

TP 1 DNS - R303

GRONDIN Benjamin - BUT 2 Réseau et télécommunication

Table des matières

Introduction.....	2
Objectif de ce TP.....	2
Table d'adressage.....	2
1.1 Mise en place et configuration du serveur web.....	3
Configuration de l'interface enp0s3 sur le serveur DNS primaire.....	3
Installation du serveur apache.....	3
Modification et vérification de la page web.....	4
1.2 - Mise en place et configuration du serveur DNS primaire.....	5
Configuration de l'interface enp0s3 sur le serveur DNS primaire.....	5
Enregistrer les configurations sur l'interface.....	5
Installation du serveur Bind9.....	5
Vérification de l'installation de bin9.....	6
Configuration du fichier named.conf.local.....	6
Création du dossier zones.....	7
Création et configuration du fichier de la zone directe db.benjamin.rt.....	7
Création et édition du fichier de la zone inverse db.1.168.192.in-addr.arpa.....	8
Test de configuration des fichiers de zone.....	8
Test de résolution directe avec nslookup.....	9
Test de résolution inverse avec nslookup.....	9
Test de la page web avec le nom de domaine associée.....	10
1.3 - Mise en place et configuration du serveur DNS secondaire.....	10
Configuration de l'interface enp0s3 sur le serveur DNS secondaire.....	10
Installation du serveur bind9.....	10
Vérification de l'installation de bin9.....	11
Configuration du fichier named.conf.local.....	11
Configuration des fichiers de zone directe et inverse sur le serveur DNS primaire....	11
Vérification du transfert de zone.....	12
Test de la page web avec le nom de domaine associée.....	13
Compte rendu.....	14

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 héberge ces services. Ainsi, avec le développement d'internet et la multiplication des services en lignes 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. Pour cela, a été mise en place une solution très sollicitée aujourd'hui dans nos réseaux informatiques, le DNS ou Domain Name System. Le DNS a pour rôle d'associer à une adresse IP un nom de domaine pleinement qualifié. Par exemple, au lieu de taper l'adresse "142.250.187.227" dans votre navigateur vous taperez plutôt www.google.fr. Ainsi, il existe dans le monde une multitude de serveurs DNS qui communiquent entre eux pour aider les utilisateurs à trouver un site donné.

Objectif de ce TP

Dans ce TP nous allons mettre en place une architecture réseau comprenant 3 machines distinctes. Ainsi, nous déployerons un serveur DNS primaire qui fera autorité sur la zone benjamin.rt et un serveur Web qui héberge une page Html que nous accèderons par un nom de domaine configuré sur le serveur DNS primaire. Pour finir, nous déployerons un serveur DNS secondaire qui sera redondant en cas de panne du serveur DNS primaire. Lors de ce TP, Nous utiliserons des serveurs ubuntu 18.04 pour les serveur DNS primaire et secondaire, ainsi que pour le serveur Web.

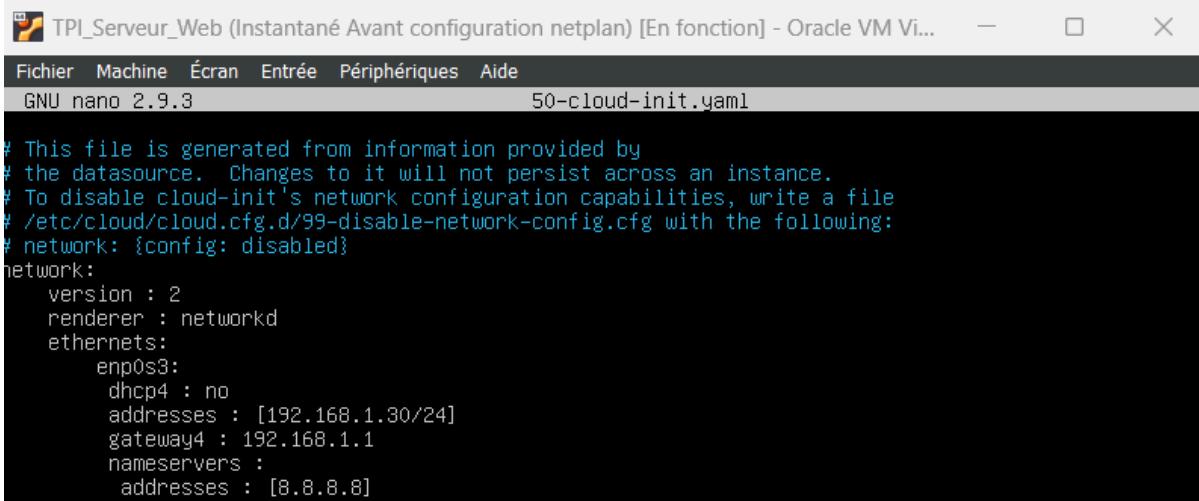
Table d'adressage

Machines	Adresse IP	Masques
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

