



Open-E JovianDSS with Proxmox VE: NFS Plugin

How-To Guide

Version: 1.0
Last Update: September 8th 2026
Supported Open-E JovianDSS up33 and newer
Storage plugin: v1.1.0

Table of Contents

1. Executive Summary and Introduction
2. Questions and Answers
3. Solution Architecture and Network Topology
4. Proxmox VE Cluster Setup (network, HA rules, fencing)
5. Creating the Open-E JovianDSS VMs (step by step)
6. Open-E JovianDSS Configuration (bonds, ping nodes, REST API, VIPs, NFS)
7. Plugin Installation and Storage Registration
8. VM Deployment (Microsoft Windows / Linux)
9. Snapshots and Data Protection (OODP)
10. Restore Scenarios
11. Pre-Production Validation (snapshots, rollback, failover)
12. Final Checklist and Known Issues

Plugin sources and latest documentation: <https://github.com/open-e/JovianDSS-Proxmox>

1.1 Executive Summary

Significant cost reduction of the virtual computing environment

This guide presents a cost-effective virtualization solution using Proxmox VE and Open-E JovianDSS in a hyperconverged High Availability setup. Unlike standard deployments requiring at least three servers, this architecture runs the entire hypervisor and storage workload on two servers. A third, low-cost node supplies the Proxmox VE HA quorum vote and can optionally serve as a backup and disaster recovery repository.

Plugin-based NFS storage with snapshots and rollback

The two main servers provide highly available, plugin-based NFS storage for Proxmox VE. VM disks are .raw files on a ZFS-backed dataset, and the plugin adds snapshot and rollback support for them - functionality not available with native NFS storage in Proxmox VE.

Enterprise-grade data protection with instant recovery

With Open-E JovianDSS On- and Off-site Data Protection (OODP), the system provides continuous backups with a near-zero RPO. The third node acts as a dedicated backup repository, so VMs can be restarted there if both main servers fail. If the third node is to be used for disaster recovery, it must be powerful enough to run all the virtual machines it would have to restore.

Key benefits

Hyperconverged architecture - storage and compute share the same two nodes. Continuous OODP protection with near-zero RPO. With the pve-config-backup service installed (9.8), VM/CT configuration files and disks are protected together, so recovery does not depend on rebuilding configuration by hand. The design scales up for large deployments while staying affordable for SMBs.

1.2 Introduction

Why this best practices guide?

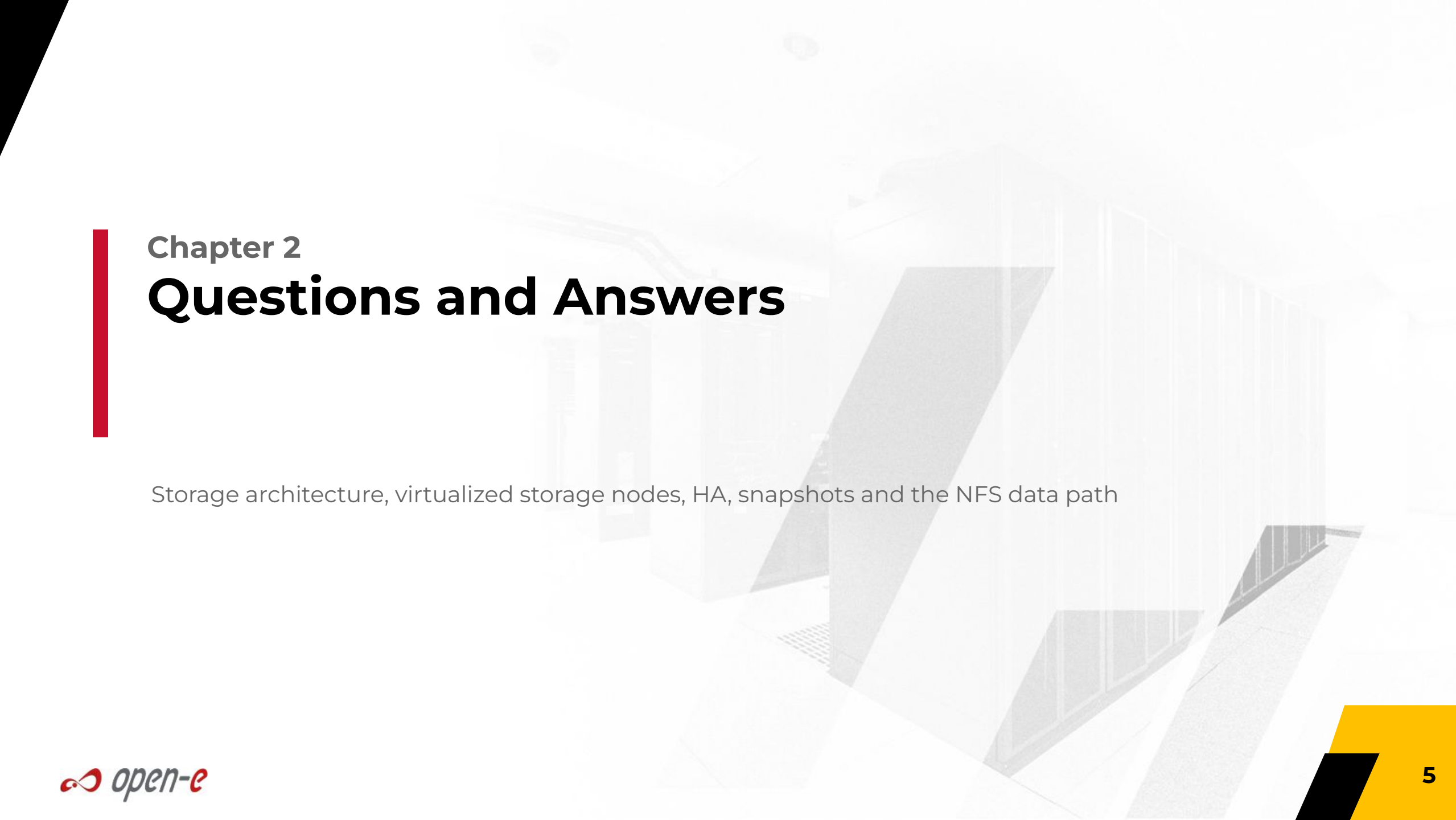
Open-E provides this guide to help organizations reduce the cost of virtualization infrastructure while keeping the high availability and continuous data protection expected from a turnkey solution. The recommended setup uses only two high-end servers for hypervisor, storage and application workloads. A third, more cost-effective server acts as the quorum node and can run a third instance of Open-E JovianDSS dedicated to backup and disaster recovery.

Why NFS for high-availability data storage?

NFS keeps VM disks as .raw files on a ZFS dataset, which makes day-to-day administration - and above all backup and disaster recovery - simpler than block storage: a dataset snapshot captures every VM on it at once, and replication moves whole datasets rather than individual volumes. The plugin adds Proxmox-native snapshot and rollback on top of those files. The nconnect mount option opens several TCP connections to the same storage address, which raises throughput on a single data path (chapter 7.4). A companion guide covering the iSCSI plugin is available separately from Open-E.

Why are both non-shared and shared storage cluster options considered?

Each has distinct advantages when cost matters. The non-shared option (Open-E JovianDSS Advanced Metro Cluster) is typically preferred below roughly 50 TB, suiting full-flash 2-way mirrors or HDD 4-way mirrors. Above that, mirroring becomes expensive, and shared-storage arrays with 2- or 3-parity redundancy give better storage efficiency; these use the Open-E JovianDSS Standard HA Cluster. The Standard HA Cluster also supports the mirror-path feature for NVMe cache and special allocation class devices, which accelerates HDD-based pools substantially.



Chapter 2

Questions and Answers

Storage architecture, virtualized storage nodes, HA, snapshots and the NFS data path

2.1 Data Storage Architecture

How is the network configured?

Several layouts are valid. The minimum is two NICs for management and storage access, plus one or two passthrough NICs for the storage mirror path - used for non-shared storage, or for NVMe cache in a shared-storage cluster. In this guide two NICs carry management, each with its own Linux bridge, and both bridges take part in the Proxmox VE HA cluster. One is sufficient; two are recommended, so that a single link failure cannot isolate a node.

Two further NICs are joined into a Linux bond in balance-alb mode. The bond carries no IP address of its own - a bridge sits on top of it, and that bridge holds the NFS data addresses. Both pools share this interface: the address of the second pool is an alias on the same bridge (chapter 4.3).

Which NICs are used for the mirror path?

The mirror path carries the entire write stream between the two nodes, which makes it the most demanding link in the design. The adapter has to be at least 10 GbE and has to be supported by Open-E JovianDSS; an RDMA-capable adapter is preferred, because RDMA lowers both CPU cost and latency on this path. NVIDIA ConnectX is one such family - the card used throughout this guide is a ConnectX-5 (3.5) - but any supported adapter meeting those requirements is valid. The path should be a direct point-to-point link wherever possible; it also operates through a switch, but a direct link is the more reliable option.

2.2 Virtualized Storage Nodes

How are ping nodes configured in an HA storage cluster?

On bare metal the natural ping targets are storage clients, addressed in the same subnet as the NIC holding the Virtual IP. In a hyperconverged build the storage VIP and the hypervisor are connected through a virtual switch inside the Proxmox VE host, so that link cannot go down and a ping across it carries no diagnostic value. The target must lie outside the storage cluster: a switch management address, the gateway, the BMC of each server, or the third Proxmox VE host, which is always external because the storage VMs are pinned to pve1 and pve2.

Where the nodes are located in two buildings, all ping nodes belong in one of them. Spreading them across both creates a split-brain risk if the link between buildings fails.

How are disks and controllers assigned to the Open-E JovianDSS VM?

In passthrough. The HBA, NVMe controller or hardware RAID controller is added to the VM as a PCI device (Hardware -> Add -> PCI Device) and the whole controller is then managed inside the VM. On all-NVMe systems, where the disks are attached directly to motherboard PCIe slots, each NVMe disk is passed through individually.

2.3 Proxmox VE High Availability

How are application VMs kept on two hosts only?

On Proxmox VE 9 each VM is added under Datacenter -> HA -> Resources and then constrained by a node-affinity rule under Datacenter -> HA -> Rules that lists only the hosts allowed to run it. A strict rule keeps production off the quorum node entirely. Chapter 4.5 covers the procedure.

Note: older documentation describes HA Groups (Datacenter -> HA -> Groups). Proxmox VE 9 replaced groups with rules - there is no Groups view any more, and the matching CLI options are gone.

How is fencing configured?

Through the watchdog. In `/etc/default/pve-ha-manager`, `WATCHDOG_MODULE=ipmi_watchdog` selects the hardware watchdog; left commented out, Proxmox VE keeps the `softdog` software fallback. Chapter 4.6 covers this.

How should a node shutdown behave?

Datacenter -> Options -> HA Settings. The default is Conditional, kept for backward compatibility. Migrate moves guests to another host when a node is shut down.

2.4 Snapshots and Data Protection

How does Take Snapshot become available in the Proxmox VE GUI?

The plugin provides it. Native NFS storage in Proxmox VE can snapshot qcow2 disks only; this plugin brings the same function to .raw disks, which is its primary purpose - .raw retains the performance advantage while gaining snapshots and rollback. Taking a snapshot creates a ZFS snapshot of the dataset - the NAS volume in the Open-E JovianDSS GUI - and no configuration beyond the documented setup is required.

What happens on rollback?

The volume is restored to the selected snapshot. Two properties matter in practice, both measured in the Open-E lab: rollback copies the whole disk file, so it takes time proportional to DISK SIZE rather than to the amount of changed data - 32 GiB took 36 seconds - and the VM remains stopped for the rollback duration. Snapshots taken AFTER the one selected for rollback are NOT destroyed: the snapshot tree branches, and both lines remain available (chapter 9.2). The VM definition is restored with it: Proxmox VE keeps a copy of the configuration inside every snapshot and puts it back on rollback, so hardware changes made after the snapshot - an added disk, more memory - are reverted as well. That is a Proxmox VE mechanism, independent of the storage and of pve-config-backup (9.8).

A snapshot is an object of the dataset, not of one VM. It captures every VM stored on that dataset at the same instant - convenient for consistency, and a factor when planning datasets.

2.5 The NFS Data Path

How is active-active load balancing arranged?

With two pools. Each pool carries at least one dataset exported over NFS, and each is imported on a different storage node - the first pool on node-a, the second on node-b. Both nodes then serve data at once, and either can take over the pool of the other. Proxmox VE registers one storage entry per pool.

How many paths does NFS use?

One per pool. Each pool has a single data Virtual IP, and that VIP moves to the surviving node during failover. Throughput comes from the nconnect mount option, which opens several TCP connections to the same address - it widens one path rather than adding a second, so it is a performance setting and not a redundancy mechanism (chapter 7.4).

Where do VM and container configuration files reside?

In /etc/pve, not on the dataset. Proxmox VE keeps them in its cluster filesystem, so a dataset snapshot captures the disks only. The optional pve-config-backup service copies them into the folders that contain their disks, which brings them inside the snapshot (chapter 9.8).



Chapter 3

Solution Architecture and Network Topology

Concept, three reference topologies and the IP addressing plan

3.1 Concept: A VM Disk Is a File on a ZFS Dataset

A VM disk is an ordinary .raw file on a ZFS dataset, reached over NFS

Proxmox VE

Guest VM - disk is a file, not a block device
scsi0: nfs-Pool-0:104/vm-104-disk-0.raw
Storage type: joviandss-nfs (shared)
Snapshot and rollback are driven by the plugin, which native NFS storage cannot do

Same file is visible from every cluster node,
which is what makes live migration and HA work

NFS data path

mount: 10.15.21.142:/Pools/Pool-0/nfs-Pool-0
options: vers=3, nconnect=4
nconnect opens 4 TCP connections to the same address - more throughput on one path, not a second path and not redundancy

One data VIP per pool; the VIP follows the pool to the surviving node during storage failover

Open-E JovianDSS

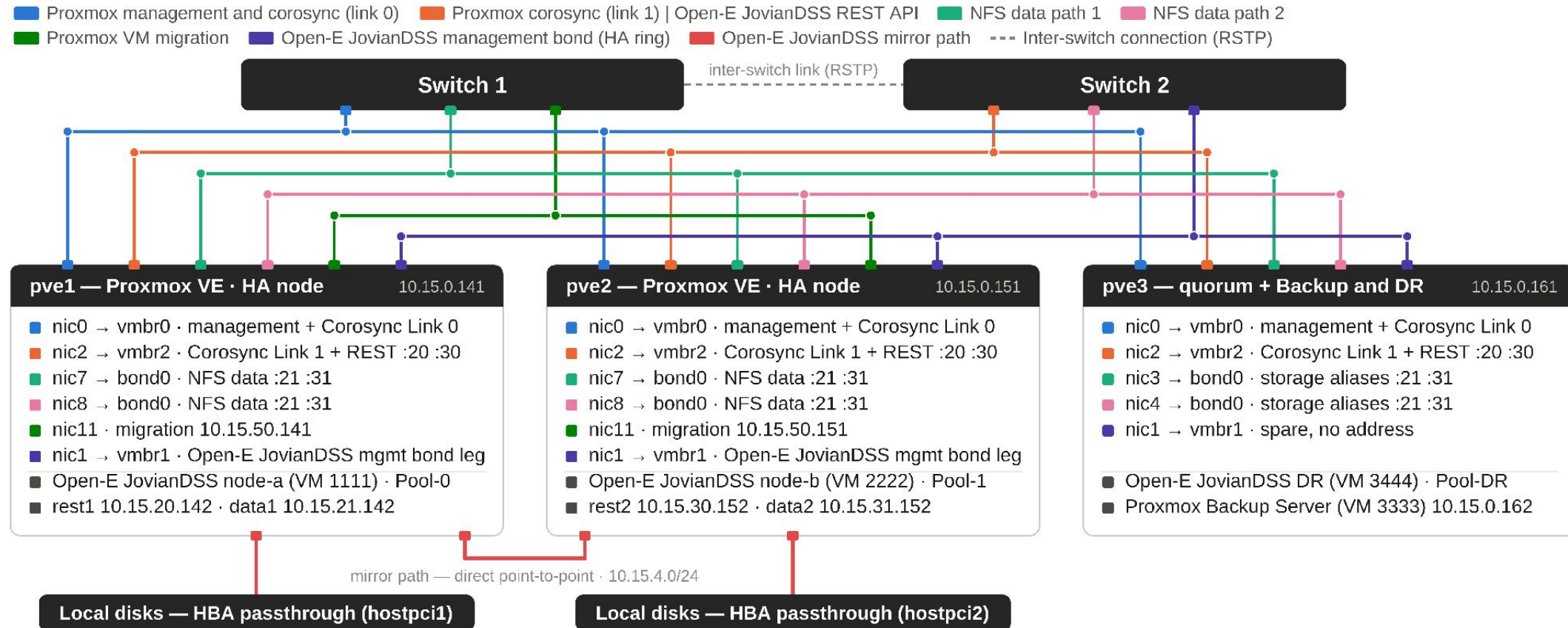
ZFS dataset: Pool-0/nfs-Pool-0
Holds the .raw files of every VM on it
A snapshot is a dataset-level object - it captures all VMs on the dataset at once
OODP replicates the dataset to the DR node

REST API on a separate control VIP carries the plugin commands; it never carries VM data

Reads and writes travel this path; the plugin talks to JovianDSS out of band over REST.

