

Webinar

Proxmox VE from zero

all your microphones are muted

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

use Chat for discussion, networking or applause

1

Design and architecture

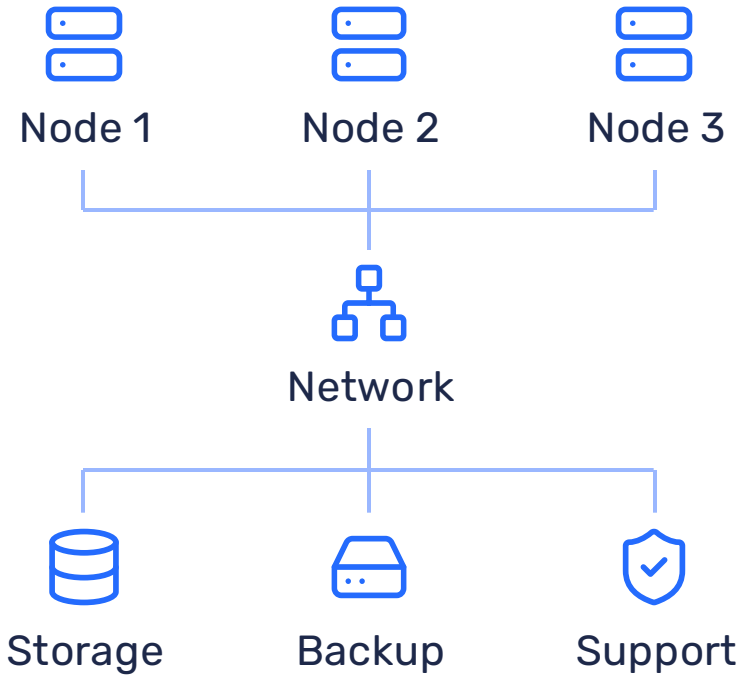


initMAX

Plan compute, storage, networking and recovery

- Cluster or standalone node
- Storage architecture
- Network architecture
- Combined architecture
- Backup architecture
- Support for systems

Architecture at a glance

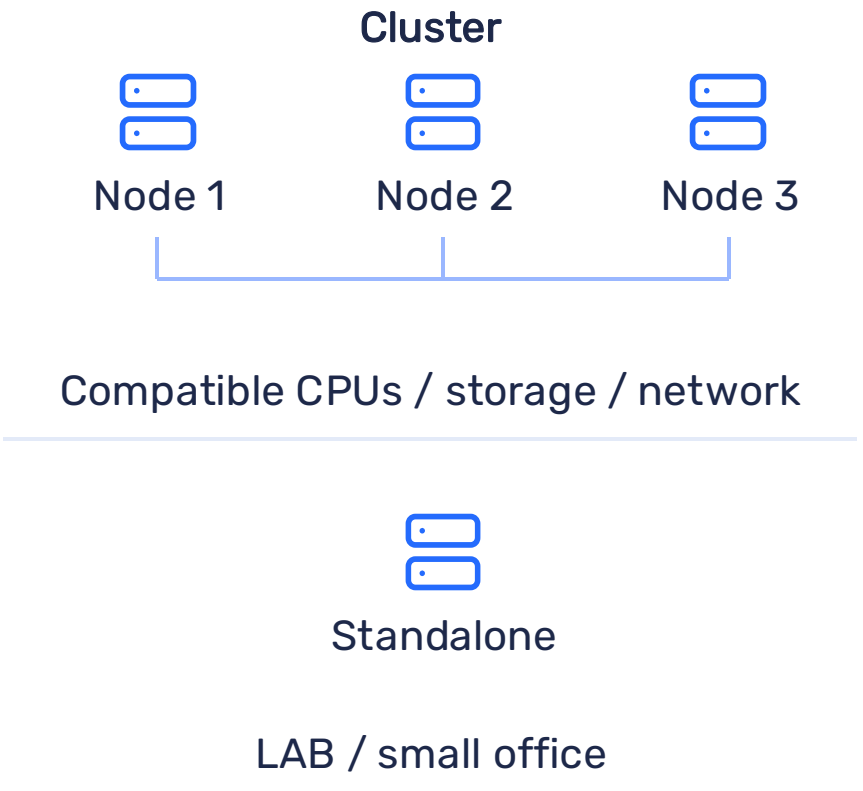


initMAX

Choose standalone or cluster to match availability

- **Cluster design:**
  - Multiple servers
  - Compatible CPUs and consistent storage/network
- **Standalone design:**
  - Used for LAB or small offices
  - Recommended HW support and DR solution

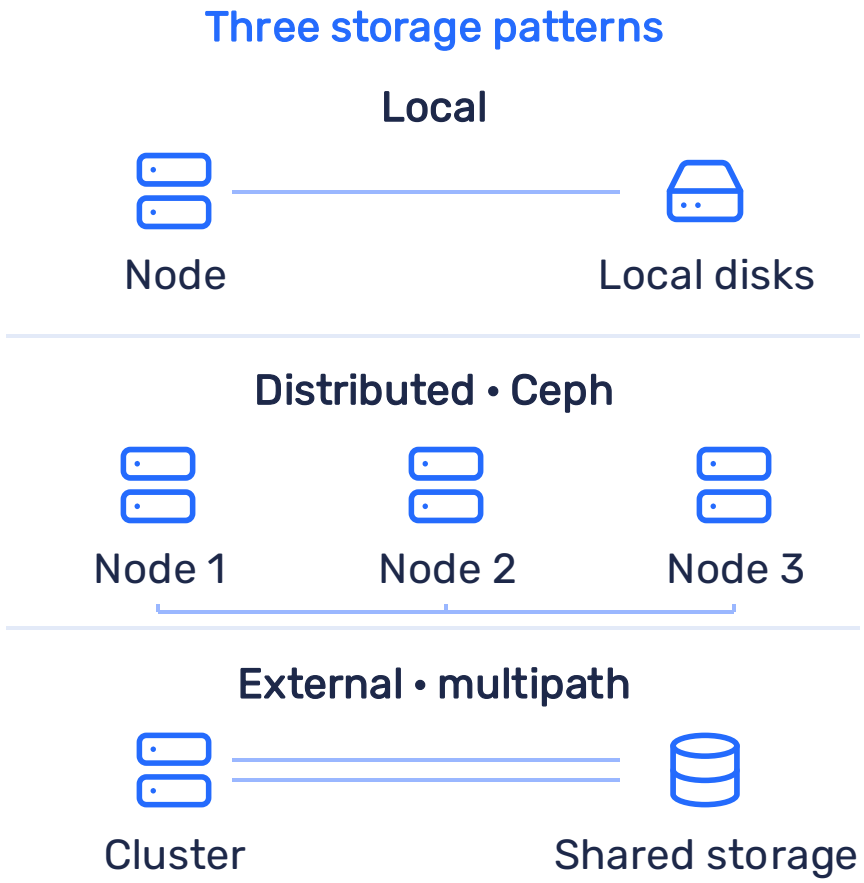
Cluster vs. standalone



initMAX

Choose local, distributed or external storage

- **Local storage**
  - Local disks and pools
  - Standalone servers / specialised applications
- **Distributed storage**
  - Cluster-wide distributed data (Ceph)
  - Plan capacity and the storage network
- **External storage:**
  - Storage shared in cluster
  - Multiple tiers and multipath

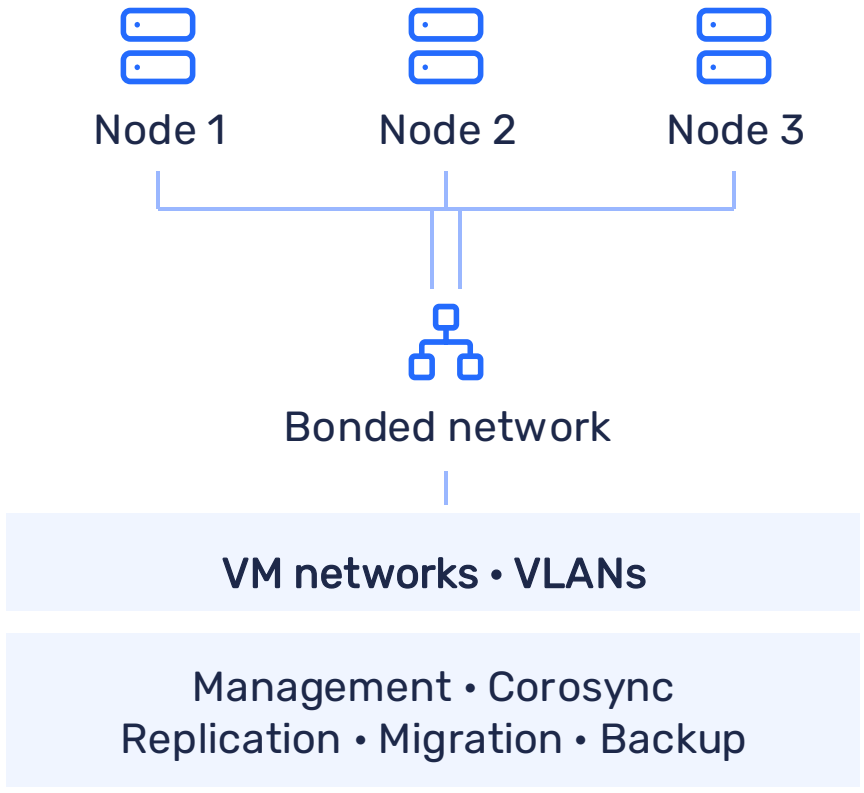


initMAX

Separate guest traffic from cluster services

- **Network for VMs**
  - Isolated networks, multiple VLANs
  - Bonded network
  - Test network
- **Service networks:**
  - Management
  - Corosync
  - Ceph/ZFS replication
  - VM migration
  - Backup
  - ...

Separate traffic by purpose

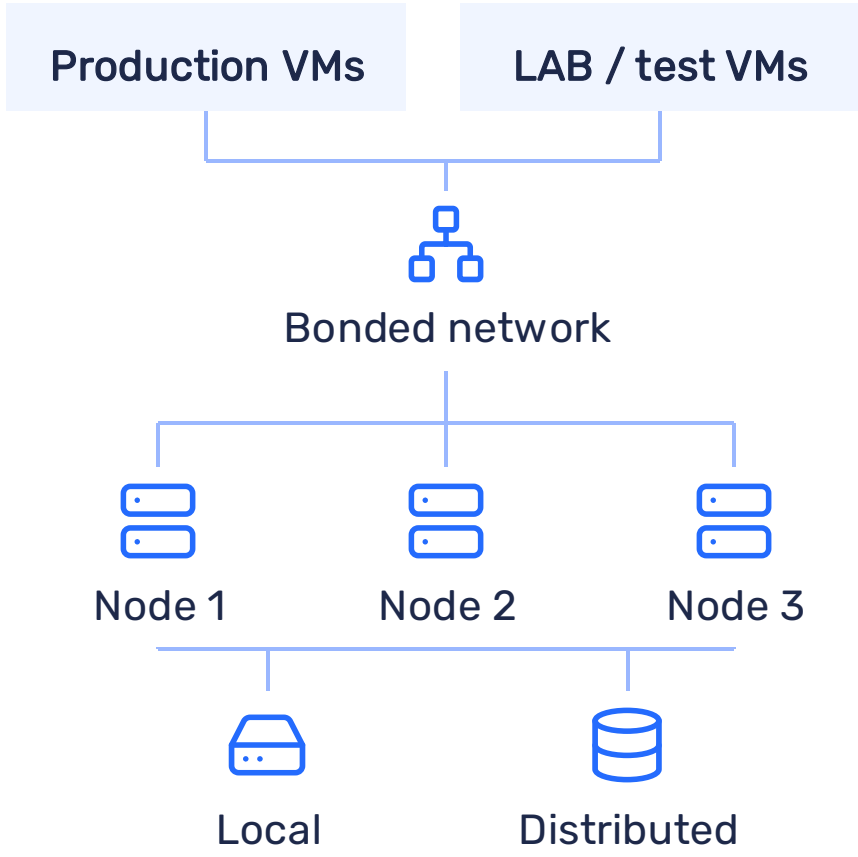


initMAX

Combine workloads, keep clear boundaries

- **Combined architecture**
  - Local and distributed storage
  - Bonded links with traffic separation
  - Workloads with different priorities
  - Mixed CPUs: define a migration baseline
  - Cluster layout with tested failure domains

One combined architecture

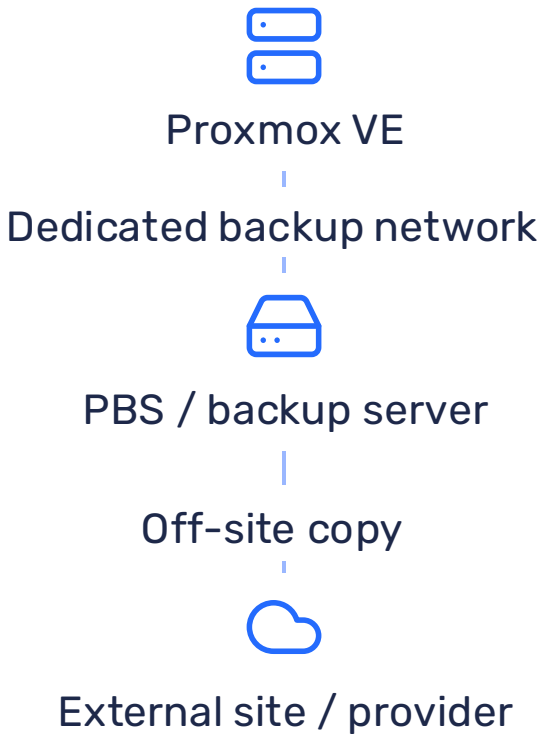


Design and architecture

Keep recovery separate from production

- **Backup/DR architecture**
  - Proxmox Backup Server (PBS) or different vendor
  - Dedicated backup network
- **External backup:**
  - Backup on external site
  - Backup providers

Backup and off-site recovery



Support for systems

Subscription tiers and initMAX prices with 10% discount

	Community	Basic	Standard	Premium
Proxmox list price / **socket** / year	€120	€370	€550	€1,100
initMAX price (guaranteed 10% discount)	€108	€333	€495	€990
Enterprise repository / stable updates	Yes	Yes	Yes	Yes
Vendor support tickets per year	None	3	10	Unlimited
First response (business hours)	No guarantee	1 business day	4 hours	2 hours

Prices in EUR excluding VAT, **per occupied CPU socket**, one-year term. All nodes in a cluster must use the same tier. Source: initmax.com (Proxmox Authorized Reseller), proxmox.com price list, October 2026.

