Haute disponibilité de l’infrastructure réseau de GSB

Modification des liens trunk

Pour SW4 :

```
SWE4(config-if-Et1)#interface eth15-16
SWE4(config-if-Et15-16)#switchport mode trunk
```

```
SWE4(config-if-Et15-16)#interface eth1-16
SWE4(config-if-Et1-16)#no shutdown
```

```
SWE4(config-if-Et15-16)#no switchport trunk allowed vlan 20,30,99,150
SWE4(config-if-Et15-16)#sh run
```

Résultat :

```
interface Ethernet15
   switchport access vlan 666
   switchport mode trunk
!
interface Ethernet16
   switchport trunk native vlan 999
   switchport mode trunk
```

Ce que je dois obtenir : pour chaque interface, je dois avoir le port en mode trunk et comme native vlan, le vlan 666 afin d'assurer que tout le trafic non étiqueté soit correctement assigné au VLAN 666.

Explications :

Pour autoriser tous les Vlan à passer, je vais autoriser tous les liens trunk des switch. C’est une solution plus sécurisée car, si on est amené à adapter le réseau et créer d’autres Vlan, on peut facilement tout contrôler car il faudrait ajouter le vlan au lien trunk pour qu’il puisse passer.
Je fais cette même manipulation pour les 2 autres switch :

Pour SW5 :

```
SWE5(config)#interface eth15-16
SWE5(config-if-Et15-16)#switchport mode trunk
```

Pour vérifier que tous les liens sont bien trunk, j’utilise la commande sh run :

```
interface Ethernet15
   switchport access vlan 666
   switchport mode trunk
!
interface Ethernet16
   switchport trunk native vlan 999
   switchport mode trunk
```

SW6 :

```
interface Ethernet15
   switchport access vlan 666
   switchport mode trunk
!
interface Ethernet16
   switchport trunk native vlan 999
   switchport mode trunk
```

Exemple pour les premiers ports.

Je vais faire la même manipulation pour Mutlab, Mutlab2 et SW6.
Je prends pour exemple Mutlab pour montrer que toutes les interfaces sont bien trunk et que il laisse passer tous les vlan :

```
MUTLAB(config-if-Et14)#interface eth4-6
MUTLAB(config-if-Et4-6)#no switchport trunk allowed vlan 20,30,99,150
```

Quand je fais la commande pour afficher l'état des liens:

```
interface Ethernet4
   switchport trunk native vlan 999
   switchport mode trunk
!
interface Ethernet5
   switchport trunk native vlan 999
   switchport mode trunk
```

```
interface Ethernet14
   switchport access vlan 666
   switchport mode trunk
```

Pour MUTLAB :

Il faut a chaque fois 2 lignes :

```
MUTLAB(config)#interface eth6
MUTLAB(config-if-Et6)#no switchport access vlan 666
```

Quand je vérifie :

```
interface Ethernet4
   switchport trunk native vlan 999
   switchport mode trunk
!
interface Ethernet5
   switchport trunk native vlan 999
   switchport mode trunk
!
interface Ethernet6
   switchport trunk native vlan 999
   switchport mode trunk
```

```
interface Ethernet14
   switchport access vlan 666
   switchport mode trunk
```

J'ai bien mes 2 lignes pour chaque interface

Pour MUTLAB 2 :

```
interface Ethernet4
   switchport trunk native vlan 999
   switchport mode trunk
!
interface Ethernet5
   switchport trunk native vlan 999
   switchport mode trunk
!
interface Ethernet6
   switchport trunk native vlan 999
   switchport mode trunk
```

```
interface Ethernet14
   switchport trunk native vlan 999
   switchport mode trunk
```

Il faut aussi créer tous les vlan sur tous les switch pour qu'il laisse passer n'importe quelle machine qui serait susceptible de passer dans des switch.

Sur SW4 :

```
SWE4(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et16
10    Reseau&Systeme                   active    Cpu
20    Direction                        active    Et1, Et2, Et3, Et4
30    Administratif                    active    Et5, Et6, Et7, Et8
99    Management                       active    Cpu
150   Visiteurs                        active    Et9, Et10, Et11, Et12
666   Blackhole                        active
999   Native                           active

SWE4(config)#vlan300
% Invalid input
SWE4(config)#vlan 300
SWE4(config-vlan-300)#name Serveurs
SWE4(config-vlan-300)#exit
SWE4(config)#
```

Il ne manquait que le vlan 300 pour ce switch.

Voici sa table mise à jour :

```
SWE4(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et16
10    Reseau&Systeme                   active    Cpu
20    Direction                        active    Et1, Et2, Et3, Et4
30    Administratif                    active    Et5, Et6, Et7, Et8
99    Management                       active    Cpu
150   Visiteurs                        active    Et9, Et10, Et11, Et12
300   Serveurs                         active
666   Blackhole                        active
999   Native                           active
```

Le vlan 300 apparait bien dans la table et les machine de ce vlan pourront passer par ce switch.

Pour SW5 :

Pour ce switch, c'était encore le vlan 300 qu'il manquait.

```
SWE5(config)#vlan 300
SWE5(config-vlan-300)#name Serveurs
SWE5(config-vlan-300)#exit
SWE5(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et16
10    Reseau&Systeme                   active    Et1, Et2, Et3, Et4
20    Direction                        active
30    Administratif                    active    Et5, Et6, Et7, Et8
99    Management                       active    Cpu
150   Visiteurs                        active
300   Serveurs                         active
666   Blackhole                        active
999   Native                           active
```

Pour SW6:

```
SWE6(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et16
99    Management                       active    Cpu
300   Serveurs                         active    Et1, Et2, Et3, Et4, Et5, Et6
                                                 Et7, Et8
666   blackhole                        active
999   Native                           active
```

Cette fois ci, il manque les vlan 10,20,30 et 150.

```
SWE6(config)#vlan10
% Invalid input
SWE6(config)#vlan 10
SWE6(config-vlan-10)#name Reseaux&Systeme
SWE6(config-vlan-10)#exit
SWE6(config)#vlan 20
SWE6(config-vlan-20)#name Direction
SWE6(config-vlan-20)#exit
SWE6(config)#vlan 30
SWE6(config-vlan-30)#Administratif
% Invalid input
SWE6(config-vlan-30)#name Administratif
SWE6(config-vlan-30)#exit
SWE6(config)#vlan 150
SWE6(config-vlan-150)#name Visiteurs
SWE6(config-vlan-150)#exit
SWE6(config)#vlan 300
SWE6(config-vlan-300)#name Serveurs
SWE6(config-vlan-300)#exit
```

Une fois ajouté :

```
SWE6(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et16
10    Reseaux&Systeme                  active
20    Direction                        active
30    Administratif                    active
99    Management                       active    Cpu
150   Visiteurs                        active
300   Serveurs                         active    Et1, Et2, Et3, Et4, Et5, Et6
                                                 Et7, Et8
666   blackhole                        active
999   Native                           active
```

Pour finir, je rajoute les vlan sur MUTLAB 2, la même configuration que sur MUTLAB (en utilisant les mêmes commandes que précédemment):

```
MUTLAB2(config)#sh vlan
VLAN  Name                             Status    Ports
----- -------------------------------- --------- -------------------------------
1     default                          active    Et1, Et2, Et3, Et4, Et6, Et7
                                                 Et8, Et9, Et10, Et11, Et12
                                                 Et13, Et14, Et15, Et16
10    Reseau&Systeme                   active
20    Direction                        active
30    Administration                   active
99    Management                       active
150   Visiteurs                        active
300   Serveurs                         active
```

Configuration du protocole MSTP

Pour configurer le protocole MSTP, qui va permettre la priorisation de chemin de certains vlan à utiliser des switch. Cela a pour but de répartir la charge des différents switch

