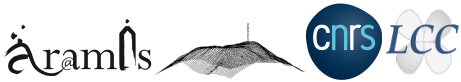
Réseaux, Hyperviseurs, Stockages, relions-les tous !

Jérôme COLOMBET

https://homepages.lcc-toulouse.fr/colombet/journees_proxmox_et_ceph_juin2022.pdf

7 et 8 juin 2022



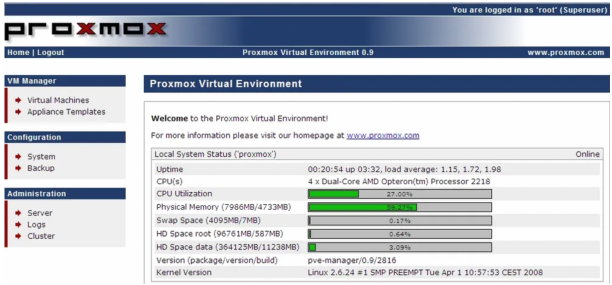
Contexte de cette présentation

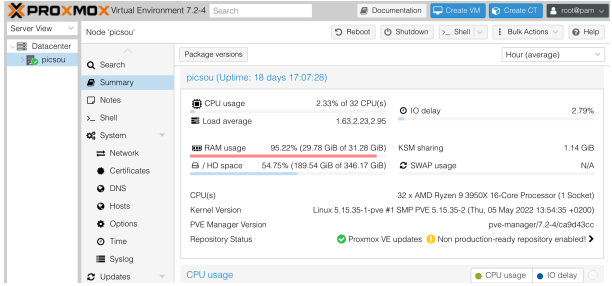
- Proxmox est majoritairement utilisés dans nos structures de l'ESR
- Nos hyperviseurs sont généralement couplés
  - aux stockages centraux (liens dédiés ou mutualisés)
  - aux réseaux d'infrastructures (normés et propriétaires)
  - dans certains cas, aux cluster HPC (ipxe, diskless, ...)
- Mais toutes ces interconnexions :
  - comment construisez-vous ces réseaux ?
    1. en fonction des ressources (stagiaires, budgets, ...) ?
    2. en fonction des projets (campus , recherche, ...) ?
    3. en fonction de vos envies de geek ?
  - vous pensez bien sûr à les documenter (pour vos collègues, un audit, ...)
  - mais de toute façon vous allez les oublier après des années d'utilisation
- Quelques configurations réseaux, de type bridge, mesh, bond et openvswitch

Politiquement - Proxmox

- **L**aboratoire de **C**himie de **C**oordination, **UPR 8241** sur un campus propre CNRS
- Proche de la nouvelle attraction touristique, le Téléo : « Téléphérique Urbain Toulouse »
- Le service RICS, 3 informaticiens pour **300** personnes et un bâtiment de 11000m^2
- Les solutions techniques
  - 3 salles serveurs reparties sur le campus 205
  - Backbone 10G sur l'ensemble des bâtiments, 600 prises
  - 3 cluster HA-PRA de 6 nœuds basé sur Proxmox VE 7
  - 2 cluster HPC via l'ordonnanceur OAR (Centos 6, Debian 10)
  - Multiples stockages ZFS accessibles via NFS, iSCSI et SMB (data, VMs, mails, ...)
- Quelques exemples de solution en place à la fin de présentation ...

Clin d'oeil aux évolutions de Proxmox ...





2008

PROXMOX

Logo Proxmox - Source Wikipedia

2022

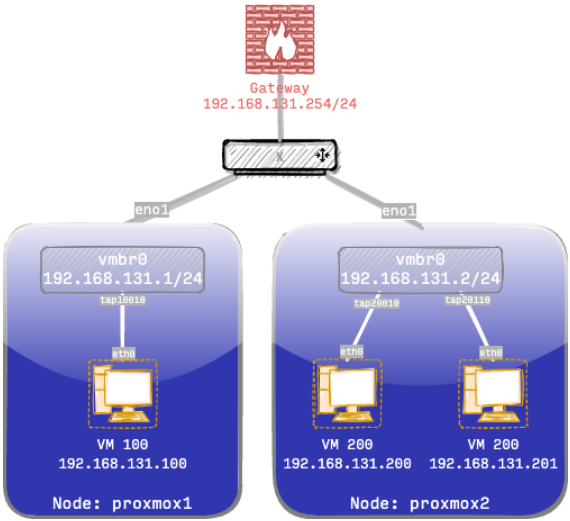
SiarsV2 2018 : Protection infrastructure Linux
Mise en place d'une infrastructure virtualisée

