

I. Introduction Générale – Mise en place d'un serveur DNS et DHCP sous Linux

Dans toute infrastructure réseau, la configuration automatique et la résolution des noms sont deux piliers fondamentaux pour assurer le bon fonctionnement et la fluidité des échanges entre machines. Pour répondre à ces besoins, deux services essentiels entrent en jeu : le DHCP (Dynamic Host Configuration Protocol) et le DNS (Domain Name System).

Ce projet a pour objectif de mettre en place, sous un système Linux (type Debian/Ubuntu), un serveur local intégrant à la fois les rôles de serveur DHCP et de serveur DNS, afin de :

- Fournir dynamiquement des adresses IP aux clients du réseau local (via le DHCP),
- Et permettre la traduction des noms d'hôtes en adresses IP, et inversement (via le DNS).

1. Pourquoi ces deux services sont-ils complémentaires ?

Imagine un réseau local sans DHCP : chaque appareil doit être configuré manuellement, ce qui est source d'erreurs et de conflits d'adresse IP. Inversement, sans DNS, les utilisateurs seraient obligés d'accéder aux services via des adresses numériques — ce qui est peu convivial et difficile à maintenir.

En les combinant :

- Le serveur DHCP attribue automatiquement des adresses IP à chaque machine du réseau dès sa connexion ;
- Le serveur DNS permet ensuite à ces machines d'être identifiées par des noms clairs et personnalisés (comme serveur1.local), ce qui facilite la communication, la gestion et la maintenance.

2. Objectif pédagogique

Ce projet a également une portée pédagogique : comprendre, installer, configurer et tester ces deux services sous Linux permet de renforcer ses compétences en administration réseau, en scripting système, et en analyse de configuration. C'est aussi une étape essentielle pour quiconque souhaite se professionnaliser dans les domaines de la cybersécurité, de la gestion de parc informatique ou de l'automatisation des infrastructures.

II. Le serveur DNS

1. Qu'est-ce que le DNS ? (Définition détaillée et illustrée)

Le **DNS**, pour **Domain Name System**, est un système essentiel de l'infrastructure d'Internet et des réseaux locaux. Il agit comme un **service de traduction** entre les noms de domaine compréhensibles par l'humain (comme `www.google.com`) et les adresses IP numériques utilisées par les machines pour communiquer entre elles (comme `142.250.186.4`).

Sans le DNS, nous serions obligés de mémoriser des séries de chiffres pour chaque site ou machine que nous voulons contacter — ce qui, on en conviendra, serait impraticable.

2. Un peu d'histoire...

À la naissance d'Internet (dans les années 70-80), la correspondance entre les noms et les adresses IP était gérée **manuellement** dans un fichier unique appelé `hosts.txt`, maintenu par l'université de Stanford. Ce fichier était distribué régulièrement à tous les ordinateurs connectés.

Mais avec l'explosion du nombre de machines, ce système centralisé a vite montré ses limites : lenteur, erreurs, surcharge... C'est ainsi qu'en 1983, **Paul Mockapetris**, un ingénieur américain, a conçu le **DNS** tel qu'on le connaît aujourd'hui : un système **décentralisé, hiérarchique et distribué**, inspiré du fonctionnement d'un annuaire téléphonique ou d'un arbre généalogique.

3. Comment ça marche ?

Le DNS repose sur une structure en **arbre inversé**, avec :

- Une **racine** (`.`),
- Des **domaines de premier niveau** (comme `.com`, `.org`, `.net`, ou des pays comme `.fr`, `.cd`, etc.),
- Des **sous-domaines** (comme `openai.com`, `google.fr`, etc.),
- Et enfin des **noms d'hôtes** (comme `chat.openai.com` ou `serveur1.monreseau.local`).

Quand un utilisateur tape un nom dans son navigateur, le système DNS est consulté pour obtenir l'adresse IP correspondante. Ce processus peut passer par plusieurs serveurs :

- Le **DNS local** (souvent celui du FAI ou de l'entreprise),
- Puis les **serveurs racine**,
- Puis les **serveurs autoritaires** pour chaque niveau du domaine.

4. Et en réseau local ?

Dans un réseau interne (école, entreprise, labo...), on peut créer un **serveur DNS privé** pour :

- Réduire la dépendance à Internet,
- Accélérer la résolution de noms internes,
- Gérer des noms personnalisés comme `serveur1.local` ou `imprimante.monreseau.local`,

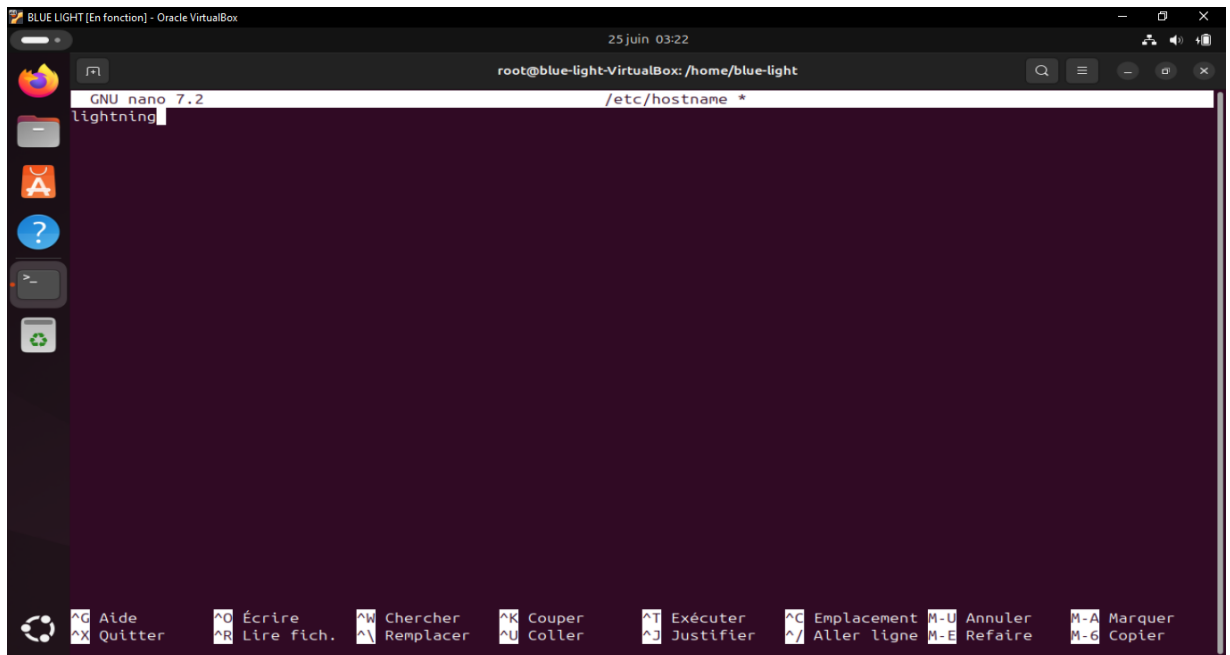
- Et même faire de la **résolution inverse** (retrouver le nom à partir de l'adresse IP).

