Training
Proxmox Virtual Environment



© inett GmbH / Proxmox Server Solutions GmbH
Stand 31.08.2026

Alle Rechte vorbehalten. Nachdruck, Vervielfältigung und Veröffentlichung nicht gestattet.

inett GmbH -- Altenkesseler Str. 17 / B8 -- 66115 Saarbrücken
+49 681 410 993-0 -- info@inett.de -- www.inett.de

Willkommen

Proxmox VE

Clustering und verteilter Storage

Training



© inett GmbH -- info@inett.de -- www.inett.de

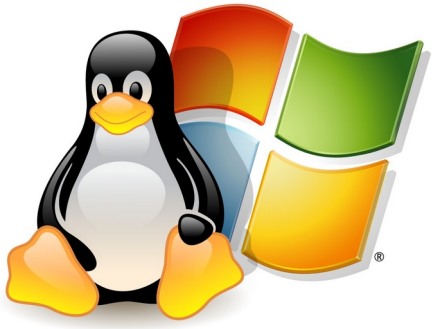
Über den Trainer

- Guillaume Delanoy
- Linux seit ~1999 (Kernel 2.0 ...)
- Proxmox VE seit ~2015
- Linux Trainer seit 2017
- Linux & Open Source orientiert
- Linux Consultant
- https://www.linkedin.com/in/guillaume-delanoy-2757ab98/





Vorstellung inett GmbH



- Gründung 2007 in Saarbrücken
- Erweiterung 2023 in Schengen/Luxemburg
- 25+ Mitarbeitende
- 2009 mit Proxmox gestartet
- 24/7 Support für Proxmox, Ceph, OPNsense, Linux, Open Source
- Proxmox, Ceph, OPNsense, Checkmk, Ansible, ...
- Consulting, Training, Managed Services, Projekte
  in DE, AT, CH, LU, NL, SW, PL, UK, BE, ...

© inett GmbH -- info@inett.de -- www.inett.de

Vorstellung Training Partner



© OpenStreetMap Contributors - https://www.openstreetmap.org/node/5989105999

Trainingsüberblick



Training Zeitplan



- 09:00 – Start
- 10:30 – Kurze Pause
- 12:30 – Mittagspause
- 15:00 – Kurze Pause
- 17:00 – Ende

©inett GmbH -- info@inett.de -- www.inett.de

Agenda des Trainings:

- Willkommen
- Trainingsüberblick
- Einstieg in die Trainingsumgebung
- Proxmox VE Cluster
- Installation der PVE Cluster Nodes
- HCI: Hyper-Converged Infrastructure
- HCI mit PVE und Ceph Storage Cluster
- Proxmox VE Guest Migration

Agenda des Trainings:

- PVE Open-Source Storage Replication Framework (ZFS)
- PVE High Availability
- PVE Benutzerverwaltung
- PVE REST API
- PVE Firewall
- Verwaltung der Zertifikate
- Durchreichen: PCI(e) Passthrough
- Optionale Themen PVE clustering & shared storage

Ziele dieses Trainings

- Konzepte und Features von Proxmox VE und Ceph Clustern verstehen
- Wissen im Bereich Clusterbau vertiefen
- Fragen & Antworten aus der Praxis
- Ein nettes Proxmox Zertifikat erhalten
- "Training Plus“ Promotion (20%)

© inett GmbH -- info@inett.de -- www.inett.de

Vorstellungsrunde

[Alle Angaben natürlich freiwillig]

- Informationen zum Teilnehmer
- Informationen zum Unternehmen
- Wird Proxmox VE bereits eingesetzt oder ist ein Einsatz in Planung?
- Ist bereits Erfahrung mit Proxmox VE vorhanden?
- Themenwünsche - gibt es spezielle Themen, die im Training behandelt werden sollten?
- Ziel des Trainings (Wissen aneignen, Wissen vertiefen, konkrete Aufgabenstellungen)?
- Weitere Anmerkungen?

inett

Einstieg in die Trainingsumgebung



© inett GmbH -- info@inett.de -- www.inett.de

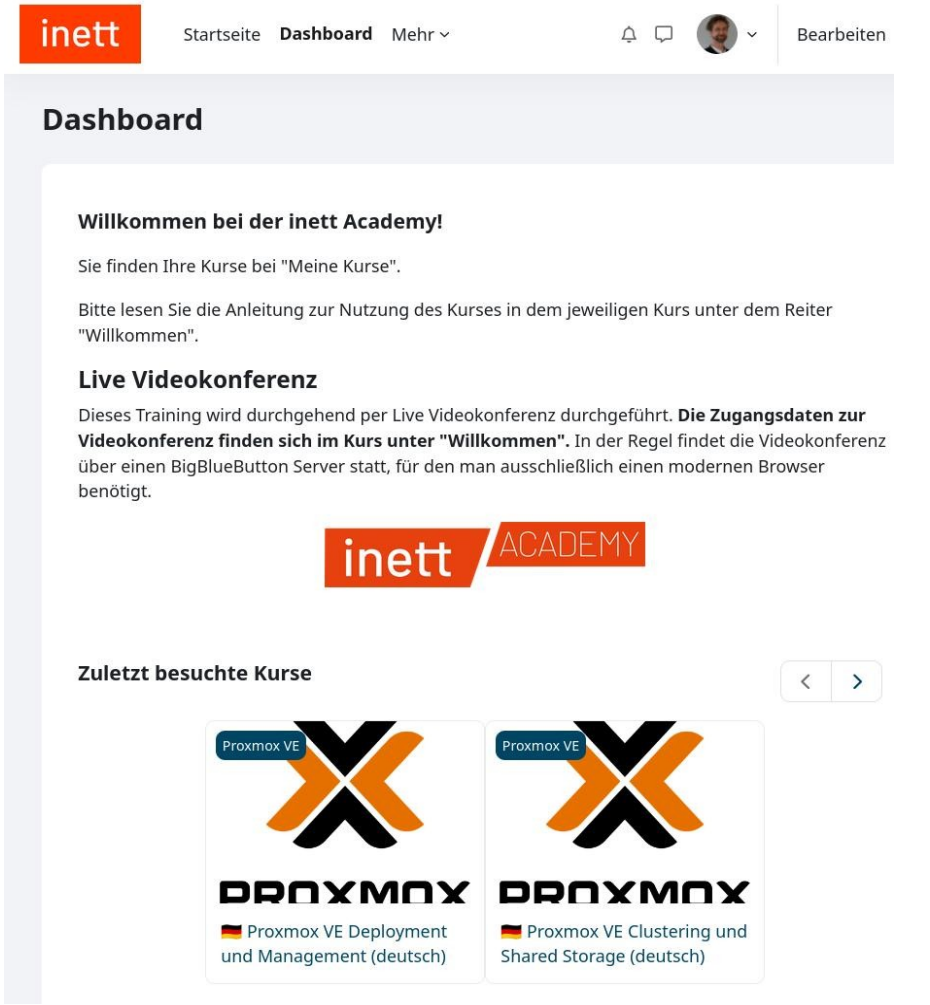
Kursumgebung

- Unter der Website

  https://training.inett.de/lms/

  befindet sich die inett Kursumgebung mit allen Lerninhalten

- Die Inhalte besprechen wir im Kurs nacheinander



© inett GmbH -- info@inett.de -- www.inett.de

Live Videokonferenz bei Online Trainings

- Begleitende Live Videokonferenz über BigBlueButton oder Teams
- Zugangsdaten sind in der Kursumgebung verlinkt
- Jedem Teilnehmer steht eine komplette Trainingsumgebung zur Verfügung. Diese Umgebung besteht aus:
  - 1 Linux Desktop
  - 4 Proxmox Virtual Environment Server (PVE)
  - 1 Proxmox Backup Server

