notre **nom de domaine local** : « **formation.local** »
- Soit un poste client sur le réseau local : « **poste-client-user01** » avec l'IP **192.168.4.150**
 - En DHCP **avec réservation par adresse MAC**
- Soit un autre poste client sur le réseau local : « **poste-client-user02** » avec l'IP **172.20.45.50**
 - En DHCP **dans notre deuxième réseau local**
- Soit notre **serveur DHCP relais** nommé : « **srv-dhcp-relais** »
- Adresse IP pour l'interface « **enp0s9** » du serveur « **srv-dhcp-dns** » : **172.20.0.10**

DNS : Prérequis nécessaire avant installation DNS Maître

Même si nous le savons déjà, nous allons taper la commande « **hostname** » sur notre serveur DHCP-DNS « srv-dhcp-dns » afin de **connaître le nom du système** sur lequel on installera Bind :

```
root@srv-dhcp-dns:~# hostname  
srv-dhcp-dns  
root@srv-dhcp-dns:~#
```

Si on veut le changer pour lui donner un **nom plus significatif** de sa fonction de server (ça ne sera pas le cas ici), il faut aller dans le fichier « **/etc/hostname** » :

```
root@srv-dhcp-dns:~# vi /etc/hostname
```

```
srv-dhcp-dns
```

Puis, il faut redémarrer le serveur afin de prendre en compte le changement du nouveau hostname.

DNS : Prérequis nécessaire avant installation DNS Maître

Nous allons maintenant compléter le fichier « **/etc/host.conf** ». Ce fichier **indique quels services de conversion des noms sont disponibles, et dans quel ordre il faut les appliquer.**

C'est la **partie cliente du système** sur lequel va être installé Bind. **Un même système peut être à la fois client et serveur**, c'est-à-dire, serveur DNS "pour lui-même".

```
root@srv-dhcp-dns:~# vi /etc/host.conf
```

Nous allons donc indiquer dans quel ordre appliquer cette recherche et enregistrer le fichier :

- **order hosts** = d'abord dans le fichier /etc/hosts
- **bind** = puis par le DNS en cas d'echec
- **multi on** : Autoriser plusieurs adresses par nom

```
order hosts, bind
multi on
```

```
~
```

DNS : Prérequis nécessaire avant installation DNS Maître

Nous allons maintenant compléter le fichier « **/etc/hosts** ». Ce fichier **permet d'attribuer des noms d'hôtes à chacune des adresses IP.**

Il s'agit là encore de l'aspect client du système. On renseigne tous les clients du réseau local ainsi que le nom de domaine de ce système en tant que client.

```
root@srv-dhcp-dns:~# vi /etc/hosts
```

Par défaut, nous avons ces informations qui sont inscrits :

```
127.0.0.1      localhost
127.0.1.1      srv-dhcp-dns.formation.local  srv-dhcp-dns

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
```

DNS : Prérequis nécessaire avant installation DNS Maître

Nous ajoutons ces **informations issues de nos prérequis** (et on enregistre) :

```
127.0.0.1      localhost
127.0.1.1      srv-dhcp-dns.formation.local      srv-dhcp-dns
192.168.4.10   srv-dhcp-dns.formation.local      srv-dhcp-dns

192.168.4.150  poste-user01.formation.local      poste-user01

172.20.45.50   poste-user02.formation.local      poste-user02
172.20.0.10    srv-dhcp-relais.formation.local    srv-dhcp-relais

# The following lines are desirable for IPv6 capable hosts
::1            localhost ip6-localhost ip6-loopback
ff02::1        ip6-allnodes
ff02::2        ip6-allrouters
```


DNS : Prérequis nécessaire avant installation DNS Maître

Nous pouvons maintenant, **recharger la configuration réseau pour prendre en compte les modifications** avec la commande « **/etc/init.d/networking restart** » :

```
root@srv-dhcp-dns:/etc/bind# /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
root@srv-dhcp-dns:/etc/bind#
```

Semifir

DNS : Prérequis nécessaire avant installation DNS Maître

Maintenant, nous allons **déclarer notre nom de domaine « formation.local »** dans le fichier « **/etc/resolv.conf** » sans oublier, de **retirer les DNS extérieurs**, afin que Bind soit consulté.

```
root@srv-dhcp-dns:~# vi /etc/resolv.conf
```

Par défaut, le fichier ressemble à ça (**par rapport à notre configuration dans le TD DHCP**) :

```
nameserver 192.168.0.254
~
~
```

→ C'est l'IP de notre passerelle Box Internet

Nous modifions donc ce fichier avec les nouvelles données et on enregistre :

```
domain formation.local
search formation.local
nameserver 192.168.4.10
#nameserver 8.8.8.8
#nameserver 8.8.4.4
```

- # Nom de notre domaine local
- # Nom de notre domaine local
- # IP de notre serveur DHCP-DNS
- # DNS 1 de Google
- # DNS 2 de Google

DNS : Prérequis nécessaire avant installation DNS Maître

Faire exactement la même chose sur le serveur DHCP relais
concernant le fichier « `/etc/resolv.conf` » !



Semifir

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

Sur le système voué à servir de serveur DNS, **s'il a été installé un environnement de bureau**, lors du redémarrage du système, la nouvelle configuration du fichier « **/etc/resolv.conf** » **sera effacée par Network Manager**.

Deux solutions pour résoudre ce problème :

- soit on **configure Network Manager**
- soit on se **crée un script de démarrage** pour effacer les modifications de Network Manager

Nous allons choisir la solution via le script !

!! Attention !!

Si vous **décidez par vous-même de supprimer Network Manager**, notez bien que ça **déstabilise complètement le système**. Cette suppression s'effectue avec cette commande (pour information) :

```
root@poste-user01:/home/reine# apt-get remove --purge network-manager-gnome network-manager
```

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

1^{ère} solution : Configurer Network Manager

Sur l'interface graphique, il faut aller dans : **Système → Préférences → Connexions réseau**

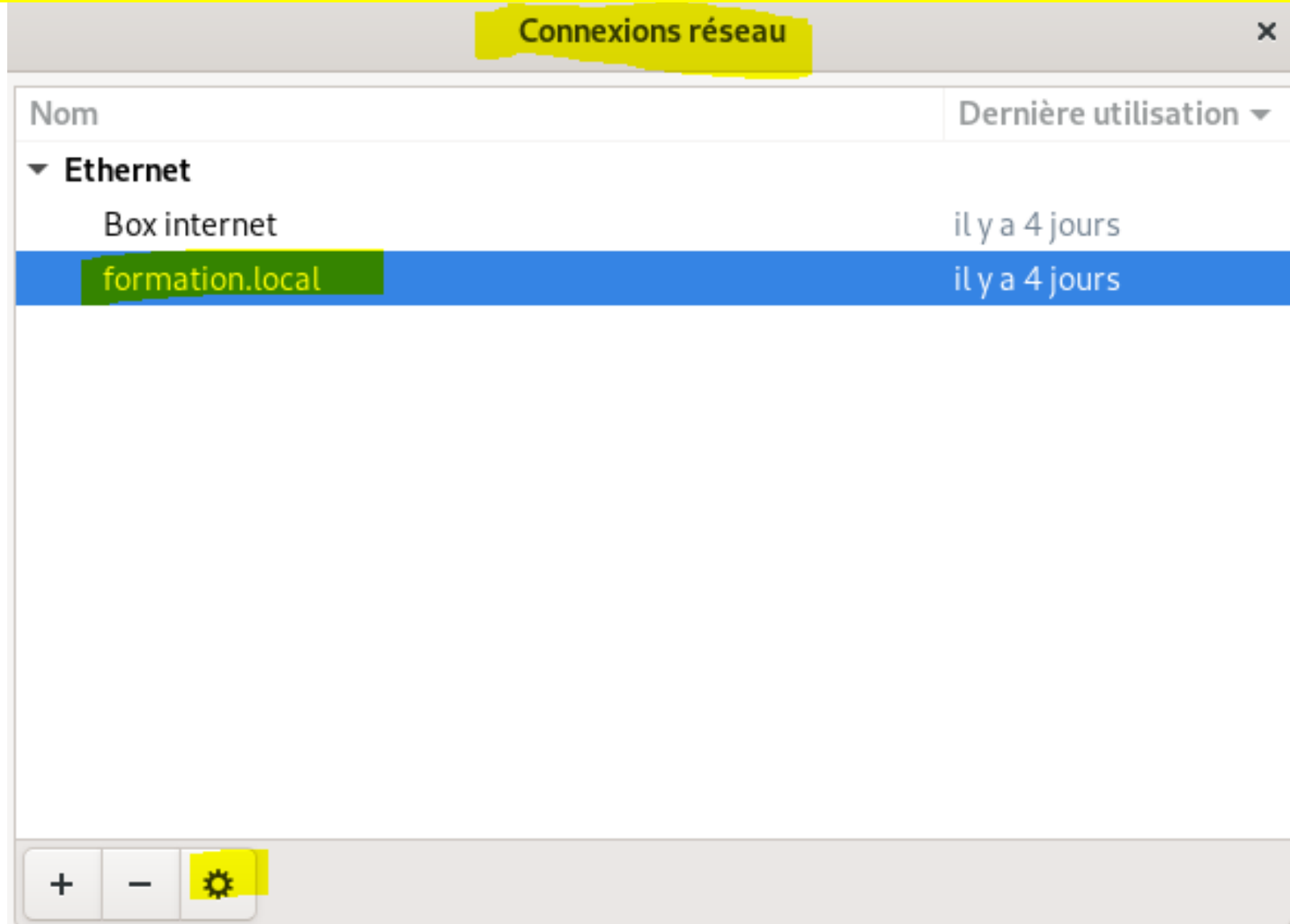
Puis il faut **modifier toutes les connexions** que vous avez dans tous les onglets (Filaire, Sans fil, etc...), en faisant, **pour chacune d'entre-elles** :

