Configurations

L’adressage IP

IPv4

Configuration d’une interface en IPv6
Procédure adressage ipv4 sur routeur : Router> enable Router# configure terminal Router(config)# interface GigabitEthernet0/0 Router(config-if)# ip address 192.168.1.1 255.255.255.0 Router(config-if)# no shutdown Router(config-if)# exit Router(config)# exit Router# show ip interface brief

Cette procédure configure l’adressage IPv4 d’un routeur Cisco sur l’interface GigabitEthernet0/0.
Après être passé en mode privilégié (>enable) puis en mode de configuration globale (#config), l’administrateur sélectionne l’interface et lui attribue une adresse IP (ici 192.168.1.1) avec le masque de sous-réseau 255.255.255.0, ce qui permet au routeur de

communiquer sur le réseau local correspondant. La commande no shutdown active l'interface, qui est désactivée par défaut. On sort ensuite du mode de configuration, et la commande show ip interface brief permet de vérifier l'état des interfaces et de confirmer que l'adresse a bien été appliquée et que l'interface est opérationnelle.

IPv6

Configuration d'une interface en IPv6
Router> enable Router# configure terminal Router(config)# ipv6 unicast-routing Router(config)# interface <nom interface> Router(config-if)# ipv6 address <address ipv6> **SI ajout link local** : Router(config-if)# ipv6 address FE80::<id du link> link-local Router(config-if)# no shutdown Router(config-if)# exit Router# show ipv6 interface brief Router# copy running-config startup-config

Ici, on configure l'adressage IPv6 sur le routeur. Après être passé en mode privilégié puis en mode de configuration globale, on active le routage IPv6 (ipv6 unicast-routing), ce qui permet au routeur de traiter et de transmettre des paquets IPv6. Ensuite, il accède à une interface réseau et lui attribue une adresse IPv6 globale, et éventuellement une adresse link-local (FE80::) pour la communication locale entre voisins. Ici encore, la commande no shutdown active l'interface. La commande show ipv6 interface brief permet de vérifier l'état et les adresses configurées sur les interfaces. Enfin, copy running-config startup-config enregistre la configuration en mémoire de démarrage afin qu'elle soit conservée après un redémarrage du routeur.

Installation/configuration d'OSPF

A Savoir :

Area 0 → Connexion WAN
Area 1 → Connexions LAN (Site principal + site 1)

Adresses de loopback :

Routeur 1	1.1.1.1

Routeur 2	2.2.2.2
Routeur 4	3.3.3.3
Routeur 5	4.4.4.4
Routeur 6	5.5.5.5

Liste des configurations OSPF de chaque routeur :

Routeur 1 :	Router1# configure terminal interface loopback0 ip address 1.1.1.1 255.255.255.255 no shutdown router ospf 1 router-id 1.1.1.1 network 1.1.1.1 0.0.0.0 area 0 network 192.168.10.0 0.0.0.31 area 1 network 192.168.20.0 0.0.0.127 area 1 network 192.168.30.0 0.0.0.15 area 1 network 172.16.100.0 0.0.0.255 area 0 network 10.0.3.0 0.0.0.255 area 0 write
Routeur 2 :	Router2# conf terminale interface Loopback0 ip address 2.2.2.2 255.255.255.255 ip ospf 1 area 0 interface GigabitEthernet0/0/1 ip ospf 1 area 0 router ospf 1 router-id 2.2.2.2 network 2.2.2.2 0.0.0.0 area 0

	network 192.168.11.0 0.0.0.15 area 2 network 192.168.21.0 0.0.0.31 area 2 network 172.16.100.10 0.0.0.3 area 0 write
Routeur 4 :	Router4# configure terminal interface Loopback0 ip address 3.3.3.3 255.255.255.255 ip ospf 2 area 0 interface GigabitEthernet0/0 ip ospf 2 area 0 router ospf 2 router-id 3.3.3.3 network 172.16.100.16 0.0.0.3 area 0 network 172.16.100.4 0.0.0.3 area 0 write
Routeur 5 :	Router1# configure terminal interface loopback0 ip address 4.4.4.4 255.255.255.255 no shutdown interface GigabitEthernet0/0 p address 172.16.100.10 255.255.255.252 ip ospf 2 area 0 router ospf 2 router-id 4.4.4.4 network 4.4.4.4 0.0.0.0 area 0 network 192.168.10.0 0.0.0.31 area 1 network 192.168.20.0 0.0.0.127 area 1 network 172.16.100.0 0.0.0.255 area 0 network 10.0.3.0 0.0.0.255 area 0 write
Routeur 6 :	Router6# configure terminal interface Loopback0

	ip address 5.5.5.5 255.255.255.255 ip ospf 2 area 0 interface GigabitEthernet0/0 ip ospf 2 area 0 router ospf 2 router-id 5.5.5.5 network 172.16.100.18 0.0.0.3 area 0 network 172.16.100.9 0.0.0.3 area 0 write

---

Liste des configurations OSPF IPV6 de chaque routeur :