© inett GmbH -- info@inett.de -- www.inett.de

Bedienung der Trainingsumgebung



Zugang für die Dauer des Trainings mit den folgenden Logindaten möglich:

- *Guacamole Login:*

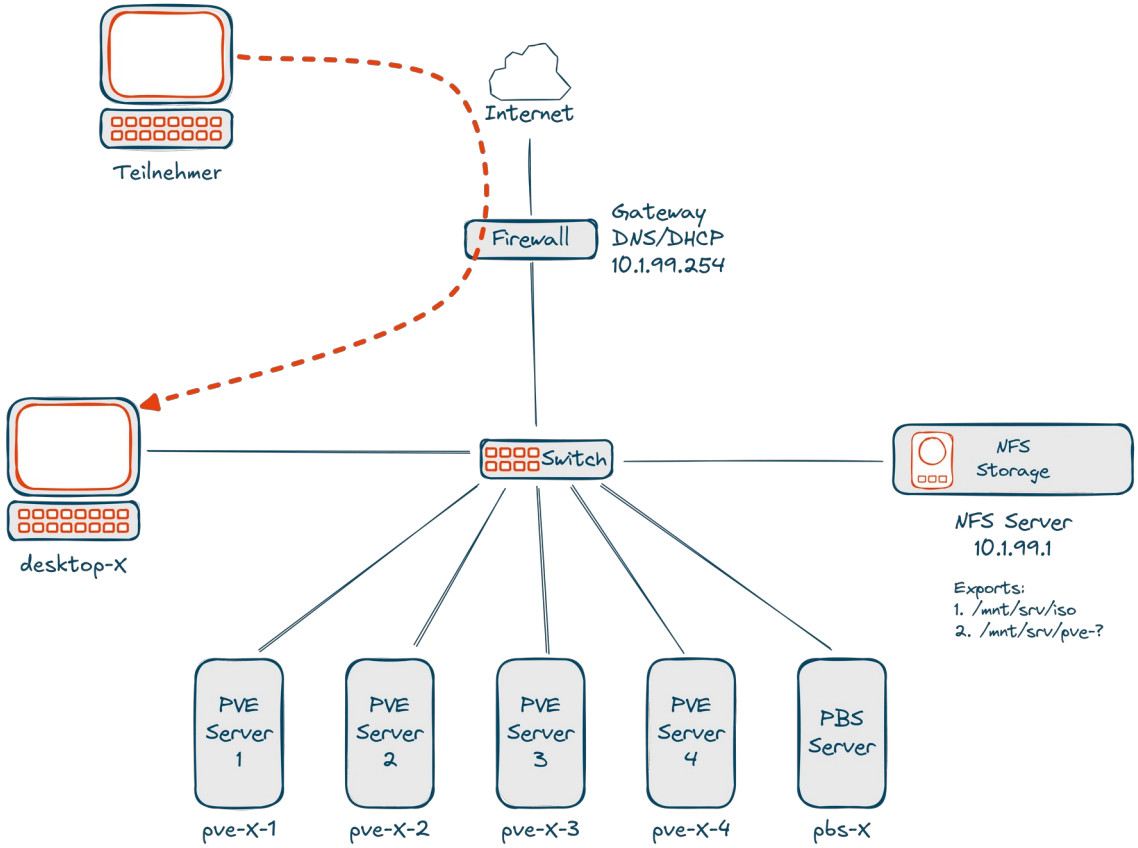
Guacamole Username	**pve-X**
Guacamole Password	**inett**

- *Linux Desktop Login:*

Linux Desktop Username	**inett**
Linux Desktop Password	**inett**

© inett GmbH -- info@inett.de -- www.inett.de

Bedienung der Trainingsumgebung



© inett GmbH -- info@inett.de -- www.inett.de

Das Unternehmen Proxmox



Proxmox Server Solutions GmbH

- Gegründet 2005 in Wien, Österreich (Europäisches Unternehmen)
- Produkte:

- **Proxmox Virtual Environment**
  Hypervisor (KVM / Qemu and LXC Guests)
- **Proxmox Backup Server**
  Backup for hosts and guests
- **Proxmox Offline Mirror**
  Offline repository for air-gapped hosts
- **Proxmox Datacenter Manager**
  Manage multiple clusters from a single dashboard
- **Proxmox Mail Gateway**
  Email security



© inett GmbH -- info@inett.de -- www.inett.de

Proxmox Server Solutions GmbH

- Aktive Community
- Enterprise Support & Training
- Weltweites Partnernetzwerk
- Proxmox entwickelt Open Source Software zur Vereinfachung des Server Managements
- In privater Hand, unabhängig von Venture Capital Investoren



Enterprise support & training



Global partner ecosystem



Vibrant Community



Proxmox Mail Gateway [2005]



Proxmox Virtual Environment [2008]



Proxmox Backup Server [2020]

Das Proxmox Geschäftsmodell

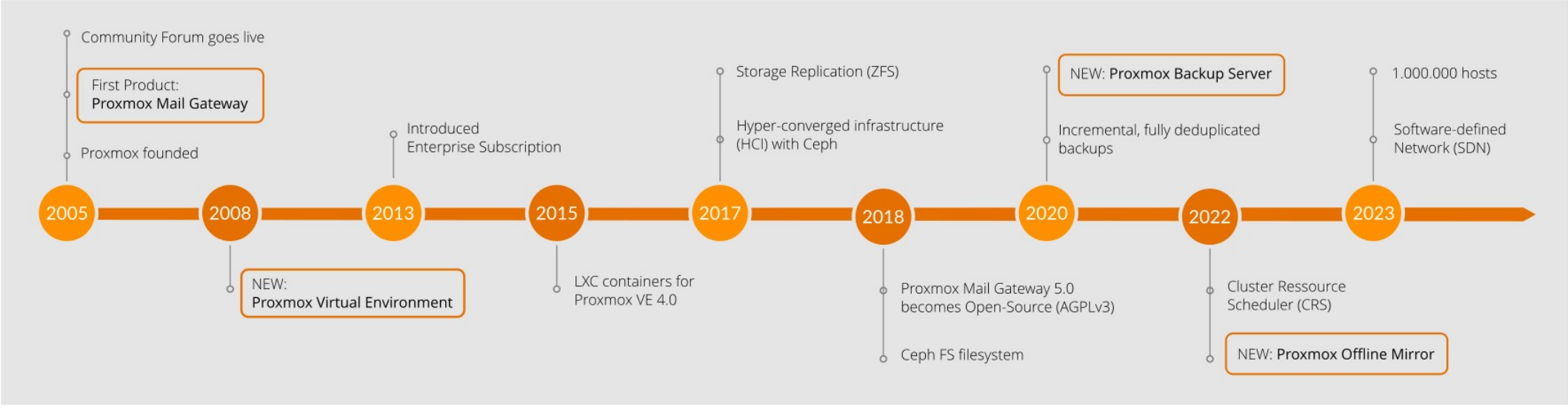
- Proxmox VE ist freie Open-Source Software
- Lizensiert unter AGPLv3
- Im Gegensatz zu „*commercial open-source*“ oder „*open core*“ ist der Quellcode sowie alle Funktionen zu 100% verfügbar
- Finanzierung über Subscriptions und Trainings helfen, das Produkt weiter zu entwickeln

