Contexte

Dans le cadre de ton BTS SIO, j'ai eu l'opportunité de déployer un serveur DHCP et un serveur DNS sur des machines virtuelles Linux sans interface graphique.

Difficultés rencontrées

Lors de la mise en place d'un serveur DHCP/DNS sous Linux dans le cadre de mon BTS SIO, j'ai été confronté à plusieurs difficultés, principalement liées à la syntaxe des fichiers de configuration. Contrairement aux environnements graphiques ou à Windows, Linux repose fortement sur l'édition manuelle de fichiers texte, ce qui rend la configuration très sensible à la moindre erreur de syntaxe (espaces, tabulations, ponctuation, noms de fichiers, etc.).

Environnement technologique

1. Système d'exploitation

Debian 12 (Linux) : Le serveur DHCP et DNS a été installé sur une machine virtuelle tournant sous Debian 12, une distribution Linux stable et largement utilisée en entreprise pour les services réseau.

2. Outils et langages utilisés

Shell (Bash) : L'installation et la configuration ont été réalisées en ligne de commande à travers le shell Bash, nécessitant une bonne rigueur dans la syntaxe et les commandes Linux.

Fichiers de configuration :

- /etc/dhcp/dhcpd.conf : pour la configuration du serveur DHCP.
- /etc/bind/named.conf.* : pour la configuration du serveur DNS (BIND).

Services système :

- isc-dhcp-server pour la gestion du DHCP.
- bind9 pour le service DNS.

3. Infrastructure réseau

Réseau local virtuel (LAN) : Le serveur était intégré dans un réseau local, avec des clients test pour vérifier l'attribution automatique d'adresses IP et la résolution DNS.

Protocole DHCP (Dynamic Host Configuration Protocol) : utilisé pour attribuer automatiquement des adresses IP aux machines clientes.
Protocole DNS (Domain Name System) : utilisé pour résoudre les noms de domaine en adresses IP dans le réseau local.

