

Proxmox Clustering with 2 Nodes

 apalrd.net/posts/2022/pve_quorum

February 13, 2022

2022-02-13

I'm experimenting with Proxmox Virtual Environment (PVE), the same hypervisor I run on my [Minilab](#). It supports clustering and high availability, and I'd like to implement the cluster option. Clustering without HA allows multiple nodes to be managed from a single user interface, and for VMs to be offline migrated between nodes. This sounds pretty useful for me, even without the high availability features like live migration. However, any cluster relies on a node voting scheme which requires agreement (quorum) from all of the nodes, and the cluster won't function without quorum being met. With only two nodes in a cluster, it's impossible to achieve quorum and no VMs will start. The usual solution to this is to add a new device specifically to vote (a 'QDevice') running on another server or a Raspberry Pi, but it's also possible to give one node an extra vote to break the ties.

My minilab uses relatively little power (it's fed by a ~90W DC power brick) and is self contained with local storage, and it's adequate for many of my continuously running services, but I still have to rely on VirtualBox on my workstation for testing a lot of VMs at once. Ideally, I could setup a higher performance node for testing and power it down when I don't need it, with the ability to migrate testing VMs to the production mini-server. In this use case, I can be fairly certain that the minilab will always be running but the testing server might not be, so giving the minilab an extra vote means it can maintain quorum on its own, even if the 'megablab' is shut down. This would allow me to operate without a separate Qdevice and still cluster the two nodes together.

Quick Video

This video is a really quick tutorial which focuses on how to do it, and not the background behind why.

The Proxmox Cluster Filesystem

Proxmox stores the vast majority of configuration files in the Proxmox Cluster Filesystem (pmxcfs). Configuration for VMs and containers are stored in the cluster filesystem, along with configuration for each node and the cluster itself. Corosync is used to replicate the cluster filesystem across all of the nodes in the cluster, and deal with distributed locking of cluster configuration files. The cluster filesystem is used even for single-node Proxmox installations, but without synchronization across the cluster.

The cluster filesystem is mounted at `/etc/pve`, so files in this path can be edited by any node and synchronized automatically. The cluster configuration file itself is located at `/etc/pve/corosync.conf`. We need to edit this configuration file to change the quorum settings, but we need to be very careful not to mess anything up since corosync will propagate any change we make to all nodes immediately.

Editing the Configuration File

Proxmox has some warnings on editing the corosync file - [see here](#). Specifically, they recommend creating a copy of the file when editing it, so accidental saving of the unfinished file doesn't cause problems. So, we will do that.

```
cp /etc/pve/corosync.conf /etc/pve/corosync.new.conf
```

Now, we can edit the corosync config file in nano to give minilab two quorum votes, so it can run on its own.

```
nano /etc/pve/corosync.new.conf
```

First, in the nodelist section, increase **quorum votes** to **2** for the node that we want to run on its own:

```
nodelist {
  node {
    name: minilab
    nodeid: 1
    quorum votes: 2
    ring0 addr: 10.0.0.3
  }
  node {
    name: megalab
    nodeid: 2
    quorum votes: 1
    ring0 addr: 10.0.0.4
  }
}
```

Increment the **config version** key in the totem section by 1 (in this example, from 2 to 3):

```
totem {
  cluster name: cluster
  config version: 3
  interface {
    linknumber: 0
  }
  ip version: ipv4-6
  link mode: passive
  secauth: on
  version: 2
}
```

And finally, copy the modified file back to the original:

```
mv /etc/pve/corosync.conf /etc/pve/corosync.conf.bak
mv /etc/pve/corosync.new.conf /etc/pve/corosync.conf
```

If you really mess things up, it's possible that the nodes won't quorate. If this happens, the entire Proxmox cluster filesystem becomes read-only. You can temporarily reduce the required quorum to 1 to continue:

```
pvecm expected 1.
```

Conclusions

For homelabs which don't have enough nodes to maintain quorum, or which don't run all of the nodes all the time, increasing the votes for the continuously running servers can help keep the cluster online without adding additional QDevices and depending on those to be online.