Routeur 1	Router1(config)# ipv6 unicast-routing ---------- OSPFv3 process ---------- ipv6 router ospf 10 router-id 1.1.1.1 exit ---------- WAN ---------- interface GigabitEthernet0/0/1 ipv6 ospf 10 area 0 no shutdown ---------- VLAN 10 ---------- interface GigabitEthernet0/0/0.10 ipv6 ospf 10 area 1 ---------- VLAN 20 ---------- interface GigabitEthernet0/0/0.20 ipv6 ospf 10 area 1 ---------- Interface vers serveur ---------- interface GigabitEthernet0/0/2

	ipv6 ospf 10 area 1 write
Routeur 2	Router2(config)# ipv6 unicast-routing ---------- OSPFv3 process ---------- ipv6 router ospf 10 router-id 2.2.2.2 exit ---------- WAN ---------- interface GigabitEthernet0/0/1 ipv6 ospf 10 area 0 no shutdown ---------- VLAN 11 ---------- interface GigabitEthernet0/0/0.11 ipv6 ospf 10 area 1 ---------- VLAN 21 ---------- interface GigabitEthernet0/0/0.21 ipv6 ospf 10 area 1 write
Routeur 4	Router4(config)# ipv6 unicast-routing ---------- OSPFv3 process ---------- ipv6 router ospf 14 router-id 3.3.3.3 exit ---------- Routeur 1 ---------- interface GigabitEthernet0/2 ipv6 ospf 14 area 0 no shutdown ---------- Routeur 6 ---------- interface GigabitEthernet0/0 ipv6 ospf 14 area 0 ---------- Routeur 5 ---------- interface GigabitEthernet0/1 ipv6 ospf 14 area 0 write

Routeur 5	Router5(config)# ipv6 unicast-routing ---------- OSPFv3 process ---------- ipv6 router ospf 15 router-id 4.4.4.4 exit ---------- Routeur 2 ---------- interface GigabitEthernet0/2 ipv6 ospf 15 area 0 no shutdown ---------- Routeur 6 ---------- interface GigabitEthernet0/0 ipv6 ospf 15 area 0 ---------- Routeur 4 ---------- interface GigabitEthernet0/1 ipv6 ospf 15 area 0 write
Routeur 6	Router6(config)# ipv6 unicast-routing ---------- OSPFv3 process ---------- ipv6 router ospf 16 router-id 5.5.5.5 exit ---------- Routeur 5 ---------- interface GigabitEthernet0/1 ipv6 ospf 16 area 0 no shutdown ---------- Routeur 4 ---------- interface GigabitEthernet0/2 ipv6 ospf 16 area 0 write

Installation et configuration de DHCP

Dans cette partie nous allons vous présenter comment nous avons configuré DHCPv4 ainsi que DHCPv6.

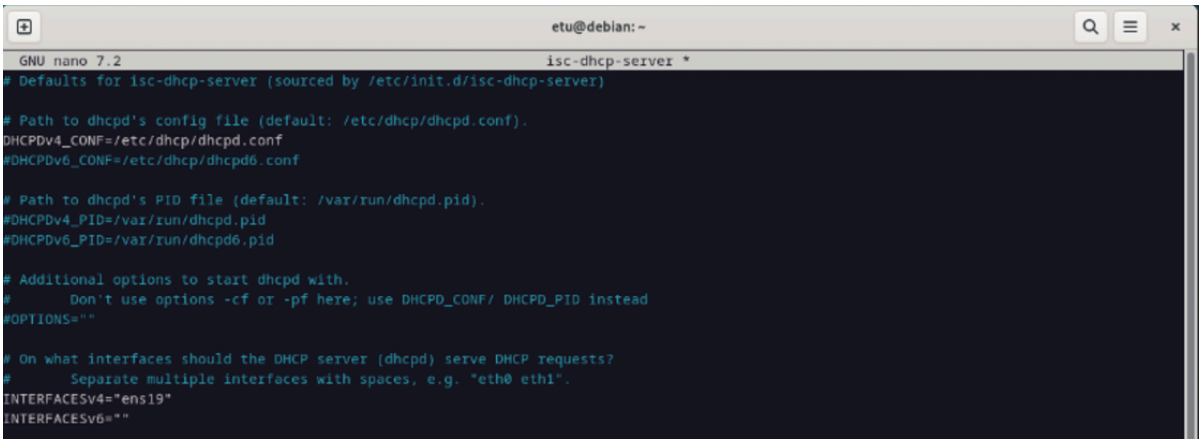
L’objectif de la mise en place des service DHCPv4 et DHCPv6 est de permettre aux hôtes présents dans les différents VLAN de notre réseau d’obtenir automatiquement une adresse IPv4 et IPv6 à partir du serveur DHCP centralisé présent sur notre machine machine réseau dans situé dans le site principale de l’association. Ainsi, à partir de cela chaque poste pourra accéder à des informations complémentaires tel que le DNS, les passerelles, les autres services...etc.

