

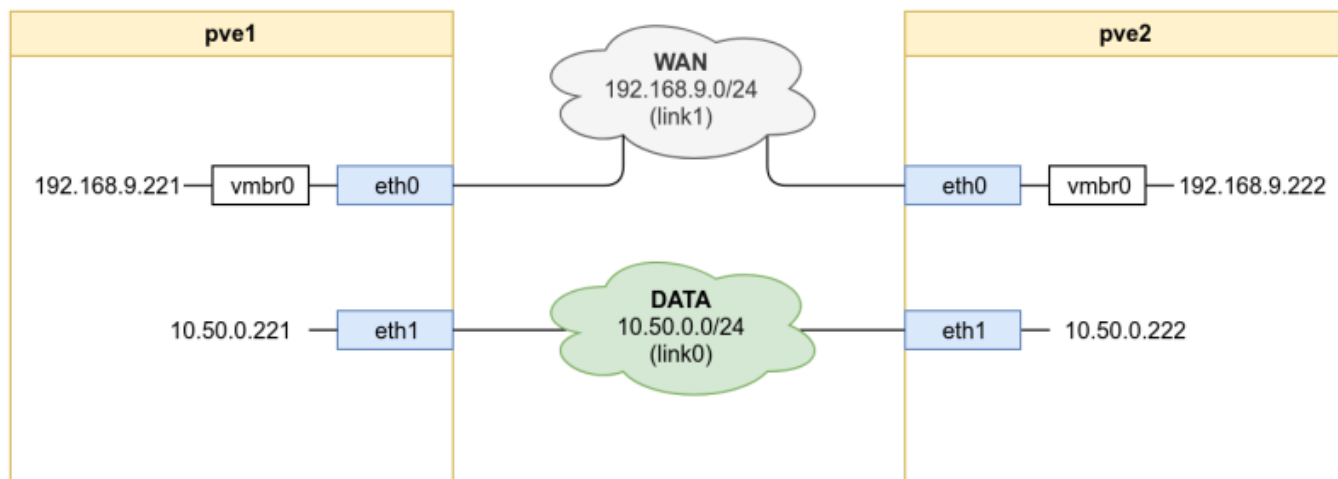
# Cluster deux nœuds Proxmox



Cette solution comporte des risques de [Split-brain](#). Il faut évidemment préférer faire un cluster avec 3 nœuds, ou au pire avec un Qdevice.

L'objectif est de faire un cluster autonome sans SPOF, sans baies de disques hors-de-prix.

Voici le diagramme réseau qui va être utilisé pour ce topic.

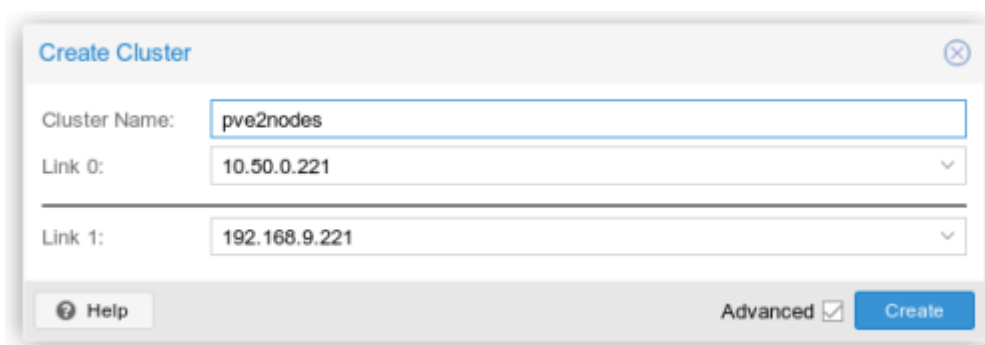


Le disque **/dev/sda** est le disque racine comportant les partitions (LVM) créé par l'ISO proxmox, et le disque **/dev/sdb** est le disque utilisé comme volume DRBD entre les nœuds.

Je pars du principe que les deux serveurs sont configurés et que la configuration réseau est faite.

