

Setting Up a Proxmox Cluster: Networking, Shared Storage, VM Migrations, and High Availability

 manjit28.medium.com/setting-up-a-proxmox-cluster-networking-shared-storage-vm-migrations-and-high-availability-15ea4708991f

Manjit Singh

22 сентября 2024 г.



Manjit Singh

In this article, we go through setting up a Proxmox Cluster using three hosts, focusing on networking configurations, shared storage integration, and VM migrations. Whether running production applications or managing a homelab, this guide will show how clustering can improve resource efficiency, provide redundancy, and ensure continuous uptime for your VMs

This is a follow up to the post where we had setup an individual node:

Setting Up Proxmox VE

This post is about setting up Proxmox on a bare metal server. But first of all, what is Proxmox? It's a virtualization...

manjit28.medium.com

I have setup two more nodes following above procedure. At this stage, they are all independent with their own management UIs. Now we'll set up the Proxmox cluster. First, let us do some prep work before we put nodes in a cluster.

Networking

It is not required but very good practice to keep management networks from VM networks. As I had four ports on each server, I went with following design:

Port 1 on each server — Management Network

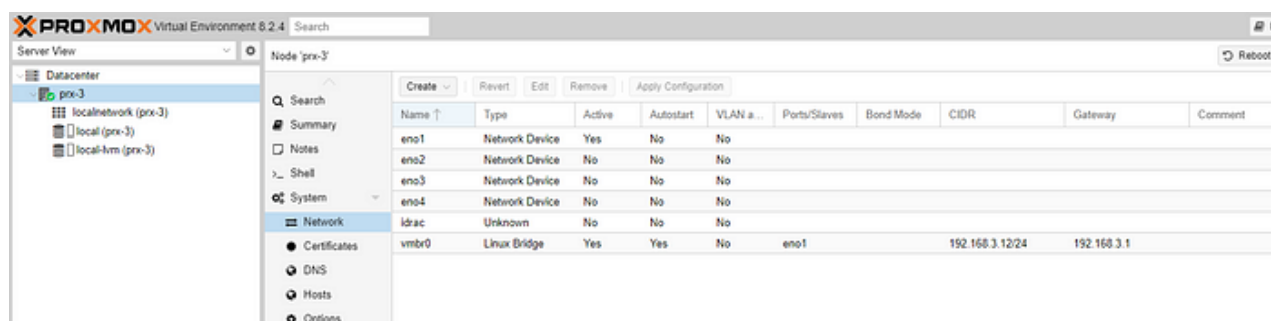
This is what I provided to Proxmox at installation time, 192.168.3.x range
Linux Bridge was created by Proxmox

Port 2 on each server — VM Network (Data Plane)

In my case, this is connected to different switch/subnet, 192.168.4.x range
We'll now create Linux Bridge to use this port

Linux Bridge: "A Linux Bridge allows the physical network interface to act as a virtual switch inside our server. This helps connect our virtual machines to different network segments."

Log on on to one node and get to Network tab (process needs to be done on all nodes):



Proxmox Network — Setup Bridge

I am going to use eno2 that is connected to a separate subnet and use this network for VMs. First bridge created by Proxmox is named as vmbr0, so we'll name the new one as vmbr1. In comments, I put 'vm-net4' to represent usage and subnet but you can put any description here:

Create: Linux Bridge

Name:

vmbr1

IPv4/CIDR:

192.168.4.12/24

Gateway (IPv4):

IPv6/CIDR:

Gateway (IPv6):

Autostart:

☒

VLAN aware:

☐

Bridge ports:

eno2

Comment:

vm-net4

MTU:

1500

Help

Advanced ☒

Create

Create Linux Bridge

Leave Gateway blank here. Click on Apply Configuration, otherwise changes will only take effect on the next restart.

| Create | Revert | Edit | Remove | Apply Configuration | | | | | |
|--------|----------------|--------|-----------|---------------------|--------------|-----------|-----------------|-------------|---------|
| Name ↑ | Type | Active | Autostart | VLAN a... | Ports/Slaves | Bond Mode | CIDR | Gateway | Comment |
| eno1 | Network Device | Yes | No | No | | | | | |
| eno2 | Network Device | No | No | No | | | | | |
| eno3 | Network Device | No | No | No | | | | | |
| eno4 | Network Device | No | No | No | | | | | |
| idrac | Unknown | No | No | No | | | | | |
| vmbr0 | Linux Bridge | Yes | Yes | No | eno1 | | 192.168.3.12/24 | 192.168.3.1 | |
| vmbr1 | Linux Bridge | No | Yes | No | eno2 | | 192.168.4.12/24 | | vm-net4 |