- 1) Cliquez sur la connexion à modifier ;
- 2) Bouton « Modifier » ;
- 3) Onglet « Paramètres IPv4 » (et aussi IPv6 si vous l'utilisez) ;
- 4) Méthode : Adresses automatiques uniquement (DHCP) ;
- 5) Serveurs DNS : 127.0.0.1

Puis appliquez les modifications. Si la connexion est partagée entre tous les utilisateurs, un mot de passe administrateur vous sera demandé.

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)



DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

Modification de formation.local x

Nom de la connexion formation.local

Général Ethernet Sécurité 802.1X DCB Proxy **Paramètres IPv4** Paramètres IPv6

Méthode Automatique (DHCP)

Adresse statique supplémentaire

Adresse	Masque de réseau	Passerelle

Ajouter

Supprimer

Serveurs DNS supplémentaires 127.0.0.1

Domaines de recherche supplémentaires

ID de client DHCP

☐ Requiert un adressage IPv4 pour que cette connexion fonctionne

Routes...

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

On peut alors éditer le fichier « **/etc/resolv.conf** » afin qu'il ressemble à ceci :

```
domain formation.local  
search formation.local  
nameserver 127.0.0.1
```

Semifir

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

2ème solution : Script de démarrage pour effacer les modifications de Network Manager

On va modifier le fichier avec un script, en même temps que résoudre le problème « Network Manager », donc **inutile d'éditer** « **/etc/resolv.conf** » **après l'exécution du script**.

Création du script pour Network Manager

On se déplace dans le répertoire « **/etc/NetworkManager/dispatcher.d/** » afin de **créer le script** à l'intérieur de celui-ci et nous faisons un « **ls** » pour voir les fichiers existants :

```
root@poste-user01:~# cd /etc/NetworkManager/dispatcher.d/
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls
01-ifupdown  no-wait.d  pre-down.d  pre-up.d
root@poste-user01:/etc/NetworkManager/dispatcher.d#
```

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

Nous allons donc créer le fichier du script « **99-dns** » avec la commande « **touch** » et nous refaisons un « **ls** » afin de s'assurer que le fichier a bien été créé. Puis on l'édite et on enregistre :

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# touch 99-dns
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls
01-ifupdown 99-dns no-wait.d pre-down.d pre-up.d
root@poste-user01:/etc/NetworkManager/dispatcher.d#
root@poste-user01:/etc/NetworkManager/dispatcher.d# vi 99-dns
```

****99-dns** est un nom aléatoire qui a été donné ici mais respectant bien la nomenclature de ce répertoire.*

```
#!/bin/sh
echo "domain formation.local" > /etc/resolv.conf
echo "search formation.local" >> /etc/resolv.conf
echo "nameserver 192.168.4.10" >> /etc/resolv.conf
echo "#nameserver 8.8.8.8" >> /etc/resolv.conf
echo "#nameserver 8.8.4.4" >> /etc/resolv.conf
```

```
# Nom de notre domaine local
# Nom de notre domaine local
# IP de notre serveur DHCP-DNS
# DNS 1 de Google
# DNS 2 de Google
```


DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

Bind sera ainsi, le server DNS du système sur lequel il est installé. On peut simplement commenter les anciens paramètres du fichier afin d'avoir sous la main les DNS de Google par exemple (en cas où).

Avant l'exécution du script, il faut **mettre les droits utilisateurs** « **rwxr-xr-x** » sur ce fichier.

$$U = \text{rwx} = 4 + 2 + 1 = 7$$

$$G = \text{r-x} = 4 + 0 + 1 = 5$$

$$O = \text{r-x} = 4 + 0 + 1 = 5$$

La commande a utilisé est : « **chmod UGO /etc/NetworkManager/dispatcher.d/99-dns** »

Vu qu'on est déjà dans le répertoire, la commande a utilisé est donc → « **chmod 755 99-dns** »

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# chmod 755 99-dns
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls -ld
drwxr-xr-x 5 root root 4096 janv. 26 09:38 .
root@poste-user01:/etc/NetworkManager/dispatcher.d#
```

DNS : Prérequis nécessaire avant installation DNS Maître

Ne pas faire ce slide, c'est juste pour votre information en cas de serveur en environnement de bureau (interface graphique)

Maintenant, on **exécute le script** avec la commande « **bash** » :

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# bash 99-dns
```

On regarde le fichier « **/etc/resolv.conf** » avec la commande « **less** » ou « **cat** » afin de voir si le script a bien renseigné les informations qu'on lui a fournit :

```
root@poste-user01:~# less /etc/resolv.conf  
root@poste-user01:~#
```

Ce qui retournera ces informations :

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4
```

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4  
/etc/resolv.conf (END)
```

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Bloquer les entrées
dans « /etc/resolv.conf »

DNS : Bloquer les entrées dans « /etc/resolv.conf »

chattr (Change Attribute) est un utilitaire Linux en ligne de commande qui est **utilisé pour définir / annuler certains attributs d'un fichier** dans le système Linux **pour sécuriser la suppression ou la modification accidentelle de fichiers et dossiers importants**, même si vous êtes connecté en tant qu'utilisateur root.

Utilisez la commande ci-dessous pour empêcher « resolv.conf » ou tout autres fichiers de le remplacer après le redémarrage du service réseau et/ou de votre machine.

Pour bloquer la modification et désactiver l'insertion dans le fichier, utilisez la commande ci-dessous:

« **chattr +i /etc/resolv.conf** » : pour bloquer le fichier (par exemple, pour bloquer le DNS local)
Personne/Rien ne pourra le modifier, y compris l'utilisateur racine.

Pour annuler la modification et réactiver l'insertion dans le fichier, utilisez la commande ci-dessous:

« **chattr -i /etc/resolv.conf** » : pour débloquer le fichier (par exemple, pour activer le DNS internet)

DNS : Bloquer les entrées dans « /etc/resolv.conf »

```
root@srv-dhcp-dns:~# vi /etc/resolv.conf
```

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4  
#nameserver 192.168.0.254
```

```
root@srv-dhcp-dns:~# chattr +i /etc/resolv.conf  
root@srv-dhcp-dns:~#
```

DNS : Bloquer les entrées dans « /etc/resolv.conf »

```
root@srv-dhcp-dns:~# cat /etc/resolv.conf
domain formation.local
search formation.local
nameserver 192.168.4.10
#nameserver 8.8.8.8
#nameserver 8.8.4.4
#nameserver 192.168.0.254
root@srv-dhcp-dns:~#
root@srv-dhcp-dns:~# ping google.fr
ping: google.fr: Échec temporaire dans la résolution du nom
root@srv-dhcp-dns:~#
root@srv-dhcp-dns:~#
```

Semifir

DNS : Bloquer les entrées dans « /etc/resolv.conf »

```
domain formation.local
search formation.local
#nameserver 192.168.4.10
nameserver 8.8.8.8
#nameserver 8.8.4.4
#nameserver 192.168.0.254
```

```
root@srv-dhcp-dns:~# vi /etc/resolv.conf
```

```
E45: 'readonly' option is set (add ! to override)
```

DNS : Bloquer les entrées dans « /etc/resolv.conf »


```
root@srv-dhcp-dns:~# chattr -i /etc/resolv.conf  
root@srv-dhcp-dns:~#
```

```
root@srv-dhcp-dns:~# vi /etc/resolv.conf
```

```
domain formation.local  
search formation.local  
#nameserver 192.168.4.10  
nameserver 8.8.8.8  
#nameserver 8.8.4.4  
#nameserver 192.168.0.254  
~  
~
```

DNS : Bloquer les entrées dans « /etc/resolv.conf »

```
root@srv-dhcp-dns:~# cat /etc/resolv.conf
domain formation.local
search formation.local
#nameserver 192.168.4.10
nameserver 8.8.8.8
#nameserver 8.8.4.4
#nameserver 192.168.0.254
root@srv-dhcp-dns:~# ping google.fr
PING google.fr (172.217.22.131) 56(84) bytes of data.
64 bytes from par21s12-in-f3.1e100.net (172.217.22.131): icmp_seq=1 ttl=118 time=11.9 ms
64 bytes from par21s12-in-f3.1e100.net (172.217.22.131): icmp_seq=2 ttl=118 time=11.6 ms
^C
--- google.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 11.577/11.721/11.866/0.180 ms
root@srv-dhcp-dns:~#
```

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Installation du serveur DNS Maître avec Bind9

DNS : Installation du serveur DNS Maître avec Bind9

Dans le cas où vous **partagez votre connexion** internet, il est très **utile d'utiliser un serveur DNS cache**.

