

initMAX

Webinar

Proxmox Cluster: Clustering, HA and Storage

all your microphones are muted

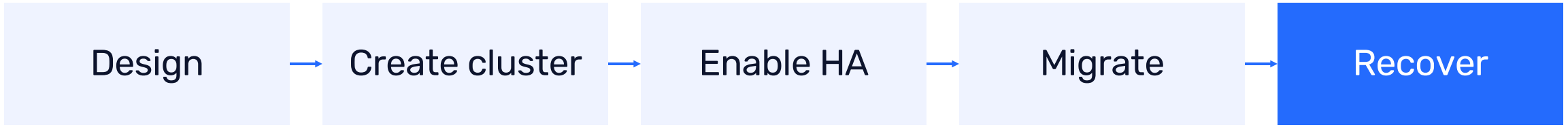
ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

initMAX

Today’s route

Clustering, HA and storage in one workflow

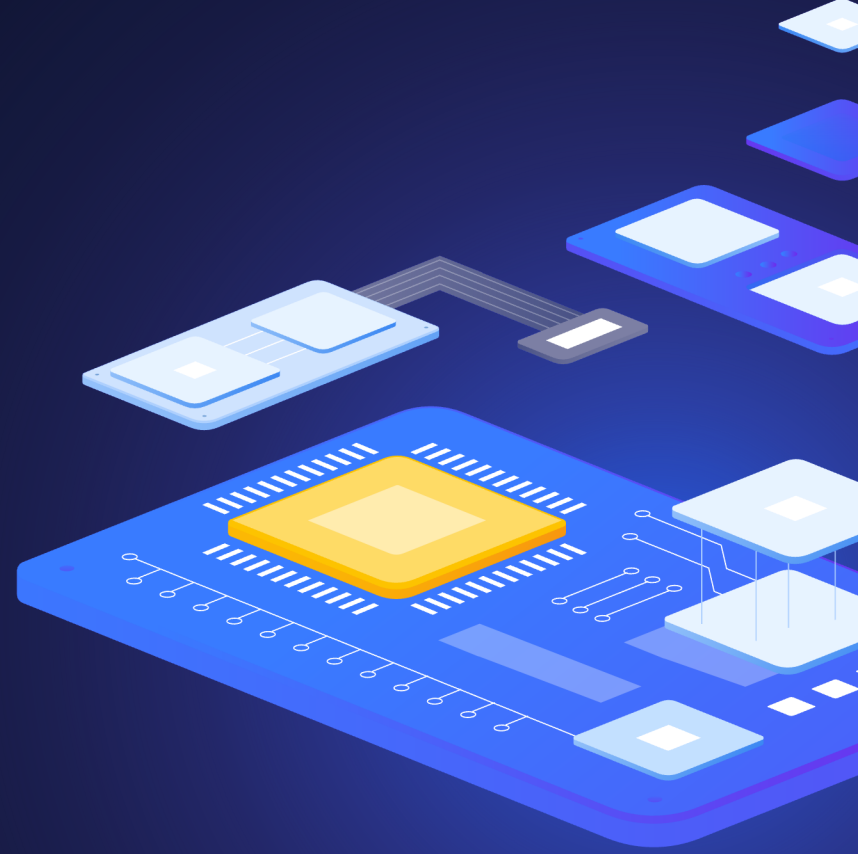


The same questions at every step

Where are the disks? Who may start the VM? Can the target host run it?

1

Cluster design and architecture



What the cluster actually shares

Configuration

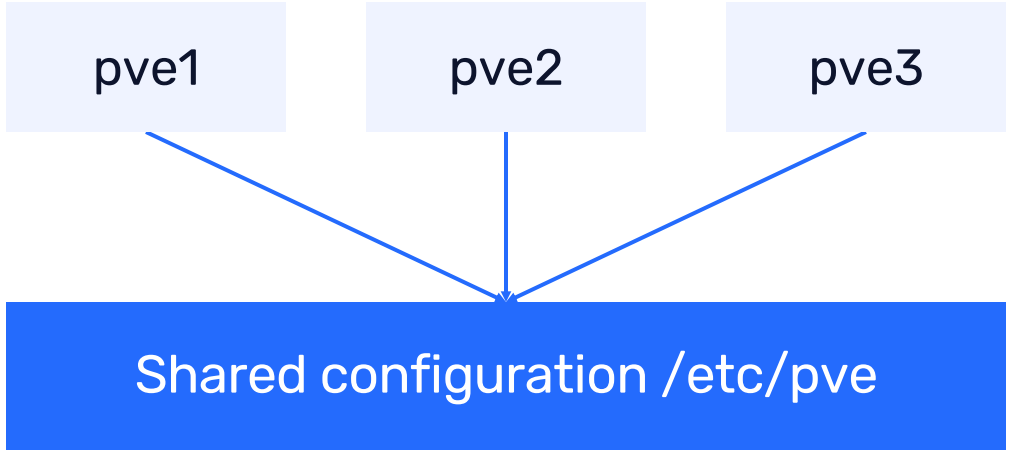
Corosync and pmxcfs distribute cluster state and configuration.

Guest disks

Their location follows the selected storage architecture.

HA resources

A VM or CT becomes managed only after it is added to HA.



VM disks follow the storage design

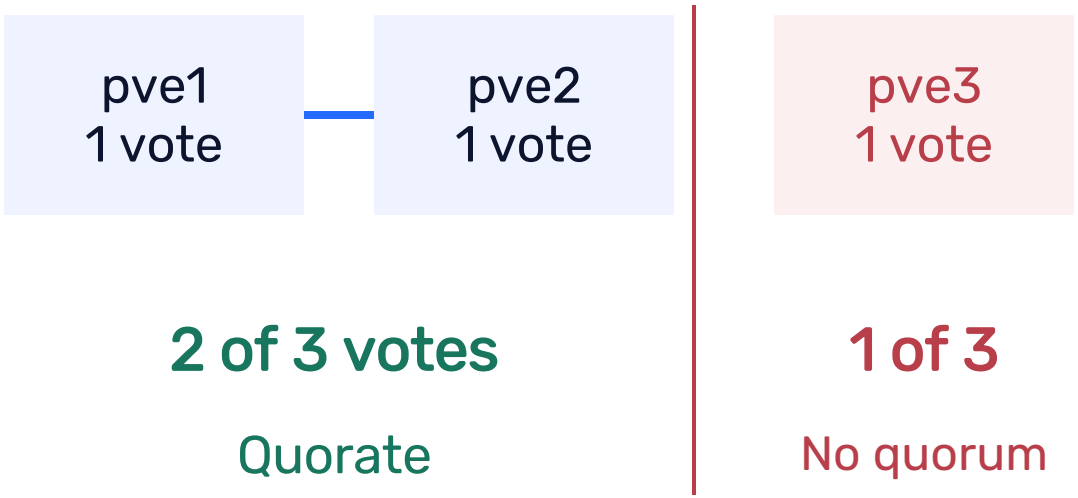
Quorum and a network partition

Majority of expected votes

A partition needs more than half of all expected votes to retain quorum.

Three nodes, one vote each

Two connected nodes retain quorum. The isolated node cannot form a second quorate cluster.



initMAX

Node count and quorum tolerance

Nodes	Votes required	Node losses tolerated
2	2 of 2	0
3	2 of 3	1
4	3 of 4	1
5	3 of 5	2

One vote per node, no QDevice. Storage and HA capacity are separate checks.

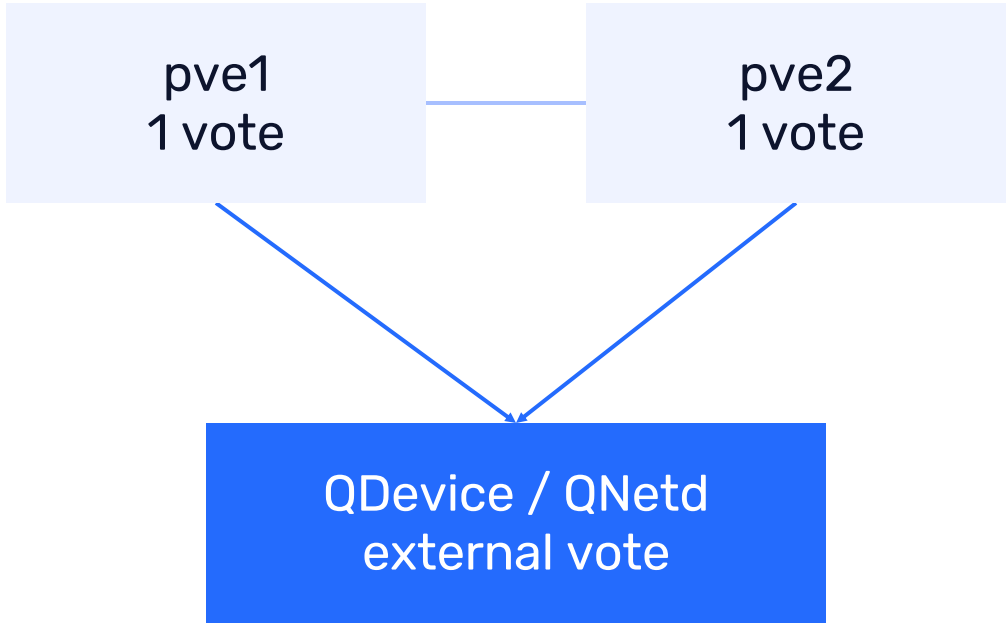
Two nodes with an independent QDevice

One additional vote

The surviving node needs the external arbitrator to retain a majority after its peer fails.

Independent placement

Keep the QNetd host outside the two-node failure domain. It stores no VM disks.



initMAX

Cluster design and architecture

Corosync quorum and Ceph quorum

Corosync

Cluster membership and configuration.
QDevice affects this voting system.

PVE quorum
Corosync votes

Ceph MON and OSD

MONs maintain storage maps. OSDs store objects and their replicas.

Storage quorum
Ceph monitors

Check both independently

Cluster design and architecture

Capacity after a host failure

Illustrative three-host design	RAM
Installed: 3 × 128 GiB	384 GiB
After losing one host: 2 × 128 GiB	256 GiB
Host reserve: 2 × 16 GiB	−32 GiB
Remaining guest budget in this example	224 GiB

Also budget CPU, storage IOPS and network traffic during recovery.

initMAX

Storage architecture

Three storage patterns for HA

Pattern	Guest disk availability	Main design question
Local ZFS + replication	Copy on configured target nodes	How much recent data may be lost?
Ceph RBD	Distributed pool across hosts	Can storage stay healthy after failure?
External shared storage	Same volume reachable by hosts	Are array, paths and switches redundant?

HA requires usable disks on an eligible target, plus enough compute capacity.

Storage architecture

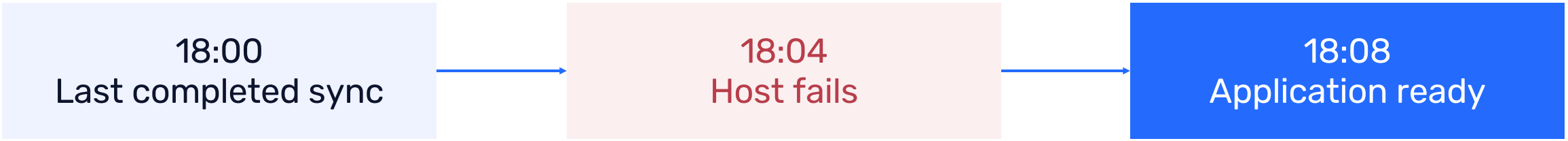
Recovery point and recovery time

RPO: recent changes at risk

How far back must the recovered data go after the failure?

RTO: time to restore the service

How long until users can work with the application again?



