Cluster Proxmox Hyperconvergé avec 2 liens

Cette documentation est une alternative plus avancée a la solution de Cluster Proxmox Hyperconvergé avec un seul lien. Je vous conseille de d'abord commencer par l'autre.

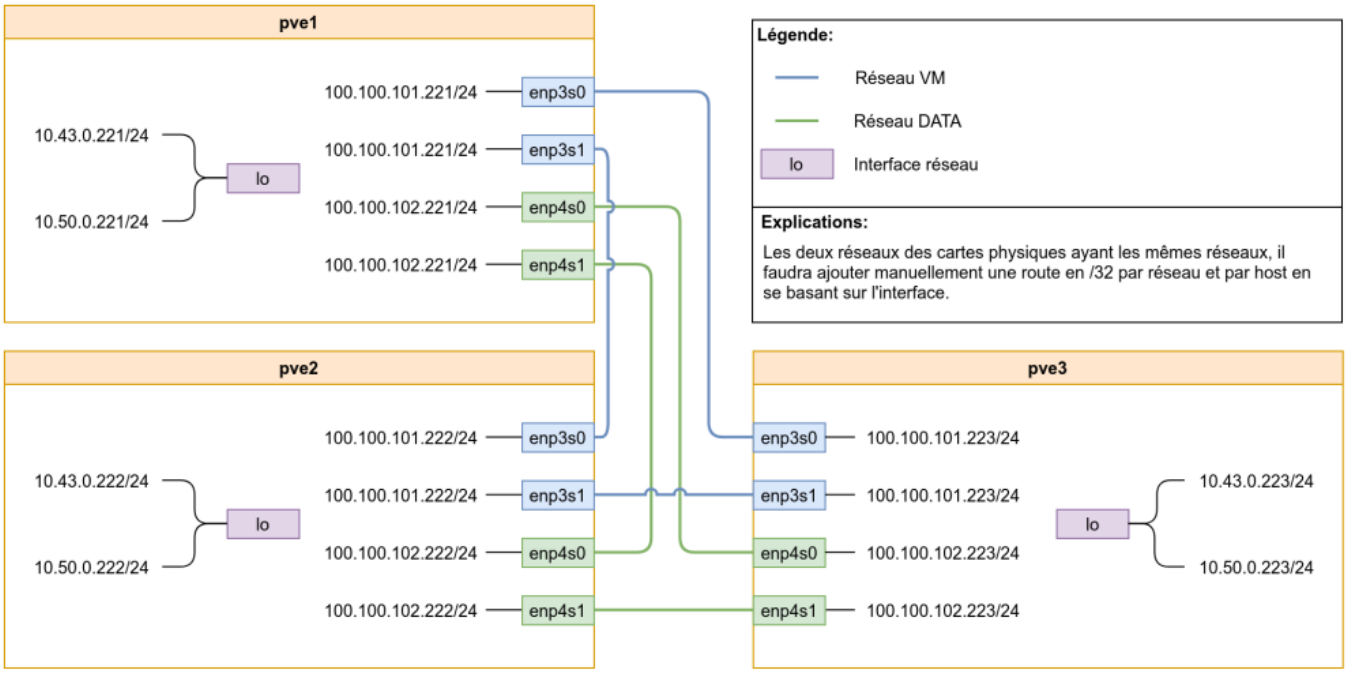
Cette documentation risque d'être obsolète a la sortie de **pve-network** qui intègre le module SDN. Elle sera réécrite quand le module sortira.

Informations préliminaires

La solution repose sur du BGP, du routage en mode peer to peer, du VxLAN, du CEPH.

Il faut bien comprendre que l'on va réaliser n'est pas natif a Proxmox et va demander de contourner certaines contraintes de l'interface en les configurant en CLI.

L'objectif va être de réaliser cette infrastructure :



avec les spécificités suivante:

- Les IPs dans le réseau 10.50.0.0 et 10.43.0.0 sont routé en eBGP.
- Les IPs du réseau 10.50.0.0 servent pour les liaisons CEPH, et sont routé en priorité sur le réseau physique DATA
- Les IPs du réseau 10.43.0.0 servent pour les liaisons entre VM, et sont routé en priorité sur le réseau physique VM. L'IP dans ce réseau devra être utilisé comme “router-id”.
- Si le réseau DATA est injoignable, le noeud passera par le réseau VM, et vice-versa.
- Les IPs pour les rings corosync sont eux routés en statique.