Par contre, pour que vos postes clients qui utilisent cette connexion partagée se servent de ce serveur cache DNS, n'oubliez surtout pas de **configurer tous les postes** pour qu'ils **utilisent comme serveur DNS votre serveur** et pas un autre.

Pour cela, donnez comme adresse de serveur DNS, l'adresse interne (côté LAN donc) de votre serveur.

Pour commencer, nous allons mettre à jour la liste des paquets existants dont fait parti « bind9 » avec la commande « **apt-get update** » :

```
root@srv-dhcp-dns:~# apt-get update
Réception de :1 http://security.debian.org/debian-security buster/updates InRelease [65,4 kB]
Atteint :2 http://deb.debian.org/debian buster InRelease
Réception de :3 http://deb.debian.org/debian buster-updates InRelease [51,9 kB]
Réception de :4 http://security.debian.org/debian-security buster/updates/main Sources [170 kB]
Réception de :5 http://security.debian.org/debian-security buster/updates/main amd64 Packages [270 k
B]
557 ko réceptionnés en 1s (1 027 ko/s)
Lecture des listes de paquets... Fait
root@srv-dhcp-dns:~# _
```

DNS : Installation du serveur DNS Maître avec Bind9

Et on installe le paquet « **bind9** » avec la commande « **apt-get install bind9** » :

```
root@srv-dhcp-dns:~# apt-get install bind9
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  bind9utils dns-root-data net-tools python3-ply
Paquets suggérés :
  bind9-doc dnsutils resolvconf ufw python-ply-doc
Les NOUVEAUX paquets suivants seront installés :
  bind9 bind9utils dns-root-data net-tools python3-ply
0 mis à jour, 5 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 1 385 ko dans les archives.
Après cette opération, 5 123 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 0
```


A top-down view of a wooden desk. In the upper left, a portion of a silver laptop is visible, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Configuration du serveur DNS Maître avec Bind9

DNS : Configuration du serveur DNS Maître avec Bind9

Une fois le serveur installé et démarré, nous allons configurer notre premier site.

Tout d'abord, nous allons (**en une seule commande**) nous déplacer dans le répertoire de bind9 « **/etc/bind/** » que nous venons d'installer et **observer** les différents fichiers :

```
root@srv-dhcp-dns:~# cd /etc/bind/ && ls
bind.keys  db.127  db.empty  named.conf          named.conf.local  rndc.key
db.0       db.255  db.local  named.conf.default-zones  named.conf.options  zones.rfc1918
root@srv-dhcp-dns:/etc/bind#
```

La **mise en place** d'un nouveau nom de domaine, aussi appelé **zone**, se fait par la **création d'un fichier (similaire au fichier « db.local »)**. Ce fichier **contient l'ensemble des enregistrements DNS du domaine**. Ce sont ces informations qui seront envoyées lors d'une requête DNS. Ils donnent notamment les adresses IPs de plusieurs services, les IPs des sous-domaines, le temps de vie avant revérification des informations (TTL), etc...

Nous allons donc prendre le **fichier « db.local » comme modèle** (copie) afin de **créer** notre fichier « **db.formation.local** » et l'observer :

DNS : Configuration du serveur DNS Maître avec Bind9

```
root@srv-dhcp-dns:/etc/bind# cp db.local db.formation.local
root@srv-dhcp-dns:/etc/bind# ls
bind.keys  db.255          db.local        named.conf.local  zones.rfc1918
db.0       db.empty        named.conf      named.conf.options
db.127     db.formation.local named.conf.default-zones rndc.key
root@srv-dhcp-dns:/etc/bind#
root@srv-dhcp-dns:/etc/bind# vi db.formation.local _
```

Par défaut, le fichier ressemble à ceci :

```
i
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
~
~
```

DNS : Configuration du serveur DNS Maître avec Bind9

Un DNS est constitué de plusieurs enregistrements, les **RR ou Ressources Records**, définissant les **diverses informations relatives au domaine**. Le **premier** enregistrement est consacré à la **résolution de noms**, dans notre cas, il s'agit du fichier db.example.com. Le **second** sera quant à lui en rapport avec la **résolution de noms inverses** ; il s'agit du fichier « db.formation.local »

Il existe **différents types d'enregistrements représentant chacun un type information différent**. Ci-dessous figure une liste des enregistrements couramment utilisées.

Enregistrement A

C'est l'enregistrement le plus courant. Il fait **correspondre une adresse IPv4 à un nom d'hôte**.

www IN A A.B.C.D

A.B.C.D étant une adresse IPv4

Enregistrement AAAA

Variante de l'enregistrement A, il fait correspondre une adresse **IPv6 à un nom d'hôte**.

www IN AAAA ::A

DNS : Configuration du serveur DNS Maître avec Bind9

Enregistrement SOA

Il permet de **définir les informations relatives à la zone**. En l'occurrence le nom du serveur DNS primaire « srv-dhcp-dns.formation.local. » et l'adresse mail du contact technique (« root.example.com. » ; le « @ » est remplacé par un point). Il est composé de plusieurs champs :

- **Serial** : est un entier non signé 32 bits. C'est le **numéro de série à incrémenter à chaque modification du fichier**. Il permet au serveur secondaire de **recharger les informations qu'ils ont**. L'usage général vient à le formater de cette manière YYYYMMDDXX, soit pour la première modification du 01/04/2007 -> 2007040101, pour la seconde 2007040102.
- **Refresh** : définit la **période de rafraîchissement** des données.
- **Retry** : **si** une **erreur** survient au cours du dernier rafraîchissement, celle-ci sera **répétée au bout du délai Retry**.
- **Expire** : le **serveur** sera **considéré** comme **non disponible** au bout du délai Expire.
- **Negative cache TTL** : définit la **durée de vie d'une réponse NXDOMAIN** de notre part.

DNS : Configuration du serveur DNS Maître avec Bind9

Enregistrement CNAME (Canonical Name)

Il permet de **créer un alias pointant vers un autre enregistrement du domaine courant ou d'un domaine externe.**

Il est possible de **créer un enregistrement CNAME pointant vers un autre enregistrement CNAME** mais cette pratique double le nombre de requêtes, il est donc déconseillé de la pratiquer.

mail	IN	CNAME	www
ftp	IN	CNAME	ftp.domain.tld.
www	IN	A	A.B.C.D

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

Enregistrement MX (Mail Exchange)

Il donne le serveur d'envoi d'emails. Cet enregistrement doit **pointer obligatoirement vers un enregistrement de type A et pas un enregistrement CNAME.**

Il est possible de **définir une priorité sur chaque enregistrement pour donner le serveur email à requêter en priorité.** Si ce serveur est **indisponible**, le serveur ayant la **priorité la plus proche** sera requêté à la place.

	IN	MX	10	mail1
	IN	MX	50	mail2
mail1	IN	A		A.B.C.D
mail2	IN	A		A.B.C.D

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

Enregistrement NS (Name Server)

Il **définit les serveurs DNS du domaine**. Cet enregistrement doit **pointer obligatoirement vers un enregistrement de type A et pas un enregistrement CNAME**.