Je vais donc créer l'instance 1, celle de MUTLAB, qui contiendra les VLAN 10, 20 et 30:

```
MUTLAB(config)#spanning-tree mode mstp
MUTLAB(config)#spanning-tree mst configuration
MUTLAB(config-mst)#name MUTLAB1
MUTLAB(config-mst)#instance 1 vlan 10,20,30
MUTLAB(config-mst)#
```

Puis l'instance 2, celle de MUTLAB2, qui contiendra les VLAN 99,150 et 300:

```
MUTLAB2(config-mst)#spanning-tree mode mstp
MUTLAB2(config)#spanning-tree mst configuration
MUTLAB2(config-mst)#name MUTLAB2
MUTLAB2(config-mst)#instance 2 vlan 99,150,300
MUTLAB2(config-mst)#
```

Je vais maintenant attribuer un pont racine pour chaque instance, MUTLAB sera le pont racine de l'instance 1 et MUTLAB2 sera celui de l'instance 2.

```
MUTLAB(config-mst)#revision 1
MUTLAB(config-mst)#spanning-tree mst 1 priority 4096
MUTLAB(config)#
```

Je nomme l'instance 1, révision 1 pour identifier la configuration MSTP.
Puis je donne une priorité de 4096 à MUTLAB pour qu'il soit "élu" pont racine.

Je vérifie ensuite à l'aide de la commande sh spanning-tree pour voir la configuration.
La ligne "This bridge is the root" indique bien que le switch a été élu pont racine.

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             This bridge is the root

  Bridge ID  Priority    4097  (priority 4096 sys-id-ext 1)
             Address     28b3.de6a.e38b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec
```

Toujours sur MUTLAB, je crée l'instance 2 pour ensuite lui attribuer une priorité moins haute donc un nombre plus important pour que ce soit le switch MUTLAB2 qui soit élu.

```
MUTLAB(config)#spanning-tree mode mstp
MUTLAB(config)#spanning-tree mst configuration
MUTLAB(config-mst)#instance 2 vlan 99,150,300
MUTLAB(config-mst)#revision 2
MUTLAB(config-mst)#spanning-tree mst 2 priority 8192
```

Ducoup sur mutlab1, quand je fais la commande sh spanning-tree , le mst2 est donc bien renseigné mais le switch n'est pas élu racine :

Je vais faire pareil pour l'instance 2 avec MUTLAB 2 :

```
MUTLAB2(config-mst)#spanning-tree mst 2 priority 4096
MUTLAB2(config)#instance 1 vlan 10,20,30
% Invalid input
MUTLAB2(config)#spanning-tree mst configuration
MUTLAB2(config-mst)#instance 1 vlan 10,20,30
MUTLAB2(config-mst)#spanning-tree mst 1 priority 8192
MUTLAB2(config)#
```

J'obtiens ce résultat:

```
MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0209.dbb6.e70e
             Cost        20000
             Port        14 (Ethernet14)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec
```

MUTLAB n'est pas élu racine car il a une priorité plus haute et il a détecter qu'il devait renvoyer vers eth14 pour atteindre la switch racine (MUTLAB2).

Exemple pour MUTLAB :
Je vais vérifier les vlan associé à l'instance 1 pour MUTLAB à l'aide de la commande show spanning-tree mst configuration qui va me permettre de voir les vlan associés à la configuration MSTP:

```
MUTLAB(config)#show spanning-tree mst configuration
Name      [MUTLAB1]
Revision  1    Instances configured 2

Instance  Vlans mapped
--------  ----------------------------------------------------------------------
0         1-9,11-19,21-29,31-4094
1         10,20,30
-------------------------------------------------------------------------------
```

Pour l'instance 1, j'ai bien les vlan 10,20,30 renseignés.

Il faut ensuite que je configure toute les MST sur tous les switch :

```
MUTLAB(config)#spanning-tree mst configuration
MUTLAB(config-mst)#instance 2
% Incomplete command
MUTLAB(config-mst)#instance 2 vlan 99,150,300
```

J'ai renseigné les valeurs des vlan de l'instance 2, je vais faire la même manipulation pour les autres switch.
Pour chaque switch:

```
SWE5(config)#spanning-tree mode mstp
SWE5(config)#spanning-tree mst configuration
SWE5(config-mst)#instance 1 vlan 10,20,30
SWE5(config-mst)#instance 1 vlan 99,150,300
SWE5(config-mst)#exit
SWE5(config)#
```

Pour MUTLAB 2 :

```
MUTLAB2(config)#spanning-tree mst configuration
MUTLAB2(config-mst)#instance 1 vlan 10,20,30
MUTLAB2(config-mst)#show spanning-tree
```

On vérifie la conf sur les switch :

SW4:

```
spanning-tree mode mstp
!
spanning-tree mst configuration
   instance 1 vlan  10,20,30
   instance 2 vlan  99,150,300
```

SW5:

```
spanning-tree mode mstp
!
spanning-tree mst configuration
   instance 1 vlan  10,20,30
   instance 2 vlan  99,150,300
!
```

SW6:

```
spanning-tree mode mstp
!
spanning-tree mst configuration
   instance 1 vlan  10,20,30
   instance 2 vlan  99,150,300
!
```

Je vérifie la conf avec la commande suivante:

Voici les configurations des mst sur chaque switch :

Pour MUTLAB:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             This bridge is the root

  Bridge ID  Priority     4097  (priority 4096 sys-id-ext 1)
             Address     28b3.de6a.e38b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p

Et5              designated forwarding 20000     128.5    P2p

Et6              designated forwarding 20000     128.6    P2p

Et14             designated forwarding 20000     128.14   P2p


MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0e0f.1471.0fc7
             Cost        20000
             Port        14 (Ethernet14)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority     8194  (priority 8192 sys-id-ext 2)
             Address     28b3.de6a.e38b
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p

Et5              designated forwarding 20000     128.5    P2p

Et6              designated forwarding 20000     128.6    P2p

Et14             root       forwarding 20000     128.14   P2p
```

Pour MUTLAB2:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             Cost        20000
             Port        14 (Ethernet14)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority     8193  (priority 8192 sys-id-ext 1)
             Address     0e0f.1471.0fc7
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p

Et5              designated forwarding 20000     128.5    P2p

Et6              designated forwarding 20000     128.6    P2p

Et14             root       forwarding 20000     128.14   P2p


MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0e0f.1471.0fc7
             This bridge is the root

  Bridge ID  Priority     4098  (priority 4096 sys-id-ext 2)
             Address     0e0f.1471.0fc7
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et4              designated forwarding 20000     128.4    P2p

Et5              designated forwarding 20000     128.5    P2p

Et6              designated forwarding 20000     128.6    P2p

Et14             designated forwarding 20000     128.14   P2p


MUTLAB2(config)#
```

SW4:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             Cost        20000
             Port        16 (Ethernet16)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     28b3.deaf.eedc
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              designated forwarding 20000     128.1    P2p Edge

Et2              designated forwarding 20000     128.2    P2p Edge

Et3              designated forwarding 20000     128.3    P2p Edge

Et4              designated forwarding 20000     128.4    P2p Edge

Et5              designated forwarding 20000     128.5    P2p Edge

Et6              designated forwarding 20000     128.6    P2p Edge

Et7              designated forwarding 20000     128.7    P2p Edge

Et8              designated forwarding 20000     128.8    P2p Edge

Et15             alternate  discarding 20000     128.15   P2p

Et16             root       forwarding 20000     128.16   P2p


MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0e0f.1471.0fc7
             Cost        20000
             Port        15 (Ethernet15)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
             Address     28b3.deaf.eedc
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et9              designated forwarding 20000     128.9    P2p Edge

Et10             designated forwarding 20000     128.10   P2p Edge

Et11             designated forwarding 20000     128.11   P2p Edge

Et12             designated forwarding 20000     128.12   P2p Edge

Et15             root       forwarding 20000     128.15   P2p

Et16             alternate  discarding 20000     128.16   P2p
```