The REST control channel carries plugin commands only - never VM data.

3.2 Non-shared Storage with Backup and DR (the lab as built)

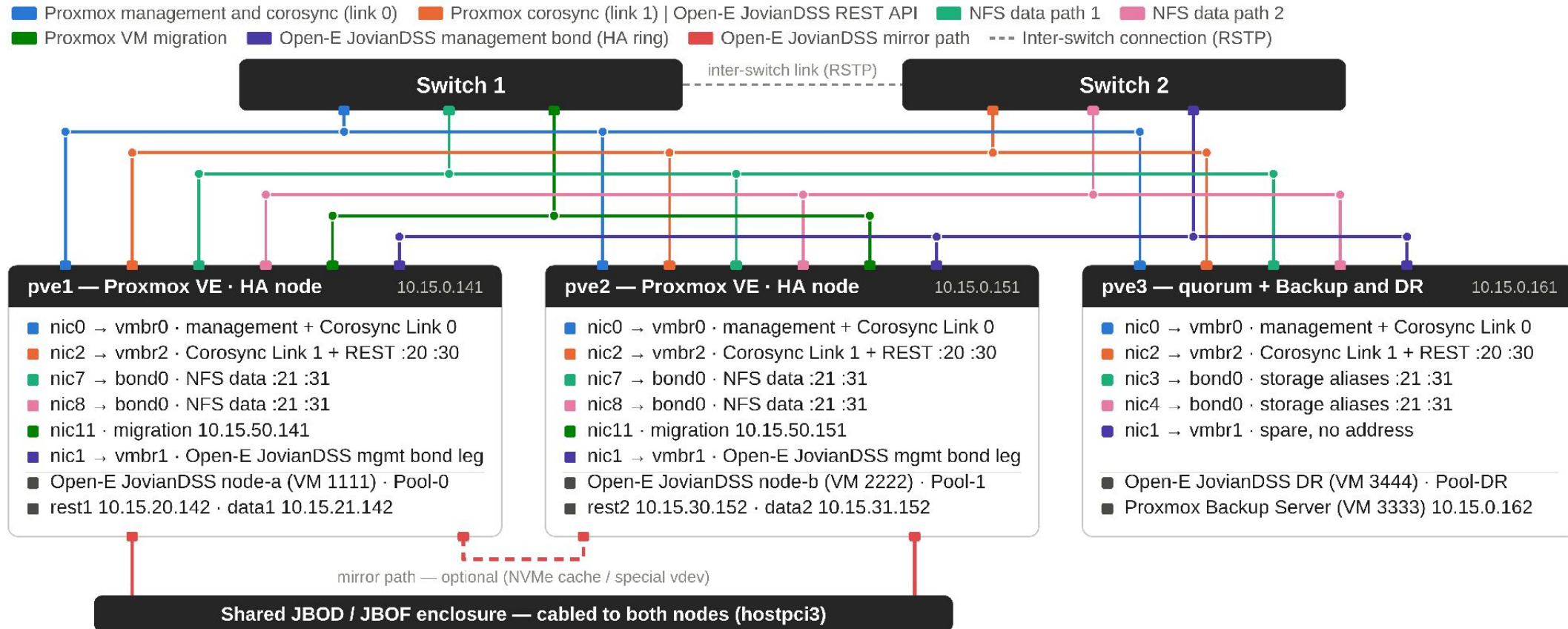


Each node owns its disks, presented through an HBA in PCI passthrough (hostpci1 / hostpci2); the two pools stay in sync over a direct point-to-point mirroring path on 10.15.4.0/24, at least 10 GbE. Pool data VIPs are 10.15.21.142 (Pool-0) and 10.15.31.152 (Pool-1); interfaces carrying production VIPs should be at least 10 GbE, as should the OODP interface on the Backup and DR node.

A dot marks a real connection. The third node votes for quorum and holds Backup and DR; it runs no guest workload.

Reference topology as built and validated in the Open-E lab. Not all services are switch-redundant in this layout; where full redundancy is required, additional interfaces and links should be provided and validated for the target environment.

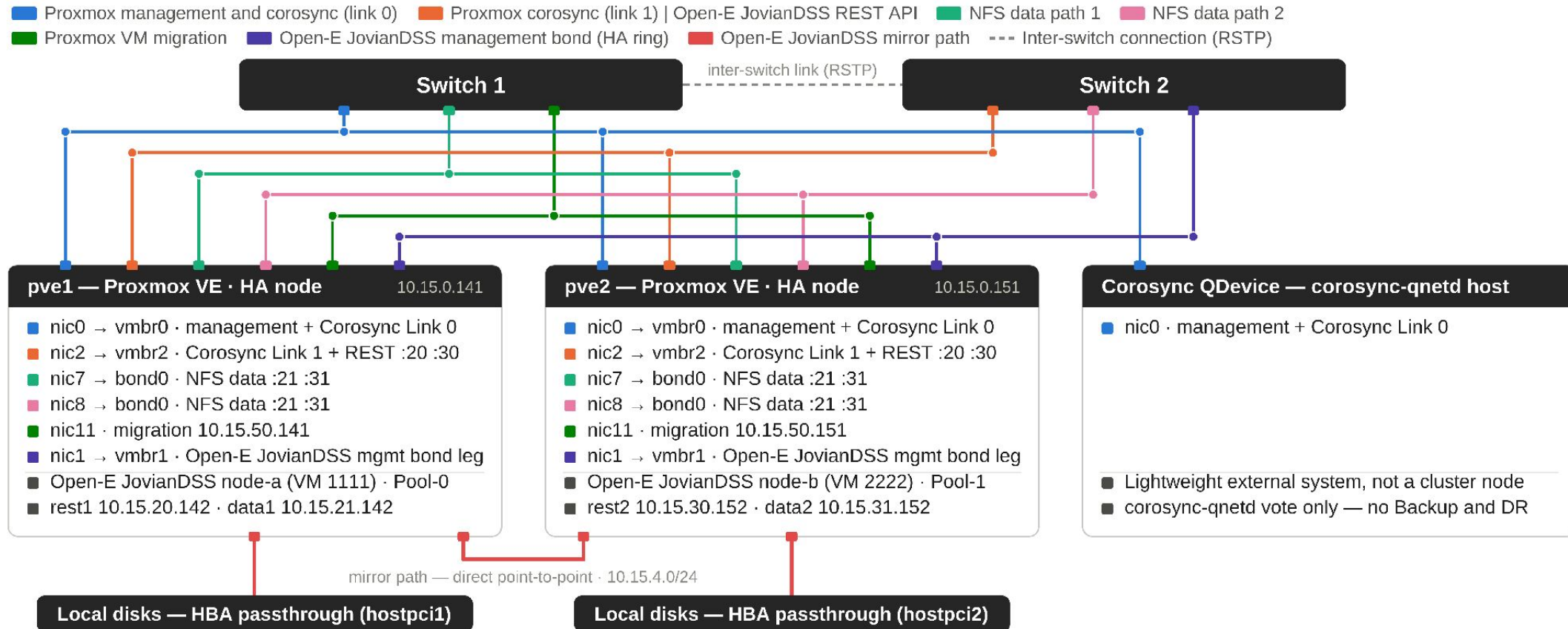
3.3 Shared Storage (JBOD / JBOF) with Backup and DR



Both nodes are cabled to one JBOD / JBOF enclosure (hostpci3) and reach the same disks, so 2- or 3-parity redundancy replaces mirroring - the efficient choice for large capacities. The Ethernet mirroring path becomes optional here and is kept for NVMe cache or special allocation class devices. Pool data VIPs are 10.15.21.142 (Pool-0) and 10.15.31.152 (Pool-1); interfaces carrying production VIPs should be at least 10 GbE, as should the OODP interface on the Backup and DR node.

Illustrative reference topology. Not all services are switch-redundant in this layout; where full redundancy is required, additional interfaces and links should be provided and validated for the target environment.

3.4 Non-shared Storage with a Corosync QDevice



Each node owns its disks, presented through an HBA in PCI passthrough (hostpci1 / hostpci2); the two pools stay in sync over a direct point-to-point mirroring path on 10.15.4.0/24, at least 10 GbE. The third vote comes from a Corosync QDevice rather than a third appliance, which is the cheapest quorum available. It provides a Proxmox VE quorum vote only - this topology has no Backup and DR instance.

The QDevice is reachable over Link 0 only - a management-network outage removes both ring 0 and the QDevice vote.

Illustrative reference topology. Not all services are switch-redundant in this layout; where full redundancy is required, additional interfaces and links should be provided and validated for the target environment.

3.5 IP Addressing Plan

Network / purpose	Bridge / iface	PROXMOX VE pve1 / pve2 / pve3	Open-E JovianDSS node-a / node-b	Virtual IPs and notes
10.15.0.0/24 management + Corosync Link 0	vmbr0 (nic0)	pve1: 10.15.0.141/24 pve2: 10.15.0.151/24 pve3: 10.15.0.161/24	node-a: bond0 10.15.0.142 node-b: bond0 10.15.0.152	DR instance: 10.15.0.163 Proxmox Backup Server: 10.15.0.162
10.15.1.0/24 Corosync Link 1 + REST base	vmbr2 (nic2)	pve1: 10.15.1.141/24 pve2: 10.15.1.151/24 pve3: 10.15.1.161/24	node-a: eth2 10.15.1.142 node-b: eth2 10.15.1.152	rest1: 10.15.20.142 (Pool-0) rest2: 10.15.30.152 (Pool-1)
10.15.2.0/24 NFS data	vmbr3 (bond0)	pve1: 10.15.2.141/24 pve2: 10.15.2.151/24 pve3: 10.15.2.161/24	node-a: eth6 10.15.2.142 node-b: eth6 10.15.2.152	data1: 10.15.21.142 (Pool-0) data2: 10.15.31.152 (Pool-1)
10.15.4.0/24 storage mirror (point-to-point)	passthrough NIC	—	node-a: eth5 10.15.4.142 node-b: eth5 10.15.4.152	ConnectX-5 at 100 Gbps in this lab. Requirement: at least 10 GbE, RDMA preferred (2.1). Topologies 3.2 and 3.4.
10.15.50.0/24 migration network	nic11 (no bridge)	pve1: 10.15.50.141/24 pve2: 10.15.50.151/24 pve3: —	—	pve3 has no migration NIC - it runs no guest workload to migrate.
BMC / IPMI (ping node targets)	—	pve1: 10.15.0.140 pve2: 10.15.0.150 pve3: 10.15.0.160	—	Outside the storage cluster and independent of the hypervisor (6.1).

VIP subnets are interface aliases on the Proxmox side (vmbr2:20 / :30 for REST, vmbr3:21 / :31 for NFS data - chapter 4.3) and pool Virtual IPs on the Open-E JovianDSS side (chapter 6.3). Each pool has exactly ONE data VIP. The topology diagrams in 3.2-3.4 carry the same addresses, so that the architectural options can be read against this plan directly.

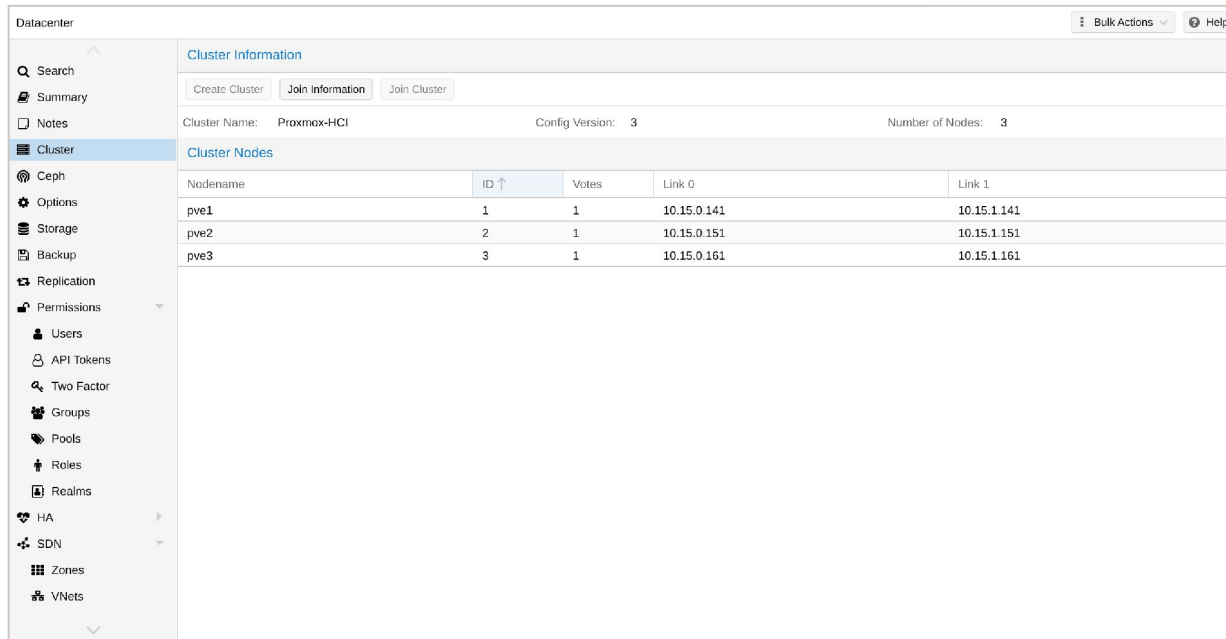


Chapter 4

Proxmox VE Cluster Setup

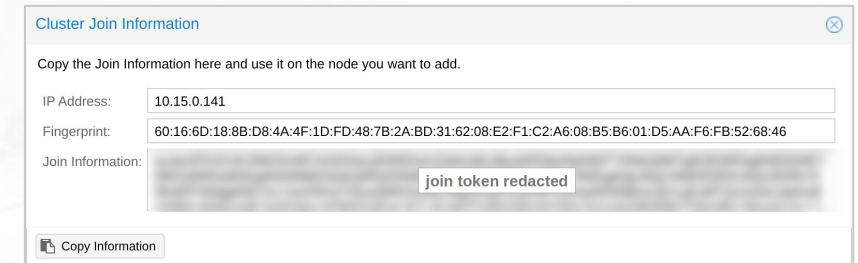
Cluster creation, bridges, VIP aliases, migration network, HA rules, fencing

4.1 Install and Create the Cluster



The screenshot shows the Proxmox Datacenter web interface. On the left is a sidebar with navigation options: Search, Summary, Notes, Cluster (selected), Ceph, Options, Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms, HA, SDN, Zones, and VNETs. The main content area is titled 'Cluster Information' and has tabs for 'Create Cluster', 'Join Information', and 'Join Cluster'. Below these tabs, it shows 'Cluster Name: Proxmox-HCI', 'Config Version: 3', and 'Number of Nodes: 3'. A 'Cluster Nodes' table lists the nodes:

Nodename	ID ↑	Votes	Link 0	Link 1
pve1	1	1	10.15.0.141	10.15.1.141
pve2	2	1	10.15.0.151	10.15.1.151
pve3	3	1	10.15.0.161	10.15.1.161



The 'Cluster Join Information' dialog box contains the following fields:

- IP Address: 10.15.0.141
- Fingerprint: 60:16:6D:18:8B:D8:4A:4F:1D:FD:48:7B:2A:BD:31:62:08:E2:F1:C2:A6:08:B5:B6:01:D5:AA:F6:FB:52:68:46
- Join Information: [redacted] join token redacted

A 'Copy Information' button is located at the bottom left of the dialog.

Install Proxmox VE on all three nodes

The system disk must sit on a controller that will NOT be passed through to the Open-E JovianDSS VMs. That controller is assigned to the storage VM and is no longer available to the hypervisor, which must still boot without it.

Datacenter -> Cluster lists the nodes; Join Information carries the fingerprint a joining node needs.

```
root@pve1:~# pvecmd create Proxmox-HCI --link0 10.15.0.141 --link1 10.15.1.141
root@pve2:~# pvecmd add 10.15.0.141 --link0 10.15.0.151 --link1 10.15.1.151
root@pve3:~# pvecmd add 10.15.0.141 --link0 10.15.0.161 --link1 10.15.1.161
```

Corosync is configured with two links: Link 0 on the management bridge, Link 1 on the second bridge. The second link is optional but strongly recommended - with a single link, any management-network disturbance can result in loss of quorum. pvecmd status must report all three nodes quorate before the next step.