	IN	NS	domain.tld.
ns	IN	A	A.B.C.D

Enregistrement TXT

Il permet de **définir un enregistrement contenant un texte libre**. Cet enregistrement est notamment utilisé pour **confirmer le détenteur du domaine pour pouvoir utiliser certains services externes** tel que Google Webmaster tools ou encore un service d'envoi de mails (Mandrill, Mailgun, ...).

domain.tld.	IN	TXT	"text"
-------------	----	-----	--------

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

Nous allons donc le modifier et enregistrer les changements :

```
;
; BIND data file for local enp0s8 interface
;
$TTL      604800
@         IN      SOA      srv-dhcp-dns.formation.local. root.formation.local. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       srv-dhcp-dns.formation.local.
@         IN      A        192.168.4.10
srv-dhcp-dns  IN      A      192.168.4.10
```

DNS : Configuration du serveur DNS Maître avec Bind9

Avant de poursuivre, nous allons **tester nos enregistrements** créés pour vérifier s'ils sont **corrects** afin d'éviter des erreurs au redémarrage de bind.

La commande « **dig** » va **vérifier la syntaxe du fichier passé en paramètre**. Nous allons donc l'utiliser pour vérifier sur notre domaine formation.local.

Si la commande « dig » n'est pas disponible, il faut installer le paquet « dnsutils » :

```
root@srv-dhcp-dns:/etc/bind# apt-get install dnsutils
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libirs161
Paquets suggérés :
  rblcheck
Les NOUVEAUX paquets suivants seront installés :
  dnsutils libirs161
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 602 ko dans les archives.
Après cette opération, 1 027 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] _
```

DNS : Configuration du serveur DNS Maître avec Bind9

La commande « **dig srv-dhcp-dns.formation.local** » nous retourne ceci :

```
root@srv-dhcp-dns:~# dig srv-dhcp-dns.formation.local

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> srv-dhcp-dns.formation.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63906
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; COOKIE: 79f43df3ab6cbdc444aae6896012e444856ddc8b69806241 (good)
;; QUESTION SECTION:
;srv-dhcp-dns.formation.local.  IN      A

;; ANSWER SECTION:
srv-dhcp-dns.formation.local. 604800 IN A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.             604800 IN      NS      srv-dhcp-dns.formation.local.

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: jeu. janv. 28 17:20:20 CET 2021
;; MSG SIZE rcvd: 115

root@srv-dhcp-dns:~#
```

Il ne vous reste plus qu'à comparer le retour de la commande avec les enregistrements que vous avez rentrés précédemment pour le domaine. Les enregistrements sont normalement les mêmes.

DNS : Configuration du serveur DNS Maître avec Bind9

Maintenant, nous allons créer la **recherche inverse** en prenant le **fichier « db.127 » comme modèle** (copie) afin de **créer** notre fichier « **db.192** » et l'observer :

```
root@srv-dhcp-dns:/etc/bind# cp db.127 db.192
root@srv-dhcp-dns:/etc/bind# ls
bind.keys  db.192      db.formation.local  named.conf.default-zones  rndc.key
db.0       db.255     db.local            named.conf.local          zones.rfc1918
db.127     db.empty   named.conf           named.conf.options
root@srv-dhcp-dns:/etc/bind#
root@srv-dhcp-dns:/etc/bind# vi db.192
```

Par défaut, le fichier ressemble à ceci :

```
i
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                                1           ; Serial
                                604800      ; Refresh
                                86400       ; Retry
                                2419200     ; Expire
                                604800 )    ; Negative Cache TTL
;
@         IN      NS       localhost.
1.0.0     IN      PTR      localhost.
~
```

DNS : Configuration du serveur DNS Maître avec Bind9

On va donc le modifier et enregistrer les changements :

```
;
; BIND reverse data file for local enp0s8 interface
;
$TTL      604800
@         IN      SOA      formation.local. root.formation.local. (
                                1          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       formation.local.
1         IN      PTR      formation.local
1         IN      PTR      srv-dhcp-dns.formation.local
10        IN      PTR      srv-dhcp-dns.formation.local.
```

DNS : Configuration du serveur DNS Maître avec Bind9

La commande « **dig 192.168.4.10** » nous retourne ceci :

```
root@srv-dhcp-dns:~# dig 192.168.4.10

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> 192.168.4.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 52912
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: deda02617a69b4757472b2556012dc7b0efd64ddb0cf31ec (good)
;; QUESTION SECTION:
;192.168.4.10.                IN      A

;; Query time: 1 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: jeu. janv. 28 16:47:07 CET 2021
;; MSG SIZE rcvd: 69

root@srv-dhcp-dns:~#
```

Il ne vous reste plus qu'à comparer le retour de la commande avec les enregistrements que vous avez rentrés précédemment pour le domaine. Les enregistrements sont normalement les mêmes.

DNS : Configuration du serveur DNS Maître avec Bind9

La configuration du domaine terminée, il est nécessaire maintenant d'**inclure cette configuration dans la liste des domaines de bind9** en modifiant le fichier « **named.conf.local** » :

```
root@srv-dhcp-dns:/etc/bind# ls
bind.keys  db.192      db.formation.local  named.conf.default-zones  rndc.key
db.0       db.255      db.local            named.conf.local          zones.rfc1918
db.127     db.empty   named.conf          named.conf.options
root@srv-dhcp-dns:/etc/bind#
root@srv-dhcp-dns:/etc/bind# vi named.conf.local _
```

Par défaut, le fichier est « vide » et ressemble à 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";
```


DNS : Configuration du serveur DNS Maître avec Bind9

Pour **définir une déclaration de « zone »**, il faut d'abord savoir ce que c'est et ce qu'elle contient.

En effet, une déclaration de « zone » **définit les caractéristiques d'une zone** tels que l'**emplacement de ses fichiers de configuration** et les **options spécifiques à la zone**. Cette déclaration peut-être utilisée pour remplacer les déclarations globales d'options.

Une déclaration de zone se présente sous le format suivant :

```
zone <zone-name> <zone-class> {  
    <zone-options>;  
    [<zone-options>; ...]  
};
```

- <zone-name> = nom de la zone
- <zone-class> = à la classe optionnelle de la zone
- <zone-options> = représente une liste des options caractérisant la zone.

DNS : Configuration du serveur DNS Maître avec Bind9

L'attribut **<zone-name>** de la déclaration de zone est particulièrement important, puisqu'il **représente la valeur par défaut assignée à la directive \$ORIGIN utilisés au sein du fichier de zone correspondant** qui se trouve dans le répertoire « **/var/named/** ». Le démon « **named** » **attache le nom de la zone à tout nom de domaine** qui n'est pas pleinement qualifié, listé dans le fichier de zone.

Par exemple, si une déclaration de zone définit l'espace de nom pour « **example.com** », utilisez « **example.com** » comme **<zone-name>** afin qu'il soit placé à la fin des noms d'hôtes au sein du fichier de zone « **example.com** ».

De nombreuses options de « **zone** » sont disponibles, dont **beaucoup dépendent l'une de l'autre pour fonctionner correctement**. Ci-dessous figure une liste des options couramment utilisées.

- **allow-query** : spécifie les clients qui sont autorisés à requérir des informations à propos de cette zone. Par défaut toutes les requêtes d'informations sont autorisées.
- **allow-transfer** : spécifie les serveurs esclaves qui sont autorisés à requérir un transfert des informations de la zone. Par défaut toutes les requêtes de transfert sont autorisées.

DNS : Configuration du serveur DNS Maître avec Bind9

- **allow-update** : spécifie les hôtes qui sont autorisés à mettre à jour dynamiquement des informations dans leur zone. Par défaut aucune requête de mise à jour dynamique n'est autorisée.

Soyez très prudent lorsque vous autorisez des hôtes à mettre à jour des informations à propos de leur zone. **Ne mettez en œuvre cette option que si vous accordez une confiance absolue à l'hôte.** De manière générale, il est préférable de laisser un administrateur mettre à jour manuellement les enregistrements de la zone et recharger le service « named » service.

- **file** : spécifie le nom du fichier qui contient les données de configuration de la zone, dans le répertoire de travail « named ».
- **masters** : spécifie les adresses IP à partir desquelles demander des informations sur la zone faisant autorité. Cette option ne doit être utilisée que si la zone est définie en tant que « type slave ».

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

- **notify** : établit si named notifie les serveurs esclaves lorsqu'une zone est mise à jour. Cette directive accepte les options suivantes :
 - **yes** : notifie les serveurs esclaves.
 - **no** : ne notifie pas les serveurs esclaves.
 - **explicit** : notifie seulement les serveurs esclaves spécifiés dans une liste also-notify à l'intérieur d'une déclaration de zone.
- **zone-statistics** : configure « named » pour qu'il conserve des statistiques concernant cette zone, en les écrivant soit dans l'emplacement par défaut (« /var/named/named.stats »), soit à l'emplacement expressément désigné par l'option « statistics-file » dans la déclaration « server ».

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

- **type** : définit le type de zone. Les types énumérés ci-dessous peuvent être utilisés.
 - **forward** : retransmet toutes les requêtes d'informations à propos de cette zone vers d'autres serveurs de noms
 - **hint** : un type spécial de zone utilisé pour diriger des transactions vers les serveurs de noms racines qui résolvent des requêtes lorsqu'une zone n'est pas connue autrement. Aucune configuration au-delà de la valeur par défaut n'est nécessaire avec une zone « hint ».
 - **master** : désigne le serveur de noms faisant autorité pour cette zone. Une zone devrait être configurée comme de type « master » (maître) si les fichiers de configuration de la zone se trouvent sur le système.
 - **slave** : désigne le serveur de noms comme serveur esclave pour cette zone. Cette option spécifie également l'adresse IP du serveur de noms maître pour cette zone.

DNS : Configuration du serveur DNS Maître avec Bind9

Un exemple de déclaration de zone pour le serveur de noms primaire hébergeant example.com (192.168.0.1) dont le type est « master » :

