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

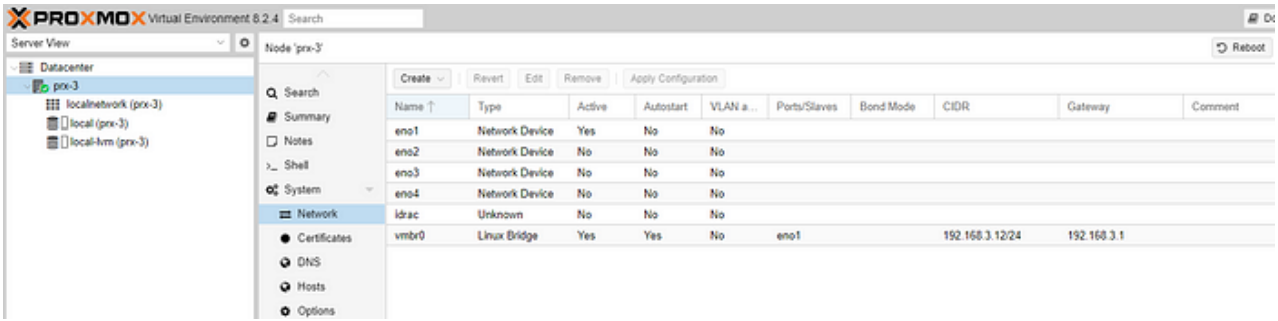
It is not required but very good practice to keep management networks from VM networks. As I had four ports on each server, I went with following design:

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

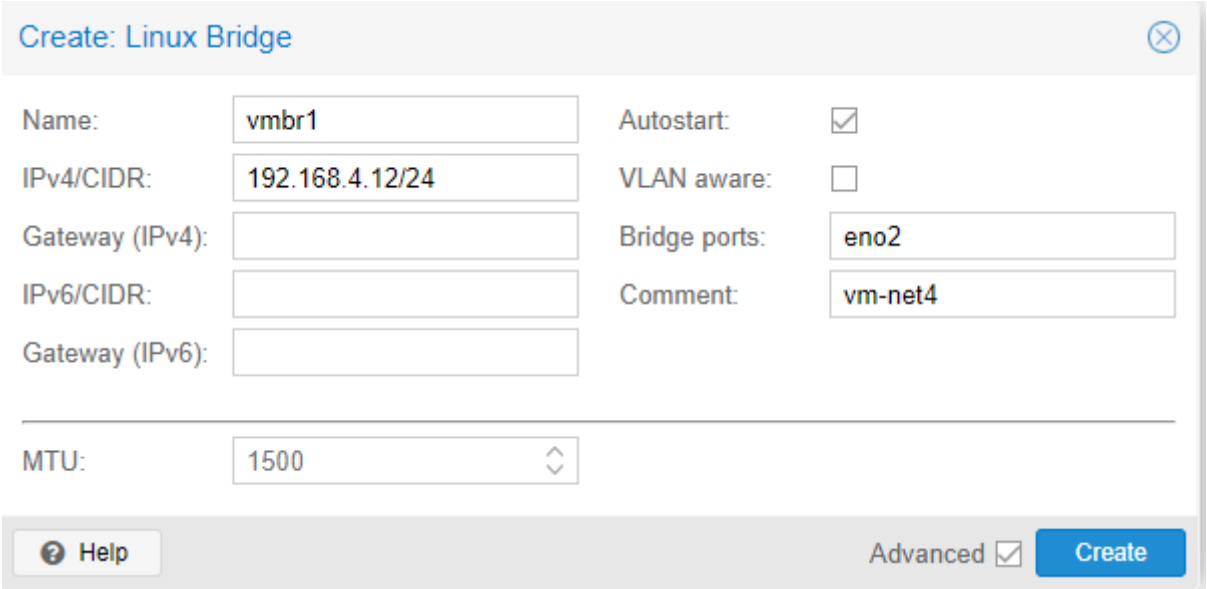
Linux Bridge: "A Linux Bridge allows the physical network interface to act as a virtual switch inside our server. This helps connect our virtual machines to different network segments."

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



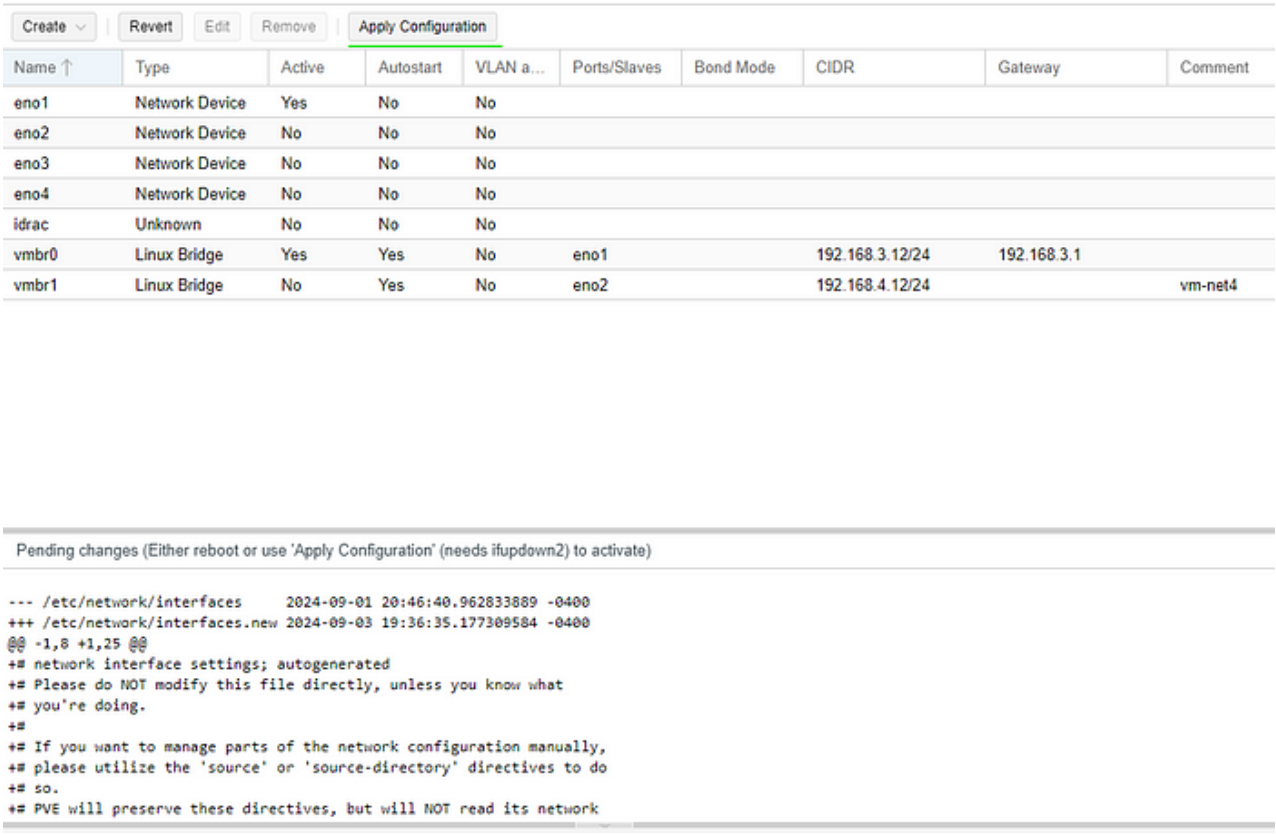
Proxmox Network — Setup Bridge

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:



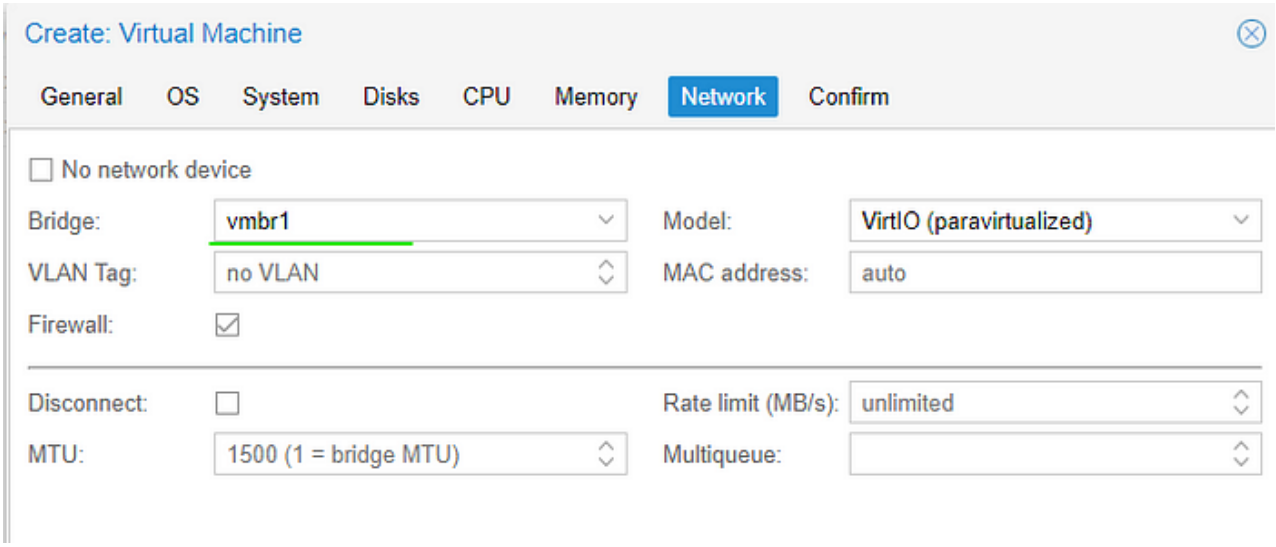
Create Linux Bridge

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



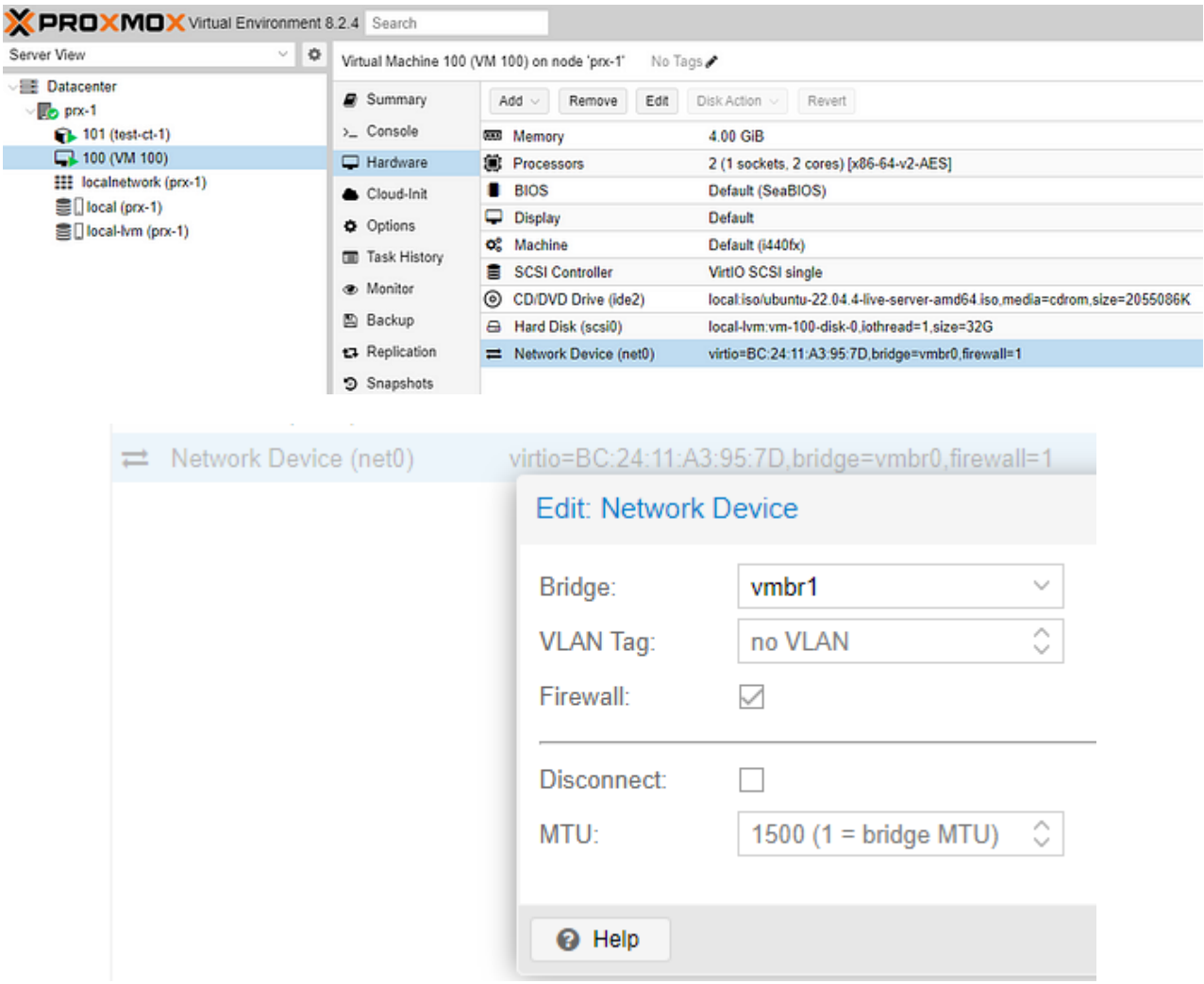
Apply Network Configuration

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



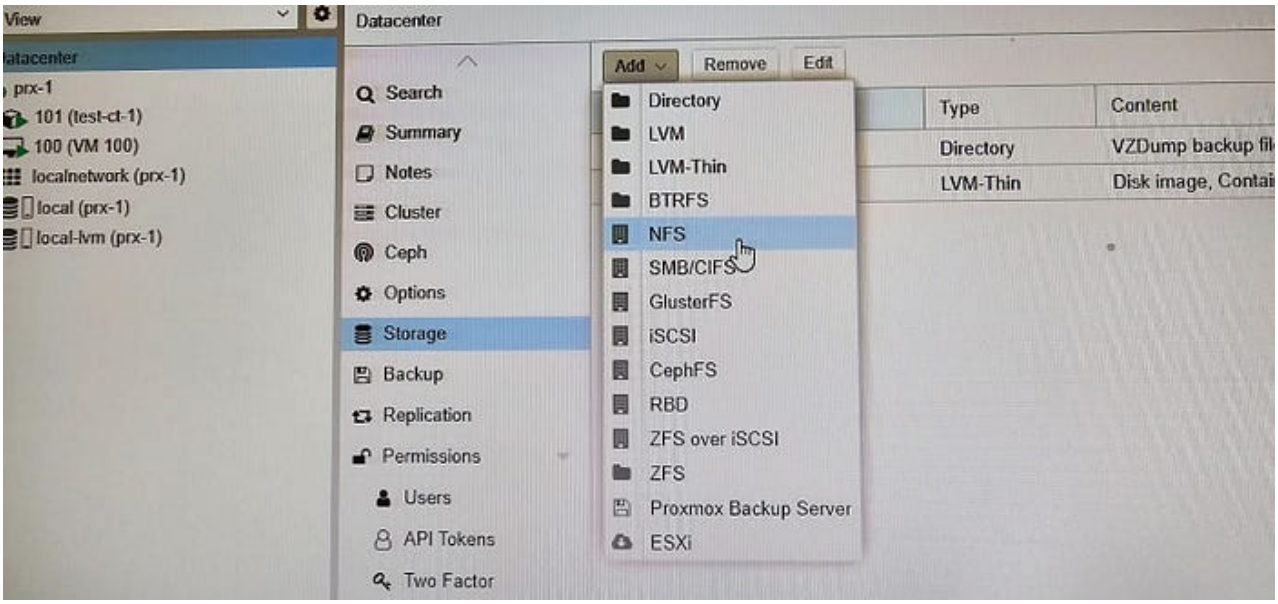
Select network for VM creation

We can change it for existing VMs also:

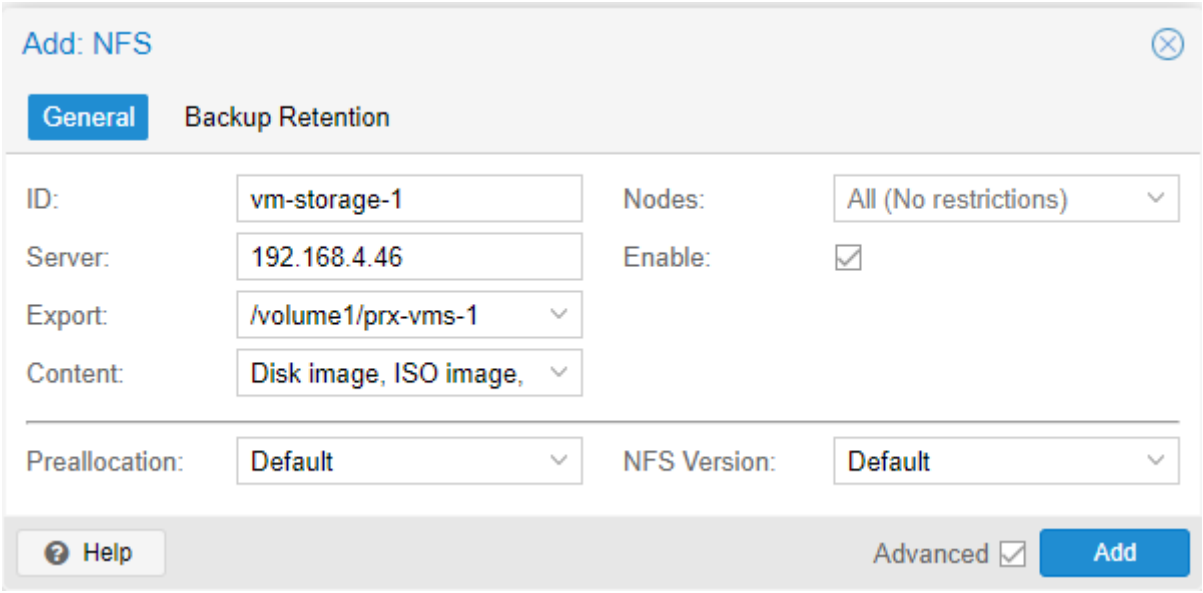


Shared Storage

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:

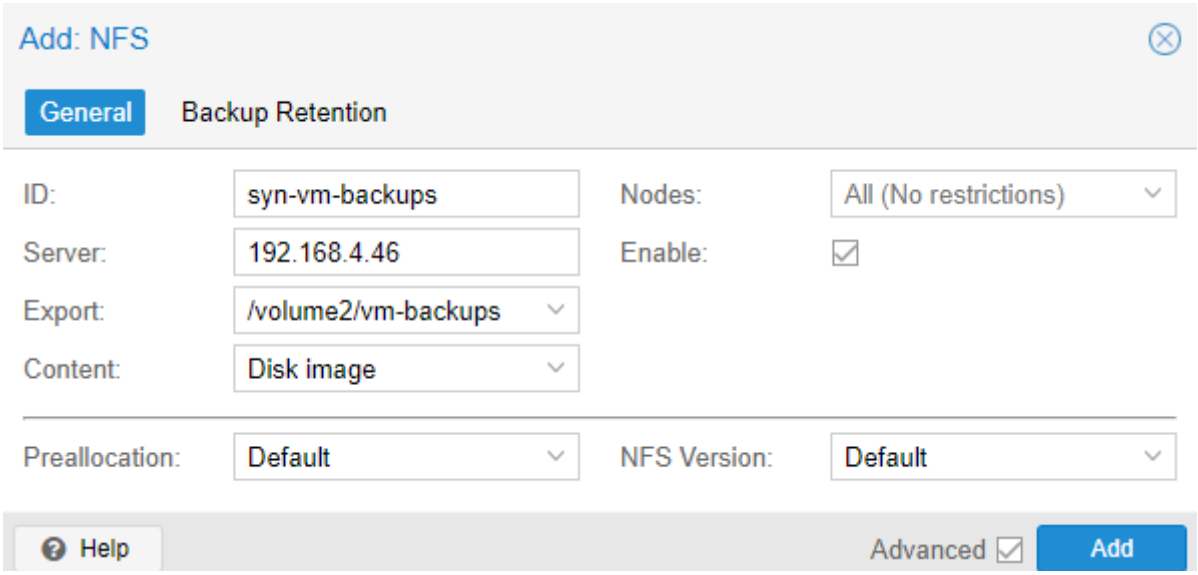


Add Shared Storage to Proxmox



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

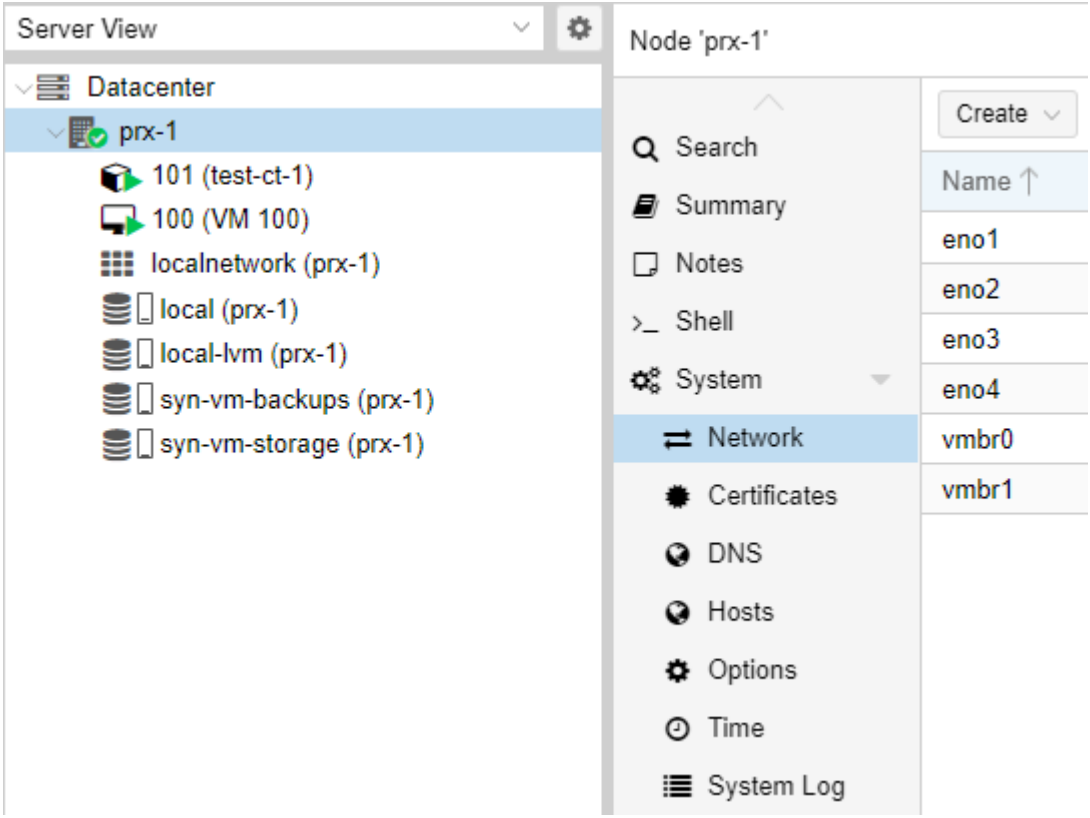
We have defined one for virtual machines above, let us define one for backups also:

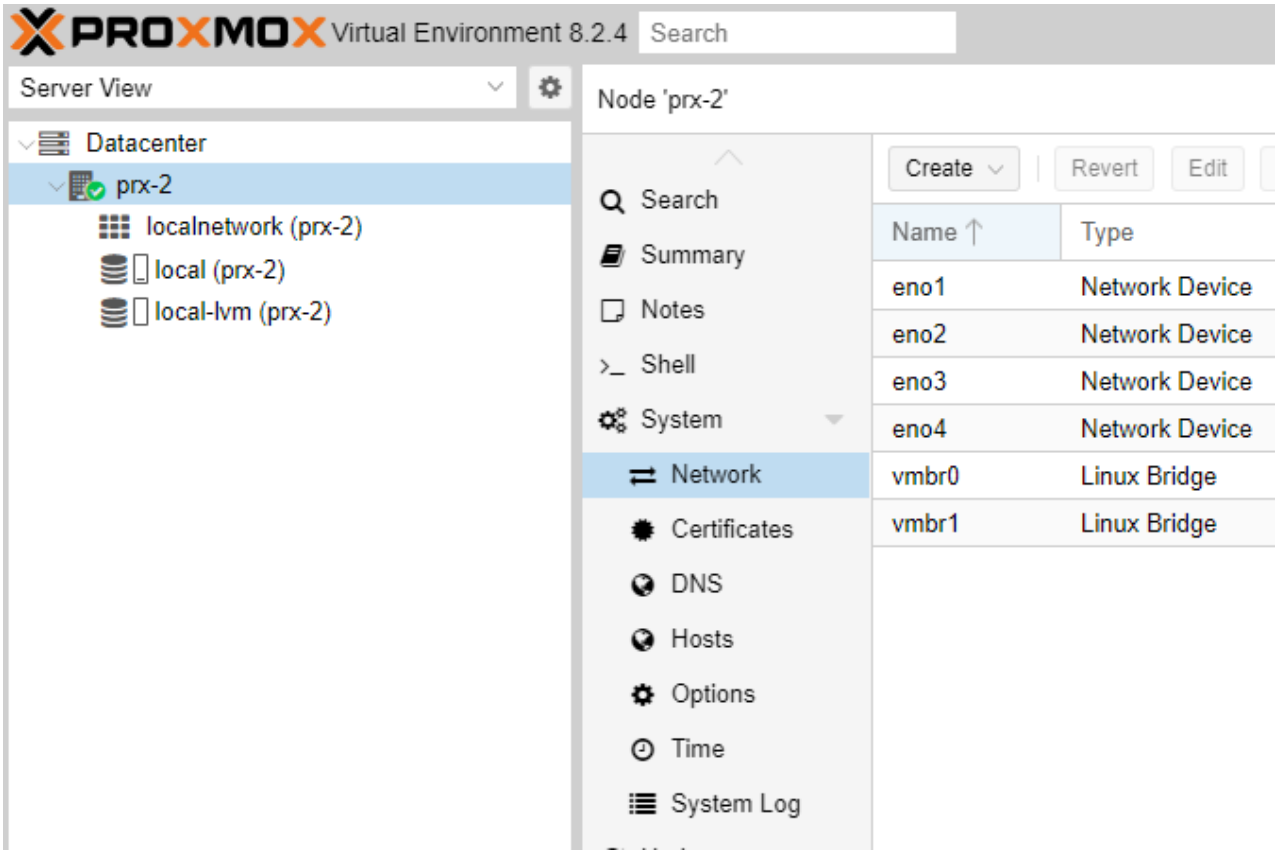


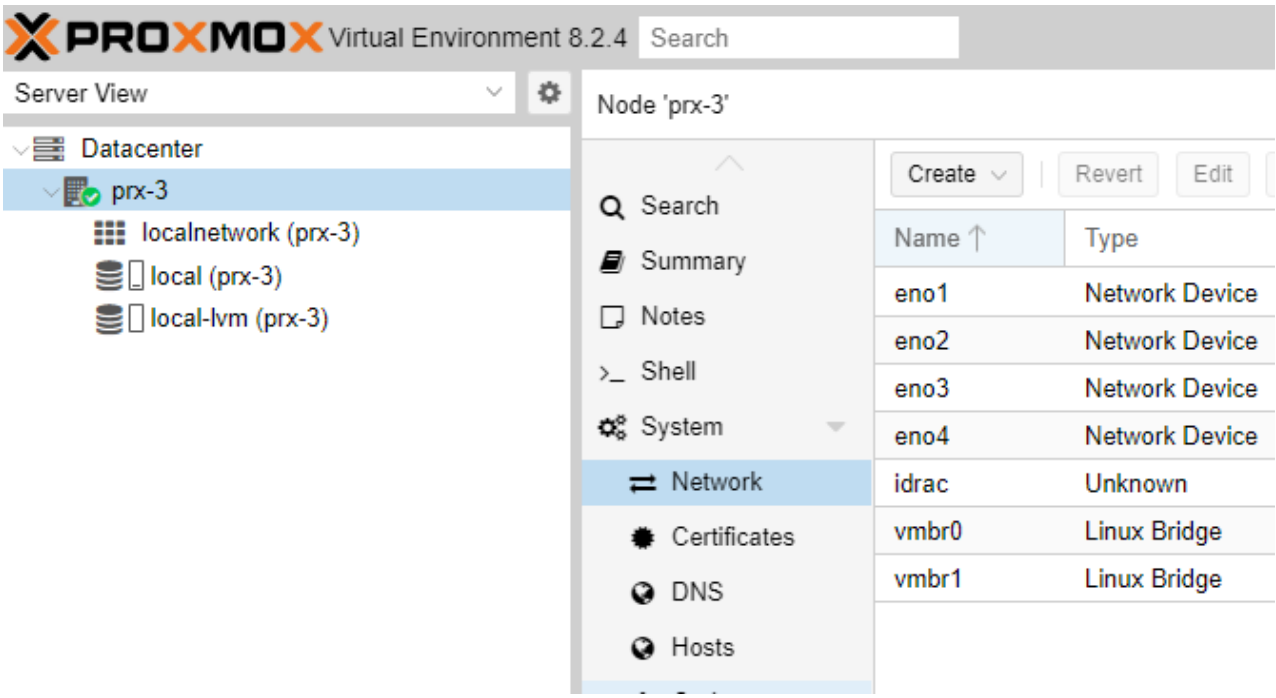
With shared storage configured, we can now leverage it to seamlessly move VMs between nodes in the cluster. This flexibility is a key benefit of clustering, as it allows for workload balancing and high availability.

Clustering

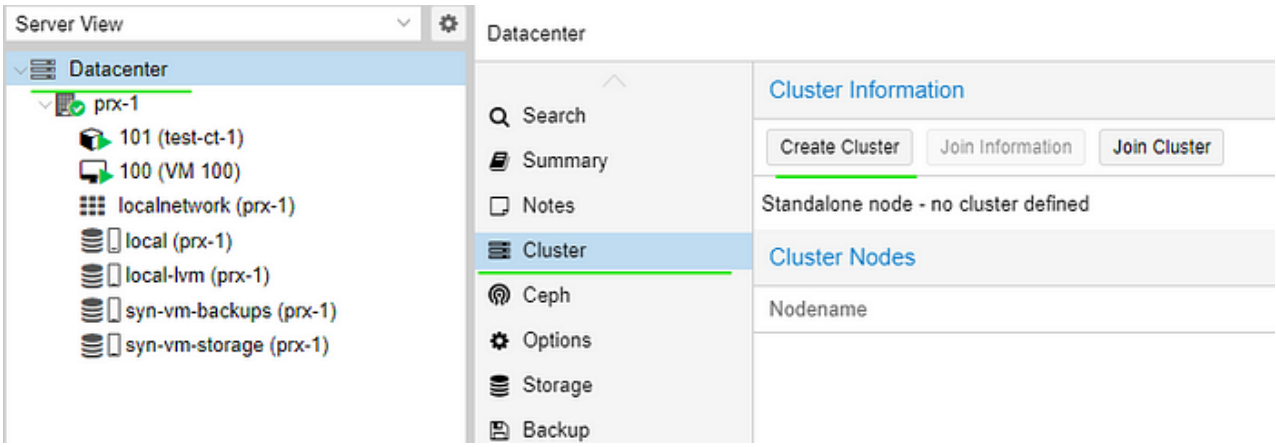
At this stage, we have three separate Proxmox hosts. We have set up two networks on each and added shared storage to one. At this stage we are good for creating the cluster.


Server View
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)
Node '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
Datacenter
prx-2
localnetwork (prx-2)
local (prx-2)
local-lvm (prx-2)
Node '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

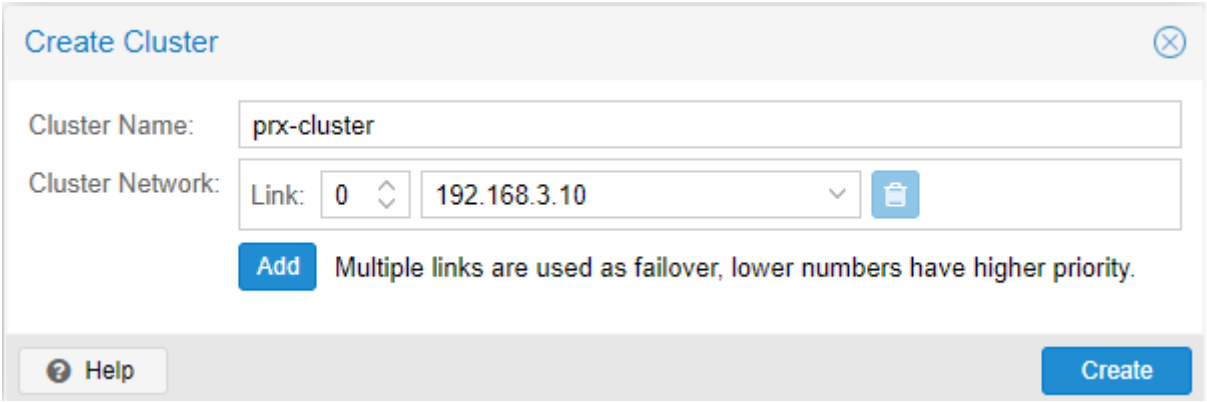


Let us Create the cluster now:

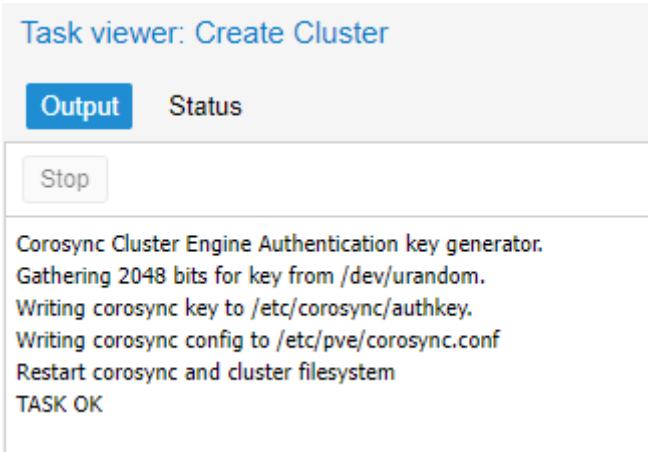


Create Cluster

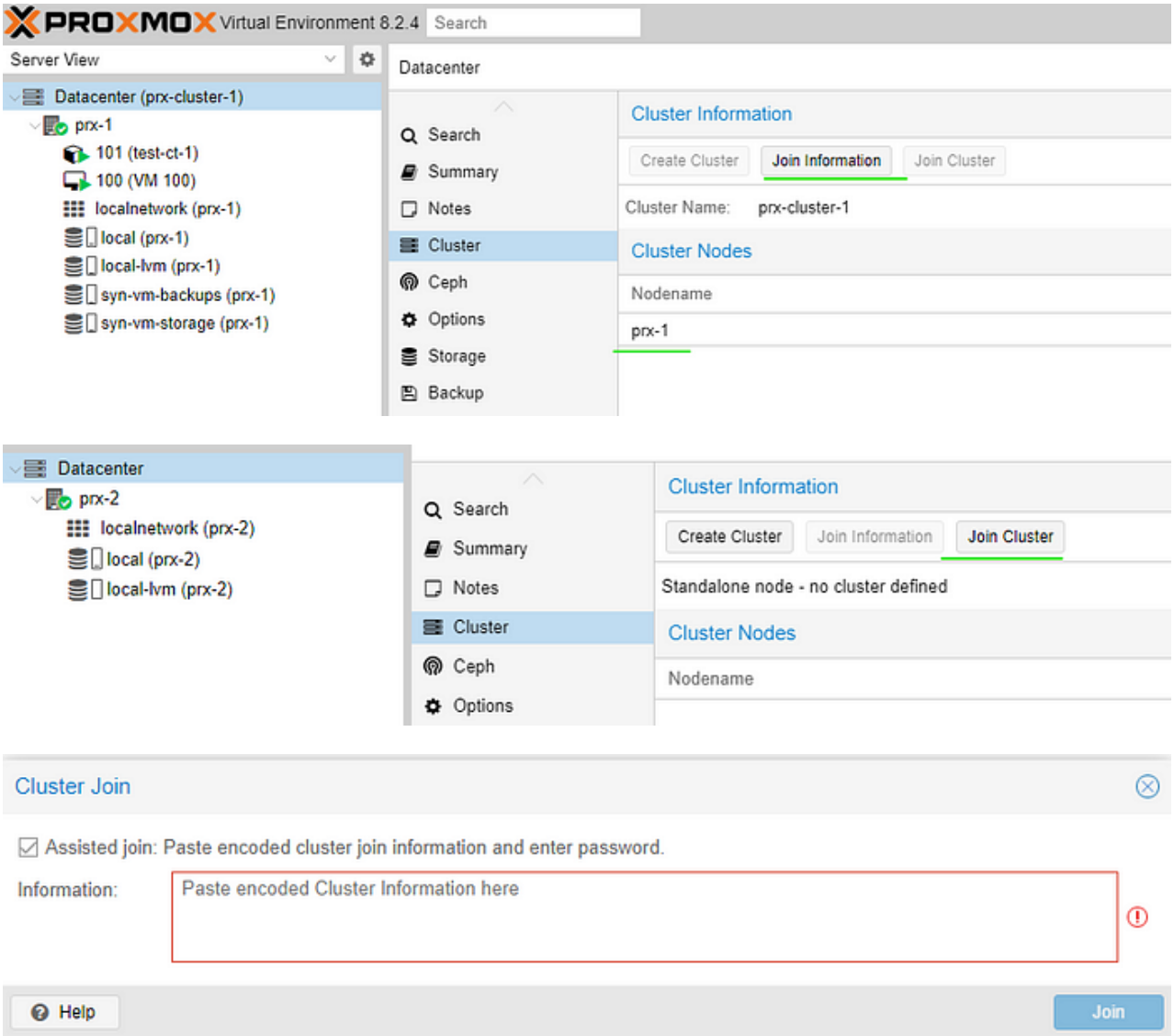
Pick a cluster name and give it node IP:

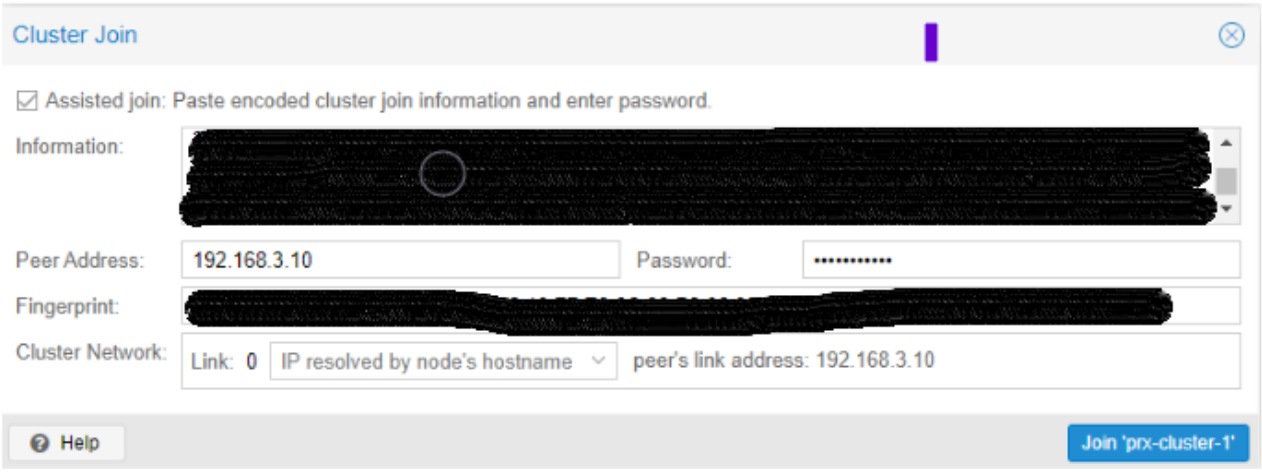


Now we have a cluster with one node. By clicking on Join Information button, we can grab information that we need to take to other nodes to add them to same cluster:

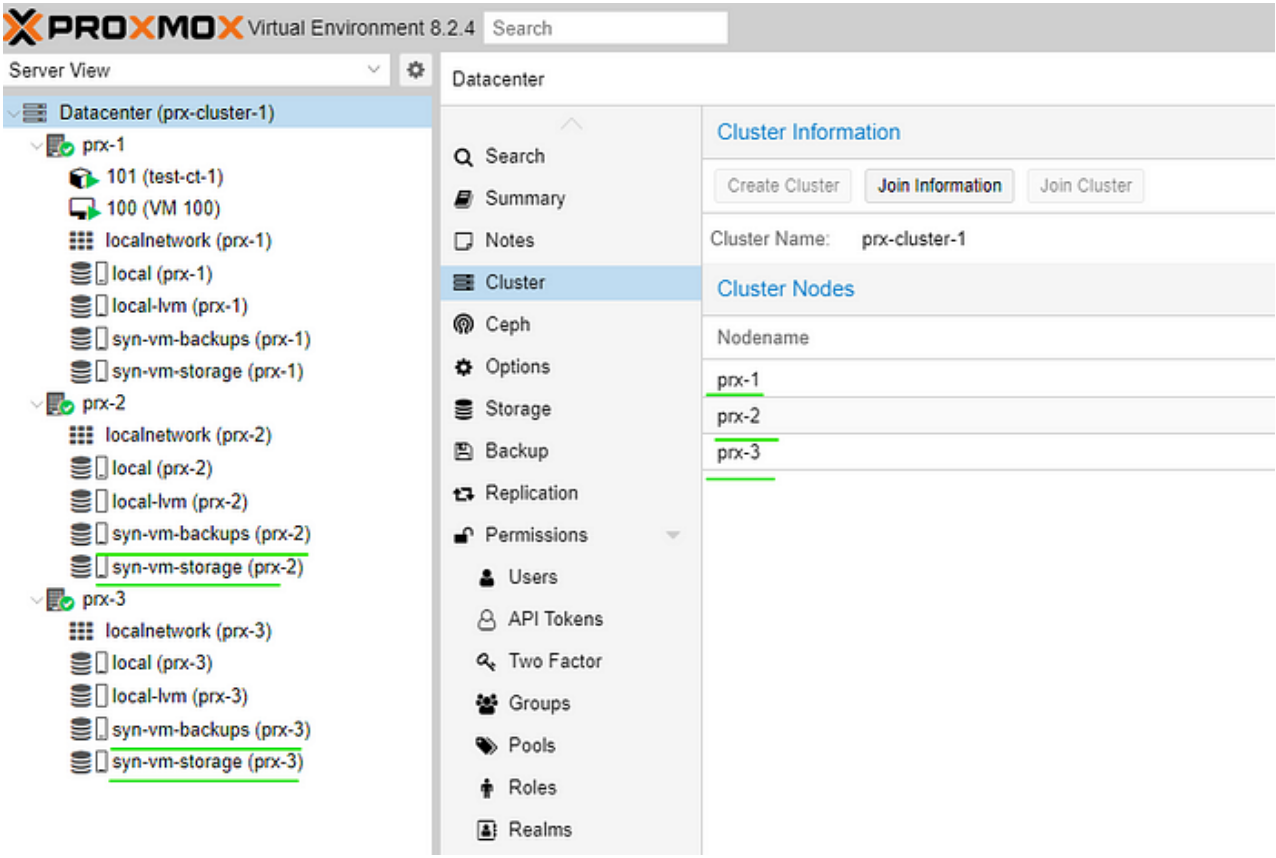


Create Cluster — Output





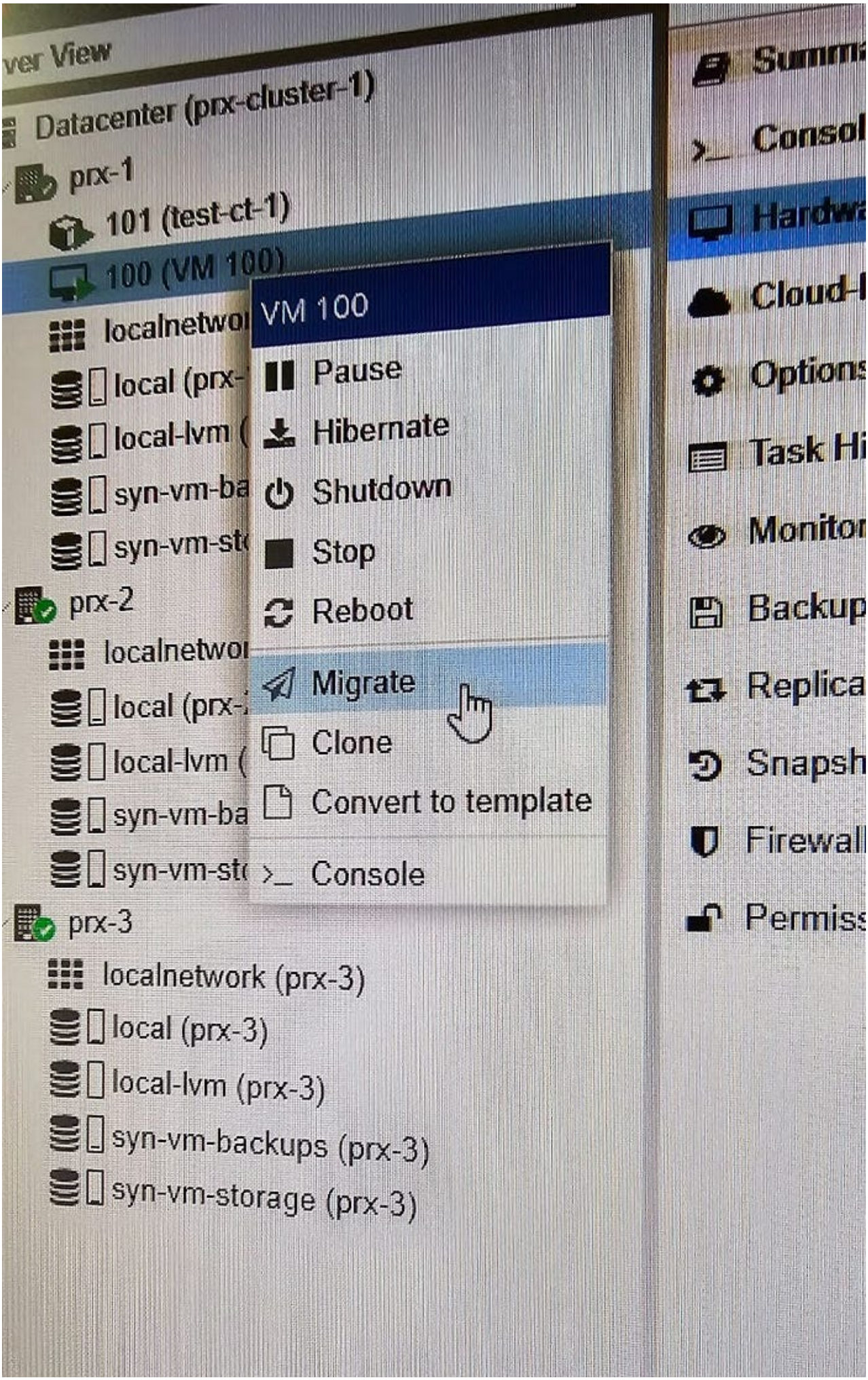
Now we can refresh page and see all nodes in cluster:

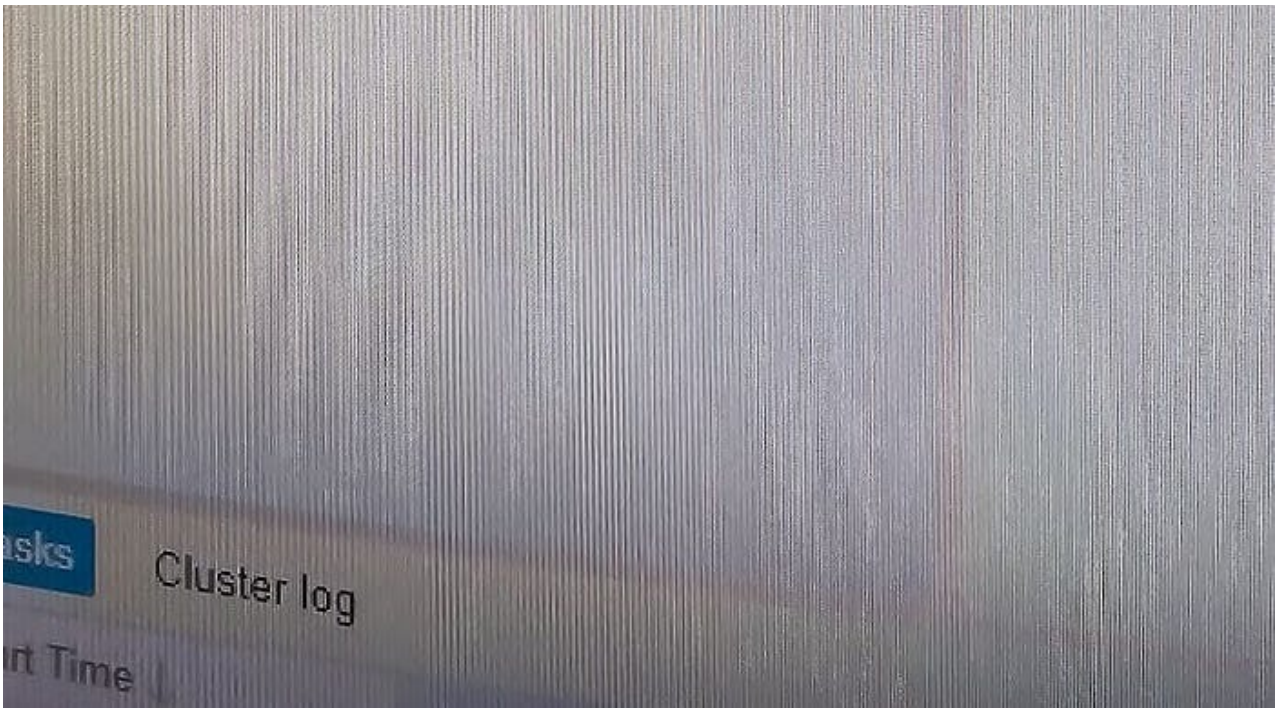


Also, in above we can see that Shared storage is accessible to all nodes even though we added it to only one node.

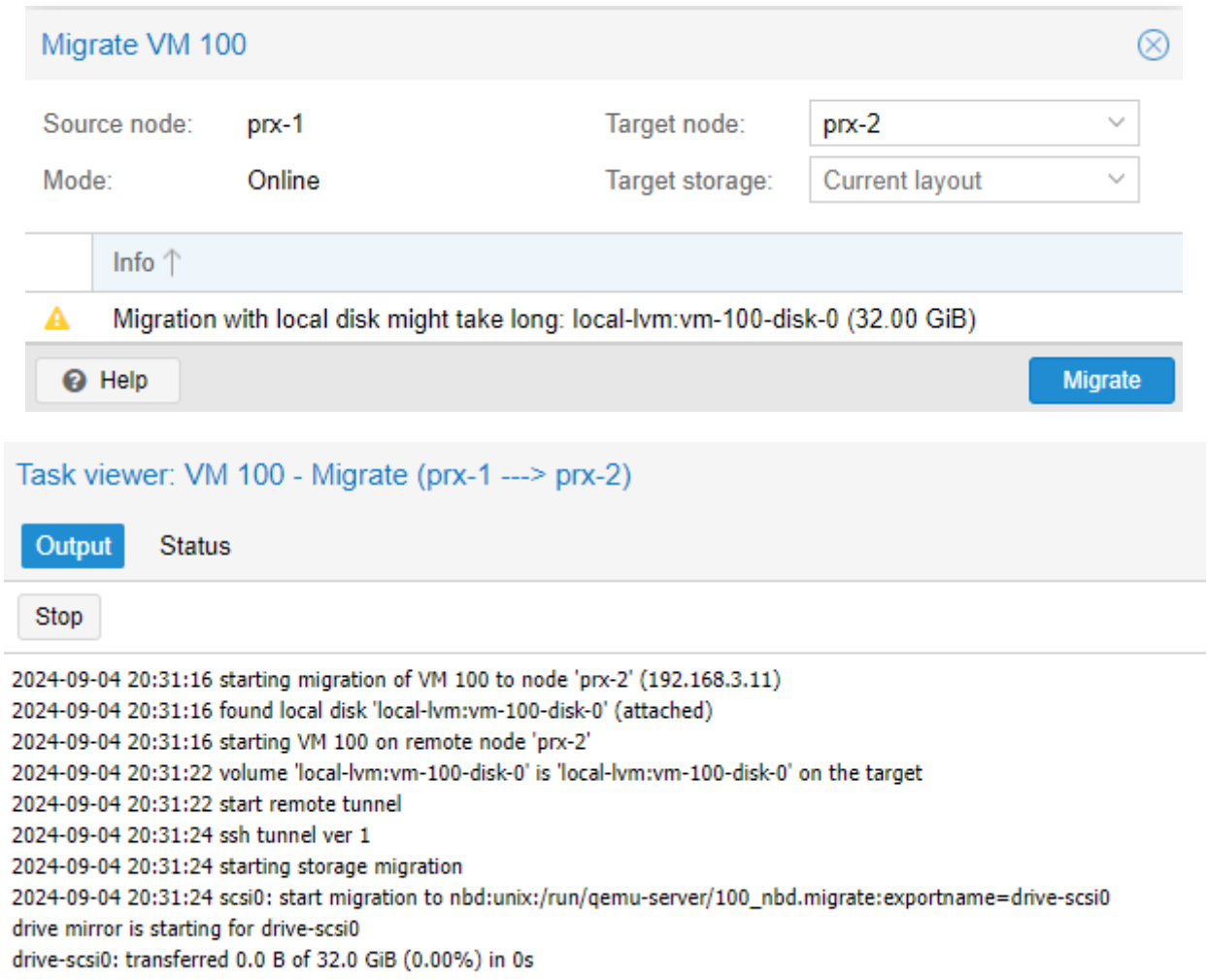
Virtual Machine Migrations

We can test our cluster by Migrating VM:


ver View
Datacenter (prx-cluster-1)
prx-1
101 (test-ct-1)
100 (VM 100)
localnetwor
local (prx-
local-lvm (
syn-vm-ba
syn-vm-st
prx-2
localnetwor
local (prx-
local-lvm (
syn-vm-ba
syn-vm-st
prx-3
localnetwork (prx-3)
local (prx-3)
local-lvm (prx-3)
syn-vm-backups (prx-3)
syn-vm-storage (prx-3)
VM 100
Pause
Hibernate
Shutdown
Stop
Reboot
Migrate
Clone
Convert to template
Console
Summ
Consol
Hardwa
Cloud-I
Options
Task Hi
Monitor
Backup
Replica
Snapsh
Firewall
Permiss



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

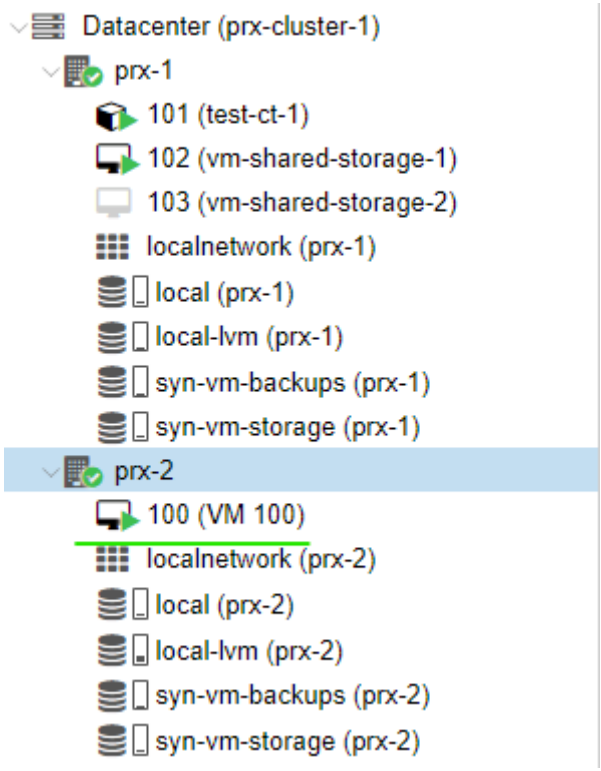


It took a few minutes to migrate over 1Gb network. The VM was responsive during this time.

```
--- 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
```