Das Proxmox Entwicklungsmodell



- Open Source *und* frei
- „It’s free as in free speech, not as in free beer“
- Entwicklung ist für jeden öffentlich einsehbar
- Jeder kann sich beteiligen in Form von Entwickeln, testen, dokumentieren, Benutzerhilfe, Ideen…

Proxmox Server Solutions GmbH
Meilensteine



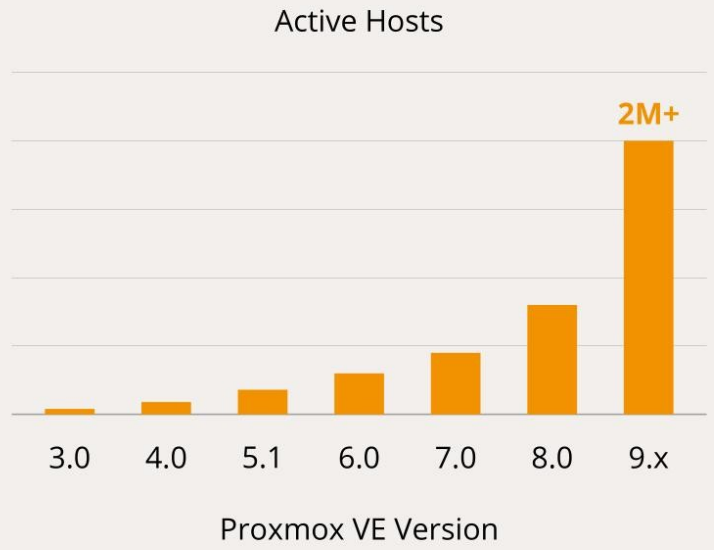
Proxmox VE Facts

Omnipresent
7 continents & in space

Enterprises of all sizes and industries

2,600 Partners
Local expertise and support

Over
2 Million



© inett GmbH -- info@inett.de -- www.inett.de

Proxmox VE Cluster



Cluster Basics: Quorum

- Ein Quorum ist eine *Abstimmung* im Cluster, um die Handlungsfähigkeit zu bestimmen
- Benötigt die Mehrzahl der Stimmen / Nodes eines Clusters
- Mehr als die Hälfte aller Cluster Nodes muss online sein *und* kommunizieren
- Nodes/2 + 1 = Quorum erreicht
- Entscheidungskriterium für HA Service Relocation und Fencing

Beispiele:

3 Node Cluster: Quorum ist 2

4 Node Cluster: Quorum ist 3

5 Node Cluster: Quorum ist 3

6 Node Cluster: Quorum ist 4

7 Node Cluster: Quorum ist 4

PVE Cluster: Features

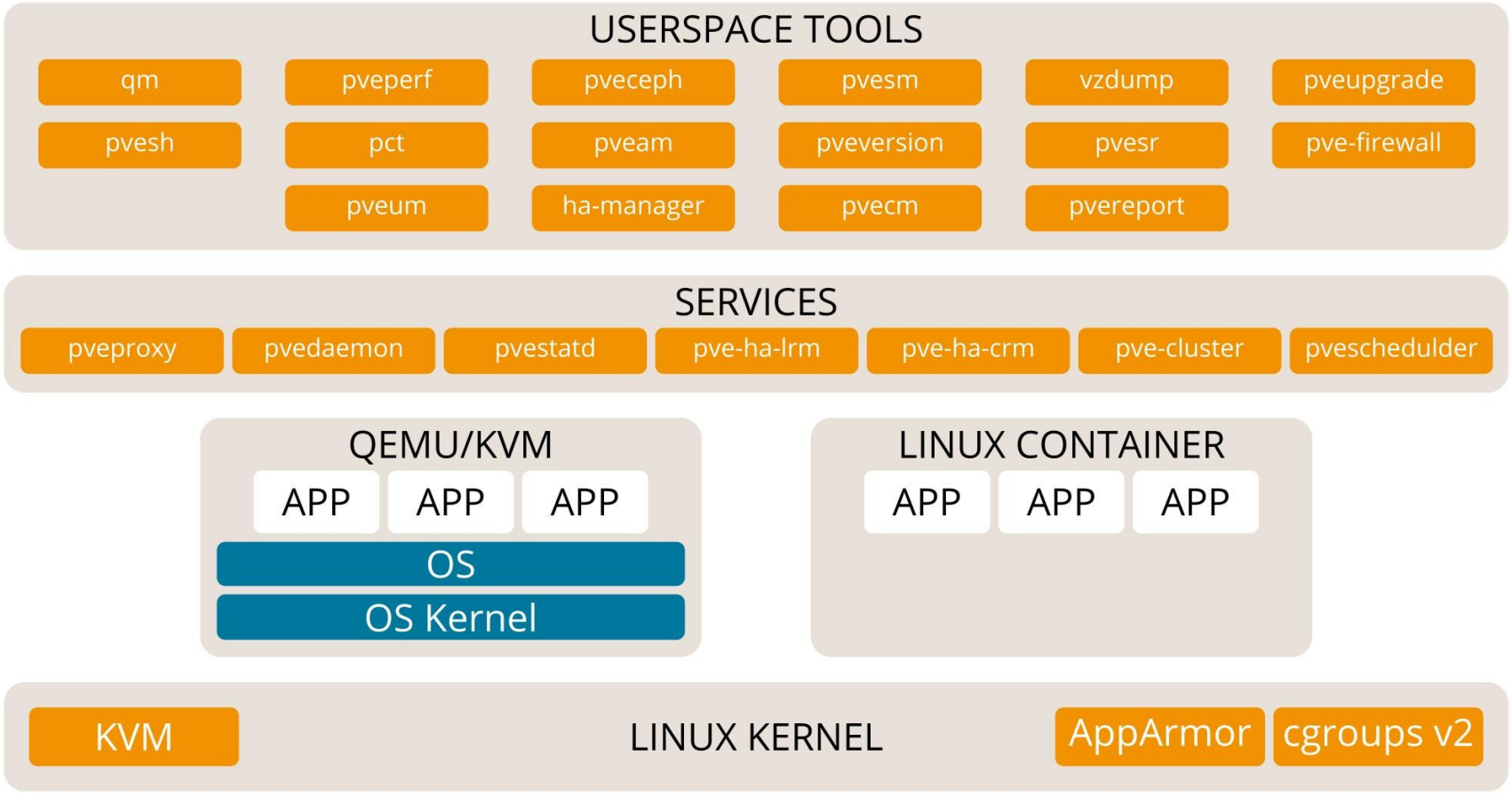
- Zentrale Verwaltung über Web GUI, API, Shell
- Authentifizierung gegen PAM, Proxmox VE Authentication Server, Active Directory, LDAP und Auth + SSO mit OpenID Connect
- Zentrales, rollenbasiertes Rechtemanagement (RBAC) aller Objekte (VMs, Container, Storages, Nodes)
- Multi Master Cluster (kein Single Point of Failure)
- Proxmox Cluster File System (pmxcfs): Filesystem für Konfiguration, clusterweit repliziert in Echtzeit
- Migration virtueller Maschinen und Container zwischen physikalischen Hosts
- Clusterweite Tasks und Logs

