

yoTech24

Praxis-Lab

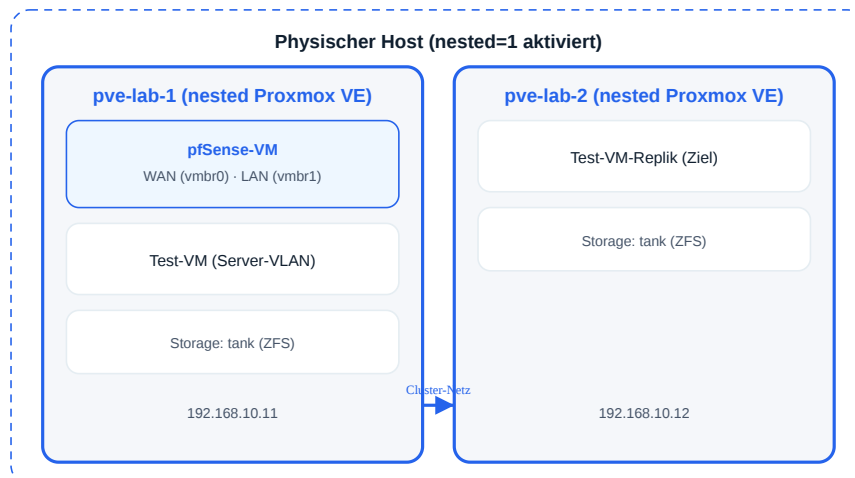
Proxmox-VE-Testlab mit 2 virtuellen Nodes · 14 Übungen (L0–L13)

Schritt-für-Schritt-Anleitung für ein Nested-Proxmox-Cluster: Installation, ZFS-Replikation, VLAN-Netz und pfSense-Firewall-Segmentierung – reine Lernumgebung, nicht für den Produktivbetrieb.

Diese PDF fasst die Kernpunkte zusammen. Die vollständigen interaktiven Folien inkl. Audio-Dialogen (Podcast) finden sich unter yotech24.com/akademie/

Testlab mit 2 virtuellen Nodes

Nested-Lab (Proxmox in Proxmox) zum Üben – 14 aufeinander aufbauende Übungen, reine Lernumgebung, nicht für den Produktivbetrieb



L0 – Systemanforderungen (physischer Host)

Prüfpunkt: Ausgabe zeigt Y.

CPU

Minimum 4 Threads mit VT-x/AMD-V, empfohlen 8+ Threads

RAM

Minimum 16 GB (eng), empfohlen 32 GB+ – Verteilung ca. 8 GB je Proxmox-Node, 2 GB pfSense, 2 GB Test-VMs

Storage

Minimum 128 GB SSD, empfohlen NVMe mit 256 GB+

Nested Virtualization

Muss aktiv sein (nested=1) – ohne das startet innerhalb der Node-VMs keine eigene Virtualisierung

Ziel: Nested Virtualization auf dem physischen Host aktivieren, einmalig.

L1 – ISO-Download



Prüfpunkt: Beide ISOs sind unter `local` sichtbar.

- Proxmox-VE-ISO von `proxmox.com` herunterladen (aktuelle Version)
- pfSense-CE-ISO von `pfsense.org` bzw. Netgate herunterladen (Alternative: OPNsense von `opnsense.org`)
- ISOs auf den physischen Host laden: **Datacenter** → **Storage (local)** → **ISO Images** → **Upload**

Ziel: beide benötigten Installations-Images auf dem physischen Host bereitlegen.

L2 – VM-Vorbereitung (2 Proxmox-Node-VMs)

Achtung: Ohne CPU-Typ `host` startet die verschachtelte Virtualisierung in der Node-VM nicht. Bei Bedarf nachträglich: `qm set <id> --cpu host`

Prüfpunkt: Beide VMs starten vom Proxmox-ISO.

CPU-Typ

`host`

Pflicht für Nested Virtualization, 2–4 Kerne

RAM

8 GB

je Node-VM

Disk 1 (System)

32 GB

VirtIO-SCSI

Disk 2 (Daten/ZFS)

32 GB

VirtIO-SCSI, wird später zum Replikations-Pool `tank`

NIC

VirtIO

an `vmbr0`

BIOS

UEFI (OVMF) oder SeaBIOS – bei beiden Nodes gleich wählen

Für **jede** der zwei Node-VMs (`pve-lab-1`, `pve-lab-2`) identisch:

L3 – Installation

Prüfpunkt: Beide Weboberflächen sind erreichbar unter `https://<IP>:8006`.