```
zone "example.com" IN {  
    type master;  
    file "example.com.zone";  
    allow-update { none; };  
};
```

Un exemple de déclaration de zone pour le serveur de noms de la zone example.com (192.168.0.1) dont le type est « slave » :

```
zone "example.com" {  
    type slave;  
    file "example.com.zone";  
    masters { 192.168.0.1; };  
};
```

DNS : Configuration du serveur DNS Maître avec Bind9

On va donc le modifier et enregistrer les changements :

```
//  
// 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 "formation.local" {  
    type master;  
    file "/etc/bind/db.formation.local";  
    allow-query { any; };  
};  
  
zone "4.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192";  
};
```


DNS : Configuration du serveur DNS Maître avec Bind9

Si nous voulons **inclure des options** dans nos configurations, il faut modifier le fichier « **named.conf.options** ».

```
root@srv-dhcp-dns:/etc/bind# ls
bind.keys  db.192      db.formation.local  named.conf.default-zones  rndc.key
db.0       db.255      db.local            named.conf.local          zones.rfc1918
db.127     db.empty    named.conf          named.conf.options
root@srv-dhcp-dns:/etc/bind#
root@srv-dhcp-dns:/etc/bind# vi named.conf.options
```

En effet, ce fichier **définit les options globales** de configuration de serveur et **établit des valeurs par défaut** pour les autres déclarations.

Cette déclaration peut être **utilisée en autres pour spécifier l'emplacement du répertoire de travail** « **named** », ou pour **déterminer les types de requêtes autorisés**.

Par défaut, le fichier ressemble à ceci :

DNS : Configuration du serveur DNS Maître avec Bind9

```
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 {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};
```

DNS : Configuration du serveur DNS Maître avec Bind9

De nombreuses options sont disponibles, dont **beaucoup dépendent l'une de l'autre pour fonctionner correctement**. Ci-dessous figure une liste des options couramment utilisées.

- **allow-query** : spécifie les hôtes autorisés à interroger ce serveur de noms. Par défaut, tous les hôtes sont autorisés à interroger le serveur de noms. Une liste de contrôle d'accès ou un ensemble d'adresses IP ou de réseaux peuvent être utilisés ici afin de n'autoriser que des hôtes précis à interroger le serveur de noms.
- **allow-recursion** : semblable à « allow-query », cette option s'applique à des demandes récursives. Par défaut, tous les hôtes sont autorisés à effectuer des demandes récursives sur le serveur de noms.
- **blackhole** : spécifie les hôtes qui ne sont pas autorisés à interroger le serveur de noms.
- **directory** : change le répertoire de travail « named » pour une valeur autre que la valeur par défaut, /var/named/.

DNS : Configuration du serveur DNS Maître avec Bind9

- **forward** : contrôle le comportement de retransmission d'une directive « forwarders ».
 - Les options suivantes sont acceptées :
 - **first** : établit que les serveurs de noms spécifiés dans la directive « forwarders » soient interrogés avant que « named » ne tente de résoudre le nom lui-même.
 - **only** : spécifie que « named » ne doit pas tenter d'effectuer lui-même une résolution de nom dans le cas où des demandes vers les serveurs de noms spécifiés dans la directive « forwarders » échoueraient.
- **forwarders** : spécifie une liste d'adresses IP valides correspondant aux serveurs de noms vers lesquels les requêtes devraient être envoyées pour la résolution.

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

- **listen-on** : spécifie l'interface réseau sur laquelle « named » prend note des requêtes. Par défaut, toutes les interfaces sont utilisées.

De cette manière, si le serveur DNS sert également de passerelle, BIND peut être configuré de telle sorte qu'il ne réponde qu'aux requêtes en provenance de l'un des réseaux.

- Une directive « listen-on » peut ressembler à l'extrait ci-dessous :

```
options {  
listen-on { 10.0.1.1; };  
};
```

- Dans cet exemple, seules les requêtes qui proviennent de l'interface de réseau servant le réseau privé (10.0.1.1) seront acceptées.

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

- **notify** : établit si « named » notifie les serveurs esclaves lorsqu'une zone est mise à jour. Les options suivantes sont acceptées :
 - **yes** : notifie les serveurs esclaves.
 - **no** : ne notifie pas les serveurs esclaves.
 - **explicit** : notifie seulement les serveurs esclaves spécifiés dans une liste « also-notify » à l'intérieur d'une déclaration de zone.
- **pid-file** : spécifie l'emplacement du fichier de processus ID créé par « named ».
- **statistics-file** : spécifie un autre emplacement des fichiers de statistiques. Par défaut, les « named » sont enregistrées dans le fichier « /var/named/named.stats ».

Semifir

DNS : Configuration du serveur DNS Maître avec Bind9

On va donc le modifier et enregistrer les changements :

```
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 {
        192.168.4.10;
        8.8.8.8;
        8.8.4.4;
        // autres dns en commentaires possibles (ex: FAI);
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    auth-nxdomain no;          # conforme a la RFC1035
    version none;
    forward only;
    // listen-on-v6 { any; };
};
```


DNS : Configuration du serveur DNS Maître avec Bind9

Avant de redémarrer le serveur, nous allons **tester le fichier de domaine** créé pour vérifier s'il est **correct** afin d'éviter des erreurs au redémarrage de bind.

La commande « **named-checkzone** », incluse dans le package de bind9, va **vérifier la syntaxe du fichier passé en paramètre**. Nous allons donc l'utiliser pour vérifier sur notre domaine formation.local :

```
root@srv-dhcp-dns:/etc/bind# named-checkzone formation.local /etc/bind/db.formation.local
zone formation.local/IN: NS 'srv-dhcp-dns.formation.local' has no address records (A or AAAA)
zone formation.local/IN: not loaded due to errors.
root@srv-dhcp-dns:/etc/bind#
```

Nous avons obtenu une erreur car il ne faut surtout pas oublier de **mettre le FQDN complet** !
A savoir « **srv-dhcp-dns.formation.local** » et non pas uniquement « formation.local » :

```
root@srv-dhcp-dns:/etc/bind# named-checkzone srv-dhcp-dns.formation.local /etc/bind/db.formation.local
zone srv-dhcp-dns.formation.local/IN: loaded serial 2
OK
root@srv-dhcp-dns:/etc/bind#
```

DNS : Configuration du serveur DNS Maître avec Bind9

Nous pouvons maintenant, **redémarrer le service pour prendre en compte les modifications** avec la commande « **/etc/init.d/bind9 restart** » ou avec « **service bind9 restart** » :

```
root@srv-dhcp-dns:/etc/bind# /etc/init.d/bind9 restart  
[ ok ] Restarting bind9 (via systemctl): bind9.service.  
root@srv-dhcp-dns:/etc/bind#
```

Semifir

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text. The background is a light-colored wooden surface.

DNS : Configuration du serveur DNS Maître avec Bind9 pour les postes clients

DNS : Configuration du serveur DNS Maître avec Bind9 pour les postes clients

Maintenant, nous allons **mettre en place la configuration des postes clients** sur notre serveur DNS Maître en éditant les fichiers « **/etc/bind/db.formation.local** » et « **/etc/bind/db.192** ».

```
root@srv-dhcp-dns:/etc/bind# vi db.formation.local _
```

```
.;
; BIND data file for local enp0s8 interface
;
$TTL      604800
@         IN      SOA      srv-dhcp-dns.formation.local. root.formation.local. (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@         IN      NS       srv-dhcp-dns.formation.local.
@         IN      A        192.168.4.10

srv-dhcp-dns    IN      A      192.168.4.10
poste-user01    IN      A      192.168.4.150
poste-user02    IN      A      172.20.45.50
srv-dhcp-relais IN      A      172.20.45.10
~
~
```

DNS : Configuration du serveur DNS Maître avec Bind9 pour les postes clients

Maintenant, nous allons **mettre en place la configuration des postes clients** sur notre serveur DNS Maître en éditant les fichiers « **/etc/bind/db.formation.local** » et « **/etc/bind/db.192** ».