SW5:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             Cost        20000
             Port        16 (Ethernet16)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     28b3.de31.7a1c
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              designated forwarding 20000     128.1    P2p Edge

Et2              designated forwarding 20000     128.2    P2p Edge

Et3              designated forwarding 20000     128.3    P2p Edge

Et4              designated forwarding 20000     128.4    P2p Edge

Et5              designated forwarding 20000     128.5    P2p Edge

Et6              designated forwarding 20000     128.6    P2p Edge

Et7              designated forwarding 20000     128.7    P2p Edge

Et8              designated forwarding 20000     128.8    P2p Edge

Et15             alternate  discarding 20000     128.15   P2p

Et16             root       forwarding 20000     128.16   P2p


MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0e0f.1471.0fc7
             Cost        20000
             Port        15 (Ethernet15)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
             Address     28b3.de31.7a1c
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et15             root       forwarding 20000     128.15   P2p

Et16             alternate  discarding 20000     128.16   P2p


SWE5(config)#
```

SW6:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    4097
             Address     28b3.de6a.e38b
             Cost        20000
             Port        16 (Ethernet16)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     28b3.defb.79f0
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et15             alternate  discarding 20000     128.15   P2p

Et16             root       forwarding 20000     128.16   P2p


MST2
  Spanning tree enabled protocol mstp
  Root ID    Priority    4098
             Address     0e0f.1471.0fc7
             Cost        20000
             Port        15 (Ethernet15)
             Hello Time  0.000 sec  Max Age  0 sec  Forward Delay  0 sec

  Bridge ID  Priority    32770  (priority 32768 sys-id-ext 2)
             Address     28b3.defb.79f0
             Hello Time  2.000 sec  Max Age 20 sec  Forward Delay 15 sec

Interface        Role       State      Cost      Prio.Nbr Type
---------------- ---------- ---------- --------- -------- --------------------
Et1              designated forwarding 20000     128.1    P2p Edge

Et2              designated forwarding 20000     128.2    P2p Edge

Et3              designated forwarding 20000     128.3    P2p Edge

Et4              designated forwarding 20000     128.4    P2p Edge

Et5              designated forwarding 20000     128.5    P2p Edge

Et6              designated forwarding 20000     128.6    P2p Edge

Et7              designated forwarding 20000     128.7    P2p Edge

Et8              designated forwarding 20000     128.8    P2p Edge

Et15             root       forwarding 20000     128.15   P2p

Et16             alternate  discarding 20000     128.16   P2p


SWE6(config)#
```

Nouveaux réseaux d’interconnexion des routeurs

Pour MUTLAB:

```
interface Ethernet15
   switchport access vlan 666
   no switchport
   ip address 172.18.0.9/30
!
interface Ethernet16
   switchport access vlan 666
   no switchport
   ip address 172.18.0.1/30
```

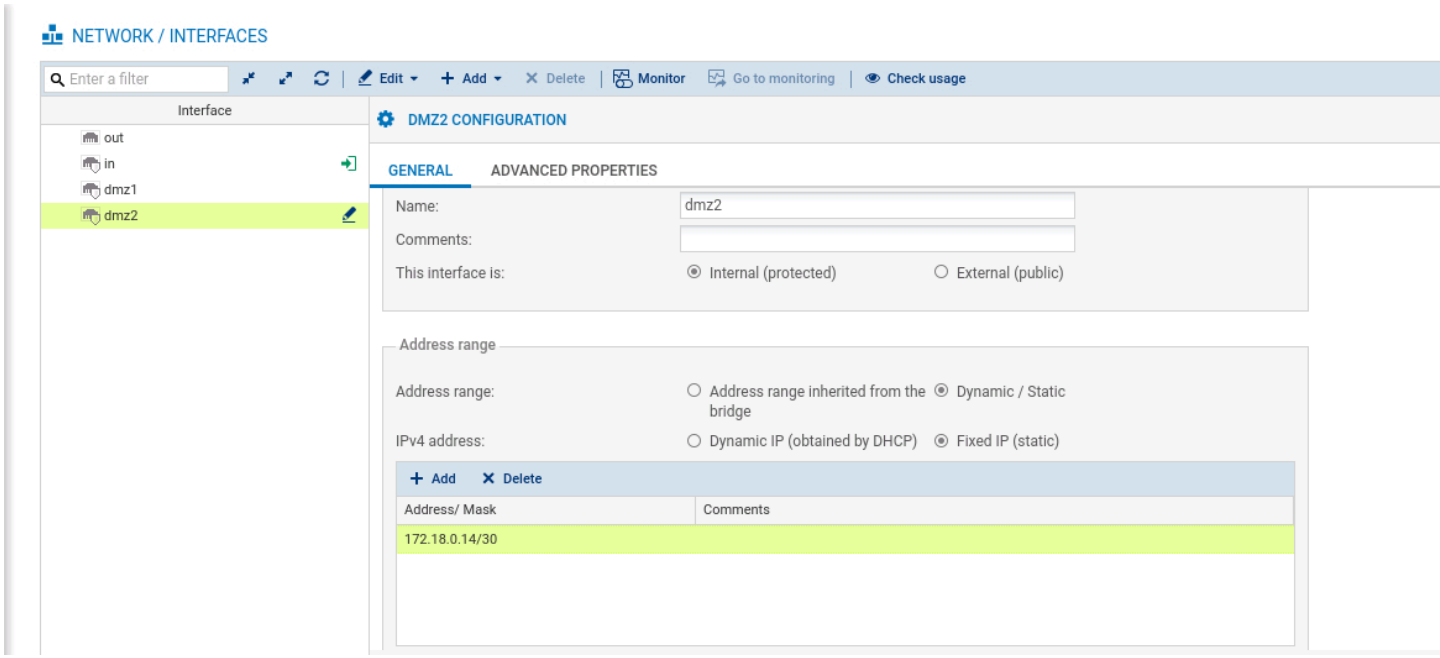
Pour MUTLAB2:

```
interface Ethernet15
   no switchport
   ip address 172.18.0.13/30
!
interface Ethernet16
   no switchport
   ip address 172.18.0.5/30
```

Ce que je dois faire :

Il faut que je configure les interfaces sur Proxilab2 pour que je puisse accéder à stormshield qui est directement installé dessus. Puis je dois faire en sorte qu'il soit accessible via les PC.

Je met une adresse statique à l'interface qui relie mutlab2 à Proxilab: (à la base en DHCP car c'est un lien qui a été ajouté et donc pas configurée)



Je vais résumer les routes sur proxilab2:

Ces règles vont permettre d'accéder au stormshield de Proxilab 2 depuis les PC
J’ai résumé en 2 routes.