- Proxmox-Installer auf beiden Nodes starten, Zielplatte: **Disk 1** (System)
- Hostname setzen: `pve-lab-1.lab.local` bzw. `pve-lab-2.lab.local`
- Feste IPs im gleichen Subnetz vergeben (z. B. 192.168.10.11 / .12)
- Nach der Installation: Enterprise-Repo deaktivieren, No-Subscription-Repo aktivieren (siehe Modul 4), dann Pakete aktualisieren

Ziel: Proxmox VE auf beiden Node-VMs installieren und aktualisieren.

L4 – Filesystem-Auswahl

ZFS & RAID-Kompatibilität: ZFS will direkten Zugriff auf die Platten, kein Hardware-RAID darunter (siehe Modul 3, „Fallstrick: RAID-Controller“) – auf echter Hardware den Controller in den HBA-/IT-Modus versetzen. In diesem Nested-Lab sind die Node-Disks ohnehin virtuelle VirtIO-SCSI-Platten ohne Hardware-RAID, das Prinzip gilt aber unverändert für den späteren Betrieb auf echten Servern.

Fallstrick: Den Pool auf **beiden** Nodes exakt gleich benennen (`tank`) – bei unterschiedlichen Namen scheitert die Replikation in L8 mit einem Storage-Fehler.

Prüfpunkt: Storage `tank` (ZFS) ist auf beiden Nodes vorhanden, gleicher Name.

- System-Disk (Disk 1): ext4/LVM oder ZFS – eine einzelne Platte ohne Redundanz reicht im Lab völlig aus, ein RAID-Verbund ist für die System-Disk nicht nötig
- **Daten-Disk 2: ZFS-Pool anlegen** – Pflicht, weil Replikation ZFS voraussetzt

Ziel: Storage-Grundlage für die spätere Replikation schaffen.

L5 – Node-Konfiguration

Prüfpunkt: `ping pve-lab-2` von Node 1 erfolgreich (und umgekehrt).

- Zeitsync (NTP) auf beiden Nodes prüfen – im Cluster kritisch (siehe Modul 4, „Zeitsynchronisation“)
- Namensauflösung über `/etc/hosts` auf beiden Nodes mit beiden Einträgen ergänzen

Ziel: Zeitsync und Namensauflösung vor dem Cluster-Aufbau absichern.

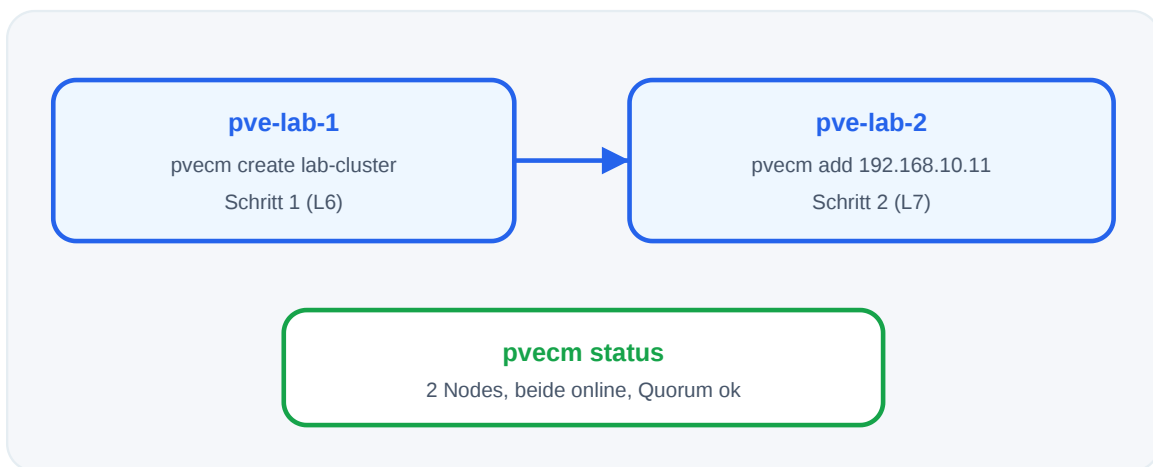
Danach gegenseitig anpingbar prüfen.