Procédure

Pour commencer, nous avons besoin d'installer **ifupdown2** [1] :

```
# apt install ifupdown2
```

et ensuite il nous faut installer FRR ;

```
# apt install frr
```

Il faut ensuite faire les configurations réseau[2] :

pve1

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto enp3s0
iface enp3s0 inet static
    address 100.100.101.221/32
    mtu 9000

auto enp3s1
iface enp3s1 inet static
    address 100.100.101.221/32
    mtu 9000

auto enp4s0
iface enp4s0 inet static
    address 100.100.102.221/32
    mtu 9000

auto enp4s1
iface enp4s1 inet static
    address 100.100.102.221/32
    mtu 9000

auto vmbr1
iface vmbr1 inet manual
    bridge-ports envxlan1
    bridge-stp off
    bridge-fd 0

auto envxlan1
iface envxlan1
    vxlan-id 1
```

```
    vxlan-learning no
```

pve2

/etc/network/interfaces

```
auto lo
iface lo inet loopback

auto enp3s0
iface enp3s0 inet static
    address 100.100.101.222/32
    mtu 9000

auto enp3s1
iface enp3s1 inet static
    address 100.100.101.222/32
    mtu 9000

auto enp4s0
iface enp4s0 inet static
    address 100.100.102.222/32
    mtu 9000

auto enp4s1
iface enp4s1 inet static
    address 100.100.102.222/32
    mtu 9000

auto vmbr1
iface vmbr1 inet manual
    bridge-ports envxlan1
    bridge-stp off
    bridge-fd 0

auto envxlan1
iface envxlan1
    vxlan-id 1
    vxlan-learning no
```

pve3

/etc/network/interfaces

```
auto lo
```

```
iface lo inet loopback

auto enp3s0
iface enp3s0 inet static
    address 100.100.101.223/32
    mtu 9000

auto enp3s1
iface enp3s1 inet static
    address 100.100.101.223/32
    mtu 9000

auto enp4s0
iface enp4s0 inet static
    address 100.100.102.223/32
    mtu 9000

auto enp4s1
iface enp4s1 inet static
    address 100.100.102.223/32
    mtu 9000

auto vmbr1
iface vmbr1 inet manual
    bridge-ports envxlan1
    bridge-stp off
    bridge-fd 0

auto envxlan1
iface envxlan1
    vxlan-id 1
    vxlan-learning no
```

puis on recharge sur chaque noeud la configuration réseau :

```
# ifreload -a
```

Maintenant, il va falloir faire des manipulations sur chaque nœud avec certaines adaptations. Voici :

pve1

Entrez en ligne de commande **frr** :

```
# vtysh
```

ensuite entrez en mode configuration :

```
pve1# conf t
```

On va définir les routes statiques afin d'épargner un peu la découverte ARP inutile :

```
pve1(config)# ip route 100.100.101.223/32 enp3s0
pve1(config)# ip route 100.100.101.222/32 enp3s1
pve1(config)# ip route 100.100.102.223/32 enp4s0
pve1(config)# ip route 100.100.102.222/32 enp4s1
```

On va ensuite ajouter les IP a l'interface loopback :

```
pve1(config)# interface lo
pve1(config-if)# ip address 10.43.0.221/32
pve1(config-if)# ip address 10.50.0.221/32
pve1(config-if)# exit
```

on va d'abord configurer les **access list** :

```
pve1(config)# access-list data_net permit 10.50.0.0/24
pve1(config)# access-list vm_net permit 10.43.0.0/24
```

Puis les **route-map** pour la priorité des flux en fonction[3] :

```
pve1(config)# route-map data_int permit 10
pve1(config-route-map)# match ip address data_net
pve1(config-route-map)# set local-preference 110
pve1(config-route-map)# exit
pve1(config)# route-map data_int permit 11
pve1(config-route-map)# match ip address vm_net
pve1(config-route-map)# set local-preference 100
pve1(config-route-map)# exit
```

```
pve1(config)# route-map vm_int permit 10
pve1(config-route-map)# match ip address vm_net
pve1(config-route-map)# set local-preference 110
pve1(config-route-map)# exit
pve1(config)# route-map vm_int permit 11
pve1(config-route-map)# match ip address data_net
pve1(config-route-map)# set local-preference 100
pve1(config-route-map)# exit
```

```
pve1(config)# route-map bgp_src_addr permit 10
pve1(config-route-map)# match ip address data_net
pve1(config-route-map)# set src 10.50.0.221
pve1(config-route-map)# exit
pve1(config)# route-map bgp_src_addr permit 11
pve1(config-route-map)# match ip address vm_net
pve1(config-route-map)# set src 10.43.0.221
pve1(config-route-map)# exit
pve1(config)# ip protocol bgp route-map bgp_src_addr
```

ensuite on va dans la configuration **bgp**

```
pve1(config)# router bgp 65001
```

on applique les configurations usuel:

```
pve1(config-router)# bgp router-id 10.43.0.221
pve1(config-router)# no bgp default ipv4-unicast
```

Ensuite on créer nos peer-group avec leur configurations:

- **pg-ipv4-data** - pour les peers du réseau DATA

  ```
  pve1(config-router)# neighbor pg-ipv4-data peer-group
  pve1(config-router)# neighbor pg-ipv4-data timers 5 15
  pve1(config-router)# neighbor pg-ipv4-data disable-connected-check
  ```

- **pg-ipv4-vm** - pour les peers du réseau des VMs