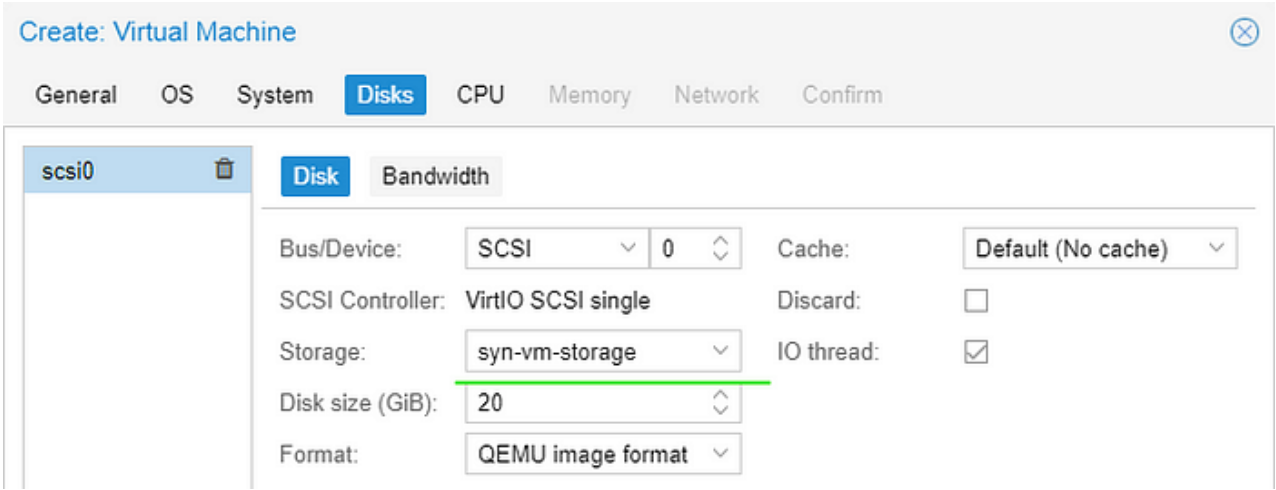
Now we can see that VM has move to prx-2 node:

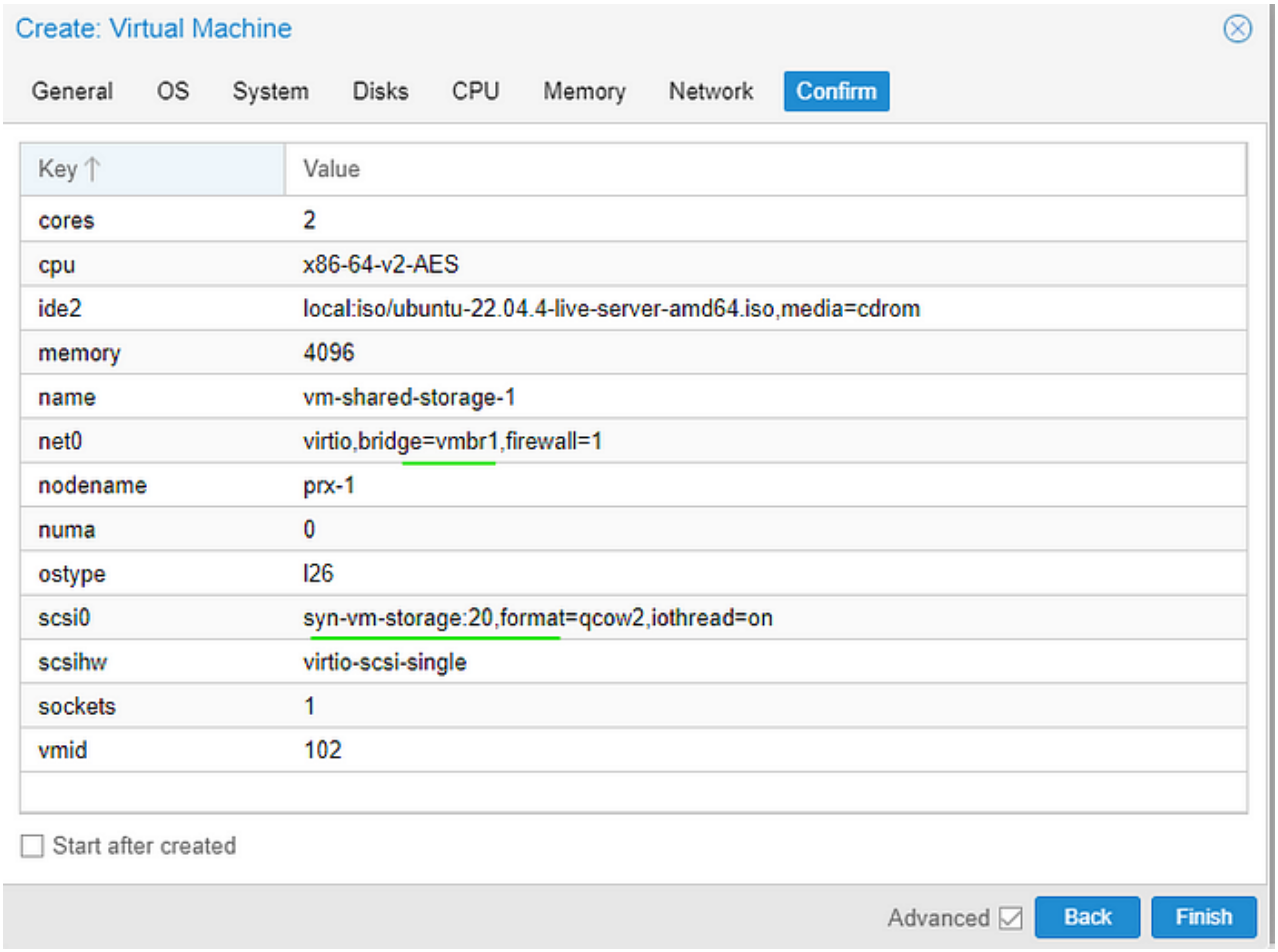
We can also verify from logs at bottom of screen:



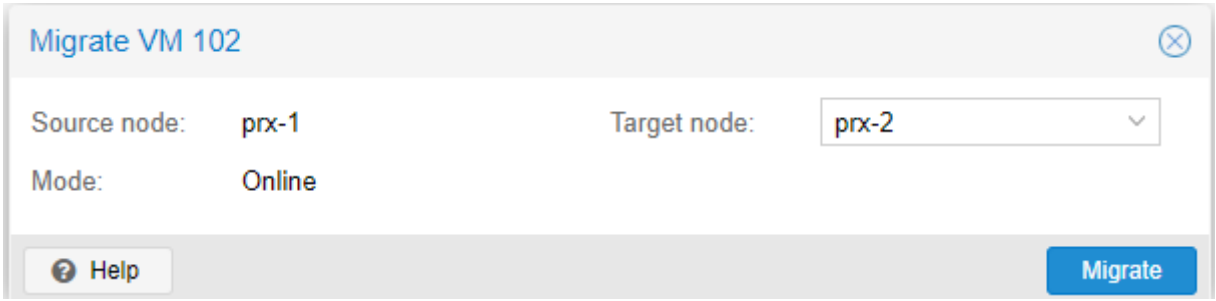
Node	User name	Description	Status
prx-3	root@pam	VM/CT 102 - Console	OK
prx-2	root@pam	VM 100 - Start	OK
prx-1	root@pam	VM 100 - Migrate	OK