4.2 Network Bridges and Interface Roles

Node 'pve1'											
Reboot Shutdown Shell Bulk Actions Help											
Create Revert Edit Remove Apply Configuration											
	Name	Alternative Names	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR	Gateway	Comment
	bond0		Linux Bond	Yes	Yes	No	nic7 nic8	balance-alb			
	nic0	enp76s0f0 enx000af721e030	Network Device	Yes	No	No					
	nic1	enp76s0f1 enx000af721e031	Network Device	Yes	Yes	No					
	nic10	enp202s0f1 enxa0369fccbb36	Network Device	No	No	No					
	nic11	enp154s0f0np0 enx043f72dc49d4	Network Device	Yes	Yes	No			10.15.50.141/24		
	nic2	enp76s0f2 enx000af721e032	Network Device	Yes	Yes	No					
	nic3	enp76s0f3 enx000af721e033	Network Device	No	No	No					
	nic4	enx3cecef61368a	Network Device	No	No	No					
	nic5		Network Device	No	No	No					
	nic6	enx3cecef61368b	Network Device	No	No	No					
	nic7	enp75s0f0np0 enx3ctdfee69cf4	Network Device	Yes	Yes	No					
	nic8	enp75s0f1np1 enx3ctdfee69cf5	Network Device	Yes	Yes	No					
	nic9	enp202s0f0 enxa0369fccbb34	Network Device	No	No	No					
	vmbr0		Linux Bridge	Yes	Yes	No	nic0		10.15.0.141/24	10.15.0.1	
	vmbr1		Linux Bridge	Yes	Yes	No	nic1				
	vmbr2		Linux Bridge	Yes	Yes	No	nic2		10.15.1.141/24		
	vmbr2:20		Unknown	No	No	No			10.15.20.141/24		
	vmbr2:30		Unknown	No	No	No			10.15.30.141/24		

Create: Linux Bridge

Name: vmbr4

Autostart: ☒

IPv4/CIDR:

VLAN aware: ☐

Gateway (IPv4):

Bridge ports:

IPv6/CIDR:

Comment:

Gateway (IPv6):

Help

Advanced ☐

Create

System -> Network -> Create -> Linux Bridge creates each bridge; Apply Configuration activates the change.

Four bridges, each with one role

vmbr0 (nic0) - management and Corosync Link 0; also the management interface of the storage VMs.

vmbr1 (nic1) - no address; the second leg of the Open-E JovianDSS management bond.

vmbr2 (nic2) - Corosync Link 1 and the base subnet for the REST control addresses.

vmbr3 (bond0) - NFS data. The only bridge that carries storage traffic.

The bond carries no IP address

bond0 joins two NICs in balance-alb mode and stays without an address; vmbr3 sits on top and holds the addressing. An IP placed on the bond itself is a common mistake and prevents the bridge above it from working as expected.

Roles stay separated: one subnet shared between Corosync and NFS data works until the first burst of storage traffic, which is precisely when Corosync must stay responsive.

4.3 VIP Subnet Aliases

Each pool VIP resides in its own subnet, added as an alias

The Proxmox nodes need an address in every subnet in which a pool VIP can appear - one alias per pool, per role. The GUI creates bridges and bonds but not aliases, so these entries are added to `/etc/network/interfaces`:

```
iface vmbr2:20 inet static      # REST control, Pool-0
    address 10.15.20.141/24
iface vmbr2:30 inet static      # REST control, Pool-1
    address 10.15.30.141/24
iface vmbr3:21 inet static      # NFS data, Pool-0
    address 10.15.21.141/24
iface vmbr3:31 inet static      # NFS data, Pool-1
    address 10.15.31.141/24
```

The GUI lists alias addresses with type “Unknown” (visible in the capture in 4.2) - that is expected, not a fault.

The same entries are repeated on every node with its own host octet. Each node needs a foothold in all four subnets, because a pool - and its VIPs - can land on any of them.

4.4 Dedicated Migration Network

Edit: Migration Settings

Type:

secure

Network:

10.15.50.0/24

X

▼

CIDR

Interface

10.15.50.141/24

nic11

Live migration is kept off the management network

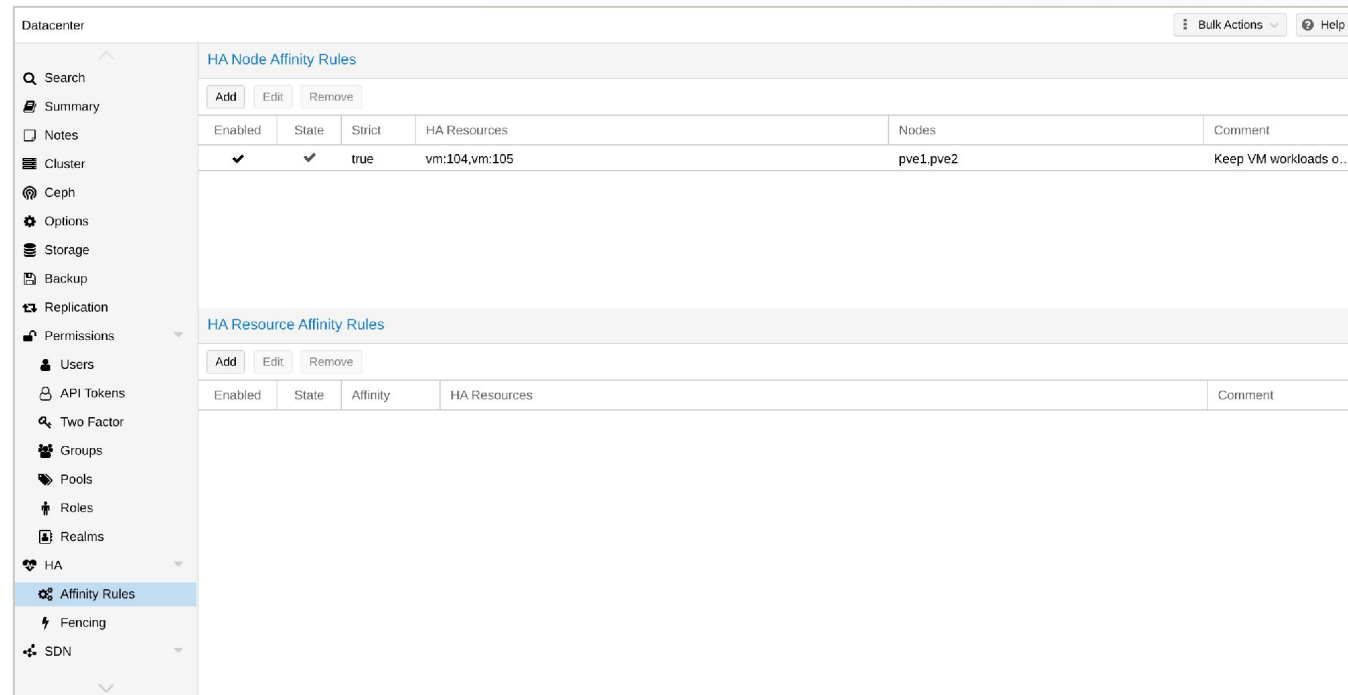
Migration moves guest memory as fast as the link allows. On the management network it competes with the GUI and with Corosync. Datacenter -> Options -> Migration Settings sets the network and the transport; the same values can be written directly to `/etc/pve/datacenter.cfg`.

```
ha: shutdown_policy=migrate
migration: secure,network=10.15.50.0/24
```

Only nodes that run guests need this interface

In this design pve1 and pve2 have a dedicated migration NIC and pve3 does not - it holds quorum and the Backup and DR instance and runs no guest workload, so it has nothing to migrate. The HA rule in 4.5 enforces that, which is what makes the missing interface safe rather than an oversight.

4.5 HA Rules and Resources



The screenshot shows the Proxmox VE 9 web interface for configuring High Availability (HA) rules. The left sidebar contains a navigation menu with options like Search, Summary, Notes, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms, HA, Affinity Rules (selected), Fencing, and SDN. The main content area is titled 'HA Node Affinity Rules' and includes a table with columns: Enabled, State, Strict, HA Resources, Nodes, and Comment. A single rule is listed with Enabled checked, State checked, Strict set to true, HA Resources set to 'vm:104,vm:105', Nodes set to 'pve1,pve2', and a comment 'Keep VM workloads o...'. Below this, the 'HA Resource Affinity Rules' section is empty, showing a table with columns: Enabled, State, Affinity, HA Resources, and Comment.

Add the guests, then constrain where they may run

Proxmox VE 9 uses rules, not groups. Each VM is added under HA -> Resources; a node-affinity rule then lists only pve1 and pve2 and is marked strict, so production never lands on the quorum node.

Resource Affinity stays empty here. A negative rule keeping the two guests apart appears prudent, but with only two permitted nodes it would leave a guest without a permitted target node when one of them fails - disabling the failover this design exists for.

4.6 Fencing and HA Shutdown Policy

Search

Summary

Notes

Cluster

Ceph

Options

Storage

Backup

Replication

Permissions

Users

API Tokens

Two Factor

Groups

Pools

Roles

Realms

HA

SDN

Zones

VNets

Arm HA

Disarm HA

CRS Settings

Status

Type	Status
quorum	OK
fencing	armed (CRM watchdog active)
master	pve1 (active, Fri Aug 21 10:55:39 2026)
lrm	pve1 (active, watchdog active, Fri Aug 21 10:55:34 2026)
lrm	pve2 (active, watchdog active, Fri Aug 21 10:55:34 2026)
lrm	pve3 (idle, watchdog standby, Fri Aug 21 10:55:36 2026)

Resources

Add

Edit

Remove

ID	State	Node	Name	Max. Restart	Max. Reloc...	Failback	Auto-Rebal...	Description
vm:104	started	pve1	ubuntu-nfs	1	1	true	true	NFS demo VM (Linux) - guide chapter 4.5
vm:105	started	pve2	win2025-nfs	1	1	true	true	NFS demo VM (Windows) - guide chapter 4.5

Edit: HA Settings

Shutdown Policy: migrate

Help

OK

Fencing runs on a watchdog

In /etc/default/pve-ha-manager, a commented-out WATCHDOG_MODULE keeps the softdog software fallback; WATCHDOG_MODULE=ipmi_watchdog switches to the hardware watchdog, which is the more reliable choice because it survives a hung kernel. Datacenter -> Options -> HA Settings sets the shutdown policy; migrate moves guests on a planned node shutdown instead of restarting them elsewhere.

The reported status matches the configuration: pve1 and pve2 show watchdog active because they hold HA resources, while pve3 remains idle with the watchdog on standby - evidence that the node-affinity rule is in force.



Chapter 5

Creating the Open-E JovianDSS VMs

The VM wizard, step by step, for the storage appliances

5.1 Create the VM: General and OS

The screenshot shows the 'Create: Virtual Machine' dialog with the 'General' tab selected. The 'Node' is set to 'pve1', 'VM ID' is '100', and 'Name' is 'jovianoss-node-a'. 'Add to HA' is unchecked. 'Start at boot' is checked. 'Start/Shutdown order' is '1', 'Startup delay' is '300', and 'Shutdown timeout' is '180'. 'vCPU Architecture' is 'Default (Host Architecture)'. There are no tags. At the bottom, 'Advanced' is checked, and 'Back' and 'Next' buttons are visible.

General

Select pve1, click Create VM, name the first Open-E JovianDSS instance and tick Advanced to reach the boot settings: Start at boot enabled, Start / Shutdown order 1, Startup delay 300 s, Shutdown timeout 180 s.

The screenshot shows the 'Create: Virtual Machine' dialog with the 'OS' tab selected. 'Use CD/DVD disc image file (iso)' is selected. 'Storage' is 'local' and 'ISO image' is '.0up33_b65410.oe_i.iso'. 'Guest OS' is 'Linux' and 'Version' is '7.x - 2.6 Kernel'. 'Use physical CD/DVD Drive' and 'Do not use any media' are unselected. At the bottom, 'Advanced' is checked, and 'Back' and 'Next' buttons are visible.

Why these values

The storage appliance must be up and exporting NFS before anything that depends on it starts. Order 1 places it first and the 300-second delay gives it five minutes to bring pools and shares online. All of it remains changeable later under VM -> Options.

OS: select the uploaded Open-E JovianDSS ISO and set Guest OS to Linux (6.x - 2.6 Kernel).

5.2 Create the VM: System and Disks

The screenshot shows the 'Create: Virtual Machine' wizard with the 'System' tab selected. The 'General' tab is also visible. The 'System' tab contains the following settings:

- Graphic card: Default
- Machine: Default (i440fx)
- Firmware: BIOS
- BIOS: Default (SeaBIOS)
- SCSI Controller: VirtIO SCSI single
- Qemu Agent: ☒
- Add TPM: ☐

At the bottom, there is a 'Help' button, an 'Advanced' checkbox (checked), and 'Back' and 'Next' buttons.

System

Tick QEMU Guest Agent. The remaining defaults are suitable - the appliance needs no special machine type or BIOS.

The screenshot shows the 'Create: Virtual Machine' wizard with the 'Disks' tab selected. The 'System' tab is also visible. The 'Disks' tab contains the following settings:

- scsi0: ☒ (selected)
- Bus/Device: SCSI 0
- Cache: Default (No cache)
- SCSI Controller: VirtIO SCSI single
- Discard: ☐
- Storage: local-lvm
- IO thread: ☒
- Disk size (GiB): 32
- Format: Raw disk image (raw)
- SSD emulation: ☐
- Backup: ☒
- Read-only: ☐
- Skip replication: ☐
- Async IO: Default (io_uring)

At the bottom, there is a 'Help' button, an 'Advanced' checkbox (checked), and 'Back' and 'Next' buttons.

Disks: keep the system disk on local-lvm

This is the one wizard setting that must be chosen deliberately. The appliance disk belongs on local-lvm - storage owned by the hypervisor - because Open-E JovianDSS has to boot before the storage it serves exists. The wizard offers the first shared storage by default, which here would make the VM depend on itself. 32 GiB is sufficient.

5.3 Create the VM: CPU, Memory, Network and Confirm

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

Sockets: 1 Type: Default (x86-64-v2-AES)

Cores: 4 Total cores: 4

VCPUs: 4 CPU units: 100

CPU limit: unlimited Enable NUMA: ☐

CPU Affinity: All Cores

Extra CPU Flags:

State	Value	Flag	Description	Supported On
Default	☐ ☒ ☐ ☐	nested-virt	Controls nested virtualization, namely 'svm' for AMD CPUs and 'vmx' for Intel CPUs. Live migration still only works if it's the same flag on both sides. Use a CPU model similar to the host, with the same vendor, not x86-64-vX!	pve1, pve2, pve3
Default	☐ ☒ ☐ ☐	aes	Activate AES instruction set for HW acceleration.	pve1, pve2, pve3
Default	☐ ☐ ☒ ☐	amd-ssbd	Improves Spectre mitigation performance with AMD	pve1, pve2

☒ Only show flags supported by at least one node Showing flags for KVM

Help Advanced ☒ Back Next

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

Memory (MiB): 65536

Minimum memory (MiB): 65536

Shares: Default (1000)

Ballooning Device: ☒

Allow KSM: ☒

Help Advanced ☒ Back Next

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

☐ No network device

Bridge: vmbrio Model: VirtIO (paravirtualized)

VLAN Tag: no VLAN MAC address: auto

Firewall: ☒

Disconnect: ☐ Rate limit (MB/s): unlimited

MTU: Same as bridge Multiqueue:

Help Advanced ☒ Back Next

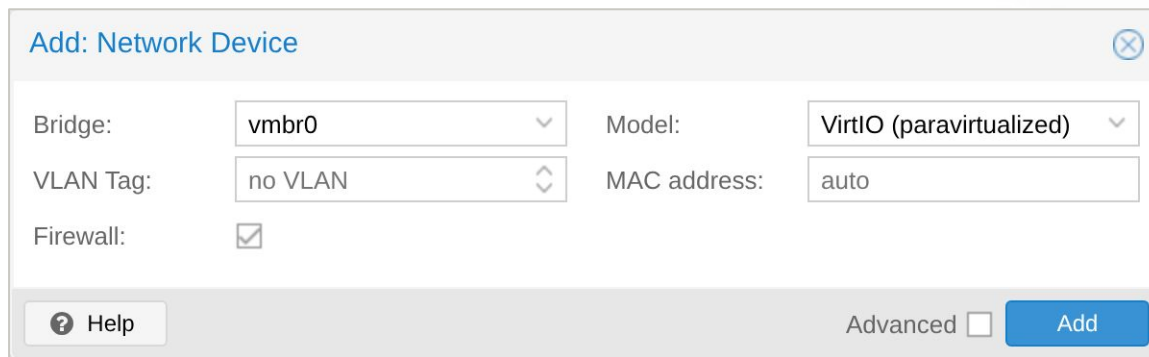
Size the appliance for the pool it will serve

Set sockets and cores under CPU and Memory in MiB. ZFS uses RAM as its primary cache, so memory is what determines read performance; the storage VMs in this guide run with 64 GB. Deduplication, if it is ever enabled, raises that requirement sharply.

One network device only

The wizard cannot add more. Attach net0 to the management bridge, review the summary and click Finish. The second interface and the passthrough devices are added afterwards, on the created VM (5.4).