PVE Cluster: Voraussetzungen

- Zuverlässiges Cluster Netzwerk (dediziert, redundant)
- Latenz zwischen Nodes unter 5ms
- Funktionierende Namensauflösung (via /etc/hosts)
- Mehrere Server (Proxmox testet bis 32 Nodes)
- Datum und Zeit müssen sychron sein (NTP)

©inett GmbH -- info@inett.de -- www.inett.de

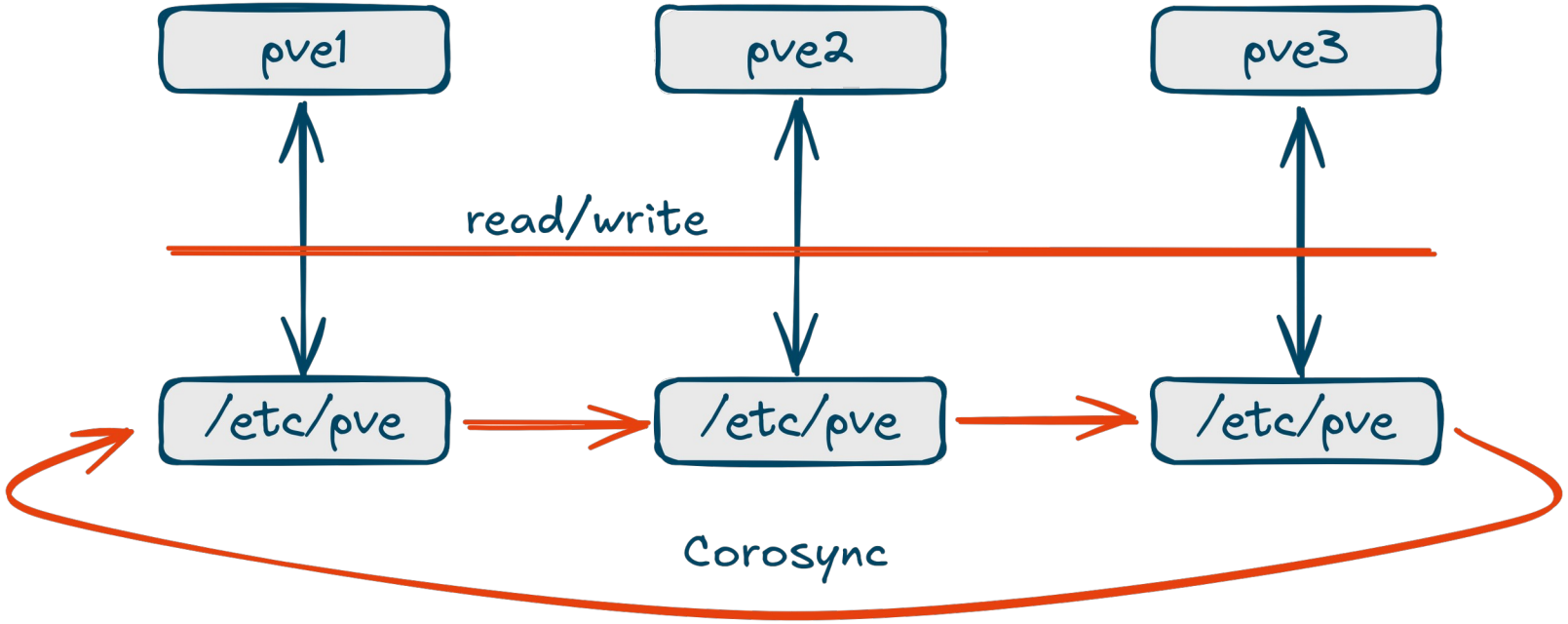
Proxmox Management Tools + Komponenten



PVE Cluster Dienste : pve-cluster

- Kernstück einer Proxmox VE Installation
- Stellt das Proxmox Cluster Filesystem bereit (pmxcfs)
- Stellt die Konfigurationsdaten als normales Filesystem bereit, eingehängt unter /etc/pve (über FUSE)
- sämtliche PVE bezogene Konfiguration ist darunter verfügbar
- Wird auch bereits für eine einzelne Node (ohne Cluster) benötigt
- Locking von Ressourcen (Standalone und Cluster)
- Verteilt statistische Daten im Cluster