2

Installation



Installation

Two installation paths, one hands-on demo

Manual install

Enter settings in the installer

Boot mode and EULA
Disks and regional settings
Administrator and network

Automatic install

Prepare the configuration in advance

Custom ISO
Web / HTTP(S)
Proxmox Datacenter Manager

Installation demo

Next: walk through the manual steps, then compare the automated approaches.

initMAX

Manual installation: from boot to summary

1. Selection of graphics mode (entering kernel parameters)
2. Acceptance of the EULA
3. Disk selection (ext4, ZFS, btrfs)
4. Location and keyboard settings
5. Setting the root password and email for alerts
6. Network configuration
7. **Installation summary**

Checkpoint: review the selected disk and installation summary before starting.

initMAX

Automated installation: deliver the answer

Where the configuration comes from

Custom ISO

answer.toml inside the ISO

Web / HTTP(S)

Answer returned by your endpoint

Proxmox Datacenter Manager

Prepared Answers served by PDM

→ **Automated installer**

Read the answer
Apply the configuration
Install Proxmox VE

Same installer, different answer delivery. Detailed comparison and examples follow.

Installation

Installer boot menu



initMAX

Installation

Installer boot options

Main menu

```
Install Proxmox VE (Graphical)
Install Proxmox VE (Terminal UI)
Install Proxmox VE (Terminal UI, Serial Console)
Advanced Options
```

Advanced Options

```
Install Proxmox VE (Graphical, nomodeset)
Install Proxmox VE (Terminal UI, nomodeset)
Install Proxmox VE (Graphical, Debug Mode)
Install Proxmox VE (Terminal UI, Debug Mode)
Install Proxmox VE (Serial Console Debug Mode)
Install Proxmox VE (Automated)
Rescue Boot
Test memory (memtest86+)
```

Graphical / Terminal UI

Interactive installation

Automated

Installation with a TOML answer file

Troubleshooting

nomodeset, debug, rescue and memory test

initMAX

Automated installation: three approaches

Approach	Answer location	Configuration change	Typical use
1. Custom ISO	answer.toml inside ISO	Rebuild the ISO	Small LAB or offline deployment
2. HTTP(S)	Your answer endpoint	Update server response	Central configuration or custom logic
3. PDM	Prepared Answers in PDM	Edit configuration in PDM	Host matching and installation overview

All three use the automated installer. PDM serves answers over HTTP(S).

Installation

Shared configuration: answer.toml

```
[global]
keyboard = "en-us"
country = "cz"
fqdn = "pve01.lab.initmax.example"
mailto = "admin@example.com"
timezone = "Europe/Prague"
root-password = "REPLACE_WITH_UNIQUE_PASSWORD"

[network]
source = "from-dhcp"

[disk-setup]
filesystem = "ext4"
# Selected disk is erased
disk-list = ["sda"]
```

Host identity

LAB hostname, keyboard, timezone and administrator.

Network

Use DHCP settings to write the installed static configuration.

Target disk

ext4 on sda. Verify the disk before booting.

1. Answer embedded in ISO

The installation medium carries the complete configuration.

```
proxmox-auto-install-assistant validate-answer answer.toml

proxmox-auto-install-assistant prepare-iso source.iso \
  --fetch-from iso \
  --answer-file answer.toml
```

Boot the prepared ISO and select Automated Installation.

Configuration changes require a new ISO. No answer server is needed.

Installation

2. Answer fetched over HTTP(S)

The prepared ISO points to your answer endpoint.

```
proxmox-auto-install-assistant prepare-iso source.iso \
  --fetch-from http \
  --url "https://provisioning.initmax.com/answer"
```

Installer request

HTTP POST with machine information, including network MAC addresses.

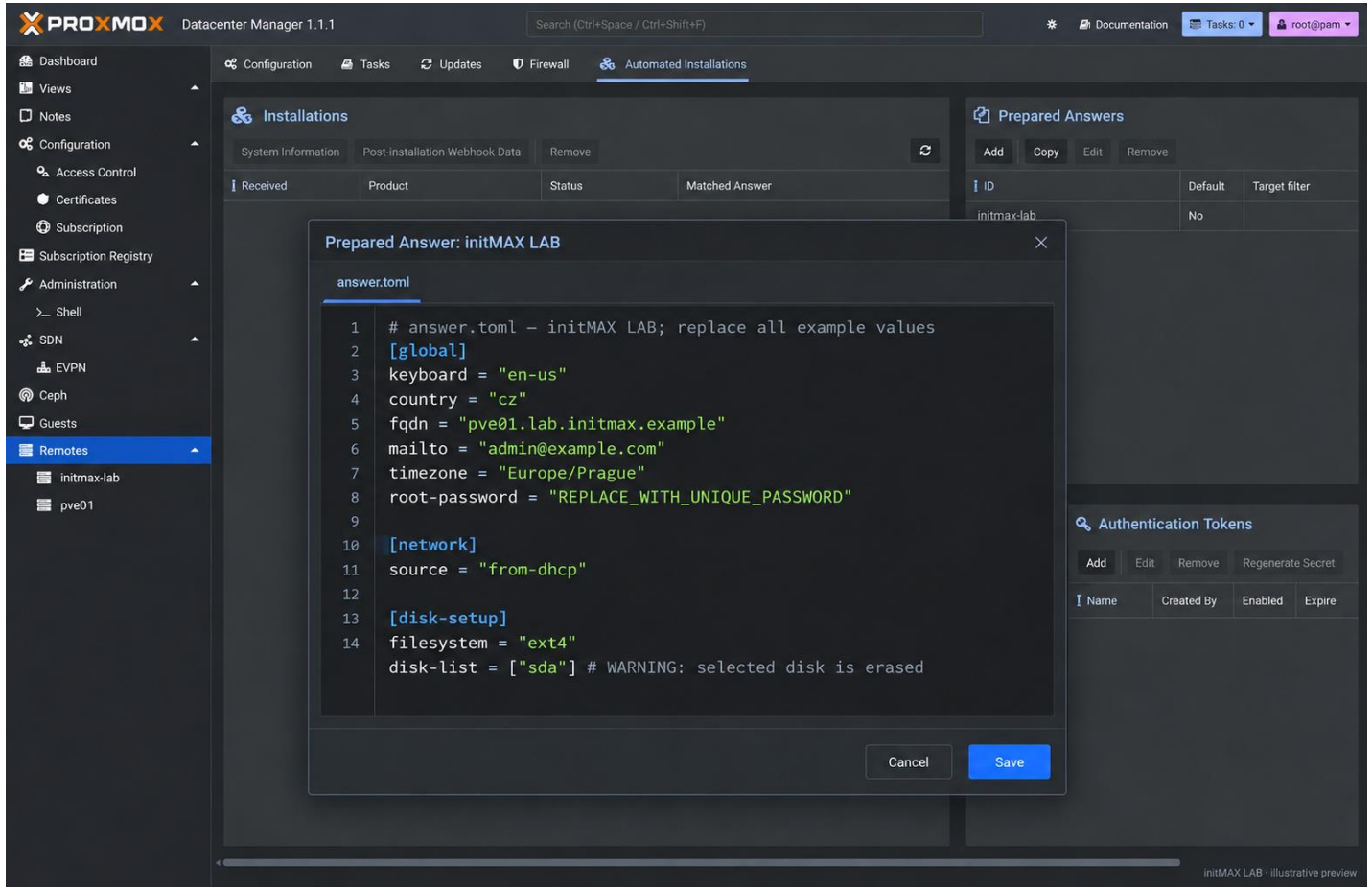
Server response

Return the answer for this host. Change it centrally before the next run.

Demo evidence: request log, selected answer and installer progress.

Installation

3. Provisioning with PDM



Illustrative preview using initMAX LAB example data

Prepared Answers

Define the LAB configuration and its target filter.

Authentication Tokens

Create a dedicated installer token and prepare the ISO.

Installations

Review the incoming machine information and matched answer.

3. PDM: prepare the installation ISO

The same HTTP fetch mode, with the PDM endpoint and installer token.

```
PDM_URL="https://pdm.lab.initmax.example"

proxmox-auto-install-assistant prepare-iso source.iso \
  --fetch-from http \
  --url "$PDM_URL/api2/json/auto-install/answer" \
  --cert-fingerprint "REPLACE_WITH_PDM_SHA256_FINGERPRINT" \
  --answer-auth-token "initmax-lab:REPLACE_WITH_SECRET"
```

Use the actual LAB URL, certificate fingerprint and dedicated installer token.

3

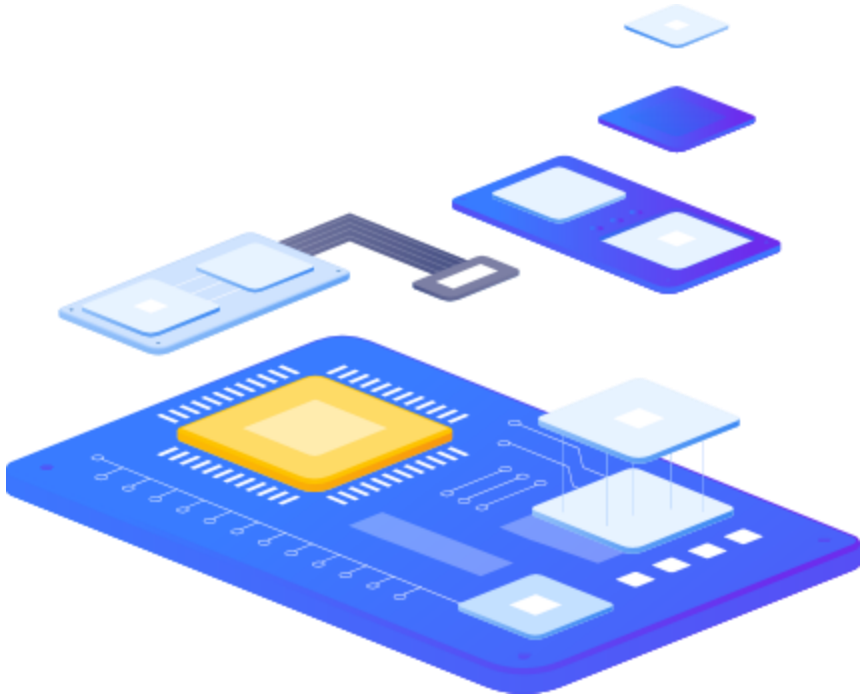
Configuration after install



initMAX

Validate the host before deploying guests

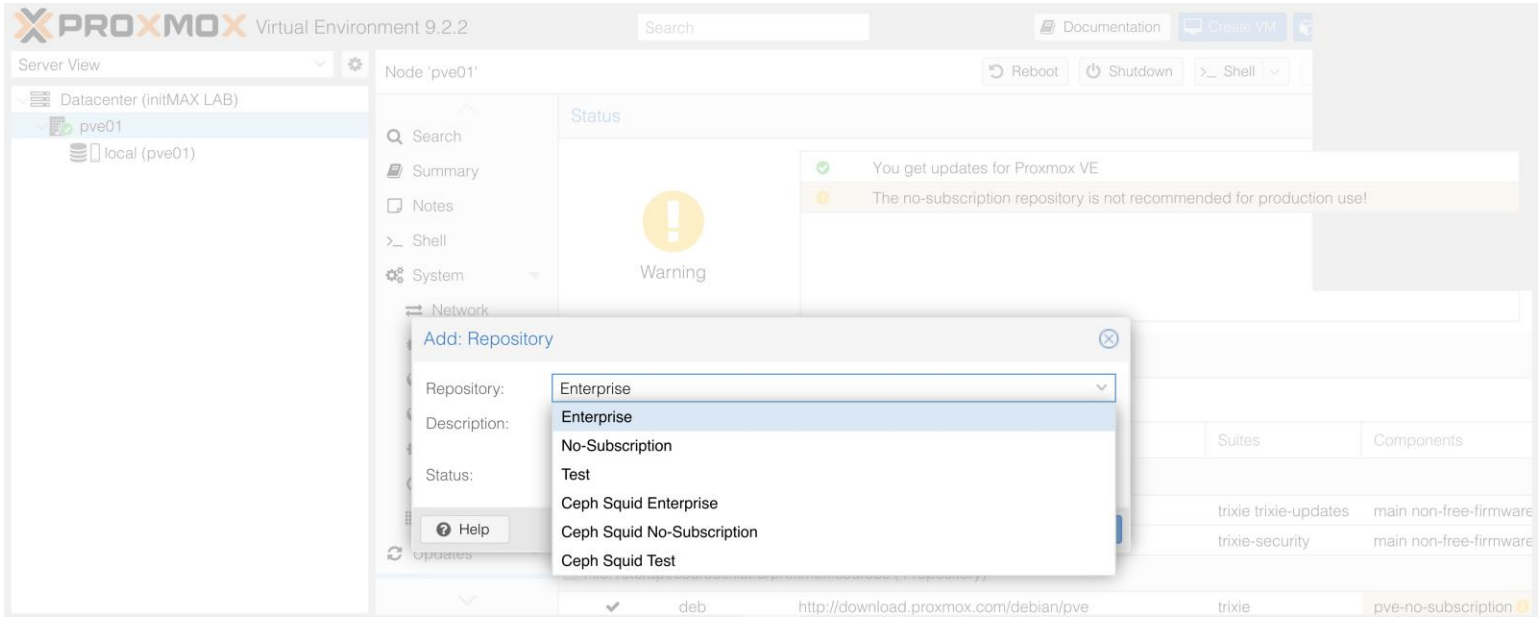
- Repository types and updates
- System checks
- Datacenter options
- Storage check
- Network check



Repository types and updates

- **Types of repository:**
  - Test
  - No-Subscription
  - Enterprise
- **Update node:**
  - Changelog
  - Upgrade
  - Version

Repository selection in the web interface

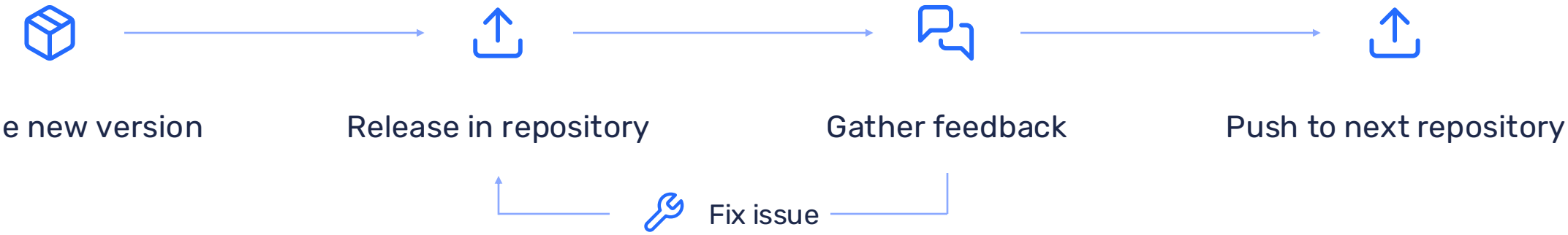


Select the repository, then review and apply node updates.