Pending changes (Either reboot or use 'Apply Configuration' (needs ifupdown2) to activate)

```

--- /etc/network/interfaces      2024-09-01 20:46:40.962833889 -0400
+++ /etc/network/interfaces.new 2024-09-03 19:36:35.177309584 -0400
@@ -1,8 +1,25 @@
+# network interface settings; autogenerated
+# Please do NOT modify this file directly, unless you know what
+# you're doing.
+#
+# If you want to manage parts of the network configuration manually,
+# please utilize the 'source' or 'source-directory' directives to do
+# so.
+# PVE will preserve these directives, but will NOT read its network

```

Apply Network Configuration

Next time we create VMs, we'll pick this new network that we just created:

Create: Virtual Machine

General
OS
System
Disks
CPU
Memory
Network
Confirm

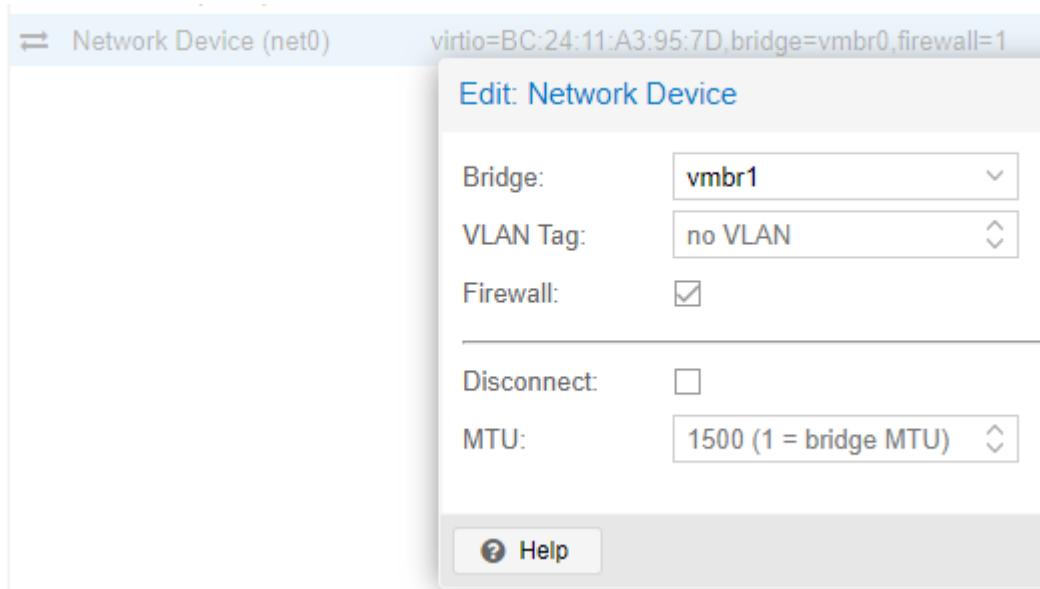
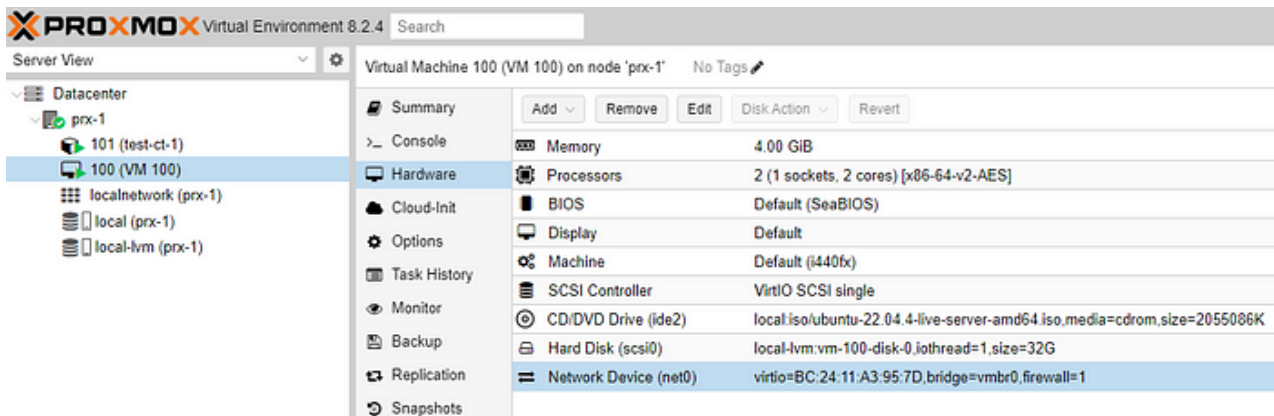
☐ No network device