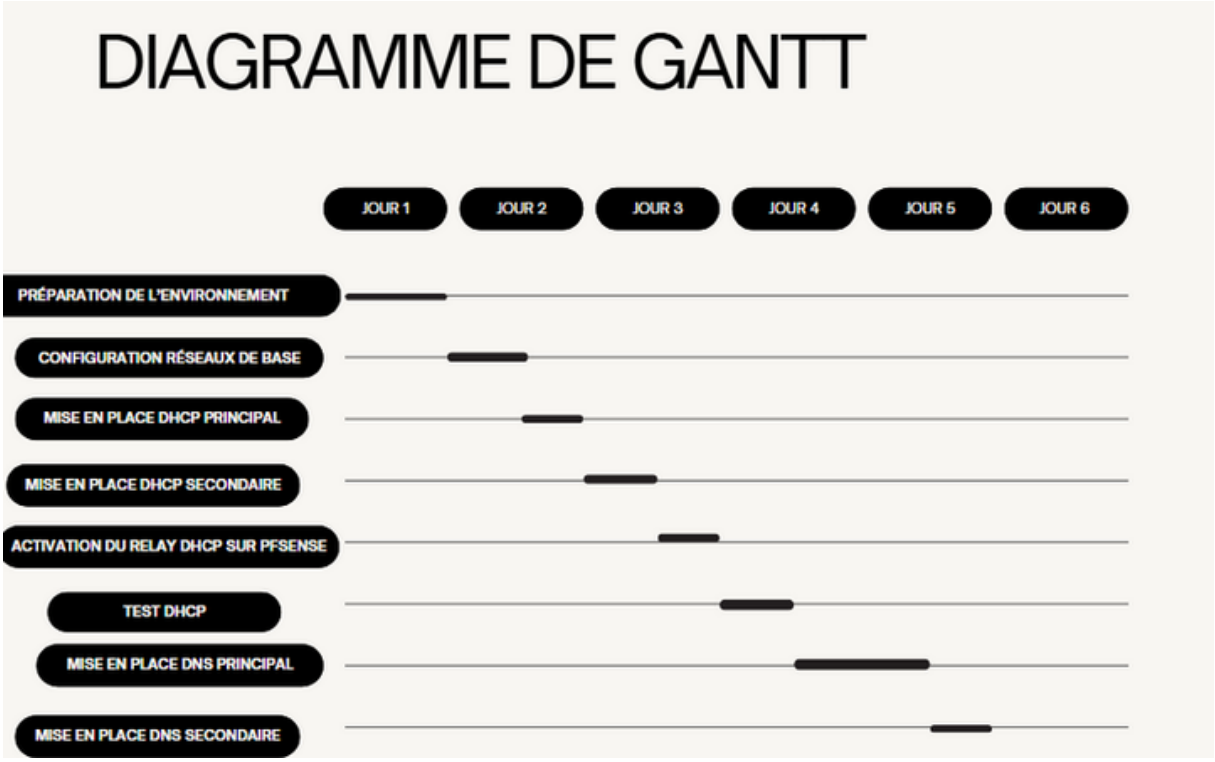
Tâches réalisées

- Mise en place du serveur DHCP
- Mise en place du serveur DNS

Compétences mobilisées

- **Gérer le patrimoine informatique** (Installation et configuration des services DHCP et DNS sous Linux)
- **Répondre aux incidents et aux demandes d'assistance** (Résolution des erreurs liées à la syntaxe dans les fichiers de configuration)
- **Organiser son développement professionnel** (Apprentissage de la configuration réseau sous Linux, recherche de solutions via la veille technique)

Diagramme de GANTT



Bilan personnel

Ce projet m'a permis de renforcer mes compétences en administration système sous Linux et d'avoir une vision plus concrète de la gestion des services réseau essentiels (DHCP et DNS). J'ai appris à diagnostiquer des erreurs de configuration, à utiliser efficacement le terminal Linux, et à comprendre l'importance d'une documentation claire et d'une approche structurée. C'est une expérience qui m'a donné envie d'approfondir mes connaissances sur l'administration de services réseaux open source.

Mise en place serveur DHCP / DNS

Prérequis :

Nous aurons besoin d'un routeur Pfsense, de deux serveurs linux non-graphiques et d'un client Debian ou Windows pour les tests.

Ces deux serveurs devront avoir les packages **isc-dhcp-server** ainsi que **bind9** de préinstallés.

Premièrement nous configurons le fichiers **/etc/hosts** de nos deux serveurs :

```
 GNU nano 7.2                                                    /etc/hosts
              localhost
              deb-srv1.                       deb-srv1

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

```
 GNU nano 7.2                                                    /etc/hosts
              localhost
              deb-srv2.                       deb-srv2

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

Puis le fichier **/etc/resolv.conf** de notre client Debian :

```
 GNU nano 7.2                    /etc/resolv.conf *
# Generated by NetworkManager
domain [redacted]
search [redacted]
nameserver [redacted]
nameserver [redacted]
```

Configuration du service DHCP en failover :

Voici la configuration du serveur principal dhcp via le fichier de configuration **/etc/dhcp/dhpd.conf**

Nous y indiquons quel est le serveur principal et quel est le secondaire, puis nous lui donnons une plage d'adresses à distribuer, enfin nous faisons une réservation d'adresse pour notre client.

```
  GNU nano 7.2                                                  /etc/dhcp/dhcpd.conf *
option domain-name "caloone.local";
option domain-name-servers

default-lease-time 21966;
max-lease-time 42000;

ddns-update-style interim;

authoritative;

log-facility local7;

subnet               netmask 255.255.255.0 {
}

failover peer "lancli-failover" {
        primary;
        address             .;
        port 647;

        peer address              ;
        peer port 647;

        max-response-delay 60;
        max-unacked-updates 10;
        load balance max seconds 3;
        mclt 3600;
        split 128;
}

subnet            netmask 255.255.255.0 {
        option routers          ;
        pool {
                failover peer "lancli-failover";
                range                        ;
                }
}
```