  ```
  pve1(config-router)# neighbor pg-ipv4-vm peer-group
  pve1(config-router)# neighbor pg-ipv4-vm timers 5 15
  pve1(config-router)# neighbor pg-ipv4-vm disable-connected-check
  ```

- **pg-evpn** - pour la partie BGP EVPN

  ```
  pve1(config-router)# neighbor pg-evpn peer-group
  pve1(config-router)# neighbor pg-evpn timers 5 15
  pve1(config-router)# neighbor pg-evpn ebgp-multihop 255
  pve1(config-router)# neighbor pg-evpn update-source 10.43.0.221
  ```

Une fois fait on configure le routage IPv4 :

```
pve1(config-router)# address-family ipv4 unicast
pve1(config-router-af)# neighbor pg-ipv4-data activate
pve1(config-router-af)# neighbor pg-ipv4-vm activate
pve1(config-router-af)# neighbor pg-ipv4-data route-map data_int in
pve1(config-router-af)# neighbor pg-ipv4-vm route-map vm_int in
pve1(config-router-af)# network 10.43.0.221/32
pve1(config-router-af)# network 10.50.0.221/32
pve1(config-router-af)# exit
```

Puis la configuration EVPN :

```
pve1(config-router)# address-family l2vpn evpn
pve1(config-router-af)# neighbor pg-evpn activate
pve1(config-router-af)# advertise-all-vni
pve1(config-router-af)# advertise-default-gw
pve1(config-router-af)# exit
```

Ensuite on configure les neighbors :

```
pve1(config-router)# neighbor 100.100.101.222 remote-as 65002
pve1(config-router)# neighbor 100.100.101.222 peer-group pg-ipv4-vm
pve1(config-router)# neighbor 100.100.102.222 remote-as 65002
pve1(config-router)# neighbor 100.100.102.222 peer-group pg-ipv4-data
pve1(config-router)# neighbor 100.100.101.223 remote-as 65003
pve1(config-router)# neighbor 100.100.101.223 peer-group pg-ipv4-vm
pve1(config-router)# neighbor 100.100.102.223 remote-as 65003
pve1(config-router)# neighbor 100.100.102.223 peer-group pg-ipv4-data
pve1(config-router)# neighbor 10.43.0.222 remote-as 65002
pve1(config-router)# neighbor 10.43.0.222 peer-group pg-evpn
pve1(config-router)# neighbor 10.43.0.223 remote-as 65003
pve1(config-router)# neighbor 10.43.0.223 peer-group pg-evpn
```

Ensuite on clear les sessions BGP :

```
pve1(config-router-af)# end
pve1# clear bgp ipv4 *
pve1# clear bgp l2vpn evpn *
```

Après avoir attendu quelques secondes on peux vérifier si les sessions BGP sont actives :

```
pve1# sh bgp summary

IPv4 Unicast Summary:
BGP router identifier 10.30.0.221, local AS number 65001 vrf-id 0
BGP table version 16
RIB entries 11, using 1760 bytes of memory
Peers 4, using 83 KiB of memory
Peer groups 3, using 192 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
100.100.102.222 4      65002     154     158        0    0    0 00:03:28
4
100.100.102.223 4      65003     157     158        0    0    0 00:03:28
4
100.100.101.222 4      65002     157     155        0    0    0 00:03:32
4
100.100.101.223 4      65003     159     155        0    0    0 00:03:32
4

Total number of neighbors 4

L2VPN EVPN Summary:
BGP router identifier 10.30.0.221, local AS number 65001 vrf-id 0
BGP table version 0
RIB entries 5, using 800 bytes of memory
Peers 2, using 41 KiB of memory
Peer groups 3, using 192 bytes of memory
```

```
Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
10.43.0.222     4      65002      47      50        0    0    0 00:03:02
4
10.43.0.223     4      65003      51      55        0    0    0 00:03:30
4

Total number of neighbors 2
```

Une fois validé, on enregistre et on quitte :

```
pve1# write memory
pve1# exit
```

pve2

Entrez en ligne de commande **frr** :

```
# vtysh
```

ensuite entrez en mode configuration :

```
pve2# conf t
```

On va définir les routes statiques afin d'epargner un peu la découverte ARP inutile :

```
pve2(config)# ip route 100.100.101.221/32 enp3s0
pve2(config)# ip route 100.100.101.223/32 enp3s1
pve2(config)# ip route 100.100.102.221/32 enp4s0
pve2(config)# ip route 100.100.102.223/32 enp4s1
```

On va ensuite ajouter les IP a l'interface loopback :

```
pve2(config)# interface lo
pve2(config-if)# ip address 10.43.0.222/32
pve2(config-if)# ip address 10.50.0.222/32
pve2(config-if)# exit
```

on va d'abord configurer les **access list** :