Proxmox Cluster Filesystem Sync

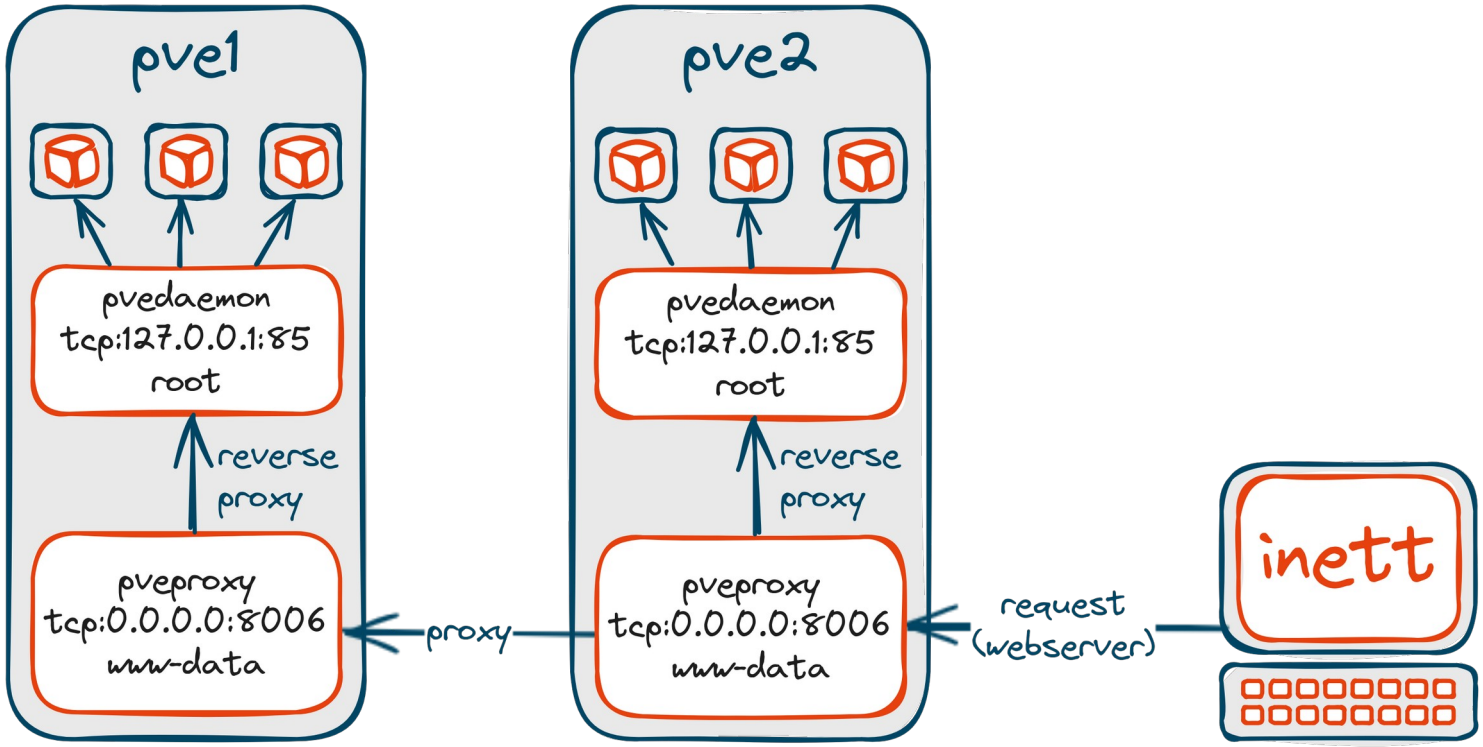


PVE Cluster Dienste : pvedaemon & pveproxy

- **pvedaemon**
  - REST API Server
  - Führt API Aufrufe mit root Rechten aus
  - ist nur über lokalen Port verfügbar (tcp:127.0.0.1:85)
  - Anfragen kommen über pveproxy

- **pveproxy**
  - REST API Proxy Server
  - Gebunden an TCP Port 8006 (tcp:0.0.0.0:8006)
  - Läuft unter www-data (nicht root)
  - leitet Anfragen entweder an pvedaemon oder zu pveproxy auf anderen Cluster Nodes weiter.
  - Anfragen, die keine root Berechtigung benötigen, werden direkt von pveproxy beantwortet

PVE cluster Dienste : pvedaemon & pveproxy



© inett GmbH -- info@inett.de -- www.inett.de

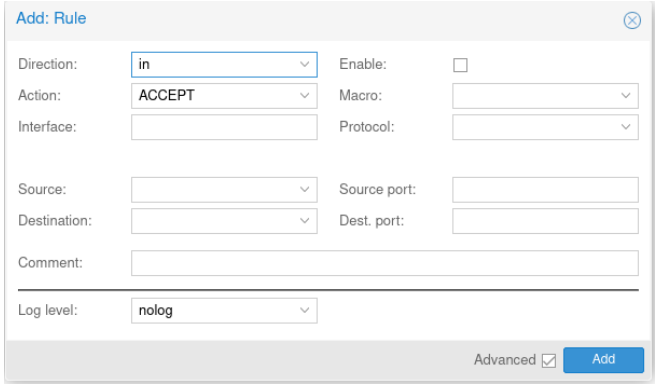
PVE Cluster Dienste: pvestatd

- PVE Status Daemon
- Monitort Status aller Ressourcen (VMs, Container, Storage)
- Sendet Status an alle anderen Nodes
- Letzter Status auch verfügbar unter `/etc/pve/.rrd`
- Auto Ballooning / dyn. Speichermanagement für QEMU/KVM VMs

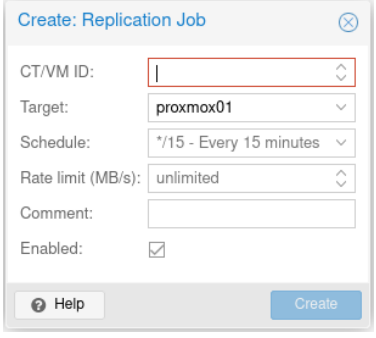
© inett GmbH -- info@inett.de -- www.inett.de

PVE Dienste: pve-firewall, pvesr

- pve-firewall
  - Zone Firewall
  - Objekt-orientierte Konfiguration
  - Standardmäßig *deaktiviert*



- pvesr
  - PVE Storage Replication
  - Spezifisch für ZFS Pools



© inett GmbH -- info@inett.de -- www.inett.de

Linux Cluster Dienst

- *Corosync*
  - Cluster Engine
  - Service Group management
  - Stellt den Quorum bereit
  - Connection Manager für Cluster Membership Management, Messaging, Event Notification, Quorum, State Transitions

Linux Standard Dienste

- pvescheduler (cron bis PVE 7.0)
  - zeitlich geplante Aufgaben → Backup, Updates
- systemd/journalctl syslog
  - Log handling mit journalctl
- Postfix MTA
  - Mail Server (MTA, smtp) → Backup, Update Benachrichtigungen
- NTP → Chrony seit PVE 7.0 (systemd-timesyncd bis 6.4)
  - Synchronisiert Systemzeit (ntp) → Zeitsync



