

TP 1 DNS – R303

Configuration de serveurs DNS primaire et secondaire

Étudiant : GRONDIN Benjamin
Formation : BUT 2 Réseaux & Télécommunications

Contents

1	Introduction	3
1.1	Objectif de ce TP	3
2	Table d’adressage	3
3	Mise en place et configuration du serveur web	4
3.1	Configuration de l’interface enp0s3	4
3.2	Installation du serveur Apache	4
3.3	Modification et vérification de la page web	5
4	Mise en place et configuration du serveur DNS primaire	5
4.1	Configuration de l’interface enp0s3	5
4.2	Enregistrement des configurations	6
4.3	Installation du serveur Bind9	6
4.4	Vérification de l’installation	6
4.5	Configuration du fichier named.conf.local	6
4.6	Création du dossier zones	7
4.7	Création du fichier de zone directe	7
4.8	Création du fichier de zone inverse	8
4.9	Test de configuration des fichiers de zone	8
4.10	Tests de résolution avec nslookup	9
4.10.1	Test de résolution directe	9
4.10.2	Test de résolution inverse	9
4.11	Test d’accès web par nom de domaine	10
5	Mise en place et configuration du serveur DNS secondaire	10
5.1	Configuration de l’interface enp0s3	10
5.2	Installation du serveur bind9	10
5.3	Vérification de l’installation	11
5.4	Configuration du fichier named.conf.local	11
5.5	Configuration pour le transfert de zone	11
5.6	Vérification du transfert de zone	11
5.7	Tests de résolution avec le serveur secondaire	12
5.7.1	Résolution directe	12

5.7.2	Résolution inverse	12
5.8	Test d'accès web via le DNS secondaire	13
6	Conclusion	13

1 Introduction

Pour naviguer sur internet et accéder à des services en ligne tels que les sites de e-commerce, du gouvernement ou de streaming, il faut connaître l'adresse IP des machines qui hébergent ces services. Avec le développement d'internet et la multiplication des services en ligne, il a fallu trouver une solution efficace pour que les utilisateurs puissent trouver le site qu'ils cherchent avec un nom compréhensible pour l'utilisateur lambda au lieu d'une adresse IP difficile à retenir.

Définition

Le **DNS (Domain Name System)** a pour rôle d'associer à une adresse IP un nom de domaine pleinement qualifié (FQDN). Par exemple, au lieu de taper l'adresse 142.250.187.227 dans votre navigateur, vous taperez plutôt **www.google.fr**.

Il existe dans le monde une multitude de serveurs DNS qui communiquent entre eux pour aider les utilisateurs à trouver un site donné. Pour cela, a été mise en place une solution très sollicitée aujourd'hui dans nos réseaux informatiques.

1.1 Objectif de ce TP

Objectifs du TP

Dans ce TP, nous allons mettre en place une architecture réseau comprenant 3 machines distinctes :

- Un serveur DNS primaire qui fera autorité sur la zone **benjamin.rt**
- Un serveur web qui héberge une page HTML accessible par nom de domaine
- Un serveur DNS secondaire redondant en cas de panne du serveur primaire

Nous utiliserons des serveurs Ubuntu 18.04 pour l'ensemble de l'infrastructure.

2 Table d'adressage

Machines	Adresse IP	Masque
Serveur Web	192.168.1.30	255.255.255.0
Serveur DNS primaire	192.168.1.20	255.255.255.0
Serveur DNS secondaire	192.168.1.40	255.255.255.0

Table 1: Plan d'adressage de l'infrastructure DNS

Remarque

Les adresses IP utilisées lors de ce TP ont été exclues de la plage d'adresses du DHCP afin qu'elles puissent être utilisées comme adresses statiques par les différentes machines.

3 Mise en place et configuration du serveur web

3.1 Configuration de l'interface enp0s3

Nous commençons par modifier le fichier *50-cloud-init.yaml* qui contient les configurations IP et qui se situe dans le dossier */etc/netplan*.