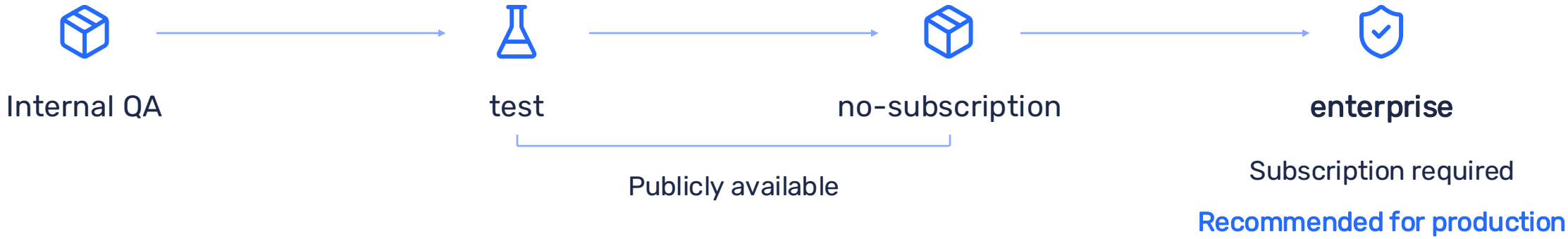
Configuration after install

Repository types and updates

Release workflow



Package repositories



Configuration after install

Demo: repositories and node updates

Live demo route

DEMO

Switch to the live Proxmox VE interface

1 Repositories

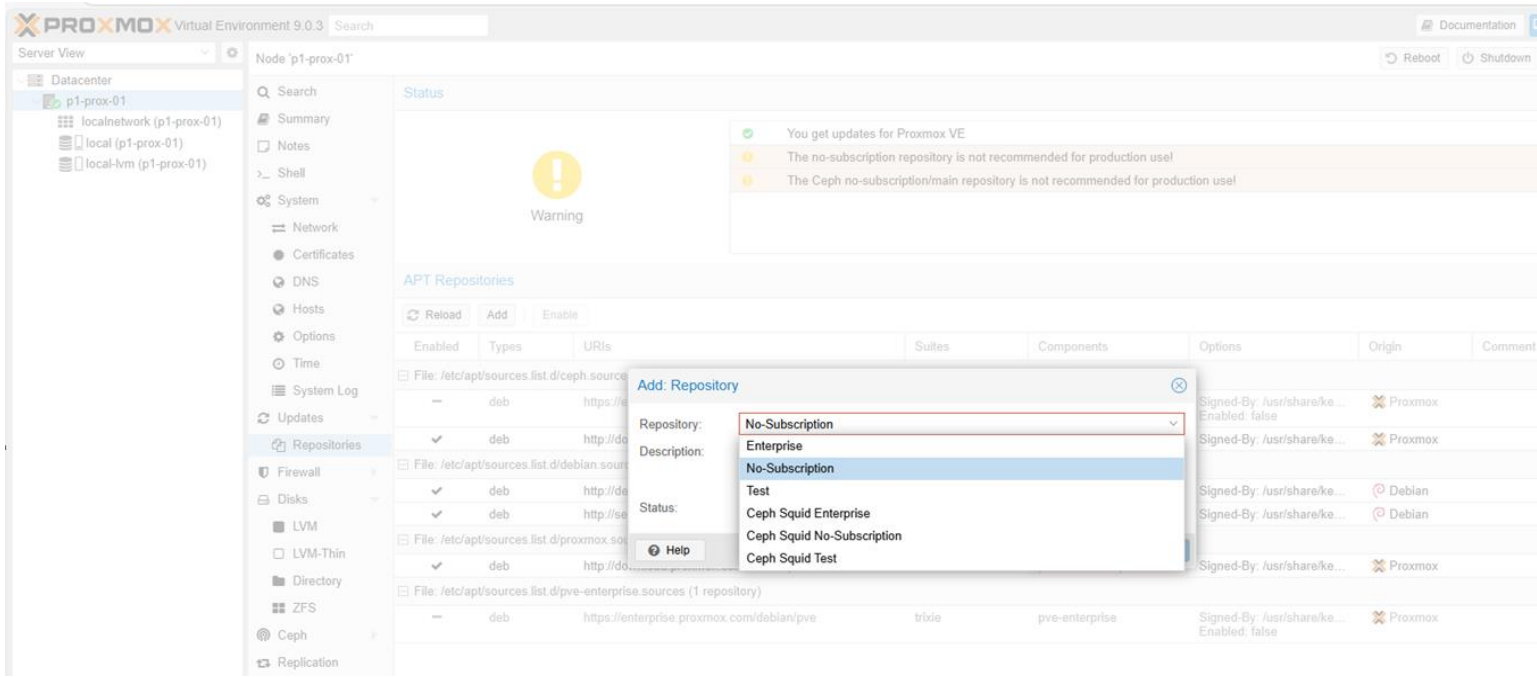
Show Test, No-Subscription and Enterprise.

2 Review updates

Refresh the package list and open the changelog.

3 Update and verify

Show the upgrade process and check the node version.



Show live: repository → changelog → upgrade → version

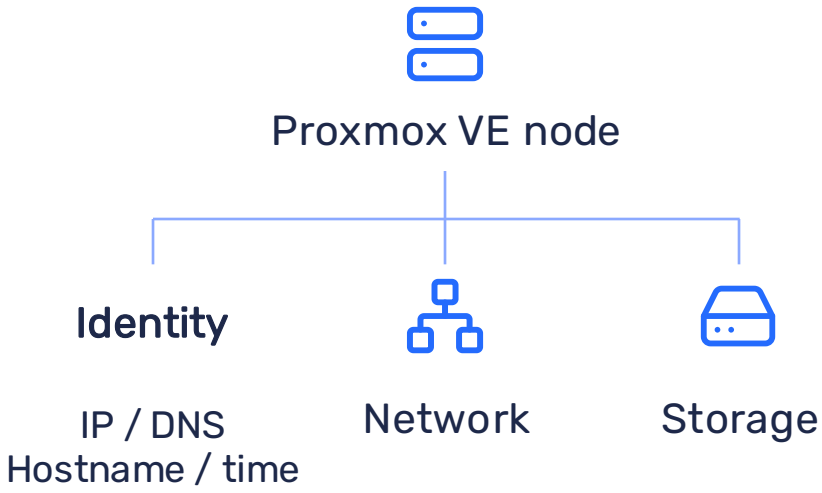
Configuration after install

1. Verify node identity and health

1 Node checks | 2 Datacenter defaults | 3 Storage checks | 4 Network tests

- **Check system:**
  - IP, DNS, hostname, time
  - Check network
  - Check storage and disk

One node, three checks



Checkpoint: verify identity and basic access before tuning the node.

Configuration after install

2. Set shared datacenter defaults

1 Node checks | 2 Datacenter defaults | 3 Storage checks | 4 Network tests

- **Options for change:**
  - VMID range
  - Tag Style Override
  - Maximal Workers
  - Bandwidth Limits

Shared policy, consistent defaults



Datacenter options

Identity & visibility	Workload limits
VMID range Tag Style Override	Maximal Workers Bandwidth Limits

Checkpoint: agree on shared defaults before deploying workloads.

Configuration after install

3. Validate storage layout and health

1 Node checks | 2 Datacenter defaults | 3 Storage checks | 4 Network tests

- **Check storage:**
  - Disk layout
  - Hide local storage
  - S.M.A.R.T.
  - Wearout

From disk health to storage view

Physical disks

S.M.A.R.T. / wearout

Disk layout

Review the installed layout

Storage visibility

Hide local storage

Checkpoint: confirm disk health, layout and intended storage visibility.

Configuration after install

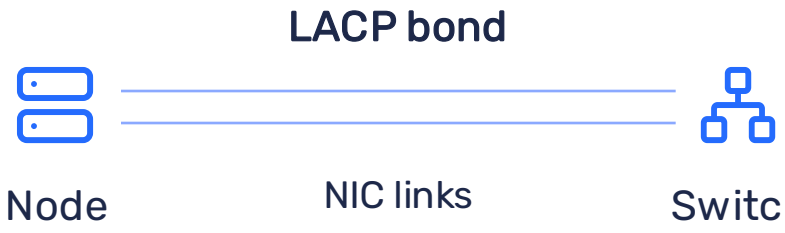
initMAX

4. Test network paths and failover

1 Node checks | 2 Datacenter defaults | 3 Storage checks | 4 Network tests

- **Network checking list:**
  - Speed of connected NIC
  - Connection to port in switch (CDP/LLDP)
  - VLAN/MTU
  - LACP bonding and test failure
  - REAL Bandwidth Test

Test the complete network path



1 Link speed • CDP/LLDP

2 VLAN / MTU • LACP failover

3 Real bandwidth test

Checkpoint: test link loss and measure real throughput before handover.

4

Create your first VM



initMAX

VM and LXC use different isolation models

Virtual machine

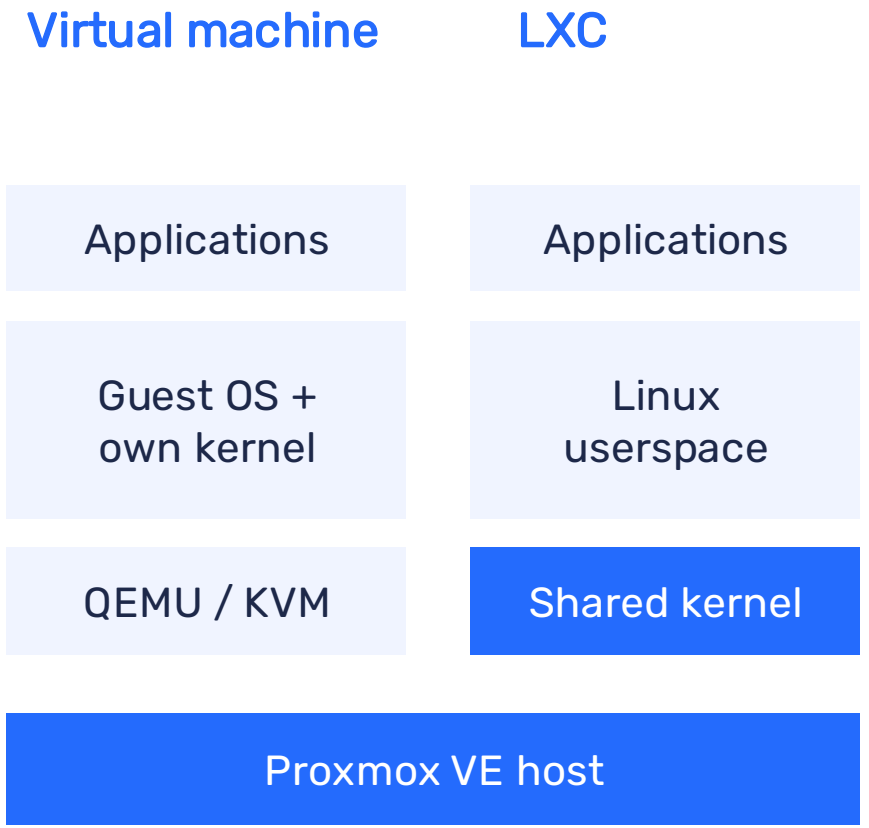
A separate guest OS and kernel.
Virtual hardware provided by QEMU/KVM.

Linux container

Linux userspace sharing the host kernel.
No separate guest kernel.

Choose the guest model first

VM for a separate OS; LXC for Linux services.



initMAX

General: identify the VM and control startup

Node, VM ID and Name

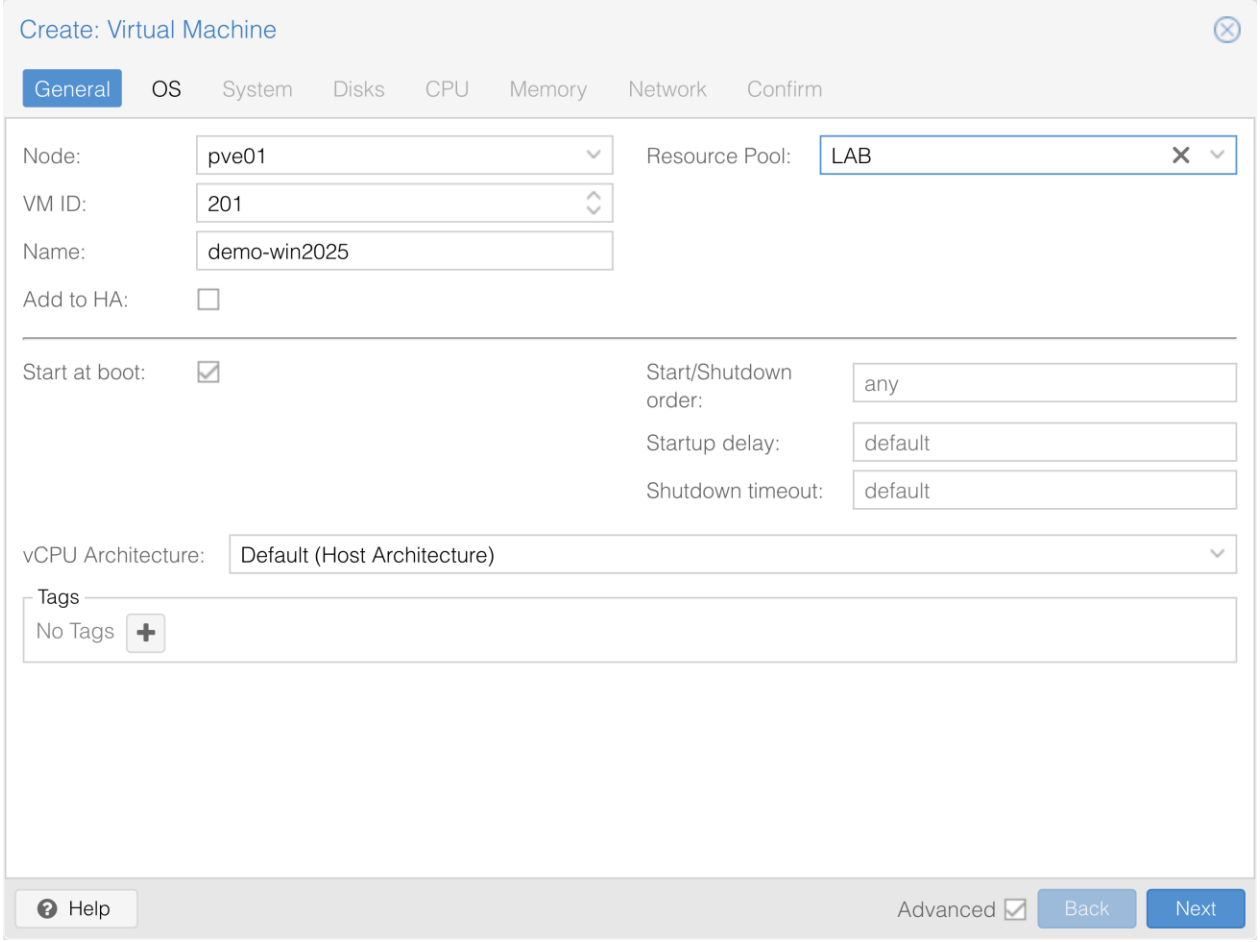
pve01 hosts VM 201. The ID is unique in the cluster; demo-win2025 gives administrators a recognizable name.

Resource Pool: LAB

A pool groups guests for organization and permissions.

Start at boot

Enabled here: the VM starts when its host boots. Startup order and delay can sequence dependent services.