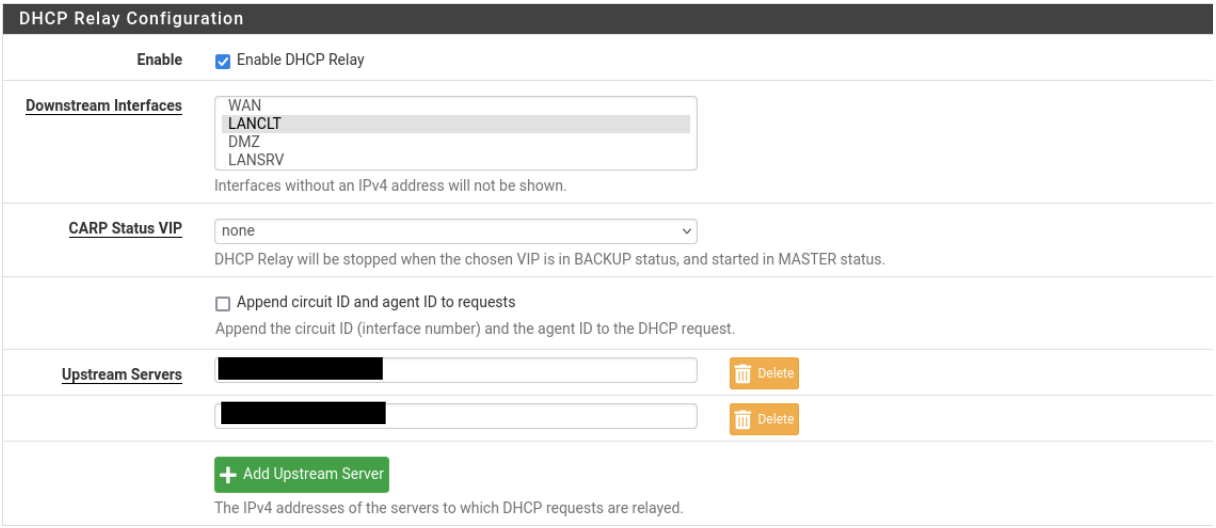
Voici la configuration du serveur dhcp secondaire, via le fichier **/etc/dhcp/dhcpd.conf**

```
  GNU nano 7.2                                                    /etc/dhcp/dhcpd.conf
option domain-name "caloone.local";
option domain-name-servers               ,               ;

default-lease-time 21966;
max-lease-time 42000;

ddns-update-style interim;

#authoritative;

log-facility local7;


subnet               netmask 255.255.255.0 {
}

failover peer "lancli-failover" {
        secondary;
        address 192.168.100.2;
        port 647;

        peer address              ;
        peer port 647;

        max-response-delay 60;
        max-unacked-updates 10;
        load balance max seconds 3;
}


subnet            netmask 255.255.255.0 {
        option routers          ;
        pool {
                failover peer "lancli-failover";
                range ..........  ..........;
        }

}
```

Une fois le fichier dhcpd.conf configurée, il est nécessaire de configurer la ou les interfaces sur lesquelles le service doit être à l'écoute, via le fichier **/etc/default/isc-dhcp-server**, sur les deux serveurs dhcp :

```
  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="ens33"
INTERFACESv6=""
```

Enfin, nous devons activer le relay DHCP sur notre router Pfsense :



Vérifions que notre client possède la bonne adresse ip, en faisant un release renew à la linux :

```
root@deb-cyber:~# dhclient -r
root@deb-cyber:~# dhclient
root@deb-cyber:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
up default qlen 1000
    link/ether 00:0c:29:8a:8f:e1 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet [redacted]/24 brd[redacted] scope global dynamic ens33
       valid_lft 1789sec preferred_lft 1789sec
    inet6 fe80::20c:29ff:fe8a:8fe1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Configuration du service DNS en failover :

Nous allons maintenant configurer le service DNS tel que le srv-deb2 soit le serveur DNS principal.

Premièrement, nous allons créer une redirection non-conditionnelle afin de rediriger de façon permanente le trafic d'un nom de domaine vers un autre grâce au fichier : **/etc/bind/named.conf.options**

```
  GNU nano 7.2                                                  /etc/bind/named.conf.options
options {
        directory "/var/cache/bind";

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

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

        // forwarders {
        //      1.1.1.1;
        // };

        //========================================================================
        // 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 no;
        allow-query { any; };
        listen-on-v6 { any; };
};
```

Voici la configuration du fichier **/etc/bind/named.conf.local** du serveur principal. Ce fichier permet d'indiquer quelles sont nos zones directes et inverses, et d'autoriser le transfert de ces zones à notre serveur secondaire :

```
  GNU nano 7.2                                                    /etc/bind/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";

zone "caloone.local" {
        type master;
        file "db.caloone.local";
        allow-transfer {             ; };
        allow-query { any; };
        notify yes;
};

zone "          .in-addr.arpa" {
        type master;
        file "db.inv.          ";
        allow-transfer {              ; };
        allow-query { any; };
        notify yes;
};

zone "      .in-addr.arpa" {
        type master;
        file "db.inv.      ";
        allow-transfer {              ; };
        allow-query { any; };
        notify yes;
};
```

Nous allons maintenant créer et configurer les fichiers de configuration de nos zones. Pour cela nous avons à disposition des exemples de ces fichiers dans le dossier **/etc/bind**, à copier dans le dossier **/var/cache/bind/** afin de récupérer les en-têtes SOA et NS :

```
root@deb-srv2:~# cd /etc/bind
root@deb-srv2:/etc/bind# ls
bind.keys  db.0  db.127  db.255  db.empty  db.local  named.conf  named.conf.default-zones  named.conf.local  named.conf.
root@deb-srv2:/etc/bind# cat db.local
;
; 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
root@deb-srv2:/etc/bind# cp db.local /var/cache/bind/db.caloone.local_
```

De même pour nos zones inverses :

```
root@deb-srv2:/etc/bind# cp db.local /var/cache/bind/db.inv.100.168.192
```

```
root@deb-srv2:/etc/bind# cp db.local /var/cache/bind/db.inv.20.172
```

Configurons notre fichier de zone directe **/var/cache/bind/db.caloone.local** :