Listing 1: Configuration netplan du serveur web

```
1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     enp0s3:
6       dhcp4: no
7       addresses: [192.168.1.30/24]
8       gateway4: 192.168.1.1
9       nameservers:
10        addresses: [8.8.8.8]
```

Nous testons la configuration réseau et appuyons sur Entrée s'il n'y a pas de problème :

Listing 2: Test de la configuration netplan

```
benjamin@server_web:~$ sudo netplan try
[sudo] password for benjamin:
Warning: Stopping systemd-networkd.service, but it can still be
        activated by: systemd-networkd.socket
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 100 seconds
```

3.2 Installation du serveur Apache

Après avoir configuré l'adresse IP, nous pouvons installer le serveur Apache :

Listing 3: Installation d'Apache

```
benjamin@server_web:/etc/netplan$ sudo apt install apache2
```

Vérification du statut du service :

Listing 4: Vérification du service Apache

```
benjamin@server_web:~$ sudo systemctl status apache2
* apache2.service - The Apache HTTP Server
```

```

Loaded: loaded (/lib/systemd/system/apache2.service; enabled;
        vendor preset: enabled)
Drop-In: /lib/systemd/system/apache2.service.d
         apache2-systemd.conf
Active: active (running) since Sat 2024-10-05 14:10:10 UTC;
        53min ago
Process: 920 ExecStart=/usr/sbin/apachectl start (code=exited,
        status=0/SUCCESS)

Main PID: 1155 (apache2)
Tasks: 55 (limit: 2311)
CGroup: /system.slice/apache2.service
        |-1155 /usr/sbin/apache2 -k start
        |-1161 /usr/sbin/apache2 -k start
        |-1162 /usr/sbin/apache2 -k start

```

Avec la commande `systemctl status apache2`, on peut voir que le service Apache est opérationnel.

3.3 Modification et vérification de la page web

Dans le fichier `index.html` situé dans le dossier `/var/www/html/`, nous pouvons configurer le fichier HTML et le personnaliser afin d'identifier notre serveur :

Listing 5: Modification du fichier index.html

```

benjamin@server_web:~$ cd /var/www/html/
benjamin@server_web:/var/www/html$ ls -l
total 12
-rw-r--r-- 1 root root 10930 Sep 15 08:03 index.html

```

Remarque

On peut voir que l'on a accès à la page HTML hébergée sur le serveur web qui a pour adresse IP 192.168.1.30. Dans la suite de ce TP, on pourra accéder à ce même site via un nom de domaine.

4 Mise en place et configuration du serveur DNS primaire

4.1 Configuration de l'interface enp0s3

Pour commencer, il faut configurer les interfaces réseau de nos machines virtuelles. Nous attribuons des adresses statiques à nos interfaces dans le fichier `50-cloud-init.yaml` situé dans le dossier `/etc/netplan` :

Listing 6: Configuration netplan du serveur DNS primaire

```

1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     enp0s3:
6       dhcp4: no

```

```

7      addresses: [192.168.1.20/24]
8      gateway4: 192.168.1.1
9      nameservers:
10     addresses: [8.8.8.8]

```

4.2 Enregistrement des configurations

Avec la commande `netplan try`, on teste le fichier de configuration pour vérifier les possibles erreurs. S'il n'y a pas d'erreur, on applique la configuration en appuyant sur Entrée.

4.3 Installation du serveur Bind9

Après avoir configuré les interfaces réseau, nous installons le package bind9 :

Listing 7: Installation de Bind9

```
benjamin@dns_server:/etc/netplan$ sudo apt install bind9
```

4.4 Vérification de l'installation

Vérification que le service bind9 est bien installé et activé :

Listing 8: Vérification du service bind9

```

benjamin@dns_server:/etc/netplan$ systemctl status bind9
* bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled;
           vendor preset: enabled)
   Active: active (running) since Sun 2024-09-15 12:34:34 UTC;
           10min ago
   Docs: man:named(8)
  Main PID: 845 (named)
    Tasks: 7 (limit: 4653)
   CGroup: /system.slice/bind9.service
           L-845 /usr/sbin/named -f -u bind

```

4.5 Configuration du fichier named.conf.local

Nous configurons le fichier *named.conf.local* pour déclarer les zones directe et inverse :

Listing 9: Configuration named.conf.local

```

1 //
2 // Do any local configuration here
3 //
4
5 // Consider adding the 1918 zones here, if they are not used in your
6 // organization
7 //include "/etc/bind/zones.rfc1918";
8
9 #La zone directe :
10 zone "benjamin.rt"{