Paquets nécessaires à la mise de ces services :

- isc-dhcp-server : paquet permettant l’installation de serveurs et de relais DHCP ;
- tcpdump : outil de capture de paquets ;
- iptables : gestion des règles de pare-feu ;
- iptables-persistent : conservation des règles de pare-feu ;
- dnsutils : outils réseau pour la résolution de nom, dont dig ;

Mise en place de DHCPv4

Nous avons tout d’abord mis en place le service DHCPv4 pour ce faire nous avons configuré le fichier /etc/default/isc-dhcp-server. Voici ci-dessous le fichier :

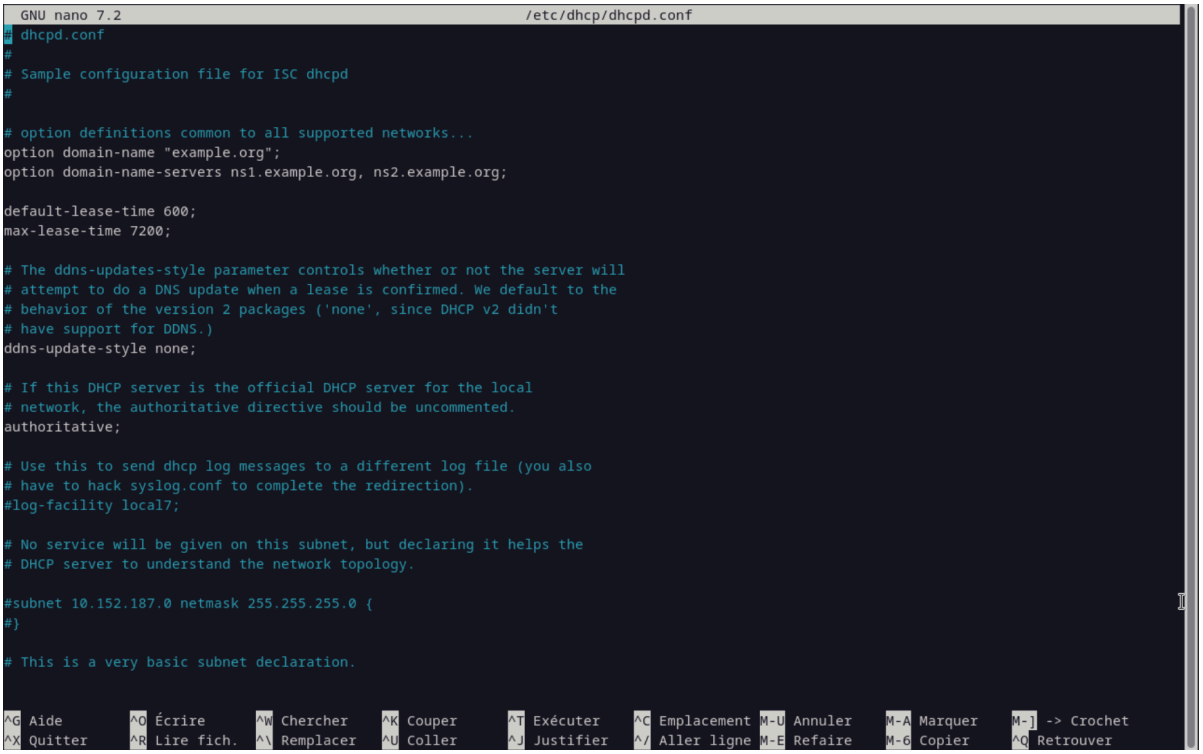


Ce fichier est celui qui configure le daemon en lui-même. Il indique de quelle manière il fonctionne.

`DHCPDv4_CONF=/etc/dhcp/dhcpd.conf`
Permet de spécifier le chemin absolu vers les fichiers de configuration DHCPv4

Ensuite, la ligne :
`INTERFACESv4=”ens19”`
permet d’indiquer quelle interface réseau doit-être utilisé pour le service DHCPv4

Suite à cela nous avons pu indiquer dans le fichier /etc/dhcp/dhcpd.conf le type et la méthode d’attribution des adresses IP sur le réseau :