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

1.1 Mise en place et configuration du serveur web

Configuration de l'interface enp0s3 sur le serveur DNS primaire

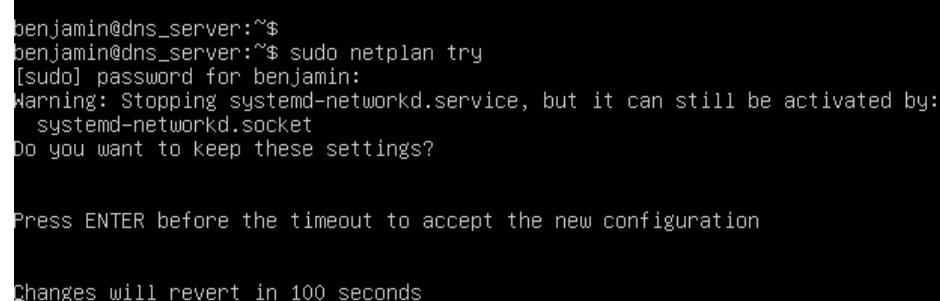


The screenshot shows a terminal window titled "TPI_Serveur_Web (Instantané Avant configuration netplan) [En fonction] - Oracle VM Vi...". The window contains the content of the 50-cloud-init.yaml file. The file is a YAML configuration for network interfaces. It includes a header note about being generated from a datasource and persisting across instances. It defines a network interface 'enp0s3' with version 2, renderer 'networkd', and static IP configuration (addresses: 192.168.1.30/24, gateway: 192.168.1.1, nameservers: 8.8.8.8).

```
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}

network:
  version : 2
  renderer : networkd
  ethernets:
    enp0s3:
      dhcp4 : no
      addresses : [192.168.1.30/24]
      gateway4 : 192.168.1.1
      nameservers :
        addresses : [8.8.8.8]
```

On commence par modifier le fichier 50-cloud-init.yaml qui contient les configurations IP et qui se situe dans le dossier /etc/netplan. Dans ce fichier on ajoutera les lignes comme montrées ci-dessus pour que l'adresse IP soit statique.



The screenshot shows a terminal session where the user runs the command `sudo netplan try`. The system prompts for a password for the user 'benjamin'. It then displays a warning about stopping the `systemd-networkd.service` and asks if the user wants to keep the settings. The user is instructed to press Enter to accept the new configuration. A message at the bottom indicates that changes will revert in 100 seconds.

```
benjamin@dns_server:~$ 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

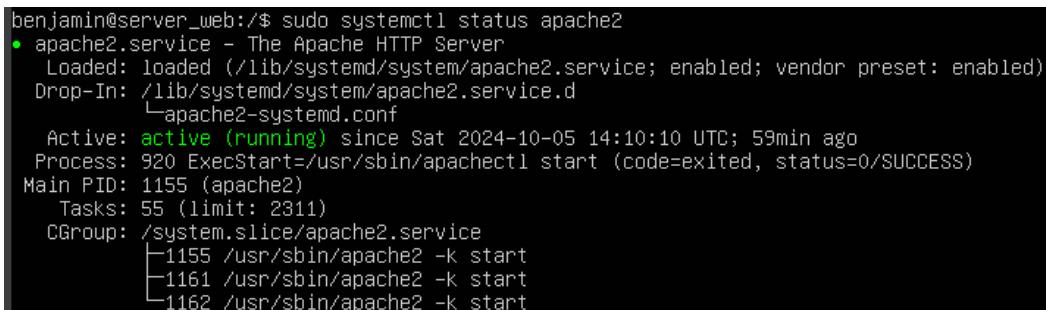
Changes will revert in 100 seconds
```

On peut tester la configuration réseau et appuyer sur Entrer si il n'y a pas de problème dans le fichier.

Installation du serveur apache

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

Après avoir configuré l'adresse IP on peut installer le serveur apache. Pour ce faire, tapez la ligne de commande montrée ci-dessus.



The screenshot shows the output of the command `sudo systemctl status apache2`. It displays the status of the `apache2.service` unit, which is active and running. It provides details about the process ID (Main PID: 1155), tasks (55), and cgroups (1155, 1161, 1162) associated with the service.