© inett GmbH -- info@inett.de -- www.inett.de

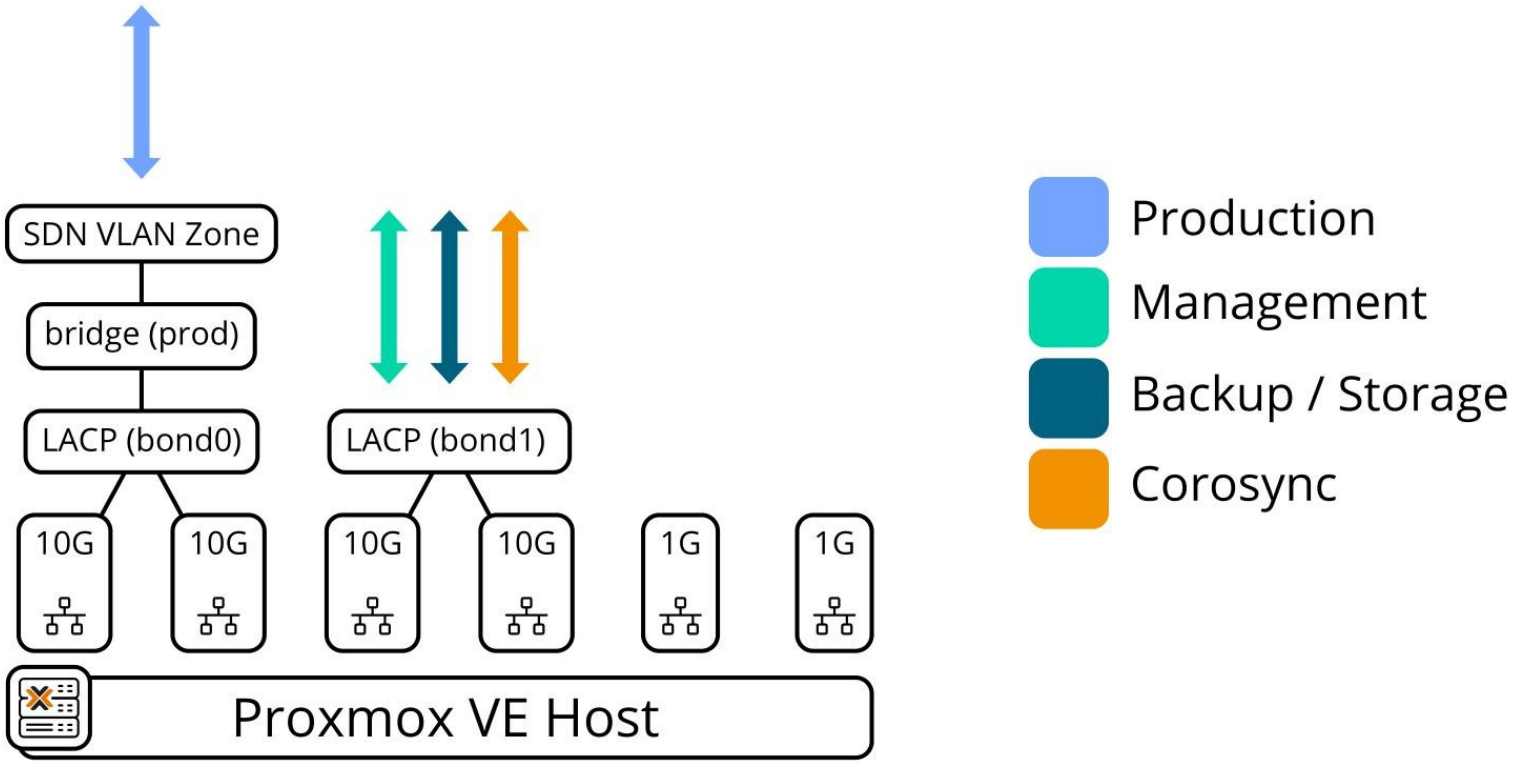
Chrony (NTP)

- Auf isolierten Umgebungen **unbedingt** prüfen
- Check mit:
  - # chronyc tracking
  - # chronyc sources
- Alternative 1: Chrony auf internen NTP konfigurieren
- Alternative 2: Chrony auf einer Node als Server betreiben, andere Nodes als Clients anbinden
- Konfiguration in `/etc/chrony/chrony.conf`
  - https://www.redhat.com/sysadmin/chrony-time-services-linux