Bridge:
vmbr1
Model:
VirtIO (paravirtualized)
VLAN Tag:
no VLAN
MAC address:
auto
Firewall:
☒

Disconnect:
☐
Rate limit (MB/s):
unlimited
MTU:
1500 (1 = bridge MTU)
Multiqueue:

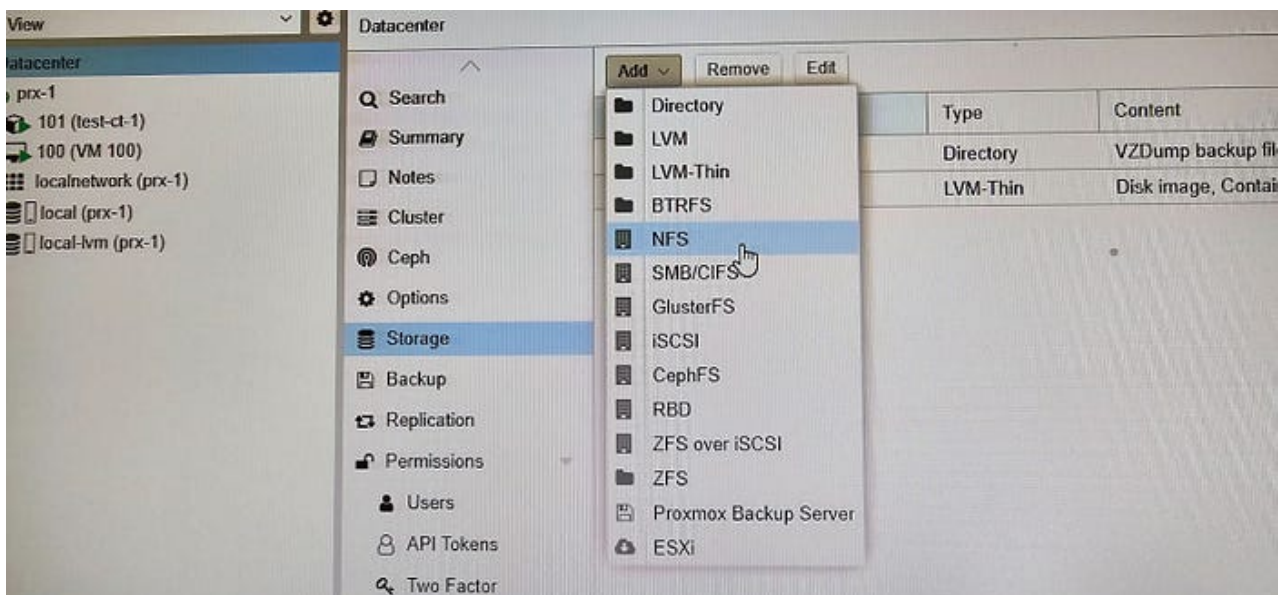
Select network for VM creation

We can change it for existing VMs also:



Shared Storage

Even though we have not set up the Cluster yet, we can define Shared storage at Datacenter level and not node level. I have access to NFS share from my Synology but as we can see below, we have many options to pick from:



Add Shared Storage to Proxmox

Add: NFS

General
Backup Retention

ID:
vm-storage-1
Nodes:
All (No restrictions)

Server:
192.168.4.46
Enable:
☒

Export:
/volume1/prx-vm-1
Content:
Disk image, ISO image,

Preallocation:
Default
NFS Version:
Default

? Help
Advanced ☒
Add

In Content above, we decide what this shared storage can be used for, like for VM disks, ISO images etc.