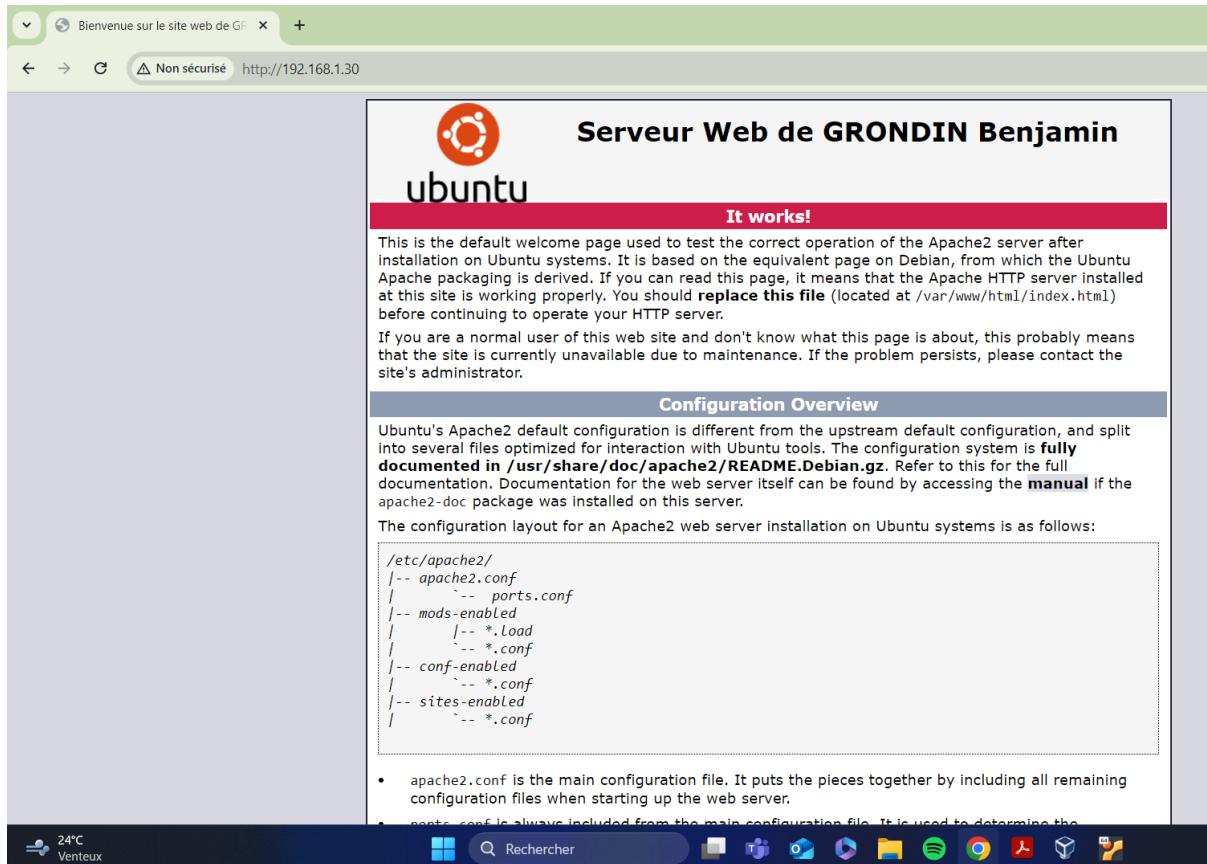
```
benjamin@server_web:/$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
             └─apache2-systemd.conf
     Active: active (running) since Sat 2024-10-05 14:10:10 UTC; 59min 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.

Modification et vérification de la page web

```
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
```

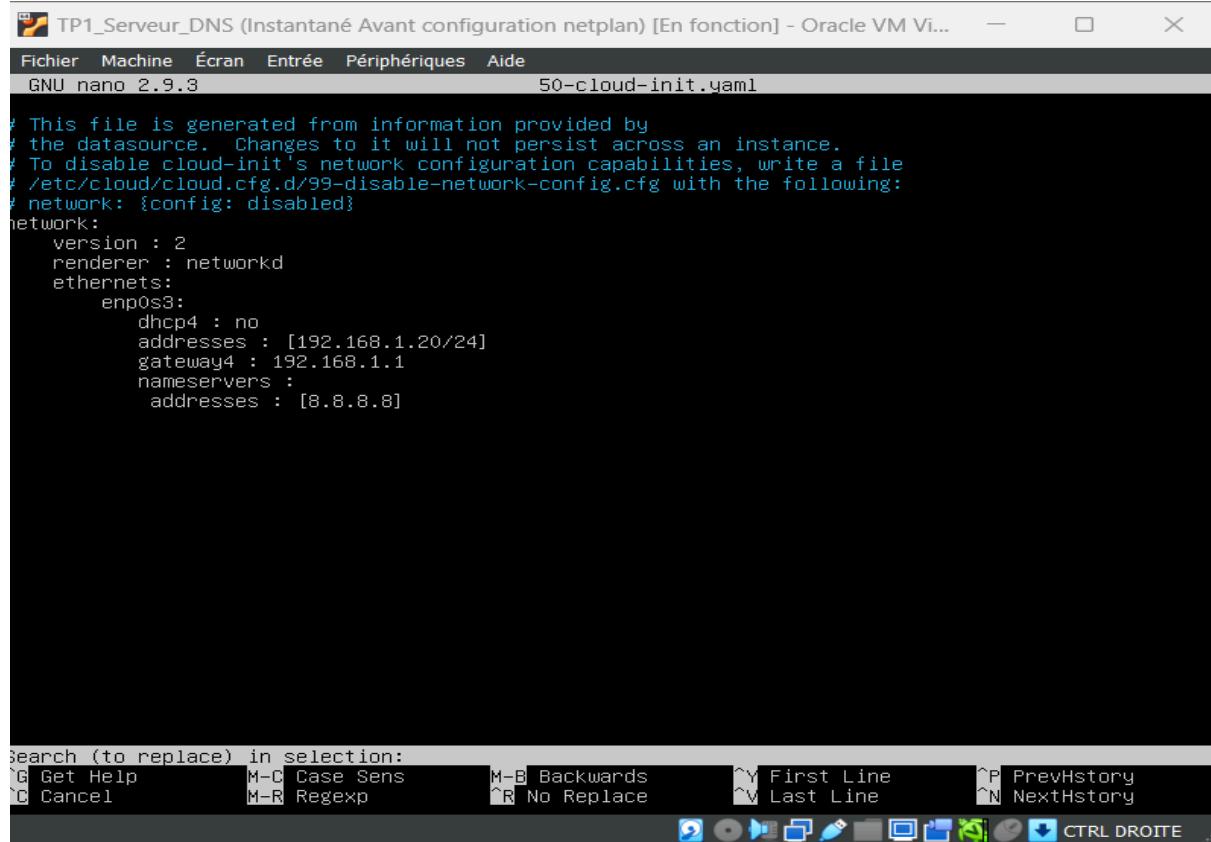
Dans le fichier index.html situé dans le dossier `/var/www/html/` on peut configurer le fichier html et la personnaliser afin d'identifier notre serveur.



On peut voir ci-dessus 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 mais par un nom de domaine.

1.2 - Mise en place et configuration du serveur DNS primaire

Configuration de l'interface enp0s3 sur le serveur DNS primaire



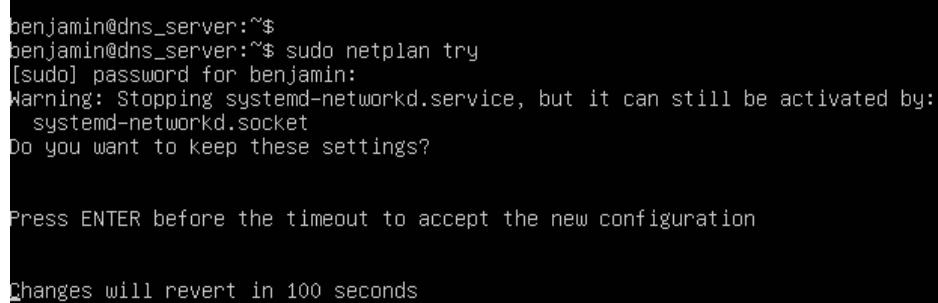
The screenshot shows a terminal window titled "TP1_Serveur_DNS (Instantané Avant configuration netplan) [En fonction] - Oracle VM Vi...". The window contains the content of the file "50-cloud-init.yaml". The file defines a network interface "enp0s3" with static IP settings: address [192.168.1.20/24], gateway [192.168.1.1], and nameservers [8.8.8.8]. The terminal also displays a search bar and various keyboard shortcuts for navigating the file.