© inett GmbH -- info@inett.de -- www.inett.de

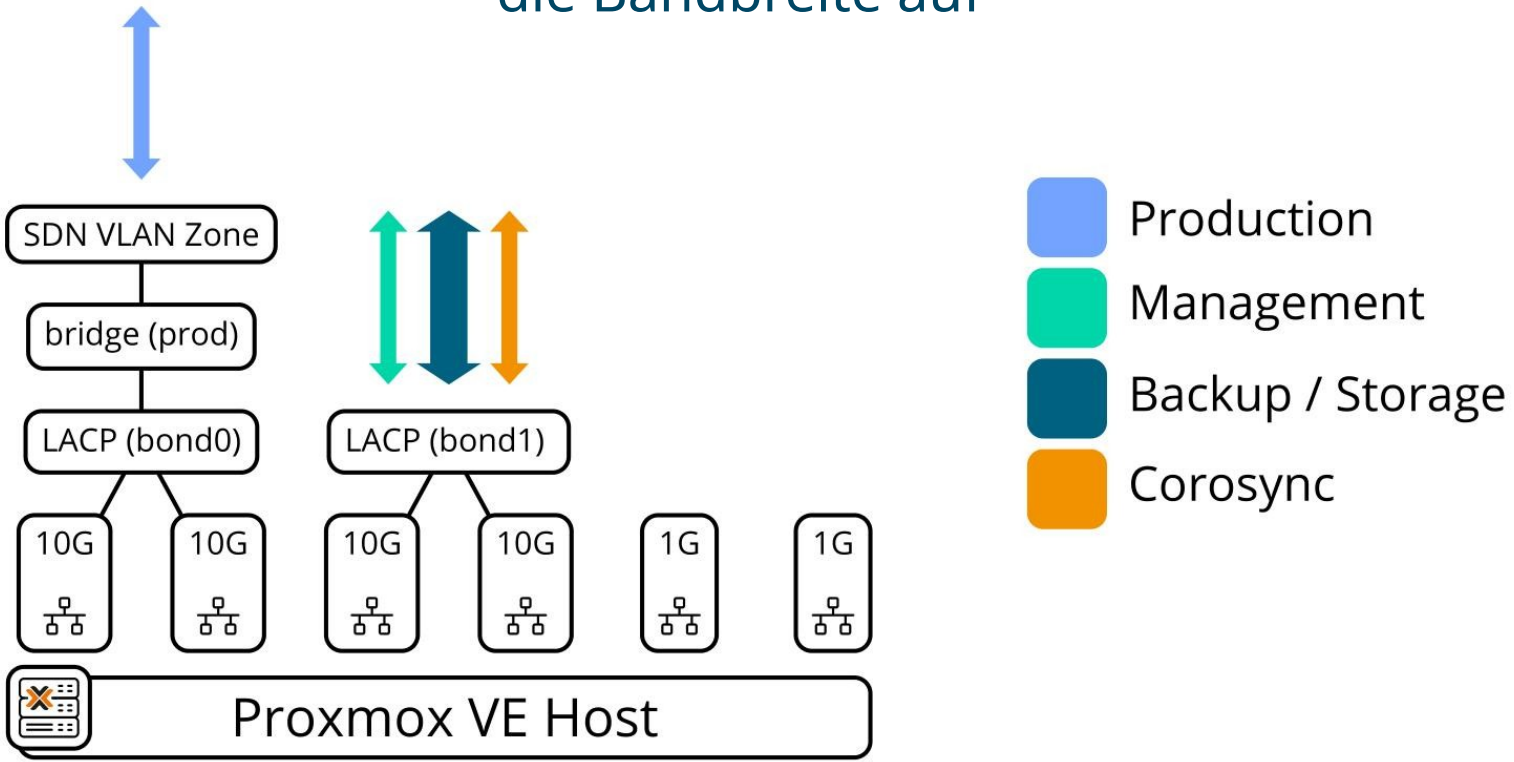
Netzwerk



- Production
- Management
- Backup / Storage
- Corosync

Netzwerk

- Backup und/oder Storage braucht die Bandbreite auf

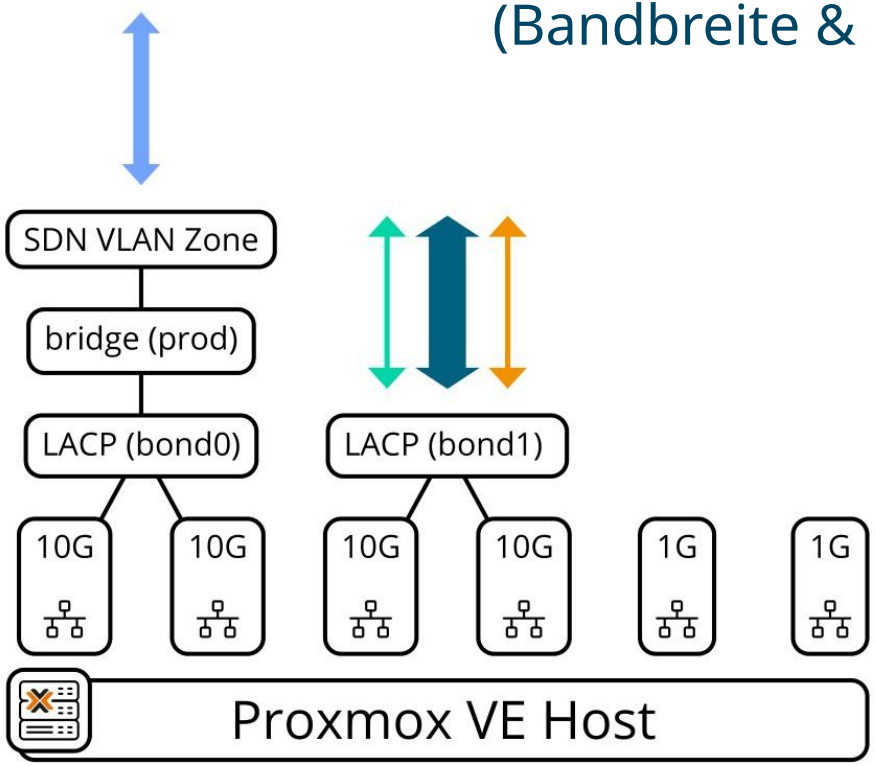


© inett GmbH -- info@inett.de -- www.inett.de

Netzwerk

- Andere Dienste leiden darunter

  (Bandbreite & Latenz)