5.4 Add the Second NIC and the PCI Devices



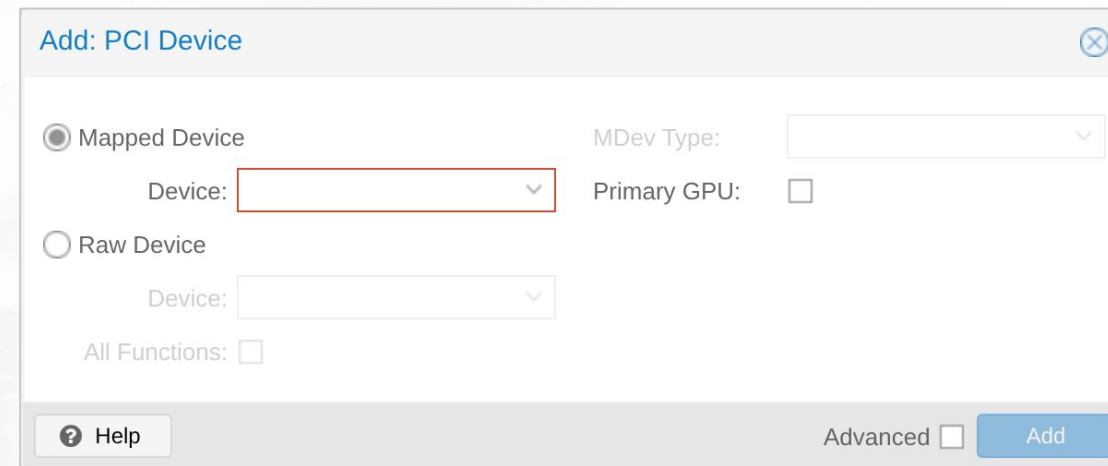
Add: Network Device

Bridge: Model:

VLAN Tag: MAC address:

Firewall: ☒

[? Help](#) ☐ Advanced



Add: PCI Device

☒ Mapped Device ☐ Raw Device

Device: MDev Type: Primary GPU: ☐

Device: All Functions: ☐

[? Help](#) ☐ Advanced

The settings unavailable in the wizard are applied to the created VM: Hardware -> Add

Network Device (net1) - attached to the NFS data bridge. Two interfaces for the two physical links are not required: the bond underneath the bridge already provides that.

PCI Device - the HBA, NVMe controller or RAID controller that owns the disks, plus the NIC ports used for the mirror path. The whole controller then belongs to the VM and is managed inside it. On all-NVMe systems, where disks sit directly in motherboard PCIe slots, each disk is passed through individually.

Order matters for the disks

A controller passed through disappears from the hypervisor. The Proxmox VE system disk must not be on it (5.2), or the node will not boot once the controller is assigned to the VM.

The VM is then started and Open-E JovianDSS installed. The procedure repeats for node-b and, in the topologies with a Backup and DR node, for the third instance.

5.5 Hardware Summary

Virtual Machine 1111 (JovianDSS-nodeA) on node 'pve1' No Tags

Start Shutdown Migrate Console More Help

Summary

Console

Hardware

Cloud-Init

Options

Task History

Monitor

Backup

Replication

Snapshots

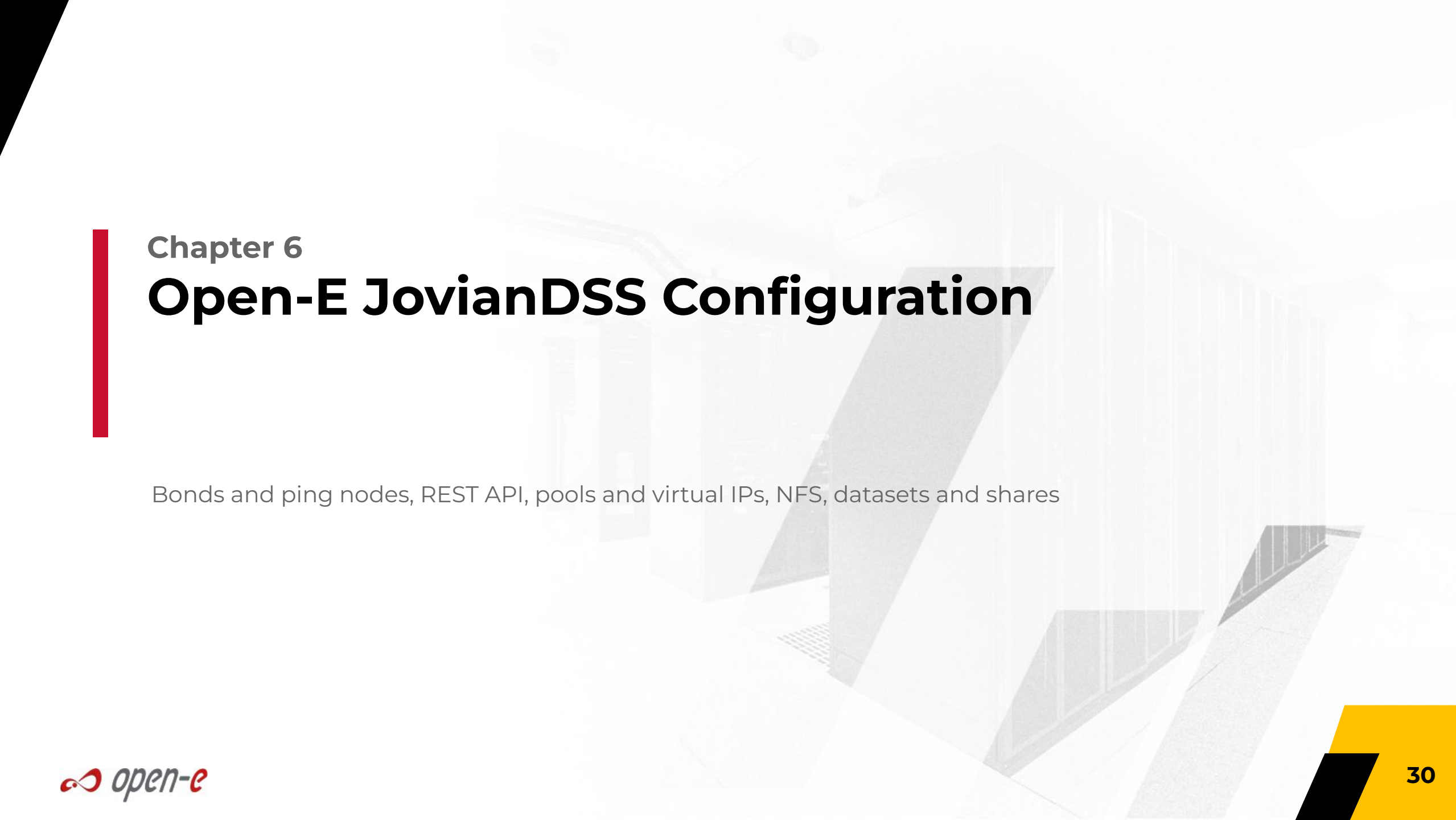
Firewall

Permissions

Add Remove Edit Disk Action Revert

Memory	64.00 GiB
Processors	8 (1 sockets, 8 cores) [host]
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	none,media=cdrom
Hard Disk (scsi0)	local-lvm:vm-1111-disk-0,aio=native,cache=directsync,iotread=1,size=100G
Network Device (net0)	virtio=BC:24:11:D6:E7:9E,bridge=vmbr0,firewall=1
Network Device (net1)	virtio=BC:24:11:DE:E1:F1,bridge=vmbr1,firewall=1
Network Device (net2)	virtio=BC:24:11:9F:C4:0F,bridge=vmbr2,firewall=1
Network Device (net3)	virtio=BC:24:11:65:07:24,bridge=vmbr3,firewall=1
PCI Device (hostpci0)	0000:cb:00.0
PCI Device (hostpci1)	0000:cc:00.0
PCI Device (hostpci2)	0000:b1:00.0
PCI Device (hostpci3)	0000:b2:00.0
PCI Device (hostpci4)	0000:99:00.0
PCI Device (hostpci5)	0000:9a:00.1

A correctly built storage appliance: system disk on local-lvm, two network devices - management and NFS data - and the passthrough devices carrying the disks and the mirror path. The VM is started and Open-E JovianDSS installed only once the hardware list looks like this.



Chapter 6

Open-E JovianDSS Configuration

Bonds and ping nodes, REST API, pools and virtual IPs, NFS, datasets and shares

6.1 Interfaces, Bond and Ping Nodes

Interfaces

+ Add VLAN

+ Create bond interface

Name	IP	DHCP	Vendor	Negotiated speed	Cable	Status	Options
bond0	10.15.0.142	No	Ethernet Bonding Driver	N/A	cable	Active	Options
eth0 (bond0)	N/A	No	Red Hat, Inc. Virtio network device	N/A	cable	Bond slave	Options
eth1 (bond0)	N/A	No	Red Hat, Inc. Virtio network device	N/A	cable	Bond slave	Options
eth2	10.15.1.142	No	Red Hat, Inc. Virtio network device	N/A	cable	Active	Options
eth5	10.15.4.142	No	Mellanox Technologies MT27800 Family [ConnectX-5]	100000 Mbps	cable	Active	Options
eth6	10.15.2.142	No	Red Hat, Inc. Virtio network device	N/A	cable	Active	Options

Editing rings not allowed while failover is started
In order to add or remove a ring please stop the failover first.

+ Add ring

Local node interface	Remote node interface	Status	Options
1 bond0 (10.15.0.142)	bond0 (10.15.0.152)	Active	

+ Add ping node

IP	Local status	Remote status	Options
1 10.15.0.140	Reachable	Reachable	✕ Delete
2 10.15.0.150	Reachable	Reachable	✕ Delete
3 10.15.0.160	Reachable	Reachable	✕ Delete

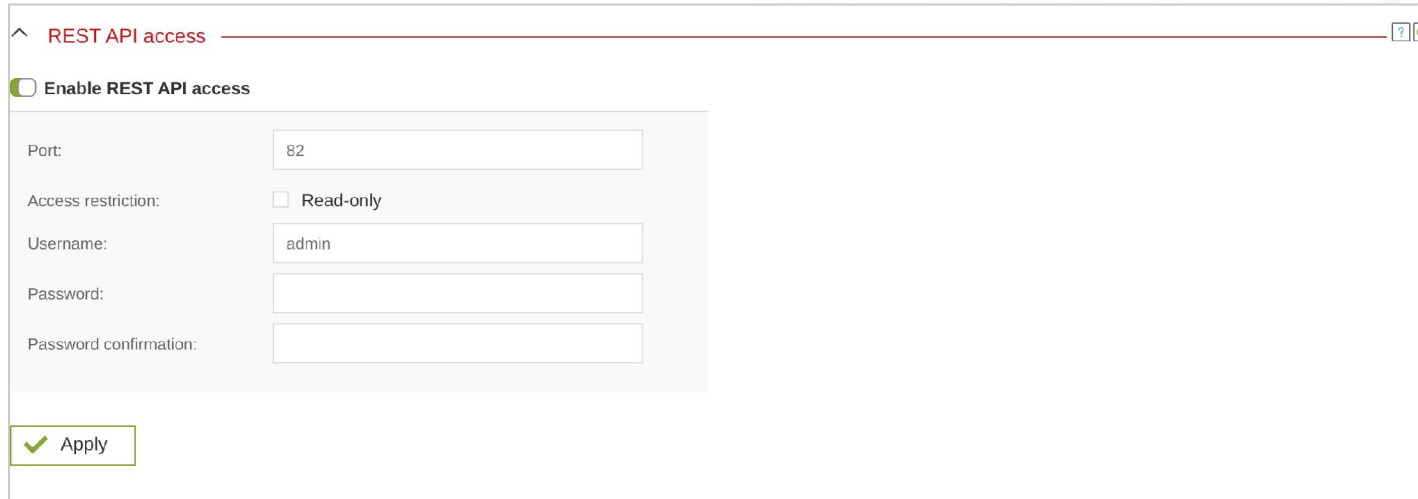
No bond is needed inside Open-E JovianDSS for data access

The data path is already redundant one layer down: the Proxmox bridge that carries NFS sits on a bond of two physical NICs (4.2). System Settings -> Network -> Create bond interface remains available and is used for the management interface and, where required, for the mirror path, where active-backup is the correct mode. The rings list (left) pairs each local interface with its counterpart on the partner node: bond0 for management, eth5 for the mirror path.

Ping nodes never point at the local hypervisor

The storage cluster uses ping nodes to determine whether a node is still reachable. In a hyperconverged build the storage node is a VM on the host that would be the obvious target, and that link runs through a virtual switch inside the host, so it can never fail. This lab uses the BMC addresses of the three servers - outside the storage cluster and independent of the hypervisor. At least two are required.

6.2 Enable the REST API



The screenshot shows a configuration window titled "REST API access" with a red header bar. Below the title, there is a checkbox labeled "Enable REST API access" which is currently unchecked. Underneath, there is a form with the following fields: "Port:" with a value of "82", "Access restriction:" with a "Read-only" checkbox, "Username:" with a value of "admin", "Password:" (empty), and "Password confirmation:" (empty). At the bottom left of the window, there is a green "Apply" button with a checkmark icon.

The plugin uses the REST API

Every Open-E JovianDSS instance used as a backend needs REST API access enabled - the plugin uses the REST API for all snapshot operations: creating, cloning and deleting snapshots, and performing rollback.

System Settings -> Administration -> REST API access. Enable it if it is not already on, and click Apply.

This applies to node-a, node-b and, where present, the Backup and DR instance.

The REST endpoint responds on its own control VIP, separate from the data VIP - control traffic never shares the NFS path.

6.3 Pools and Virtual IPs

StatusDisk GroupsiSCSI TargetsFC TargetsSharesSnapshotsVirtual IPsConfiguration

Virtual IPs

Search

+ Add virtual IP

	Virtual IP	Name	Netmask	Network interface	Remote network interface	State	
1	10.15.21.142	data1	255.255.255.0	eth6 (10.15.2.142)	eth6 (10.15.2.152)	Active	
2	10.15.20.142	rest1	255.255.255.0	eth2 (10.15.1.142)	eth2 (10.15.1.152)	Active	

Virtual IPs routing

Search

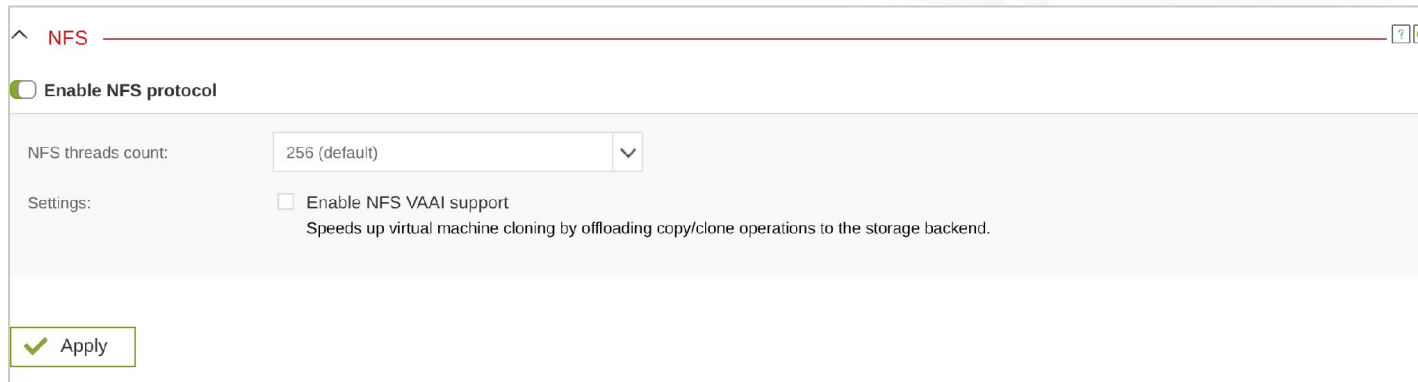
+ Add static routing

Network/Host IP	Netmask	Gateway	Virtual IP name
No items found.			

Two VIPs per pool: one for REST control, one for NFS data. Each names a local interface and the matching remote interface on the partner node - that pairing is what allows the address to move when the pool fails over. For active-active operation, Pool-0 is imported on node-a and Pool-1 on node-b, so both nodes serve data and either can take the pool of the other.

33

6.4 Enable the NFS Protocol



^ NFS

☐ Enable NFS protocol

NFS threads count: 256 (default) ▼

Settings:

☐ Enable NFS VAAI support
Speeds up virtual machine cloning by offloading copy/clone operations to the storage backend.

✓ Apply

NFS is off on a fresh install

A newly installed Open-E JovianDSS does not serve NFS until the protocol is enabled: Storage Settings -> NFS, enable, apply.

If this step is missed, everything else looks correct - the dataset exists, the share exists, the VIP responds to ping - while Proxmox VE cannot mount the export.

The protocol is enabled on every instance that serves storage, the DR node included.

6.5 Create the Dataset and the NFS Share

Add new dataset

Encryption settings

Encryption requires a passphrase in Pool → Configuration → Resource encryption.

☐ Encrypt resource

Dataset properties

Name:

Deduplication:

Number of the data copies:

Compression:

Record size:

Write cache sync requests:

Write cache sync request handling (logbias):

Read cache (primary, ARC) scope:

Read cache (secondary, L2ARC) scope:

Access time:

Small data blocks policy

Available only when a special devices group exists.