```
pve2(config)# access-list data_net permit 10.50.0.0/24
pve2(config)# access-list vm_net permit 10.43.0.0/24
```

Puis les **route-map** pour la priorité des flux en fonction[4] :

```
pve2(config)# route-map data_int permit 10
pve2(config-route-map)# match ip address data_net
pve2(config-route-map)# set local-preference 110
```

```
pve2(config-route-map)# exit
pve2(config)# route-map data_int permit 11
pve2(config-route-map)# match ip address vm_net
pve2(config-route-map)# set local-preference 100
pve2(config-route-map)# exit
```

```
pve2(config)# route-map vm_int permit 10
pve2(config-route-map)# match ip address vm_net
pve2(config-route-map)# set local-preference 110
pve2(config-route-map)# exit
pve2(config)# route-map vm_int permit 11
pve2(config-route-map)# match ip address data_net
pve2(config-route-map)# set local-preference 100
pve2(config-route-map)# exit
```

```
pve2(config)# route-map bgp_src_addr permit 10
pve2(config-route-map)# match ip address data_net
pve2(config-route-map)# set src 10.50.0.222
pve2(config-route-map)# exit
pve2(config)# route-map bgp_src_addr permit 11
pve2(config-route-map)# match ip address vm_net
pve2(config-route-map)# set src 10.43.0.222
pve2(config-route-map)# exit
pve2(config)# ip protocol bgp route-map bgp_src_addr
```

ensuite on va dans la configuration **bgp**

```
pve2(config)# router bgp 65002
```

on applique les configurations usuel:

```
pve2(config-router)# bgp router-id 10.43.0.222
pve2(config-router)# no bgp default ipv4-unicast
```

Ensuite on créer nos peer-group avec leur configurations:

- **pg-ipv4-data** - pour les peers du réseau DATA

```
pve2(config-router)# neighbor pg-ipv4-data peer-group
pve2(config-router)# neighbor pg-ipv4-data timers 5 15
pve2(config-router)# neighbor pg-ipv4-data disable-connected-check
```

- **pg-ipv4-vm** - pour les peers du réseau des VMs

```
pve2(config-router)# neighbor pg-ipv4-vm peer-group
pve2(config-router)# neighbor pg-ipv4-vm timers 5 15
pve2(config-router)# neighbor pg-ipv4-vm disable-connected-check
```

- **pg-evpn** - pour la partie BGP EVPN

```
pve2(config-router)# neighbor pg-evpn peer-group
```

```
pve2(config-router)# neighbor pg-evpn timers 5 15
pve2(config-router)# neighbor pg-evpn ebgp-multihop 255
pve2(config-router)# neighbor pg-evpn update-source 10.43.0.222
```

Une fois fait on configure le routage IPv4 :

```
pve2(config-router)# address-family ipv4 unicast
pve2(config-router-af)# neighbor pg-ipv4-data activate
pve2(config-router-af)# neighbor pg-ipv4-vm activate
pve2(config-router-af)# neighbor pg-ipv4-data route-map data_int in
pve2(config-router-af)# neighbor pg-ipv4-vm route-map vm_int in
pve2(config-router-af)# network 10.43.0.222/32
pve2(config-router-af)# network 10.50.0.222/32
pve2(config-router-af)# exit
```

Puis la configuration EVPN :

```
pve2(config-router)# address-family l2vpn evpn
pve2(config-router-af)# neighbor pg-evpn activate
pve2(config-router-af)# advertise-all-vni
pve2(config-router-af)# advertise-default-gw
pve2(config-router-af)# exit
```

Ensuite on configure les neighbors :

```
pve2(config-router)# neighbor 100.100.101.221 remote-as 65001
pve2(config-router)# neighbor 100.100.101.221 peer-group pg-ipv4-vm
pve2(config-router)# neighbor 100.100.102.221 remote-as 65001
pve2(config-router)# neighbor 100.100.102.221 peer-group pg-ipv4-data
pve2(config-router)# neighbor 100.100.101.223 remote-as 65003
pve2(config-router)# neighbor 100.100.101.223 peer-group pg-ipv4-vm
pve2(config-router)# neighbor 100.100.102.223 remote-as 65003
pve2(config-router)# neighbor 100.100.102.223 peer-group pg-ipv4-data
pve2(config-router)# neighbor 10.43.0.221 remote-as 65001
pve2(config-router)# neighbor 10.43.0.221 peer-group pg-evpn
pve2(config-router)# neighbor 10.43.0.223 remote-as 65003
pve2(config-router)# neighbor 10.43.0.223 peer-group pg-evpn
```

Ensuite on clear les sessions BGP :

```
pve2(config-router-af)# end
pve2# clear bgp ipv4 *
pve2# clear bgp l2vpn evpn *
```

Après avoir attendu quelques secondes on peux vérifier si les sessions BGP sont actives :