L6 – Cluster erstellen (auf Node 1)

Prüfpunkt: `pvecm status` zeigt 1 Node, Quorum ok.

Ziel: den Cluster initial anlegen.

L7 – Cluster beitreten (auf Node 2)



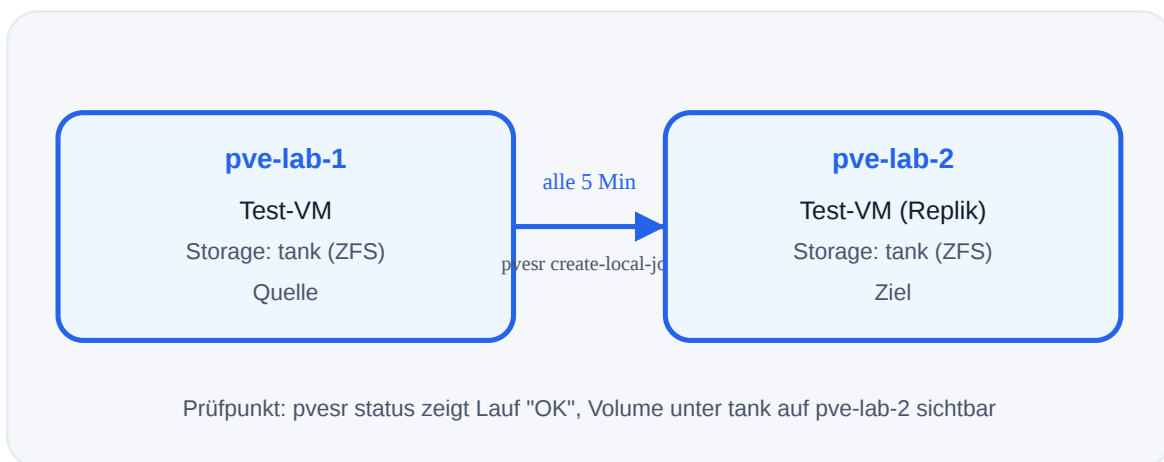
Hinweis: Für echte Hochverfügbarkeit fehlt bei zwei Nodes die dritte Stimme – optional eine kleine QDevice-VM ergänzen (siehe Modul 7).

Prüfpunkt: `pvecm status` zeigt 2 Nodes, beide online.

Ziel: den zweiten Node dem Cluster hinzufügen.

Danach in der GUI: beide Nodes im Datacenter-Baum sichtbar.

L8 – Replikation einrichten

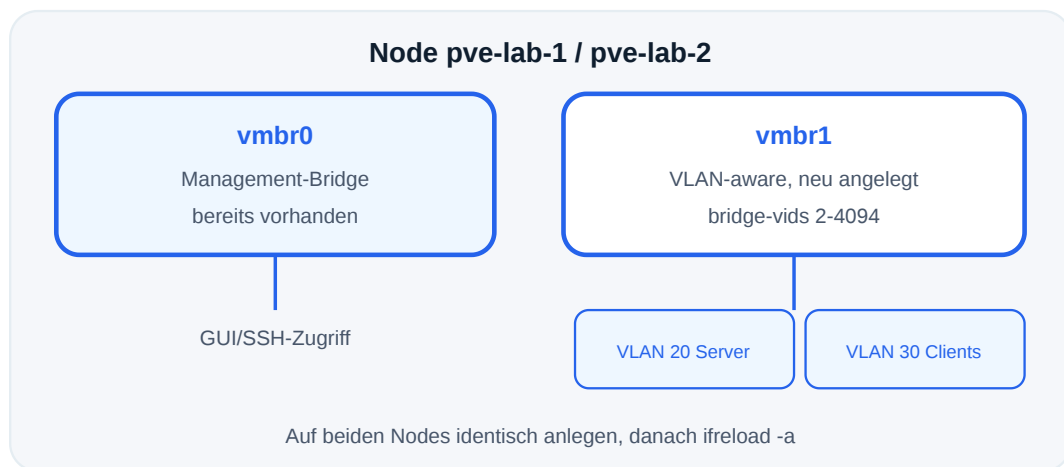


Prüfpunkt: Erster Replikationslauf zeigt OK, das VM-Volume ist auf Node 2 unter tank sichtbar.