Vocabulaire : nommage réseau

- en**oX**, ens**X** : interface physique qui écoute tout ce qui se passe via systemd
  - **en** – ethernet, **sl** – serial line IP (slip), **wl** – wlan, **ww** – wwan, **ib** – Infiniband
  - **o** – on-board device index number, **s** – hotplug slot index number
- **vlan** : interface virtuelle associée à une interface physique séparée par un point (eno1.50, bond1.30)
- **bond** : agrégation de plusieurs interfaces physiques en une interface logique
- **vmbr** : interface faisant jonction (pont) entre les ethX, vethX, tapX (vmbr0 - vmbr4094)
- **ovs bond, bridge, intport** : identique aux autres mais via le logiciel OpenvSwitch
- **tap** : interface virtuelle pour les machines de type KVM
- **veth** : interface virtuelle pour les conteneurs de type LXC, OpenVZ

Conseil, ne pas utiliser la technique ci dessous pour revenir à l'ancien nommage ensX ⇒ ethX
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"

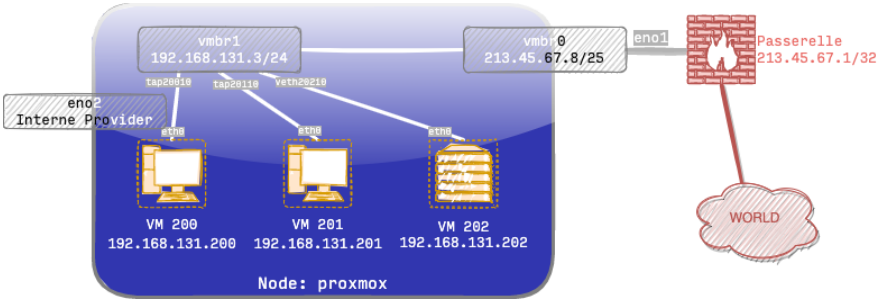
Configuration en mode bridge (par défaut)



more /etc/network/interfaces

```
auto lo
iface lo inet loopback
iface eno1 inet manual

auto vmbr0
iface vmbr0 inet static
    address 192.168.131.1/24
    gateway 192.168.131.254
    bridge_ports eno1 # physical netcard
    bridge_stp off # spanning tree off
```

Configuration en mode routage (typique en VPS)



more /etc/network/interfaces

```
iface vmbr0 inet static
        address 213.45.67.8/25 # provider ip
        gateway 213.45.67.1
        bridge_ports eno1

iface vmbr1 inet static
        address 192.168.131.1
        netmask 255.255.255.0
        bridge_ports none
```

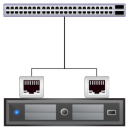
more /etc/network/interfaces

```
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '192.168.131.0/24' -o vmbr0 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '192.168.131.0/24' -o vmbr0 -j MASQUERADE
post-up iptables -t nat -A PREROUTING -i vmbr0 -p tcp –dport 443 -j DNAT –to 192.168.131.200 :443
post-down iptables -t nat -D PREROUTING -i vmbr0 -p tcp –dport 443 -j DNAT –to 192.168.131.200 :443
post-up iptables -t nat -A PREROUTING -i vmbr0 -p tcp –dport 25 -j DNAT –to 192.168.131.201 :25
post-down iptables -t nat -D PREROUTING -i vmbr0 -p tcp –dport 25 -j DNAT –to 192.168.131.201 :25
post-up iptables -t nat -A PREROUTING -i vmbr0 -p tcp –dport 143 -j DNAT –to 192.168.131.202 :143
post-down iptables -t nat -D PREROUTING -i vmbr0 -p tcp –dport 143 -j DNAT –to 192.168.131.202 :143
```