4 minutes of changes at risk

4 minutes to recover

Illustration for asynchronous replication. Actual objectives depend on the application.

initMAX

Storage architecture

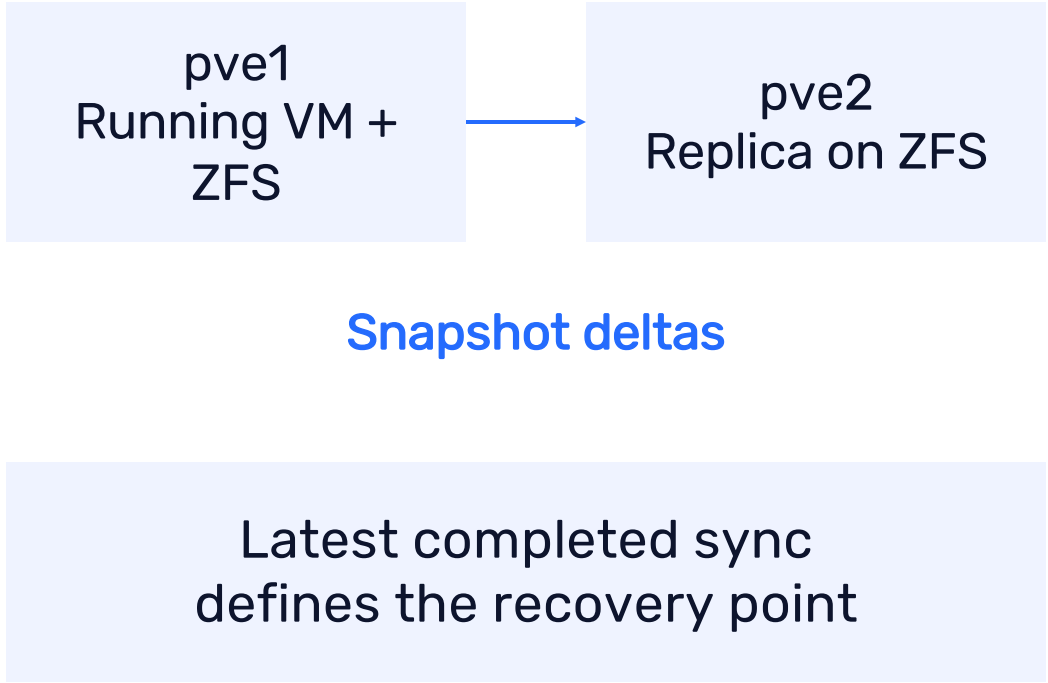
Local ZFS replication and recovery point

Scheduled, asynchronous copy

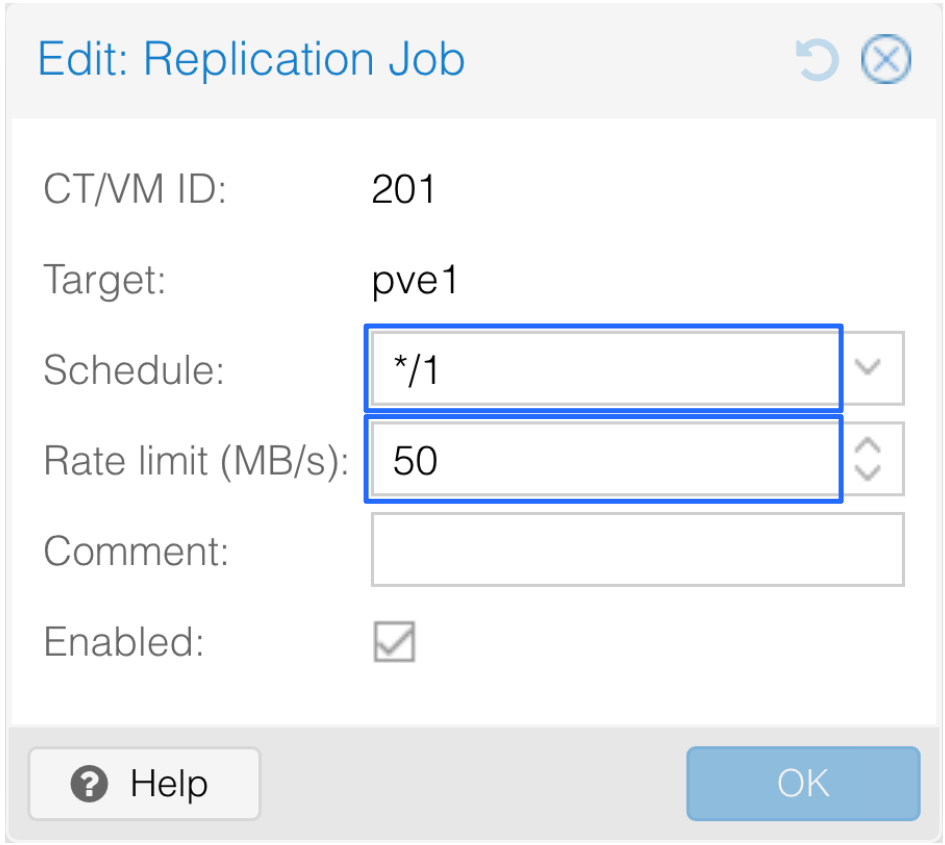
After a host fails, recovery uses the last completed replication on the target.

RPO follows actual job health

A one-minute schedule is not a one-minute guarantee if jobs fail or run late.



Replication job: target and schedule



Target node

VM 201 on pve2 replicates to pve1 in this view.

Schedule

*/1 requests a run every minute. Monitor completion.

Bandwidth

The 50 MB/s value is a lab limit, not a production default.

initMAX

Replication status: last successful run

Add | Edit | Remove | Log | Schedule now

Enabled	Guest	Job	Target	Status	Last Sync	Dur…	Next Sync
✔	201	0	pve1	✔ OK	2026-11-11 17:49:19	6.1s	pending

Last sync

Read the last successful completion.

Next sync

A schedule alone does not prove a fresh copy.

Failures

Alert on failed or delayed jobs.

initMAX

Storage architecture

Ceph sizing: hosts, monitors and disks

Layer	Baseline / recommendation	Purpose
Storage hosts	At least 3 for Proxmox HCI Ceph	Independent host failure domains
Ceph MON	3 MONs, or 5 for two MON failures	Majority agreement on cluster maps
Data disks / OSDs	Proxmox recommends at least 12 OSDs	Evenly distributed data capacity and I/O
Pool replicas	Example: size = 3, min_size = 2	Data copies and the I/O threshold

QDevice affects Proxmox quorum. It adds no Ceph monitor, disk or data replica.

initMAX

Storage architecture

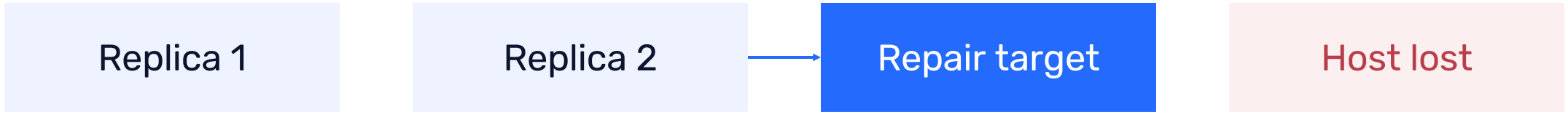
Three Ceph hosts and room for recovery

Three hosts: one object, three host placements

Replica 1	Replica 2	Host lost

Two copies remain.
No third surviving host for repair.

Four hosts: a surviving target for the third copy



Requires enough free capacity and recovery bandwidth on the surviving hosts.

Example: replicated size = 3, min_size = 2, failure domain = host. MON quorum is separate.

Storage architecture

Ceph replicas across failure domains

Replicated pool example

size = 3, min_size = 2, with replicas placed on different hosts.

Three hosts are a starting point

Plan spare capacity and recovery bandwidth. More free disks on two hosts do not create a third host.



One object, three host placements

initMAX

Storage architecture

Ceph recovery after a host failure

Illustration: a healthy replicated pool, one host lost, size = 3 / min_size = 2

Host lost OSDs down	→	Peering Replica checks	→	OSDs out New placement	→	Recovery / backfill Copies rebuilt
Affected PGs lose redundancy when OSDs become unreachable.		I/O may pause, then resume with enough usable replicas.		Timing and policy determine when CRUSH remaps the data.		Rebuild missing copies on eligible OSDs with free space.

Proxmox HA can recover the VM while Ceph repairs

After safe fencing, HA can restart the VM when quorum, disks and an eligible host with enough capacity are available.

Storage repair and application recovery have separate completion times.

initMAX

Storage architecture

Ceph capacity after losing one host

Stage	Raw capacity	Raw data	Use
Before failure	40 TiB	24 TiB	60%
Immediately after host loss	30 TiB	18 TiB	60%
After complete replica repair	30 TiB	24 TiB	80%

Repair adds 6 TiB. Average use rises from 60% to 80%.

Starting at 70% would require about 93% after complete repair.

Equal hosts, balanced data, three copies included. No growth or overhead in this illustration.

Storage architecture

Ceph capacity limits and recovery load

State	Typical default	Operational effect
nearfull	85%	Warns that an OSD is getting full.
backfillfull	90%	Can block backfill into that OSD.
full	95%	Can stop client writes.

Recovery competes with application I/O

Reserve disk space, CPU, RAM and network bandwidth. Check the fullest eligible OSD and the impact on VM latency.

Verify actual OSDMap thresholds. These limits are protection thresholds, not operating targets.

Storage architecture

Ceph repair and ZFS replication

Question	Ceph replicated pool	Local ZFS + PVE replication
Normal operation	Writes replicate across OSD hosts.	Snapshots transfer on a schedule.
Source host fails	Usable surviving replicas can serve I/O.	Use the last successful target replica.
Repair path	Rebuild copies on eligible OSDs.	A failed source cannot send new updates.
Capacity needed	Space on eligible OSDs and hosts.	Space for replicas, snapshots and deltas.

Both designs require an independent backup and a tested application recovery procedure.

Storage architecture

Ceph health before maintenance

Training environment: 4 hosts, 8 OSDs, 3 monitors

Status



HEALTH_OK

Ceph Version: 20.2.5

OSDs

	In	Out
Up	8	0
Down	0	0

Total: 8

Monitors

pve1: ✔ pve2: ✔ pve3: ✔

Health

Read the health detail before maintenance.

OSDs

Check up/in state and active recovery.

Monitors

Three MONs have their own quorum.

Storage architecture

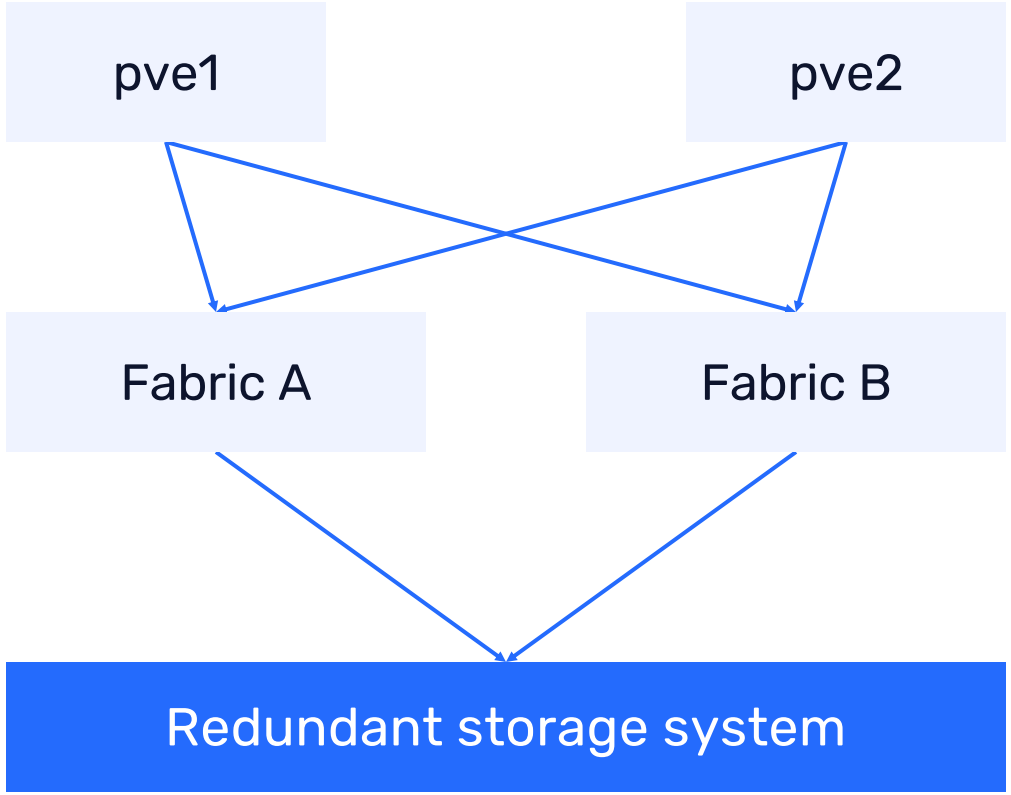
External storage and redundant paths

The array is part of the HA design

Protect controllers, switches, cabling and power, not only the hypervisor.

Protocol support matters

Use a supported storage plugin and validate its actual snapshot, migration and backup behavior.