Small data block size:

☐ Enable quota

☐ Enable reservation

Share wizard

1. Share properties

Share properties

Name:

Path:

If you want to share whole dataset leave this input empty.

Comment:

Storage Management -> pool -> Shares -> Add dataset

Record size stays at 128 KiB, a balance of IOPS and bandwidth for VM disks; 64 KiB or 16 KiB favour IOPS where the workload is dominated by small random writes. Write cache sync requests is set to Always, which is what makes the dataset safe for virtual machine disks.

Dataset Options -> Add share

The share name is the value Proxmox VE puts in its Export field (7.2). Synchronous Data Record is required for a High Availability cluster.

Access is then restricted by IP to the addresses of the Proxmox nodes in the NFS data subnet. A node left off the list cannot write, and a VM running from that node will fail - so ACLs are changed only while the guests using the share are stopped.

6.6 Verify the Open-E JovianDSS Side

Five conditions before Proxmox VE is touched

- REST API access is enabled and responds on the control VIP
- The NFS protocol is enabled
- Both pool VIPs are Active and paired with a remote interface
- The dataset exists with sync set to Always
- The share is active and its NFS service is on

```
# from a Proxmox node - the export must be visible before the storage is registered
root@pve1:~# showmount -e 10.15.21.142
```

```
# NOTE: showmount prints the share's path alias, while the storage.cfg
# "export" field needs the FULL dataset path - see chapter 7.2.
```

An empty showmount result is a stop condition. Every later symptom - a storage that will not activate, a VM that will not start - traces back to this point, and diagnosing it from the Proxmox side is considerably more time-consuming.



Chapter 7

Plugin Installation and Storage Registration

Installation on all nodes, storage registration, REST tuning, mount options

7.1 Install the Plugin on All Nodes

One command, run once, from any node in the cluster

The installer deploys the plugin to every cluster member, so it is not repeated per node:

```
root@pve1:~# curl -fsSL https://raw.githubusercontent.com/open-e/JovianDSS-Proxmox/main/install.pl \  
| perl - --all-nodes
```

The installer restarts every service that loads a storage plugin, on every node:

```
systemctl restart pvedaemon pvestatd pveproxy pve-ha-lrm.service pve-ha-crm.service
```

Confirm the version before continuing

`dpkg -l | grep joviandss` must report the same version on every node. A cluster running two plugin versions produces failures that appear random and are difficult to diagnose. v1.1.0-pre1 added pvestatd and pveproxy to that list; before it they kept running the previous plugin code after an upgrade, which is worth knowing when an older build is upgraded by hand. Older documentation points the installer at the nfs-plugin branch; that URL still resolves, but main is the repository default.

7.2 Register the Storage

The GUI cannot add this storage type - Datacenter -> Storage -> Add does not list it (right). The supported method is `/etc/pve/storage.cfg` or `pvesm add`:

```
joviandss-nfs: nfs-Pool-0
server 10.15.21.142          # the pool's DATA virtual IP
export /Pools/Pool-0/nfs-Pool-0 # FULL dataset path
path /mnt/pve/nfs-Pool-0    # local mount point
control_addresses 10.15.20.142 # REST control VIP
user_name admin             # password: pvesm set, NOT this file
ssl_cert_verify 0
content images,rootdir
options vers=3,nconnect=4
shared 1
```

	Directory
	LVM
	LVM-Thin
	BTRFS
	NFS
	SMB/CIFS
	iSCSI
	CephFS
	RBD
	ZFS over iSCSI
	ZFS
	Proxmox Backup Server
	ESXi

Two fields cause most of the failures

`export` needs the full dataset path, not the value `showmount` prints - `showmount` shows the share's path alias, and copying it produces a storage that will not mount. `server` is the data VIP while `control_addresses` is the REST VIP.

The REST password is not part of `storage.cfg`

It is supplied as `--user_password` to `pvesm add` or `pvesm set` and held outside the configuration file. From v1.1.0-pre1 the NFS plugin requires it: a storage created without one is refused outright rather than registering and failing later on the first snapshot, and a storage carrying the password inline in `storage.cfg` is refused with instructions to set it again.

7.3 REST Timeouts and Retries

Five optional properties, added in plugin v1.1.0-pre1

They govern how the plugin talks to the appliance over REST. Left unset they reproduce the behaviour of earlier versions, so an existing storage needs no change:

```
root@pve1:~# pvesm set nfs-Pool-0 --jdssc_rest_connect_timeout 10
```

jdssc_rest_connect_timeout	5	seconds	TCP connect to one address
jdssc_rest_read_timeout	570	seconds	inactivity, not a deadline
jdssc_rest_request_send_cycle_attempts	17	rounds	over all control addresses
jdssc_rest_request_send_cycle_delay	3	seconds	sleep between rounds
jdssc_rest_send_retry_on_decode_error_attempts	5	attempts	undecodable JSON reply

The connect timeout is what bounds failover

Five seconds to open a TCP connection to one control address, after which the next address in control_addresses is tried. Earlier versions had a single long timeout covering the whole request, so an unreachable controller could stall an operation for minutes.

The read timeout is an inactivity timeout

It measures the gap between bytes of a response, not the length of the response, so a slow but steady transfer is never cut off. The retry cycle sits above both: each round tries every control address, then sleeps, up to the configured number of rounds. Raising these belongs to an appliance that answers REST slowly under load, or a control link with a high round-trip time - and it is done with pvesm set or storage.cfg, because the GUI does not edit this storage type (7.5).

7.4 Mount Options and nconnect

options vers=3,nconnect=4

nconnect opens several TCP connections to the same server address. It widens one path - it does not create a second one, and it is not redundancy. Verification takes two commands, because the declared option and the live mount can differ:

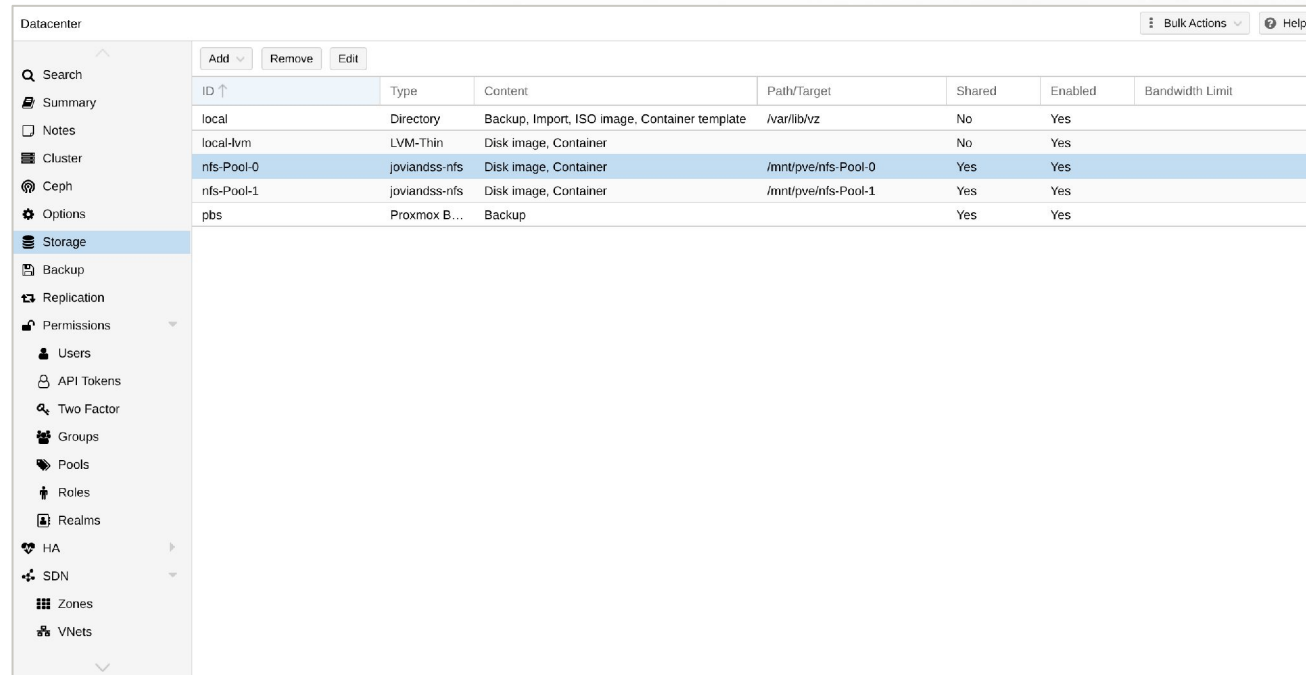
```
root@pve1:~# findmnt -no OPTIONS /mnt/pve/nfs-Pool-0
rw,relatime,vers=3,rsiz=1048576,wsiz=1048576,hard,proto=tcp,nconnect=4,...

root@pve1:~# ss -tn state established '( dport = :2049 )' | grep -c 10.15.21.142
4
```

Changing options does NOT affect a storage that is already mounted

Proxmox VE never re-applies options to a live NFS mount, so an option added to storage.cfg after the first mount is not in effect, and nothing warns about it. The procedure is: stop everything running from that storage and umount the path; pvesm status then remounts it. Every node needs checking - a single node missing nconnect is easy to overlook and halves the expected throughput.

7.5 Changing the Storage Configuration



ID ↑	Type	Content	Path/Target	Shared	Enabled	Bandwidth Limit
local	Directory	Backup, Import, ISO image, Container template	/var/lib/vz	No	Yes	
local-lvm	LVM-Thin	Disk image, Container		No	Yes	
nfs-Pool-0	joviandss-nfs	Disk image, Container	/mnt/pve/nfs-Pool-0	Yes	Yes	
nfs-Pool-1	joviandss-nfs	Disk image, Container	/mnt/pve/nfs-Pool-1	Yes	Yes	
pbs	Proxmox B...	Backup		Yes	Yes	

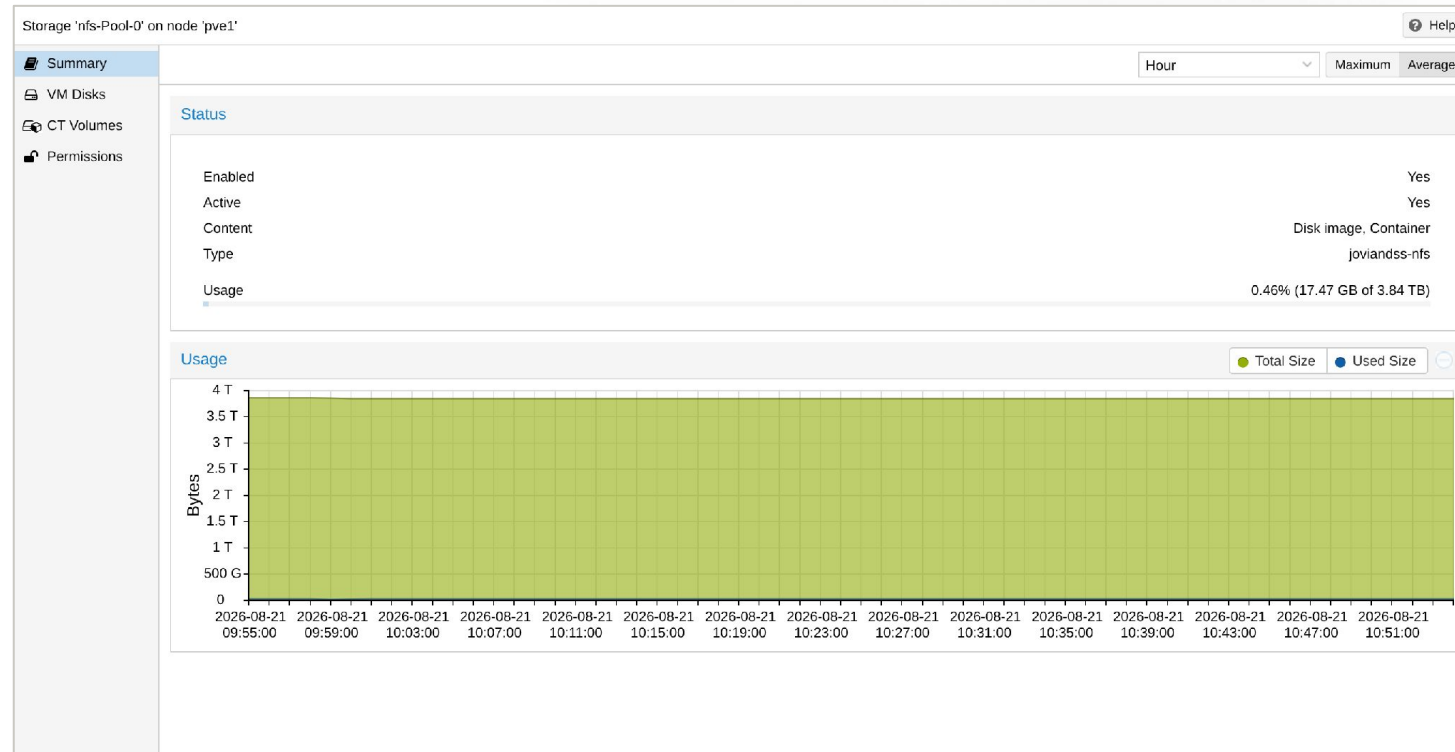
The GUI lists this storage but cannot edit it

Double-clicking the entry, or selecting it and pressing Edit, does nothing: Proxmox VE raises an internal error for the joviandss-nfs type because the plugin registers no GUI editor. Built-in storage types open their editor normally, so the empty response is specific to this plugin.

Changes are made from the command line instead - both forms are supported and take effect immediately:

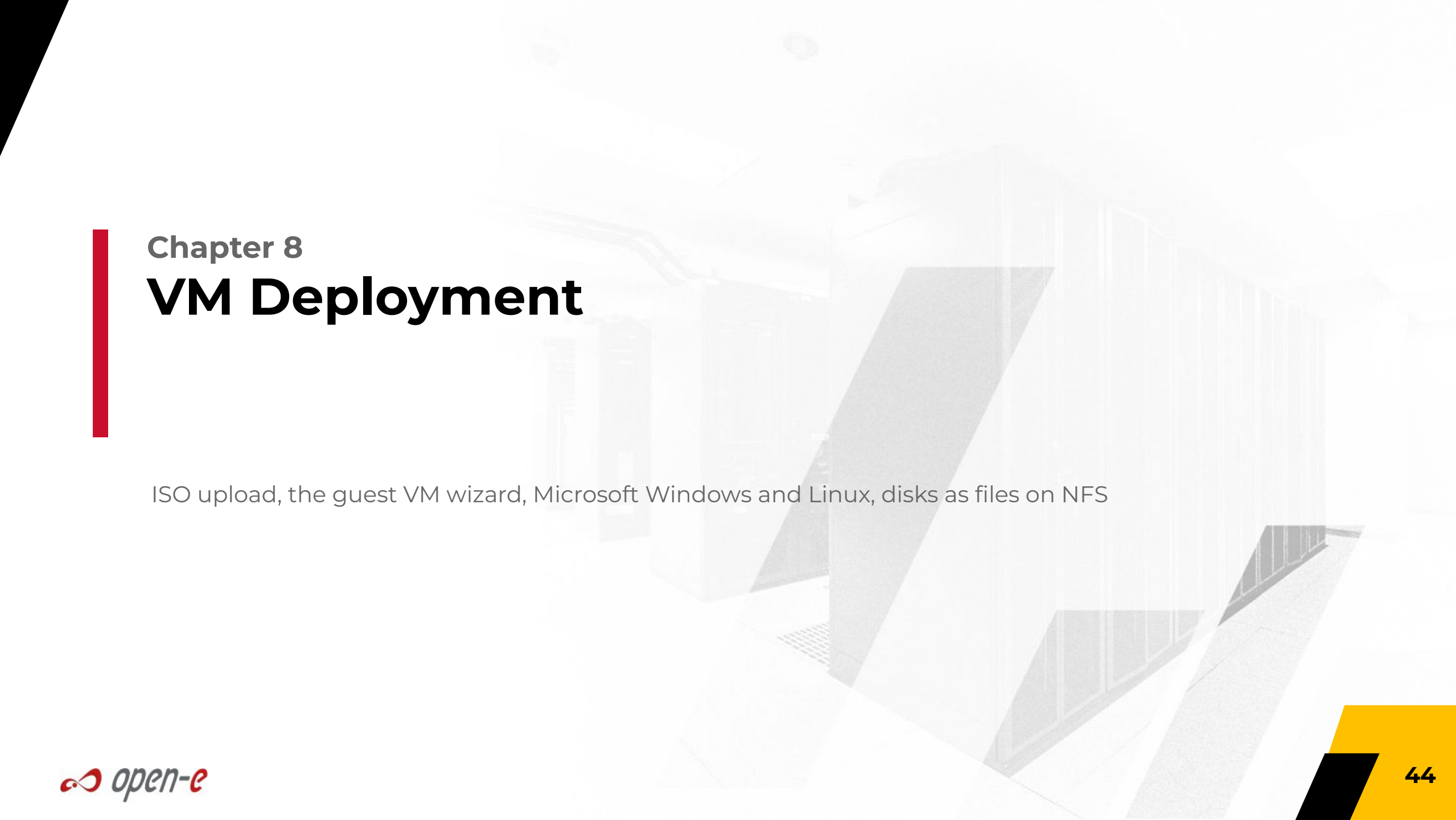
```
root@pve1:~# pvesm set nfs-Pool-0 --content images,rootdir # or edit /etc/pve/storage.cfg
```

7.6 Verify the Storage Registration



The storage must be active on every node before a VM is placed on it

pvesm status lists it as active with the real capacity of the pool. A size of zero, or a storage that activates on one node only, means the export or the ACLs are wrong. Because shared 1 is set, Proxmox VE mounts it cluster-wide, which is what makes migration and HA possible.