C'est ce type de serveur que nous allons mettre en place dans ce projet, à l'aide du paquet **BIND9**, l'un des plus anciens et plus fiables serveurs DNS open-source.

5. Image du travail fait :

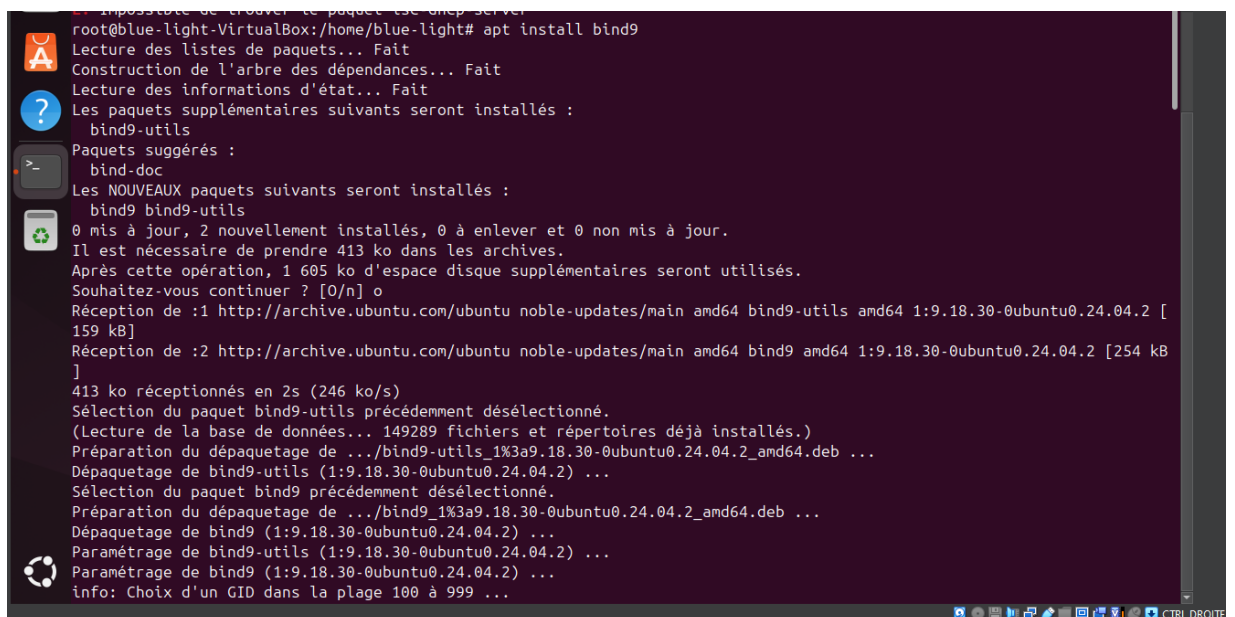
a. Réglage de l'`host_name` :

Avec la commande `nano /etc/hostname` pour régler le nom de la machine.



b. Installations de Bind9 :

Tout d'abord on installe Bind9. C'est la ressource nécessaire pour faire marcher le serveur.



```

info: Choix d'un GID dans la plage 100 à 999 ...
info: Ajout du groupe « bind » (GID 124)...
info: Choix d'un identifiant utilisateur dans la plage 100 à 999 ...

info: Ajout de l'utilisateur système « bind » (UID 122) ...
info: Ajout du nouvel utilisateur « bind » (UID 122) avec pour groupe d'appartenance « bind » ...
info: Pas de création du répertoire personnel « /var/cache/bind ».
wrote key file "/etc/bind/rndc.key"
named-resolvconf.service is a disabled or a static unit, not starting it.
Created symlink /etc/systemd/system/bind9.service → /usr/lib/systemd/system/named.service.
Created symlink /etc/systemd/system/multi-user.target.wants/named.service → /usr/lib/systemd/system/named.service.
Traitement des actions différées (« triggers ») pour man-db (2.12.0-4build2) ...
Traitement des actions différées (« triggers ») pour ufw (0.36.2-6) ...

```

c. Mise a jour des package :

Pour un meilleur confort dans la configuration il vaut mieux que toute les ressource soit à jour alors on va mettre les package à jour

```

root@blue-light-VirtualBox:/home/blue-light# apt install update
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait

```

J'ai plus la suite de la procedure.....

d. Installation de net-tools :

Déjà au début si on essaie de vérifier la configuration on a de grande chance de se manger un mur comme ceci

```

root@blue-light-VirtualBox:/home/blue-light# ifconfig
La commande « ifconfig » n'a pas été trouvée, mais peut être installée avec :
apt install net-tools

```

Alors on doit installer net-tools pour que les ressource de configuration marche

```

root@blue-light-VirtualBox:/home/blue-light# apt install net-tools
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  net-tools
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 204 ko dans les archives.
Après cette opération, 811 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://archive.ubuntu.com/ubuntu noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]
204 ko réceptionnés en 1s (138 ko/s)
Sélection du paquet net-tools précédemment désélectionné.
(Lecture de la base de données... 149377 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Dépaquetage de net-tools (2.10-0.1ubuntu4) ...
Paramétrage de net-tools (2.10-0.1ubuntu4) ...
Traitement des actions différées (« triggers ») pour man-db (2.12.0-4build2) ...

