01 / BEFORE YOU START - VIDEO 0:00-5:22

Three PCs. One cluster.

Build a basic three-node Proxmox VE cluster, then move a VM between hosts. Tech without the technobabble.

What we are building

One management view for three Proxmox nodes, with manual VM migration. This does not combine their RAM into one enormous computer, enable automatic load balancing or configure high availability.

1. Prepare the two new computers first

Install Proxmox VE on PVE2 and PVE3 before following the cluster steps. Keep both joining nodes empty: no VMs or containers. PVE1 is the starting node and can retain its existing guests.

Use wired, reliable networking; final static IPs and unique hostnames; compatible, updated Proxmox versions; root access; and enough free RAM and disk capacity on migration targets. Allow inter-node TCP 22 and 8006 and UDP 5405-5412 on your trusted network. Do not expose these ports to the internet.

Back up before clustering

Make verified backups of the existing guests on separate storage. Joining a cluster overwrites the joining node's /etc/pve configuration and inherits cluster storage settings. Do not join a populated server and hope for the best.

Update each node through **Node > Updates**, using the repositories appropriate to your subscription. Finish updates and any necessary reboot before proceeding. An update still running in another tab is not a completed update.

02 / KNOW YOUR SERVERS - VIDEO 6:07

Write it down first.

Run the same commands on each node. Record the actual results rather than trusting a label on a computer.

2. Open each node's Shell

```
hostname
hostname -f
pveversion
ip -br addr
ip route
qm list
pct list
```

Look for: a unique hostname, correct full name, active management interface and the intended static address. Check the default gateway and compare Proxmox versions. PVE2 and PVE3 should list no guests; PVE1 may contain the existing guests.

> ### Example addresses used with this guide
>
> PVE1: 192.168.1.95 | PVE2: 192.168.1.96
> PVE3: 192.168.1.97
> Gateway: 192.168.1.1 | DNS: 192.168.1.2
> These are example values from this setup. Substitute your own addresses and existing hostnames.

Use **hostname** to establish the real spelling and case of every node name. The examples below use **pve1.lan**, **pve2.lan** and **pve3.lan**. Do not rename an existing node to match this sheet.

My PVE1 name / address:

My PVE2 name / address:

My PVE3 name / address:

03 / NAME RESOLUTION - VIDEO 6:47

Make names predictable.

Repeat this on all three nodes before creating the cluster.

3. Back up and edit /etc/hosts

```
cp -a /etc/hosts /etc/hosts.before-cluster
nano /etc/hosts
```

Keep the localhost and IPv6 entries. Check the existing entry for this node, then add or correct the other two. Each management address should have one consistent full name and short-name alias. Avoid conflicting duplicate entries.

```
192.168.1.95 pve1.lan pve1
192.168.1.96 pve2.lan pve2
192.168.1.97 pve3.lan pve3
```

Use your **actual existing names and domain**. The node's own management name should resolve to its management IP, not a loopback address. Do not replace the entire hosts file with the three lines above.

In nano: **Ctrl+O**, **Enter** to save; **Ctrl+X** to exit.

Check the result

```
getent hosts pve1 pve2 pve3
getent hosts pve1.lan pve2.lan pve3.lan
```

Expected: each name resolves consistently to its intended management address on every node. If results disagree, fix them before continuing.

> ### Useful distinction
>
> Hosts entries help you and SSH find the right server. Use stable IP addresses for the cluster links themselves. A DNS record does not reserve an address or prevent a DHCP conflict.

04 / TIME CHECKS - VIDEO 7:56

Keep the clocks together.

Accurate time matters. Matching time zones alone does not synchronise clocks.

4. Check time on each node

```
timedatectl
systemctl is-active chrony
chronyc tracking
chronyc sources -v
```

Look for: System clock synchronized: yes, an active time service, a selected source marked * (often ^*) and Leap status: Normal. An active service by itself does not prove synchronisation. Newly started nodes may need a little time.