We have defined one for virtual machines above, let us define one for backups also:

Add: NFS

General
Backup Retention

ID:
syn-vm-backups
Nodes:
All (No restrictions)

Server:
192.168.4.46
Enable:
☒

Export:
/volume2/vm-backups
Content:
Disk image

Preallocation:
Default
NFS Version:
Default

? Help
Advanced ☒
Add

With shared storage configured, we can now leverage it to seamlessly move VMs between nodes in the cluster. This flexibility is a key benefit of clustering, as it allows for workload balancing and high availability.

Clustering

At this stage, we have three separate Proxmox hosts. We have set up two networks on each and added shared storage to one. At this stage we are good for creating the cluster.

Server View

Node 'prx-1'

Datacenter

- prx-1
 - 101 (test-ct-1)
 - 100 (VM 100)
 - localnetwork (prx-1)
 - local (prx-1)
 - local-lvm (prx-1)
 - syn-vm-backups (prx-1)
 - syn-vm-storage (prx-1)

Search

Summary

Notes

Shell

System

Network

Certificates

DNS

Hosts

Options

Time

System Log

Create

Name ↑

| |
|-------|
| eno1 |
| eno2 |
| eno3 |
| eno4 |
| vmbr0 |
| vmbr1 |

PROXMOX Virtual Environment 8.2.4 Search

Server View

Node 'prx-2'

Datacenter

- prx-2
 - localnetwork (prx-2)
 - local (prx-2)
 - local-lvm (prx-2)

Search

Summary

Notes

Shell

System

Network

Certificates

DNS

Hosts

Options

Time

System Log

Create

Revert

Edit

| Name ↑ | Type |
|--------|----------------|
| eno1 | Network Device |
| eno2 | Network Device |
| eno3 | Network Device |
| eno4 | Network Device |
| vmbr0 | Linux Bridge |
| vmbr1 | Linux Bridge |

PROXMOX Virtual Environment 8.2.4 Search

Server View

Datacenter

- prx-3
 - localnetwork (prx-3)
 - local (prx-3)
 - local-lvm (prx-3)

Node 'prx-3'

Create | Revert | Edit

Search

Summary

Notes

Shell

System

Network

Certificates

DNS

Hosts

| Name ↑ | Type |
|--------|----------------|
| eno1 | Network Device |
| eno2 | Network Device |
| eno3 | Network Device |
| eno4 | Network Device |
| idrac | Unknown |
| vmbr0 | Linux Bridge |
| vmbr1 | Linux Bridge |

Let us Create the cluster now:

Server View

Datacenter

- prx-1
 - 101 (test-ct-1)
 - 100 (VM 100)
 - localnetwork (prx-1)
 - local (prx-1)
 - local-lvm (prx-1)
 - syn-vm-backups (prx-1)
 - syn-vm-storage (prx-1)

Cluster

Ceph

Options

Storage

Backup

Cluster Information

Create Cluster | Join Information | Join Cluster

Standalone node - no cluster defined

Cluster Nodes

Nodename

Create Cluster

Pick a cluster name and give it node IP:

Create Cluster

Cluster Name: prx-cluster

Cluster Network: Link: 0 192.168.3.10

Add Multiple links are used as failover, lower numbers have higher priority.

Help Create

Now we have a cluster with one node. By clicking on Join Information button, we can grab information that we need to take to other nodes to add them to same cluster:

Task viewer: Create Cluster

Output

Status

Stop

Corosync Cluster Engine Authentication key generator.
Gathering 2048 bits for key from /dev/urandom.
Writing corosync key to /etc/corosync/authkey.
Writing corosync config to /etc/pve/corosync.conf
Restart corosync and cluster filesystem
TASK OK

Create Cluster — Output

The screenshot displays the Proxmox Virtual Environment 8.2.4 interface. The top bar shows the Proxmox logo and version. The left sidebar contains a tree view of the Datacenter (prx-cluster-1) and its nodes (prx-1, prx-2). The main panel is divided into two sections: 'Cluster Information' and 'Cluster Nodes'. The 'Cluster Information' section shows the cluster name 'prx-cluster-1' and the 'Join Information' tab is selected. The 'Cluster Nodes' section shows the node 'prx-1' and the 'Join Information' tab is selected. Below the main panel, a 'Cluster Join' dialog box is open, showing the 'Assisted join' option selected and a text input field for 'Paste encoded Cluster Information here'. The 'Join' button is visible at the bottom right of the dialog.