```

Voilà déjà comme ca on est prêt pour la suite...

e. Configuration du serveur :

On commence par configurer le serveur avec le point d'entrer ou le lien qui définit notre serveur dans notre cas comme dans le but de rester fidel au model on a pris l'identifiant : 192.168.20.1.

```

root@blue-light-VirtualBox:/home/blue-light# ifconfig enp0s3 192.168.20.1
root@blue-light-VirtualBox:/home/blue-light# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.20.1 netmask 255.255.255.0 broadcast 192.168.20.255
    inet6 fe80::a00:27ff:fe99:f9d9 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:99:f9:d9 txqueuelen 1000 (Ethernet)
    RX packets 10862 bytes 12746475 (12.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3004 bytes 409928 (409.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

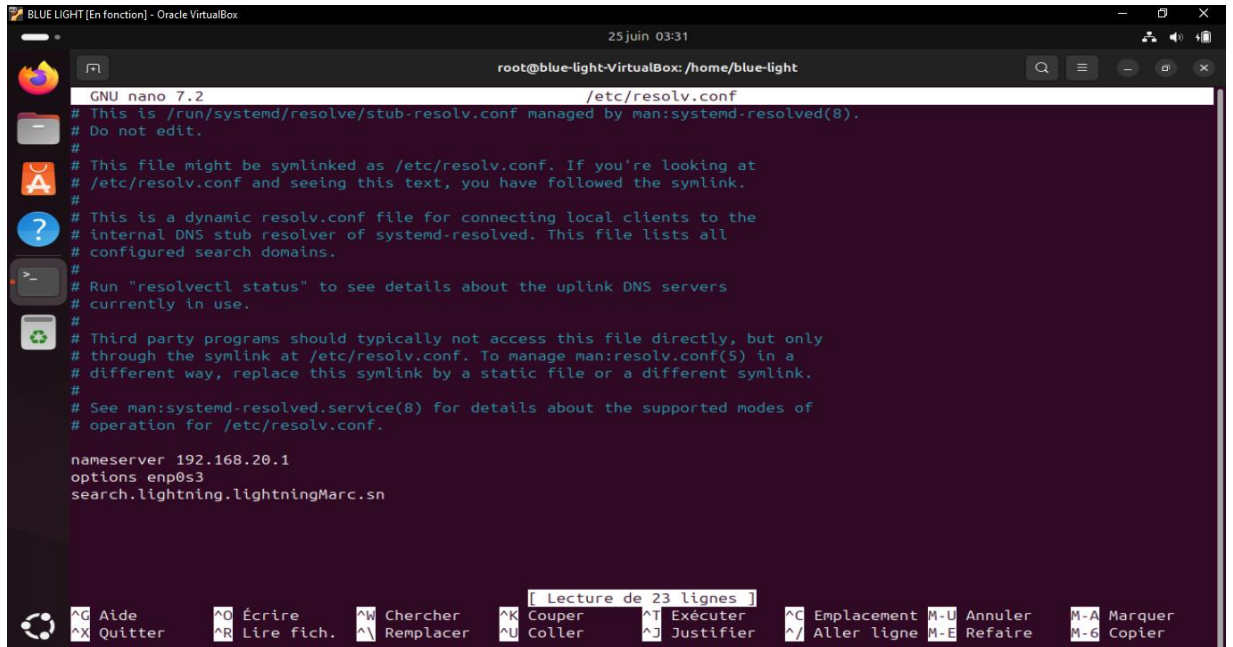
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 501 bytes 46289 (46.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 501 bytes 46289 (46.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

On passe à l'étape suivante. Avec le fichier `resolve.conf` qui se trouve dans le dossier « etc »

```
root@blue-light-VirtualBox:/home/blue-light# nano /etc/resolve.conf
```

Où on va pouvoir l'ouvrir le consulter pour connaître le nom attribuer au PC et même le modifier si voulue.



```

GNU nano 7.2 /etc/resolve.conf
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.20.1
options enps3
search lightning.lightningMarc.sn
  
```

Maintenant on vas aller sur le fichier `named.conf.local` dans le meme dossier mais sous un autre sous dossier.

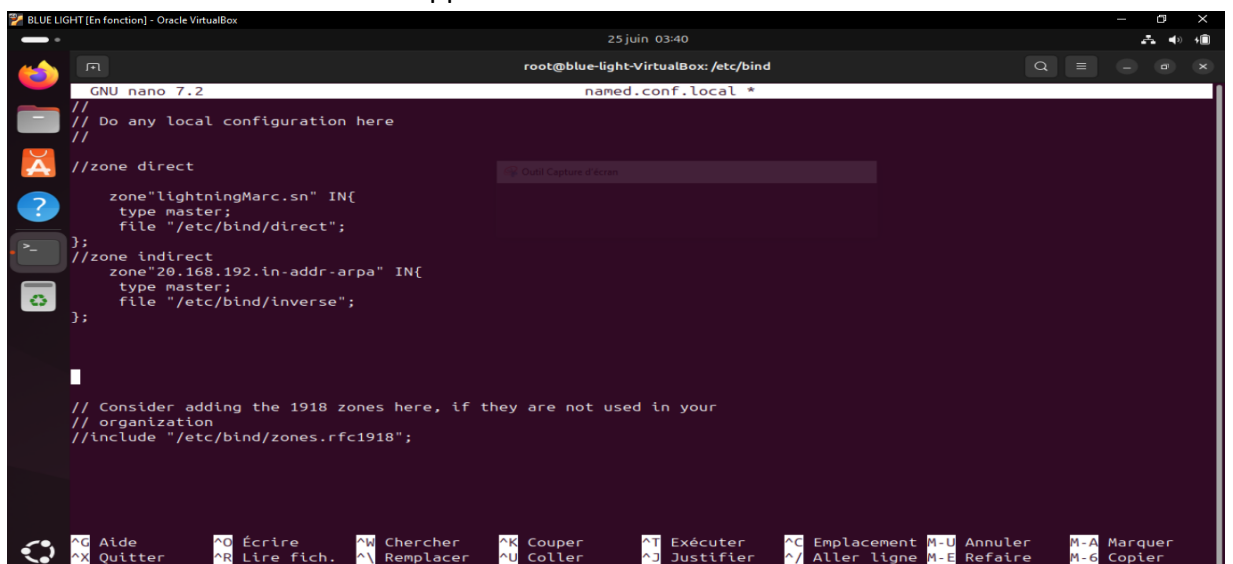
```
root@blue-light-VirtualBox:/home/blue-light# cd /etc/bind
```

```

root@blue-light-VirtualBox:/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@blue-light-VirtualBox:/etc/bind# nano named.conf.local
```

Et là on va accéder au fichier de configuration où on définit notre adresse est on le personnalise avec deux méthodes, la méthode direct et la méthode indirect ou inverse comme on va souvent l'appeler ici.



