

Configurations

L'adressage IP

[IPv4](#)

[IPv6](#)

Installation/configuration d'OSPF

[Configurations OSPF d'un routeur :](#)

[Configurations OSPF IPV6 d'un routeur :](#)

Installation et configuration de DHCP

[Mise en place de DHCPv4](#)

[Mise en place de DHCPv6](#)

Installation du DNS

[Manipulations sur la machine réseau](#)

[Configuration d'Unbound](#)

[Configurations effectuées](#)

Mise en place du NAT/PAT

[Mise en place du NAT/PAT IPV4 - routeur1](#)

[Mise en place du dynamic PAT IPV4 - routeur3](#)

Configuration de BGP

Elaboration des ACL

[Procédure de création des règles](#)

Déploiement du site sur la VM Applicative

Outil de monitoring

[Qu'est-ce qu'OpenObserve ?](#)

[À quoi ça sert ?](#)

[Comment ça fonctionne ?](#)

[Les configurations mises en places :](#)

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

Dû à notre architecture intégrant un WAN, nous avons intégré une solution OSPF pour relier le site principal et le site 1.

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

Configurations OSPF d'un 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
--	--

Configurations OSPF IPV6 d'un routeur :

<u>Routeur 1</u>	<pre> 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 </pre>
-------------------------	---

Installation et configuration de DHCP

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

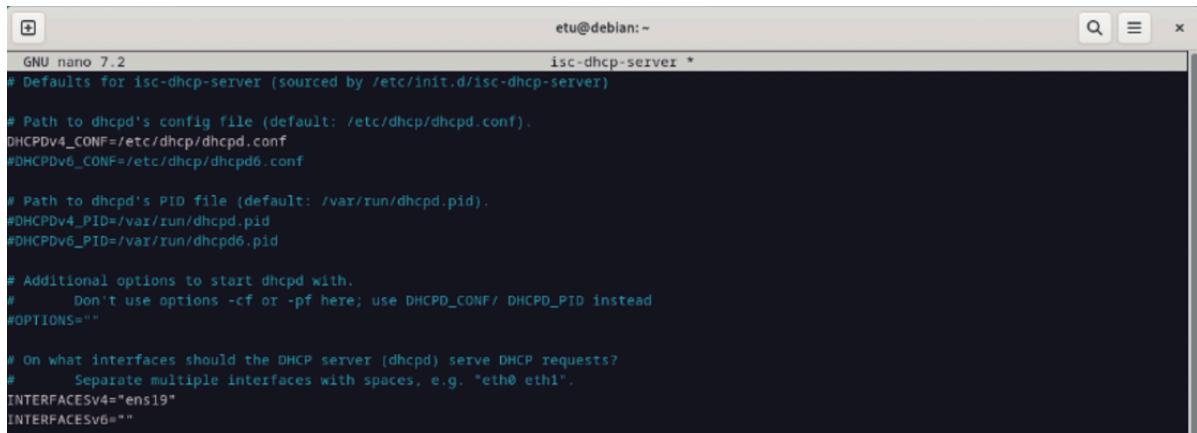
L'objectif de la mise en place des service DHCPv4 et DHCPv6 est de permettre aux hôtes présents dans les différents VLAN de notre réseau d'obtenir automatiquement une adresse IPv4 et IPv6 à partir du serveur DHCP centralisé présent sur notre machine machine réseau dans situé dans le site principale de 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 :



```
etu@debian:~$ nano /etc/default/isc-dhcp-server
GNU nano 7.2                               isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpcd's config file (default: /etc/dhcp/dhcpcd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpcd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpcd6.conf

# Path to dhcpcd's PID file (default: /var/run/dhcpcd.pid).
#DHCPDv4_PID=/var/run/dhcpcd.pid
#DHCPDv6_PID=/var/run/dhcpcd6.pid

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

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens19"
INTERFACESv6=""
```

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

`DHCPDv4_CONF=/etc/dhcp/dhcpcd.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 :

```
GNU nano 7.2                               /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpcd
#
# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# The ddns-updates-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.)
ddns-update-style none;

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

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

^G Aide      ^O Écrire      ^W Chercher      ^K Couper      ^T Exécuter      ^C Emplacement M-U Annuler      M-A Marquer      M-J -> Crochet
^X Quitter    ^R Lire fich.  ^Y Remplacer    ^U Coller       ^J Justifier    ^I Aller ligne M-E Refaire      M-B Copier       ^Q Retrouver
```

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 bail 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 chaque sites conformément à notre topologie.