```
  GNU nano 7.2                                                    /var/cache/bind/db.caloone.local
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     deb-srv2.             . root.deb-srv2.             . (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      deb-srv2.             .
@       IN      A
@       IN      NS      deb-srv1.             .
@       IN      A
@       IN      AAAA    ::1
deb-srv2        IN      A
deb-srv1        IN      A
WAN     IN      A
LANCLT  IN      A
LANSRV  IN      A
DMZ     IN      A
```

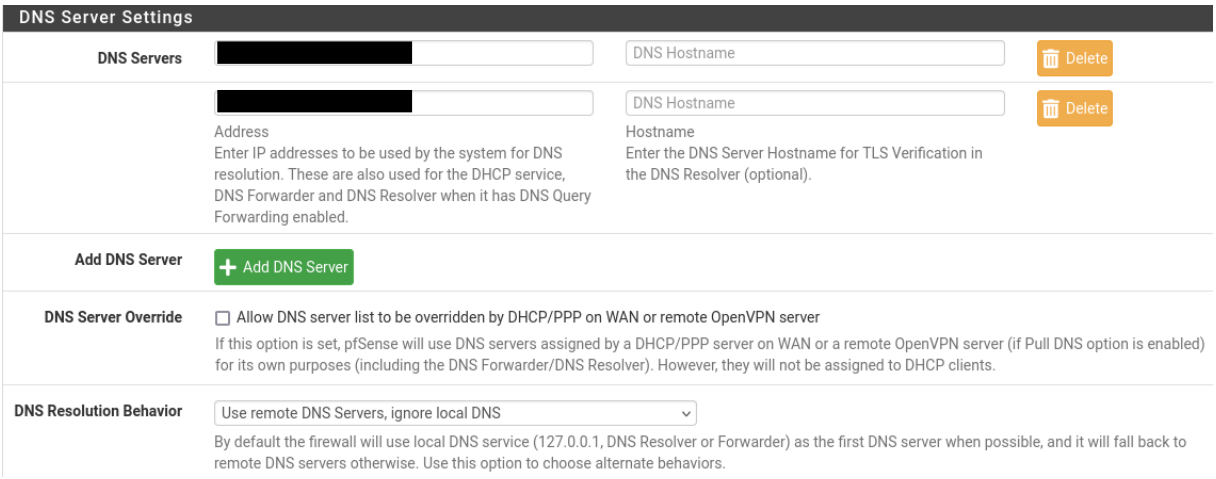
Chargeons les changements de configurations grâce à la commande : **rndc reload** puis testons nos fichiers grâce aux commandes : **named-checkconf -z** et **named-checkzone [redacted] /var/cache/bind/db.[redacted]** :

```
root@deb-srv2:~# rndc reload
server reload successful
root@deb-srv2:~# named-checkconf -z
zone caloone.local/IN: loaded serial 2
zone 100.168.192.in-addr.arpa/IN: loaded serial 2
zone 20.172.in-addr.arpa/IN: loaded serial 2
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
root@deb-srv2:~# named-checkzone              /var/cache/bind/db.
zone              ./IN: loaded serial 2
OK
```

Configurons maintenant notre serveur DNS secondaire srv-deb1 via le fichier **/etc/bind/named.conf.local** :

```
 GNU nano 7.2                                                  /etc/bind/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";

zone "caloone.local" {
        type slave;
        masters {               };;
        file "db.
        allow-query { any; };
};

zone "          .in-addr.arpa" {
        type slave;
        masters {               };;
        file "db.inv.          ";
        allow-query { any; };
};

zone "      .in-addr.arpa" {
        type slave;
        masters {               };;
        file "db.      ";
        allow-query { any; };
};
```

Ne pas oublier de renseigner nos deux serveurs DNS dans les paramètres généraux de notre Pfsense :



La configuration étant terminée, nous pouvons tester la résolution de nos différent appareils réseaux sur notre client Debian grâce à la commande **nslookup** :

PS : Le message de récursion est tout à fait normal, ne pas s'en inquiéter

```
root@DEB-CLT:~# nslookup deb-srv1
;; Got recursion not available from 192.168.100.1, trying next server
Server:         [redacted]
Address:        [redacted]

Name:   deb-srv1.[redacted]
Address: [redacted]
;; Got recursion not available from [redacted], trying next server

root@DEB-CLT:~# nslookup deb-srv2
;; Got recursion not available from 192.168.100.1, trying next server
Server:         [redacted]
Address:        [redacted]

Name:   deb-srv2.[redacted]
Address: [redacted]
;; Got recursion not available from [redacted], trying next server
```