```

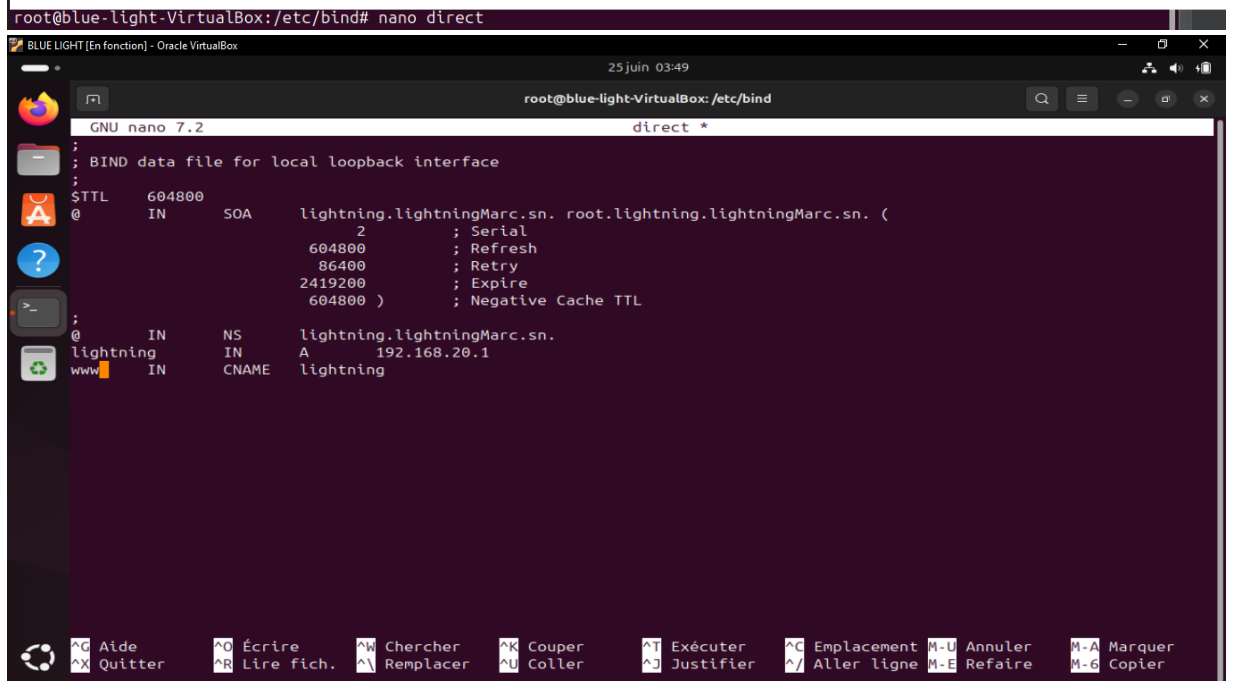
GNU nano 7.2 named.conf.local *
//
// Do any local configuration here
//
//zone direct
//
zone "lightningMarc.sn" IN {
    type master;
    file "/etc/bind/direct";
};
//zone indirect
zone "20.168.192.in-addr.arpa" IN {
    type master;
    file "/etc/bind/inverse";
};

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

Et maintenant on va copier le contenu de base du fichier db.local pour l'insérer dans le fichier « direct » qui au passa nous sert de lien de connexion direct

```
root@blue-light-VirtualBox:/etc/bind# cp db.local direct
```

Et la on ouvre le fichier direct au passage pour y apporter quelque modification pour lui mettre les bonnes valeurs des paramètres.



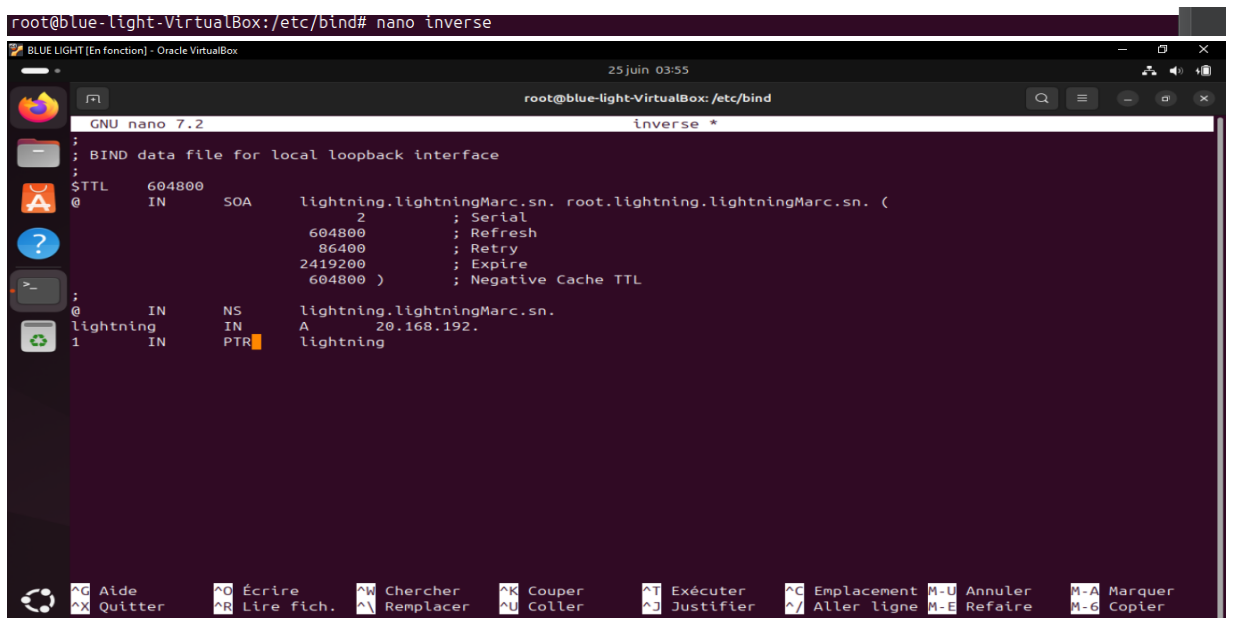
```

root@blue-light-VirtualBox:/etc/bind# nano direct
GNU nano 7.2 direct *
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA      lightning.lightningMarc.sn. root.lightning.lightningMarc.sn. (
                                2      ; Serial
                                604800  ; Refresh
                                86400   ; Retry
                                2419200 ; Expire
                                604800  ) ; Negative Cache TTL
;
@      IN      NS       lightning.lightningMarc.sn.
lightning  IN      A      192.168.20.1
www       IN      CNAME   lightning
  