Chapter 8

VM Deployment

ISO upload, the guest VM wizard, Microsoft Windows and Linux, disks as files on NFS

8.1 Upload an ISO Image

Storage 'local' on node 'pve1'

SummaryBackupsISO ImagesCT TemplatesImportPermissions

UploadDownload from URLRemove

Search: Name, Format

Name	Date	Format	Size
jdss1.0up33_b65410.oe_i.iso	2026-06-03 13:34:48	iso	1.44 GiB
ubuntu-25.04-desktop-amd64.iso	2026-06-03 13:44:42	iso	5.85 GiB
virtio-win-0.1.285.iso	2026-06-03 13:45:22	iso	753.06 MiB
Windows_Server_2025.iso	2026-06-03 13:32:31	iso	6.82 GiB

Download from URL

URL:

Enter URL to download

Query URL

File name:

Please (re-)query URL to get meta information

File size:

-

MIME type:

-

Advanced

Download

The ISO does not go on the Opwn-E JovianDSS storage

This storage is registered with content images,rootdir - VM disks and container volumes. It does not accept ISO images, which keeps installation media off the production data path.

A storage that declares the iso content type receives it instead; local on any node does so by default: select the node, open local -> ISO Images -> Upload. For large images, Download from URL fetches the file directly onto the node instead of pushing it through the browser. Adding iso to the content list of the shared storage is possible, but only through /etc/pve/storage.cfg or pvesm set - the GUI cannot edit this storage type (7.5).

An ISO on local exists on that one node only, so the VM is created on the node that holds it, or the image is uploaded to every node that might run the installer. Once the guest is installed the ISO no longer matters - the disk is on shared storage and the VM can run and migrate anywhere.

8.2 Create the Guest VM: the Wizard

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

☒ Use CD/DVD disc image file (iso)

Storage: local

Type: Linux

ISO image: 5.04-desktop-amd64.iso

Version: 7.x - 2.6 Kernel

☐ Use physical CD/DVD Drive

☐ Do not use any media

Advanced

Back

Next

General, OS and System

Name the VM and select the node it starts on. Under OS select the uploaded ISO and the matching guest type. Under System tick QEMU Guest Agent - it is what allows Proxmox VE to quiesce the guest filesystem before a snapshot, which turns a crash-consistent snapshot into an application-consistent one (9.1). A Windows guest also needs its machine type and firmware set here.

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

Graphic card: Default

SCSI Controller: VirtIO SCSI single

Machine: Default (i440fx)

Qemu Agent: ☒

Firmware

BIOS: Default (SeaBIOS)

Add TPM: ☐

Help

Advanced

Back

Next

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

Key ↑	Value
agent	1
cores	4
cpu	x86-64-v2-AES
ide2	local:iso/ubuntu-25.04-desktop-amd64.iso,media=cdrom
memory	4096
name	ubuntu-nfs-demo
net0	virtio,bridge=vbr0,firewall=1
nodename	pve1
numa	0
ostype	l26
scsi0	nfs-Pool-0:32,iothread=on
scsilhw	virtio-scsi-single
sockets	1
vmid	100

☐ Start after created

Advanced

Back

Finish

CPU, Memory, Network and Confirm

Sockets, cores and memory follow the workload; none of it is storage-specific. The network device attaches to the bridge used by guests - normally not the NFS data bridge, so that guest traffic and storage traffic stay apart.

The summary is reviewed before Finish. The value that matters is scsi0: the disk is on the JovianDSS NFS storage. It carries no format= entry, because raw is the only format the storage offers and Proxmox VE omits a format that equals the storage default (8.3).

8.3 Create the Guest VM: Disks

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: nfs-Pool-0 IO thread: ☒

Disk size (GiB): 32

Format: Raw disk image (raw)

+ Add

Import

Help Advanced ☐ Back Next

Select the JovianDSS NFS storage

Unlike the storage appliance in chapter 5, a guest belongs on the shared NFS storage - that is what makes live migration and HA failover possible.

Format is fixed at Raw disk image

The field is greyed out and set to raw. From plugin v1.1.0-pre1 the storage declares raw as its only supported format, so Proxmox VE offers no alternative. Earlier versions preselected qcow2 here and a wizard run straight through produced a qcow2 file, which removes the reason for using this plugin: Proxmox-native snapshots and rollback on .raw disks, which keep the performance advantage.

The command line is refused as well - the allocation fails

```
root@pve1:~# pvesm alloc nfs-Pool-0 999 \
                vm-999-disk-0.qcow2 1G --format qcow2
unsupported format 'qcow2' -
storage 'nfs-Pool-0' only supports raw
```

The remaining defaults are suitable: IO thread stays on, and cache can remain at the default - the demo VMs in this guide run with directsync on NFS without issues.

8.4 Windows and Linux Guests

Virtual Machine 105 (win2025-nfs) on node 'pve2'No Tags

Start

Shutdown

Migrate

Console

More

Help

Summary

Console

Hardware

Cloud-init

Options

Task History

Monitor

Backup

Replication

Snapshots

Firewall

Permissions

AddRemoveEditDisk ActionRevert

Memory4.00 GiB

Processors4 (1 sockets, 4 cores) [x86-64-v2-AES]

BIOSOVMF (UEFI)

DisplayDefault

Machinepc-q35-11.0

SCSI ControllerVirtIO SCSI single

CD/DVD Drive (ide0)none.media=cdrom

CD/DVD Drive (ide2)none.media=cdrom

Hard Disk (scsi0)nfs-Pool-1:105/vm-105-disk-1.raw,aio=native,cache=directsync,iothread=1,size=32G

Network Device (net0)virtio=BC:24:11:70:84:1A,bridge=vbr0,firewall=1,link_down=1

EFI Disknfs-Pool-1:105/vm-105-disk-0.raw,efitype=4m,ms-cert=2023k,pre-enrolled-keys=1,size=528K

TPM Statenfs-Pool-1:105/vm-105-disk-2.raw,size=4M,version=v2.0

Virtual Machine 104 (ubuntu-nfs) on node 'pve1'

No Tags

▶ Start

⏻ Shutdown

↗ Migrate

>_ Console

More

⚙ Help

Summary

>_ Console

Hardware

Cloud-init

Options

Task History

Monitor

Backup

Replication

Snapshots

Firewall

Permissions

Add

Remove

Edit

Disk Action

Revert

Memory

4.00 GiB

Processors

4 (1 sockets, 4 cores) [x86-64-v2-AES]

BIOS

Default (SeaBIOS)

Display

Default

Machine

Default (i440fx)

SCSI Controller

VirtIO SCSI single

CD/DVD Drive (ide2)

none.media=cdrom

Hard Disk (scsi0)

nfs-Pool-0:104/vm-104-disk-0.raw,aio=native,cache=directsync,iothread=1,size=32G

Network Device (net0)

virtio=BC:24:11:F3:B8:CC,bridge=vbr0,firewall=1,link_down=1

Microsoft Windows

A Windows guest brings two extra objects, both on the NFS storage beside the disk: an EFI disk and a TPM state volume, 528 KiB and 4 MiB. They carry the Secure Boot keys and the TPM identity of the machine and are copied together with it. A full clone of such a VM remains functional - the copied TPM state boots correctly, verified in this lab.

Linux

The simple case: one .raw disk on the NFS storage, VirtIO SCSI single as the controller, and qemu-guest-agent installed inside the guest. Without the agent Proxmox VE cannot freeze the filesystem and snapshots drop from application-consistent to merely crash-consistent.

8.5 Verify: the Disks Are Files on NFS

Storage 'nfs-Pool-0' on node 'pve1'

Remove

Search: Name, Format

Name	Date	Format	Size
vm-104-disk-0.raw	2026-08-21 10:54:47	raw	32.00 GiB

Summary

VM Disks

CT Volumes

Permissions

This is the point at which the design becomes visible: every VM disk is an ordinary file in a directory on the dataset, listed here by the VM that owns it. That is why a dataset snapshot captures whole machines at once, why replication moves them as a unit, and why configuration files placed beside them by pve-config-backup are protected by the same snapshot (9.8).



Chapter 9

Snapshots and Data Protection

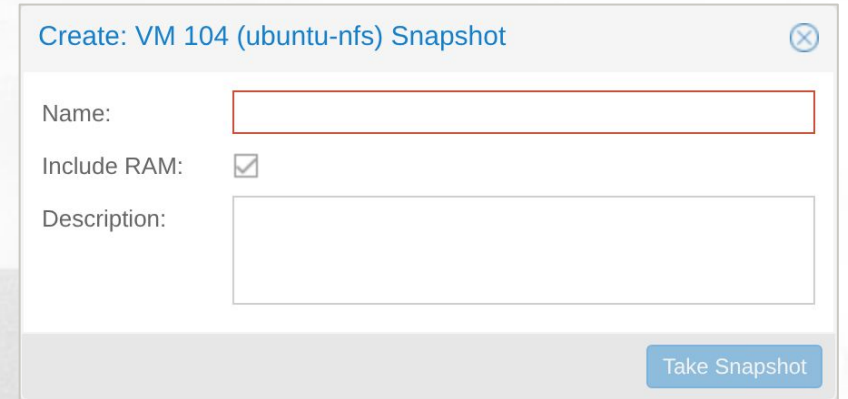
How snapshots work, rollback, clones and templates, OODP and the DR instance

9.1 How Snapshots Work

A snapshot belongs to the dataset, not to one VM

VM -> Snapshots -> Take Snapshot creates a ZFS snapshot of the dataset on Open-E JovianDSS. Because every VM disk on that dataset is a file inside it, the snapshot captures all of them at the same instant - along with any configuration files that pve-config-backup has placed beside them (9.8).

With the guest agent installed the result is application-consistent.



Create: VM 104 (ubuntu-nfs) Snapshot

Name:

Include RAM: ☒

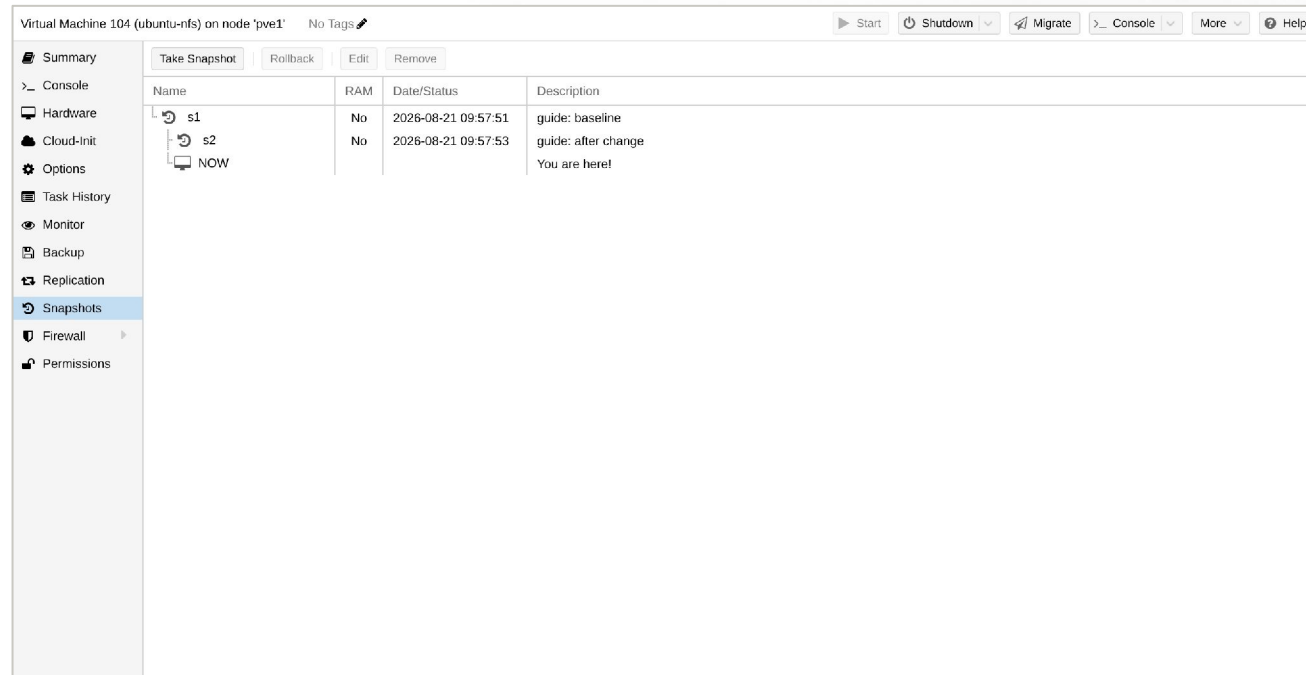
Description:

Take Snapshot

```
root@pve1:~# qm snapshot 104 s1 --description 'baseline'
freeze guest filesystem
snapshotting 'drive-scsi0' (nfs-Pool-0:104/vm-104-disk-0.raw)
thaw guest filesystem
real    0m1.228s
```

The freeze/thaw pair comes from qemu-guest-agent: the guest flushes and quiesces its filesystem before the snapshot is taken. Without the agent the snapshot still works, but it is only crash-consistent - equivalent to an abrupt power loss.

9.2 Rollback



Newer snapshots survive - the tree branches

A rollback to s1 does not destroy s2. NOW becomes a sibling of s2 under s1 and both lines stay reachable, so a rollback is reversible.

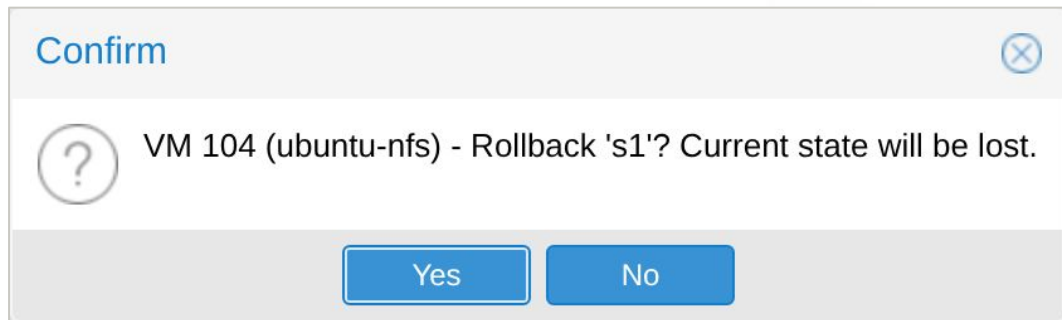
The maintenance window follows the disk size, not the change

Rollback copies the whole disk file. In this lab a 32 GiB disk took 36 seconds at about 964 MB/s, and the duration is the same whether one byte changed or all of them. The VM is stopped for the duration.

9.3 Rollback and High Availability

A VM under HA control cannot be rolled back directly

The plugin refuses, rather than allowing HA to restart or move the resource during the operation. Nothing is locked and the guest continues to run.



```
root@pve1:~# qm rollback 104 s1
Rollback blocked: vm:104 is controlled by High Availability (state: started).
Disable HA management before retrying:
  Web UI: Datacenter -> HA -> Resources -> set state to ignored
  CLI:    ha-manager set vm:104 --state ignored
```

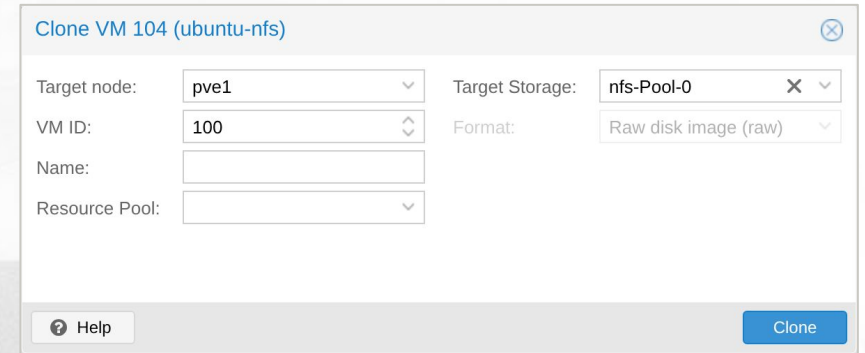
The documented procedure

ha-manager set vm:104 --state ignored, roll back, then ha-manager set vm:104 --state started. The resource returns through queued and starting to started. The last step is essential - a VM left ignored will not fail over.