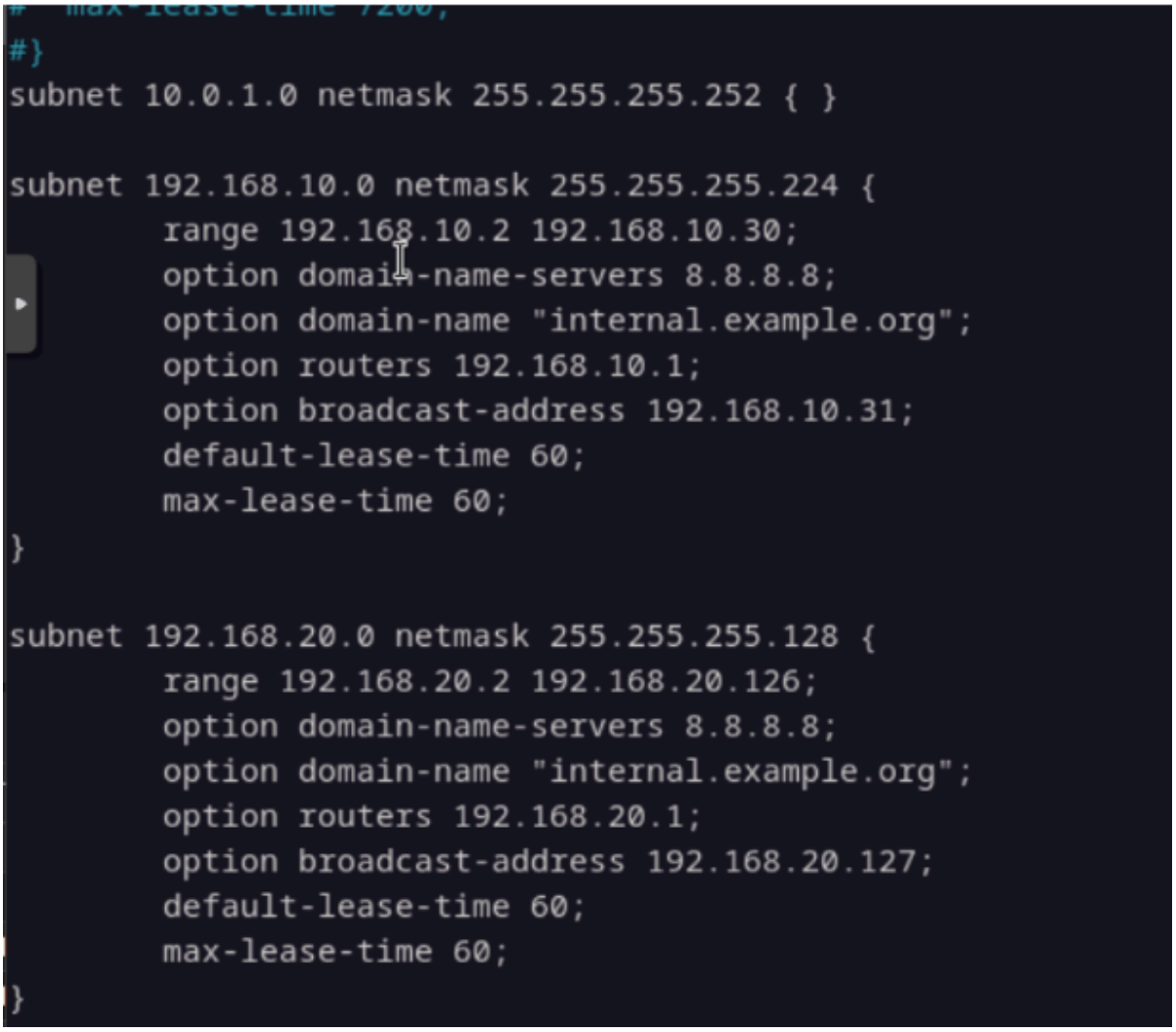
Dans cette partie nous avons pu indiquer que par défaut la durée d'un bail DHCP est de 10 minutes et que la durée maximum d'un baill DHCP sera de 120 minutes.

`ddns-update-style none;` désactive la mise-à-jour dynamique du DNS par le DHCP.

`authoritative;` indique que le serveur installé sur cette machine est le serveur principal du réseau, et que celui-ci fait autorité.

Dans la suite du fichier nous avons pu indiquer de quelle manières seront distribuées les adresses ip sur chaques sites conformément à notre topologie.
Voici la configuration pour chaque VLAN du réseau :

Site Principal, VLAN 10 et VLAN 20 :



```
#}
subnet 10.0.1.0 netmask 255.255.255.252 { }

subnet 192.168.10.0 netmask 255.255.255.224 {
        range 192.168.10.2 192.168.10.30;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.10.1;
        option broadcast-address 192.168.10.31;
        default-lease-time 60;
        max-lease-time 60;
}

subnet 192.168.20.0 netmask 255.255.255.128 {
        range 192.168.20.2 192.168.20.126;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.20.1;
        option broadcast-address 192.168.20.127;
        default-lease-time 60;
        max-lease-time 60;
}
```

Site 1 VLAN 11 et VLAN 21 :

```
#Site1 vlan11
subnet 192.168.11.0 netmask 255.255.255.240 {
        range 192.168.11.2 192.168.11.14;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.11.1;
        option broadcast-address 192.168.11.15;
        default-lease-time 60;
        max-lease-time 60;
}

#Site1 vlan21
subnet 192.168.21.0 netmask 255.255.255.224 {
        range 192.168.21.2 192.168.21.30;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.21.1;
        option broadcast-address 192.168.11.31;
        default-lease-time 60;
        max-lease-time 60;
}
```