```
root@srv-dhcp-dns:/etc/bind# vi db.192 _
```

```
;
; BIND reverse data file for local enp0s8 interface
;
$TTL      604800
@         IN      SOA      formation.local. root.formation.local. (
                                1           ; Serial
                                604800       ; Refresh
                                86400        ; Retry
                                2419200      ; Expire
                                604800 )     ; Negative Cache TTL
;
@         IN      NS       formation.local.
1         IN      PTR      formation.local
1         IN      PTR      srv-dhcp-dns.formation.local
10        IN      PTR      srv-dhcp-dns.formation.local.
150       IN      PTR      poste-user01.
```

DNS : Configuration du serveur DNS Maître avec Bind9 pour les postes clients

Nous pouvons maintenant, **redémarrer le service pour prendre en compte les modifications** avec la commande « **/etc/init.d/bind9 restart** » ou avec « **service bind9 restart** » :

```
root@srv-dhcp-dns:/etc/bind# /etc/init.d/bind9 restart  
[ ok ] Restarting bind9 (via systemctl): bind9.service.  
root@srv-dhcp-dns:/etc/bind#
```

Semifir

A top-down view of a wooden desk. In the upper left, a silver laptop is open, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white mug filled with dark coffee. In the bottom right corner, there is a chocolate muffin. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Configuration des postes clients

DNS : Configuration des postes clients

Sur chacun des postes clients, il faut **configurer** les fichiers « **/etc/host.conf** » et « **/etc/resolv.conf** ».

Nous allons commencer par modifier le fichier « **/etc/host.conf** » afin que le serveur bind du réseau local soit interrogé par le client :

```
root@poste-user01:~# vi /etc/host.conf
```

```
root@poste-user02:/home/princesse# vi /etc/host.conf
```

En y ajoutant ces paramètres :

```
order hosts,bind
multi on
#nospoof on
```

DNS : Configuration des postes clients

- **order** : indique l'ordre des requêtes : ici, d'abord le fichier hosts, puis, en cas d'échec, le serveur de noms qui sera le serveur Bind quand le fichier « /etc/resolv.conf » aura été modifier pour se faire.
- **multi mis à on** : plusieurs adresses IP peuvent être associées à un même nom.
- **nospoof** : oblige, par sécurité, à vérifier la concordance entre adresse IP et nom lors de la résolution d'adresses inverse.

Le client va **lire** le fichier « **hosts.conf** » et **rechercher l'adresse correspondant** au nom demandé d'abord dans le fichier hosts local.

Si la requête **échoue**, il va **s'adresser à Bind**, le serveur DNS du réseau local, **qui va lui-même demander à des forwarders s'il ne sait pas répondre**. Pour qu'il trouve l'adresse de ce serveur DNS, il **consulte** le fichier « **/etc/resolv.conf** » qu'il est donc **nécessaire** de modifier.

Semifir

DNS : Configuration des postes clients

Nous continuons en modifiant le fichier « **/etc/resolv.conf** » afin que le serveur bind du réseau local soit interrogé par le client.

Il y a deux solutions :

Solution N°1 : Installer un script client pour /etc/resolv.conf

Ce qui permet là aussi de ne plus être embêté par Network Manager, mais cette fois il va permettre de renseigner le système client DNS par l'adresse IP du serveur local bind.

Nous allons choisir cette solution via le script !

Solution 2 : Configurer Network Manager

Semifir

DNS : Configuration des postes clients

1ère solution : Installer un script client pour `/etc/resolv.conf` (**sur tous les postes clients**)

On va modifier le fichier avec un script, en même temps que résoudre le problème « Network Manager », donc **inutile d'éditer** « `/etc/resolv.conf` » **après l'exécution du script**.

Création du script pour Network Manager

On se déplace dans le répertoire « `/etc/NetworkManager/dispatcher.d/` » afin de **créer le script** à l'intérieur de celui-ci et nous faisons un « **ls** » pour voir les fichiers existants :

```
root@poste-user01:~# cd /etc/NetworkManager/dispatcher.d/
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls
01-ifupdown  no-wait.d  pre-down.d  pre-up.d
root@poste-user01:/etc/NetworkManager/dispatcher.d#
```

DNS : Configuration des postes clients

Nous allons donc créer le fichier du script « **99-dns** » avec la commande « **touch** » et nous refaisons un « **ls** » afin de s'assurer que le fichier a bien été créé. Puis on l'édite et on enregistre :

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# touch 99-dns
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls
01-ifupdown 99-dns no-wait.d pre-down.d pre-up.d
root@poste-user01:/etc/NetworkManager/dispatcher.d#
root@poste-user01:/etc/NetworkManager/dispatcher.d# vi 99-dns
```

****99-dns** est un nom aléatoire qui a été donné ici mais respectant bien la nomenclature de ce répertoire.*

```
#!/bin/sh
echo "domain formation.local" > /etc/resolv.conf
echo "search formation.local" >> /etc/resolv.conf
echo "nameserver 192.168.4.10" >> /etc/resolv.conf
echo "#nameserver 8.8.8.8" >> /etc/resolv.conf
echo "#nameserver 8.8.4.4" >> /etc/resolv.conf
```

```
# Nom de notre domaine local
# Nom de notre domaine local
# IP de notre serveur DHCP-DNS
# DNS 1 de Google
# DNS 2 de Google
```

DNS : Configuration des postes clients

Bind sera ainsi, le server DNS du système sur lequel il est installé. On peut simplement commenter les anciens paramètres du fichier afin d'avoir sous la main les DNS de Google par exemple (en cas où).

Avant l'exécution du script, il faut **mettre les droits utilisateurs** « **rwxr-xr-x** » sur ce fichier.

U = **rw****x** = 4 + 2 + 1 = **7**

G = **r**-**x** = 4 + 0 + 1 = **5**

O = **r**-**x** = 4 + 0 + 1 = **5**

La commande a utilisé est : « **chmod UGO /etc/NetworkManager/dispatcher.d/99-dns** »

Vu qu'on est déjà dans le répertoire, la commande a utilisé est donc → « **chmod 755 99-dns** »

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# chmod 755 99-dns
root@poste-user01:/etc/NetworkManager/dispatcher.d# ls -ld
drwxr-xr-x 5 root root 4096 janv. 26 09:38 .
root@poste-user01:/etc/NetworkManager/dispatcher.d#
```

DNS : Configuration des postes clients

Maintenant, on **exécute le script** avec la commande « **bash** » :

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# bash 99-dns
```

On regarde le fichier « **/etc/resolv.conf** » avec la commande « **less** » ou « **cat** » afin de voir si le script a bien renseigné les informations qu'on lui a fournit :

```
root@poste-user01:~# less /etc/resolv.conf  
root@poste-user01:~#
```

Ce qui retournera ces informations :

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4
```

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4  
/etc/resolv.conf (END)
```

DNS : Configuration des postes clients

Si vous choisissez la 1^{ère} solution, vous devez refaire ces différentes étapes sur tous les postes clients du réseau local.

En l'occurrence, ici, on refait ces étapes sur le poste-client-user02 !