Network architecture

Network traffic by purpose

Management	Administration and APIs
Corosync	Small, latency-sensitive traffic
VM networks	Consistent VLANs / bridges
Storage / replication	Sustained and recovery I/O
Migration / backup	Large transfers and bursts

initMAX

Network architecture

Corosync links and physical redundancy

Link 0 and link 1

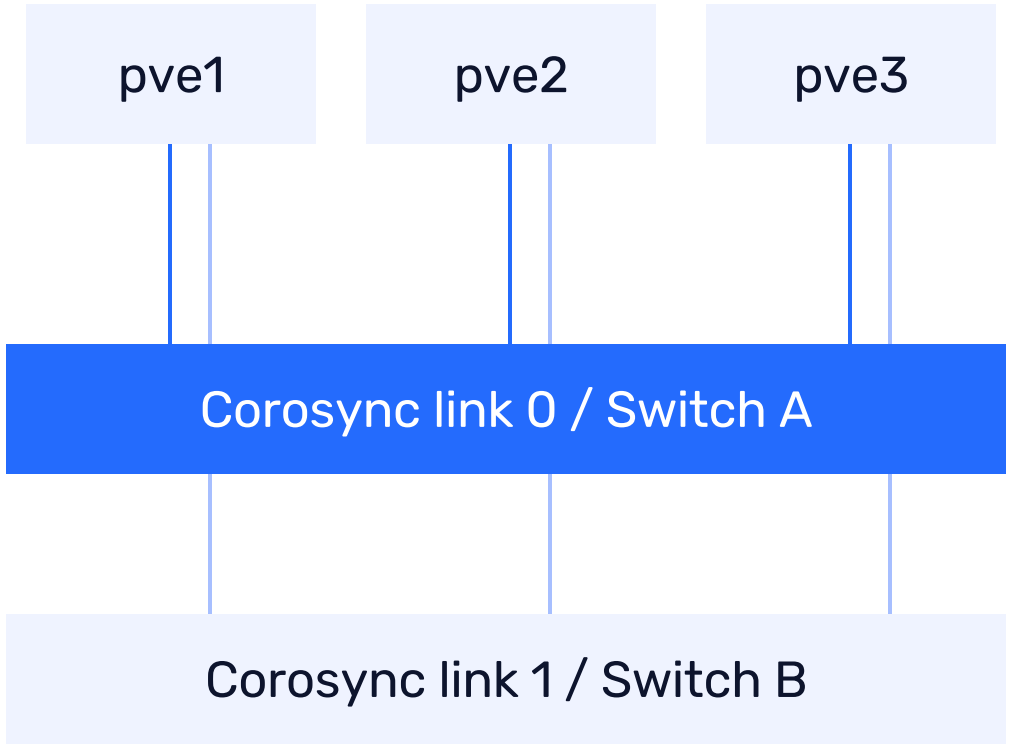
Separate physical paths reduce shared failure points.

Low latency under load

Test with storage recovery, backup and migration traffic.

Bond and VLAN limits

A bond or second VLAN may still share the same switch.



Network architecture

Cluster network checks

Service	Path to verify
Corosync	UDP 5405–5412 between all cluster nodes
SSH	TCP 22 between cluster nodes
Management	TCP 8006 from approved administration networks
Latency and time	Reliable LAN, under 5 ms, synchronized clocks

Test each Corosync path while migration or storage recovery is active.

Network architecture

Bandwidth during migration and recovery

Illustrative lower bound

64 GiB of RAM over an otherwise idle 10 Gbit/s link needs about 55 seconds for one pass.

Real migration can take longer

Protocol overhead, guest memory writes and local disks add work. Multiple migrations share the same link.

Plan recovery traffic together with guest I/O, backup and maintenance windows.

initMAX

Cluster design and architecture

Design review before building the cluster

Required failure tolerance

Which host, disk, switch or site may fail without losing the service?

Recovery objectives

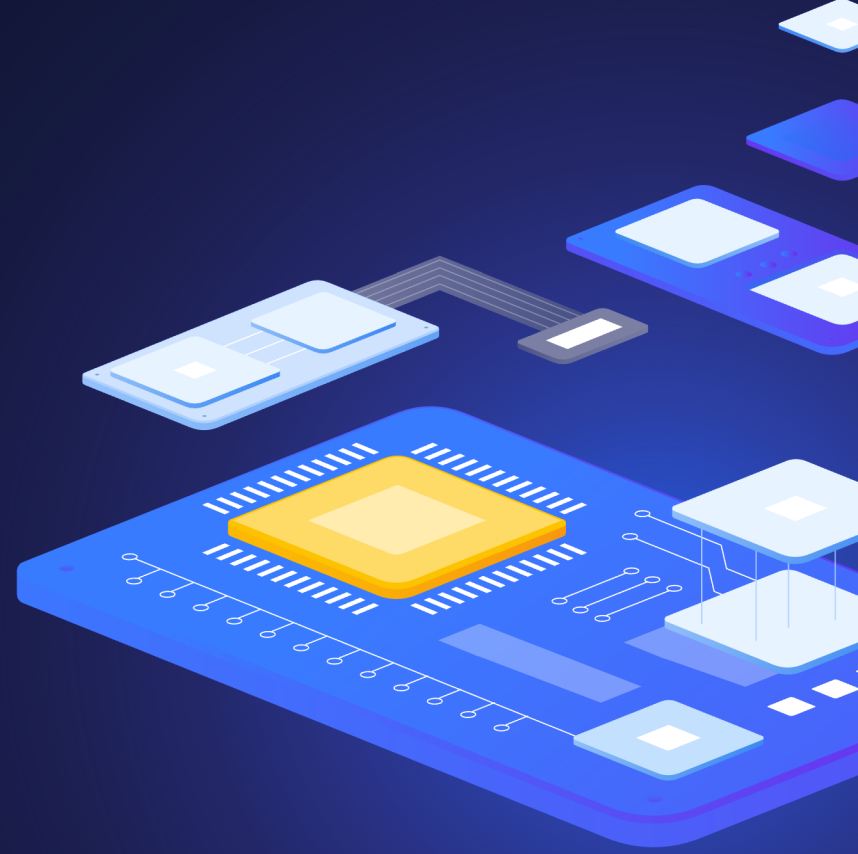
How much downtime and recent data loss can the application tolerate?

Operational proof

Check capacity, backups, monitoring and a tested recovery procedure.

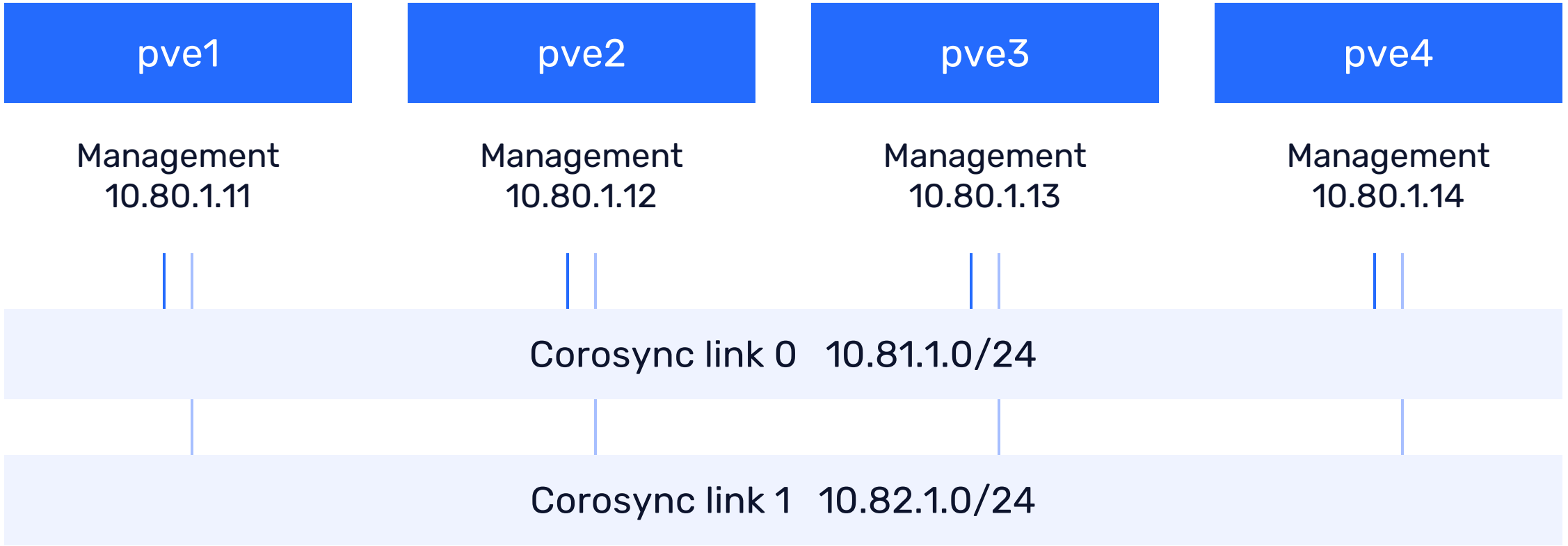
2

Create a Proxmox cluster



initMAX

The lab used in this walkthrough



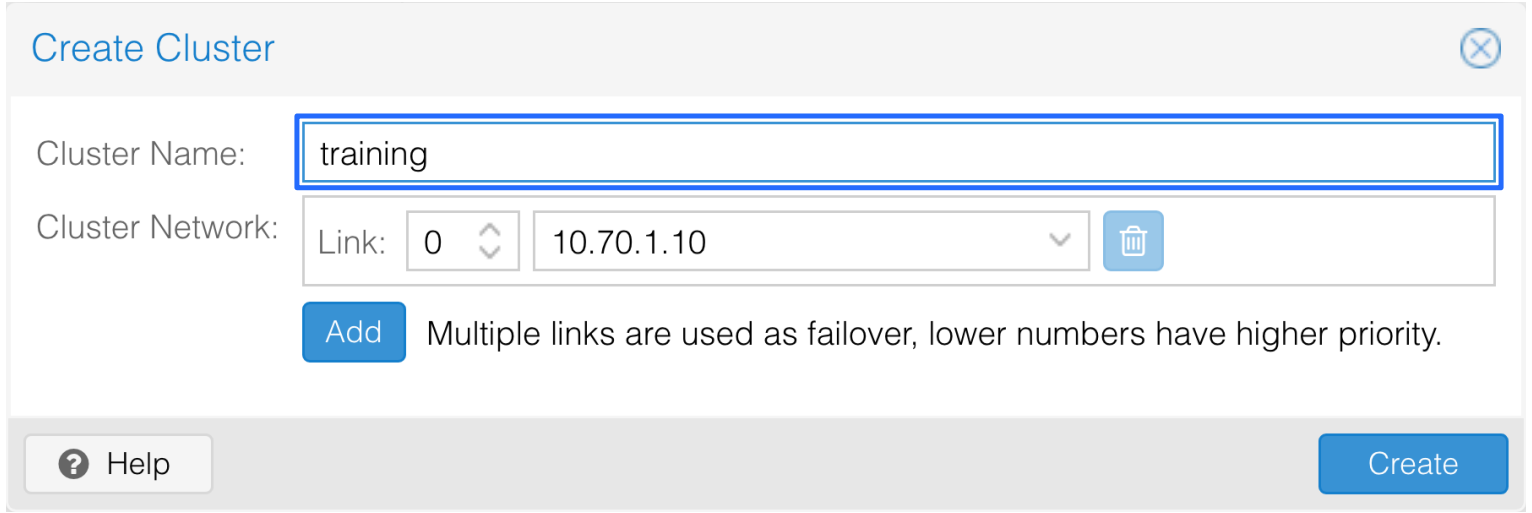
Example lab addresses. Four node votes require three for quorum.

initMAX

Preflight checks on every node

Check	Expected condition
Node identity	Unique stable hostname and correct address resolution
Version and time	Same PVE version, synchronized clocks
Cluster traffic	All node pairs reachable over the selected links
Joining node	No existing guests or conflicting VM IDs
Recovery access	Verified backup and an available console

Create the cluster on the first node



Cluster name

Choose a stable name before creating the cluster.

Local link address

Select the interface from your Corosync network plan.

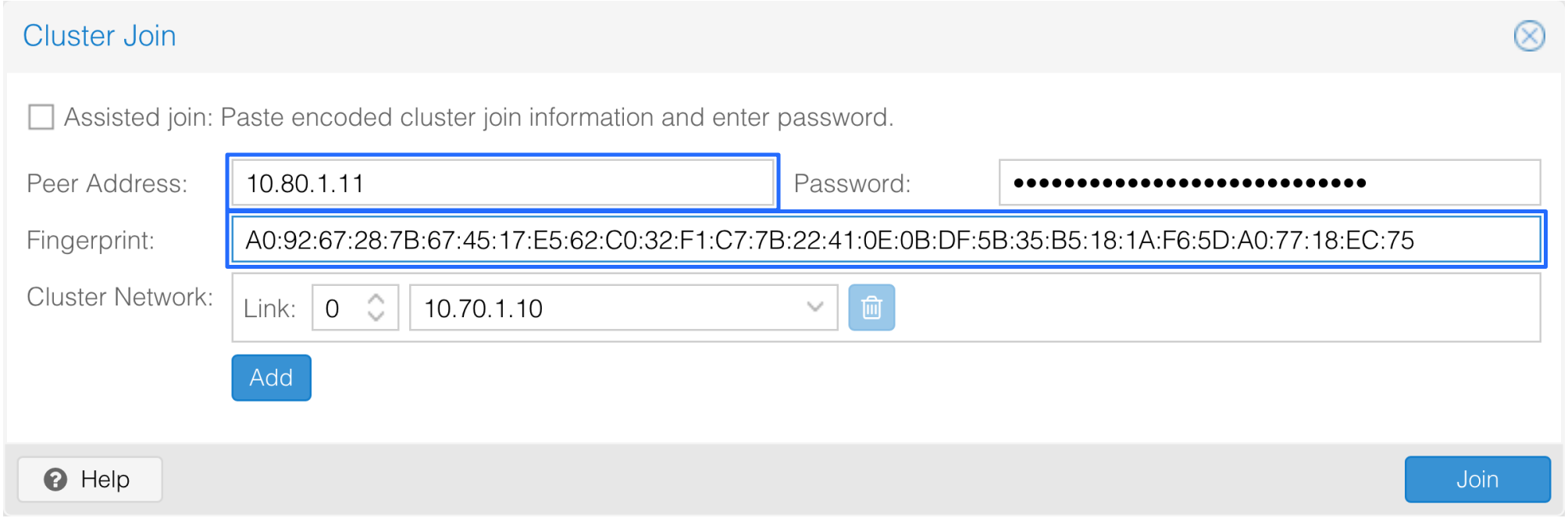
Create once

Create the cluster on one host. Other empty hosts join it.

initMAX

Create a Proxmox cluster

Join a prepared node to the cluster



Existing peer

Use an established cluster member.