Site 2 VLAN 21 et VLAN 22 :

```
#Site 2 vlan12
subnet 192.168.12.0 netmask 255.255.255.240 {
        range 192.168.12.2 192.168.12.14;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.12.1;
        option broadcast-address 192.168.12.15;
        default-lease-time 60;
        max-lease-time 60;
}
#Site 2 vlan22
subnet 192.168.22.0 netmask 255.255.255.224 {
        range 192.168.22.2 192.168.22.30;
        option domain-name-servers 8.8.8.8;
        option domain-name "internal.example.org";
        option routers 192.168.22.1;
        option broadcast-address 192.168.22.31;
        default-lease-time 60;
        max-lease-time 60;
}
```

`range` indique la plage des adresses IP devant être distribuées dans le sous-réseau ;

`option domain-name-server` indique l'adresse IP du serveur DNS utilisé ;

`option domain-name` indique le nom de domaine utilisé par les machines ;

`option routers` indique l'adresse IP de la passerelle du sous-réseau ;

`option broadcast-address` indique l'adresse de broadcast du sous-réseau ;

`default-lease-time` est une configuration de la durée par défaut du bail DHCP pour ce sous-réseau en particulier ;

`max-lease-time` est une configuration du temps maximal du bail pour ce sous-réseau en particulier ;

Mise en place de DHCPv6

Dans cette partie nous allons vous présenter comme nous avons mis en place le service DHCP pour ipv6. La configuration de ce service se distingue sur plusieurs points du service DHCP ipv4.
Tout d'abord, la configuration de DHCPv6 demande d'indiquer sur chaque routeur que les adresses et paramètres sont fournis par un serveur DHCPv6. Effectivement chaques routeurs envoie des messages RA (Router Advertisement) permettant d'indiquer quelle méthode est utilisée par le biais drapeaux (flags, M pour DHCPv6 et O pour SLAAC) . Une autre méthode est disponible même si nous ne l'utilisons pas et celle-ci est la méthode SLAAC (Stateless Address Autoconfiguration) où le routeur annonce le préfixe puis les hôtes s'auto-configurent sans serveur DHCP.

Pour ce faire, nous avons dû effectuer sur chaques interfaces des VLAN des routeurs 1, 2 et 3 les commande suivantes :

R(config)# interface <interface.n°vlan>
R(config-if)# **ipv6 nd managed-config-flag**
R(config-if)# exit

La commande **ipv6 nd managed-config-flag** active le flag **M=1** dans les messages RA. Ainsi, les clients savent qu'ils doivent contacter un **serveur DHCPv6** pour obtenir leur configuration complète.

Ensuite, nous avons eu besoin de mettre en place sur les interfaces des VLAN des routeurs 1, 2 et 3 le DHCPv6 Relay étant donné que le serveur DHCPv6 (addresse 2001:db8:1:1::2) n'est pas connecté aux VLANs, et donc chaque routeur doit relayer les requêtes DHCPv6 reçues sur ces interfaces VLAN vers le serveur DHCPv6 centralisé.

Pour ce faire, nous avons utiliser la commande suivantes :

R(config)# interface <interface.n°vlan>

R(config-if)# ipv6 dhcp relay destination 2001:db8:1:1::2

R(config-if)# exit

Une fois cela effectué nous avons configuré le service isc-dhcp-server6.servoce sur la machine réseau.

Pour ce faire nous avons créé le fichier **/etc/dhcp/dhcpd6.conf** qui est identique dans la manière d'être écrit que le fichier /etc/dhcp/dhcpd.conf.

Voici le contenu important de ce fichier :

```
# AUtoriser clients à obtenir addresse
authoritative;

# Global definitions for name server address(es) and domain search li
option dhcp6.name-servers 2001:db8:1:1::2;
option dhcp6.domain-search "internal.example.org";
```

Cette partie permet donc de spécifier le type d'autorisation côté client (ici une autorisation d'obtenir une adresse).
Mais aussi de spécifier l'adresse IPv6 du serveur DHCP (2001:db8:1:1::2) et son nom de domaine.
Ensuite Nous avons mis en place le type d'attribution d'adresses IPv6 pour chaques VLAN en suivant les directive présente sur notre Topologie :

```
# VLAN 10
subnet6 2001:db8:1:10::/64{
        range6 2001:db8:1:10::100 2001:db8:1:10::200;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}

# VLAN 20
subnet6 2001:db8:1:20::/64{
        range6 2001:db8:1:20::100 2001:db8:1:20::200;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}
```

```
#VLAN11
subnet6 2001:db8:2:11::/64{
        range6 2001:db8:2:11::100 2001:db8:2:11::1ff;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}

#VLAN21
subnet6 2001:db8:2:21::/64{
        range6 2001:db8:2:21::100 2001:db8:2:21::1ff;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}

#VLAN12
subnet6 2001:db8:3:12::/64{
        range6 2001:db8:3:12::100 2001:db8:3:12::1ff;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}

#VLAN22
subnet6 2001:db8:3:22::/64{
        range6 2001:db8:3:22::100 2001:db8:3:22::1ff;
        option dhcp6.name-servers 2001:db8:1:1::2;
        option dhcp6.domain-search "internal.example.org";
}
```