```

Maintenant encore on va faire une copie du fichier direct vers le fichier inverse. Lui maintenant gère les connexions avec un lien inverser

```
root@blue-light-VirtualBox:/etc/bind# cp direct inverse
```

Et là on ouvre le fichier pour y apporter quelque modification. Parce que il ne peut pas être à l'identique que le lien direct vous vous en douter. On va modifier l'adresse en chiffre et l'écrire inversement comme pour modèle sur le fichier named.conf.local mais le mieux est de le voir en image.



```

root@blue-light-VirtualBox:/etc/bind# nano inverse
GNU nano 7.2 inverse *
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA      lightning.lightningMarc.sn. root.lightning.lightningMarc.sn. (
                                2      ; Serial
                                604800  ; Refresh
                                86400   ; Retry
                                2419200 ; Expire
                                604800  ) ; Negative Cache TTL
;
@      IN      NS       lightning.lightningMarc.sn.
lightning  IN      A      20.168.192.
1          IN      PTR    lightning
  
```

Voilà déjà comme ça c'est okay et on va juste sauvegarder. Et checker la suite.

f. Démarrage et statuts du serveur :

Maintenant avant de tout démarrer on vas faire un p'tit check histoire d'être sûre que tout marche.

```
root@blue-light-VirtualBox:/etc/bind# named-checkconf -z
zone lightningMarc.sn/IN: loaded serial 2
dns_rdata_fromtext: /etc/bind/inverse:13: near '20.168.192.': bad dotted quad
zone 20.168.192.in-addr-arpa/IN: loading from master file /etc/bind/inverse failed: bad dotted quad
zone 20.168.192.in-addr-arpa/IN: not loaded due to errors.
```

Bon déjà là encore okay et on va juste redémarrer et vérifier le statu du serveur histoire qu'il exécute bien tous les config modifier.

```
root@blue-light-VirtualBox:/etc/bind# systemctl restart bind9
root@blue-light-VirtualBox:/etc/bind# systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-06-25 03:58:20 WAT; 15s ago
     Docs: man:named(8)
    Main PID: 4542 (named)
      Status: "running"
        Tasks: 4 (limit: 3475)
       Memory: 5.3M (peak: 5.5M)
          CPU: 71ms
      CGroup: /system.slice/named.service
              └─4542 /usr/sbin/named -f -u bind

juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './DNSKEY/IN': 2001:7fd::1#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './NS/IN': 2001:7fd::1#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './DNSKEY/IN': 2801:1b8:10::b#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './NS/IN': 2801:1b8:10::b#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './DNSKEY/IN': 2001:500:2::c#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './NS/IN': 2001:500:2::c#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './DNSKEY/IN': 2001:503:c27::2:30#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: network unreachable resolving './DNSKEY/IN': 2001:500:1::53#53
juin 25 03:58:20 blue-light-VirtualBox named[4542]: managed-keys-zone: Key 20326 for zone . is now trusted (acceptance
juin 25 03:58:20 blue-light-VirtualBox named[4542]: managed-keys-zone: Key 38696 for zone . is now trusted (acceptance

root@blue-light-VirtualBox:/etc/bind#
```

Voilà tout marche normal. Le service est activer et valide donc cette partie est Finito.

III. Le serveur DHCP

1. Qu'est-ce que le DHCP ? (Définition complète et illustrée)

Le **DHCP**, pour **Dynamic Host Configuration Protocol**, est un protocole réseau qui permet d'attribuer automatiquement une **adresse IP** et d'autres paramètres réseau (comme la passerelle, le DNS, le masque de sous-réseau...) à un appareil qui se connecte à un réseau.

En d'autres termes, il agit comme un **distributeur automatique d'adresses IP**. Sans DHCP, chaque machine devrait être **configurée manuellement**, ce qui devient vite ingérable dès que le nombre de postes dépasse une poignée.

2. Contexte et historique

Avant l'invention du DHCP, les administrateurs réseaux utilisaient un système appelé **BOOTP** (Bootstrap Protocol), créé dans les années 1980. BOOTP permettait une configuration automatique, mais manquait de flexibilité (il nécessitait une configuration statique sur le serveur pour chaque client).

Le **DHCP** a été conçu au début des années 1990 comme une **évolution de BOOTP**, capable de gérer dynamiquement l'allocation d'adresses IP, sans nécessiter de liste prédéfinie de tous les clients. Ce fut une **révolution pour les réseaux d'entreprise et les réseaux domestiques**.

3. Anecdote : Et si on n'avait pas de DHCP ?

Imagine un lycée avec 500 ordinateurs, et un seul technicien réseau. Sans DHCP, ce technicien devrait :

- Configurer **manuellement** chaque machine avec une IP unique,
- Gérer les conflits si deux machines ont la même IP,
- Mettre à jour les IP à chaque changement de machine, redémarrage, ou changement de salle.