Rappel des modes de bonding

L'agrégation de lien regroupe plusieurs interfaces physiques sous une même interface virtuelle :

- **Mode0** : Round Robin ou équilibrage de charge, la transmission des paquets se fait de façon séquentielle sur chacune des cartes actives dans l'agrégat. Ce mode augmente la bande passante et gère la tolérance de panne
- **Mode1** : Active ou Passive, ce mode gère uniquement la tolérance de panne. Si une des interfaces est désactivée, une autre du pool prend le relais
- **Mode2** : Balance XOR, une interface est affectée à l'envoi vers une même adresse MAC
- **Mode3** : Broadcast, tout le trafic est envoyé par toutes les interfaces



Bonding - Source Dell

- **Mode4** : LACP ou norme IEEE 802.3ad, toutes les interfaces du groupe sont agrégées de façon dynamique, ce qui augmente la bande passante et gère la tolérance de panne. Le commutateur doit gérer la norme 802.ad et les interfaces doivent être compatibles mii-tool / ethtool.
- **Mode5** : balance-tlb pour transmit load balancing : seule la bande passante en sortie est load balancée selon la charge calculée en fonction de la vitesse, ceci pour chaque interface. Alors, le flux entrant est affecté à l'interface courante. Si celle-ci devient inactive, une autre prend alors l'adresse MAC et devient l'interface courante.
- **Mode6** : balance-alb pour adaptive load balancing, ce mode inclut en plus du tlb un load balancing sur le flux entrant et seulement pour un trafic.

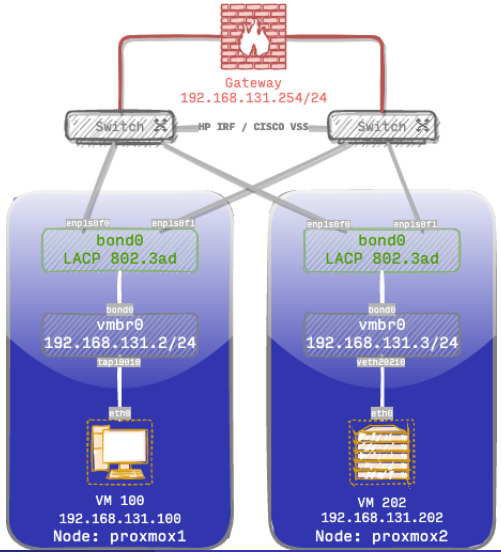
Configuration en mode Linux Bonding LACP

more /etc/network/interfaces

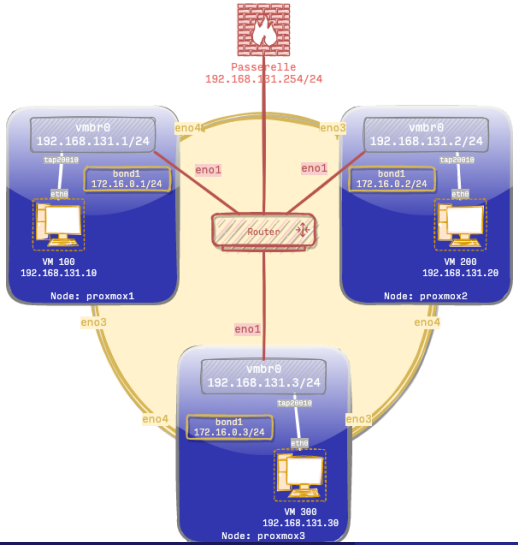
```
iface enp1s0f0 inet manual
iface enp1s0f1 inet manual

auto bond0
iface bond0 inet manual
        bond-slaves enp1s0f0 enp1s0f1
        bond-mode 802.3ad

auto vmbr0
iface vmbr0 inet static
        address 192.168.131.1/24
        gateway 192.168.131.254
        bridge-ports bond0
```

more /proc/net/bonding/bond0

```
Ethernet Channel Bonding Driver : v5.15.35-1-pve
Bonding Mode : IEEE 802.3ad Dynamic link aggregation
Transmit Hash Policy : layer3+4 (1)
MII Status : up
802.3ad info LACP active : on LACP rate : slow Min links : 0
```