Verify identity

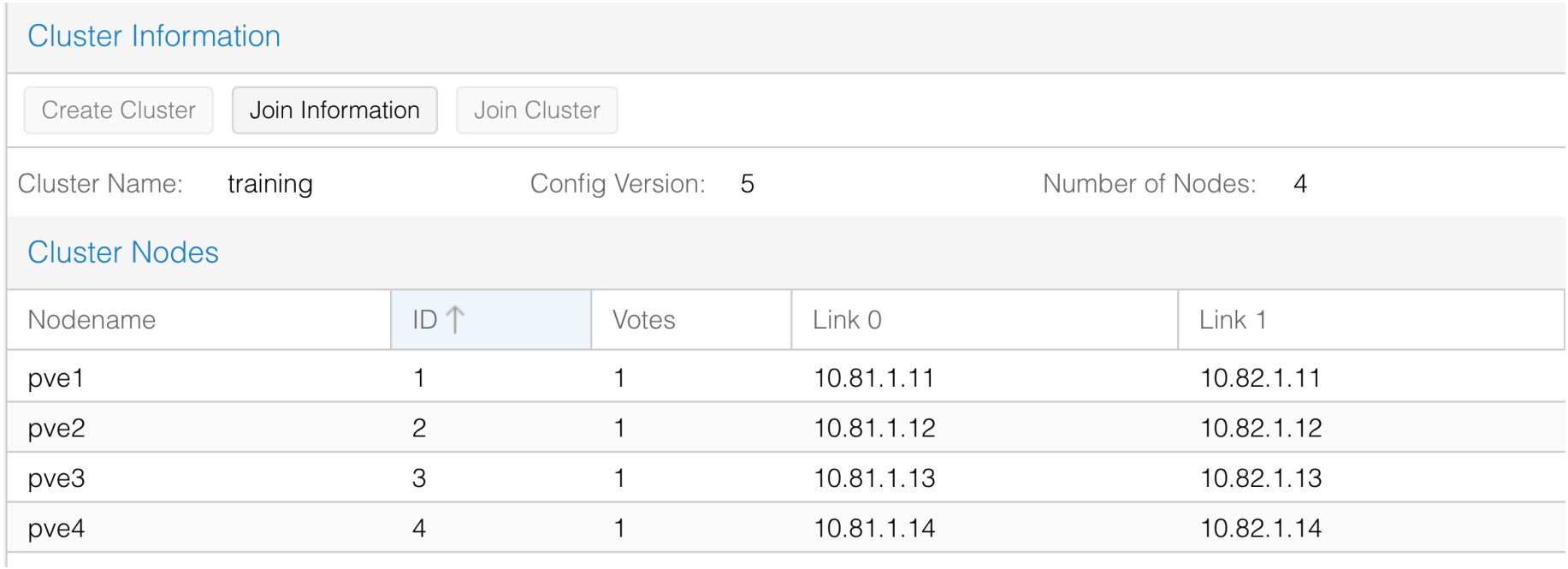
Verify the certificate fingerprint before joining.

Prepared host

Use an empty host and its planned local links.

Create a Proxmox cluster

Verify cluster membership and links



Membership

Four expected nodes, one vote each.

Corosync links

Both subnets match on every node.

Quorum

Confirm with pvecm status.

initMAX

Cluster checks in the shell

`pvecm status`	Expected votes, total votes, quorum and Quorate
`pvecm nodes`	Node IDs, names and voting membership
`chronyc tracking`	Time synchronization on a host using chrony

Run the checks from more than one node when investigating a partition.

initMAX

Demo: build and verify the cluster

DEMO

Switch to the Proxmox VE lab

1 Preflight

Check identity, time and links.

2 Create

Create training on pve1.

3 Join

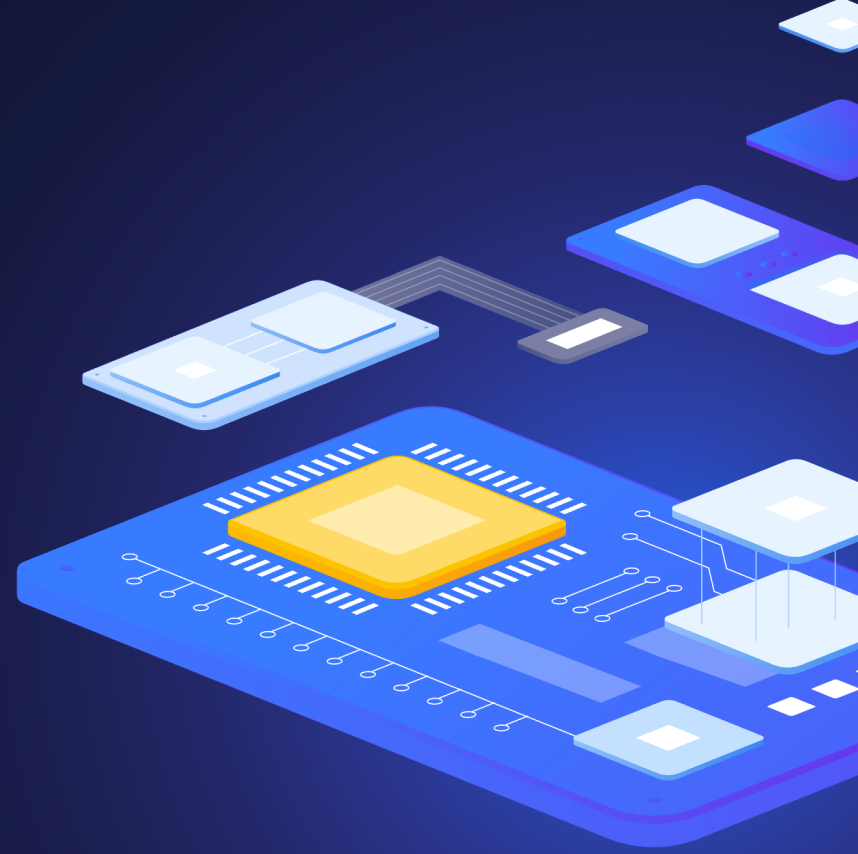
Join the remaining clean nodes.

4 Verify

Check membership, links and quorum.

3

Proxmox HA cluster



Proxmox HA cluster

HA restart and application recovery



Recovery time includes the application

The guest may be running before the database, web service and client connections recover.

HA restarts a failed workload. It does not preserve the failed host’s RAM.

HA prerequisites beyond quorum

A safe place to restart

A healthy quorate cluster, working fencing and eligible targets.

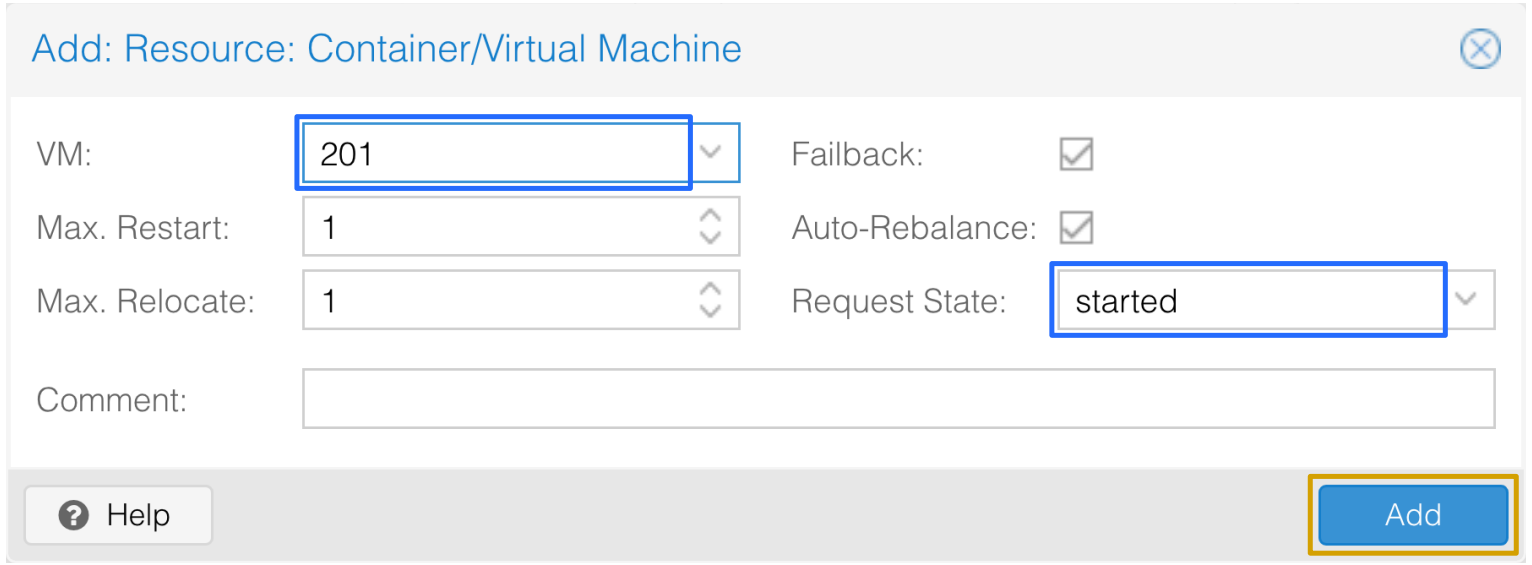
Usable disks and resources

Shared storage or a valid replica, plus CPU and memory headroom.

A recoverable application

Boot order, service checks, backups and tested recovery behavior.

Add a resource to HA management



Resource ID

The open Add form selects VM 201. Verify storage and placement first.

Requested state

started asks HA to keep the resource running.

Recovery limits

Review restart, relocation and failback behavior.

initMAX

HA status after adding the VM

Status

Type	Status
quorum	OK
fencing	armed (CRM watchdog active)

Resources

Add Edit Remove

ID	State	Node	Name	Max. Restart	Max. Reloc…	Failback	Auto-Rebal.
vm:200	started	pve4	cluster-ser…	1	1	true	true

Resource state

VM 200 is started on pve4 in this view.