```
pve2# sh bgp summary

IPv4 Unicast Summary:
BGP router identifier 10.30.0.222, local AS number 65002 vrf-id 0
```

```
BGP table version 14
RIB entries 11, using 1760 bytes of memory
Peers 4, using 83 KiB of memory
Peer groups 3, using 192 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
100.100.102.221 4      65001     281     281        0    0    0 00:13:51
4
100.100.102.223 4      65003     282     281        0    0    0 00:13:51
4
100.100.101.221 4      65001     279     281        0    0    0 00:13:55
4
100.100.101.223 4      65003     281     281        0    0    0 00:13:55
4

Total number of neighbors 4

L2VPN EVPN Summary:
BGP router identifier 10.30.0.222, local AS number 65002 vrf-id 0
BGP table version 0
RIB entries 5, using 800 bytes of memory
Peers 2, using 41 KiB of memory
Peer groups 3, using 192 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
10.43.0.221     4      65001     171     173        0    0    0 00:13:25
4
10.43.0.223     4      65003     173     177        0    0    0 00:13:42
4

Total number of neighbors 2
```

Une fois validé, on enregistre et on quitte :

```
pve2# write memory
pve2# exit
```

pve3

Entrez en ligne de commande **frr** :

```
# vtysh
```

ensuite entrez en mode configuration :

```
pve3# conf t
```

On va définir les routes statiques afin d'epargner un peu la découverte ARP inutile :

```
pve3(config)# ip route 100.100.101.221/32 enp3s0
pve3(config)# ip route 100.100.101.222/32 enp3s1
pve3(config)# ip route 100.100.102.221/32 enp4s0
pve3(config)# ip route 100.100.102.222/32 enp4s1
```

On va ensuite ajouter les IP a l'interface loopback :

```
pve3(config)# interface lo
pve3(config-if)# ip address 10.43.0.223/32
pve3(config-if)# ip address 10.50.0.223/32
pve3(config-if)# exit
```

on va d'abord configurer les **access list** :

```
pve3(config)# access-list data_net permit 10.50.0.0/24
pve3(config)# access-list vm_net permit 10.43.0.0/24
```

Puis les **route-map** pour la priorité des flux en fonction[5] :

```
pve3(config)# route-map data_int permit 10
pve3(config-route-map)# match ip address data_net
pve3(config-route-map)# set local-preference 110
pve3(config-route-map)# exit
pve3(config)# route-map data_int permit 11
pve3(config-route-map)# match ip address vm_net
pve3(config-route-map)# set local-preference 100
pve3(config-route-map)# exit
```

```
pve3(config)# route-map vm_int permit 10
pve3(config-route-map)# match ip address vm_net
pve3(config-route-map)# set local-preference 110
pve3(config-route-map)# exit
pve3(config)# route-map vm_int permit 11
pve3(config-route-map)# match ip address data_net
pve3(config-route-map)# set local-preference 100
pve3(config-route-map)# exit
```

```
pve3(config)# route-map bgp_src_addr permit 10
pve3(config-route-map)# match ip address data_net
pve3(config-route-map)# set src 10.50.0.223
pve3(config-route-map)# exit
pve3(config)# route-map bgp_src_addr permit 11
pve3(config-route-map)# match ip address vm_net
pve3(config-route-map)# set src 10.43.0.223
pve3(config-route-map)# exit
pve3(config)# ip protocol bgp route-map bgp_src_addr
```

ensuite on va dans la configuration **bgp**

```
pve3(config)# router bgp 65003
```

on applique les configurations usuel:

```
pve3(config-router)# bgp router-id 10.43.0.223
pve3(config-router)# no bgp default ipv4-unicast
```

Ensuite on créer nos peer-group avec leur configurations:

- **pg-ipv4-data** - pour les peers du réseau DATA

```
pve3(config-router)# neighbor pg-ipv4-data peer-group
pve3(config-router)# neighbor pg-ipv4-data timers 5 15
pve3(config-router)# neighbor pg-ipv4-data disable-connected-check
```

- **pg-ipv4-vm** - pour les peers du réseau des VMs

```
pve3(config-router)# neighbor pg-ipv4-vm peer-group
pve3(config-router)# neighbor pg-ipv4-vm timers 5 15
pve3(config-router)# neighbor pg-ipv4-vm disable-connected-check
```

- **pg-evpn** - pour la partie BGP EVPN

```
pve3(config-router)# neighbor pg-evpn peer-group
pve3(config-router)# neighbor pg-evpn timers 5 15
pve3(config-router)# neighbor pg-evpn ebgp-multihop 255
pve3(config-router)# neighbor pg-evpn update-source 10.43.0.223
```

Une fois fait on configure le routage IPv4 :