Lab screenshot • Proxmox VE Administration Guide

initMAX

OS: separate installation media from drivers

Installation ISO

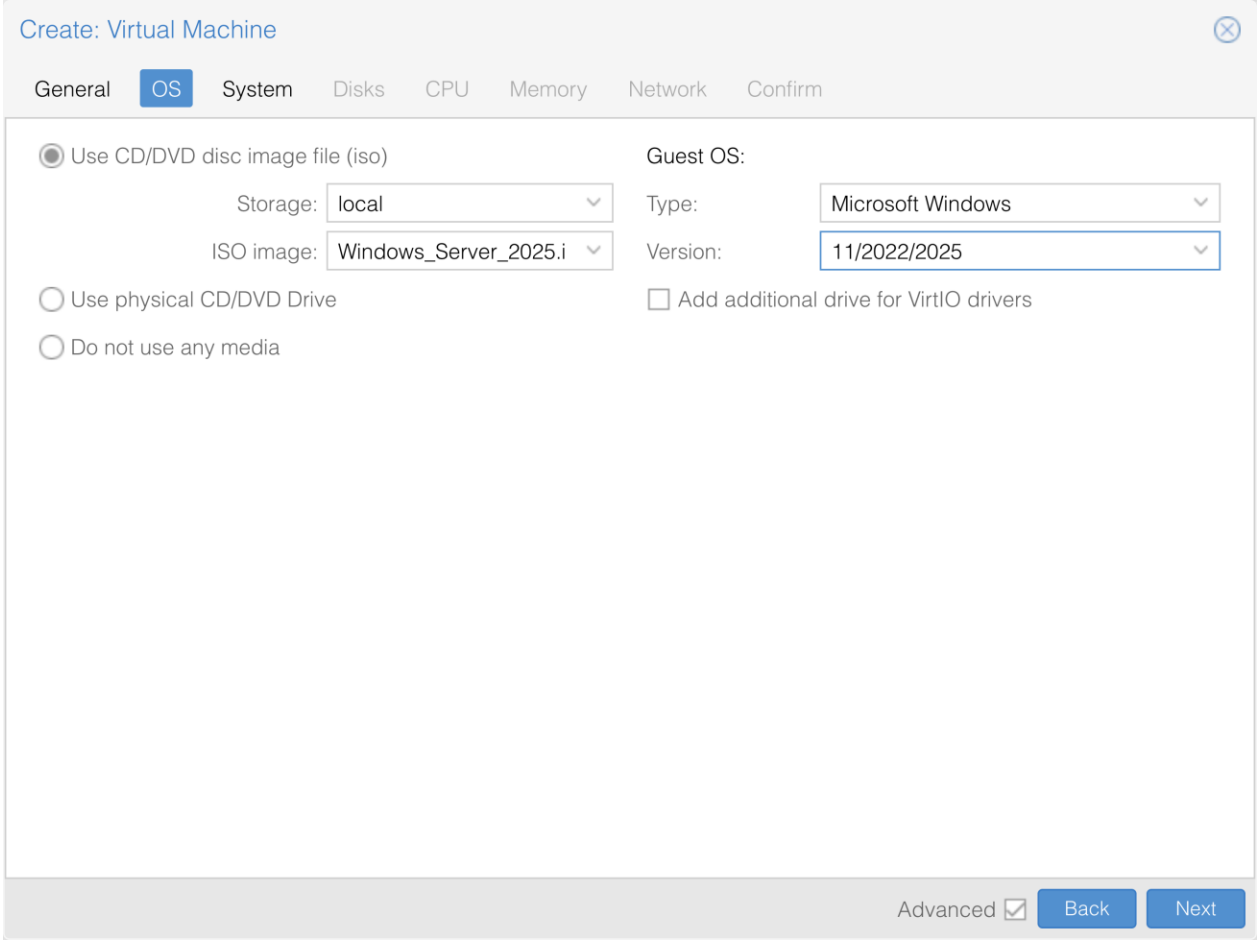
Windows_Server_2025.iso on local is the installer. Selecting it does not install Windows until you boot the VM.

Guest OS profile

Microsoft Windows • 11/2022/2025 selects the guest profile. It is separate from the contents of the ISO.

VirtIO drivers

The extra driver drive is unchecked here. Attach a VirtIO driver ISO before installation if Windows needs disk or network drivers.



Lab screenshot • Proxmox VE Administration Guide

initMAX

System: choose firmware and guest integration

q35, UEFI and EFI disk

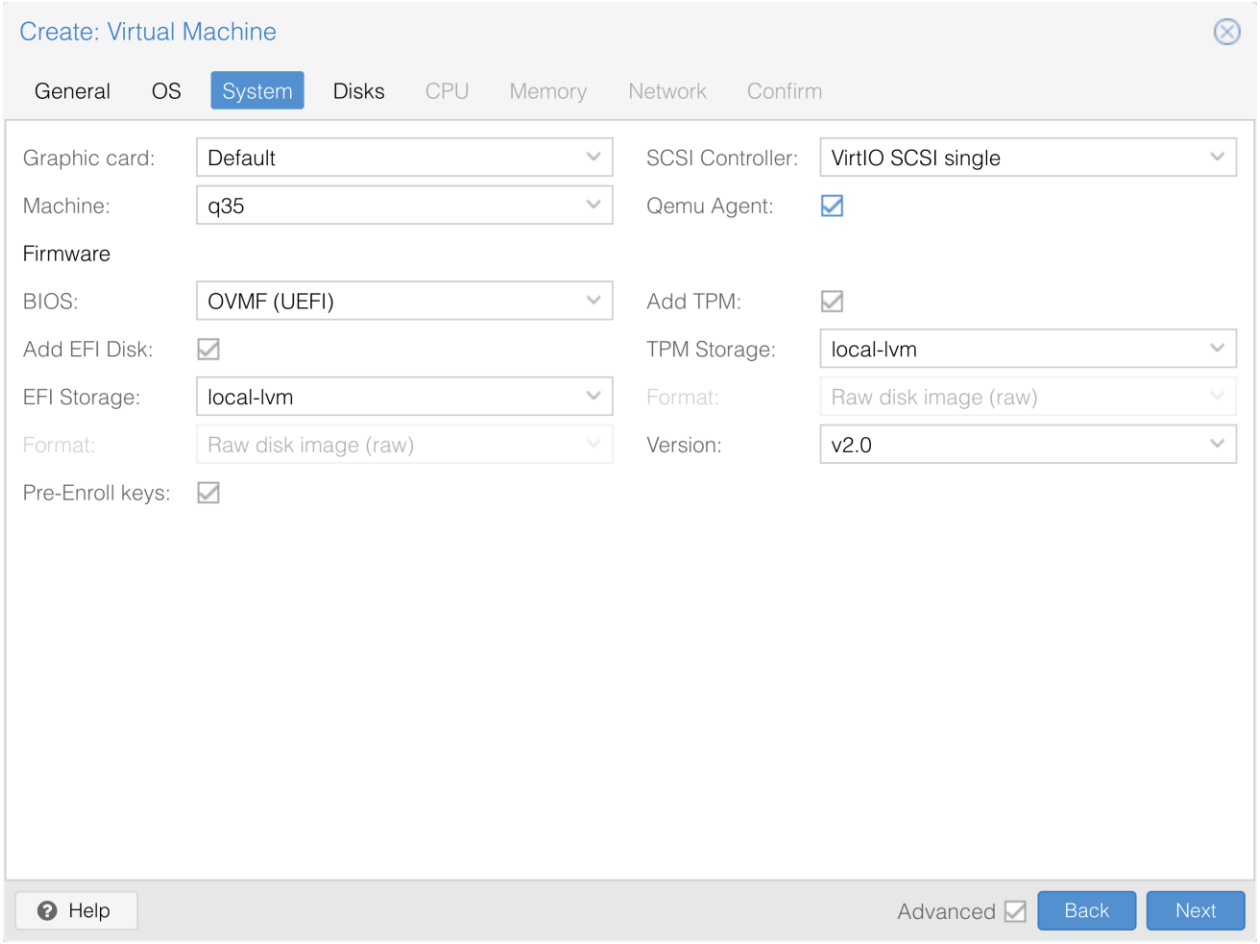
q35 is the virtual machine platform; OVMF provides UEFI. The EFI disk stores firmware settings, not Windows files.

TPM 2.0 and keys

TPM 2.0 stores TPM state; pre-enrolled keys belong to UEFI Secure Boot. Preserve both TPM and EFI disks in backups.

QEMU Guest Agent

The checkbox enables communication. Install and run the agent inside Windows too; the checkbox alone is not enough.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Disks: choose capacity and the I/O path

64 GiB on local-lvm

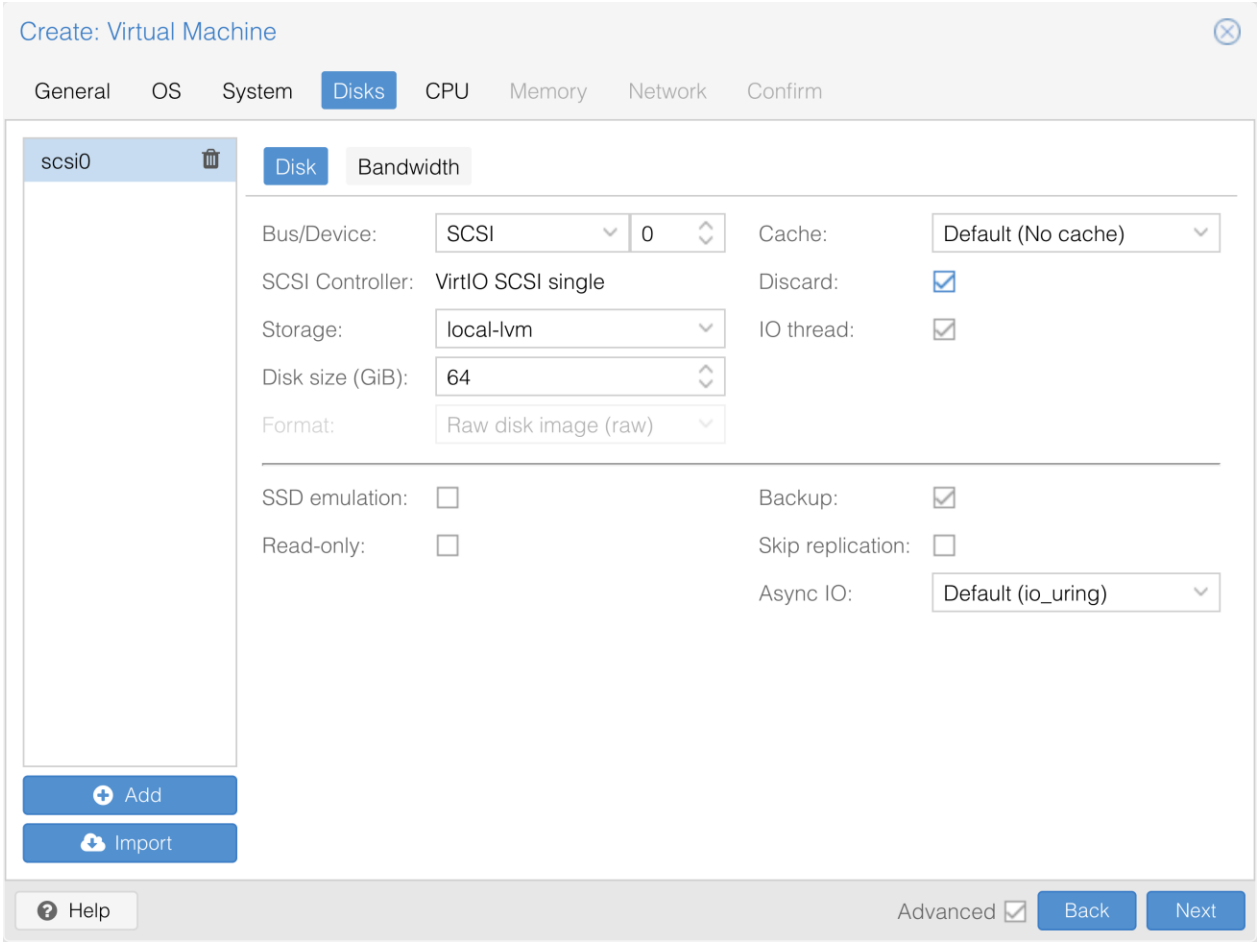
scsi0 is the system disk. Capacity must cover Windows, updates and applications; monitor free space on the host too.

VirtIO SCSI single + IO thread

The paravirtualized controller uses guest drivers. IO thread - one I/O thread per storage controller.

Discard and cache

Discard passes guest TRIM requests to supported storage. No cache avoids the host page cache; it does not disable all storage caches.



Lab screenshot • Proxmox VE Administration Guide

initMAX

CPU: balance guest capacity and portability

1 socket × 4 cores

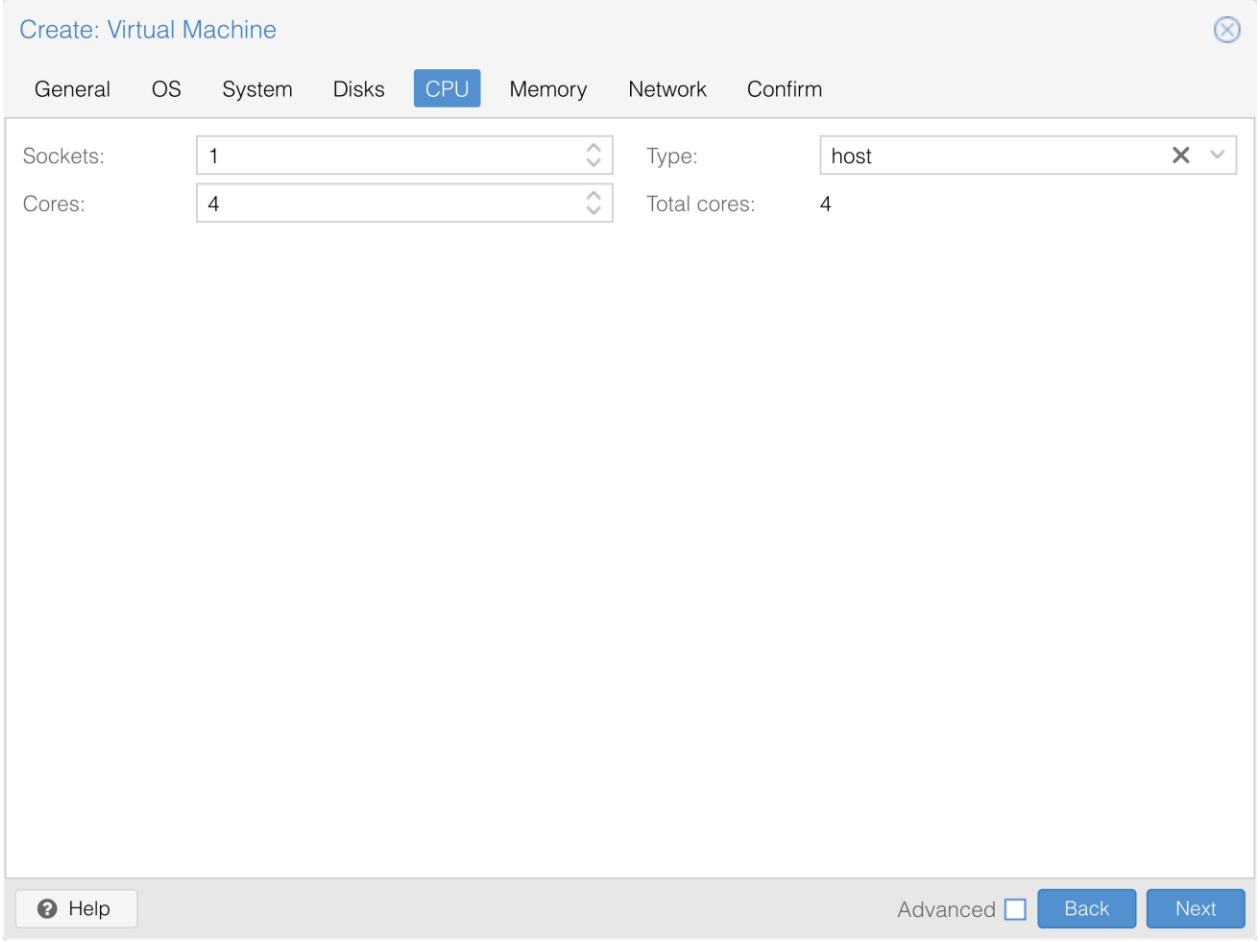
The guest sees four vCPUs. These are scheduled on host CPUs, not four physical cores reserved exclusively for this VM.