PROXMOX Virtual Environment 8.2.4 Search

Server View

Datacenter (prx-cluster-1)

- prx-1
 - 101 (test-ct-1)
 - 100 (VM 100)
 - localnetwork (prx-1)
 - local (prx-1)
 - local-lvm (prx-1)
 - syn-vm-backups (prx-1)
 - syn-vm-storage (prx-1)

Search

Summary

Notes

Cluster

Ceph

Options

Storage

Backup

Cluster Information

Create Cluster Join Information Join Cluster

Cluster Name: prx-cluster-1

Cluster Nodes

Nodename

prx-1

Datacenter

- prx-2
 - localnetwork (prx-2)
 - local (prx-2)
 - local-lvm (prx-2)

Search

Summary

Notes

Cluster

Ceph

Options

Cluster Information

Create Cluster Join Information Join Cluster

Standalone node - no cluster defined

Cluster Nodes

Nodename

Cluster Join

☒ Assisted join: Paste encoded cluster join information and enter password.

Information: Paste encoded Cluster Information here

Help Join

Cluster Join

☒ Assisted join: Paste encoded cluster join information and enter password.

Information:

Peer Address:

192.168.3.10

Password:

Fingerprint:

Cluster Network:

Link: 0

IP resolved by node's hostname

peer's link address: 192.168.3.10

Help

Join 'prx-cluster-1'

Now we can refresh page and see all nodes in cluster:

PROXMOX Virtual Environment 8.2.4

Server View ▼ ⚙️ **Datacenter**

▼ **Datacenter (prx-cluster-1)**

- ▼ **prx-1**
 - 101 (test-ct-1)
 - 100 (VM 100)
 - localnetwork (prx-1)
 - local (prx-1)
 - local-lvm (prx-1)
 - syn-vm-backups (prx-1)
 - syn-vm-storage (prx-1)
- ▼ **prx-2**
 - localnetwork (prx-2)
 - local (prx-2)
 - local-lvm (prx-2)
 - syn-vm-backups (prx-2)
 - syn-vm-storage (prx-2)
- ▼ **prx-3**
 - localnetwork (prx-3)
 - local (prx-3)
 - local-lvm (prx-3)
 - syn-vm-backups (prx-3)
 - syn-vm-storage (prx-3)

Cluster Information

Create Cluster Join Information Join Cluster

Cluster Name: prx-cluster-1

Cluster Nodes

Nodename

- prx-1
- prx-2
- prx-3

Also, in above we can see that Shared storage is accessible to all nodes even though we added it to only one node.

Virtual Machine Migrations

We can test our cluster by Migrating VM:

ver View

Datacenter (prx-cluster-1)

prx-1

101 (test-ct-1)

100 (VM 100)

localnetwork VM 100

local (prx-1) Pause
local-lvm (prx-1) Hibernate
syn-vm-backups (prx-1) Shutdown
syn-vm-storage (prx-1) Stop

prx-2

Reboot

localnetwork

Migrate

local (prx-2)

Clone

local-lvm (prx-2)

Convert to template

syn-vm-backups (prx-2)

syn-vm-storage (prx-2)

Console

prx-3

localnetwork (prx-3)

local (prx-3)

local-lvm (prx-3)

syn-vm-backups (prx-3)

syn-vm-storage (prx-3)