Concernant le temps de location de chaque adresses ip nous avons gardé celles de base proposées par le service DHCP, comme présenté ci-dessous :

```
default-lease-time 2592000;

# IPv6 address preferred lifetime
#  (at the end the address is deprecated, i.e., the client should use
#   other addresses for new connections)
#  (set to 7 days, the  usual IPv6 default)
preferred-lifetime 604800;

# T1, the delay before Renew
#  (default is 1/2 preferred lifetime)
#  (set to 1 hour)
option dhcp-renewal-time 3600;

# T2, the delay before Rebind (if Renews failed)
#  (default is 3/4 preferred lifetime)
#  (set to 2 hours)
option dhcp-rebinding-time 7200;

# Enable RFC 5007 support (same than for DHCPv4)
allow leasequery;
```

Pour finir, Nous avons eu besoin de créer un nouveau service directement sur le système, c'est pour cela que nous avons eu besoin de créer un fichier **/etc/systemd/system/isc-dhcp-server6.service**.

Voici le contenu du fichier en question :

```
  GNU nano 7.2                          /etc/systemd/system/isc-dhcp-server6.service
[unit]
Description=ISC DHCPv6 Server
After=network.target

[Service]
ExecStart=/usr/sbin/dhcpd -6 -cf /etc/dhcp/dhcpd6.conf ens19
PIDFile=/var/run/dhcpd6.pid

[install]
WantedBy=multi-user.target
```

Ce fichier comporte le texte descriptif du service que le systemd affichera lorsque nous effectuerons systemctl status (« ISC DHCPv6 Server). Et quand est-ce que systemd doit démarrer ce service, dans notre cas le service démarrera après l'unité network.target, cette unité permet d'indiquer que l'étape d'initialisation réseau a bien été atteinte).

Ensuite, dans la partie [Service] la commande *ExecStart=/usr/sbin/dhcpd -6 -cf /etc/dhcp/dhcpd6.conf ens19* indique que lorsque le service démarre nous lançons le serveur DHCP en mode ipv6 en spécifiant qu'il utilisera le fichier de configuration dhcpd6.conf que nous avons créer et qu'il se lancera bien sur l'interface réseau ens19.

Pour finir nous indiquons que quand nous effectuons systemctl enable isc-dhcp-server6.service, alors le service sera démarré automatiquement au boot. Nous avons fait en sorte qu'il est le comportement standard pour les services non-gui.