```

```
11     type master;
12     file "/etc/bind/zones/db.benjamin.rt";
13     allow-transfer {192.168.1.40;};
14 };
15
16 #La zone inverse :
17 zone "1.168.192.in-addr.arpa"{
18     type master;
19     file "/etc/bind/zones/db.1.168.192.in-addr.arpa";
20     allow-transfer {192.168.1.40;};
21 };
```

Remarque

Nous déclarons une zone directe (`benjamin.rt`) et une zone inverse (`1.168.192.in-addr.arpa`). Le serveur est de type **master** car il est utilisé comme serveur DNS primaire. La ligne `allow-transfer` autorise le transfert de zone au serveur DNS secondaire (`192.168.1.40`).

4.6 Création du dossier zones

Avec la commande `mkdir zones`, nous créons le dossier zones dans lequel seront stockés les fichiers de configuration, car ce dernier n'existe pas par défaut.

4.7 Création du fichier de zone directe

Nous créons le fichier de zone avec la commande `sudo nano db.benjamin.rt` :

Listing 10: Fichier de zone directe db.benjamin.rt

```
1 ;
2 ; BIND reverse data file for local loopback interface
3 ;
4 $TTL      604800
5 @ IN SOA  ns.benjamin.rt. root.benjamin.rt. (
6             1             ; Serial
7             604800        ; Refresh
8             86400         ; Retry
9             2419200       ; Expire
10            604800 )      ; Negative Cache TTL
11 ;
12 @      IN NS   ns.benjamin.rt.
13 ns     IN A    192.168.1.20
14 web    IN A    192.168.1.30
```

Remarque

Explication des enregistrements :

- **SOA (Start of Authority)** : `ns.benjamin.rt` est le serveur qui fait autorité sur la zone
- **@ IN NS** : `ns.benjamin.rt` sera utilisé comme nom de domaine de la zone
- **ns IN A** : Attribue à une machine physique configurée en IPv4 l'autorité sur la zone
- **web IN A** : Attribue un nom de domaine au serveur web (`web.benjamin.rt`)

4.8 Création du fichier de zone inverse

Après avoir configuré le fichier de zone directe, nous créons le fichier de zone inverse `db.1.168.192.in-addr.arpa` :

Listing 11: Fichier de zone inverse

```

1 ;
2 ; BIND reverse data file for local loopback interface
3 ;
4 $TTL      604800
5 @   IN  SOA      ns.benjamin.rt. root.benjamin.rt. (
6           1           ; Serial
7           604800      ; Refresh
8           86400       ; Retry
9           2419200     ; Expire
10          604800 )    ; Negative Cache TTL
11 ;
12 @       IN  NS    ns.benjamin.rt.
13 20      IN  PTR    ns.benjamin.rt.
14 30      IN  PTR    web.benjamin.rt.
```

Remarque

Les enregistrements **PTR (Pointed to Record)** permettent de pointer une adresse IP vers un nom de domaine :

- 20 IN PTR : Pointe 192.168.1.20 vers `ns.benjamin.rt`
- 30 IN PTR : Pointe 192.168.1.30 vers `web.benjamin.rt`

4.9 Test de configuration des fichiers de zone

Redémarrage du service `bind9` pour prendre en compte les nouvelles configurations :

Listing 12: Redémarrage de `bind9`

```
benjamin@dns_server:/etc/bind/zones$ sudo systemctl restart bind9
```


Vérification que les fichiers de zones peuvent être chargés :

Listing 13: Vérification de la configuration

```
benjamin@dns_server:/etc/bind/zones$ named-checkconf -z /etc/bind/named.conf
zone benjamin.rt/IN: loaded serial 1
zone 1.168.192.in-addr.arpa/IN: loaded serial 1
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
benjamin@dns_server:/etc/bind/zones$ _
```

Vérification de la syntaxe des fichiers de zones :

Listing 14: Vérification de la syntaxe

```
benjamin@dns_server:/etc/bind/zones$ sudo named-checkzone
/etc/bind/zones/db.benjamin.rt /etc/bind/zones/db.1.168.192.in-addr.arpa
zone /etc/bind/zones/db.benjamin.rt/IN: loaded serial 2
OK
benjamin@dns_server:/etc/bind/zones$ _
```