```
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}

network:
  version : 2
  renderer : networkd
  ethernets:
    enp0s3:
      dhcp4 : no
      addresses : [192.168.1.20/24]
      gateway4 : 192.168.1.1
      nameservers :
        addresses : [8.8.8.8]
```

Pour commencer, il faut configurer les interfaces réseau de nos machines virtuelles. Pour ce faire, on commence par attribuer des adresses statiques à nos interfaces comme montrée ci-dessus. Ici aussi, Pour configurer l'interface réseau il faut aller dans le fichier 50-cloud-init.yaml situé dans dossier /etc/netplan.

Enregistrer les configurations sur l'interface



The screenshot shows a terminal window with the following commands and output:

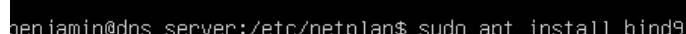
```
benjamin@dns_server:~$ sudo netplan try
[sudo] password for benjamin:
Warning: Stopping systemd-networkd.service, but it can still be activated by:
  systemd-networkd.socket
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 100 seconds
```

Avec la commande "netplan try" on peut tester le fichier de configuration 50-cloud-init.yaml afin de vérifier les possibles erreurs. Si il n'y a pas d'erreur dans le fichier, on applique la configuration en appuyant sur la touche Entrer.

Installation du serveur Bind9



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

Après avoir configuré les interfaces réseau, on peut à présent installer le package bind9 avec la commande ci-dessus.

Vérification de l'installation de 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
          └─845 /usr/sbin/named -f -u bind
```

Avec la commande systemctl status bind9 on voit que le service bind9 est bien installé et activé. On peut à présent configurer notre serveur DNS primaire.

Configuration du fichier named.conf.local

```
TP1_Serveur_DNS (Avant suppression named.conf.local) [En fonction] - Oracle VM

Fichier Machine Écran Entrée Périphériques Aide
GNU nano 2.9.3                               named.conf.local

// Do any local configuration here
//

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

#La zone directe :
zone "benjamin.rt"{
  type master;
  file "/etc/bind/zones/db.benjamin.rt";
  allow-transfer {192.168.1.40;};
};

#La zone inverse :
zone "1.168.192.in-addr.arpa"{
  type master;
  file "/etc/bind/zones/db.1.168.192.in-addr.arpa";
  allow-transfer {192.168.1.40;};
};
```

A présent, il faut configurer le fichier named.conf.local comme ci-dessus afin de déclarer une zone directe, ici benjamin.rt, et une zone inverse, 1.168.192.in-addr.arpa. Ensuite, on indique que ce serveur est de type Master car il est utilisé comme serveur DNS primaires sur la zone. La ligne allow-transfer est utilisée pour autoriser le transfert de zone au serveur DNS secondaire ayant pour adresse 192.168.1.40, elle nous servira pour l'autre partie de ce TP. Ceci étant fait, il faut désormais créer les fichiers db.benjamin.rt et 1.168.192.in-addr.arpa dans le dossier zones.