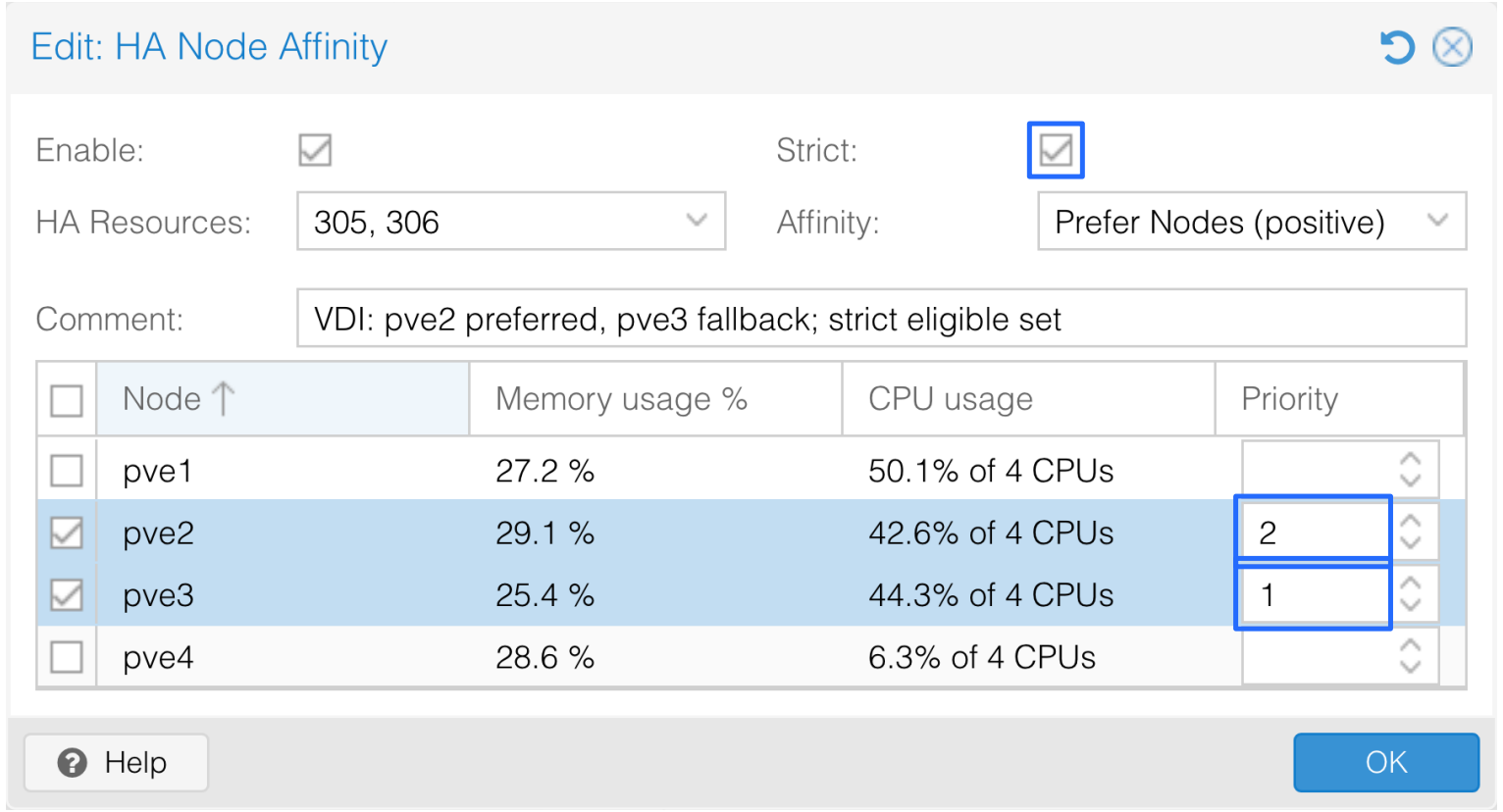
Cluster protection

Quorum is OK and fencing is armed.

Application

A running VM still needs a service check.

Node affinity: eligible hosts and priorities



Resources

The example applies to VMs 305 and 306.

Priority

pve2 is preferred over pve3 in this example.

Strict mode

Only listed eligible nodes may host the resources.

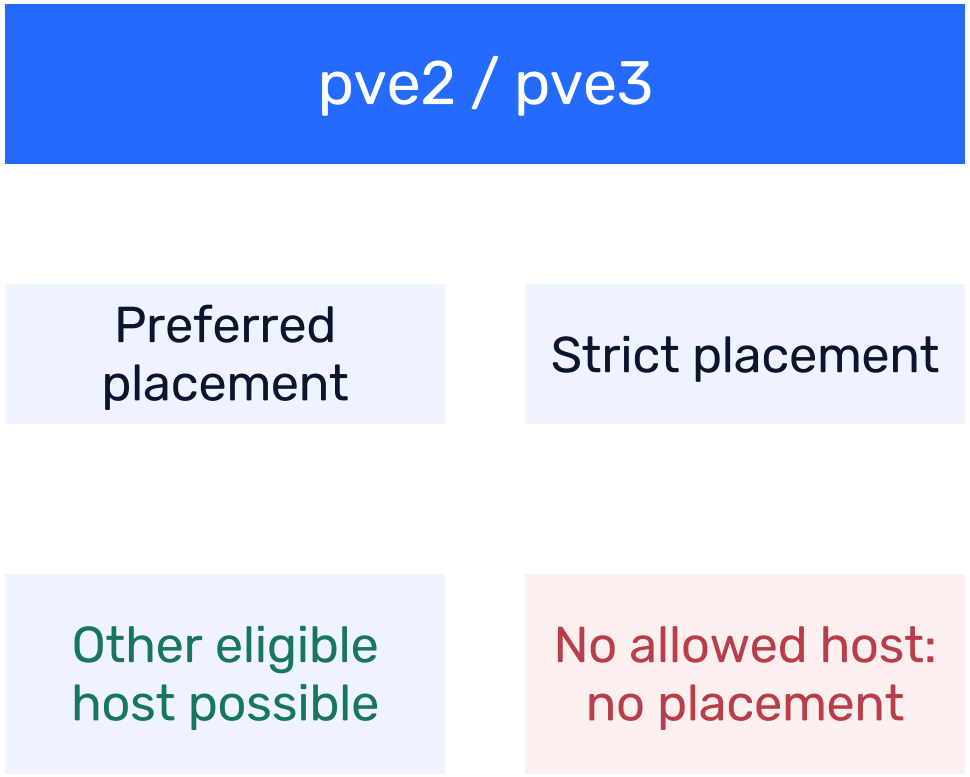
Strict placement and preferred placement

Preference

Use the preferred nodes when possible. Other eligible hosts can remain a fallback.

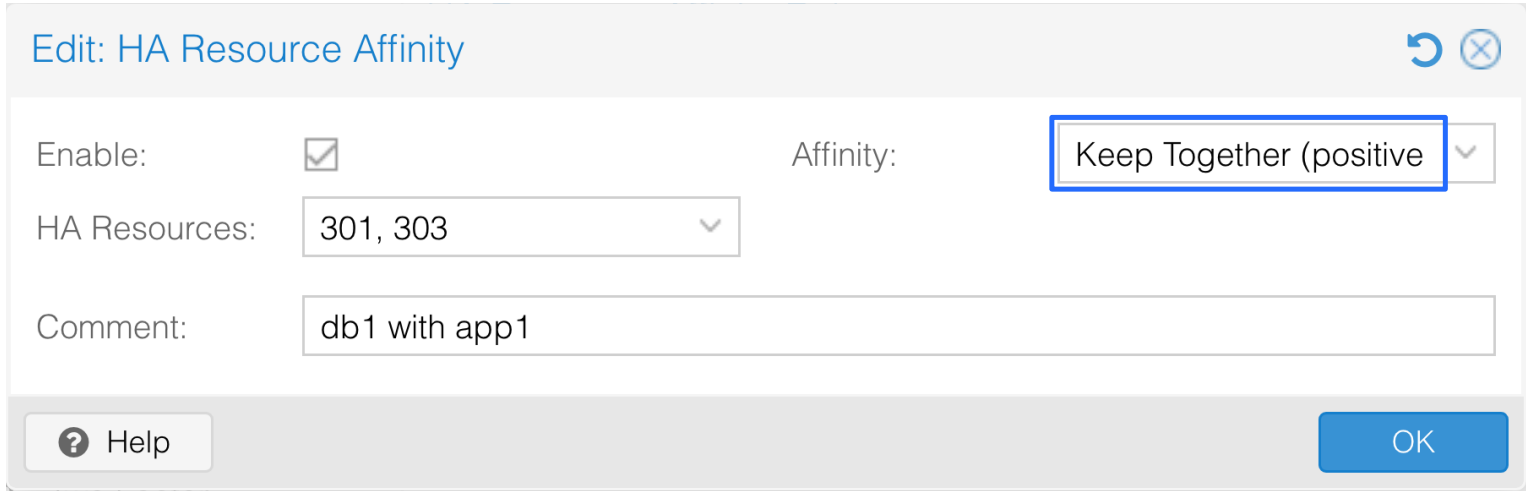
Strict constraint

If all allowed hosts are unavailable, HA cannot place the resource on an unrelated node.



initMAX

Resource affinity: colocate or separate



Resource set

This example uses VMs 301 and 303.

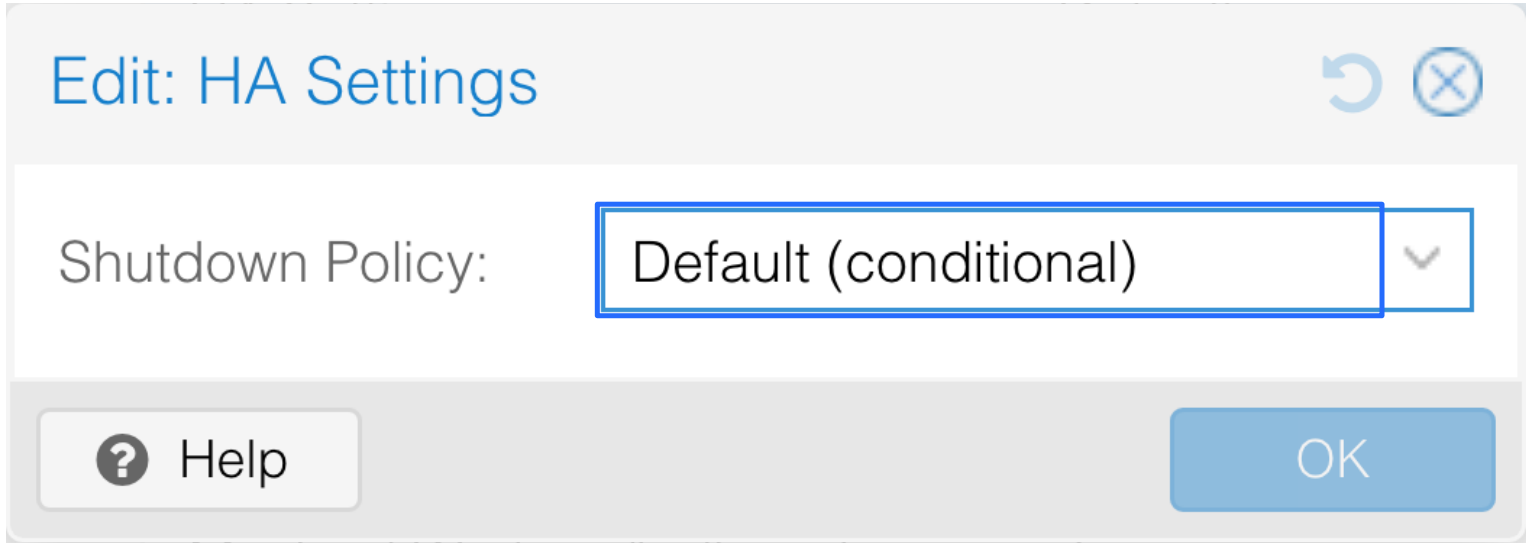
Keep together

Positive affinity colocates the selected resources.

Keep separate

Use negative affinity for host-level separation.

Shutdown policy and HA behavior



Cluster setting

Shutdown Policy affects orderly shutdown behavior.

Read the policy

conditional, migrate, failover and freeze differ.

Planned updates

Explicit maintenance is easier to verify than assumptions.

initMAX

HA shutdown policies

Policy	Behavior during orderly node shutdown
migrate	Move HA services away before shutdown. Requires an eligible target.
failover	Stop services. Recover elsewhere if the node stays offline.
freeze	Stop and freeze services until the original node returns.
conditional	Reboot: restart on the same node. Poweroff: recover elsewhere.

Shutdown freeze stops services. Disarm freeze keeps their current running state.

initMAX

Demo: add HA and review placement

DEMO

Switch to the Proxmox VE lab

1 Select

Choose a recoverable test VM.

2 Add

Request started in HA.

3 Constrain

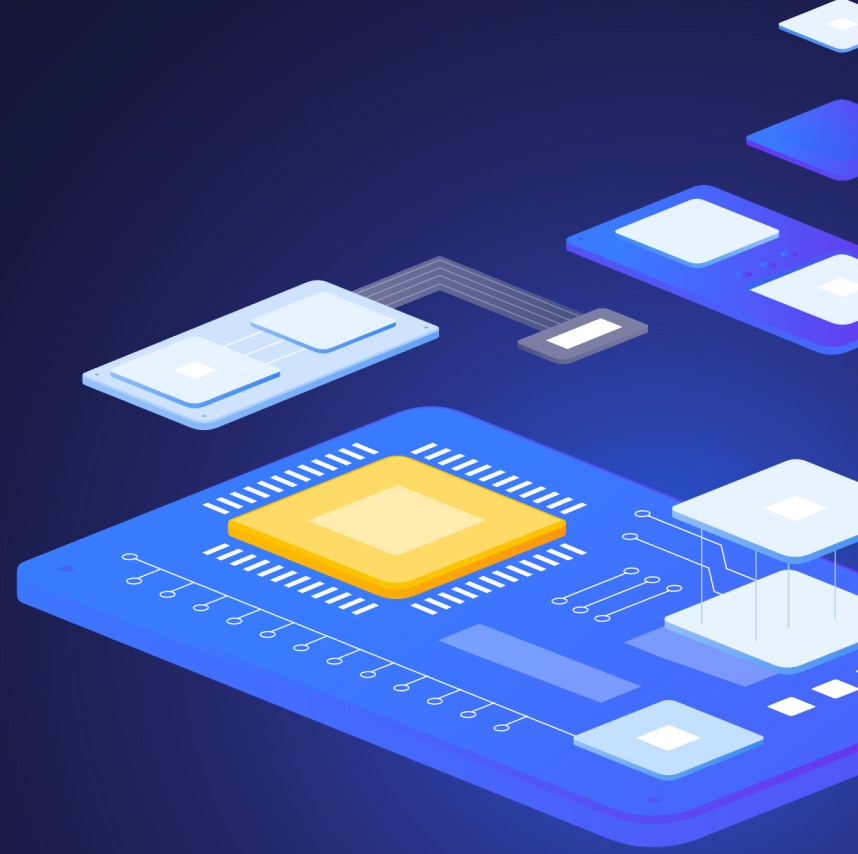
Review storage and affinity rules.