```
pve3(config-router)# address-family ipv4 unicast
pve3(config-router-af)# neighbor pg-ipv4-data activate
pve3(config-router-af)# neighbor pg-ipv4-vm activate
pve3(config-router-af)# neighbor pg-ipv4-data route-map data_int in
pve3(config-router-af)# neighbor pg-ipv4-vm route-map vm_int in
pve3(config-router-af)# network 10.43.0.223/32
pve3(config-router-af)# network 10.50.0.223/32
pve3(config-router-af)# exit
```

Puis la configuration EVPN :

```
pve3(config-router)# address-family l2vpn evpn
pve3(config-router-af)# neighbor pg-evpn activate
pve3(config-router-af)# advertise-all-vni
pve3(config-router-af)# advertise-default-gw
pve3(config-router-af)# exit
```

Ensuite on configure les neighbors :

```
pve3(config-router)# neighbor 100.100.101.221 remote-as 65001
```

```
pve3(config-router)# neighbor 100.100.101.221 peer-group pg-ipv4-vm
pve3(config-router)# neighbor 100.100.102.221 remote-as 65001
pve3(config-router)# neighbor 100.100.102.221 peer-group pg-ipv4-data
pve3(config-router)# neighbor 100.100.101.222 remote-as 65002
pve3(config-router)# neighbor 100.100.101.222 peer-group pg-ipv4-vm
pve3(config-router)# neighbor 100.100.102.222 remote-as 65002
pve3(config-router)# neighbor 100.100.102.222 peer-group pg-ipv4-data
pve3(config-router)# neighbor 10.43.0.221 remote-as 65001
pve3(config-router)# neighbor 10.43.0.221 peer-group pg-evpn
pve3(config-router)# neighbor 10.43.0.222 remote-as 65002
pve3(config-router)# neighbor 10.43.0.222 peer-group pg-evpn
```

Ensuite on clear les sessions BGP :

```
pve3(config-router-af)# end
pve3# clear bgp ipv4 *
pve3# clear bgp l2vpn evpn *
```

Après avoir attendu quelques secondes on peux vérifier si les sessions BGP sont actives :

```
pve3# sh bgp summary

IPv4 Unicast Summary:
BGP router identifier 10.30.0.223, local AS number 65003 vrf-id 0
BGP table version 15
RIB entries 11, using 1760 bytes of memory
Peers 4, using 83 KiB of memory
Peer groups 3, using 192 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
100.100.102.221 4      65001     156     159        0    0    0 00:03:29
4
100.100.102.222 4      65002     154     159        0    0    0 00:03:29
4
100.100.101.221 4      65001     155     159        0    0    0 00:03:33
4
100.100.101.222 4      65002     155     159        0    0    0 00:03:33
4

Total number of neighbors 4

L2VPN EVPN Summary:
BGP router identifier 10.30.0.223, local AS number 65003 vrf-id 0
BGP table version 0
RIB entries 5, using 800 bytes of memory
Peers 2, using 41 KiB of memory
Peer groups 3, using 192 bytes of memory

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down
State/PfxRcd
```

```
10.43.0.221     4      65001      53      52        0    0    0 00:03:31
4
10.43.0.222     4      65002      51      51        0    0    0 00:03:21
4

Total number of neighbors 2
```

Une fois validé, on enregistre et on quitte :

```
pve3# write memory
pve3# exit
```

Configuration du Cluster

On va créer le cluster **en CLI**. Pour ça, connectez vous sur le premier noeud, et tapez la commande :

```
# pvecm create pve-switchless
```

une fois fait, il faudra vous connecter sur les deux autres noeuds et taper la commande suivante en vous laissant guider [6]:

```
# pvecm add 10.1.9.221
```

```
Please enter superuser (root) password for '10.1.9.221':
                                                        Password for
root@10.1.9.221: *************
Establishing API connection with host '10.1.9.221'
The authenticity of host '10.1.9.221' can't be established.
X509 SHA256 key fingerprint is
59:28:DC:C3:12:1E:4A:C8:5A:9A:34:52:16:FB:47:C8:30:08:43:29:C9:B0:C8:64:33:4
8:96:46:92:5D:76:61.
Are you sure you want to continue connecting (yes/no)? yes
Login succeeded.
Request addition of this node
Join request OK, finishing setup locally
stopping pve-cluster service
backup old database to '/var/lib/pve-
cluster/backup/config-1570610507.sql.gz'
waiting for quorum...OK
(re)generate node files
generate new node certificate
merge authorized SSH keys and known hosts
generated new node certificate, restart pveproxy and pvedaemon services
successfully added node 'pve2' to cluster.
```

Une fois fait, il vous faudra modifier sur l'un des noeud le fichier **/etc/pve/corosync.conf**, pour y modifier le ring existant, et y ajouter le second ring[7] :