Création du dossier zones

```
benjamin@dns_server:/etc/bind$ ls -l
total 68
-rw-r--r-- 1 root root 2761 Sep 20 2022 bind.keys
-rw-r--r-- 1 root root 237 Sep 30 2019 db.0
-rw-r--r-- 1 root bind 271 Sep 22 12:56 db.1.168.192.in-addr.arpa
-rw-r--r-- 1 root root 271 Sep 30 2019 db.127
-rw-r--r-- 1 root root 237 Sep 30 2019 db.255
-rw-r--r-- 1 root bind 271 Sep 15 13:39 db.benjamin.rt
-rw-r--r-- 1 root root 353 Sep 30 2019 db.empty
-rw-r--r-- 1 root root 270 Sep 30 2019 db.local
-rw-r--r-- 1 root root 3171 Sep 30 2019 db.root
-rw-r--r-- 1 root bind 463 Sep 30 2019 named.conf
-rw-r--r-- 1 root bind 490 Sep 30 2019 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Sep 30 2019 named.conf.local
-rw-r--r-- 1 root bind 165 Sep 15 13:17 named.conf.local-copy
-rw-r--r-- 1 root bind 381 Sep 15 13:28 named.conf.locale
-rw-r--r-- 1 root bind 890 Sep 30 2019 named.conf.options
-rw-r----- 1 bind bind 77 Sep 15 07:54 rndc.key
-rw-r--r-- 1 root root 1317 Sep 30 2019 zones.rfc1918
benjamin@dns_server:/etc/bind$
```

```
benjamin@dns_server:/etc/bind$ ls -l
total 64
-rw-r--r-- 1 root root 2761 Sep 20 2022 bind.keys
-rw-r--r-- 1 root root 237 Sep 30 2019 db.0
-rw-r--r-- 1 root root 271 Sep 30 2019 db.127
-rw-r--r-- 1 root root 237 Sep 30 2019 db.255
-rw-r--r-- 1 root root 353 Sep 30 2019 db.empty
-rw-r--r-- 1 root root 270 Sep 30 2019 db.local
-rw-r--r-- 1 root root 3171 Sep 30 2019 db.root
-rw-r--r-- 1 root bind 463 Sep 30 2019 named.conf
-rw-r--r-- 1 root bind 490 Sep 30 2019 named.conf.default-zones
-rw-r--r-- 1 root bind 165 Sep 30 2019 named.conf.local
-rw-r--r-- 1 root bind 165 Sep 15 13:17 named.conf.local-copy
-rw-r--r-- 1 root bind 381 Sep 15 13:28 named.conf.locale
-rw-r--r-- 1 root bind 890 Sep 30 2019 named.conf.options
-rw-r----- 1 bind bind 77 Sep 15 07:54 rndc.key
drwxr-sr-x 2 root bind 4096 Sep 22 13:07 zones
-rw-r--r-- 1 root root 1317 Sep 30 2019 zones.rfc1918
```

Avec la commande “mkdir zones”, on créer le dossier zones dans laquelle sera stocké les fichiers de configurations de zones car ce dernier n'existe pas.

Création et configuration du fichier de la zone directe db.benjamin.rt

```
GNU nano 2.9.3                                     db.benjamin.rt

;
; BIND reverse data file for local loopback interface
;

$TTL    604800
@      IN      SOA     ns.benjamin.rt. root.benjamin.rt. (
                      1           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                     604800 )    ; Negative Cache TTL
;
@      IN      NS      ns.benjamin.rt.
ns    IN      A       192.168.1.20
web  IN      A       192.168.1.30
```

On crée le fichier de zone avec la commande “sudo nano db.benjamin.rt”. Par la suite, dans le fichier de zone on ajoute les lignes présentées ci-dessus. On peut voir à côté du SOA, qui signifie Start of authority, que ns.benjamin.rt est le serveur qui fait autorité sur la zone. Puis, la ligne @ IN NS ns.benjamin.rt qui signe que ns.benjamin.rt sera utilisée comme nom de domaine de la zone. Ensuite, ns IN A 192.168.1.20 permet d'attribuer à une machine physique configurée en ipv4 l'autorité sur la zone. Pour finir, web IN A permet d'attribuer un nom de domaine au serveur web. Ici, le nom de domaine du serveur web sera web.benjamin.rt.

Création et édition du fichier de la zone inverse db.1.168.192.in-addr.arpa

```
GNU nano 2.9.3                               db.1.168.192.in-addr.arpa                         Modified

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@      IN      SOA     ns.benjamin.rt. root.benjamin.rt. (
                      1          ; Serial
                      604800    ; Refresh
                      86400     ; Retry
                     2419200   ; Expire
                      604800 )  ; Negative Cache TTL
;
@      IN      NS      ns.benjamin.rt.
20    IN      PTR     ns.benjamin.rt.
30    IN      PTR     web.benjamin.rt.
```

Après avoir configuré le fichier de zone directe, il faut à présent créer le fichier de zone indirecte db.1.168.192.in-addr.arpa. On peut voir sur cette capture d'écrans que les fichiers de la zone inverse et directe se ressemblent. En revanche, on peut voir deux lignes qui diffèrent. Premièrement, 20 IN PTR, PTR signifie Pointed to record et il permet de pointer l'adresse 192.168.1.20 vers le nom de domaine ns.benjamin.rt. Deuxièmement, la ligne du dessous pointera vers le site web.benjamin.rt.