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

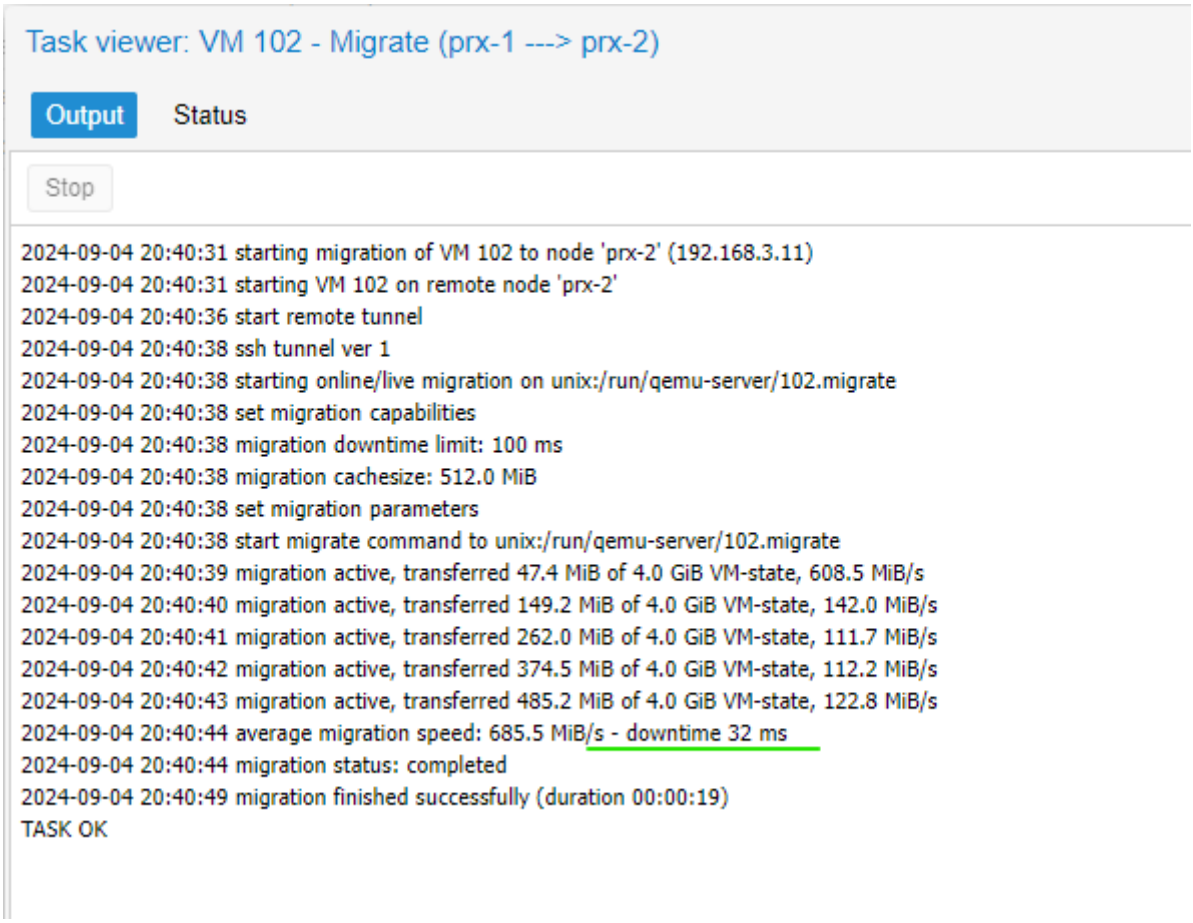


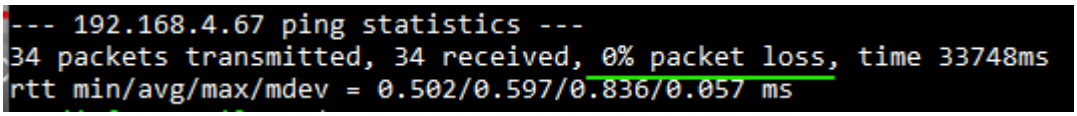


Migrate VM with shared storage:



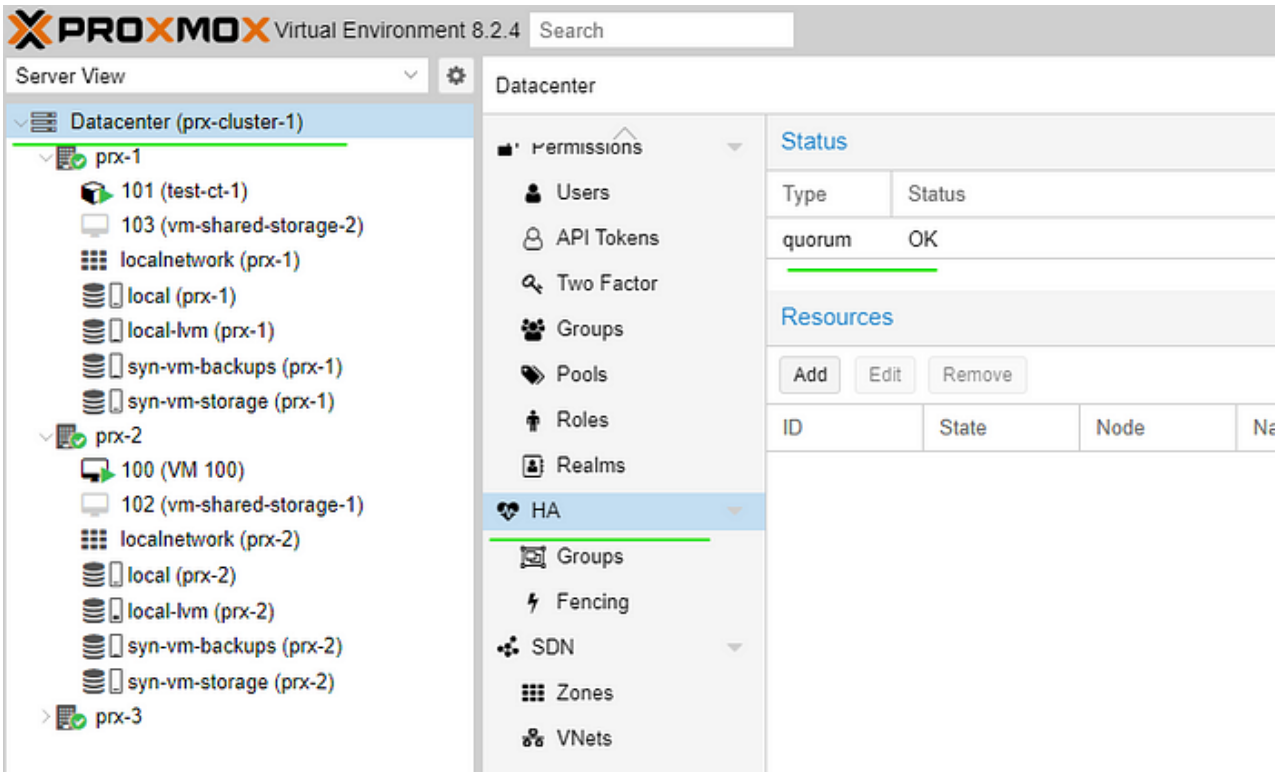
It is very quick:



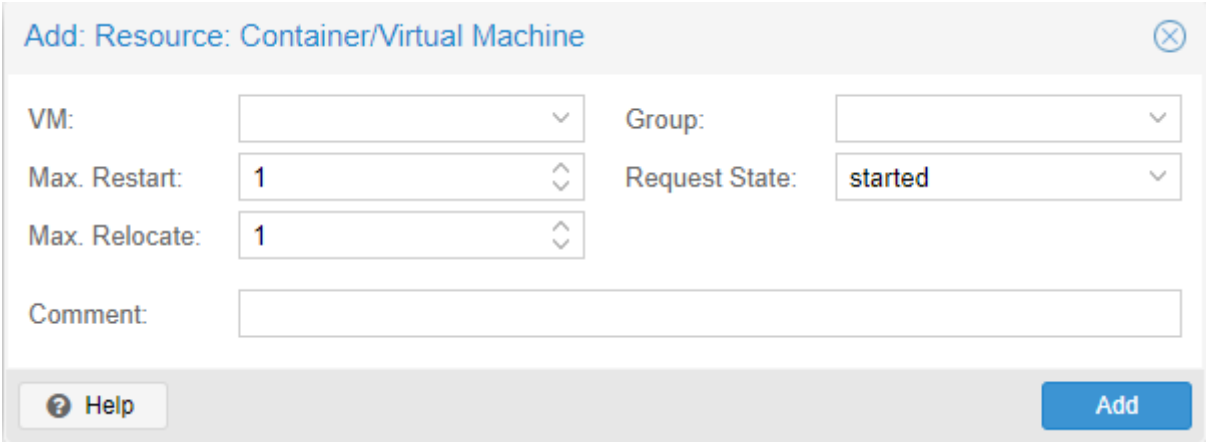


High Availability

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.

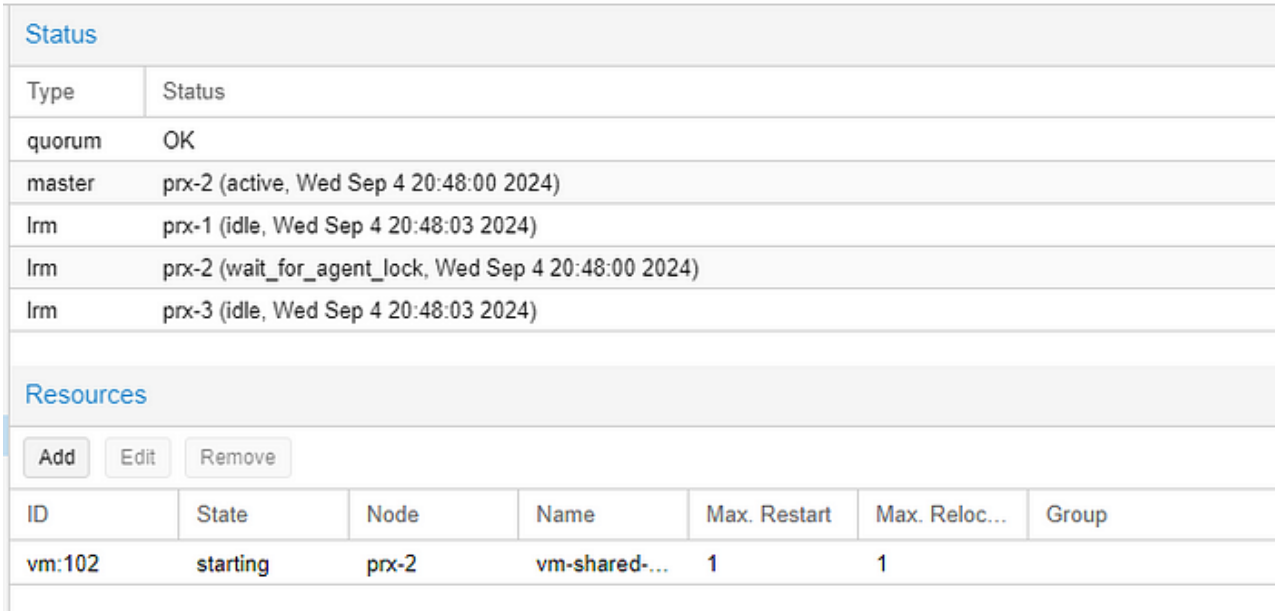


Click on Add under Resources
And select VM to add to HA



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!