```
logging {
```

```
  debug: off
  to_syslog: yes
}

nodelist {
  node {
    name: pve1
    nodeid: 1
    quorum_votes: 1
    ring0_addr: 100.100.101.221
    ring1_addr: 100.100.102.221
  }
  node {
    name: pve2
    nodeid: 2
    quorum_votes: 1
    ring0_addr: 100.100.101.222
    ring1_addr: 100.100.102.222
  }
  node {
    name: pve3
    nodeid: 3
    quorum_votes: 1
    ring0_addr: 100.100.101.223
    ring1_addr: 100.100.102.223
  }
}

quorum {
  provider: corosync_votequorum
}

totem {
  cluster_name: pve-switchless
  config_version: 4
  interface {
    linknumber: 0
  }
  interface {
    linknumber: 1
  }
  ip_version: ipv4-6
  secauth: on
  version: 2
}
```

Il faut ensuite installer ceph sur chaque nœud **en CLI** :

```
# pveceph install
```

Puis sur l'un des noeud, initialiser le cluster :

```
# pveceph init --network 10.50.0.0/24
```

Il va falloir modifier le fichier **/etc/pve/ceph.conf** pour :

- Retirer les deux lignes suivantes :

```
     cluster_network = 10.50.0.0/24
     public_network = 10.50.0.0/24
```

- Ajouter ceci a la fin du fichier :

```
[mon.pve1]
     host = pve1
     mon_addr = 10.50.0.221
     public_addr = 10.50.0.221
     public_bind_addr = 10.50.0.221

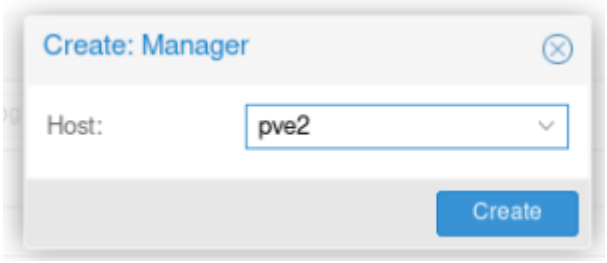
[mon.pve2]
     host = pve2
     mon_addr = 10.50.0.222
     public_addr = 10.50.0.222
     public_bind_addr = 10.50.0.222