Test de configuration des fichiers de zone

```
benjamin@dns_server:/etc/bind/zones$ sudo systemctl restart bind9_
```

Avec cette commande on redémarre le service bind9 afin que les nouvelles configurations soient prises en compte.

```
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$ _
```

Ensuite, avec la commande *named-checkconf -z /etc/bind/named.conf*, on vérifie que les fichiers de zones peuvent être chargés. Ici, c'est le cas.

```
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$ _
```

Avec la commande ci-dessus, *named-checkzone /etc/bind/zones/db.benjamin.rt /etc/bind/zones/db.1.168.192.in-addr.arpa*, on vérifie que les fichiers de zones sont bien configurés. On peut voir que oui car il en ressort OK.

Test de 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
```

```
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
```

On peut voir sur les deux captures d'écrans ci-dessus que les résolutions d'adresse fonctionnent. Ainsi, on peut voir dans les lignes de commande que l'on a tapé le nom de domaine suivie de l'adresse ip du serveur DNS que l'on souhaite interroger pour trouver l'information.

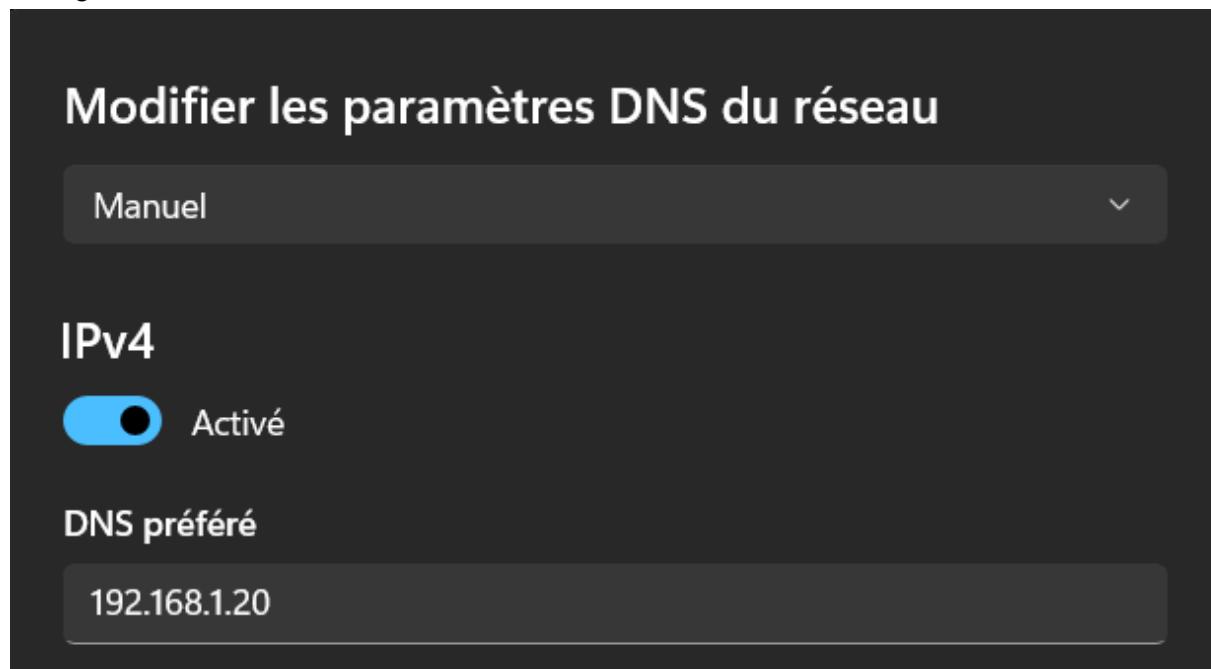
Test de résolution inverse avec nslookup