Autant dire que ce serait un **cauchemar logistique**. Grâce au DHCP, tout se fait **automatiquement** dès qu'un appareil se connecte.

4. Fonctionnement du DHCP

Le DHCP repose sur un échange en **quatre étapes**, souvent appelé le **DORA** :

- **Discover** : Le client envoie un message pour chercher un serveur DHCP.
- **Offer** : Le serveur répond avec une proposition d'adresse IP.
- **Request** : Le client accepte l'offre en faisant une demande officielle.
- **Acknowledge** : Le serveur confirme l'attribution de l'adresse.

Tout cela se fait en quelques millisecondes dès que l'appareil est connecté.

5. Et en réseau local ?

Dans un réseau privé (maison, entreprise, université), le serveur DHCP joue un rôle crucial :

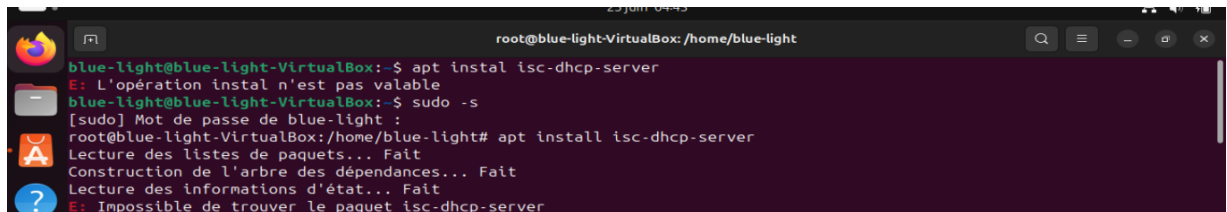
- Il garantit une **distribution rapide** des IP,
- Évite les **conflits d'adresse IP**,
- Permet une **gestion centralisée** (on peut voir qui a reçu quelle IP, quand, pour combien de temps),
- Et peut même **réserver certaines IP** pour des appareils critiques (imprimantes, serveurs, caméras...).

Dans notre projet, nous allons configurer un serveur DHCP local sous Linux, à l'aide du paquet **ISC DHCP Server**, très répandu dans le monde Unix/Linux.

6. Image du Projet

a) Installation de isc-dhcp-server :

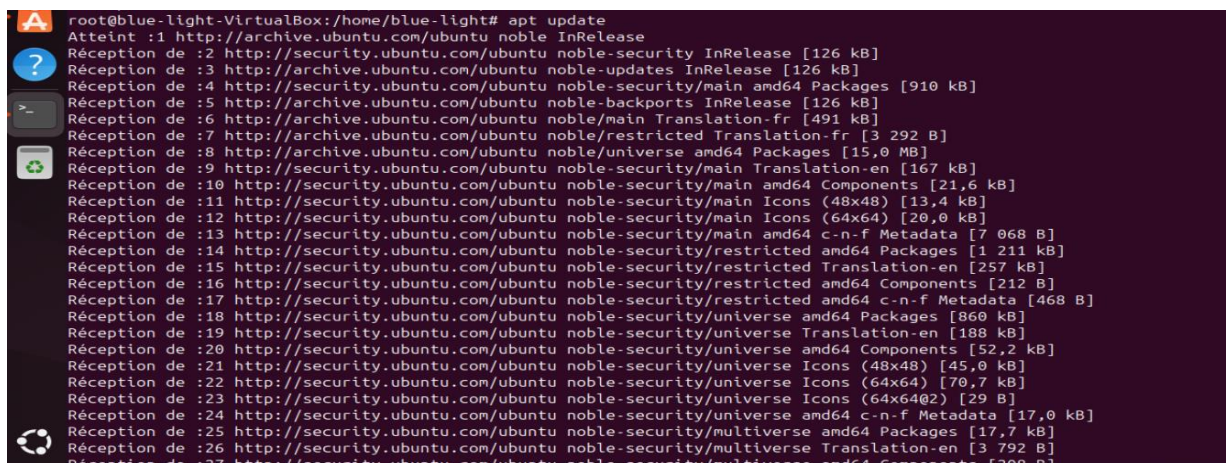
Déjà on commence très court au début de l'installation un bug grave. Et très perturbant. Et ce justement lors de l'installation de la ressource essentielle



```

root@blue-light-VirtualBox:/home/blue-light
blue-light@blue-light-VirtualBox:~$ apt instal isc-dhcp-server
E: L'opération instal n'est pas valable
blue-light@blue-light-VirtualBox:~$ sudo -s
[sudo] Mot de passe de blue-light :
root@blue-light-VirtualBox:/home/blue-light# apt install isc-dhcp-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
E: Impossible de trouver le paquet isc-dhcp-server
  
