2025/11/05 01:19 1/4 Cluster deux nœuds Proxmox

Cluster deux nœuds Proxmox

Cette solution comporte des risques de Split-brain. Il faut évidement 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



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/cororsync.conf** d'un des 2 nœuds, comme ceci :

```
logging {
  debug: off
  to_syslog: yes
}
nodelist {
  node {
    name: pvel
```

2025/11/05 01:19 2/4 Cluster deux nœuds Proxmox

```
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 entré 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;
```

2025/11/05 01:19 3/4 Cluster deux nœuds Proxmox

```
startup {
    become-primary-on both;
}
syncer {
}
net {
    allow-two-primaries;
}
on pvel {
    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 :

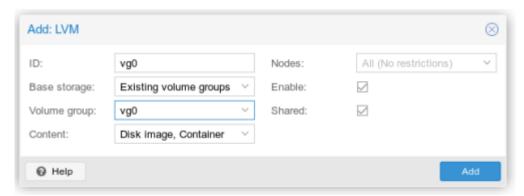
```
# drbdadm primary pv0
```

Une fois synchronisé, on créer le volume LVM :

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

et on l'ajoute dans l'interface proxmox :

2025/11/05 01:19 4/4 Cluster deux nœuds 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 chaques disques et désactivons tout détection.

puis il faut mettre a jour l'initramfs :

update-initramfs -u

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

Last update: 2019/09/26 07:57