4 Verify

Check actual state and application.

4

Live migration



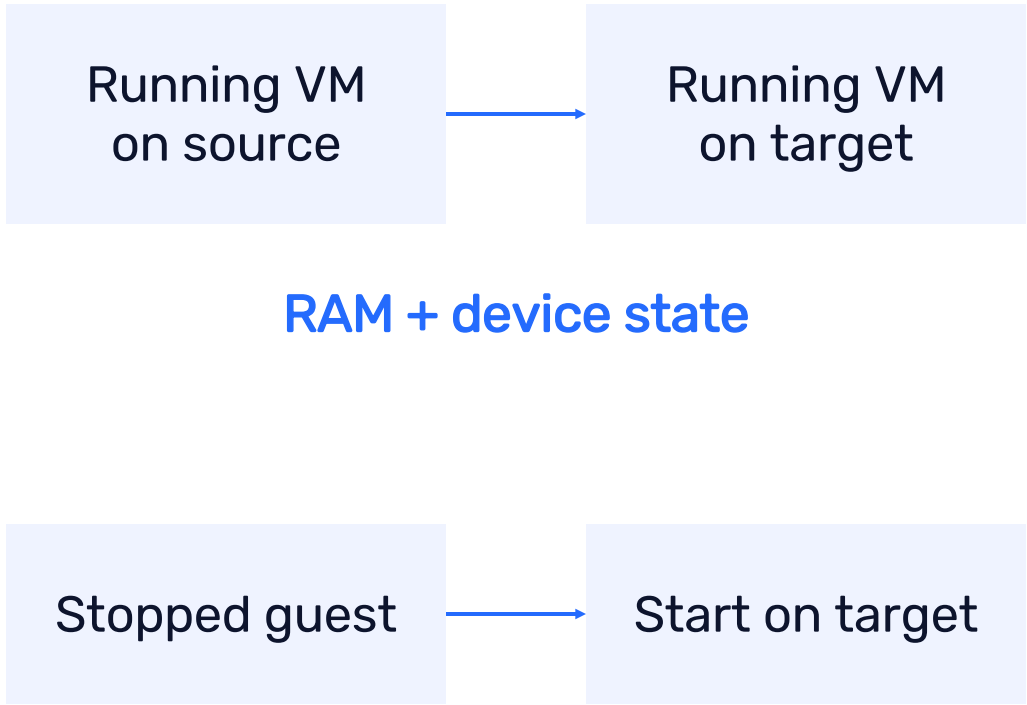
Live migration and restart migration

Live VM migration

Source and target are running. Memory is copied, then execution switches to the target.

Restart or offline migration

The guest stops or is already stopped. Its next start occurs on the target.



initMAX

Live migration prerequisites

Area	Check before migration
CPU and version	Compatible guest CPU model and supported package versions
Guest devices	No blocking local device or passthrough dependency
Network	Matching bridges / VLANs and sufficient transfer capacity
Target resources	Enough RAM, CPU and accessible storage
Application	A continuous service probe and a recovery plan

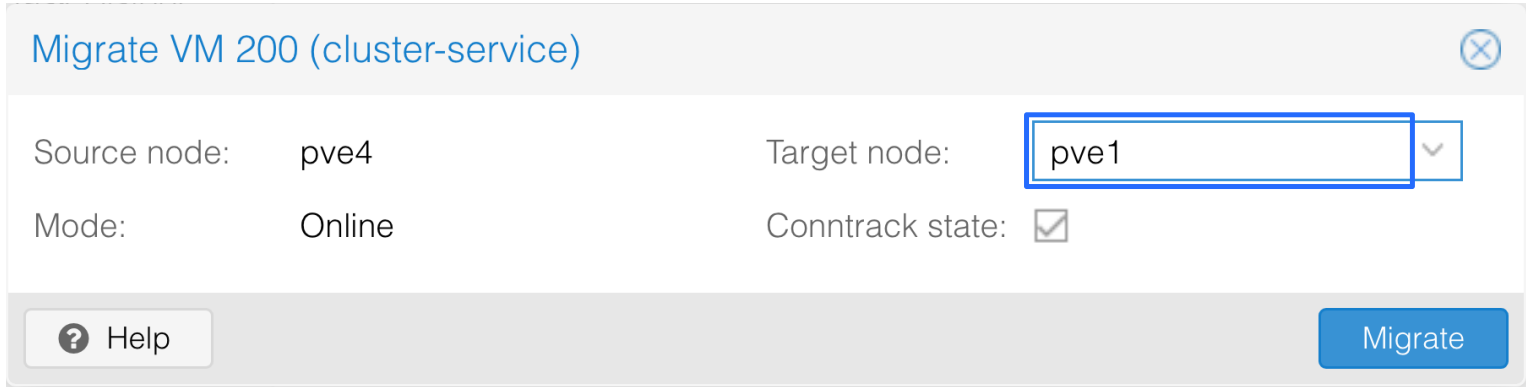
Live migration

Shared disks and local disk migration

Disk location	What must move	Operational effect
Shared storage	RAM and execution state	Less bulk disk transfer
Local disk	Disk data plus RAM/state	More bandwidth and target space
Valid ZFS replica	Final storage deltas plus state	Replica can reduce transfer work

“Shared” must describe real shared access. Checking a box does not copy data.

Online migration: choose the target



VM and source

VM 200 is running on pve4 in this view.

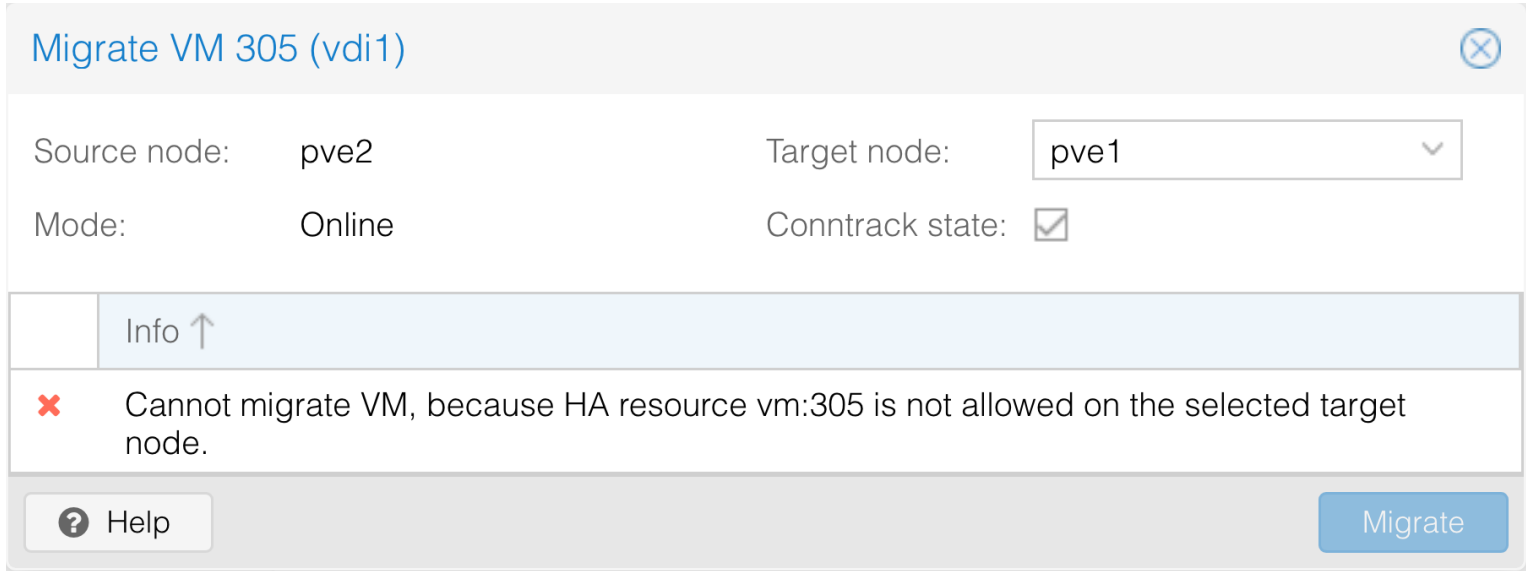
Target and mode

The form selects pve1 and Online.

Final review

Confirm capacity, disks and the application probe.

Migration blocked by HA placement



Separate example

VM 305 uses the strict rule shown earlier.

Target rejected

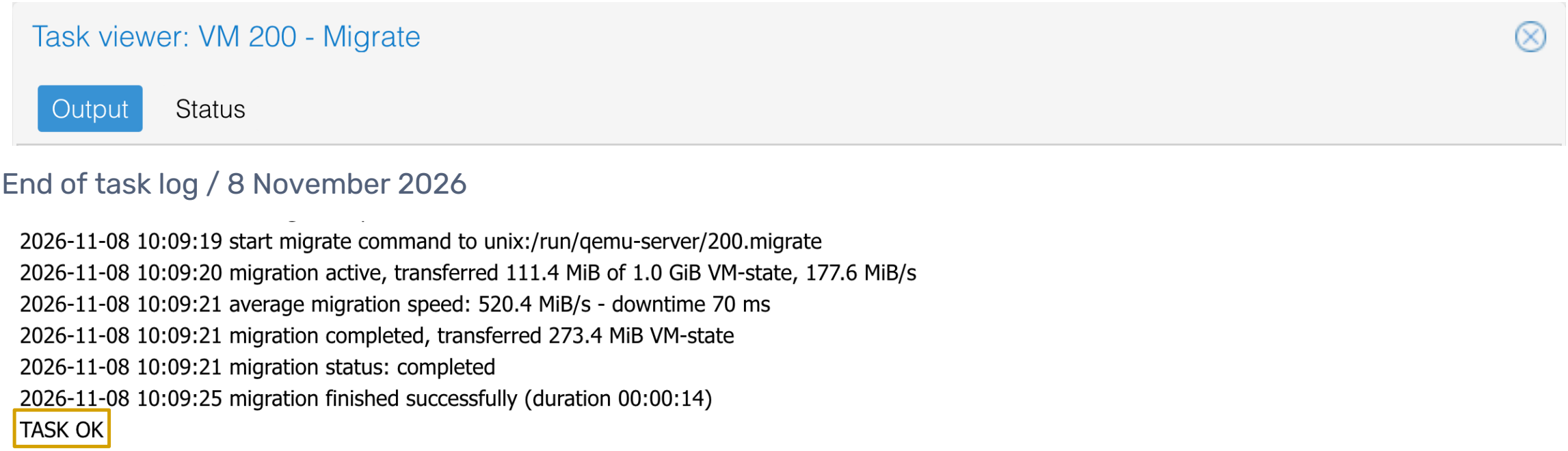
pve1 is outside the allowed pve2 / pve3 set.

Resolve the dependency

Choose an eligible host and verify its storage.

Live migration

Migration task and application check



Task result

Read the end of the log and confirm successful completion.