## Création du cluster Proxmox

Comme pour un cluster "classique", dans l'interface de l'un des nœud on va créer un cluster



Puis avec l'autre nœud, rejoindre le cluster



**Fix Me!**

Une fois fait, il faut modifier la configuration corosync, pour qu'il fonctionne en mode deux nœuds, en ajoutant les lignes **two\_node: 1** et **expected\_votes: 1** dans la section **quorum** du fichier **/etc/pve/corosync.conf** d'un des 2 nœuds, comme ceci :

```
logging {
    debug: off
    to_syslog: yes
}
nodelist {
    node {
        name: pve1
        nodeid: 1
        quorum_votes: 1
        ring0_addr: 10.50.0.221
        ring1_addr: 192.168.9.221
    }
    node {
        name: pve2
        nodeid: 2
        quorum_votes: 1
        ring0_addr: 10.50.0.222
        ring1_addr: 192.168.9.222
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1
    expected_votes: 1
}

totem {
    cluster_name: pve2nodes
    config_version: 2
    interface {
        linknumber: 0
    }
    interface {
        linknumber: 1
    }
    ip_version: ipv4-6
    secauth: on
    version: 2
}
```

et puis restart le daemon corosync sur les deux nœuds :

```
# systemctl restart corosync
```

## Création du volume partagé

Pour que les VM puissent entrer en mode "HA", il faut obligatoirement que le disque de celle-ci soit disponible sur chaque nœud, même si le nœud voisin n'est plus joignable.

Pour cela, on va créer un volume drbd, sur lequel on va placer un volume LVM.

On va donc créer le volume DRBD, en créant le fichier **/etc/drbd.d/pv0.res** :

```
resource pv0 {
    meta-disk internal;
    device /dev/drbd0;
    startup {
        become-primary-on both;
    }
    syncer {
    }
    net {
        allow-two-primaries;
    }
    on pve1 {
        disk /dev/sdb;
        address 10.50.0.221:7798;
    }
    on pve2 {
        disk /dev/sdb;
        address 10.50.0.222:7798;
    }
}
```

On va suite initialiser le volume DRBD :

- Sur les deux nœuds :

```
# drbdadm create-md pv0
```

```
# drbdadm up pv0
```

- puis, sur l'un des deux nœud :

```
# drbdadm -- --overwrite-data-of-peer primary pv0
```

- Vérifier l'état de la synchronisation en tapant sur l'un des deux nœuds :

```
# cat /proc/drbd
```

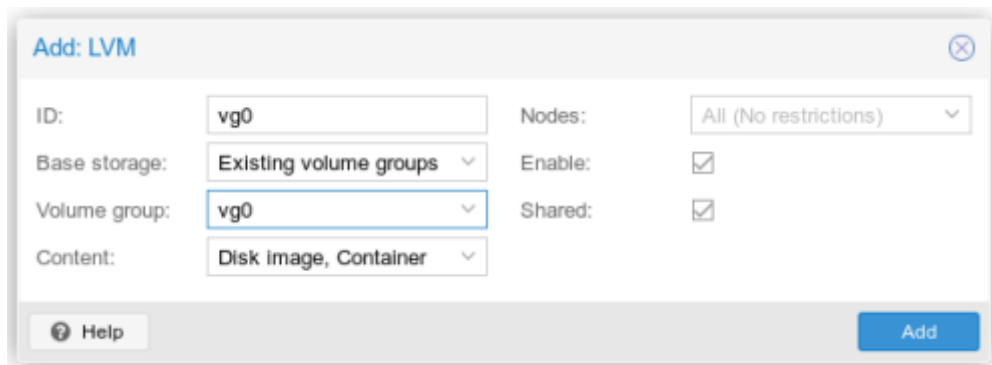
- Une fois la synchronisation faite, tapez sur le second nœud :

```
# drbdadm primary pv0
```

Une fois synchronisé, on crée le volume LVM :

```
# vgcreate vg0 /dev/drbd0
```

et on l'ajoute dans l'interface proxmox :



il faut impérativement cocher la case **Shared**

Il faut ensuite limiter la détection des volumes LVM, afin qu'il évite de "monter" le **vg** avant le DRBD.

Pour cela on va modifier, dans le fichier **/etc/lvm/lvm.conf**, la variable **global\_filter** :

```
global_filter = [ "a|/dev/drbd0|", "a|/dev/sda.*|", "r|.*|" ]
```

cette variable fonctionne en expression régulière. Nous autorisons manuellement chaque disque et désactivons toute détection.

puis il faut mettre à jour l'**initramfs** :

```
# update-initramfs -u
```

From:  
<https://wiki.virtit.fr/> - VirtIT

Permanent link:  
[https://wiki.virtit.fr/doku.php/kb:linux:proxmox:cluster\\_deux\\_n%C5%93uds\\_proxmox?rev=1569517796](https://wiki.virtit.fr/doku.php/kb:linux:proxmox:cluster_deux_n%C5%93uds_proxmox?rev=1569517796)

Last update: 2019/09/26 17:09