```

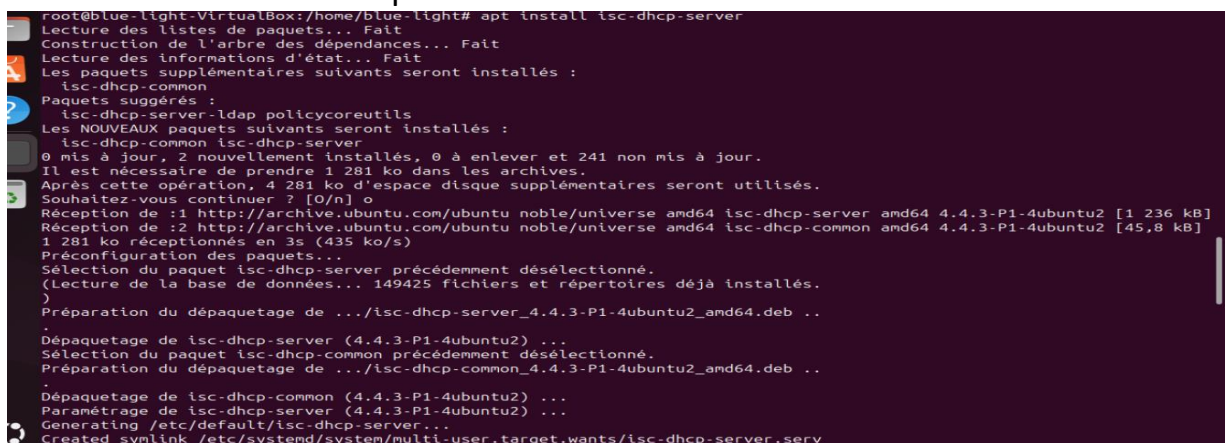
Problème très simple à régler il faut mettre à jour les ressources apt. En tout cas ça a marché chez moi.



```

root@blue-light-VirtualBox:/home/blue-light# apt update
Atteint :1 http://archive.ubuntu.com/ubuntu noble InRelease
Réception de :2 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Réception de :3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Réception de :4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [910 kB]
Réception de :5 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Réception de :6 http://archive.ubuntu.com/ubuntu noble/main Translation-fr [491 kB]
Réception de :7 http://archive.ubuntu.com/ubuntu noble/restricted Translation-fr [3 292 B]
Réception de :8 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15,0 MB]
Réception de :9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [167 kB]
Réception de :10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21,6 kB]
Réception de :11 http://security.ubuntu.com/ubuntu noble-security/main Icons (48x48) [13,4 kB]
Réception de :12 http://security.ubuntu.com/ubuntu noble-security/main Icons (64x64) [20,0 kB]
Réception de :13 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [7 068 B]
Réception de :14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [1 211 kB]
Réception de :15 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [257 kB]
Réception de :16 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Réception de :17 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [468 B]
Réception de :18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [860 kB]
Réception de :19 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [188 kB]
Réception de :20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52,2 kB]
Réception de :21 http://security.ubuntu.com/ubuntu noble-security/universe Icons (48x48) [45,0 kB]
Réception de :22 http://security.ubuntu.com/ubuntu noble-security/universe Icons (64x64) [70,7 kB]
Réception de :23 http://security.ubuntu.com/ubuntu noble-security/universe Icons (64x64@2) [29 B]
Réception de :24 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [17,0 kB]
Réception de :25 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [17,7 kB]
Réception de :26 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [3 792 B]
Réception de :27 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
  
```

Et maintenant on peut installer la ressource



```

root@blue-light-VirtualBox:/home/blue-light# apt install isc-dhcp-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  isc-dhcp-common
Paquets suggérés :
  isc-dhcp-server-ldap polycoreutils
Les NOUVEAUX paquets suivants seront installés :
  isc-dhcp-common isc-dhcp-server
0 mis à jour, 2 nouvellement installés, 0 à enlever et 241 non mis à jour.
Il est nécessaire de prendre 1 281 ko dans les archives.
Après cette opération, 4 281 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://archive.ubuntu.com/ubuntu noble/universe amd64 isc-dhcp-server amd64 4.4.3-P1-4ubuntu2 [1 236 kB]
Réception de :2 http://archive.ubuntu.com/ubuntu noble/universe amd64 isc-dhcp-common amd64 4.4.3-P1-4ubuntu2 [45,8 kB]
1 281 ko réceptionnés en 3s (435 ko/s)
Préconfiguration des paquets...
Sélection du paquet isc-dhcp-server précédemment désélectionné.
(Lecture de la base de données... 149425 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../isc-dhcp-server_4.4.3-P1-4ubuntu2_amd64.deb ...
Dépaquetage de isc-dhcp-server (4.4.3-P1-4ubuntu2) ...
Sélection du paquet isc-dhcp-common précédemment désélectionné.
Préparation du dépaquetage de .../isc-dhcp-common_4.4.3-P1-4ubuntu2_amd64.deb ...
Dépaquetage de isc-dhcp-common (4.4.3-P1-4ubuntu2) ...
Paramétrage de isc-dhcp-server (4.4.3-P1-4ubuntu2) ...
Generating /etc/default/isc-dhcp-server...
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server.serv
  
```

b) Configuration du serveur dhcp :

Bein maintenant on va passer a une autre etape on ouvre le fichier isc-dhcp-server qui se trouve sur le dossier « etc » sous dossier « default »

```
root@blue-light-VirtualBox:/home/blue-light# nano /etc/default/isc-dhcp-server
```

Et on l'ouvre pour configurer selon la démarche suivante.

```

GNU nano 7.2 /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
  
```

Et aussi on va ouvrir le dernier fichier de configuration en ce qui nous concerne. Le fichier dhcpd.conf du dossier « dhcpd » qui est un sous dossier d' « etc »

```
root@blue-light-VirtualBox:/home/blue-light# nano /etc/dhcp/dhcpd.conf
```

Et on l'ouvre pour le configurer

```

GNU nano 7.2 /etc/dhcp/dhcpd.conf *
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
subnet 192.168.20.0 netmask 255.255.255.0 {
  range 192.168.20.10 192.168.20.100;
  option domain-name "lightningMarc.sn";
  option domain-name-servers 192.168.20.1, lightning.lightningMarc.sn;

  default-lease-time 86400;
  max-lease-time 172800;
}

# The ddns-update-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
# have support for DDNS.)
ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement  ^M-U Annuler  ^M-A Marquer
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne ^M-E Refaire   ^M-6 Copier
  
```


Maintenant on va checker la configuration en « ifconfig » pour s'assurer que l'adresse est toujours la même et que le dhcp est configuré avec de nouvelle valeur.

```
root@blue-light-VirtualBox:/home/blue-light# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.20.1 netmask 255.255.255.0 broadcast 192.168.20.255
    inet6 fe80::a00:27ff:fe99:f9d9 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:99:f9:d9 txqueuelen 1000 (Ethernet)
    RX packets 320559 bytes 479643292 (479.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 115459 bytes 9974272 (9.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 712 bytes 69148 (69.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 712 bytes 69148 (69.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

c) Phase test (CHECK) :

Et maintenant on check les signaux de notre serveur pour voir s'il y a des erreurs.