Je vais mettre la route pour accéder à proxilab 2:
Ces commandes proviennent de RTROUT que j’ai adapté à RTROUT2

Routage dynamique OSPF

Paramétrage du NAT sur RTROUT2:

Les règles NAT permettent de masquer les adresses IP privées pour qu'elles apparaissent comme une seule adresse publique en sortie:

```
set nat source rule 10 source address '192.168.0.0/16'
set nat source rule 10 translation address 'masquerade'
set nat source rule 11 description 'NAT for internal servers'
set nat source rule 11 outbound-interface 'eth0'
set nat source rule 11 source address '172.16.0.0/17'
set nat source rule 11 translation address 'masquerade'
set nat source rule 12 description 'NAT for ProxSILAB'
set nat source rule 12 outbound-interface 'eth0'
set nat source rule 12 source address '172.30.0.2'
set nat source rule 12 translation address 'masquerade'
```

```
set protocols static route 10.0.0.0/24 next-hop 172.30.0.2
set protocols static route 172.16.0.0/14 next-hop 172.30.0.2
set protocols static route 192.168.0.0/16 next-hop 172.30.0.2
```

```
set service dns forwarding allow-from '172.16.0.10/32'
set service dns forwarding listen-address '172.30.0.1'
set service dns forwarding system
set service ssh listen-address '172.30.0.1'
set system config-management commit-revisions '100'
set system host-name 'RTROUT2'
```

Grâce à la commande sh conf, voici le résultat des commandes entrée précédemment

```
    source {
        rule 10 {
            description "NAT for users"
            outbound-interface eth0
            source {
                address 192.168.0.0/16
            }
            translation {
                address masquerade
            }
        }
        rule 11 {
            description "NAT for internal servers"
            outbound-interface eth0
            source {
                address 172.16.0.0/17
            }
            translation {
                address masquerade
            }
        }
        rule 12 {
            description "NAT for ProxSILAB"
            outbound-interface eth0
            source {
                address 172.30.0.2
            }
            translation {
                address masquerade
            }
        }
    }
}
```

Je vais ensuite configurer le routage OSPF:

Pour RTROUT2:

Je rajoute l'adresse IP sur l'interface eth1:

```
vyos@RTROUT2:~$ set interfaces ethernet eth1 address '172.30.0.1/30'
```

Je vérifie qu'elle a bien été ajoutée:

OBJOBJ

OBJvyos@RTROUT2:~$ sh interfaces

```
Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface        IP Address                        S/L  Description
---------        ----------                        ---  -----------
eth0             100.64.122.207/24                 u/u
eth1             172.30.0.1/30                     u/u
eth2             -                                 u/D
lo               127.0.0.1/8                       u/u
                 ::1/128
```

Nous avons donc ajouté 2 routes. La première qui va sur internet, et la 2e est la route directement connecté (C) .

```
vyos@RTROUT2:~$ sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r -
rejected route

S>* 0.0.0.0/0 [210/0] via 100.64.122.1, eth0, 02:27:01
S>* 10.0.0.0/24 [1/0] via 172.30.0.2, eth1, 00:37:33
C>* 100.64.122.0/24 is directly connected, eth0, 02:27:01
S>* 172.16.0.0/14 [1/0] via 172.30.0.2, eth1, 00:37:33
C>* 172.30.0.0/30 is directly connected, eth1, 00:37:33
S>* 192.168.0.0/16 [1/0] via 172.30.0.2, eth1, 00:37:33
```

Je vais permettre au réseau 172.30.0.0 à pouvoir détecter les routes automatiquement :

```
#set protocols ospf area 0 network '172.30.0.0/30'
#set protocols ospf area 0 network '172.18.0.0/30'
#set protocols ospf area 0 network '172.18.0.12/30'
```

A la base ma route par défaut est créée par dhcp mais elle a un poids plus important (210) que le routes par ospf (110), cela veut dire que les routes créées par ospf étaient prioritaire par rapport à la route par défaut. Il faut donc supprimer la route crée avec le dhcp avec la commande : (Sur les deux RTROUT)

```
#set protocols ospf area 0 network '172.18.0.0/30'
```

Cela va supprimer la route par défaut. Puis on va rajouter la route de facon statique qui aura un poids de 1. Les routes statiques sont prioritaire et ont donc le poids minimum :

Pour ajouter la route par défaut de façon statique (sur les deux RTROUT) :

```
vyos@RTROUT:~$ set protocols static route 0.0.0.0/0 next-hop
100.64.122.1
```

La route est normalement créé, on va vérifier sur les deux RTROUT qu'elle est bien présente :

```
S>* 0.0.0.0/0 [1/0] via 100.64.122.1, eth0, 00:07:41
```

Il faut aussi ajouter la route par défaut par ospf à l'aide de la commande suivante :

```
vyos@RTROUT:~$ set protocols ospf default-information originate
```

Je configure OSPF sur les 2 MUTLAB :

```
MUTLAB(config)#interface eth15
MUTLAB(config)#router ospf area 0
MUTLAB(config)#interface eth16
MUTLAB(config)#router ospf area 0
```

Je fais la même chose sur MUTLAB2:

```
MUTLAB2(config)#interface eth15
MUTLAB2(config)#router ospf area 0
MUTLAB2(config)#interface eth16
MUTLAB2(config)#router ospf area 0
```

Il faut ensuite empêcher les requêtes OSPF de se propager vers les PC, car cela pollue le réseau inutilement :

- Il faut donc passer les interfaces 4,5,6 en mode passive
- Les interface 15 et 16 en mode no passive interface

Sur MUTLAB2 :
Je n'oublie pas d'activer l'ip routing comme sur MUTLAB

```
MUTLAB(config)#router ospf 1
MUTLAB(config-router-ospf)#network 172.18.0.4/30 area 0.0.0.0
MUTLAB(config-router-ospf)#network 172.18.0.12/30 area 0.0.0.0
MUTLAB(config-router-ospf)#max-lsa 1200
```

Voici le résultat sur MUTLAB et MUTLAB2 :

Sur MUTLAB:

```
interface Ethernet15
   switchport access vlan 666
   no switchport
   ip address 172.18.0.9/30
   ip ospf area 0.0.0.0
!
interface Ethernet16
   switchport access vlan 666
   no switchport
   ip address 172.18.0.1/30
   ip ospf area 0.0.0.0
!

router ospf 1

   no passive-interface Ethernet4

   no passive-interface Ethernet5

   no passive-interface Ethernet6

   network 172.16.0.0/16 area 0.0.0.0

   network 172.18.0.4/30 area 0.0.0.0

   network 172.18.0.12/30 area 0.0.0.0

   network 192.168.0.0/16 area 0.0.0.0

   max-lsa 1200

!
```

Sur MUTLAB2:

```
interface Ethernet15
   no switchport
   ip address 172.18.0.13/30
   ip ospf area 0.0.0.0
!
interface Ethernet16
   no switchport
   ip address 172.18.0.5/30
   ip ospf area 0.0.0.0
router ospf 1

   passive-interface Ethernet4

   passive-interface Ethernet5

   passive-interface Ethernet6

   no passive-interface Ethernet15

   no passive-interface Ethernet16

   network 172.16.0.0/16 area 0.0.0.0

   network 172.18.0.0/30 area 0.0.0.0

   network 172.18.0.8/30 area 0.0.0.0

   network 192.168.0.0/16 area 0.0.0.0

   max-lsa 1200
```

J’ajoute les SVI à MUTLAB2 (elles sont déjà présente sur MUTLAB):

```
interface Vlan10
   ip address 192.168.10.2/24
   ip helper-address 172.16.0.10
!
interface Vlan20
   ip address 192.168.20.2/24
   ip helper-address 172.16.0.10
!
interface Vlan30
   ip address 192.168.30.2/24
   ip helper-address 172.16.0.10
!
interface Vlan99
   ip address 172.16.128.2/17
!
interface Vlan150
   ip address 192.168.150.2/24
   ip helper-address 172.16.0.10
!
interface Vlan300
   ip address 172.16.0.2/17
!
```

Voici le résultat sur mes MUTLAB :