Summary

Console

Hardware

Cloud-Init

Options

Task History

Monitor

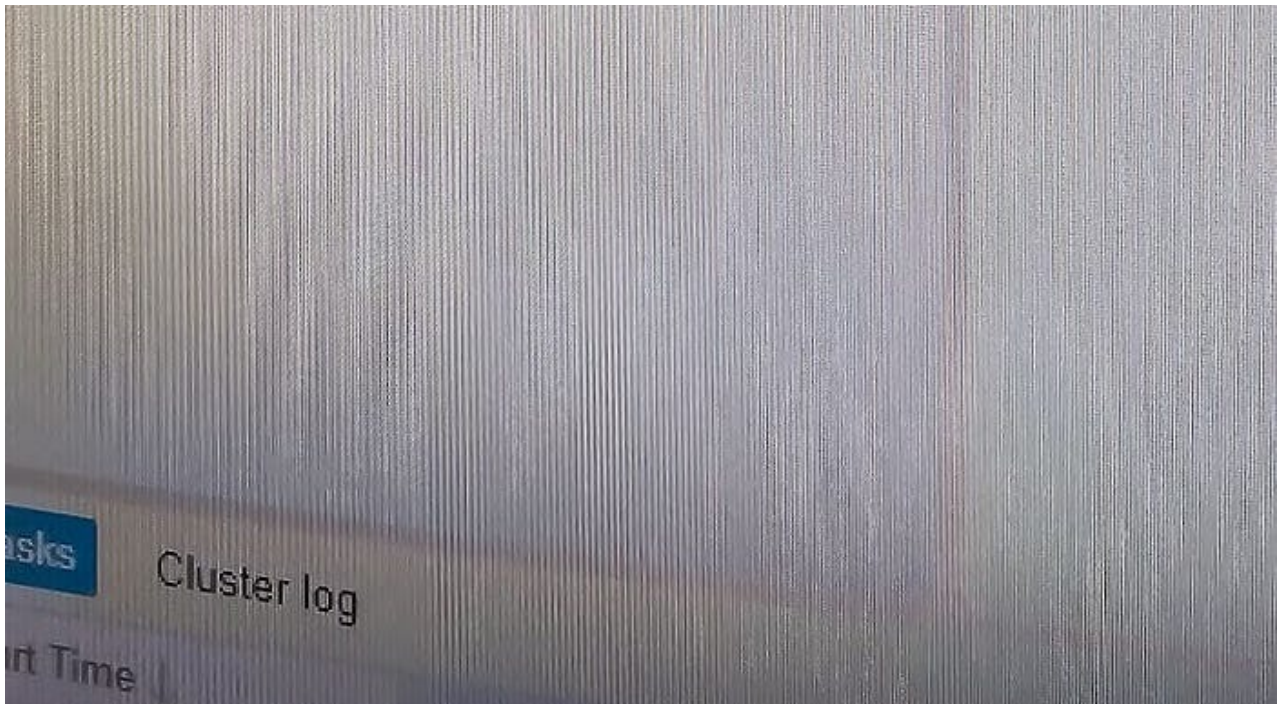
Backup

Replication

Snapshots

Firewall

Permissions



This VM was created using local disk. Migrating VM with local disk:

Migrate VM 100

Source node: prx-1

Target node: prx-2

Mode: Online

Target storage: Current layout

Info ↑

⚠ Migration with local disk might take long: local-lvm:vm-100-disk-0 (32.00 GiB)

Help

Migrate

Task viewer: VM 100 - Migrate (prx-1 ----> prx-2)

Output

Status

Stop

2024-09-04 20:31:16 starting migration of VM 100 to node 'prx-2' (192.168.3.11)
2024-09-04 20:31:16 found local disk 'local-lvm:vm-100-disk-0' (attached)
2024-09-04 20:31:16 starting VM 100 on remote node 'prx-2'
2024-09-04 20:31:22 volume 'local-lvm:vm-100-disk-0' is 'local-lvm:vm-100-disk-0' on the target
2024-09-04 20:31:22 start remote tunnel
2024-09-04 20:31:24 ssh tunnel ver 1
2024-09-04 20:31:24 starting storage migration
2024-09-04 20:31:24 scsi0: start migration to nbd:unix:/run/qemu-server/100_nbd.migrate:exportname=drive-scsi0
drive mirror is starting for drive-scsi0
drive-scsi0: transferred 0.0 B of 32.0 GiB (0.00%) in 0s

It took a few minutes to migrate over 1Gb network. The VM was responsive during this time.