9.4 Clones and Templates

Full clones work, including from a running VM and from a template

VM -> More -> Clone, or `qm clone`. Cloning a running guest quiesces it through the agent first, exactly as a snapshot does. Selecting a JovianDSS NFS storage as the target leaves Format greyed out on Raw disk image - the clone gets the only format the storage supports (8.3).



Clone VM 104 (ubuntu-nfs)

Target node: Target Storage:

VM ID: Format:

Name:

Resource Pool:

```
root@pve1:~# qm clone 104 900 --full 1 --storage nfs-Pool-0
issuing guest agent 'guest-fsfreeze-thaw' command
real    0m38.333s          # 32 GiB - a file copy, like rollback

root@pve1:~# qm clone 900 901          # linked clone from a template
Linked clone feature is not supported for 'nfs-Pool-0:900/vm-900-disk-0.raw' (scsi0)
```

Linked clones are not available on this storage

`qm template` works and full clones from a template work, but a linked clone is refused. Every clone therefore costs the full size of the disk and the time to copy it, so a fleet built from one golden image consumes full capacity instead of sharing blocks. VDI-style deployments must size the pool accordingly.

9.5 OODP and the Backup and DR Instance

StatusDisk GroupsiSCSI TargetsFC TargetsSharesSnapshotsVirtual IPsConfiguration

Virtual IPs

Search

+ Add virtual IP

Virtual IP	Name	Netmask	Network interface	State
No virtual IPs found.				

Virtual IPs routing

Search

+ Add static routing

Network/Host IP	Netmask	Gateway	Virtual IP name
No items found.			

On- and Off-site Data Protection

OODP is the replication built into Open-E JovianDSS: it takes snapshots of a dataset on a schedule and sends them to a second Open-E JovianDSS system, keeping its own retention plan at each end. Intervals go down to one minute, which is what makes a near-zero RPO possible. It moves snapshot deltas rather than files, so a short interval costs little even with thousands of VMs on the dataset.

The unit of replication is the dataset - which is exactly the unit that holds whole virtual machines: their disks, and with pve-config-backup installed, their configuration files as well.

A third Open-E JovianDSS, on the quorum node

The Backup and DR instance is another Open-E JovianDSS VM running on the third node with its own pool. It receives OODP replicas and, in a disaster, serves them so that the VMs can be restarted there. Sizing follows the intended role: capacity alone for a repository; CPU and memory as well if it must run the workload after both main servers are lost. Its OODP interface should be at least 10 GbE, like the production VIP interfaces.

OODP and Proxmox Backup Server are complementary: OODP gives a short RPO and a fast restart, PBS gives deduplicated long-retention backups with per-file restore.

9.6 OODP Setup: Destination Dataset and Backup Task

The screenshot shows the 'Backup task wizard' window with the 'Source configuration' step selected. The left sidebar lists five steps: 1. Source configuration, 2. Destination configuration, 3. vCenter / vSphere server integration, 4. Task properties, and 5. Summary. The main area is titled 'Source configuration' and contains the following fields:

- Source:** A text field labeled 'Resource path:' containing 'Pool-0/nfs-Pool-0' and a 'Browse...' button.
- Retention-Interval plan for source:** A section with a note: 'Snapshots from source will be sent every 5 minutes (i.e. the shortest interval from the source retention rules)'. Below this are three retention rules, each with a table of values and a 'should be kept for' field.

Retention rules:			
A snapshot taken every	5	minute(s)	should be kept for
	↓	↓	1
	↑	↑	hour(s)
	↑	↑	↓
A snapshot taken every	15	minute(s)	should be kept for
	↓	↓	3
	↑	↑	day(s)
	↑	↑	↓
A snapshot taken every	1	hour(s)	should be kept for
	↓	↓	2
	↑	↑	week(s)
	↑	↑	↓

At the bottom of the retention rules section, there is a '+ Add another rule' button. At the bottom right of the main area, there is a 'Restore to default' button with a circular arrow icon. At the very bottom of the window, there are 'Cancel' and 'Next >' buttons.

Step 1 - on the Backup and DR system

Storage Management -> pool -> Shares -> Add dataset creates the dataset that will receive the snapshots. It is then marked Options -> Backup task -> Set as backup destination; without that flag the wizard on the production side shows it as unavailable.

No network share is added to it. The repository is overwritten at every interval and is not meant to be browsed. Where direct access to a replica is genuinely needed, the share is created read-only. A name carrying the origin keeps replicas from several sources distinguishable.

Step 2 - on the production system

Shares tab, the dataset holding the VMs, Options -> Add to backup task. That is what enrolls the dataset into OODP. The direction is always source-driven: production pushes to the DR system.

OODP is implemented on top of znapzend, and the entire task definition is stored in ZFS user properties on the source dataset - schedule, retention plans and destination. The task therefore remains attached to the dataset, and it is managed from the product GUI because the REST API does not expose backup tasks.

9.7 OODP Setup: Retention Plans, Apply and Verify

The screenshot shows the 'Backup task wizard' window with the 'Summary' tab selected. The left sidebar lists five steps: 1. Source configuration, 2. Destination configuration, 3. vCenter / vSphere server integration, 4. Task properties, and 5. Summary. The main content area is divided into two sections: 'Source' and 'Destination 1'. The 'Source' section shows 'Resource path: Pool-0/nfs-Pool-0' and a retention plan with three rules: 'A snapshot taken every 5 minutes should be kept for 1 hour', 'A snapshot taken every 15 minutes should be kept for 3 days', and 'A snapshot taken every 1 hour should be kept for 2 weeks'. The 'Destination 1' section shows 'Destination path: 10.15.0.163:40000/Pool-DR/nfs-Pool-0-dr' and a retention plan with four rules: 'A snapshot taken every 5 minutes should be kept for 12 hours', 'A snapshot taken every 15 minutes should be kept for 1 week', 'A snapshot taken every 1 hour should be kept for 3 weeks', and 'A snapshot taken every 12 hours should be kept for 3 months'. At the bottom are 'Cancel', 'Back', and 'Apply' buttons.

The screenshot shows the 'Backup task wizard' window with the 'Destination configuration' tab selected. The left sidebar is the same as the previous screen. The main content area has a 'Destination configuration' section with a checkbox 'Set remote destination for backup task' and a '+ Add another destination' button. Below this is a section for 'Destination 1' with a 'Remove resource' button. The 'Destination server and resource path' section contains a 'Destination server' dropdown set to '10.15.0.163:40000' and a 'Resource path' field set to 'Pool-DR/nfs-Pool-0-dr' with a 'Browse...' button. The 'Retention-Interval plan for destination' section has a text box stating 'Snapshots from source will be sent every 5 minutes (i.e. the shortest interval from the source retention rules)' and a 'Retention rules:' section with a table of rules. At the bottom are 'Cancel', 'Back', and 'Next' buttons.

Two independent plans

Production keeps a short, dense history for quick recovery; the DR system keeps a longer one for cases that reach further back. The shortest source interval is the RPO - the amount of work that is accepted as lost.

The destination pool is sized before the task is enabled: the replica needs room for the dataset plus every snapshot the destination plan retains. In this lab a 16 GiB dataset produced a 20 GiB replica and then added roughly 5 GiB a day of deltas from an idle guest.

Apply, then verify twice

First in the dataset list: the source dataset shows two markers, A for auto-snapshots and B for the backup task. An A without a B means snapshots are taken but nothing leaves production.

Then at the destination: snapshots must arrive and their timestamps must advance. A task that was accepted but cannot reach its destination must be detected on day one rather than during a recovery. In this lab the first full replication of a 16 GiB dataset took 189 seconds.

9.8 Backing Up the VM and CT Configuration Files

By default they are NOT stored with the disks

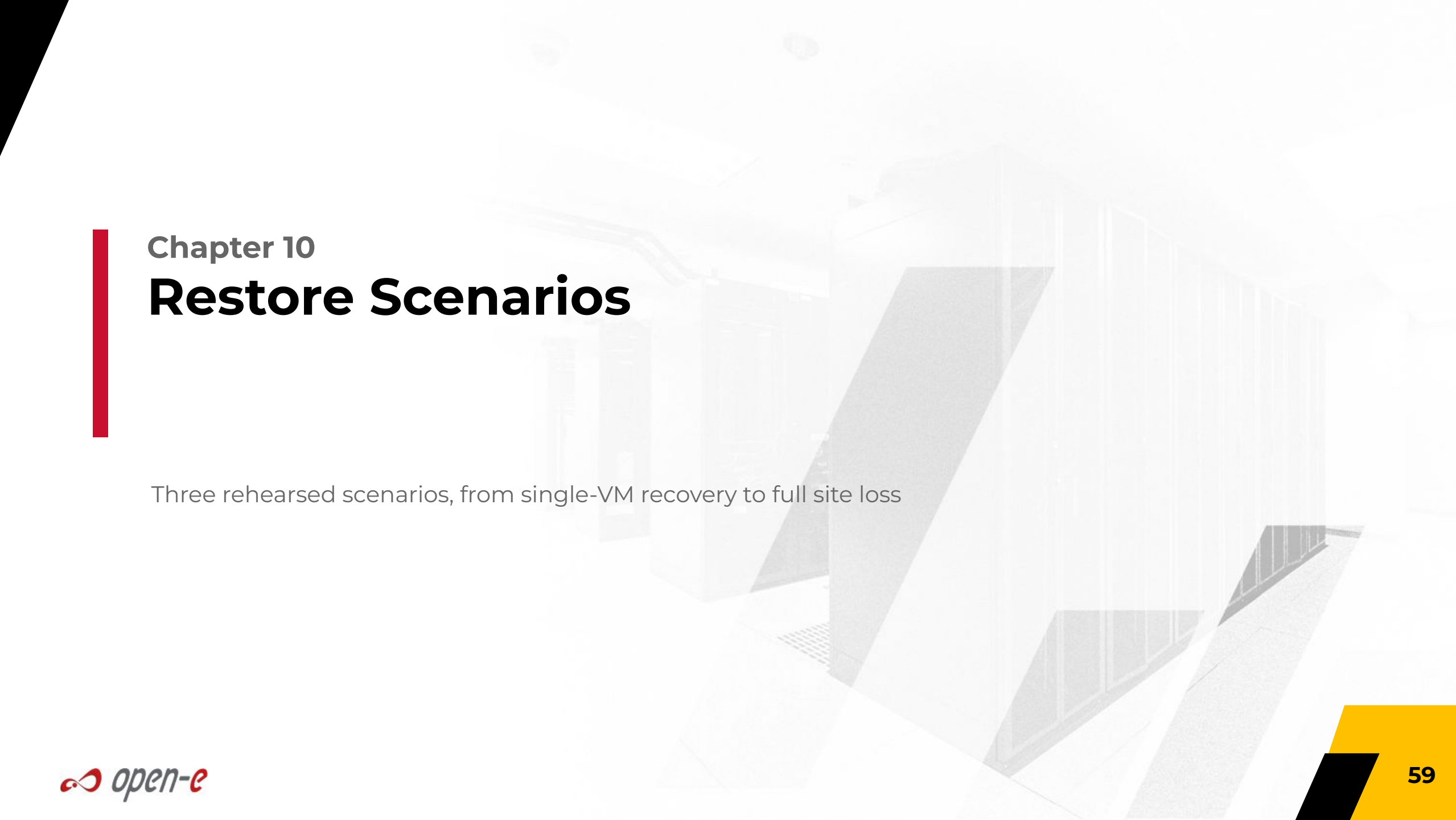
Proxmox VE keeps a VM definition in `/etc/pve`, which is not on the dataset. A snapshot therefore captures the disks and nothing else - verified in this lab: a restored clone contained `images/<vmid>/<disk>.raw` and no `.conf` file at all.

The pve-config-backup service closes that gap

Open-E publishes a small service - github.com/open-e/pve-config-backup - that runs under `systemd` and replicates VM and CT configuration files into the folders that contain their disks. With it installed, the configuration is inside the snapshot and a restore is a copy and an edit. Without it, the hardware profile has to be rebuilt by hand (chapter 10.3).

Why this matters for recovery

A disk restored without its definition means reconstructing the machine manually - controller, firmware, machine type, network model - and a single wrong value produces a guest that will not boot. The service is installed before it is needed; it becomes relevant only during a restore.



Chapter 10

Restore Scenarios

Three rehearsed scenarios, from single-VM recovery to full site loss

10.1 Restore Scenarios: Overview

Every restore starts the same way: clone a snapshot on the DR system

A clone is created instantly and consumes almost no space until it is written to, so one clone per restore is the safe pattern. A name carrying the moment it was taken - `cl_yymmdd_hhmm` - keeps several clones distinguishable during an incident.

Scenario 1 - one virtual machine is lost or corrupted

Clone, share, attach the clone to Proxmox VE as storage, rebuild the VM around the restored disk. The production dataset is never touched, so the attempt can be made while the original still runs.

Scenario 2 - one file inside a guest

Nothing is booted. The disk image is read directly on the Proxmox host and the file copied out - faster, and it cannot disturb the running service.

Scenario 3 - the whole site

The clone is the entire dataset, so one clone brings back every VM stored on it. There is no per-VM restore step: the clone is registered once and each machine it contains is rebuilt from it.

10.2 Scenario 1: Clone the Snapshot and Share It

Clone "autosnap_2026-08-21-124000"

Encryption settings

Encryption requires a passphrase in Pool → Configuration → Resource encryption.

☐ Encrypt resource

Dataset properties

Name:

cl_260821_1300

All values below are inherited from the current dataset settings.

Deduplication:

Disabled (default)

Number of the data copies:

1 (default)

Compression:

None

Record size:

128 KiB (default)

Write cache sync requests:

Always

Write cache sync request handling (logbias):

Write log device (Latency)

Read cache (primary, ARC) scope:

All (default)

Read cache (secondary, L2ARC) scope:

All (default)

Access time:

Enabled

Small data blocks policy

Available only when a special devices group exists.

Small data block size:

Disable for the dataset

☐ Enable quota

☐ Enable reservation

Cancel

Apply

Step 1 - clone

On the Backup and DR system: Storage Management -> pool -> Snapshots. Each row carries an action icon with Delete, Snapshot details and Clone.

The snapshot for the required point in time is cloned under a name identifying that moment. The clone appears in the Shares tab beside the ordinary datasets and carries its origin snapshot with it, so its provenance stays visible.

Cloning does not consume the snapshot and does not disturb replication. A clone taken from the wrong point in time is deleted and another one made.

Step 2 - share

In the Shares tab, the Options menu of the clone offers Add share. The share is named to match the clone, and the NFS protocol must be enabled on it - the clone is not accessible to Proxmox VE until it is exported.

Restore shares are temporary by nature, and matching names make it clear later which share belongs to which clone.

Access control applies here as it does in production (6.5): the share is restricted to the Proxmox nodes performing the restore. A restore share holds the same data as production and requires the same protection.

10.3 Scenario 1: Attach the Clone and Rebuild the VM

Confirm the export, add a storage entry pointing at the DR system, then rebuild the machine around the restored disk:

```
root@pve1:~# showmount -e 10.15.0.163
/c1_260821_1300 *                # the ALIAS - not what export needs

joviandss-nfs: nfs-restore
  export /Pools/Pool-DR/c1_260821_1300    # FULL path
  path /mnt/pve/nfs-restore
  server 10.15.0.163
  control_addresses 10.15.0.163
  content images,rootdir
  shared 1

root@pve1:~# ls /mnt/pve/nfs-restore/images/104/
vm-104-disk-0.raw                # the disk - and no .conf beside it

root@pve1:~# qm create 110 --name restored-from-dr --memory 4096 --cores 4 \
               --ostype l26 --scsihw virtio-scsi-single --agent 1
root@pve1:~# qm set 110 --scsi0 nfs-restore:110/vm-110-disk-0.raw
root@pve1:~# qm set 110 --boot order=scsi0
```

The restore storage needs its REST password too (7.2) - one without it is refused at creation. The VM definition is present only if pve-config-backup was installed before the snapshot was taken (9.8); otherwise the hardware profile is recreated by hand, matching the controller, firmware and machine type of the original - a mismatch there produces a guest that will not boot. The restore uses a NEW VMID while the original still runs, so a restore from the wrong point in time destroys nothing.

10.4 Scenario 1: Verify the Restore

A started VM is not a verified restore. The data has to be proven:

```
root@pve1:~# qm start 110
root@pve1:~# qm agent 110 get-osinfo | grep pretty-name
    "pretty-name" : "Ubuntu 25.04"

root@pve1:~# qm guest exec 110 -- /bin/cat /root/marker.txt
BEFORE-S1
2026-08-21T09:57:41+02:00          # matches production exactly
```

What this proves

The entire chain functioned as intended: production dataset, OODP replica on the DR system, snapshot, clone, NFS share, Proxmox storage, running guest. In this lab the restored VM answered its guest agent 20 seconds after start.

The check uses content that can be predicted - a marker file, a log with a known last entry, a database row. A successful boot is not verification; matching content is.