MUTLAB:

```
MUTLAB(config)#sh ip route ospf

Gateway of last resort:
 O E2     0.0.0.0/0 [110/10]
           via 172.18.0.10, Ethernet15
           via 172.18.0.2, Ethernet16

 O        172.18.0.4/30 [110/20]
           via 172.18.0.10, Ethernet15
 O        172.18.0.12/30 [110/20]
```

```
           via 172.18.0.2, Ethernet16
 O         172.30.0.0/30 [110/20]
           via 172.18.0.10, Ethernet15
 O         172.31.0.0/30 [110/20]
           via 172.18.0.2, Ethernet16
```

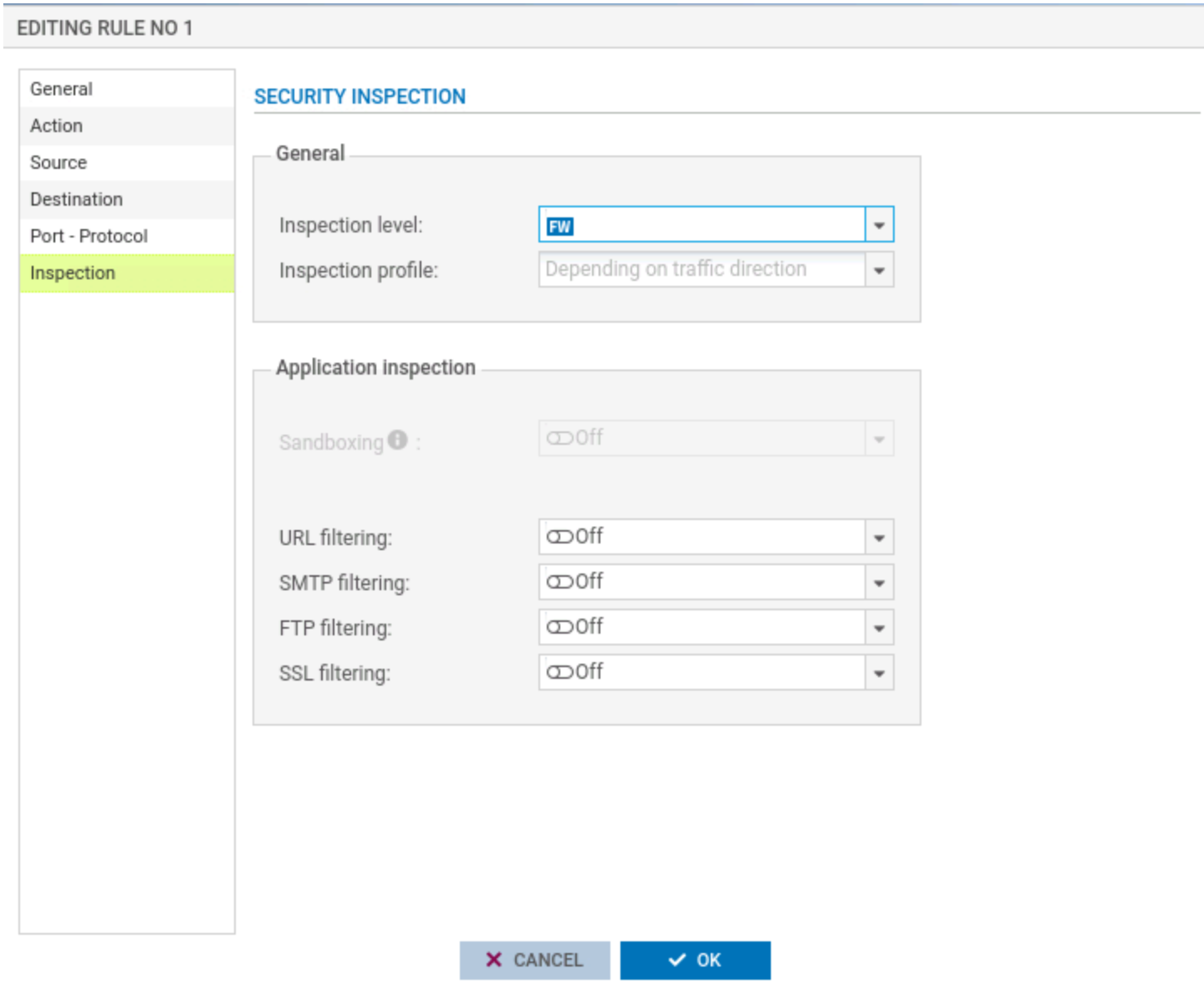
MUTLAB2:

```
MUTLAB2(config-if-Vl300)#sh ip route ospf
Gateway of last resort:
 O E2     0.0.0.0/0 [110/10]
           via 172.18.0.14, Ethernet15
           via 172.18.0.6, Ethernet16

 O         172.18.0.0/30 [110/20]
           via 172.18.0.14, Ethernet15
 O         172.18.0.8/30 [110/20]
           via 172.18.0.6, Ethernet16
 O         172.30.0.0/30 [110/20]
           via 172.18.0.6, Ethernet16
 O         172.31.0.0/30 [110/20]
           via 172.18.0.14, Ethernet15
```

Il y a bien 4 routes crée par le protocole OSPF, et une en plus qui est la route par défaut

Sur les Proxilab :
Je les mets en mode FW, qui indique au Stormshield d'agir juste comme Pare-Feu.



Configuration pour OSPF sur les proxilab:

Pour Proxilab :

```
# Le protocole direct génère automatiquement des routes pour toutes
les interfaces réseau.
protocol direct {
}

# Ce pseudo-protocole synchronise les tables de routage de BIRD avec
le noyau.
```

```
protocol kernel {
      learn;          # Apprend les routes du noyau
      persist;        # Ne pas supprimer les routes à l'arrêt de BIRD
      scan time 20;         # Scanner les routes du noyau toutes les
20 secondes
      import all;     # Importer toutes les routes
      export all;     # Exporter toutes les routes
      preference 254;       # Préserver les routes existantes
}

# Ce pseudo-protocole surveille les événements up/down des
interfaces.
protocol device {
      scan time 10;         # Scanner les interfaces toutes les 10
secondes
}

# Ajout de la configuration OSPF
protocol ospf OSPF1 {
      # Configuration de base
      router id 3.3.3.3;  # ID unique du routeur

      area 0 {
      # Configuration pour l'interface in
      interface "in" {
            #cost 10;       # Coût de l'interface (optionnel)
            type broadcast; # Type de réseau
      };
      # Configuration pour l'interface out
      interface "out" {
            #cost 10;       # Coût de l'interface (optionnel)
            type broadcast; # Type de réseau
      };
      # Configuration pour l'interface dmz2
      interface "dmz2" {
            #cost 20;
            type broadcast;
```

```
      };
      };

      # Importation et exportation des routes
      import all;           # Importer toutes les routes OSPF
      export none;          # Exporter aucune route (modifiable selon
les besoins)
}
```

Pour Proxilab2:

```
# Le protocole direct génère automatiquement des routes pour toutes
les interfaces réseau.
protocol direct {
}

# Ce pseudo-protocole synchronise les tables de routage de BIRD avec
le noyau.
protocol kernel {
      learn;           # Apprend les routes du noyau
      persist;         # Ne pas supprimer les routes à l'arrêt de BIRD
      scan time 20;         # Scanner les routes du noyau toutes les
20 secondes
      import all;      # Importer toutes les routes
      export all;      # Exporter toutes les routes
      preference 254;       # Préserver les routes existantes
}

# Ce pseudo-protocole surveille les événements up/down des
interfaces.
protocol device {
      scan time 10;         # Scanner les interfaces toutes les 10
secondes
}

# Ajout de la configuration OSPF
protocol ospf OSPF1 {
```

```
      # Configuration de base
      router id 4.4.4.4;  # ID unique du routeur

      area 0 {
      # Configuration pour l'interface in
      interface "in" {
            #cost 10;       # Coût de l'interface (optionnel)
            type broadcast; # Type de réseau
      };
      # Configuration pour l'interface out
      interface "out" {
            #cost 10;       # Coût de l'interface (optionnel)
            type broadcast; # Type de réseau
      };
      # Configuration pour l'interface dmz2
      interface "dmz2" {
            #cost 20;
            type broadcast;
      };
      };

      # Importation et exportation des routes
      import all;           # Importer toutes les routes OSPF
      export none;          # Exporter aucune route (modifiable selon
les besoins)
}
```