CPU type: host

Exposes host CPU capabilities. Before migration, verify that the destination CPU supports the guest CPU feature set.

NUMA

Disabled in this lab. Size small VMs conservatively; extra vCPUs do not automatically make an application faster.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Memory: understand the fixed 8 GiB allocation

8192 MiB = 8 GiB

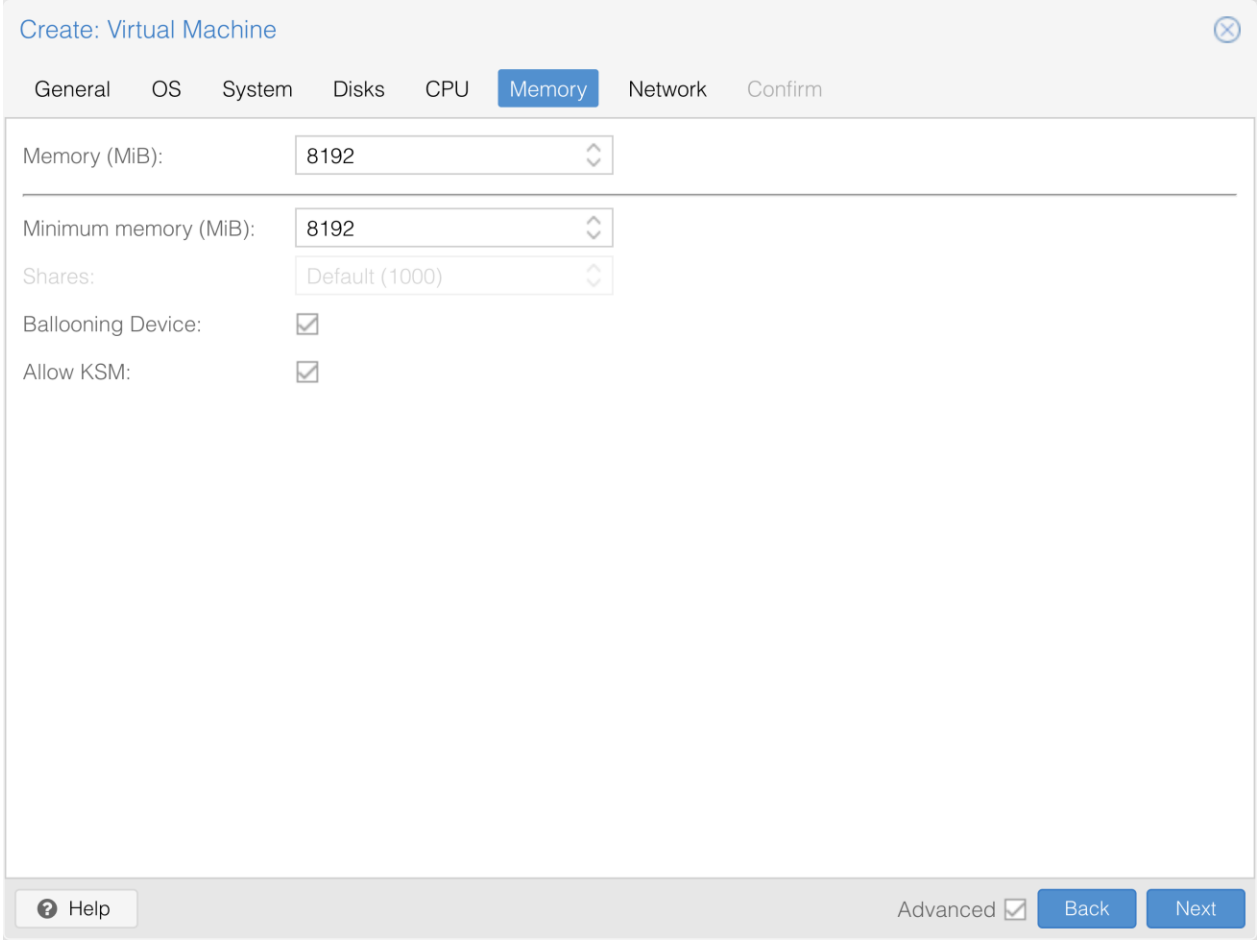
Memory and minimum memory are both 8192 MiB. This leaves no dynamic range between minimum and maximum.

Ballooning device

Enabled here. A guest balloon driver supports reporting and memory control; it is different from the QEMU Guest Agent.

Allow KSM

Allows identical memory pages to be shared by the host. Do not budget capacity on the assumption that deduplication will occur.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Network: connect VirtIO to the right bridge

Bridge: vmbr0

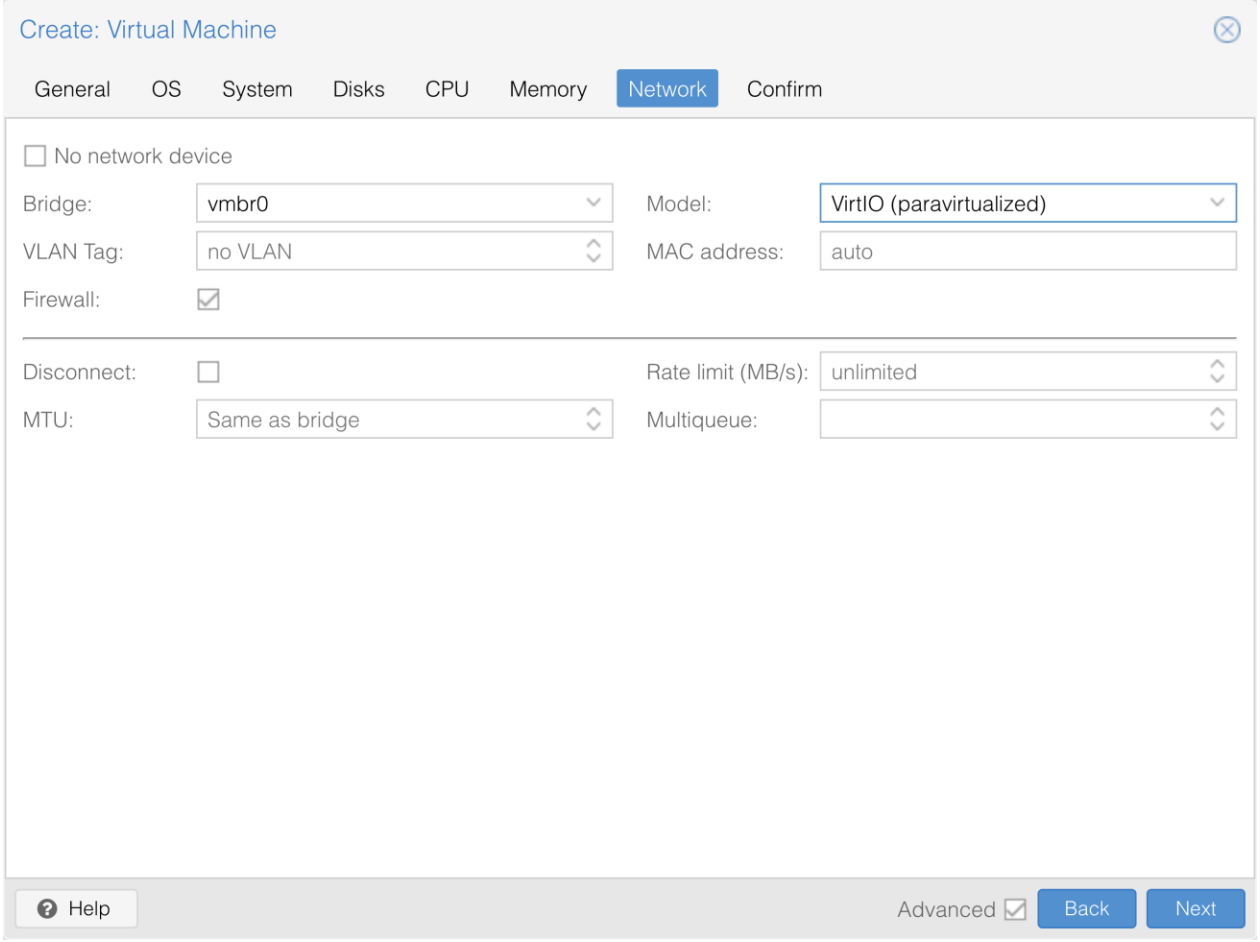
Connects the virtual NIC to a host bridge. Reachability depends on the bridge uplink and the physical network.

VirtIO and VLAN

VirtIO needs a Windows driver. No VLAN tag is configured here; do not add a tag unless the network design requires one.

Firewall and guest IP

The NIC firewall flag is enabled; rules and higher-level firewall settings still matter. Configure IP and DNS inside Windows.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Confirm: verify the platform before creation

Identity and resources

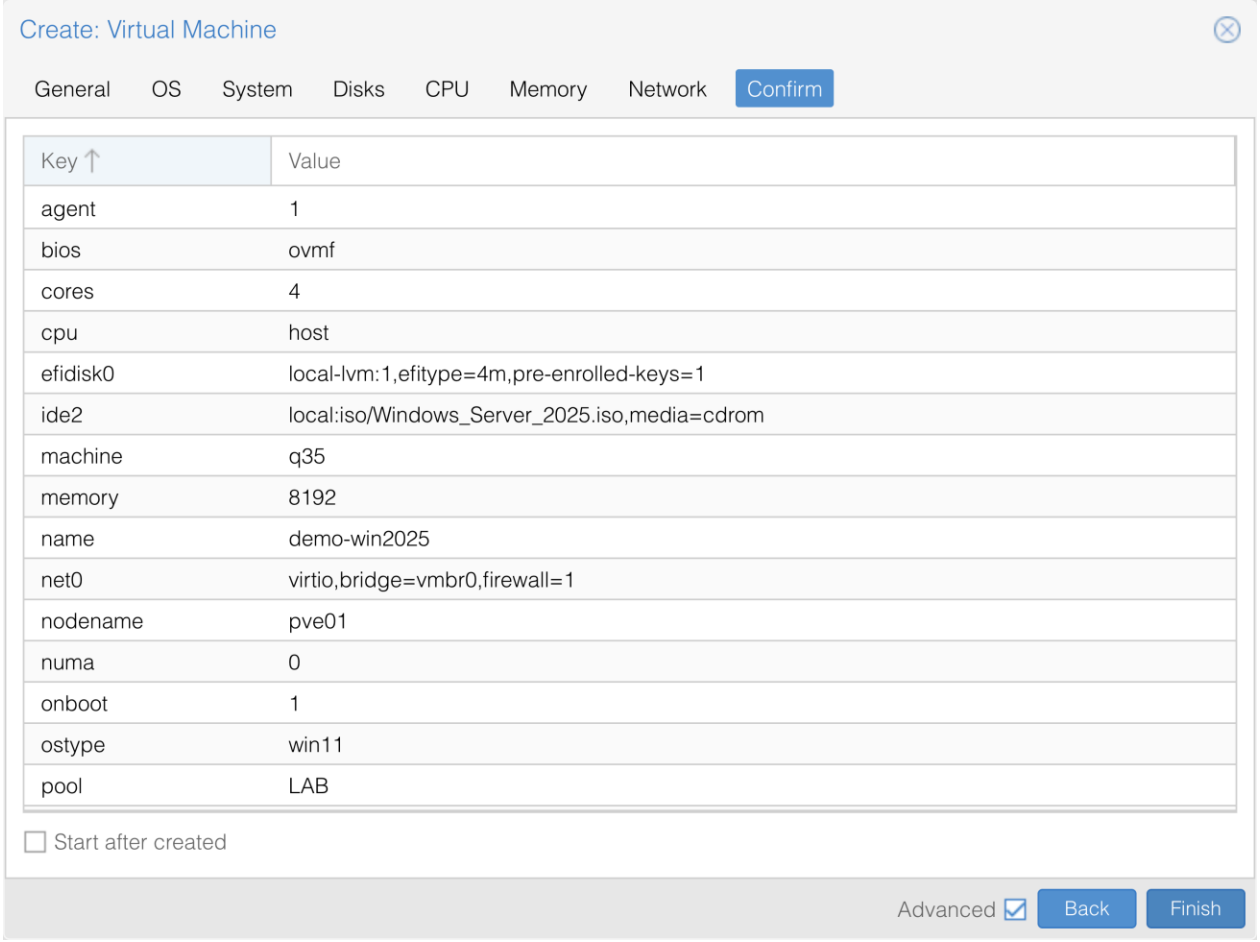
Check pve01, VM 201, demo-win2025 and LAB against the plan. This review shows four cores and 8192 MiB RAM.

Firmware and installation media

Verify OVMF, q35, EFI settings and the Windows Server 2025 ISO. A wrong boot mode can prevent the guest from booting.