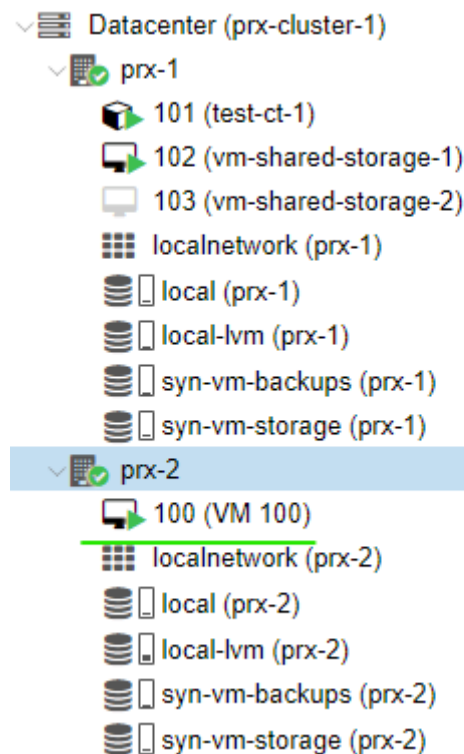
```

--- 192.168.4.66 ping statistics ---
284 packets transmitted, 280 received, 1.40845% packet loss, time 289727ms
rtt min/avg/max/mdev = 0.494/0.609/0.855/0.040 ms

```

Now we can see that VM has move to prx-2 node:

We can also verify from logs at bottom of screen:



| Node | User name | Description | Status |
|-------|-----------|---------------------|--------|
| prx-3 | root@pam | VM/CT 102 - Console | OK |
| prx-2 | root@pam | VM 100 - Start | OK |
| prx-1 | root@pam | VM 100 - Migrate | OK |

Now let us Create VM using shared storage. Process is same except that we select disk from shared storage:

Create: Virtual Machine

General OS System **Disks** CPU Memory Network Confirm

scsi0

Disk Bandwidth

Bus/Device: SCSI 0 Cache: Default (No cache)

SCSI Controller: VirtIO SCSI single Discard: ☐

Storage: syn-vm-storage IO thread: ☒

Disk size (GiB): 20

Format: QEMU image format

Create: Virtual Machine

General
OS
System
Disks
CPU
Memory
Network
Confirm

| Key ↑ | Value |
|----------|--|
| cores | 2 |
| cpu | x86-64-v2-AES |
| ide2 | local:iso/ubuntu-22.04.4-live-server-amd64.iso,media=cdrom |
| memory | 4096 |
| name | vm-shared-storage-1 |
| net0 | virtio,bridge=vbr1,firewall=1 |
| nodename | prx-1 |
| numa | 0 |
| ostype | l26 |
| scsi0 | syn-vm-storage:20,format=qcow2,iothread=on |
| scsihw | virtio-scsi-single |
| sockets | 1 |
| vmid | 102 |

☐ Start after created

Advanced ☒
Back
Finish

Migrate VM with shared storage:

Migrate VM 102

Source node: prx-1
Target node: prx-2

Mode: Online

? Help
Migrate

It is very quick:

Task viewer: VM 102 - Migrate (prx-1 ----> prx-2)

Output

Status

Stop

```
2024-09-04 20:40:31 starting migration of VM 102 to node 'prx-2' (192.168.3.11)
2024-09-04 20:40:31 starting VM 102 on remote node 'prx-2'
2024-09-04 20:40:36 start remote tunnel
2024-09-04 20:40:38 ssh tunnel ver 1
2024-09-04 20:40:38 starting online/live migration on unix:/run/qemu-server/102.migrate
2024-09-04 20:40:38 set migration capabilities
2024-09-04 20:40:38 migration downtime limit: 100 ms
2024-09-04 20:40:38 migration cachesize: 512.0 MiB
2024-09-04 20:40:38 set migration parameters
2024-09-04 20:40:38 start migrate command to unix:/run/qemu-server/102.migrate
2024-09-04 20:40:39 migration active, transferred 47.4 MiB of 4.0 GiB VM-state, 608.5 MiB/s
2024-09-04 20:40:40 migration active, transferred 149.2 MiB of 4.0 GiB VM-state, 142.0 MiB/s
2024-09-04 20:40:41 migration active, transferred 262.0 MiB of 4.0 GiB VM-state, 111.7 MiB/s
2024-09-04 20:40:42 migration active, transferred 374.5 MiB of 4.0 GiB VM-state, 112.2 MiB/s
2024-09-04 20:40:43 migration active, transferred 485.2 MiB of 4.0 GiB VM-state, 122.8 MiB/s
2024-09-04 20:40:44 average migration speed: 685.5 MiB/s - downtime 32 ms
2024-09-04 20:40:44 migration status: completed
2024-09-04 20:40:49 migration finished successfully (duration 00:00:19)
TASK OK
```

```
--- 192.168.4.67 ping statistics ---
34 packets transmitted, 34 received, 0% packet loss, time 33748ms
rtt min/avg/max/mdev = 0.502/0.597/0.836/0.057 ms
```

High Availability

So far we have migrated VMs manually. There are use cases for doing so. But if want VMs to migrate when there is an issue with a given node, then we need High Availability.