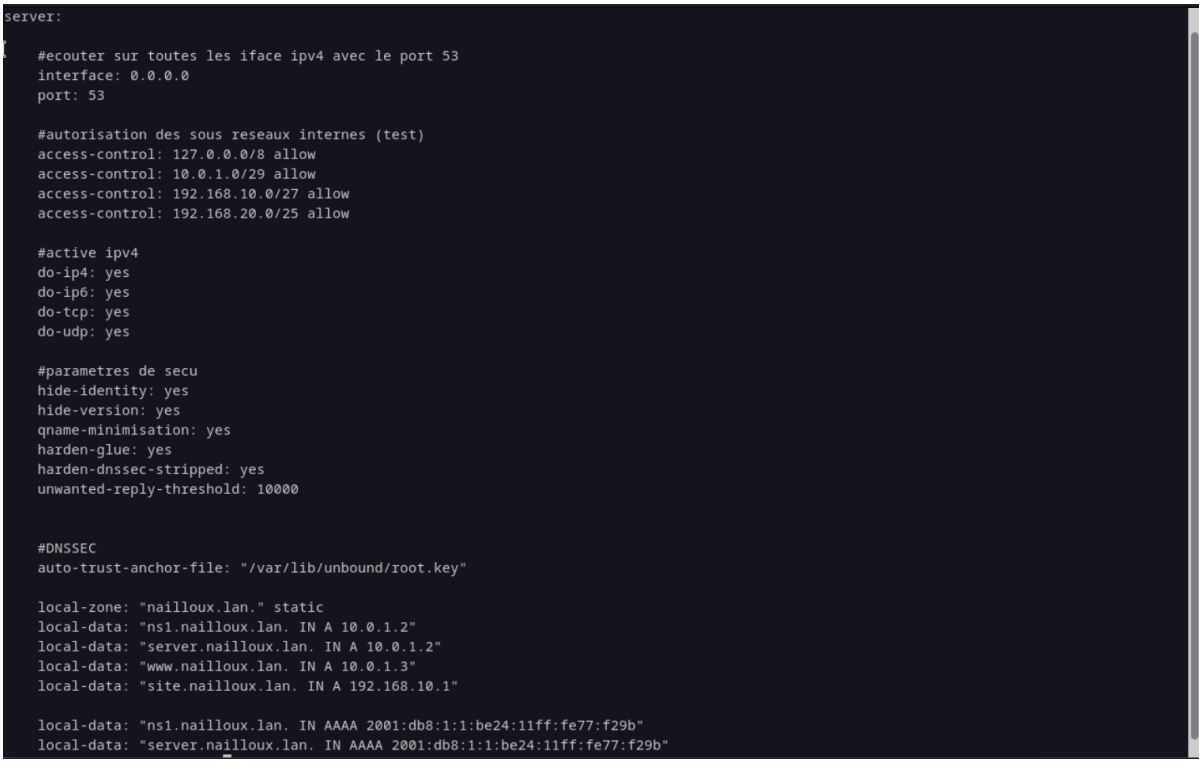
Installation du DNS

Manipulations sur la machine réseau

Installation des paquets nécessaires
apt install -y unbound

- unbound : paquet permettant l'installation d'un résolveur DNS local

Dans le fichier /etc/unbound/unbound.conf :



```
server:

    #ecouter sur toutes les iface ipv4 avec le port 53
    interface: 0.0.0.0
    port: 53

    #autorisation des sous reseaux internes (test)
    access-control: 127.0.0.0/8 allow
    access-control: 10.0.1.0/29 allow
    access-control: 192.168.10.0/27 allow
    access-control: 192.168.20.0/25 allow

    #active ipv4
    do-ip4: yes
    do-ip6: yes
    do-tcp: yes
    do-udp: yes

    #parametres de secu
    hide-identity: yes
    hide-version: yes
    qname-minimisation: yes
    harden-glue: yes
    harden-dnssec-stripped: yes
    unwanted-reply-threshold: 10000


    #DNSSEC
    auto-trust-anchor-file: "/var/lib/unbound/root.key"

    local-zone: "nailloux.lan." static
    local-data: "ns1.nailloux.lan. IN A 10.0.1.2"
    local-data: "server.nailloux.lan. IN A 10.0.1.2"
    local-data: "www.nailloux.lan. IN A 10.0.1.3"
    local-data: "site.nailloux.lan. IN A 192.168.10.1"

    local-data: "ns1.nailloux.lan. IN AAAA 2001:db8:1:1:be24:11ff:fe77:f29b"
    local-data: "server.nailloux.lan. IN AAAA 2001:db8:1:1:be24:11ff:fe77:f29b"
```

Il est le principal fichier de configuration du DNS. Il contient les paramètres de fonctionnement du résolveur.

`access-control: X.X.X.X allow` indique une que le résolveur résout les noms de domaine si les requêtes viennent du sous-réseau X.X.X.X ;

`hide-identity: yes` masque l'identité du DNS ;
`hide-version: yes` masque la version de Unbound ;
`qname-minimisation: yes` retire des requêtes les informations non strictement nécessaires à l'obtention d'une réponse ;
`harden-glue: yes` vérifie la cohérence des enregistrements glue (les IP des serveurs de noms renvoyées dans les réponses).
`harden-dnssec-stripped: yes` refuse les réponses DNS qui devraient contenir des signatures DNSSEC mais qui en sont dépourvues ;
`unwanted-reply-threshold: 10000` définit un seuil de réponses DNS "non sollicitées" (c'est-à-dire qui ne correspondent pas à une requête envoyée) ;

`auto-trust-anchor-file: "/chemin/vers/root.key"` chemin absolu vers le

fichier.key contenant la clé racine DNSSEC ;

`local-zone: "nailloux.lan" static` définit une zone DNS à traiter de façon spécifique ;

`local-data: "xyz.nailloux.lan" IN A X.X.X.X` associe le nom xyz.nailloux.lan à l'adresse IPv4 X.X.X.

`local-data: "xyz.nailloux.lan" IN AAAA X:X:X:X:X:X:X:X` associe xyz.nailloux.lan à l'adresse IPv6 X:X:X:X:X:X:X:X

Manipulations sur la machine applicative

Dans le fichier /etc/resolv.conf :

```
root@debian:~# cat /etc/resolv.conf
# Generated by NetworkManager
#nameserver 100.100.100.254
nameserver 10.0.1.2
```

On met en commentaire le résolveur DNS la défaut 100.100.100.254, et on ajoute une ligne avec l'@IP de notre propre résolveur. L'IP est celle de la machine réseau.

Dans le fichier /opt/Site/nginx/nginx.conf :

```
server {
    listen 80;
    server_name www.nailloux.lan nailloux.lan;
    return 301 https://$host$request_uri;
}


server {
    listen 443 ssl;
    server_name www.nailloux.lan nailloux.lan;
    ssl_certificate /etc/nginx/certs/nailloux.crt;
    ssl_certificate_key /etc/nginx/certs/nailloux.key;

    absolute_redirect off;
    port_in_redirect off;


    client_max_body_size 30M;

    access_log /var/log/nginx/nailloux_access.log;
    error_log /var/log/nginx/nailloux_error.log;

    location / {
        proxy_pass http://app:80;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;

        proxy_connect_timeout 300;
```

```
        proxy_send_timeout 300;
        proxy_read_timeout 300;
        send_timeout 300;
    }

    location ~* \.(jpg|jpeg|png|gif|ico|css|js|svg|woff|woff2|ttf|eot)$ {
        proxy_pass http://app:80;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto https;
        expires 30d;
        add_header Cache-Control "public, immutable";
    }
}
```

Dans le fichier DockerFile on ajoute :

```
# Set ServerName globally to prevent Apache warnings
RUN echo "ServerName www.nailloux.lan" >> /etc/apache2/apache2.conf
```

Et dans le fichier DockerCompose :

```
services:
  nginx:
    image: nginx:alpine
    container_name: nailloux-club-nginx
    restart: unless-stopped
    ports:
      - "80:80"
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf:ro
    networks:
      - nailloux-network
    extra_hosts:
      - "www.nailloux.lan:10.0.1.3"
    depends_on:
      - app
```

On ajoute la ligne ‘extra_hosts:’ .