Voici le résultat sur mes RTROUT :

Pour RTROUT :

```
vyos@RTROUT:~$ sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
```

```
       > - selected route, * - FIB route, q - queued route, r -
rejected route

O   0.0.0.0/0 [110/10] via 172.31.0.2, eth1, 00:24:50
S>* 0.0.0.0/0 [1/0] via 100.64.122.1, eth0, 01:14:00
C>* 100.64.122.0/24 is directly connected, eth0, 01:14:00
O>* 172.16.0.0/17 [110/30] via 172.31.0.2, eth1, 00:02:39
O>* 172.16.128.0/17 [110/30] via 172.31.0.2, eth1, 00:02:39
O>* 172.18.0.0/30 [110/20] via 172.31.0.2, eth1, 01:13:56
O>* 172.18.0.4/30 [110/30] via 172.31.0.2, eth1, 00:19:11
O>* 172.18.0.8/30 [110/30] via 172.31.0.2, eth1, 00:24:51
O>* 172.18.0.12/30 [110/20] via 172.31.0.2, eth1, 01:13:56
O>* 172.30.0.0/30 [110/40] via 172.31.0.2, eth1, 00:24:51
O   172.31.0.0/30 [110/10] is directly connected, eth1, 01:14:00
C>* 172.31.0.0/30 is directly connected, eth1, 01:14:04
O>* 192.168.10.0/24 [110/30] via 172.31.0.2, eth1, 00:05:55
O>* 192.168.20.0/24 [110/30] via 172.31.0.2, eth1, 00:05:55
O>* 192.168.30.0/24 [110/30] via 172.31.0.2, eth1, 00:05:55
O>* 192.168.150.0/24 [110/30] via 172.31.0.2, eth1, 00:05:55
```

Pour RTROUT2:

```
vyos@RTROUT2:~$ sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued route, r -
rejected route

O   0.0.0.0/0 [110/10] via 172.30.0.2, eth1, 00:16:35
S>* 0.0.0.0/0 [1/0] via 100.64.122.1, eth0, 01:26:42
C>* 100.64.122.0/24 is directly connected, eth0, 01:26:42
O>* 172.16.0.0/17 [110/30] via 172.30.0.2, eth1, 00:02:42
O>* 172.16.128.0/17 [110/30] via 172.30.0.2, eth1, 00:02:42
O>* 172.18.0.0/30 [110/30] via 172.30.0.2, eth1, 00:24:54
O>* 172.18.0.4/30 [110/20] via 172.30.0.2, eth1, 01:26:04
```

```
O>* 172.18.0.8/30 [110/20] via 172.30.0.2, eth1, 01:26:04
O>* 172.18.0.12/30 [110/30] via 172.30.0.2, eth1, 00:19:14
O   172.30.0.0/30 [110/10] is directly connected, eth1, 01:26:46
C>* 172.30.0.0/30 is directly connected, eth1, 01:26:48
O>* 172.31.0.0/30 [110/40] via 172.30.0.2, eth1, 00:24:54
O>* 192.168.10.0/24 [110/30] via 172.30.0.2, eth1, 00:05:58
O>* 192.168.20.0/24 [110/30] via 172.30.0.2, eth1, 00:05:58
O>* 192.168.30.0/24 [110/30] via 172.30.0.2, eth1, 00:05:58
O>* 192.168.150.0/24 [110/30] via 172.30.0.2, eth1, 00:05:58
```

Vérification : Je fais un sh ospf neighbor

Sur RTROUT :

```
vyos@RTROUT:~$ sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface
                RXmtL RqstL DBsmL
3.3.3.3           1 Full/Backup       32.045s 172.31.0.2      eth1:172.31.0.1
                  0     0     0
```

Sur RTROUT2:

```
vyos@RTROUT2:~$ sh ip ospf neighbor

Neighbor ID     Pri State           Dead Time Address         Interface
                RXmtL RqstL DBsmL
4.4.4.4           1 Full/Backup       35.127s 172.30.0.2      eth1:172.30.0.1
                  0     0     0
```

Sur MUTLAB:

```
MUTLAB(config)#sh ip ospf neighbor
Neighbor ID     Instance VRF      Pri State                  Dead Time   Address
         Interface
3.3.3.3              1       default  1   FULL/BDR              00:00:36    172.18.
0.2      Ethernet16
4.4.4.4              1       default  1   FULL/BDR              00:00:37    172.18.
0.10     Ethernet15
```

Sur MUTLAB2:

```
MUTLAB2(config-if-Vl300)#sh ip ospf neighbor
Neighbor ID     Instance VRF      Pri State                  Dead Time   Address
         Interface
4.4.4.4         1        default  1   FULL/BDR              00:00:33    172.18.
0.6      Ethernet16
3.3.3.3         1        default  1   FULL/BDR              00:00:32    172.18.
0.14     Ethernet15
```

Configuration de VRRP

Nous allons configurer VRRP entre MUTLAB et MUTLAB2 pour garantir que les passerelles restent disponibles même si une panne arrive.

Sur MUTLAB:

Vlan10 :

```
MUTLAB(config)#interface Vlan10
MUTLAB(config-if-Vl10)#vrrp 10 ipv4 192.168.10.254
MUTLAB(config-if-Vl10)# vrrp 10 priority-level 200
MUTLAB(config-if-Vl10)#vrrp 10 preempt
```

Vlan20 :

```
MUTLAB(config-if-Vl10)#interface Vlan20
MUTLAB(config-if-Vl20)# vrrp 20 ipv4 192.168.20.254
MUTLAB(config-if-Vl20)# vrrp 20 priority-level 200
MUTLAB(config-if-Vl20)# vrrp 20 preempt
```

Vlan30 :

```
MUTLAB(config-if-Vl20)#interface Vlan30
MUTLAB(config-if-Vl30)# vrrp 30 ipv4 192.168.30.254
MUTLAB(config-if-Vl30)# vrrp 30 priority-level 200
MUTLAB(config-if-Vl30)# vrrp 30 preempt
```

vlan99:

```
MUTLAB(config-if-Vl30)#interface Vlan99
MUTLAB(config-if-Vl99)# vrrp 99 ipv4 172.16.255.254
MUTLAB(config-if-Vl99)# vrrp 99 priority-level 100
MUTLAB(config-if-Vl99)# vrrp 99 preempt
```

Vlan150 :

```
MUTLAB(config-if-Vl99)#interface Vlan150
MUTLAB(config-if-Vl150)# vrrp 150 ipv4 192.168.150.254
MUTLAB(config-if-Vl150)# vrrp 150 priority-level 100
MUTLAB(config-if-Vl150)# vrrp 150 preempt
```

Vlan300:

```
MUTLAB(config-if-Vl150)# interface Vlan300
MUTLAB(config-if-Vl300)# vrrp 200 ipv4 172.16.127.254
MUTLAB(config-if-Vl300)# vrrp 200 priority-level 100
MUTLAB(config-if-Vl300)# vrrp 200 preempt
```

Sur MUTLAB2:

Vlan10:

```
MUTLAB2(config)#interface Vlan10
MUTLAB2(config-if-Vl10)# vrrp 10 ipv4 192.168.10.254
MUTLAB2(config-if-Vl10)# vrrp 10 priority-level 100
MUTLAB2(config-if-Vl10)# vrrp 10 preempt
```

Vlan20:

```
MUTLAB2(config-if-Vl10)#interface Vlan20
MUTLAB2(config-if-Vl20)# vrrp 20 ipv4 192.168.20.254
MUTLAB2(config-if-Vl20)# vrrp 20 priority-level 100
MUTLAB2(config-if-Vl20)# vrrp 20 preempt
```

Vlan30:

```
MUTLAB2(config-if-Vl20)#interface Vlan30
MUTLAB2(config-if-Vl30)# vrrp 30 ipv4 192.168.30.254
MUTLAB2(config-if-Vl30)# vrrp 30 priority-level 100
MUTLAB2(config-if-Vl30)# vrrp 30 preempt
```

Vlan99:

```
MUTLAB2(config-if-Vl30)#interface Vlan99
MUTLAB2(config-if-Vl99)# vrrp 99 ipv4 172.16.255.254
MUTLAB2(config-if-Vl99)# vrrp 99 priority-level 200
MUTLAB2(config-if-Vl99)# vrrp 99 preempt
```

Vlan150:

```
MUTLAB2(config-if-Vl99)#interface Vlan150
```

```
MUTLAB2(config-if-Vl150)# vrrp 150 ipv4 192.168.150.254
MUTLAB2(config-if-Vl150)# vrrp 150 priority-level 200
MUTLAB2(config-if-Vl150)# vrrp 150 preempt
```

Vlan300:

```
MUTLAB2(config-if-Vl150)#interface Vlan300
MUTLAB2(config-if-Vl300)# vrrp 200 ipv4 172.16.127.254
MUTLAB2(config-if-Vl300)# vrrp 200 priority-level 200
MUTLAB2(config-if-Vl300)# vrrp 200 preempt
```

Les configurations sur MUTLAB et MUTLAB2 ont été ajustées pour que MUTLAB soit prioritaire pour les VLANs 10, 20, et 30 avec des priorités de 120 et les vlan99,150 et 300 ont une priorité plus haute donc moins importante (200), tandis que MUTLAB2 soit prioritaire pour les VLANs 99, 150, et 300 avec des priorités de 100 et les vlan 10,20,30 ont un poids de 200 car moins prioritaire, ce qui va garantir que chaque routeur gère les VLANs qui lui sont attribués tout en assurant une redondance via VRRP.
L'option preempt va permettre au routeur ayant la priorité la plus élevée de prendre le relais en cas de défaillance.

Voici le résultat de ces commandes :

Pour MUTLAB :

```
Vlan10 - Group 10
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 00:17:27 ago
  Virtual IPv4 address is 192.168.10.254
  Virtual MAC address is 0000.5e00.010a
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 192.168.10.1 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s

Vlan20 - Group 20
  VRF is default
  VRRP Version 2
```

```
  State is Master
  Last state transition was 00:17:12 ago
  Virtual IPv4 address is 192.168.20.254
  Virtual MAC address is 0000.5e00.0114
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 192.168.20.1 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s

Vlan30 - Group 30
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 00:17:02 ago
  Virtual IPv4 address is 192.168.30.254
  Virtual MAC address is 0000.5e00.011e
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 192.168.30.1 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s

Vlan99 - Group 99
  VRF is default
  VRRP Version 2
  State is Backup
  Last state transition was 00:15:27 ago
  Virtual IPv4 address is 172.16.255.254
  Virtual MAC address is 0000.5e00.0163
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
```

```
  Priority is 100
  Master Router is 172.16.128.2, priority is 100
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s

Vlan150 - Group 150

  VRF is default
  VRRP Version 2
  State is Backup
  Last state transition was 00:15:21 ago
  Virtual IPv4 address is 192.168.150.254
  Virtual MAC address is 0000.5e00.0196
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 192.168.150.2, priority is 100
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s
```

Pour MUTLAB2:

```
Vlan10 - Group 10
  VRF is default
  VRRP Version 2
  State is Backup
  Last state transition was 00:17:06 ago
  Virtual IPv4 address is 192.168.10.254
  Virtual MAC address is 0000.5e00.010a
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 192.168.10.1, priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s
```

```
Vlan20 - Group 20
  VRF is default
  VRRP Version 2
  State is Backup
  Last state transition was 00:16:51 ago
  Virtual IPv4 address is 192.168.20.254
  Virtual MAC address is 0000.5e00.0114
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 192.168.20.1, priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s

Vlan30 - Group 30
  VRF is default
  VRRP Version 2
  State is Backup
  Last state transition was 00:16:40 ago
  Virtual IPv4 address is 192.168.30.254
  Virtual MAC address is 0000.5e00.011e
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 192.168.30.1, priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s

Vlan99 - Group 99
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 00:15:06 ago
  Virtual IPv4 address is 172.16.255.254
  Virtual MAC address is 0000.5e00.0163
  Mac Address Advertisement interval is 30s
```

```
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 172.16.128.2 (local), priority is 100
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s

Vlan150 - Group 150
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 00:15:00 ago
  Virtual IPv4 address is 192.168.150.254
  Virtual MAC address is 0000.5e00.0196
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 100
  Master Router is 192.168.150.2 (local), priority is 100
  Master Advertisement interval is 1s
  Skew time is 0.600s
  Master Down interval is 3.600s

Vlan300 - Group 200
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 00:10:34 ago
  Virtual IPv4 address is 172.16.127.254
  Virtual MAC address is 0000.5e00.01c8
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 172.16.0.2 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s
```

On voit bien que pour MUTLAB, les vlan 10,20,30 sont prioritaires (Master Router is 172.16.0.2 (local)) et pour MUTLAB2, ce sont les vlan 99,150 et 300. Pour savoir si un vlan n'est pas prioritaire, on peut le vérifier car la ligne est diférente (`Master Router is 172.16.0.1 (local))`. La communication entre les 2 MUTLAB se fait bien puisqu'ils détective qui a la plus forte/faible priorité afin de choisir qui a la priorité sur l'autre pour chaque vlan.

Vérification : Je lance un ping depuis rezolab vers internet (8.8.8.8), comme il fait partie du vlan 300, c'est MUTLAB2 qui est prioritaire, si je shutdown l'interface du vlan300, MUTLAB2 ne sera plus considéré comme le routeur prioritaire du vlan 300, c'est donc MUTLAB qui devient automatiquement prioritaire. Et inversement, si pendant que le ping est interrompu, on réactive l'interface, cela va switcher automatiquement vers MUTLAB2.

Preuve de la vérification :

Je lance un ping vers internet depuis rezolab

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

[root@rezolab named]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=108 time=9.25 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=108 time=6.75 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=108 time=7.62 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=108 time=7.35 ms
```

Le ping doit passer vers MUTLAB2 car il fait partie du vlan 300. C'est donc MUTLAB2 qui est prioritaire.