Configuration en mode mesh (réseau du pauvre)



3 méthodes pour un réseau mesh (RSTP, Routed, Broadcast)

Exemple en mode Broadcast

```
iface eno1 inet manual
iface eno3 inet manual
iface eno4 inet manual

auto bond1
iface bond1 inet static
    address 172.16.0.1/24
    slaves eno3 eno4
    bond_mode broadcast
auto vmbr0
iface vmbr0 inet static
    address 192.168.131.1/24
    gateway 192.168.1.254
    bridge_ports eno1
```

⇒ Attention si vous utilisez un STP propriétaire de type Juniper, Cisco, HP pensé à vérifier la compatibilité et la cohérence du protocole.

Mise en cluster de 3 nœuds Proxmox via le réseau MESH

Synchroniser le temps et nommer vos nœuds

```
# timedatectl
# more /etc/hosts
172.16.0.1 proxmox1
172.16.0.2 proxmox2
172.16.0.3 proxmox3
```

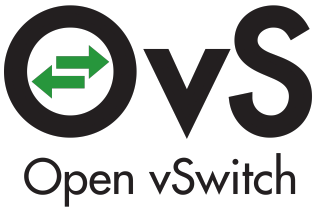
Sur proxmox1 créer et nommer le cluster

```
# pvecm create starwars –link0 172.16.0.1,priority=20 –link1 192.168.131.1,priority=15
```

Intégrer les 2 autres nœuds

```
# pvecm add proxmox1 –link0 172.16.0.x,priority=20 –link1 192.168.131.x,priority=15
```

OpenvSwitch - définition [1]



Logo OpenvSwitch - Source Wikipedia

OpenvSwitch, c'est un commutateur :

- virtuel logiciel multicouche open source (Apache2)
- qui travail au niveau 2 OSI et 3 via iptable
- disponible sous BSD, Linux, Windows
- idéal pour les environnements VMs
- majeure partie du code écrit en C

⇒ *pour résumer*, une de ses fonctions principales est de créer des ports pour le système comme des interfaces réseau ou bien de lui attribuer une interface réseau réelle comme port.

1. `https://docs.openvswitch.org/en/latest/intro/what-is-ovs/`

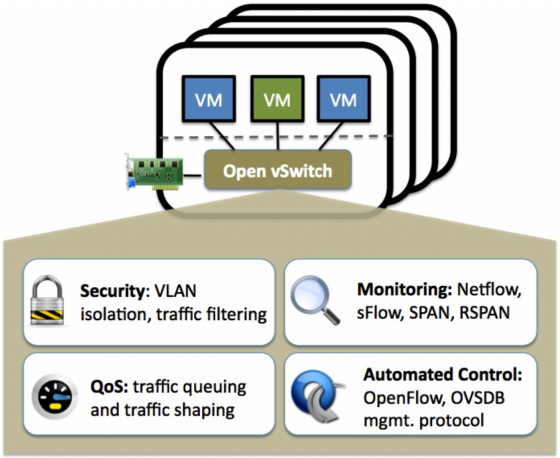
OpenvSwitch - fonctionnalités

OpenvSwitch supporte les fonctions suivantes :

- Standard 802.1Q Vlan (Ports Trunk & Access)
- Interconnexion en bonding avec/sans LACP
- QoS avec des règles type traffic shaping
- Sondes de monitoring (Netflow, sFlow, ...)