If chrony is installed but stopped

```
systemctl enable --now chrony
chronyc sources -v
chronyc tracking
```

If no time source is reachable, check DNS, the configured sources and outbound NTP access (UDP 123). See the service log with:

```
journalctl -u chrony -n 30 --no-pager
```

Only if the clock needs a deliberate step

After chrony has selected a valid source, `chronyc makestep` applies the outstanding correction immediately. It can jump time forward or backwards and disturb running applications. Use it during preparation or an appropriate maintenance window, then recheck tracking.

Chrony is the normal choice on modern Proxmox installations. If another NTP service is intentionally installed, inspect that service instead; do not run two competing time daemons. Nodes need accurate common time, not identical time-zone labels.

05 / REPEATABLE PREFLIGHT - VIDEO 10:12

Same checks. Three times.

Run this block on PVE1, then PVE2, then PVE3. Adapt the names first.

5. Confirm the basics

```
hostname
pveversion
getent hosts pve1 pve2 pve3
ping -c 3 pve1
ping -c 3 pve2
ping -c 3 pve3
timedatectl
chronyc tracking
qm list
pct list
pvecm status
```

Names: correct addresses, consistently on all nodes.
Pings: replies from all three nodes with no loss on a healthy LAN. Ping does not test cluster or SSH ports.
Time: synchronised; investigate errors before joining.
Versions: compatible and updated; use the same release for this build.
Guests: joining nodes empty; PVE1's existing guests accounted for.

Before the cluster exists

A pvecm status error about missing Corosync configuration is expected on a standalone node. After joining, that same error is a problem. If a node already belongs to a different cluster, stop and investigate; do not join it blindly.

Check that you can open each node's management interface at **https://NODE-IP:8006**. If firewall rules are enabled, verify the required inter-node ports from page 1. A green ping is encouraging; it is not a networking degree.

06 / CREATE AND JOIN - VIDEO 11:41-14:59

Create once. Join twice.

Start on the populated node, PVE1. Add the two empty nodes one at a time.

6. Create the cluster on PVE1

Log in to PVE1. Select **Datacenter > Cluster > Create Cluster**. Choose a unique cluster name, for example **home-cluster**. Confirm that Link 0 uses PVE1's intended cluster-network address. Create the cluster and wait for the task to complete.

The cluster name is a long-term choice. Avoid renaming nodes or changing management IPs as part of this procedure.

7. Join PVE2

On PVE1, open **Datacenter > Cluster > Join Information** and copy the information. Log in directly to PVE2, choose **Datacenter > Cluster > Join Cluster**, paste it and enter PVE1's root password. Check the destination and PVE2's Link 0 address. Join and wait for completion.

> ### A browser session can disconnect
>
> Joining replaces the node certificate with one signed by the cluster CA. Refresh and log in again when needed. Check task and cluster status before retrying; a dropped browser session does not prove the join failed.

8. Join PVE3

Repeat the join procedure on the empty PVE3. Do not create another cluster there. Confirm all three nodes appear under Datacenter before moving on.

Joining inherits the cluster storage configuration. Under **Datacenter > Storage**, check node restrictions and actual backing storage. An identically named storage entry does not magically create a disk or a shared filesystem.

07 / VERIFY THE CLUSTER - VIDEO 14:59

Check the votes.

Run the following on each node and compare the results.

9. Verify membership and services

```
pvecm status
pvecm nodes
systemctl is-active corosync
systemctl is-active pve-cluster
```

Expected in this basic three-node setup:
Nodes: 3
Expected votes: 3
Total votes: 3
Quorum: 2
Quorate: Yes
Three node names and their correct addresses; the local marker changes depending on where you run the command.

Quorum is a majority

With one vote per node, any two of these three nodes provide a majority. One node alone does not. A two-node cluster is possible, but has different failure behaviour; an external QDevice can supply a deciding vote. A Raspberry Pi used as a QDevice is a voting helper, not a Proxmox VM host.

If something is missing

```
journalctl -u corosync -n 50 --no-pager
journalctl -u pve-cluster -n 50 --no-pager
```

Check the missing node's power, network, firewall, addresses and time. Do not lower expected votes as a routine workaround. Restore communication and a healthy majority before making cluster changes.