PROXMOX Virtual Environment 8.2.4

Server View ⌵ ⚙ **Datcenter**

Datcenter (prx-cluster-1)

- prx-1
 - 101 (test-ct-1)
 - 103 (vm-shared-storage-2)
 - localnetwork (prx-1)
 - local (prx-1)
 - local-lvm (prx-1)
 - syn-vm-backups (prx-1)
 - syn-vm-storage (prx-1)
- prx-2
 - 100 (VM 100)
 - 102 (vm-shared-storage-1)
 - localnetwork (prx-2)
 - local (prx-2)
 - local-lvm (prx-2)
 - syn-vm-backups (prx-2)
 - syn-vm-storage (prx-2)
- prx-3

Permissions

- Users
- API Tokens
- Two Factor
- Groups
- Pools
- Roles
- Realms
- HA**
 - Groups
 - Fencing
- SDN
 - Zones
 - VNets

Status

| Type | Status |
|--------|--------|
| quorum | OK |

Resources

Add Edit Remove

| ID | State | Node | Na |
|----|-------|------|----|
|----|-------|------|----|

Click on Add under Resources
And select VM to add to HA

Add: Resource: Container/Virtual Machine ⌵

| | | | |
|----------------|--------------------------------|----------------|--------------------------------------|
| VM: | <input type="text"/> | Group: | <input type="text"/> |
| Max. Restart: | <input type="text" value="1"/> | Request State: | <input type="text" value="started"/> |
| Max. Relocate: | <input type="text" value="1"/> | | |
| Comment: | <input type="text"/> | | |

ⓘ Help Add

We can leave Max Restarts and Max Relocate to 1. We do not want migration to be tried multiple times, we would rather know about underlying issue first at that time.

VM was stopped when adding, so it is starting now as requested state above was Started:

Status

| Type | Status |
|--------|--|
| quorum | OK |
| master | prx-2 (active, Wed Sep 4 20:48:00 2024) |
| lrm | prx-1 (idle, Wed Sep 4 20:48:03 2024) |
| lrm | prx-2 (wait_for_agent_lock, Wed Sep 4 20:48:00 2024) |
| lrm | prx-3 (idle, Wed Sep 4 20:48:03 2024) |

Resources

Add

Edit

Remove

| ID | State | Node | Name | Max. Restart | Max. Reloc... | Group |
|--------|----------|-------|---------------|--------------|---------------|-------|
| vm:102 | starting | prx-2 | vm-shared-... | 1 | 1 | |

Now, if something happens to the node on which the VM is running, it'll be moved to another node automatically (hence HA). I tested it by unplugging the network cable from the server on which VM was running and it moved to another node.

Use Cases

While High Availability (HA) is a powerful feature with clear use cases, it's important to assess whether it's necessary based on specific requirements. In scenarios where redundancy is already built into the application layer, such as Elasticsearch or Kafka clusters, we might prioritize high-speed local storage over fast VM migration, even if that means reduced mobility for the virtual machines. Ensuring data consistency often takes precedence in such cases. Conversely, for services like NGINX acting as a load balancer, HA can be a valuable option, as configuration updates are infrequent and the service benefits more from uptime and reliability than instantaneous data replication

Summary

In this article, we explored the process of setting up a Proxmox cluster using three hosts, focusing on configuring networking, shared storage, and VM migrations. By carefully designing the cluster architecture, including the management and VM networks, adding shared storage accessible by all nodes, and enabling HA, we've built a platform that can dynamically handle resource failures.

Next Steps

There is still lot more to do regarding usage of this cluster. In next posts, we'll go over backups/snapshots, user management and optimizing templates to quickly launch new virtual machines.

Thanks!