4.10 Tests de résolution avec nslookup

4.10.1 Test de résolution directe

Listing 15: Résolution directe avec nslookup

```
benjamin@dns_server:/etc/bind/zones$ nslookup ns.benjamin.rt 192.168.1.20
Server:          192.168.1.20
Address:         192.168.1.20#53

Name:   ns.benjamin.rt
Address: 192.168.1.20
```

Listing 16: Résolution du serveur web

```
benjamin@dns_server:/etc/bind/zones$ nslookup web.benjamin.rt 192.168.1.20
Server:          192.168.1.20
Address:         192.168.1.20#53

Name:   web.benjamin.rt
Address: 192.168.1.30
```

Remarque

Les résolutions d'adresse fonctionnent. On tape le nom de domaine suivi de l'adresse IP du serveur DNS que l'on souhaite interroger.

4.10.2 Test de résolution inverse

Listing 17: Résolution inverse

```
benjamin@dns_server:/etc/bind/zones$ nslookup 192.168.1.20 192.168.1.20
20.1.168.192.in-addr.arpa      name = ns.benjamin.rt.
```

Listing 18: Résolution inverse du serveur web

```
benjamin@dns_server:/etc/bind/zones$ nslookup 192.168.1.30 192.168.1.20
30.1.168.192.in-addr.arpa      name = web.benjamin.rt.
```

Remarque

En tapant les adresses IP suivies de l'adresse du serveur DNS à interroger, on obtient les noms de domaine associés.

4.11 Test d'accès web par nom de domaine

Après avoir configuré notre machine cliente pour qu'elle utilise le serveur DNS primaire (192.168.1.20), nous pouvons accéder à la page web hébergée sur le serveur web via le nom de domaine `web.benjamin.rt`.

Remarque

La page web s'affiche correctement avec le titre "Serveur Web de GRONDIN Benjamin", confirmant que la résolution DNS fonctionne parfaitement.

5 Mise en place et configuration du serveur DNS secondaire

5.1 Configuration de l'interface enp0s3

Nous effectuons la configuration IP statique comme précédemment :

Listing 19: Configuration netplan DNS secondaire

```
1 network:
2   version: 2
3   renderer: networkd
4   ethernets:
5     enp0s3:
6       dhcp4: no
7       addresses: [192.168.1.40/24]
8       gateway4: 192.168.1.1
9       nameservers:
10        addresses: [8.8.8.8]
```

5.2 Installation du serveur bind9

Installation du package bind9 qui fera office de serveur DNS secondaire :

Listing 20: Installation de Bind9

```
benjamin@dns_server:/etc/netplan$ sudo apt install bind9
```

5.3 Vérification de l'installation

Comme pour le serveur DNS primaire, nous vérifions que l'installation a bien été effectuée avec `systemctl status bind9`.

5.4 Configuration du fichier `named.conf.local`

Après avoir installé bind9, nous configurons le fichier `named.conf.local` :

Listing 21: Configuration `named.conf.local` (secondaire)

```
1 #La zone directe :
2 zone "benjamin.rt"{
3     type slave;
4     file "/var/cache/bind/db.benjamin.rt";
5     masters {192.168.1.20;};
6 };
7
8 #La zone inverse :
9 zone "1.168.192.in-addr.arpa"{
10    type slave;
11    file "/var/cache/bind/db.1.168.192.in-addr.arpa";
12    masters {192.168.1.20;};
13 };
```

Remarque

À la différence du fichier configuré sur le serveur DNS primaire, le type est **slave**, signifiant qu'il recevra les configurations du serveur DNS primaire. Le paramètre `masters{192.168.1.20;}` spécifie l'adresse du serveur maître.

5.5 Configuration pour le transfert de zone

Sur le serveur DNS primaire, nous modifions le numéro de série (serial) dans les fichiers de zone pour initier le transfert :

Listing 22: Modification du serial

```
1 $TTL    604800
2 @ IN SOA ns.benjamin.rt. root.benjamin.rt. (
3         2_          ; Serial (modifi de 1  2)
4         604800      ; Refresh
5         86400       ; Retry
6         2419200     ; Expire
7         604800 )    ; Negative Cache TTL
```

Remarque

Le changement du numéro de série (de 1 à 2) permettra au serveur DNS secondaire de détecter qu'une mise à jour est disponible et d'initier le transfert de zone.