Agent is not installed yet

agent=1 records the enabled option. After installing Windows, install the guest agent and verify that it responds.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Confirm: check disks, network and first boot

Disk and TPM state

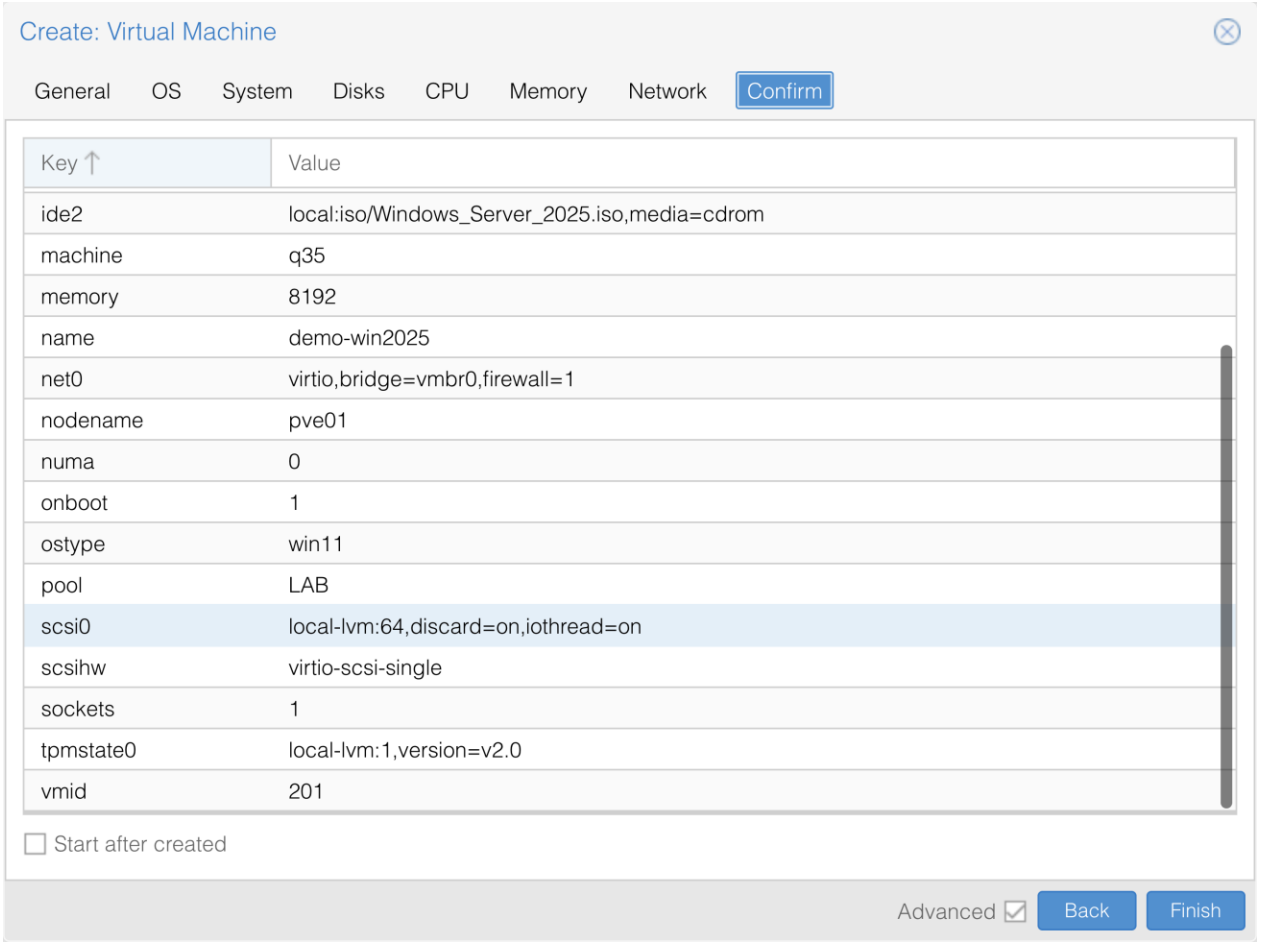
Review scsi0: local-lvm, 64 GiB, discard and IO thread. Confirm VirtIO SCSI single and TPM version 2.0.

Network and boot policy

net0 uses VirtIO on vmbr0 with the firewall flag. onboot=1 is host-start policy, not immediate startup after creation.

Finish, then validate

Start after created is unchecked. Finish creates the VM; then start it, install Windows and test disk, network and agent.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Post-install: drivers, guest tools and validation

1 Attach the VirtIO driver ISO

Mount virtio-win ISO as a second CD/DVD. If Windows Setup cannot see the SCSI disk, load its virtioscsi driver first.

2 Install the guest tools

In Windows, run virtio-win-guest-tools.exe as administrator. Install VirtIO drivers and the QEMU Guest Agent component.

3 Enable and verify the agent

Enable Options → QEMU Guest Agent and check its Windows service. Stop/start the VM if the agent channel was newly enabled.

4 Patch and check devices

Run Windows Update and reboot. Check Device Manager: disk, network and balloon devices must have working drivers.

5 Return to normal boot

Eject installer and driver ISOs; put the system disk first in boot order. Confirm the VM boots without installation media.

6 Validate before handover

Check IP, DNS, time and connectivity. On the host: qm agent 201 ping. Test clean shutdown and run a backup/restore check.

Sources: Proxmox VE Windows guest best practices • QEMU Guest Agent

initMAX

Demo: create and validate the first VM

Live demo route

1 Identify

Select node, VM ID and name.

2 Configure

Choose ISO, disks, CPU, memory and network.

3 Review and create

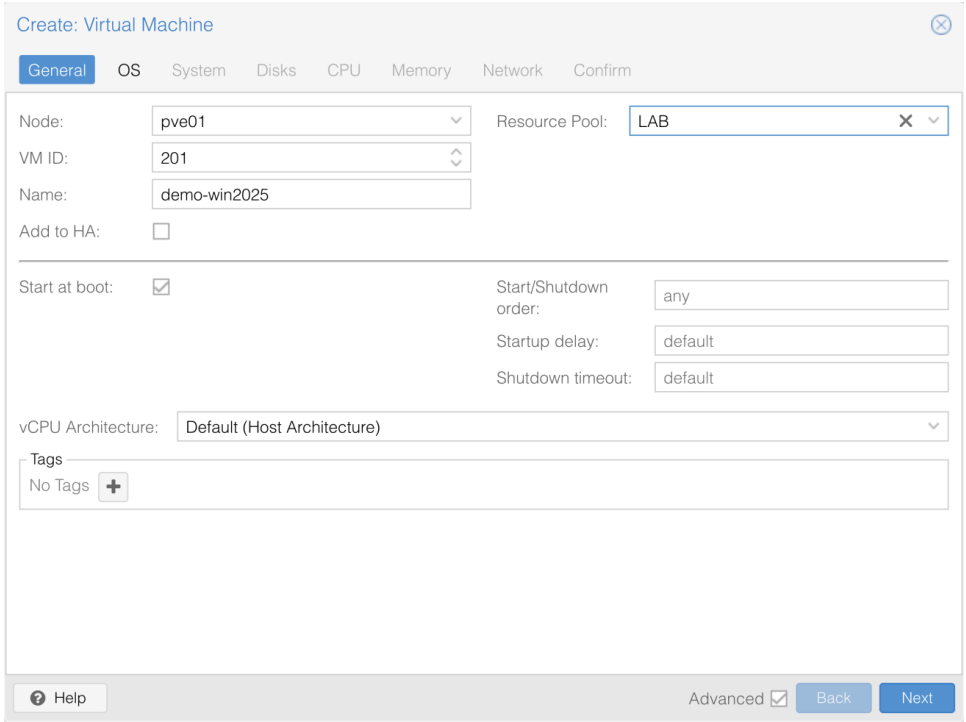
Check the summary, then create the VM.

4 Tools and validation

Install tools; test agent, network and shutdown.

DEMO

Switch to the live Proxmox VE interface



Show live: create → install OS → guest tools → validate

5

Create your first LXC



initMAX

General: choose the container isolation

Unprivileged container

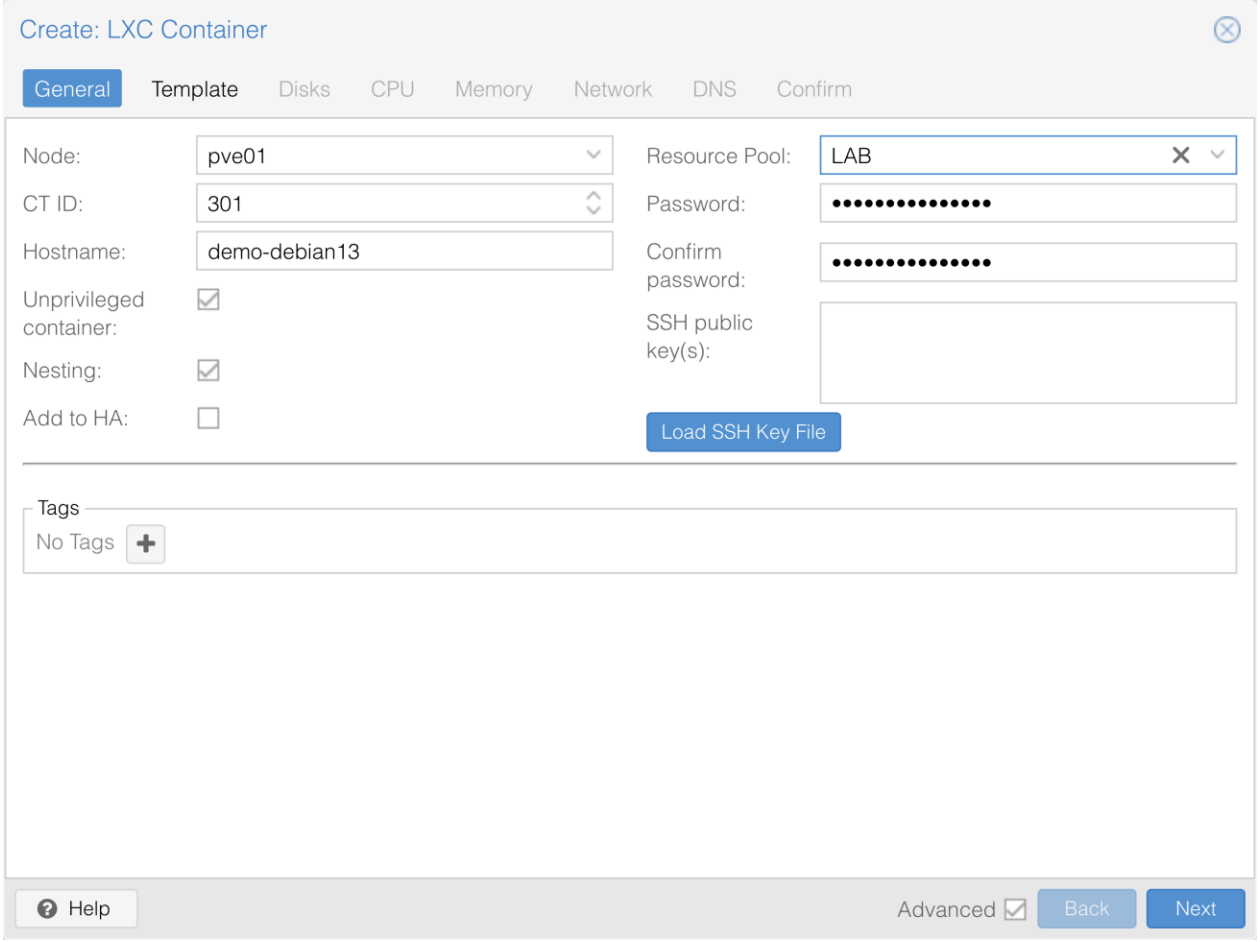
Checked here: container root is mapped to an unprivileged host user. The container still shares the host kernel.

Nesting

Enabled for this lab. It exposes additional host interfaces for nested workloads; enable it only when required.

Identity and access

CT 301 is demo-debian13 on pve01 in LAB. Password fields are set; the SSH public key field is empty.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Template: deploy userspace, not an installer

Debian 13 template

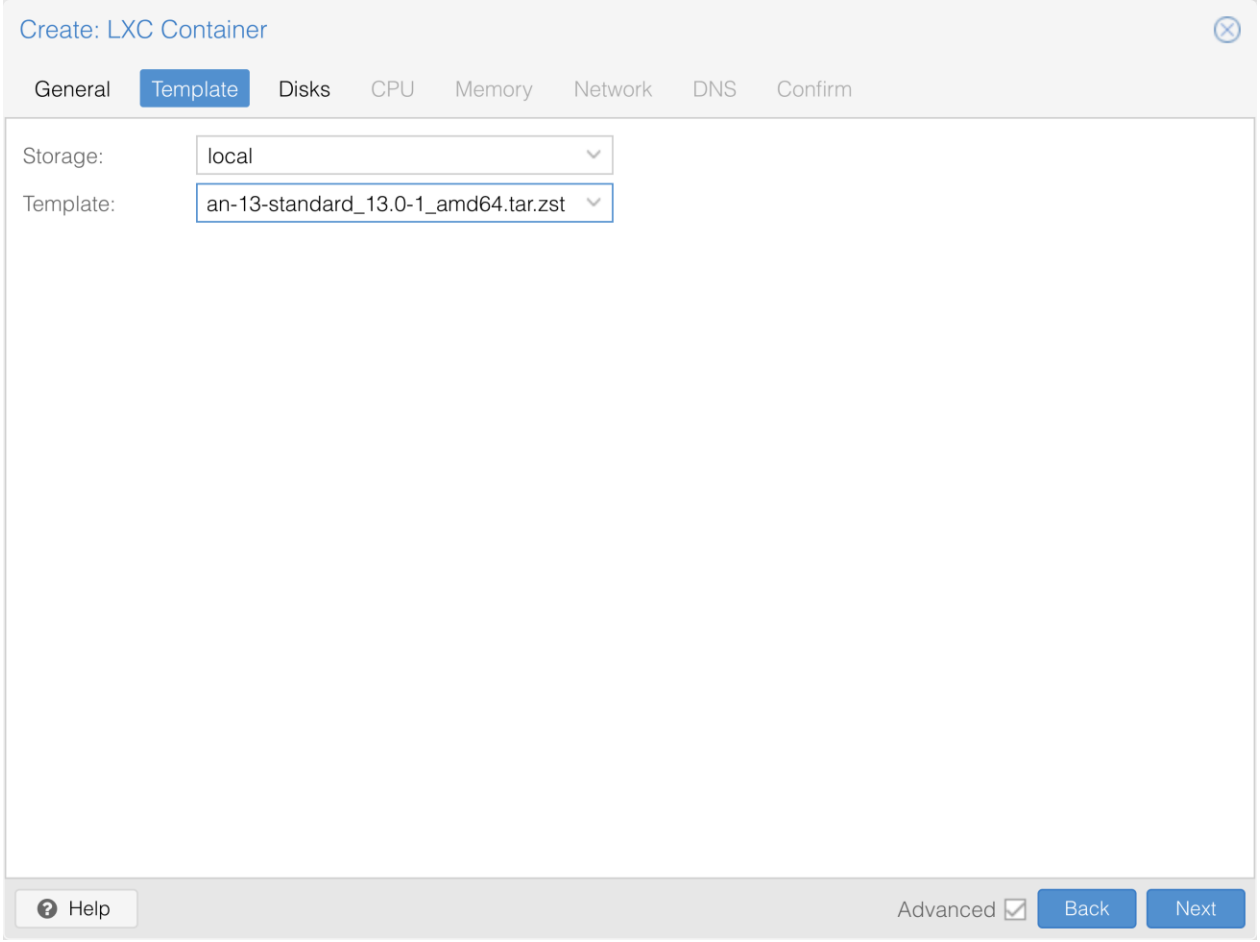
The selected archive contains the initial Debian root filesystem. Unlike a VM ISO, it is unpacked rather than booted.