[mon.pve3]
     host = pve3
     mon_addr = 10.50.0.223
     public_addr = 10.50.0.223
     public_bind_addr = 10.50.0.223
```

puis initialiser sur chaque noeud, le monitor ceph en y adaptant l'IP [8] :

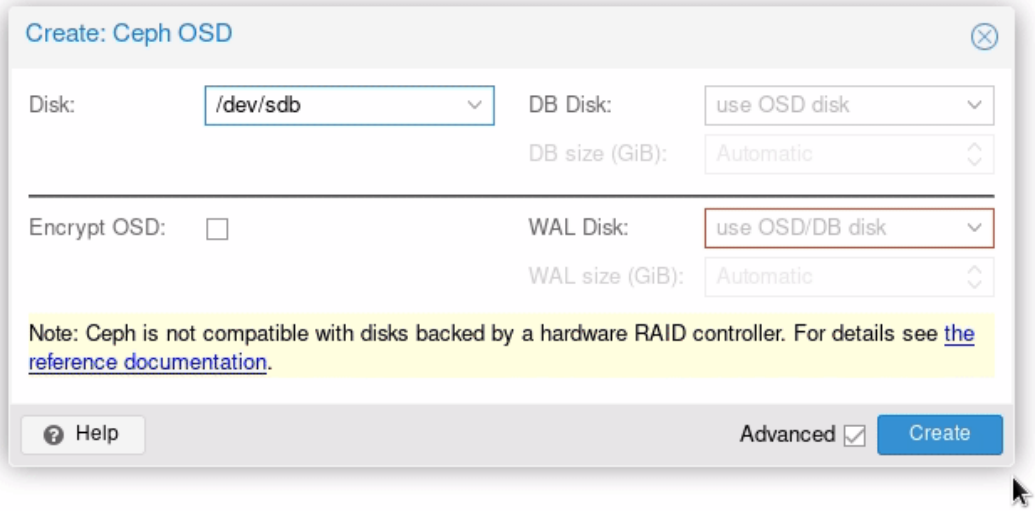
```
# pveceph createmon --mon-address 10.50.0.221
```

Une fois fait, on peut repasser sur l'interface pour ajouter les managers manquant :

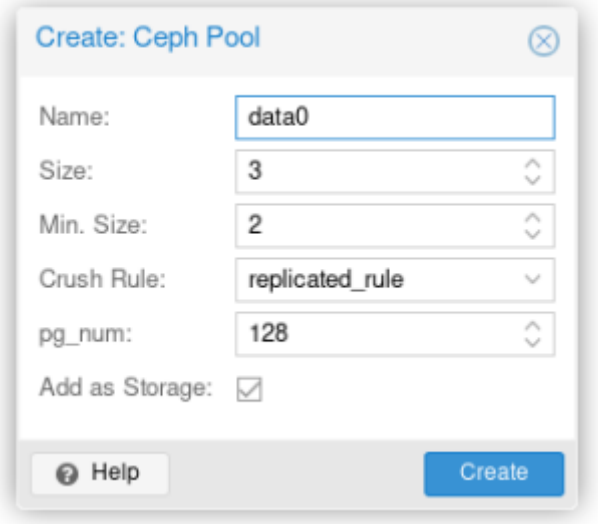


Puis les OSD sur chaque nœud :

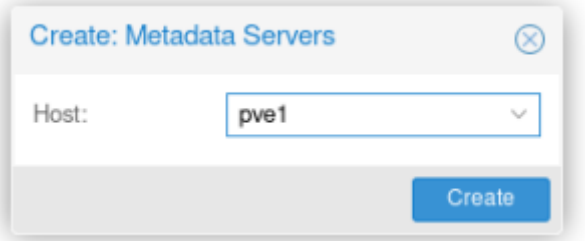
			14.2.4				
hdd	bluestore	up / in	14.2.4	0.01859	1.00	5.27	19.00 GiB



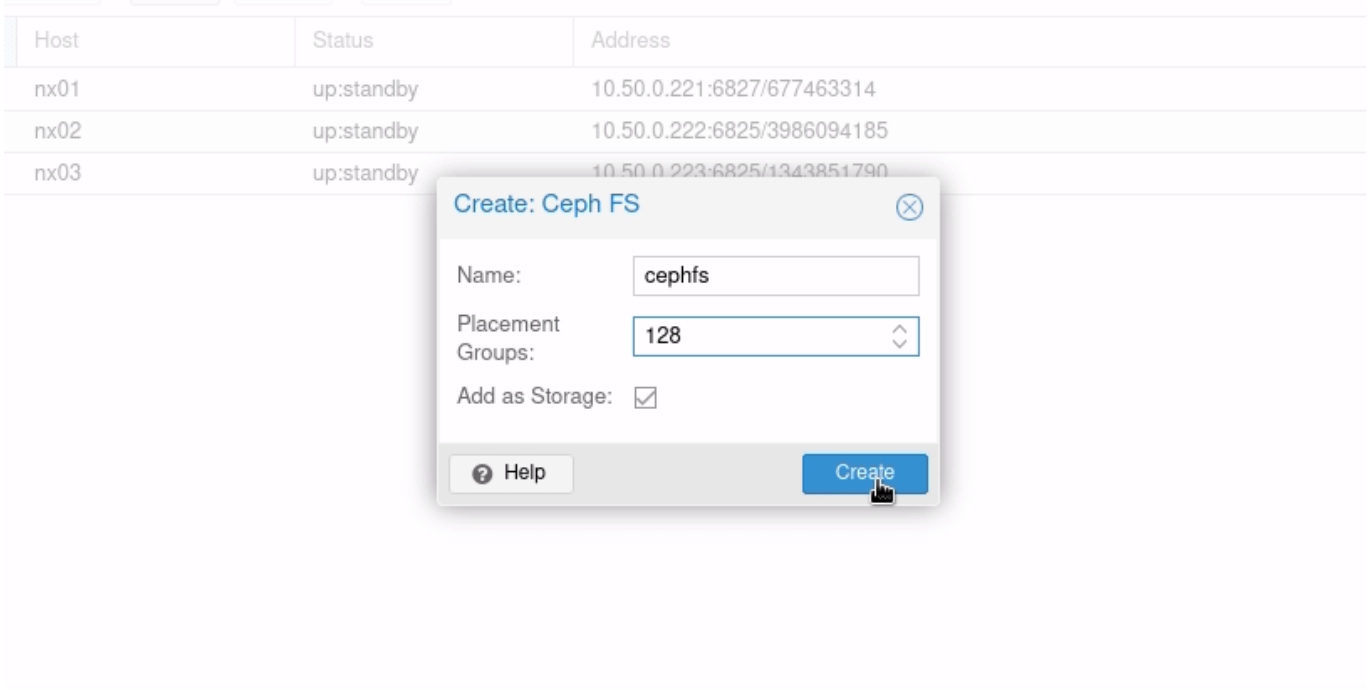
Et pour finir le pool de données :



On peux aussi créer un pool CephFS si nécessaire en créant les MDS :



Puis le pool CephFS :



1)

Attention, toutes les configurations réseau seront déchargées, donc vous allez perdre la main si vous êtes en SSH

2)

Les interfaces vxlan doivent obligatoirement commencer par le préfix “en” et termine par un numéro

3) , 4) , 5)

Pour rappel, la route priorisée est celle avec la **local preference** la plus élevée

6) , 8)

A lancer nœud par nœud, pas en même temps

7)

Il vous faudra incrementer la valeur de config_version

From:
https://wiki.virtit.fr/ - **VirtIT**

Permanent link:
https://wiki.virtit.fr/doku.php/kb:linux:proxmox:hyperconverged-proxmox-2-liens?rev=1581780350

Last update: **2020/02/15 15:25**