```
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.
```

```
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.
```

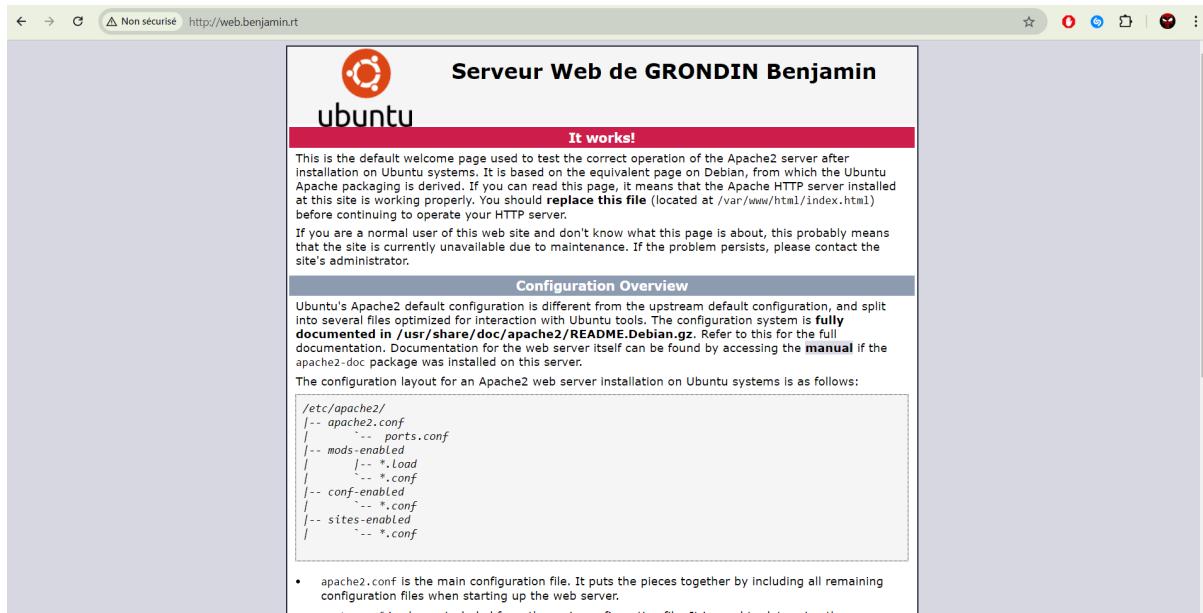
Ici, on peut voir qu'en tapant les adresses IP du serveur web ou du serveur de DNS, toujours suivie de l'adresse du serveur DNS que l'on souhaite interroger, on obtient les noms de domaine associées.

Configuration du DNS sur une machine cliente



Pour pouvoir interroger le serveur DNS et accéder à la page web il faut à présent configurer notre machine en entrant l'adresse ip du serveur DNS primaire comme ci-dessus.

Test de la page web avec le nom de domaine associée



Pour finir on peut voir que l'on peut accéder à la page web hébergée sur le serveur web 192.168.1.30 via le nom de domaine pleinement qualifié web.benjamin.rt.

1.3 - Mise en place et configuration du serveur DNS secondaire

Configuration de l'interface enp0s3 sur le serveur DNS secondaire

```
GNU nano 2.9.3                                     50-cloud-init.yaml

# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    version : 2
    renderer : networkd
    ethernets:
        enp0s3:
            dhcp4 : no
            addresses : [192.168.1.40/24]
            gateway4 : 192.168.1.1
            nameservers :
                addresses : [8.8.8.8]
```

On effectue la configuration IP statique comme dans la partie 1.1 et 1.2.

Installation du serveur bind9

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

On installe le serveur bind9 qui fera office de serveur DNS secondaire.

Vérification de l'installation de 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
           └─845 /usr/sbin/named -f -u bind
```

Comme pour le serveur DNS primaire on vérifie que l'installation a bien été effectuée.

Configuration du fichier named.conf.local

```
#La zone directe :
zone "benjamin.rt"{
  type slave;
  file "/var/cache/bind/db.benjamin.rt";
  masters {192.168.1.20;};
};

#La zone inverse :
zone "1.168.192.in-addr.arpa"{
  type slave;
  file "/var/cache/bind/db.1.168.192.in-addr.arpa";
  masters {192.168.1.20;};
};
```

Après avoir installé bind9, il faut maintenant configurer le fichier named.conf.local. A la différence du même fichier configuré sur le serveur DNS primaire, on peut noter que le type slave signifie qu'il recevra les configuration du serveur DNS primaire dont le rôle et l'adresse est spécifié plus bas par le paramètre master{192.168.1.20;};.

Configuration des fichiers de zone directe et inverse sur le serveur DNS primaire

```
GNU nano 2.9.3                                     db.benjamin.rt

;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.benjamin.rt. root.benjamin.rt. (
                      2_              ; Serial
                      604800          ; Refresh
                      86400           ; Retry
                      2419200         ; Expire
                      604800 )        ; Negative Cache TTL
;
@       IN      NS      ns.benjamin.rt.
ns      IN      A       192.168.1.20
web    IN      A       192.168.1.30
```

```

GNU nano 2.9.3                               db.1.168.192.in-addr.arpa

;

; BIND reverse data file for local loopback interface
;

$TTL    604800
@       IN      SOA     ns.benjamin.rt. root.benjamin.rt. (
                      2_              ; Serial
                      604800          ; Refresh
                      86400           ; Retry
                     2419200         ; Expire
                     604800 )        ; Negative Cache TTL
;
@       IN      NS      ns.benjamin.rt.
20      IN      PTR     ns.benjamin.rt.
30      IN      PTR     web.benjamin.rt.

```

Il faut à présent modifier le sérial des fichiers db.benjamin.rt et db.1.168.192.in-addr.arpa. Ainsi, on modifie le serial en le passant de 1 à 2. Cela permettra le transfert de zone vers le DNS secondaire.

Vérification du transfert de zone

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

Pour confirmer les nouvelles configurations et initier le transfert de zone on redémarre le serveur DNS secondaire avec la commande ci-dessus.

```

A/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, 159 bytes, 0.002 secs (79500 bytes/sec)

```

Ensuite, on tape la commande `tail -f` suivie du chemin du fichier "/var/log/syslog" pour vérifier les nouvelles informations inscrites dans le fichier syslog. On peut voir dans la capture d'écrans ci-dessus que le transfert de zone a bien été effectué.