Template storage: local

local stores the template archive. The next step selects separate storage for the running container root filesystem.

Shared host kernel

The template supplies userspace, not a separate kernel. Choose an application and distribution suitable for LXC.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Disks: size the container root filesystem

rootfs: 16 GiB

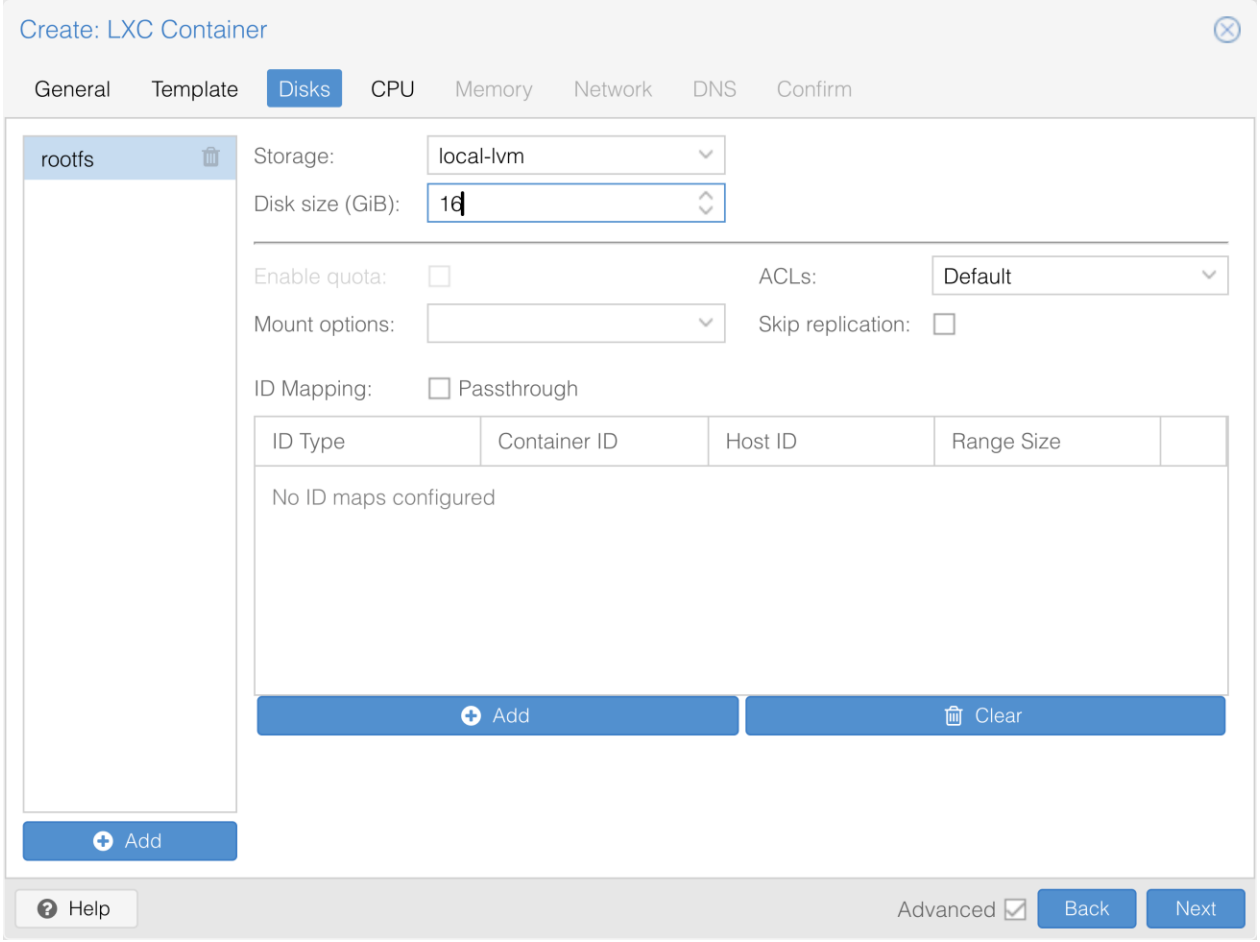
local-lvm stores the container root filesystem. Include packages, logs and application data when planning capacity.

Permissions and ACLs

ACLs remain at Default. Do not change quota or mount options without checking storage support and application needs.

ID mapping

Passthrough is unchecked and no custom maps are configured. Host bind mounts require careful ownership and UID/GID planning.



Lab screenshot • Proxmox VE Administration Guide

initMAX

CPU: distinguish core count, limit and weight

Cores: 2

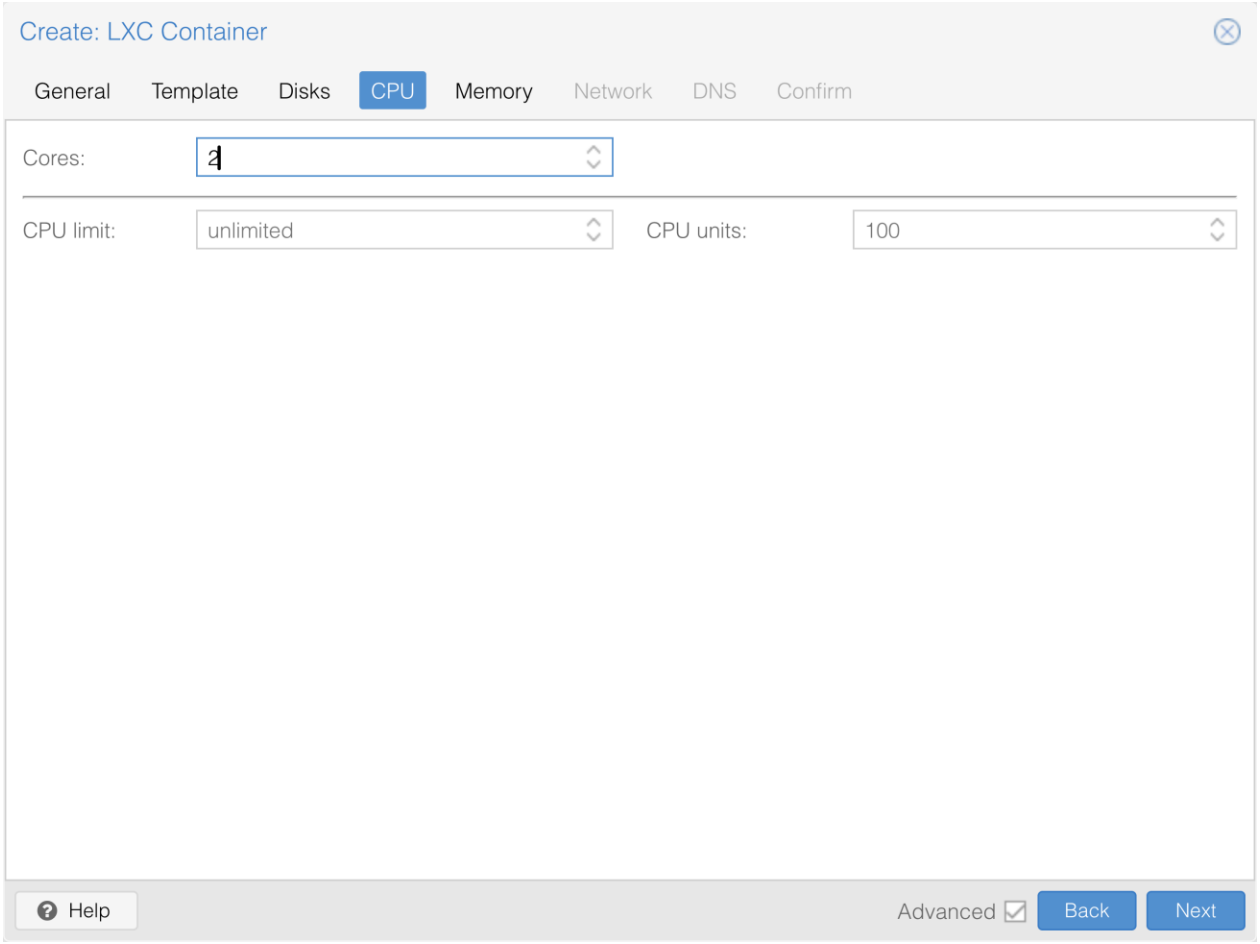
Restricts the CPUs available to the container. Tasks still run under the host scheduler; cores are not exclusively reserved.

CPU limit: unlimited

There is no additional CPU-time cap here. A CPU limit is a separate control from the number of available cores.

CPU units: 100

Relative CPU weight under contention, not 100 percent CPU. Equal weights mean equal priority, not guaranteed performance.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Memory: set RAM and swap as separate limits

Memory: 2048 MiB

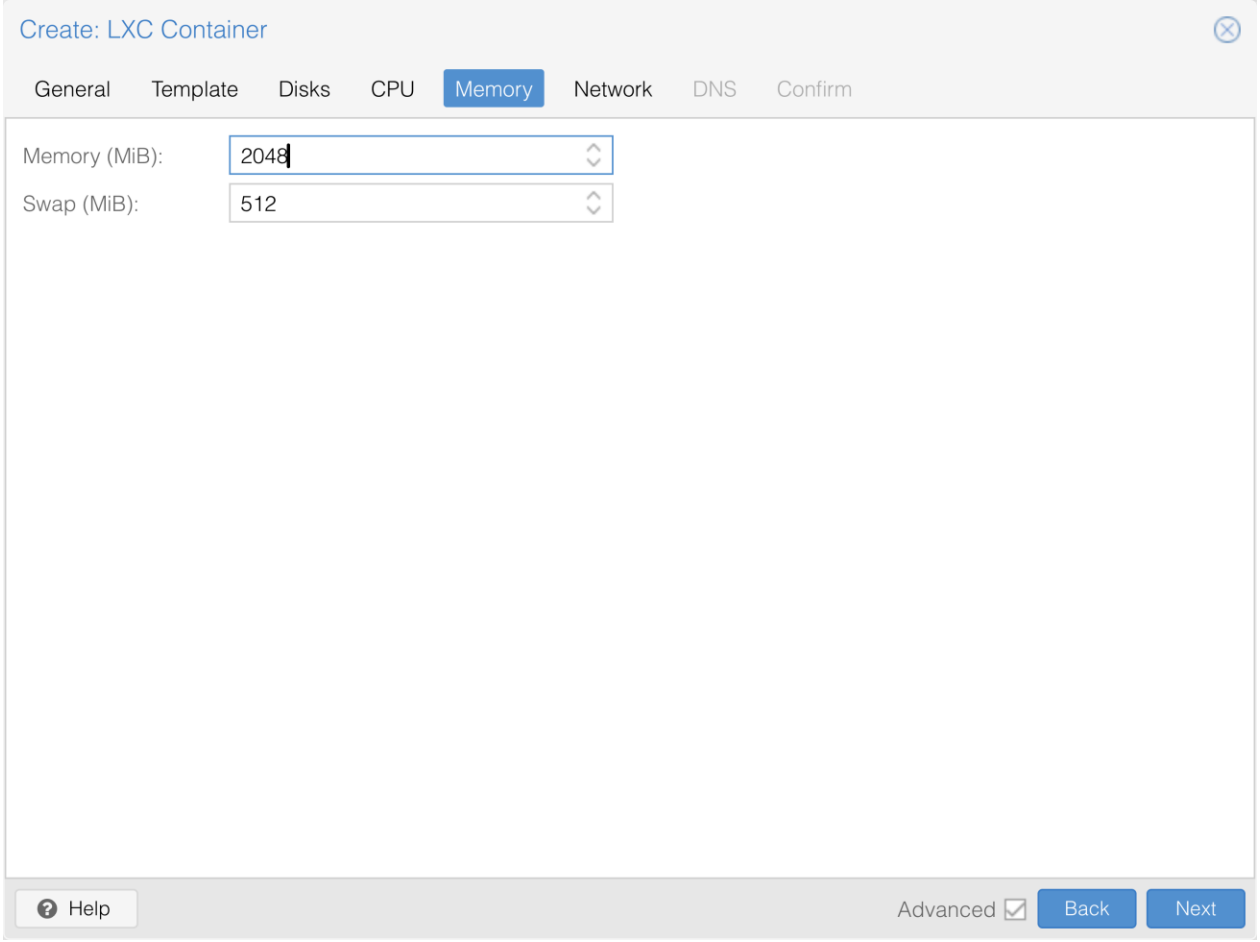
Sets a 2 GiB memory limit for this container. Size it for the services and leave headroom for the Proxmox host.

Swap: 512 MiB

Allows additional swap usage if host swap is available. Swap is slower than RAM and is not a performance upgrade.

Watch memory pressure

Check application use after startup. Memory exhaustion can trigger out-of-memory kills; increase limits or reduce the workload.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Network: separate bridge, address and gateway

eth0 on vmbr0

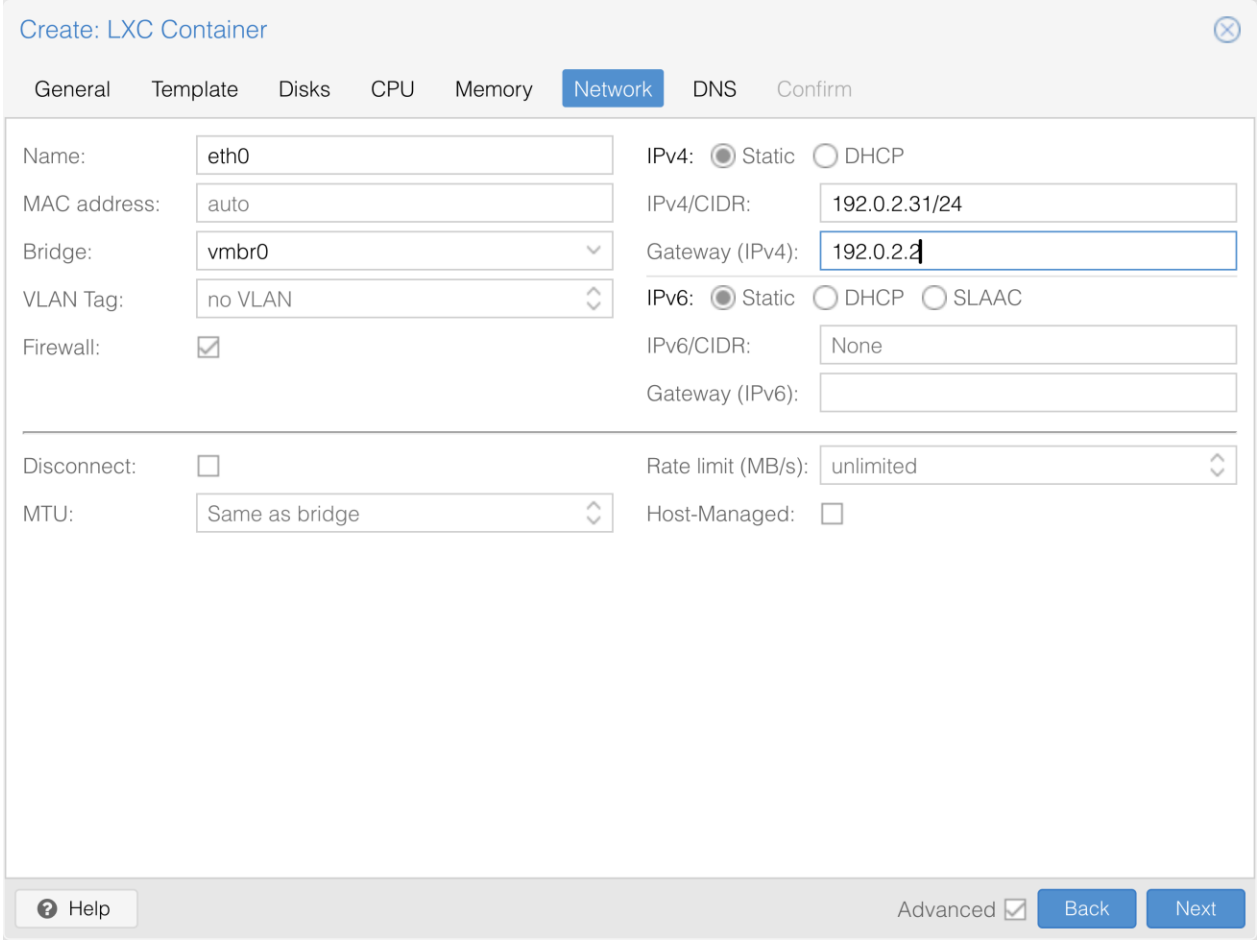
eth0 is the interface inside the container. vmbr0 is the host bridge; no VLAN tag is configured in this lab.