Semifir

DNS : Configuration des postes clients

2ème solution : Configurer Network Manager

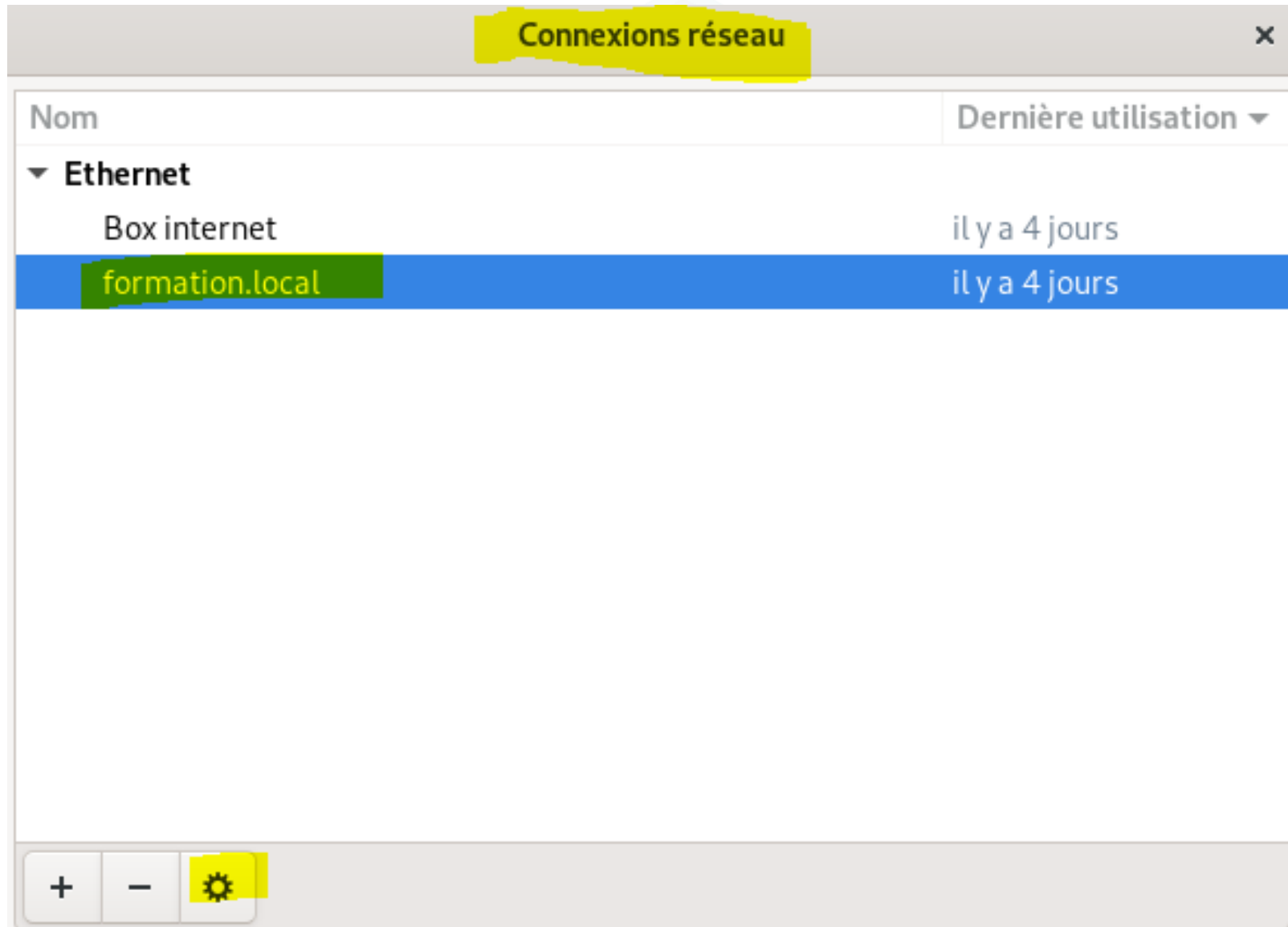
Sur l'interface graphique, il faut aller dans : **Système → Préférences → Connexions réseau**

Puis il faut **modifier toutes les connexions** que vous avez dans tous les onglets (Filaire, Sans fil, etc...), en faisant, **pour chacune d'entre-elles** :

- 1) Cliquez sur la connexion à modifier ;
- 2) Bouton « Modifier » ;
- 3) Onglet « Paramètres IPv4 » (et aussi IPv6 si vous l'utilisez) ;
- 4) Méthode : Adresses automatiques uniquement (DHCP) ;
- 5) Serveurs DNS : IP du serveur DNS local à savoir 192.168.4.10

Puis appliquez les modifications. Si la connexion est partagée entre tous les utilisateurs, un mot de passe administrateur vous sera demandé.

DNS : Configuration des postes clients



DNS : Configuration des postes clients

Modification de formation.local x

Nom de la connexion formation.local

Général

Ethernet

Sécurité 802.1X

DCB

Proxy

Paramètres IPv4

Paramètres IPv6

Méthode Automatique (DHCP) ▼

Adresse statique supplémentaire

Adresse	Masque de réseau	Passerelle

Ajouter

Supprimer

Serveurs DNS supplémentaires 192.168.4.10

Domaines de recherche supplémentaires

ID de client DHCP

☐ Requiert un adressage IPv4 pour que cette connexion fonctionne

Routes...

DNS : Configuration des postes clients

On peut alors éditer le fichier « **/etc/resolv.conf** » afin qu'il ressemble à ceci :

```
domain formation.local  
search formation.local  
nameserver 192.168.4.10  
#nameserver 8.8.8.8  
#nameserver 8.8.4.4  
/etc/resolv.conf (END)
```

Semifir

DNS : Configuration des postes clients

Si vous choisissez la 2^{ème} solution, vous devez refaire ces différentes étapes sur tous les postes clients du réseau local.

En l'occurrence, ici, on refait ces étapes sur le poste-client-user02 !

Semifir

A top-down view of a wooden desk. In the upper left, a portion of a silver laptop is visible, showing its keyboard and trackpad. To the right of the laptop is a white computer mouse. Further right is a white ceramic cup filled with dark coffee. In the bottom right corner, there is a single chocolate muffin. In the bottom left corner, a portion of a black smartphone is visible. A yellow rectangular box is positioned on the left side of the image, partially overlapping the laptop and the text.

DNS : Vérification des relations DNS/clients

DNS : Vérification des relations DNS/clients

On va commencer par vérifier que le serveur DNS se connaisse lui-même.

Nous tapons la commande « **hostname** » pour avoir le **nom complet sur le système** avec Bind :

```
root@srv-dhcp-dns:/etc/bind# hostname
srv-dhcp-dns
root@srv-dhcp-dns:/etc/bind#
```

Avec la commande « **nslookup** », on va demander l'**adresse associée à cet hôte « srv-dhcp-dns »** :

```
root@srv-dhcp-dns:/etc/bind# nslookup
> srv-dhcp-dns
Server:          192.168.4.10
Address:         192.168.4.10#53

Name:   srv-dhcp-dns.formation.local
Address: 192.168.4.10
> _
```

DNS : Vérification des relations DNS/clients

Idem pour la zone inverse, on va faire l'inverse de ce que l'on vient de demander.

Avec la commande « **nslookup** », on va demander le nom associé à cette **adresse** « **192.168.4.10** » :

```
root@srv-dhcp-dns:/etc/bind# nslookup
> 192.168.4.10
10.4.168.192.in-addr.arpa      name = srv-dhcp-dns.formation.local.
>
```

Nous avons une réponse dans les deux cas, donc tout est fonctionnel.

Semifir

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig srv-dhcp-dns** » = **interrogation du hostname**

```
root@srv-dhcp-dns:/etc/bind# dig srv-dhcp-dns

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> srv-dhcp-dns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 27133
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 13286cc8b6594886c382a80b60116c7b46de005537fdcc84 (good)
;; QUESTION SECTION:
;srv-dhcp-dns.                IN      A

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 14:36:59 CET 2021
;; MSG SIZE rcvd: 69

root@srv-dhcp-dns:/etc/bind#
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig formation.local** »
= **interrogation simple du nom de domaine**

```
root@srv-dhcp-dns:/etc/bind# dig formation.local

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> formation.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 26614
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 283d13cae82253163d6fecff60116cb07433cdbf637fa364 (good)
;; QUESTION SECTION:
;formation.local.                IN      A

;; ANSWER SECTION:
formation.local.                604800  IN      A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.                604800  IN      NS      srv-dhcp-dns.formation.local.

;; ADDITIONAL SECTION:
srv-dhcp-dns.formation.local. 604800  IN      A      192.168.4.10

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 14:37:52 CET 2021
;; MSG SIZE rcvd: 131

root@srv-dhcp-dns:/etc/bind#
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig -x 192.168.4.10** » = **Recherche DNS inversée (PTR doit être établi en amont)**

```
root@srv-dhcp-dns:/etc/bind# dig -x @192.168.4.10

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> -x @192.168.4.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 55720
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: d53162e5a4baede0feb288dc60116cefdf410a4b62648e1c (good)
;; QUESTION SECTION:
;10.4.168.\@192.in-addr.arpa.    IN      PTR

;; Query time: 1 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 14:38:55 CET 2021
;; MSG SIZE  rcvd: 83

root@srv-dhcp-dns:/etc/bind#
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

Pour aller plus loin :

<https://www.hostinger.fr/tutoriels/comment-utiliser-la-commande-dig-sous-linux/>

Semifir

DNS : Vérification des relations DNS/clients

Ensuite, on va vérifier que le serveur DNS connaisse le serveur relais DHCP.

Nous tapons la commande « **nslookup** », on va **demandeur l'adresse associé au hostname** de notre serveur DHCP relais à savoir « **srv-dhcp-relais** ».