```
Vlan300 - Group 200
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 03:21:24 ago
  Virtual IPv4 address is 172.16.127.254
  Virtual MAC address is 0000.5e00.01c8
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 172.16.0.2 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s
```

MUTLAB2 est donc bien prioritaire pour le vlan 300

Je vais shutdown l'interface vlan300:

MUTLAB2(config)#interface vlan 300
MUTLAB2(config-if-Vl300)#shutdown

Je fais un sh vrrp pour voir si MUTLAB a prit le role de prioritaire pour le vlan300:
avec la commande sh vrrp:

```
Vlan300 - Group 200
  VRF is default
  VRRP Version 2
  State is Master
  Last state transition was 03:21:24 ago
  Virtual IPv4 address is 172.16.127.254
  Virtual MAC address is 0000.5e00.01c8
  Mac Address Advertisement interval is 30s
  VRRP Advertisement interval is 1s
  Preemption is enabled
  Preemption delay is 0s
  Preemption reload delay is 0s
  Priority is 200
  Master Router is 172.16.0.2 (local), priority is 200
  Master Advertisement interval is 1s
  Skew time is 0.210s
  Master Down interval is 3.210s
```

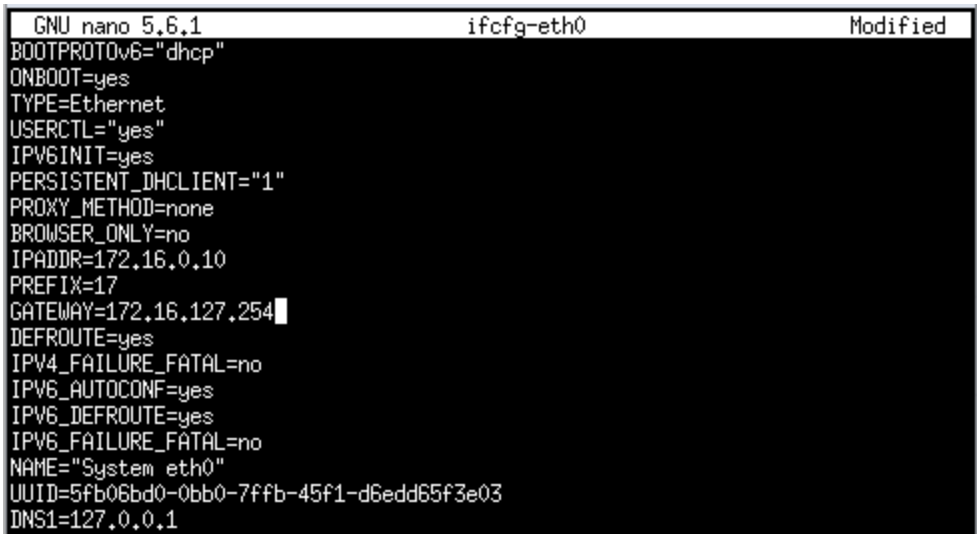
On voit bien que le ping passe toujours (léger temps d'adaptation, capture d'écran ci-dessous) et MUTLAB a bien prit le relais sur la priorité automatiquement afin qu'il n'y ait pas d'interruption.

```
64 bytes from 8.8.8.8: icmp_seq=585 ttl=108 time=7.50 ms
64 bytes from 8.8.8.8: icmp_seq=586 ttl=108 time=9.56 ms
64 bytes from 8.8.8.8: icmp_seq=587 ttl=108 time=6.96 ms
64 bytes from 8.8.8.8: icmp_seq=588 ttl=108 time=7.00 ms
64 bytes from 8.8.8.8: icmp_seq=589 ttl=108 time=7.72 ms
64 bytes from 8.8.8.8: icmp_seq=590 ttl=108 time=7.67 ms
```

Le ping passe toujours malgré le changement de route. (utilisation de MUTLAB au lieu de MUTLAB2)

Modification des services DHCP et DNS

Je change la passerelle dans nano /etc/sysconfig/network-scripts/ifcfg-eth0 :
Au lieu de mettre la SVI du vlan 300 auquel il appartient, on va mettre l'adresse ip de la VRRP.



Il va falloir ensuite modifier les passerelle pour chaque vlan :

Pour le vlan 10 :

```
# VLAN 10 - Reseau&Systeme
subnet 192.168.10.0 netmask 255.255.255.0 {
   range 192.168.10.11 192.168.10.249;
   option routers 192.168.10.254;
}
```

Pour le vlan20:

```
# VLAN 20 - Direction
subnet 192.168.20.0 netmask 255.255.255.0 {
   range 192.168.20.11 192.168.20.249;
   option routers 192.168.20.254;
}
```

Pour le vlan30:

```
# VLAN 30 - Administratif
subnet 192.168.30.0 netmask 255.255.255.0 {
   range 192.168.30.11 192.168.30.249;
   option routers 192.168.30.254;
}
```

Pour le vlan150:

```
# VLAN 150 - Visiteurs
subnet 192.168.150.0 netmask 255.255.255.0 {
```

```
   range 192.168.150.11 192.168.150.249;
   option routers 192.168.150.254;
}
```

Le vlan99 n'a pas besoin car c'est juste le management (pas besoin de DHCP).

Mise en place de l'agent relais DHCP sur MUTLAB2:

```
interface Vlan10
   ip address 192.168.10.2/24
   ip helper-address 172.16.0.10
   vrrp 10 ipv4 192.168.10.254
!
interface Vlan20
   ip address 192.168.20.2/24
   ip helper-address 172.16.0.10
   vrrp 20 ipv4 192.168.20.254
!
interface Vlan30
   ip address 192.168.30.2/24
   ip helper-address 172.16.0.10
   vrrp 30 ipv4 192.168.30.254
!
interface Vlan99
   ip address 172.16.128.2/17
   ip helper-address 172.16.0.10
   vrrp 99 ipv4 172.16.255.254
!
interface Vlan150
   ip address 192.168.150.2/24
   ip helper-address 172.16.0.10
   vrrp 150 ipv4 192.168.150.254
!
interface Vlan300
   ip address 172.16.0.2/17
   ip helper-address 172.16.0.10
   vrrp 200 priority-level 200
   vrrp 200 ipv4 172.16.127.254
!
```

Changement de passerelle de SW4,5 et 6 :

Actuellement la passerelle de chaque switch est la suivante :

```
ip route 0.0.0.0/0 172.16.128.1
```

Actuellement, la passerelle correspond à celle de MUTLAB, on va la modifier en mattant la vrrp du vlan 300 (serveur) pour que si MUTLAB tombe, ce soit MUTLAB2 qui prenne le relais de la passerelle. Pour chaque switch, on va supprimer la passerelle

Pour SW4 :

Suppression de la passerelle :

```
SWE4(config)#no ip route 0.0.0.0/0 172.16.128.1
```

Ajout de la nouvelle passerelle:

```
SWE4(config)#ip route 0.0.0.0/0 172.16.127.254
```

On peut vérifier si la nouvelle passerelle a bien été appliqué en refesant un sh ru et en cherchant la ligne :

```
ip route 0.0.0.0/0 172.16.127.254
```

Pour SW5:

```
SWE5(config)#no ip route 0.0.0.0/0 172.16.128.1
SWE5(config)#ip route 0.0.0.0/0 172.16.127.254
```

Pour SW6:

```
SWE6(config)#no ip route 0.0.0.0/0 172.16.128.1
SWE6(config)#ip route 0.0.0.0/0 172.16.127.254
```

Changement DNS :

Dans le fichier “172.in-addr.arpa.zone“

Je fais les ajouts suivants :

```
1.0.30          IN PTR  rtrout2.gsb.intra
6.0.18          IN PTR  proxilab2.gsb.intra
2.128.16        IN PTR  mutlab2.gsb.intra
```

Dans le fichier “gsb.intra.zone”:

Je fais les ajouts suivants :

```
rtrout2         IN      A       172.30.0.1
proxilab2       IN      A       172.18.0.6
mutlab2         IN      A       172.16.128.2
```

J’ai rajouté les postes manquants dans les confs dns. 172.in-addr.arpa.zone sert à la résolution inverse DNS et gsb.intra.zone pour la résolution directe DNS.

Résumé :

- gsb.intra.zone sert principalement à associer des noms aux adresses IP

- 172.in-addr.arpa.zone permet de faire l'inverse, c'est-à-dire d'associer une adresse IP à un nom