Static IPv4 and gateway

192.0.2.31/24 is the container address. 192.0.2.2 is the gateway used to reach destinations outside that subnet.

Firewall and lab addresses

The firewall flag is checked. These 192.0.2.x values are documentation/lab examples; use addresses valid for your own network.



Lab screenshot • Proxmox VE Administration Guide

initMAX

DNS: resolve names independently of routing

DNS server: 192.0.2.3

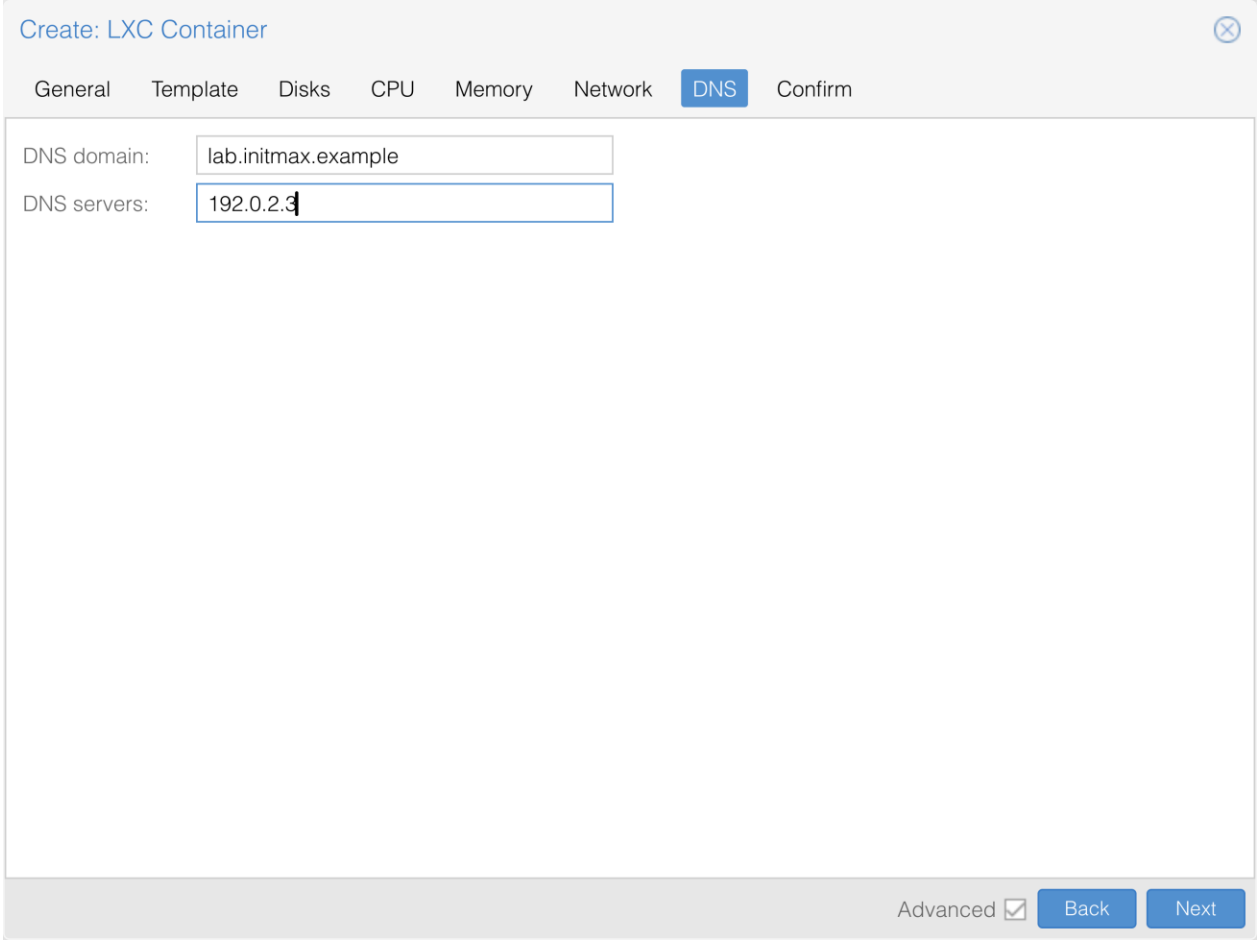
The resolver answers name lookups. It must be reachable from the container; a working gateway alone does not prove DNS works.

Search domain

lab.initmax.example is the suffix used for short-name lookups. It does not create DNS records for the container.

Validate both layers

Test IP connectivity first, then a hostname lookup. If IP works but names fail, check the resolver and domain settings.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Confirm: review the container before first start

Identity, isolation and template

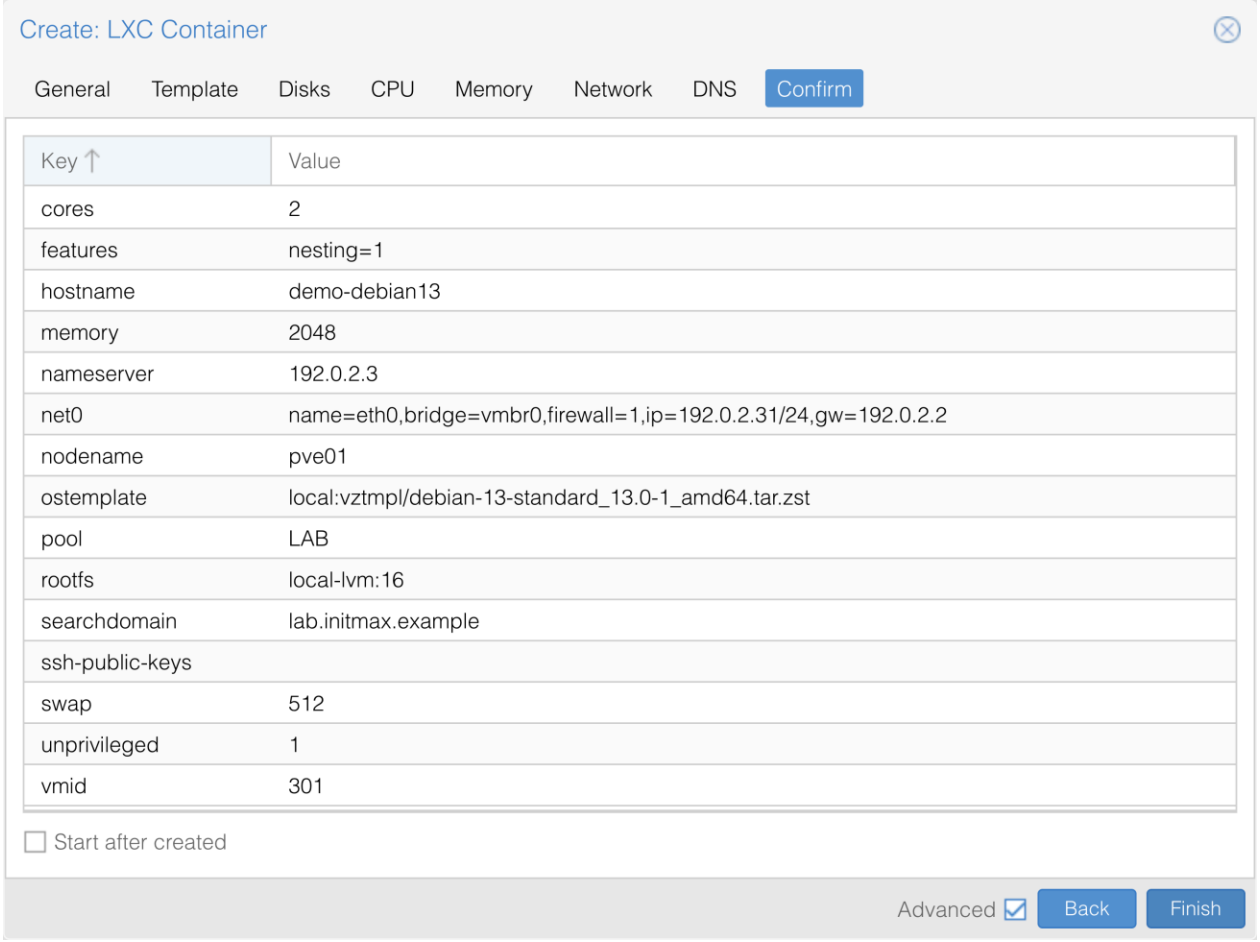
Verify CT 301, demo-debian13, pve01 and LAB. Check unprivileged=1, nesting=1 and the Debian 13 archive.

Resource and network review

Check 2 cores, 2048 MiB RAM, 512 MiB swap and 16 GiB rootfs. Verify IP, gateway and DNS against the lab plan.

Finish, start and test

Start after created is unchecked. Create the CT, start it and open the console; test IP connectivity and name resolution.



Lab screenshot • Proxmox VE Administration Guide

initMAX

Post-create: update, secure and validate the CT

1 Start and update Debian

Start CT 301 and open its console. Run apt update, then apt full-upgrade. Review package changes before confirming.

2 No VM guest tools in LXC

LXC shares the host kernel. Do not install VirtIO drivers or QEMU Guest Agent as a Proxmox integration step.

3 Configure secure access

Create an admin user and install only required packages. Configure SSH keys, then test access before restricting passwords.

4 Verify network and DNS

Check IP address and IP route. Test the gateway, DNS resolution with hosts, and access to the package repositories.

5 Check services and limits

Check systemctl --failed, time synchronization and disk space. Monitor memory, swap and service logs under expected load.

6 Back up and test recovery

Confirm firewall and boot policy. Create a backup and test restore. Review bind mounts separately: their data is not in CT backups.

Source: Proxmox VE Administration Guide • Linux Container

initMAX

Demo: create and validate the first LXC

Live demo route

1 Identify

Select CT ID, hostname and access settings.

2 Configure

Choose template, mount points and resources.

3 Review and create

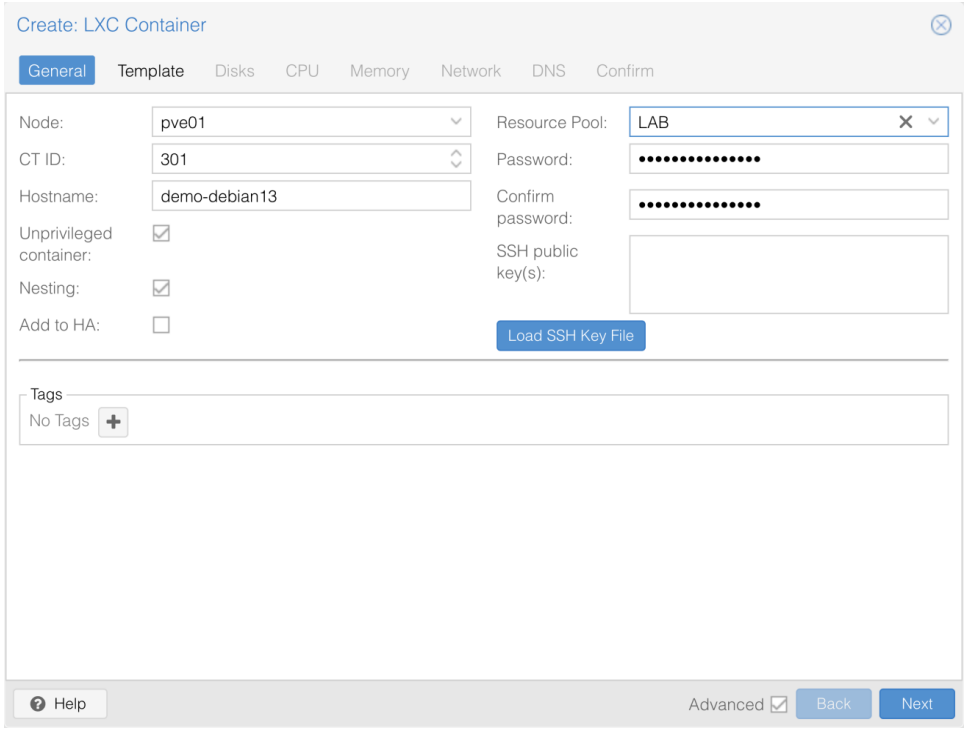
Check network/IP and DNS, then create CT.

4 Update and validate

Update packages; test DNS, services and backup.

DEMO

Switch to the live Proxmox VE interface



Show live: create → update → secure → validate

6

DEMO



7

Problems



Problems

Design mistakes that are expensive to fix

01 One network carries everything.
Storage recovery or migration can starve Corosync. Fixing it means network redesign.

02 Redundant links. One shared point of failure.
Both paths use the same switch or power feed. One failure can take all nodes down.

03 No capacity left when a node fails.
Surviving nodes cannot fit the VMs. Maintenance and failover become an outage.

04 Mixed CPUs with no migration baseline.
A VM using host CPU features may not live-migrate. Fixing it can require a shutdown.

05 HA is planned, but VM disks stay on one host.
Without shared storage or replication, host failure leaves no usable disks elsewhere.

06 Backups share the production failure domain.
The same storage, site or credentials can take out both production and recovery.

07 Snapshots become the backup strategy.
Retained changes consume space; losing the pool also loses its local snapshots.

8

WIKI



Wiki

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

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.

Accreditation is in progress and will be completed soon.

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

initMAX

Contact us

Implementation, support and training

Tomáš Heřmánek

tomas.hermanek@initmax.com

+420 732 447 184

Discuss your project with us →

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