Mise en place du NAT/PAT

Mise en place du NAT/PAT IPV4 - routeur1

```
Router1
interface GigabitEthernet0/0/2
 description LAN to App Server
 ip nat inside
 no shutdown
!
interface GigabitEthernet0/0/3
 description LAN to Router3
 ip nat outside
 no shutdown
!
ip nat inside source static tcp 10.0.1.3 80 10.0.3.1 80 (http)
ip nat inside source static tcp 10.0.1.3 443 10.0.3.1 443 (https)

Router1#show ip nat translations
Pro  Inside global      Inside local       Outside local      Outside global
tcp  10.0.3.1:80        10.0.1.3:80        ---                ---
tcp  10.0.3.1:443       10.0.1.3:443       ---                ---
Total number of translations: 2
```

Mise en place du dynamic PAT IPV4 - routeur3

```
Router3#conf t
Enter configuration commands, one per line.
End with CNTL/Z.
Router3(config)#interface g0/1
Router3(config-if)#ip nat outside
Router3(config-if)#exit
Router3(config)#interface g0/2.12
Router3(config-subif)#ip nat inside
Router3(config-subif)#exit
Router3(config)#interface g0/2.22
Router3(config-subif)#ip nat inside
Router3(config-subif)#exit
Router3(config)#access-list 1 permit
192.168.12.0 0.0.0.15
Router3(config)#access-list 1 permit
192.168.22.0 0.0.0.31
```

```
Router3(config)#ip nat inside source list 1
interface GigabitEthernet0/1 overload
```

Configuration de BGP

IPv4 et IPv6

Routeur 1	router bgp 65001 bgp log-neighbor-changes neighbor 10.0.3.2 remote-as 65003 neighbor 2001:DB8:1:2::3 remote-as 65003 ! address-family ipv4 network 10.0.1.0 mask 255.255.255.252 neighbor 10.0.3.2 activate no neighbor 2001:DB8:1:2::3 activate exit-address-family ! address-family ipv6 network 2001:DB8:1:1::/126 neighbor 2001:DB8:1:2::3 activate exit-address-family
Routeur 3	router bgp 65003 bgp log-neighbor-changes neighbor 10.0.3.1 remote-as 65001 neighbor 2001:DB8:1:2::2 remote-as 65001 ! address-family ipv4 network 192.168.12.0 mask 255.255.255.240 network 192.168.22.0 mask 255.255.255.224 neighbor 10.0.3.1 activate exit-address-family ! address-family ipv6 network 2001:DB8:3:12::/64 network 2001:DB8:3:22::/64 neighbor 2001:DB8:1:2::2 activate exit-address-family

Elaboration des ACL

Une ACL est une liste de règles permettant de filtrer ou d'autoriser du trafic sur un réseau en fonction de
certains critères (IP source, IP destination, port source, port destination, protocole, …). Une ACL permet soit d'autoriser du trafic (permit) ou de le bloquer (deny).

Procédure de création des règles

Syntaxe d'une règle ACL :
access-list <numéro|nom> <type> <protocole> <IP_source> <masque_wildcard_source> <IP_destination> <masque_wildcard_destination> [port]

Syntaxe	Champ attendu	Rôle du champ
<numéro | nom>	**au choix**	nom de la règle ACL
<type>	permit / deny	autorise ou refuse les paquets qui correspondent aux paramètres de la règle
<protocol>	tcp / udp / ospf / …	précise le protocole concerné par la règle
<IP_source>	**adresse IP de réseau**	identifie le SR auquel appliquer la règle
<masque_wildcard_source>	**masque wildcard du SR de IP_source**	identifie le SR auquel appliquer la règle
<IP_destination>	**adresse IP de réseau**	identifie le SR auquel appliquer la règle
<masque_wildcard_destination>	**masque wildcard du SR de IP_source**	identifie le SR auquel appliquer la règle
[port]	eq / lt / gt / range	champ optionnel, indique le ou les ports sur lesquels appliquer la règle : - eq <port> → port égal à X ; - lt <port> → ports supérieurs à X - gt <port> → ports inférieurs à X ; - range <port1> <port2> → ports de X à Y

En l'espèce, les services que nous utilisons et leur port correspondant sont les suivants :

Service	Port associé

HTTP	80
HTTPS	443
DNS	53
DHCPv4 (côté serveur)	67
DHCPv4 (côté client)	68
DHCPv6 (côté serveur)	546
DHCPv6 (côté client)	547