Sur Proxmox VE 7.2-4

```
# ovs-vswitchd –version
ovs-vswitchd (Open vSwitch) 2.15.0
```



Fonctionnalités OVS - Source openvswitch.org

OpenvSwitch - les commandes principales

Ajouter un bridge

```
ovs-vsctl add-br vmbrX
```

Supprimer un bridge

```
ovs-vsctl del-br vmbrX
```

Ajouter un port

```
ovs-vsctl add-port vmbrX enoX tag=X
```

Supprimer un port

```
ovs-vsctl del-port vmbrX enoX
```

Modifier un port pour en faire un trunk

```
ovs-vsctl set port enoX trunks=3,4,5
```

Supprimer un des VLANs du trunk

```
ovs-vsctl set port enoX trunks=3,4
```

Supprimer un VLAN d'un port

```
ovs-vsctl remove port enoX tag X
```

Afficher un récapitulatif du bridge :

```
ovs-vsctl show
 bridge vmbr0
    port vmbr0
            interface vmbr0
            type : internal
    port bond0
            interface eno1
            interface eno2
    port veth100i0
            tag : 4
            interface veth100i0
    port veth200i0
            tag : 3
            interface veth200i0
 ovs_version : "2.15.0"
```

OpenvSwitch - remplacer son commutateur physique

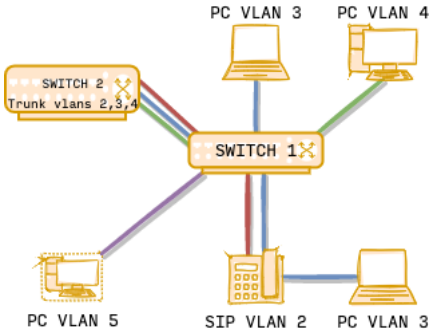


Figure – Réseau physique avec 4 vlans

Pour créer un commutateur OpenvSwitch comme le dessin ci-contre, nous utiliserons 5 cartes réseau physiques du serveur et la série de commandes suivantes :

Switch virtuel

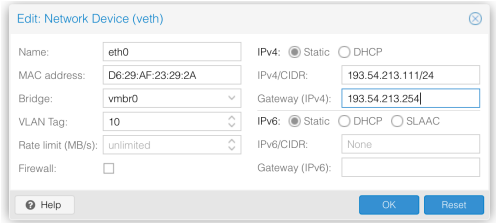
```
ovs-vsctl add-br vmbr0
ovs-vsctl add-port vmbr0 eno1 tag=3
ovs-vsctl add-port vmbr0 eno2 tag=3
ovs-vsctl add-port vmbr0 eno3 tag=4
ovs-vsctl add-port vmbr0 eno4 tag=5
ovs-vsctl add-port vmbr0 eno5 trunks=2,3,4
```

Les PC n'ayant accès qu'à un seul vlan sont en mode access. Idéal sur des mini pc avec un firewall applicatif.

OpenvSwitch - Proxmox couplé à un switch HP

Syntaxe HP 5xxx

HP : interface Bridge-Aggregation 1
HP-Bridge-Aggregation1 : link-aggregation mode dynamic
HP : interface GigabitEthernet 1/0/1
HP-GigabitEthernet1/0/1 : port link-aggregation group 1
HP : interface GigabitEthernet 2/0/1
HP-GigabitEthernet2/0/1 : port link-aggregation group 1
HP : interface Bridge-Aggregation1
HP-Bridge-Aggregation1 : port link-type trunk
HP-Bridge-Aggregation1 : port trunk permit vlan 2-4094



Name ↑	Type	Active	Autostart	Ports/Slaves	Bond Mode	CIDR	Gateway
bond0	OVS Bond	Yes	Yes	eno1 eno2	LACP (balance-tcp)		
bond1	Linux Bond	Yes	Yes	eno3 eno4	broadcast	172.16.0.3/24	
eno1	Network Device	Yes	Yes				
eno2	Network Device	Yes	Yes				
eno3	Network Device	Yes	Yes				
eno4	Network Device	Yes	Yes				
enp7s0f0	Network Device	No	No				
enp7s0f1	Network Device	No	No				
internal	OVS IntPort	Yes	Yes			10.0.0.3/24	10.0.0.254
vmbr0	OVS Bridge	Yes	Yes	bond0 internal			

OpenvSwitch - sFlow [2]