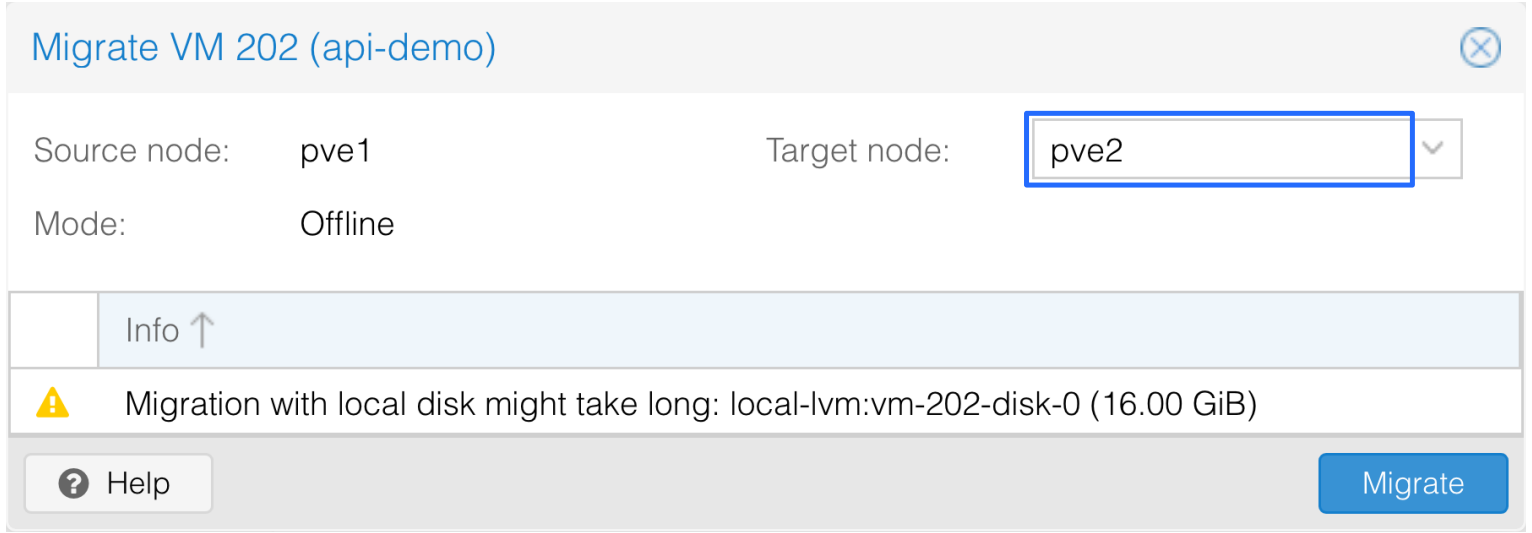
Guest location

The source and target view must agree after the move.

Service probe

Check actual response, not only TASK OK.

Local disks: read the migration warning



Separate example

VM 202 is api-demo, not VM 200.

Offline mode

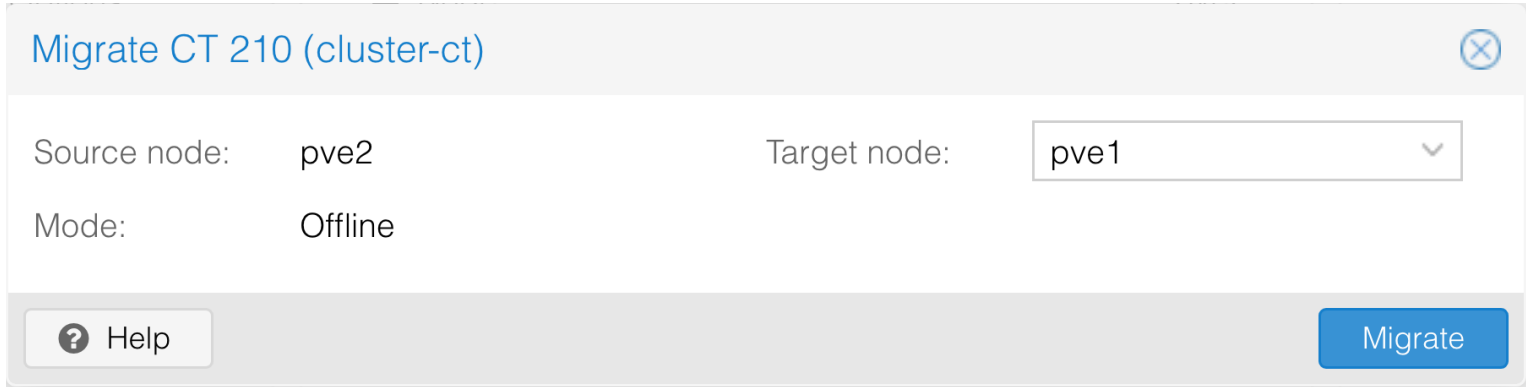
The screenshot shows a stopped VM migration.

Disk transfer

The local 16 GiB volume must reach the target.

Live migration

Container migration: offline and restart



Stopped container

The current CT 210 view uses Offline mode.

Running container

Restart migration stops the CT and starts it on the target.

Downtime

Budget application startup and validate its data.

initMAX

Demo: migrate and observe the service

DEMO

Switch to the Proxmox VE lab

1 Baseline

Record guest location and service response.

2 Migrate

Select an eligible target and mode.

3 Observe

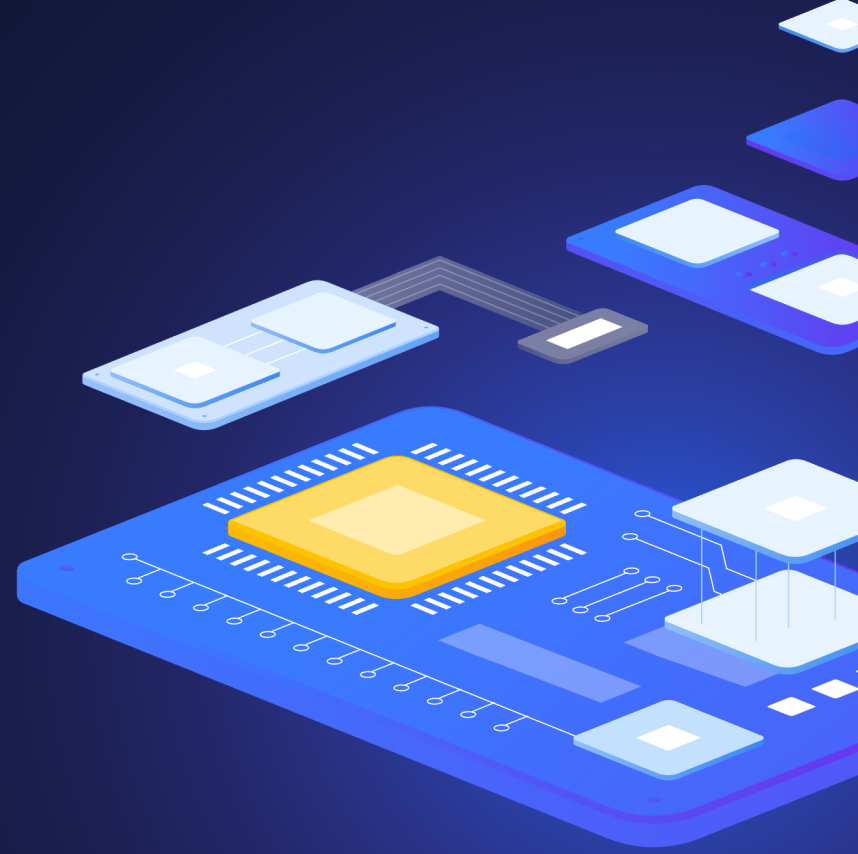
Follow the task and service probe.

4 Verify

Check location, HA state and application.

5

Failure and maintenance



initMAX

Failures and their impact

Failure	Main dependency	What to verify
One host lost	Quorum, disks and target capacity	HA recovery and service health
Corosync partition	Quorate membership and fencing	No second active instance
Storage unavailable	Actual guest disk access	Storage recovery before guest restart
Several hosts lost	Remaining votes and resources	Whether safe recovery is still possible

initMAX

Failure and maintenance

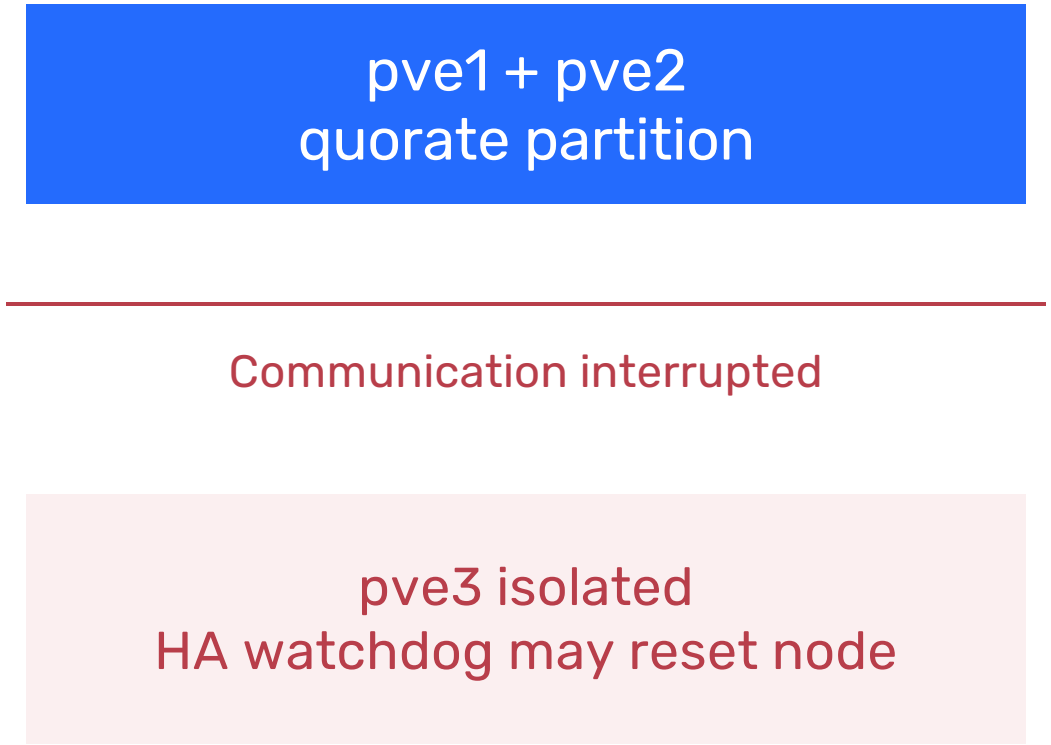
Network isolation and self-fencing

Loss of contact is ambiguous

The node might be alive but isolated. Starting a second copy immediately would be unsafe.

Fencing establishes a safe boundary

HA recovery follows fencing and quorum decisions, then starts an eligible copy.



initMAX

Failure and maintenance

Node maintenance for a planned update

1 Request maintenance

```
ha-manager crm-command node-maintenance enable pve1
```

2 Confirm evacuation

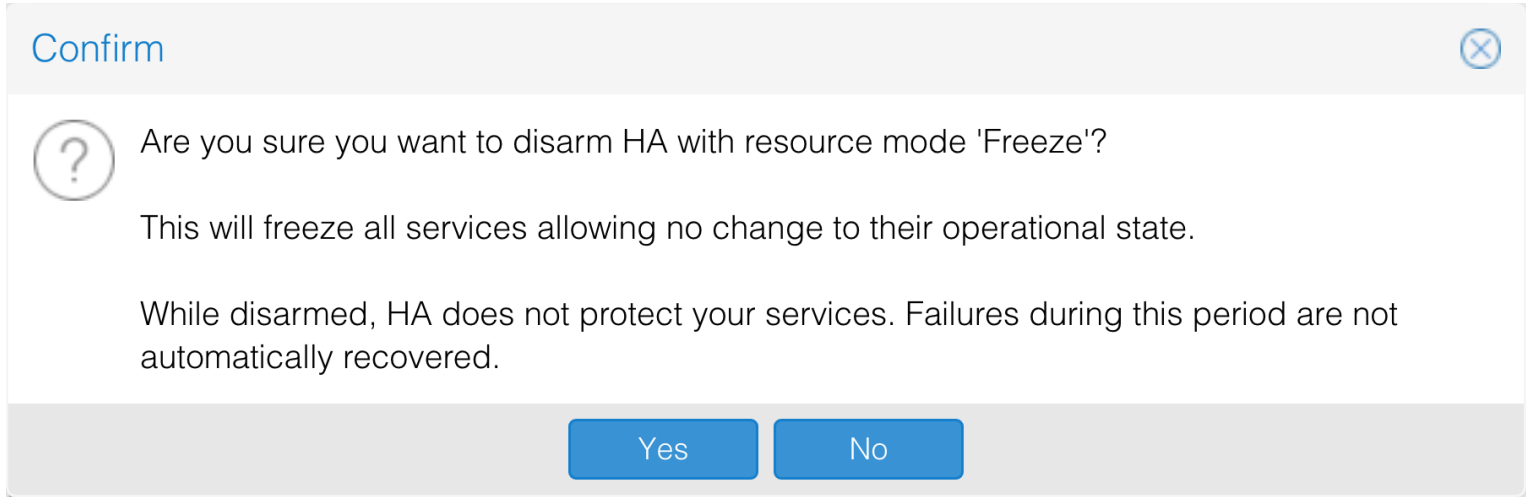
```
ha-manager status
```

3 Return the node

```
ha-manager crm-command node-maintenance disable pve1
```

Wait for HA resources to leave. Check non-HA guests and storage separately.

Cluster-wide HA disarm for maintenance



Different scope

Disarm affects HA watchdogs across the cluster.

Freeze mode

Resources keep their current state while HA actions pause.

Reduced protection

No automatic HA recovery while fully disarmed.