Voici la configuration pour chaque VLAN du réseau :

Site Principal, VLAN 10 et VLAN 20 :

```
# max-lease-time 7200;
#
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 192.168.21.31;
    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 commandes 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 (adresse 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 suivante :

```
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.service sur la machine réseau.

Pour ce faire nous avons créé le fichier **/etc/dhcp/dhcpcd6.conf** qui est identique dans la manière d'être écrit que le fichier /etc/dhcp/dhcpcd.conf.

Voici le contenu important de ce fichier :

```
# Autoriser clients à obtenir adresse
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 chaque 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 ens1
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 créé 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

Configuration d'Unbound

Installation des paquets nécessaires

```
apt install -y unbound
```

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

Configurations effectuées

Pour installer le DNS nous avons modifié le fichier de configuration /etc/unbound/unbound.conf. Dans celui-ci, nous avons ajouté les adresses IP vers lesquelles le DNS doit rediriger pour les noms de domaine nailloux.lan. Dans ce même fichier nous avons précisé les paramètres de sécurité et de confidentialité avec lesquels le DNS doit fonctionner.

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

Routeur1

```

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
```

Routeur3

```
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
no neighbor 2001:DB8:1:2::2 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

Déploiement du site sur la VM Applicative

Pour pouvoir déployer le site on a utilisé docker qui nous a permis de lancer plusieurs services en même temps dans des conteneurs isolés.

Outil de monitoring

Qu'est-ce qu'OpenObserve ?

OpenObserve est une plateforme d'observabilité qui permet de surveiller et analyser ce qui se passe dans un système informatique.

Elle sert à rassembler, stocker et visualiser différents types de données comme :

- Les logs (messages produits par les serveurs et applications)
- Les métriques (informations chiffrées comme l'utilisation du processeur ou de la mémoire)
- Les traces (le parcours d'une requête à travers plusieurs services)

À quoi ça sert ?

OpenObserve aide les administrateurs systèmes et les développeurs à :

- Comprendre l'état des serveurs et des applications
- Déetecter les erreurs ou les pannes
- Suivre les performances (par exemple si un service devient lent)
- Centraliser toutes les informations au même endroit

Grâce à cela, on peut rapidement savoir ce qui ne va pas dans une infrastructure sans devoir chercher sur chaque machine.

Comment ça fonctionne ?

Des agents (comme Vector ou Fluent Bit) envoient les données (logs ou métrics) vers OpenObserve. Il enregistre ces données dans sa base interne. L'utilisateur peut ensuite les afficher sous forme de graphiques, tableaux ou recherches dans une interface web.

Les configurations mises en place :

Pour mettre en place la solution de monitoring, on a utilisé Docker et Vector, deux outils essentiels pour déployer et gérer l'infrastructure de supervision. Docker nous a permis de lancer facilement plusieurs services dans des conteneurs isolés, comme le serveur de supervision, la base de données et le proxy Nginx. L'utilisation de Docker Compose a simplifié le déploiement en regroupant tous les services dans un seul fichier de configuration, ce qui rend le système plus clair, plus facile à maintenir et plus portable. Grâce à Docker, chaque service fonctionne indépendamment des autres, tout en restant connecté dans un même réseau virtuel.

L'agent Vector, de son coté, a été installé pour assurer la collecte et la transmission des logs vers la plateforme de supervision(vector est installé sur la vm applicatif). Il récupère automatiquement les journaux du système et des conteneurs (comme ceux d'Apache, Nginx ou du système Debian), puis les envoie au serveur d'observation(la vm réseau dans notre cas). Vector peut aussi filtrer, transformer et enrichir les données avant de les transmettre, ce qui permet d'avoir des informations claires et organisées dans l'interface de monitoring. Son rôle est donc de centraliser les logs provenant de différentes sources afin d'avoir une vision complète de l'activité des machines et de faciliter le diagnostic en cas de problème.