Test de résolution directe avec nslookup

```
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
```

```
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
```

On peut voir sur ces captures d'écrans qu'il est possible d'obtenir des informations sur les noms de domaine en interrogeant le serveur DNS secondaire

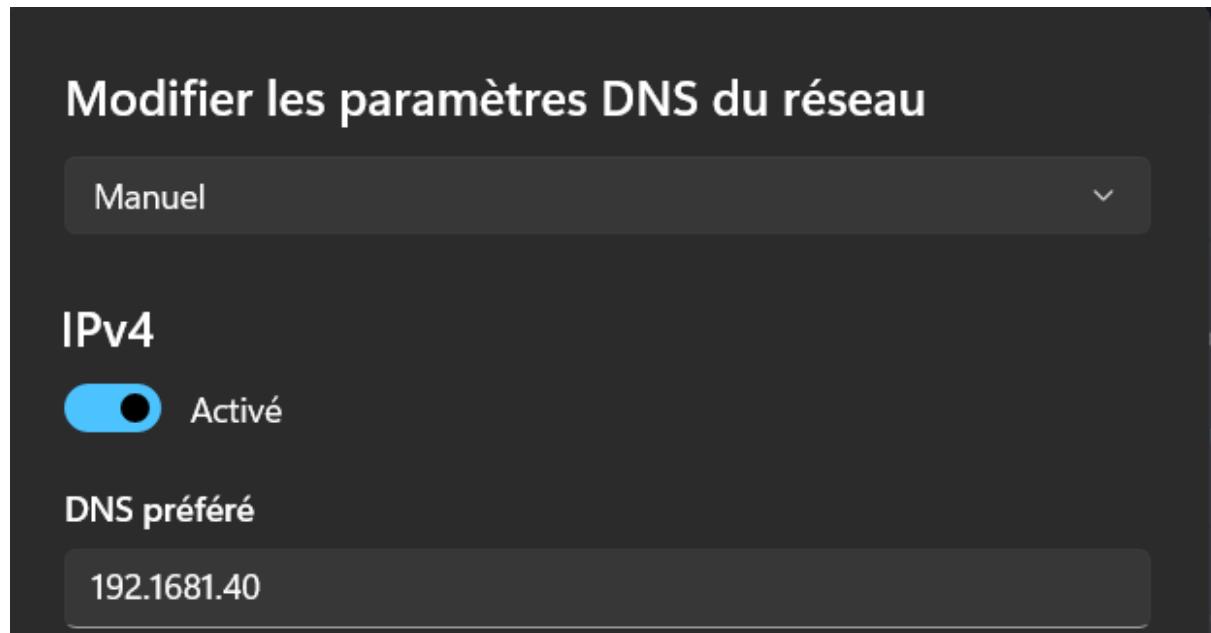
Test de résolution inverse avec nslookup

```
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.
```

```
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.
```

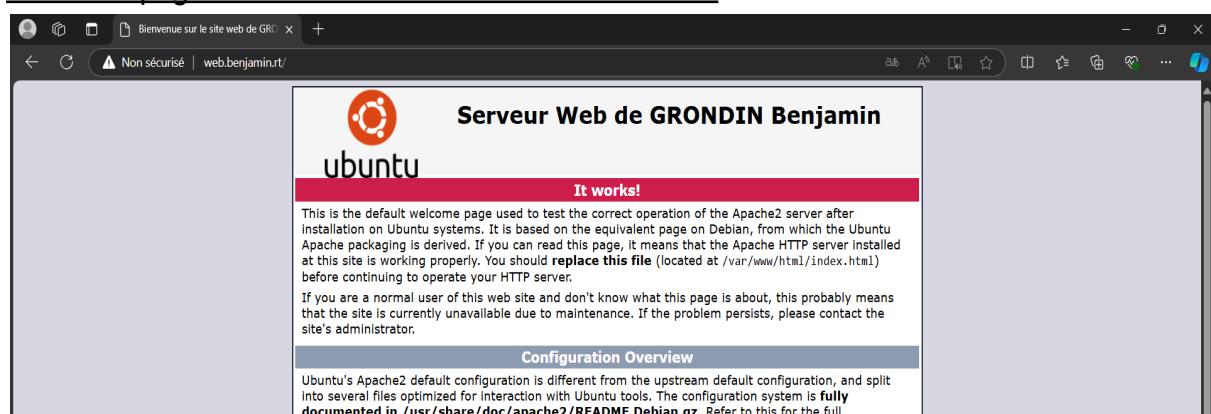
De même ici, où l'on peut voir que la résolution inverse fonctionne.

Configuration du DNS sur une machine cliente



On configure à présent notre machine cliente pour qu'elle interroge le serveur DNS secondaire pour accéder au site `web.benjamin.rt`. Ici, on configurera la machine cliente DNS secondaire comme serveur DNS primaire sur la machine cliente afin d'avoir la certitude qu'il sera le seul serveur interroger.

Test de la page web avec le nom de domaine associée



On peut voir ici que le site web est bien accessible. Pour être certain que les données du site n'ont pas été enregistrées au préalable dans le cache de notre navigateur lors du premier essaie et qu'il interroge bien le serveur DNS secondaire, on change de navigateur web.

Compte rendu

L'objectif de ce TP était de mettre en œuvre trois serveurs. Un serveur DNS primaire, un serveur DNS secondaire et un serveur web. Après avoir mis en place les différentes machines et réaliser 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. En revanche, ayant réalisé le TP cher moi je n'ai pas pu continuer la partie 4 et 5 du TP.