10.5 Scenario 2: Restore a Single File

The disk image is read on the host - no VM, no boot, no risk to the running service:

```
# Linux guest - ext4
root@pve1:~# LOOP=$(losetup -f -P -r --show /mnt/pve/nfs-restore/images/110/vm-110-disk-0.raw)
root@pve1:~# mount -o ro ${LOOP}p2 /mnt/restore-probe
mount: cannot mount /dev/loop0p2 read-only.      # dirty journal - expected
root@pve1:~# mount -o ro,noload ${LOOP}p2 /mnt/restore-probe      # works

# Windows guest - NTFS, four partitions; the largest holds the system
root@pve2:~# lsblk -no NAME,SIZE $LOOP
loop0      32G
|-loop0p1  100M   EFI system partition (vfat)
|-loop0p2   16M   Microsoft reserved
|-loop0p3  31.2G   Windows (ntfs)    <- the target
`-loop0p4  682M   recovery (ntfs)
root@pve2:~# mount -t ntfs3 -o ro ${LOOP}p3 /mnt/win-probe

root@pve1:~# cp /mnt/restore-probe/path/to/file /root/ && umount /mnt/restore-probe && losetup -D
```

Two details decide whether the mount succeeds

A guest stopped abruptly - or any crash-consistent snapshot - leaves a dirty journal, and the kernel refuses a plain read-only mount because replaying the journal requires writing; noload skips the replay. The -r flag on losetup guarantees the image itself is never modified. A cloned Windows disk must not be attached to the running guest as a shortcut: it carries the same GPT disk GUID as the original, so Windows keeps it offline or requires a new disk signature, which writes to the image.

10.6 Scenario 3: Restore Every VM at Once

One clone is the whole dataset - and the whole dataset is every machine on it

There is no separate bulk-restore procedure. The clone is registered once, and what came back is then visible in one listing:

```
root@pve1:~# ls /mnt/pve/nfs-restore/images/  
104  993                                # every VM stored on the dataset
```

Step 4 of scenario 1 then repeats for each machine

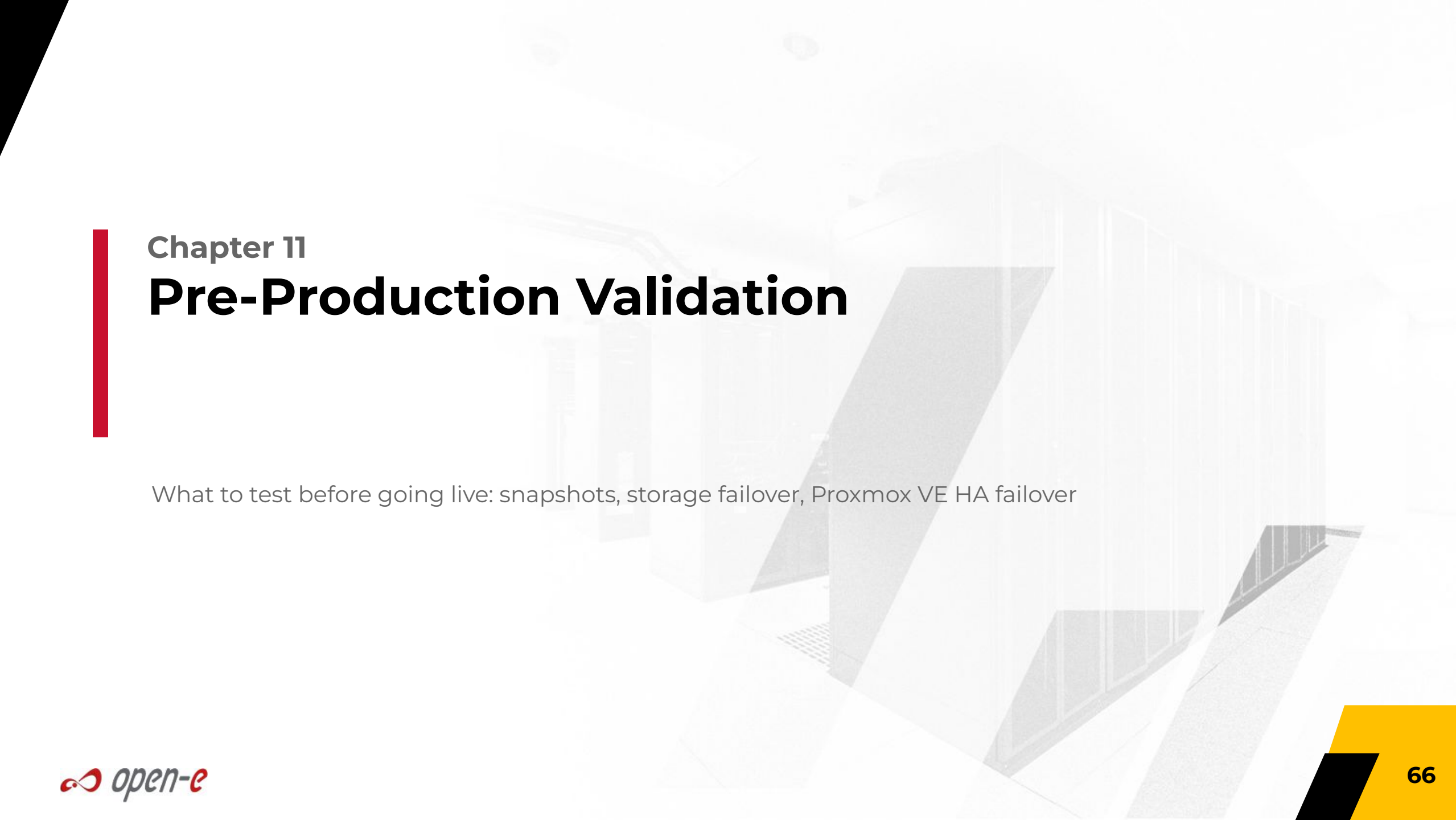
The work is per machine, but the data recovery already happened at the clone. The effect of pve-config-backup is largest here: with the definitions inside the snapshot, rebuilding twenty VMs is twenty copies and edits; without it, twenty hardware profiles have to be reconstructed.

Restore order is defined in advance, not during the incident

Which machines come back first - domain controllers, databases and anything the rest depends on - is decided in advance and written down. A recovery run in the wrong order still completes, but it delays service restoration at the point where downtime is most expensive.

Capacity on the DR system

If the third node has to run the workload rather than merely hold it, it needs the CPU and memory for the machines intended to start there (9.5). A DR node sized only as a repository will restore the data and then fail to run it.



Chapter 11

Pre-Production Validation

What to test before going live: snapshots, storage failover, Proxmox VE HA failover

11.1 Validate: Snapshots and Rollback

On a disposable VM, before production data is placed on the storage

The purpose is not to prove that the feature exists but to measure what it costs on this hardware, so that the numbers in the runbook are local ones rather than the numbers in this guide.

1. `qm snapshot <vmid> t1` # write a marker file in the guest first
2. change data in the guest, then: `qm snapshot <vmid> t2`
3. `time qm rollback <vmid> t1` # RECORD THIS NUMBER
4. `qm listsnapshot <vmid>` # t2 must still be there
5. start the VM, confirm the marker is back to its pre-t1 content

What to conclude

Step 3 gives the maintenance window for a disk of that size; it scales with disk size, not with the amount of changed data. Step 4 confirms the branch behaviour, so that a rollback is not avoided in the belief that it destroys newer snapshots. A surprise in either step is a reason to stop and find the cause before going to production.

11.2 Validate: Storage Failover

Zpool name	Active on node	Zpool storage capacity	Available space	Status
Pool-0	JovianDSS-nodeA (IP: 10.15.0.142)	3.63 TiB (154.65 GiB used)	3.47 TiB	Started
Pool-1	JovianDSS-nodeB (IP: 10.15.0.152)	3.63 TiB (150.32 GiB used)	3.48 TiB	Started

Move a pool while a VM is running from it

This is the test that proves the storage cluster, the VIPs and the NFS mount cooperate. Failover -> Failover resources shows which node owns each pool and offers the move. A continuous write runs inside the guest throughout.

```
# in the guest, before the move:
while true; do date >> /root/hb.txt; sync; sleep 1; done

# on a Proxmox node, watching the path follow the pool:
watch -n1 "ss -tn state established '( dport = :2049 )' | grep -c <data-vip>"
```

Expected behaviour - and the values measured here

The guest pauses on I/O and then continues: a hard NFS mount blocks rather than failing. In this lab the gap was 13.8 seconds, the data VIP followed the pool to the surviving node and all four nconnect links were re-established. What must NOT happen is the guest filesystem going read-only or the VM crashing.

11.3 Validate: Proxmox VE HA Failover

Fence a node under realistic conditions

Powering a node off from the GUI is a clean shutdown and tests the shutdown policy, not fencing. Fencing is tested by removing the node abruptly - disconnecting its network, or using the power button on the chassis - and allowing the watchdog to act.

```
# before: note where the HA guests run
ha-manager status

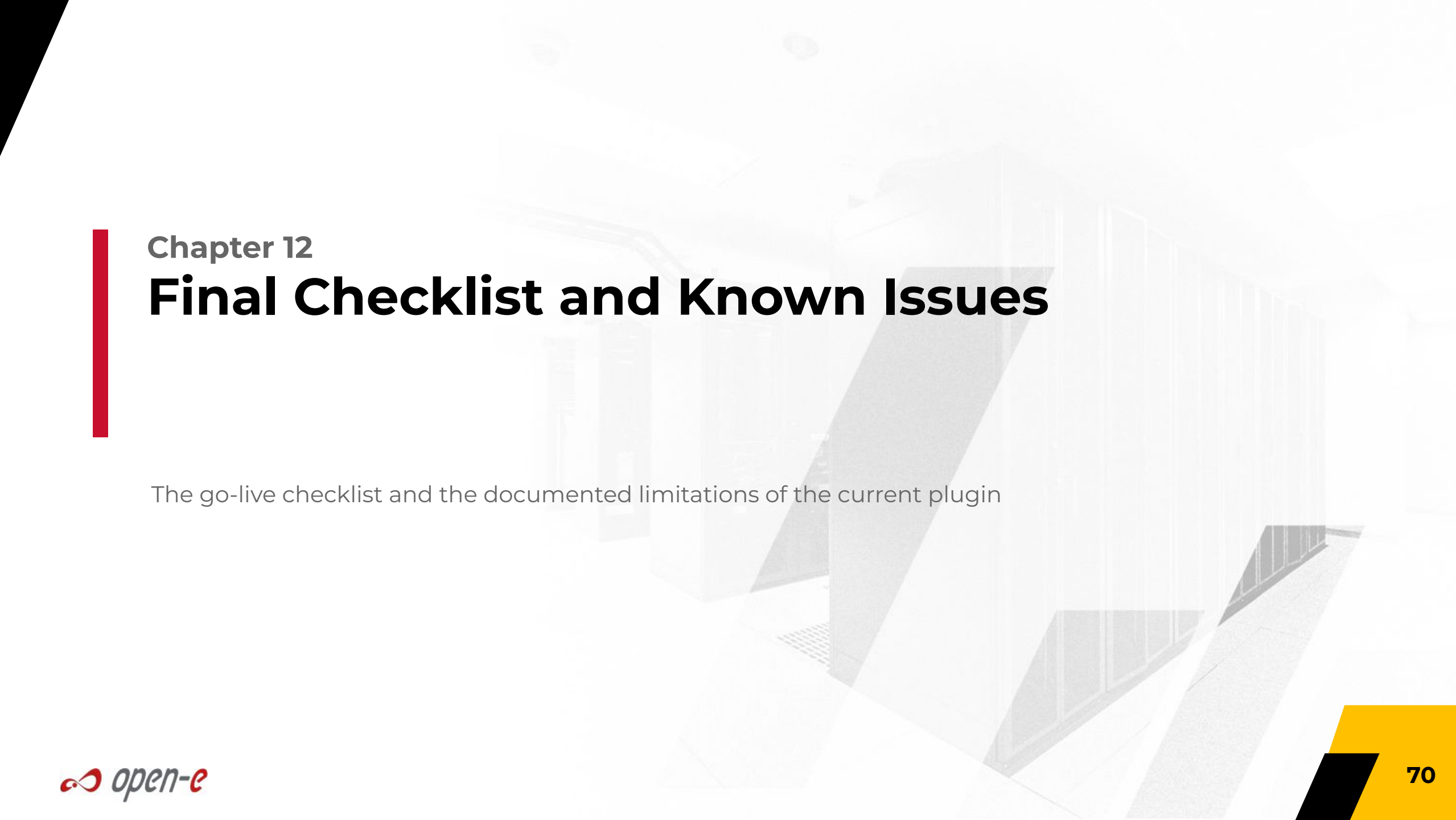
# after fencing the node, on a surviving node:
watch ha-manager status      # the guests should appear here, started
```

What to check afterwards

Guests must restart on a permitted node only - one landing on the quorum node means the rule is not strict. The storage must be mounted on the new host before the node is rejoined.

Relocation is fast; fencing dominates the recovery time

Measured with node-maintenance, which relocates without fencing: the guest moved in ~50 s and lost 1.01 s of I/O, against 13.8 s for a storage failover (11.2). The unplanned number includes the watchdog timeout and a cold boot and has to be measured separately, with physical access to the rack.



Chapter 12

Final Checklist and Known Issues

The go-live checklist and the documented limitations of the current plugin

12.1 Final Checklist

Storage side

- REST API enabled on every instance
- NFS protocol enabled (off by default)
- Both pool VIPs Active, paired with a remote interface
- Dataset sync set to Always
- Share active, IP ACLs list every Proxmox node
- Ping nodes point OUTSIDE the storage cluster
- OODP task shows both the A and B markers
- DR pool sized for the data plus its retention

Proxmox VE side

- Same plugin version on every node
- Storage active cluster-wide, shared 1
- REST password set on every joviandss-nfs storage
- nconnect confirmed on EVERY node
- Plugin v1.1.0-pre1 or newer, so raw is enforced
- HA resources added and a strict node-affinity rule set
- Watchdog and shutdown policy configured
- Migration network on its own subnet
- Rollback time measured on local hardware
- Capacity planned for data that is never reclaimed

12.2 Known Issues and Limitations

The storage cannot be added or edited from the Proxmox VE GUI

Datacenter -> Storage -> Add does not list the jovianDSS-nfs type (7.2), and selecting an existing entry and pressing Edit raises an internal error and opens nothing, because the plugin registers no GUI editor for its type. `pvesm add` / `pvesm set` and `/etc/pve/storage.cfg` work and take effect immediately.

Linked clones are not supported

qm template and full clones work; a linked clone is refused. Every clone therefore costs the full size of the disk in space and time. Pools are sized for full copies, especially for VDI-style fleets.

Rollback is a full file copy

Its duration scales with the size of the disk, not with the amount of changed data, and the VM is stopped throughout. It is not the near-instant operation a ZFS rollback would be. Newer snapshots do survive it.

Rollback is refused while HA controls the VM

By design, to stop HA moving the resource mid-operation. The resource is set to ignored, rolled back, then set to started again - the last step is essential.

Mount options do not reach an already-mounted storage

Adding or changing options in `storage.cfg` leaves a live mount untouched, with no warning. What uses it is stopped, the path unmounted, and Proxmox VE remounts it. Verification is per node - one node missing `nconnect` is easily overlooked.

A cancelled snapshot operation can leave the VM locked

Proxmox VE may end plugin cleanup before it finishes, leaving a partial snapshot in the prepare state and snapshot resources present on Open-E JovianDSS but not registered in Proxmox VE. Recovery is `qm unlock <vmid>` followed by `qm delsnapshot <vmid> <snapname>` `--force`, which also removes the `vmstate` disk.

12.3 Known Issues and Limitations (continued)

Deleted guest data is never reclaimed

NFSv3 has no deallocation primitive, so space freed inside a guest never returns to the pool. `discard=on`, `fstrim` and `Optimize-Volume` all report success and change nothing - the disk file, `df` and `zfs` used stay as they were. The host cannot do it either: `fstrim` on the mount is refused. Only deleting the whole disk releases the blocks. Measured identically on Proxmox VE and on VMware ESXi against the same dataset, so this is a property of the protocol, not of the plugin. Pools are therefore sized for peak usage, not for the usage a guest currently reports.

Snapshots persist after the VM is destroyed

`qm destroy` removes the `.raw` disks and the `vmstate` files but leaves the dataset snapshots behind; they remain as orphans, holding their blocks, until they are deleted on Open-E JovianDSS. The plugin release notes list this as a known issue rather than an artefact of this lab, so a periodic cleanup belongs in the operating routine - and in the sizing of the DR pool.

Copyright

© Open-E, Inc. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form, by any means, without the prior written consent of the publisher.

Open-E and Open-E JovianDSS are trademarks or registered trademarks of Open-E, Inc. Proxmox and Proxmox VE are trademarks of Proxmox Server Solutions GmbH. All other trademarks are the property of their respective owners.

Plugin sources and latest documentation: <https://github.com/open-e/JovianDSS-Proxmox>