Ajouter une sonde sFlow

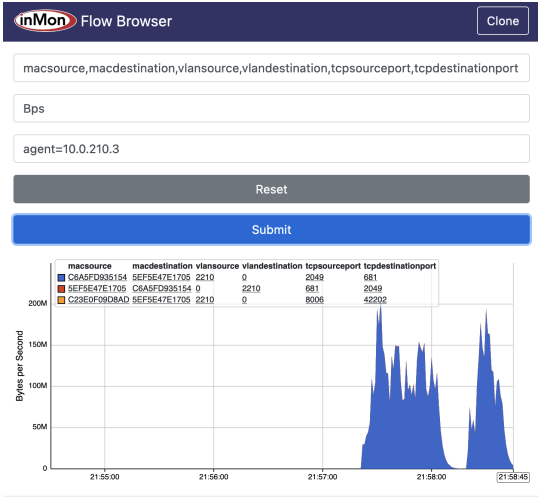
```
# ovs-vsctl – –id=@s create sFlow agent=eth0
target="192.168.131.141 :6343" header=128
sampling=2000 polling=30 – set Bridge vmbr0 sflow=@s
```

Afficher le détail d'une sonde sFlow

```
# ovs-vsctl list sflow
uuid : 774d1bd9-a4d4-4c87-84b6-19c0203daaaa
agent : eth0 header : 128 polling : 30 sampling : 2000
targets : ["192.168.131.141 :6343"]
```

Supprimer une sonde sFlow

```
# ovs-vsctl – clear Bridge vmbr0 sflow
```



2. https://sflow-rt.com/download.php#applications

OpenvSwitch - Port mirroring [3]

Les commandes suivantes configurent le bridge vmbr0 avec tap100i0 et tap200i0 comme ports trunk. Le trafic entrant/sortant sur vmbr0 ou tap100i0 est reflété sur tap200i0 et tout le trafic arrivant sur tap200i0 est supprimé.

Activer un mirror de port

```
$ ovs-vsctl add-br vmbr0
$ ovs-vsctl add-port vmbr0 tap100i0
$ ovs-vsctl add-port vmbr0 tap200i0
– –id=@p get port tap100i0
– –id=@m create mirror name=mirror0 select-all=true output-port=@p
– set bridge vmbr0 mirrors=@m
```

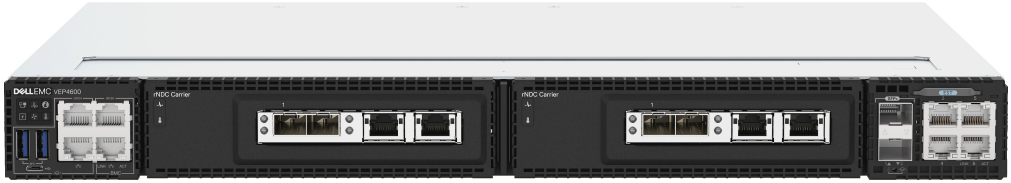
Supprimer le mirror de port

```
$ ovs-vsctl clear bridge vmbr0 mirrors
```

3. https://docs.openvswitch.org/en/latest/faq/configuration/

Pour conclure

- Je n'ai pas parlé de MTU, de IPV6 (tetaneutral.net), et de toutes les options non utilisées
- Quelques surprises après migration de PVE6 à PVE7 (pas de réseau, ifupdown2, ethX...)
- OpenvSwitch du NAC ça roule (PacketFence ou des outils maison)
- OpenvSwitch : Freeradius et Eduroam (isolation des utilisateurs pour des accès ciblés)
- Dell EMC Virtual Edge Platform 4600 est une plate-forme permettant de virtualiser des appliances réseaux type Fortinet, Palo Alto, pFsense dans un chassis réseau



Dell VEP4600 - Source Dell

Questions

Merci de votre attention !

: jerome.colombet@lcc-toulouse.fr
: https://homepages.lcc-toulouse.fr/colombet/
: https://github.com/jeromecolombet
: https://twitter.com/neoclimb