Failure and maintenance

Disarm, confirm, maintain and re-arm

Disarm in freeze mode

```
ha-manager crm-command disarm-ha freeze
```

Wait for disarmed

```
ha-manager status
```

After maintenance, re-arm

```
ha-manager crm-command arm-ha
```

Keep the disarmed window short. Recheck status and the application after re-arming.

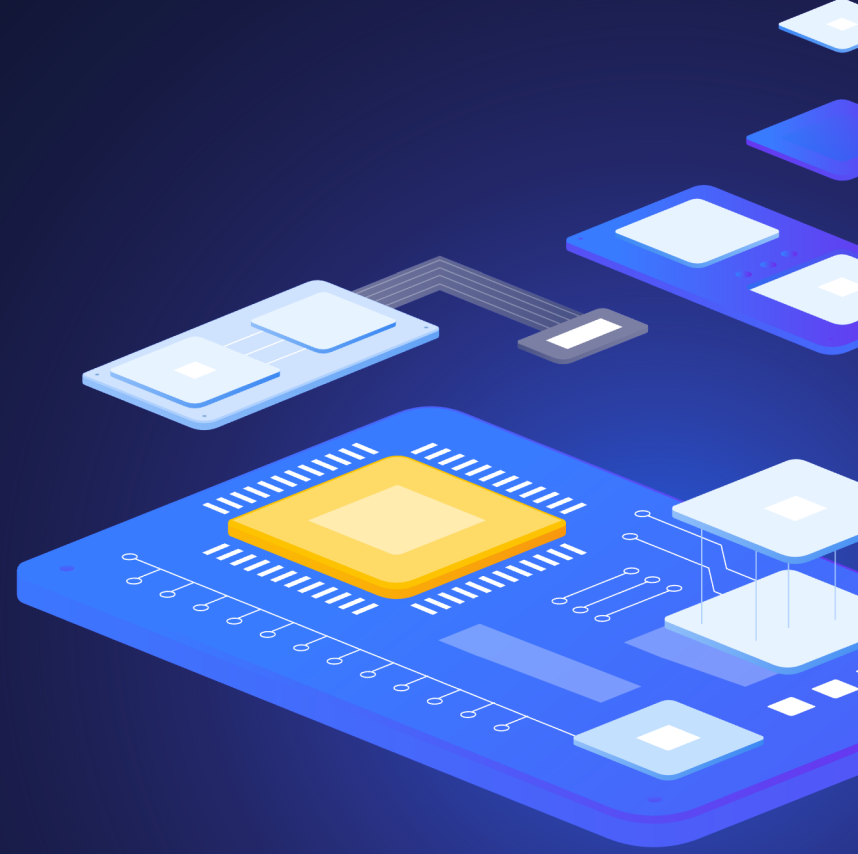
initMAX

Maintenance checklist

Before	After
Confirm backups and recovery access	Confirm cluster membership and quorum
Check storage health and recovery headroom	Inspect storage health and guest I/O
Evacuate or protect the affected workloads	Verify HA state, rules and resource location
Define rollback and application probes	Run the application probe and review alerts

6

DEMO



initMAX

Integrated lab demo

Demo: controlled failure and recovery

DEMO

Switch to the Proxmox VE lab

1 Start healthy

Record quorum, storage and HA state.

2 Lose one host

Use only an isolated, recoverable lab.

3 Observe recovery

Follow fencing, task logs and placement.

4 Validate service

Check application recovery, then restore the host.

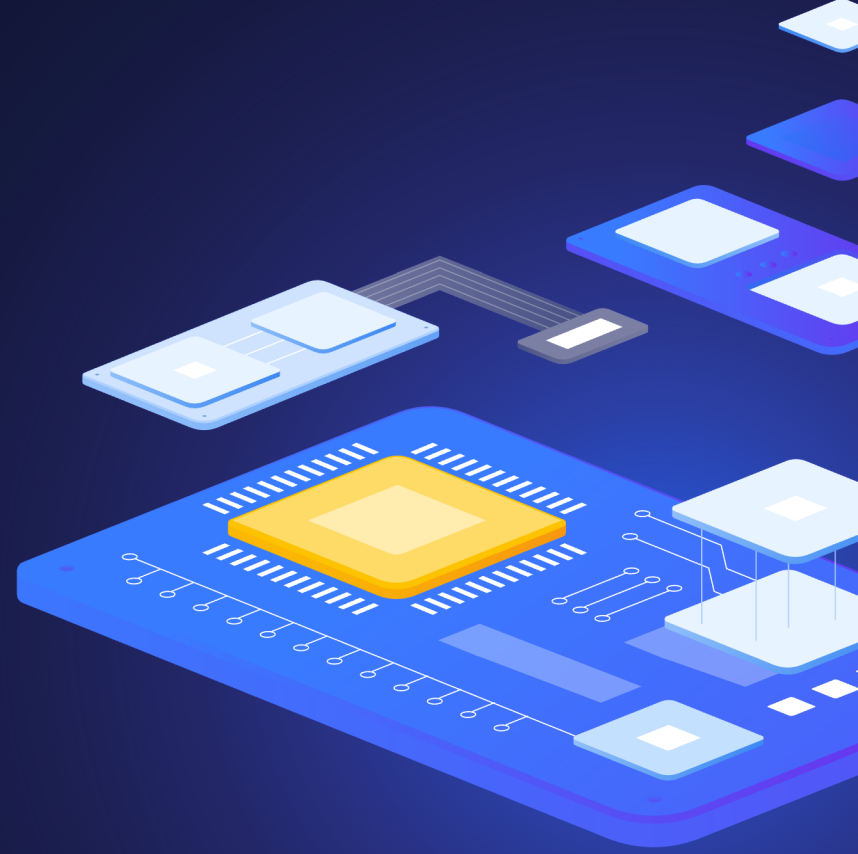
Integrated lab demo

Recovery evidence in the right order

`pvecm status`	Quorum and expected membership
`ceph -s`	Ceph health, if this design uses Ceph
`ha-manager status`	Actual HA service state and location
Application probe	Response, data consistency and recovery time

7

Problems



Problems

Design mistakes that defeat HA

One link carries every heavy transfer

Migration or storage recovery delays cluster communication.

Redundant paths share one failure point

One switch, power feed or storage controller can defeat both paths.

No room for recovery

The remaining hosts cannot fit the guests after one host is lost.

initMAX

Problems

Recovery assumptions worth testing

A replica is treated as a backup

Deletion, corruption or an old replica can invalidate the recovery assumption.

HA reports running, so the incident is closed

The application, network path or database can still be unavailable.

Quorum is bypassed to make recovery look easy

Unsafe forced actions can create two active instances or damage shared data.

Problems

A design case to discuss

Four hosts, local ZFS, one replica per critical VM

One host fails

Which remaining node has the current usable replica?

The preferred target is unavailable

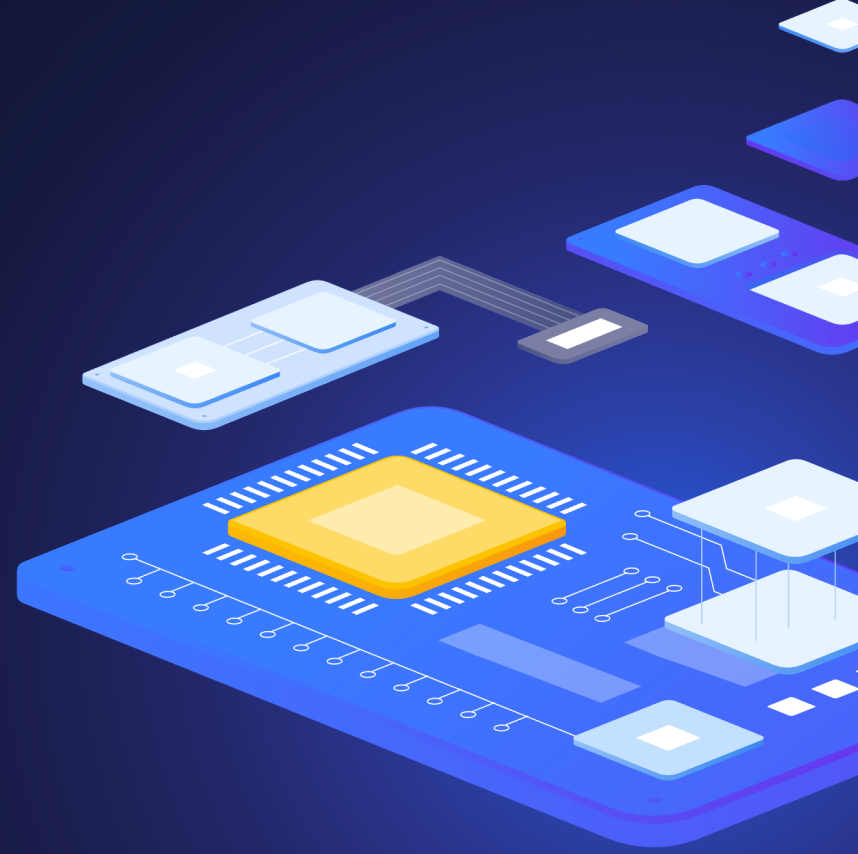
Do the placement rules permit a safe alternative?

The VM restarts

How will you verify data freshness and application recovery?

8

WIKI



Wiki

initMAX

Help us shape the Proxmox wiki

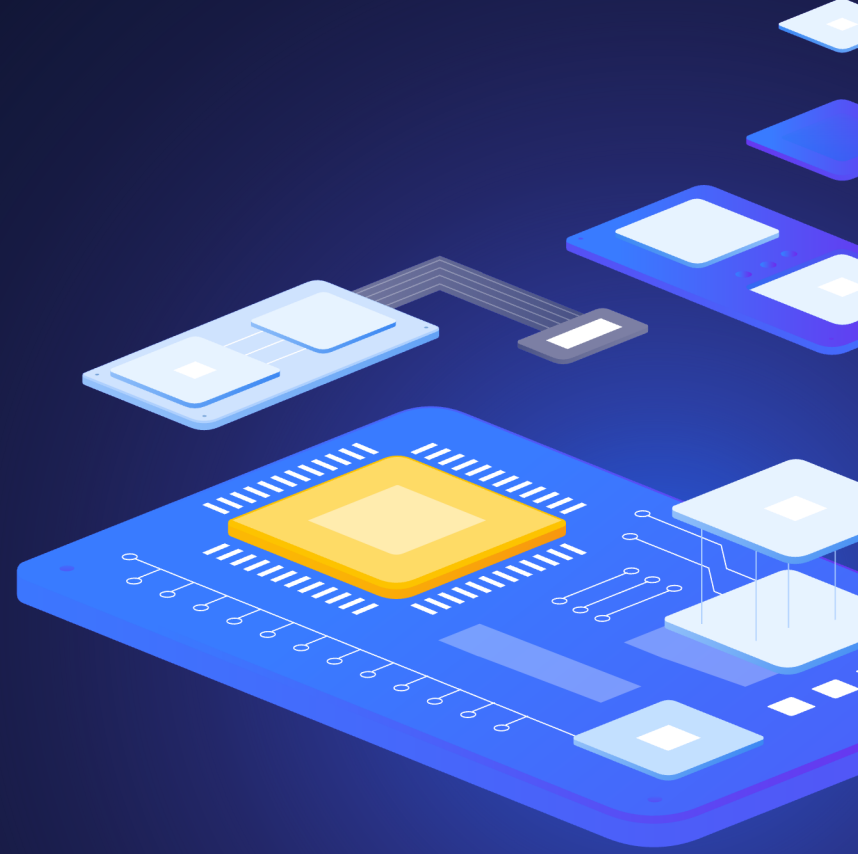
Tell us what you would like to find on the initMAX wiki about Proxmox, or what is bothering you right now.

- **Missing topics:** installation, storage, networking, HA, backup, migration from VMware
- **Production problems** you hit and want a tested fix for
- **Preferred format:** step-by-step guides, checklists, ready-made scripts

Send your suggestions in the Q&A now or to the initMAX team after the webinar.

9

Questions & discussion



initMAX services

initMAX

How initMAX can help you with Proxmox



2-hour consultation for free

Review of your design, cluster or migration plan with our engineers.

No commitment, no paperwork.



Consulting

Architecture and HA design, storage and networking, migration from VMware or Hyper-V, troubleshooting in production.



Training

Proxmox VE courses for administrators, hands-on labs.

Accredited training led by certified Proxmox trainers.

Book your free consultation at initmax.com or talk to us after the webinar.

Proxmox Cluster: Clustering, HA and Storage

Contact us

Implementation, support and training

Tomáš Heřmánek

tomas.hermanek@initmax.com

+420 732 447 184

Discuss your project with us →

initMAX

USA / International

+1 (727) 350-9332 — initmax.com

Czech Republic

+420 800 244 442 — initmax.cz

Slovakia

+421 277 270 154 — initmax.sk

LinkedIn: initMAX

X: @initmax