A cluster gives shared management and configuration. It is not a backup, and this guide does not enable HA. Keep verified guest backups on separate storage.

08 / FIRST MIGRATION - VIDEO 16:08-17:33, 19:43

Move one small VM.

Start with a non-critical test guest. The video demonstrates moving Pi-hole; plan DNS disruption before moving yours.

10. Check the destination first

Confirm spare disk space and RAM, compatible VM CPU settings, and matching network bridges/VLAN connectivity. Local disks must be copied across the network. Similar storage names are not proof of compatible storage.

Open **VM > Hardware > CD/DVD Drive > Edit**. If a local installer ISO is still attached and not needed, select **Do not use any media**. Review USB/PCI passthrough and other local devices; they can prevent migration or require deliberate changes. Do not detach a device your guest needs.

11. Migrate and verify

For a simple first test, shut down the guest cleanly and wait until it is stopped. Select **Migrate**, choose the target node, review storage options and start the task. Wait for a successful task result; check that the guest appears on the target, then start it.

Open its console and application, check network access and confirm that it serves its normal purpose. For Pi-hole, test DNS from a client. For Home Assistant, test integrations and any hardware connections.

> ### Live migration is conditional
>
> A running VM can migrate where CPU compatibility, storage, networking and device requirements allow it. It is not guaranteed for every guest. Local-storage migration can take longer because disk data also travels across the network.

Keep a record of the source, destination, guest ID and result. Do not shut down the source host until the migrated services have been checked.

09 / TROUBLESHOOTING - VIDEO 17:39-19:38

When migration says no.

Use the actual error message. Avoid turning a small problem into an interesting weekend.

Local CD/DVD or local resource error

Detach an unused local ISO, or ensure the required resource is genuinely available on the destination. USB and PCI passthrough require their own plan. Moving Zigbee2MQTT does not move its USB radio.

Permission denied (publickey,password)

Check cluster health, the target address and SSH service first. Inspect the migration task log and the destination's SSH log. Authentication failures have several possible causes; use the logs to identify the affected node and key before changing anything.

```
pvecm status
systemctl is-active ssh
journalctl -u ssh -n 30 --no-pager
```

The documented `pvecm updatecerts` command refreshes node certificates and required cluster files, and is relevant to some stale node-identity situations. Use it on the affected node only when diagnosis supports that remedy. Do not delete SSH keys or disable host-key checking to silence an error.

Storage unavailable, or insufficient space

Inspect **Datacenter > Storage**, node restrictions and the target's real capacity. Use `pvesm status` to check active storage. Do not mark local disks as shared just to make a warning disappear.

Guest starts but its service fails

Check bridges, VLANs, IP configuration, attached hardware and the application itself. Keep an independent way to resolve node names if you are moving the VM that supplies DNS.

10 / FINISH AND KEEP - VIDEO 21:40

A working cluster.

The useful outcome is three healthy nodes and a tested move, not merely three names in a sidebar.

- ☐ All three nodes appear and report Quorate: Yes.
- ☐ Name resolution and time checks pass on each node.
- ☐ Storage availability and node restrictions are correct.
- ☐ A small VM migrated successfully and its application works.
- ☐ Important guests have verified backups outside their host.

Clustering, migration and HA are different

This walkthrough sets up clustering and manual migration. HA is a separate design that restarts managed guests after failures; it needs quorum, fencing and access to suitable guest storage. Shared storage is one approach; supported storage replication is another, with recovery-point trade-offs. HA does not preserve a failed host's running memory.

Video and official references

Watch the cluster walkthrough

Proxmox: Cluster Manager, joining and quorum
Proxmox: VM management and migration
Proxmox: High Availability
Proxmox: Storage Replication
Chrony: tracking, sources and makestep

Video checkpoints: 6:07 server details; 6:47 hosts; 7:56 time; 10:12 checks; 11:41 create; 12:37 join; 14:59 quorum; 16:08 migration preparation; 17:39 troubleshooting; 19:43 successful migration demonstration.

As an Amazon Associate I earn from qualifying purchases.