5.6 Vérification du transfert de zone

Redémarrage du serveur DNS secondaire :

Listing 23: Redémarrage du serveur secondaire

```
benjamin@dns-secondaire:/etc/bind$ sudo reboot
```

Vérification dans les logs système :

Listing 24: Vérification du transfert dans syslog

```
W IN': 2600:9000:5303:8900::1#53
Oct  5 07:21:54 dns-secondaire named[1465]: zone benjamin.rt/IN:
                Transfer started.
Oct  5 07:21:54 dns-secondaire named[1465]: transfer of 'benjamin.rt/IN'
                from 192.168.1.20#53: connected using 192.168.1.40#41813
Oct  5 07:21:54 dns-secondaire named[1465]: zone benjamin.rt/IN:
                transferred serial 1
Oct  5 07:21:54 dns-secondaire named[1465]: transfer of 'benjamin.rt/IN'
                from 192.168.1.20#53: Transfer status: success
Oct  5 07:21:54 dns-secondaire named[1465]: transfer of 'benjamin.rt/IN'
                from 192.168.1.20#53: Transfer completed: 1 messages,
                5 records, 155 bytes, 0.002 secs (77500 bytes/sec)
```

Remarque

On peut voir dans les logs que le transfert de zone a bien été effectué avec succès.

5.7 Tests de résolution avec le serveur secondaire

5.7.1 Résolution directe

Listing 25: Test de résolution directe via DNS secondaire

```
benjamin@dns_server:/etc/bind/zones$ nslookup ns.benjamin.rt 192.168.1.40
Server:          192.168.1.40
Address:         192.168.1.40#53

Name:   ns.benjamin.rt
Address: 192.168.1.20
```

Listing 26: Résolution du serveur web via DNS secondaire

```
benjamin@dns_server:/etc/bind/zones$ nslookup web.benjamin.rt 192.168.1.40
Server:          192.168.1.40
Address:         192.168.1.40#53

Name:   web.benjamin.rt
Address: 192.168.1.30
```

5.7.2 Résolution inverse

Listing 27: Test de résolution inverse via DNS secondaire

```
benjamin@dns_server:/etc/bind/zones$ nslookup 192.168.1.20 192.168.1.40
20.1.168.192.in-addr.arpa      name = ns.benjamin.rt.
```

Listing 28: Résolution inverse serveur web

```
benjamin@dns_server:/etc/bind/zones$ nslookup 192.168.1.30 192.168.1.40
30.1.168.192.in-addr.arpa      name = web.benjamin.rt.
```

Remarque

On constate qu'il est possible d'obtenir des informations sur les noms de domaine en interrogeant le serveur DNS secondaire. La résolution inverse fonctionne également parfaitement.

5.8 Test d'accès web via le DNS secondaire

Nous configurons maintenant notre machine cliente pour qu'elle interroge le serveur DNS secondaire (192.168.1.40) pour accéder au site `web.benjamin.rt`.

Remarque

Le site web est bien accessible via le serveur DNS secondaire. Pour être certain que les données du site n'ont pas été enregistrées au préalable dans le cache du navigateur lors du premier essai et qu'il interroge bien le serveur DNS secondaire, nous changeons de navigateur web pour effectuer le test.

6 Conclusion

L'objectif de ce TP était de mettre en œuvre trois serveurs : un serveur DNS primaire, un serveur DNS secondaire et un serveur web.

Réalisations accomplies :

- Configuration d'un serveur DNS primaire avec autorité sur la zone `benjamin.rt`
- Création des fichiers de zone directe et inverse
- Mise en place d'un serveur web Apache accessible via nom de domaine
- Déploiement d'un serveur DNS secondaire avec transfert de zone fonctionnel
- Tests de résolution directe et inverse réussis sur les deux serveurs DNS
- Vérification de l'accès web par nom de domaine via les deux serveurs DNS

Après avoir mis en place les différentes machines et réalisé les configurations, nous avons réussi à faire fonctionner les trois serveurs. De plus, nous avons réussi à tester les configurations des machines et obtenir les résultats attendus. Le transfert de zone entre le serveur DNS primaire et secondaire fonctionne correctement, assurant ainsi la redondance du service DNS.

Remarque

En revanche, ayant réalisé le TP chez moi, je n'ai pas pu continuer les parties 4 et 5 du TP.