- Test-VM auf tank (ZFS) auf Node 1 anlegen
- In der GUI: **VM** → **Replication** → **Add**, Ziel pve-lab-2, Intervall z. B. */5 (alle 5 Minuten)

Ziel: eine Test-VM per ZFS-Replikation von Node 1 auf Node 2 absichern.

L9 – Netzwerk-Konfiguration



Prüfpunkt: vmbr1 ist als VLAN-aware Bridge auf beiden Nodes aktiv.

- **vmbr0** bleibt die Management-Bridge (schon vorhanden)
- Zusätzliche Bridge **vmbr1** für das Lab-Netz/VLANs anlegen, **VLAN aware** aktivieren

Ziel: eine zweite, VLAN-fähige Bridge für das Lab-Netz anlegen.

L10 – Nachträgliche Anpassung: VLAN, Disk, CPU, RAM

ÄNDERUNG	LIVE MÖGLICH?
VLAN-Tag an VM-NIC setzen	✓ ja
vDisk vergrößern	✓ ja (+ Gast-FS erweitern)
Neue Disk hinzufügen	✓ ja (VirtIO-SCSI-Hotplug)
CPU erhöhen	⚠ nur mit Hotplug + Gast
RAM erhöhen	⚠ nur mit Memory-Hotplug
CPU / RAM verkleinern	✗ nein, Shutdown nötig

Grün: sofort live · Gelb: an Bedingungen geknüpft · Rot: immer Shutdown

Prüfpunkt: vDisk live vergrößert und im Gast nutzbar; VLAN-Tag greift sofort.

- VLAN an VM-NIC: live, über **VM** → **Hardware** → **Netzwerk** → **VLAN-Tag setzen**
- vDisk vergrößern: live auf Hypervisor-Ebene per `qm resize <id> scsi0 +8G`; danach im Gast Partition und Dateisystem erweitern (Windows: Datenträgerverwaltung; Linux: `growpart` + `resize2fs/xfs_growfs`)
- Disk hinzufügen: live, über VirtIO-SCSI-Hotplug
- CPU erhöhen: nur mit Hotplug und passender Gast-Unterstützung, sonst Shutdown nötig (siehe Modul 5)
- RAM erhöhen: nur mit Memory-Hotplug, sonst Shutdown nötig
- CPU/RAM verkleinern: grundsätzlich nicht live, immer Shutdown nötig

L11 – pfSense-Firewall hinzufügen

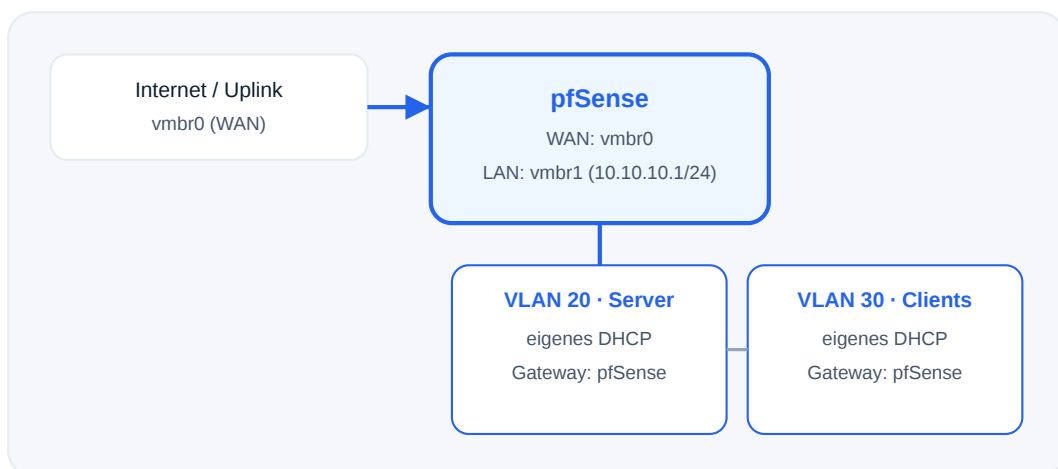


Prüfpunkt: pfSense-Web-GUI ist über LAN erreichbar, WAN hat Verbindung.