```
root@srv-dhcp-dns:/etc/bind# nslookup
> srv-dhcp-relais
Server:          192.168.4.10
Address:         192.168.4.10#53

Name:   srv-dhcp-relais.formation.local
Address: 172.20.45.10
>
```

Semifir

DNS : Vérification des relations DNS/clients

Puis, on va vérifier que le serveur DNS connaisse nos postes clients.

Nous tapons la commande « **nslookup** », on va **demandeur l'adresse associé au hostname** de nos deux clients à savoir « **poste-user01** » et « **poste-user02** ».

```
root@srv-dhcp-dns:/etc/bind# nslookup
> poste-user01
Server:          192.168.4.10
Address:         192.168.4.10#53

Name:   poste-user01.formation.local
Address: 192.168.4.150
>
```

```
root@srv-dhcp-dns:/etc/bind# nslookup
> poste-user02
Server:          192.168.4.10
Address:         192.168.4.10#53

Name:   poste-user02.formation.local
Address: 172.20.45.50
>
```

Semifir

DNS : Vérification des relations DNS/clients

Maintenant, on va vérifier que notre serveur relais DHCP ainsi que nos postes clients interrogent bien le DNS local.

Nous tapons la commande « **host -a srv-dhcp-dns** » sur notre serveur DHCP relais.

```
root@srv-dhcp-relais:~# host -a srv-dhcp-dns
Trying "srv-dhcp-dns.formation.local"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36018
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;srv-dhcp-dns.formation.local.  IN      ANY

;; ANSWER SECTION:
srv-dhcp-dns.formation.local. 604800 IN A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.             604800  IN      NS      srv-dhcp-dns.formation.local.

Received 76 bytes from 192.168.4.10#53 in 2 ms
root@srv-dhcp-relais:~#
```

DNS : Vérification des relations DNS/clients

Nous tapons la commande « **host -a srv-dhcp-dns** » sur notre poste client user01.

```
root@poste-user01:/etc/NetworkManager/dispatcher.d# host -a srv-dhcp-dns
Trying "srv-dhcp-dns.formation.local"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7323
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;srv-dhcp-dns.formation.local. IN ANY

;; ANSWER SECTION:
srv-dhcp-dns.formation.local. 604800 IN A 192.168.4.10

;; AUTHORITY SECTION:
formation.local. 604800 IN NS srv-dhcp-dns.formation.local.

Received 76 bytes from 192.168.4.10#53 in 1 ms
root@poste-user01:/etc/NetworkManager/dispatcher.d# _
```


DNS : Vérification des relations DNS/clients

Nous tapons la commande « **host -a srv-dhcp-dns** » sur notre poste client user02.

```
root@poste-user02:~# host -a srv-dhcp-dns
Trying "srv-dhcp-dns.formation.local"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6819
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;srv-dhcp-dns.formation.local.  IN      ANY

;; ANSWER SECTION:
srv-dhcp-dns.formation.local. 604800 IN A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.             604800 IN      NS      srv-dhcp-dns.formation.local.

Received 76 bytes from 192.168.4.10#53 in 1 ms
root@poste-user02:~#
```

semitir

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre serveur relais DHCP qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig formation.local** »
= **interrogation simple du nom de domaine**

Si la commande « **dig** » n'est pas disponible,
il faut installer le paquet « **dnsutils** ».

```
root@srv-dhcp-relais:~# dig formation.local

; <>> DiG 9.11.5-P4-5.1+deb10u2-Debian <>> formation.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9686
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 45ea1fafe4be907512bca1dc60117c119ea700c6f84bb79a (good)
;; QUESTION SECTION:
;formation.local.                IN      A

;; ANSWER SECTION:
formation.local.                604800  IN      A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.                604800  IN      NS      srv-dhcp-dns.formation.local.

;; ADDITIONAL SECTION:
srv-dhcp-dns.formation.local.  604800  IN      A      192.168.4.10

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 15:43:29 CET 2021
;; MSG SIZE rcvd: 131

root@srv-dhcp-relais:~#
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre serveur relais DHCP qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig -x 192.168.4.10** »
= **Recherche DNS inversée**
(PTR doit être établi en amont)

Si la commande « **dig** » n'est pas disponible, il faut installer le paquet « **dnsutils** ».

```
root@srv-dhcp-relais:~# dig -x 192.168.4.10

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> -x 192.168.4.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57740
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 969ae1cc686d05e81e52b03160117c48111a773bf5dc1522 (good)
;; QUESTION SECTION:
;10.4.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
10.4.168.192.in-addr.arpa. 604800 IN      PTR      srv-dhcp-dns.formation.local.

;; AUTHORITY SECTION:
4.168.192.in-addr.arpa. 604800 IN      NS      formation.local.

;; ADDITIONAL SECTION:
formation.local.          604800 IN      A      192.168.4.10

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 15:44:24 CET 2021
;; MSG SIZE rcvd: 154

root@srv-dhcp-relais:~#
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre poste client **user01** qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig formation.local** »
= **interrogation simple du nom de domaine**

Si la commande « **dig** » n'est pas disponible,
il faut installer le paquet « **dnsutils** ».

```
root@poste-user01:/home/reine# dig formation.local

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> formation.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47928
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 4900fdb1f946851af720d6016011add445e4fbfd3b517887 (good)
;; QUESTION SECTION:
;formation.local.                IN      A

;; ANSWER SECTION:
formation.local.                604800  IN      A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.                604800  IN      NS      srv-dhcp-dns.formation.local.

;; ADDITIONAL SECTION:
srv-dhcp-dns.formation.local.  604800  IN      A      192.168.4.10

;; Query time: 1 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 19:15:48 CET 2021
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre poste client **user01** qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig -x 192.168.4.10** »
= **Recherche DNS inversée**
(PTR doit être établi en amont)

Si la commande « **dig** » n'est pas disponible, il faut installer le paquet « **dnsutils** ».

```
root@poste-user01:/home/reine# dig -x 192.168.4.10

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> -x 192.168.4.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14423
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: fcf07eaf92655431c444fc346011ae1017cc2124516a6eca (good)
;; QUESTION SECTION:
;10.4.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
10.4.168.192.in-addr.arpa. 604800 IN      PTR      srv-dhcp-dns.formation.local.

;; AUTHORITY SECTION:
4.168.192.in-addr.arpa. 604800 IN      NS       formation.local.

;; ADDITIONAL SECTION:
formation.local.          604800 IN      A        192.168.4.10

;; Query time: 0 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 19:16:48 CET 2021
;; MSG SIZE rcvd: 154
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre poste client **user02** qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig formation.local** »
= **interrogation simple du nom de domaine**

Si la commande « **dig** » n'est pas disponible,
il faut installer le paquet « **dnsutils** ».

```
root@poste-user02:/home/princesse# dig formation.local

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> formation.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63117
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: fdbd2efe7c9c35c618af1efa60117ee868792bd6497f7cda (good)
;; QUESTION SECTION:
;formation.local.                IN      A

;; ANSWER SECTION:
formation.local.                604800  IN      A      192.168.4.10

;; AUTHORITY SECTION:
formation.local.                604800  IN      NS      srv-dhcp-dns.formation.local.

;; ADDITIONAL SECTION:
srv-dhcp-dns.formation.local.  604800 IN      A      192.168.4.10

;; Query time: 2 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 15:55:36 CET 2021
```

DNS : Vérification des relations DNS/clients

On va faire quelques commandes « **dig** » depuis notre poste client **user02** qui vont **permettre d'interroger le serveur DNS et de diagnostiquer les dysfonctionnements dans la résolution de nom.**

La commande « **dig -x 192.168.4.10** »
= **Recherche DNS inversée**
(PTR doit être établi en amont)

Si la commande « **dig** » n'est pas disponible, il faut installer le paquet « **dnsutils** ».

```
root@poste-user02:/home/princesse# dig -x 192.168.4.10

; <<>> DiG 9.11.5-P4-5.1+deb10u2-Debian <<>> -x 192.168.4.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1382
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 440d11bf47f7b64b2ab115360117f396659e1304582a115 (good)
;; QUESTION SECTION:
;10.4.168.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
10.4.168.192.in-addr.arpa. 604800 IN      PTR      srv-dhcp-dns.formation.local.

;; AUTHORITY SECTION:
4.168.192.in-addr.arpa. 604800 IN      NS       formation.local.

;; ADDITIONAL SECTION:
formation.local.          604800 IN      A        192.168.4.10

;; Query time: 1 msec
;; SERVER: 192.168.4.10#53(192.168.4.10)
;; WHEN: mer. janv. 27 15:56:57 CET 2021
;; MSG SIZE rcvd: 154
```


Schéma de l'infrastructure mise en place



Schéma de l'infrastructure mise en place