```
root@blue-light-VirtualBox:/home/blue-light# dhcpd -t
Internet Systems Consortium DHCP Server 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
```

Donc là aucun problème de tout peut avancer et on passe au suivant.

Et finalement on va redémarrer le serveur et checker le statut pour voir le résultat final que tout ça donne

```
root@blue-light-VirtualBox:/home/blue-light# systemctl restart isc-dhcp-server
Warning: The unit file, source configuration file or drop-ins of isc-dhcp-server.service changed on disk. Run 'systemct
daemon-reload' to reload units.
root@blue-light-VirtualBox:/home/blue-light# systemctl status isc-dhcp-server
Unknown command verb 'status', did you mean 'status'?
root@blue-light-VirtualBox:/home/blue-light# systemctl status isc-dhcp-server
Warning: The unit file, source configuration file or drop-ins of isc-dhcp-server.service changed on disk. Run 'systemct
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-06-25 04:40:43 WAT; 31s ago
     Docs: man:dhcpd(8)
    Main PID: 6903 (dhcpd)
      Tasks: 1 (limit: 3475)
    Memory: 3.7M (peak: 4.0M)
       CPU: 24ms
    CGroup: /system.slice/isc-dhcp-server.service
            └─6903 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf enp0s3

juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Database file: /var/lib/dhcp/dhcpd.leases
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: PID file: /run/dhcp-server/dhcpd.pid
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Wrote 0 leases to leases file.
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Listening on LPF/enp0s3/08:00:27:99:f9:d9/192.168.20.0/24
juin 25 04:40:43 blue-light-VirtualBox sh[6903]: Listening on LPF/enp0s3/08:00:27:99:f9:d9/192.168.20.0/24
juin 25 04:40:43 blue-light-VirtualBox sh[6903]: Sending on LPF/enp0s3/08:00:27:99:f9:d9/192.168.20.0/24
juin 25 04:40:43 blue-light-VirtualBox sh[6903]: Sending on Socket/fallback/fallback-net
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Sending on LPF/enp0s3/08:00:27:99:f9:d9/192.168.20.0/24
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Sending on Socket/fallback/fallback-net
juin 25 04:40:43 blue-light-VirtualBox dhcpd[6903]: Server starting service.
lines 1-22/22 (END)
```

Voilà tout est ok et tout va pour le mieux donc on peut

IV. Conclusion Générale

La mise en œuvre d'un **serveur DNS et DHCP sous Linux** représente une étape clé dans la construction d'une infrastructure réseau locale **fiable, automatisée et facile à administrer**.

D'une part, le **serveur DHCP** simplifie considérablement la gestion des adresses IP, en attribuant dynamiquement les paramètres réseau aux machines dès leur connexion. Il élimine les erreurs humaines, prévient les conflits d'adresse, et optimise l'utilisation des ressources réseau.

D'autre part, le **serveur DNS** permet une communication fluide entre les hôtes en traduisant les noms en adresses IP, rendant le réseau plus lisible, plus convivial et surtout plus facile à administrer à grande échelle. Il devient même un **point central de l'architecture logique** du réseau.

1. Bilan pédagogique

Ce projet a permis de :

- Maîtriser l'installation et la configuration de services critiques (BIND9 et DHCP),
- Comprendre l'articulation entre les services IP et les services de nommage,
- Manipuler des fichiers de configuration, des zones DNS, des baux DHCP et des interfaces réseau,
- Apprendre à tester, diagnostiquer et corriger les erreurs courantes dans un contexte réel.

2. Perspectives

Cette base de travail ouvre la voie à des projets plus complexes, comme :

- L'intégration d'un **contrôleur de domaine** (Active Directory Samba),
- La mise en place d'un **DNS secondaire** pour la redondance,
- L'automatisation de la configuration via **Ansible ou Bash**,
- Ou encore la gestion dynamique de DNS par le DHCP via les **DDNS (Dynamic DNS Updates)**.

V. BIBLIOGRAPHIE / WEBOGRAPHIE :

1. VIDEO YOUTUBE

- **TUTO INSTALLATION ET CONFIGURATION DU SERVEUR DNS SOUS UBUNTU 22.04 LTS**

De : MAREME TECHN. INFO

URL: <https://youtu.be/BTF7m3ClcyM?si=sC2UhVel09UTtLmv>

Date: 25/06/2025, 00:20

- **Installation et configuration d'un serveur DHCP sous Linux (Ubuntu 20.04)**

De : [hbs_servtech](#)

URL: https://youtu.be/yEgbdNPp2j8?si=e4QNnxXJGQ_yPEKZ

Date: 25/06/2025, 03:15

2. OUVRAGE UTILISER

- **SYSTÈME D'EXPLOITATION**
Par le Prof : KASENGEDIA MOTUMBE Pierre.
Date : 24/06/2025,

3. SITE DE DOCUMENTATION

- **CHATGPT de OpenIA** : en cas d'erreur ou de problème dans la procédure.
Date : 25/06/2025, 03 :40

VI. TABLE DES MATIERES

- 1. Introduction Générale**
 - 1.1 Contexte du projet
 - 1.2 Objectifs techniques
 - 1.3 Objectif pédagogique
- 2. Le Serveur DNS**
 - 2.1 Définition et fonctionnement du DNS
 - 2.2 Historique du DNS
 - 2.3 Architecture et résolution DNS
 - 2.4 Mise en place du serveur DNS sous Linux
 - 2.4.1 Installation de BIND9
 - 2.4.2 Configuration des fichiers DNS
 - 2.4.3 Méthode directe et méthode inverse
 - 2.4.4 Vérification et démarrage du serveur DNS
- 3. Le Serveur DHCP**
 - 3.1 Définition et utilité du DHCP
 - 3.2 Historique du DHCP
 - 3.3 Fonctionnement du protocole (DORA)
 - 3.4 Mise en place du serveur DHCP sous Linux
 - 3.4.1 Installation de isc-dhcp-server
 - 3.4.2 Configuration des fichiers DHCP
 - 3.4.3 Test et vérification du serveur DHCP
- 4. Conclusion Générale**
 - 4.1 Bilan technique
 - 4.2 Bilan pédagogique
 - 4.3 Perspectives
- 5. Bibliographie / Webographie**

Rapport rédigé par : MUFUTA MULEBA MARC (MARC LIGHTNING).

Mercis ...