- Neue VM aus der pfSense-ISO anlegen, mit zwei NICs
- **WAN** → `vmbr0` (Management/extern)
- **LAN** → `vmbr1` (Lab-Netz, VLAN-fähig)
- pfSense installieren, WAN- und LAN-IP setzen (LAN z. B. 10.10.10.1/24)

Ziel: eine Firewall-VM zwischen Management- und Lab-Netz schalten.

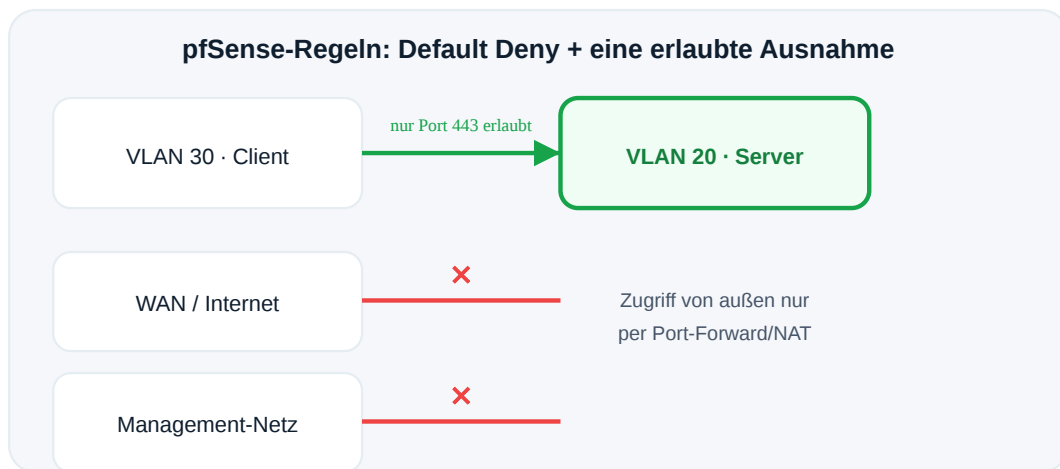
L12 – VLANs erstellen



Prüfpunkt: Eine VM im VLAN 20 bekommt eine IP aus dem Server-Netz, Gateway ist pfSense.

- In pfSense VLANs auf dem LAN-Interface definieren (z. B. VLAN 20 = Server, VLAN 30 = Clients)
- Je VLAN ein eigenes Interface samt DHCP und Gateway in pfSense anlegen
- An der jeweiligen VM-NIC in Proxmox das passende VLAN-Tag setzen (auf `vmbr1`)

L13 – Server nur hinter Firewall erreichbar



Prüfpunkt: Der Server ist nur über die freigegebene Firewall-Regel erreichbar; direkter Zugriff aus einem anderen Segment wird geblockt.

- Test-Server-VM liegt ausschließlich im **VLAN 20** (Server-Netz), kein Bein im Management-Netz
- pfSense-Regeln: Client-VLAN → Server-VLAN nur für benötigte Ports erlauben (z. B. 443), alles andere per Default Deny blocken
- Kein direkter Weg vom WAN oder Management-Netz zum Server, außer über eine explizit definierte Regel
- Zugriff von außen ausschließlich per gezieltem Port-Forward/NAT in pfSense

Abschluss & Aufräumen

Dieses Lab lässt sich beliebig oft zurücksetzen und wiederholen – genau dafür ist es als reine Nested-Umgebung gedacht, nicht für den Produktivbetrieb.

- Vor riskanten Schritten jeweils einen Snapshot der betroffenen VM anlegen, um das Lab schnell zurücksetzen zu können
- Beide Node-VMs als Proxmox-Templates sichern, damit sich das komplette Lab reproduzierbar neu aufsetzen lässt
- Als Erweiterung: einen dritten Node oder ein QDevice ergänzen und einen echten HA-Failover üben (siehe Modul 7 und 8)