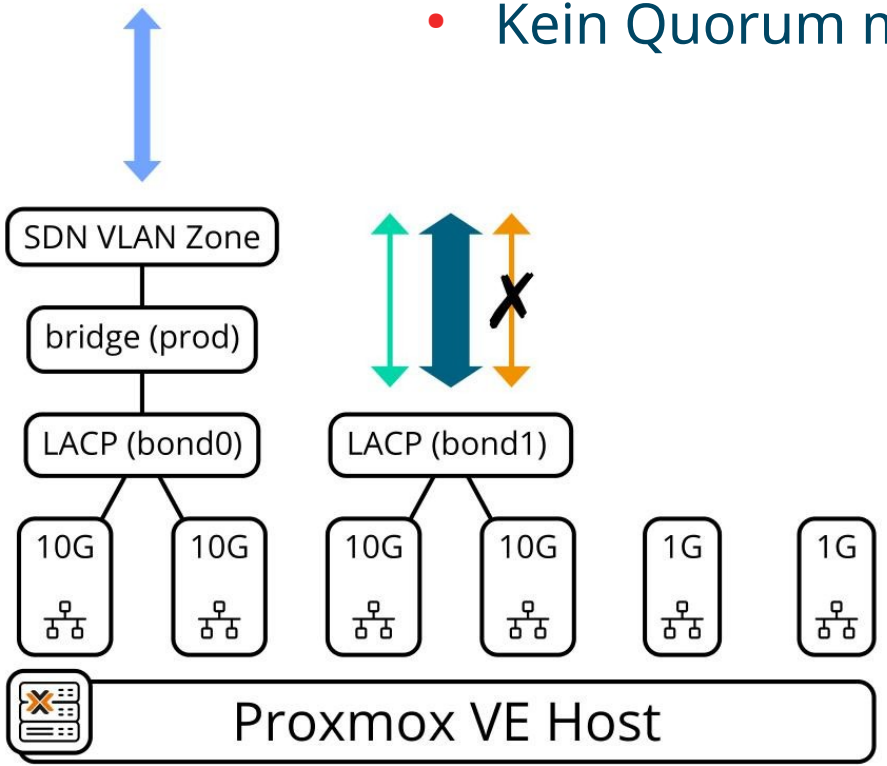




© inett GmbH -- info@inett.de -- www.inett.de

Netzwerk

- Corosync erklärt Link als down / unbrauchbar
- Kein Quorum mehr = Cluster down

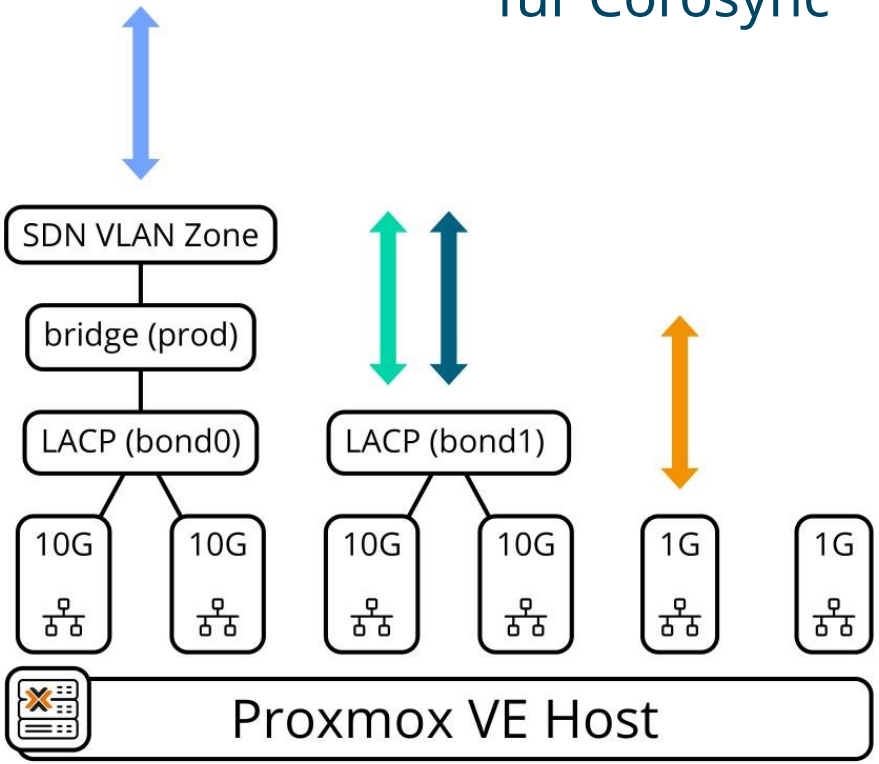




© inett GmbH -- info@inett.de -- www.inett.de

Netzwerk

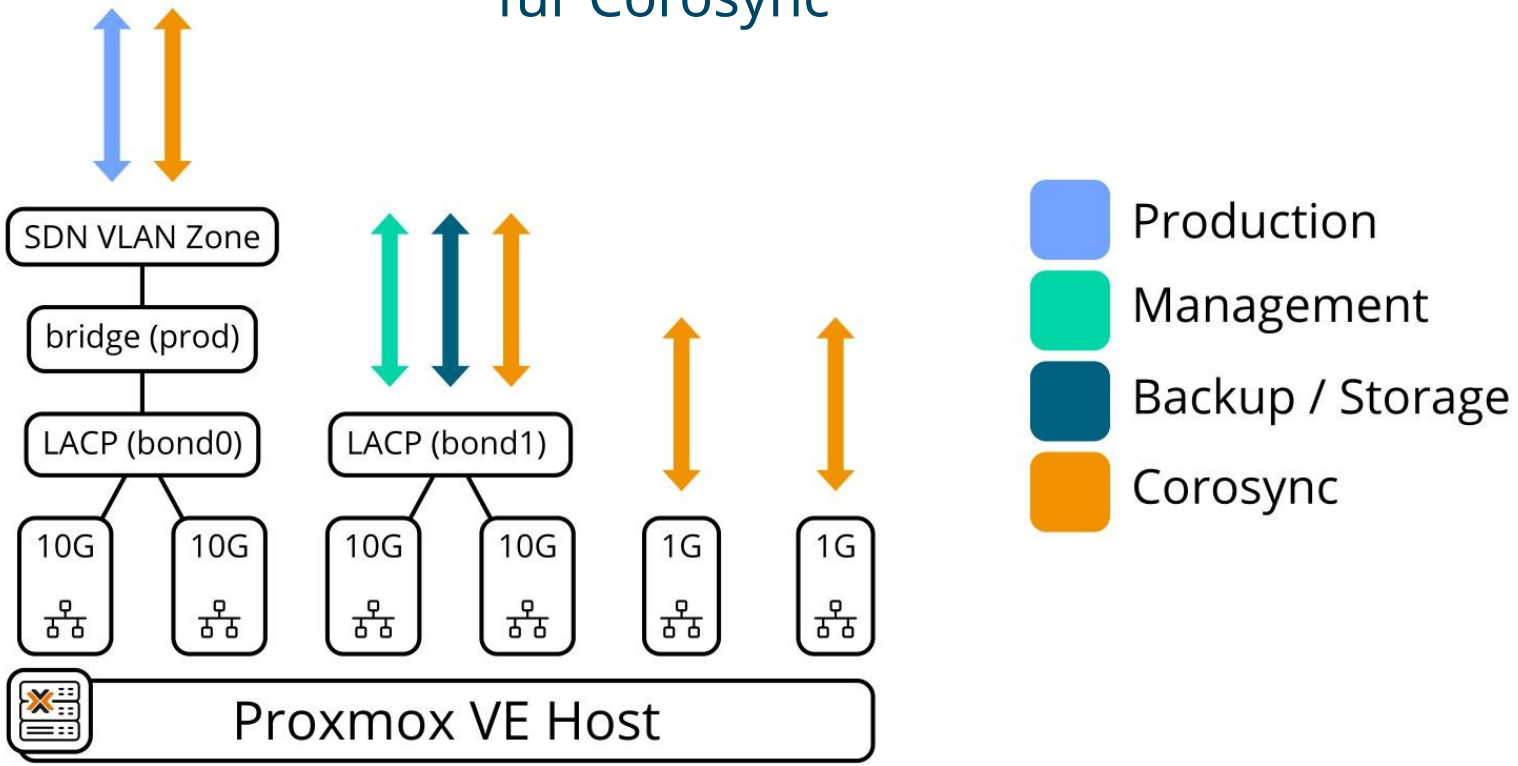
- Besser: dediziertes Netzwerk für Corosync



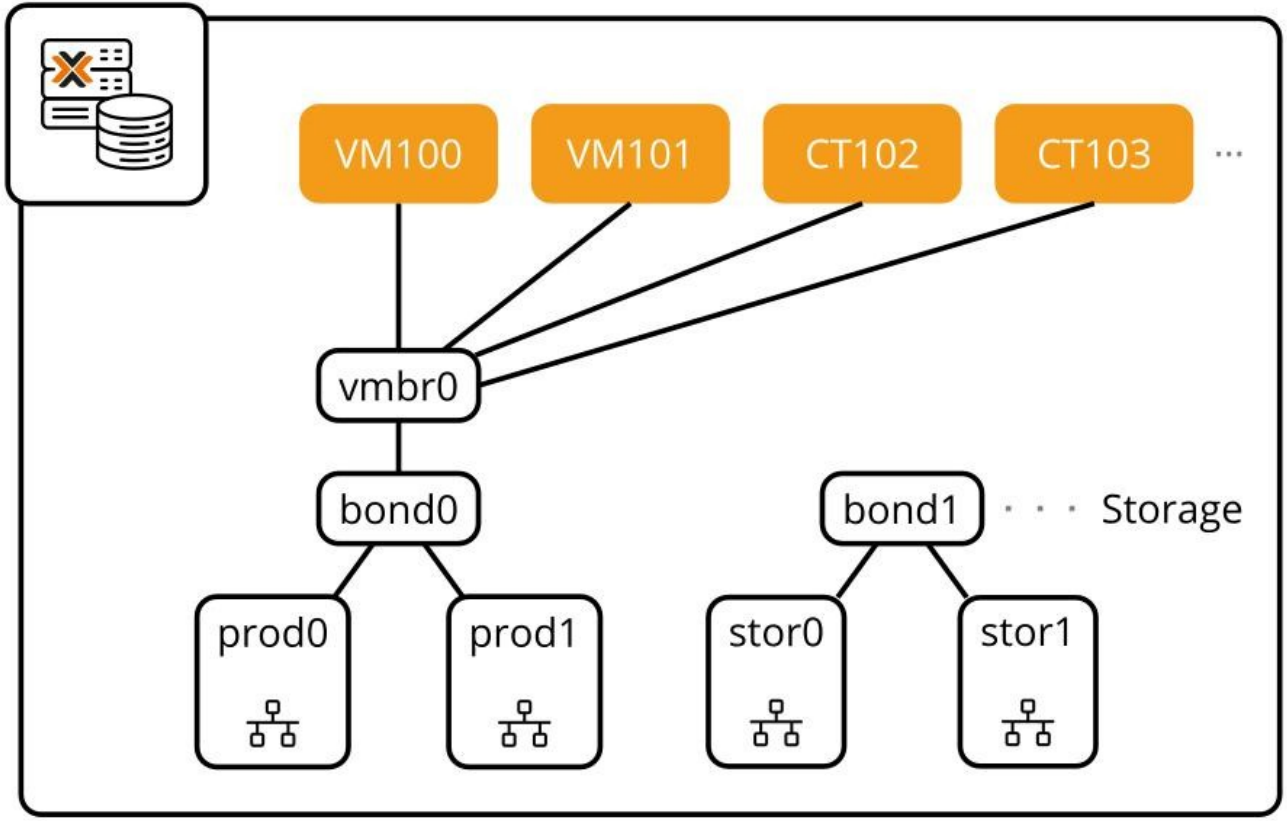


Netzwerk

- Empfohlen: redundante Netzwerke für Corosync

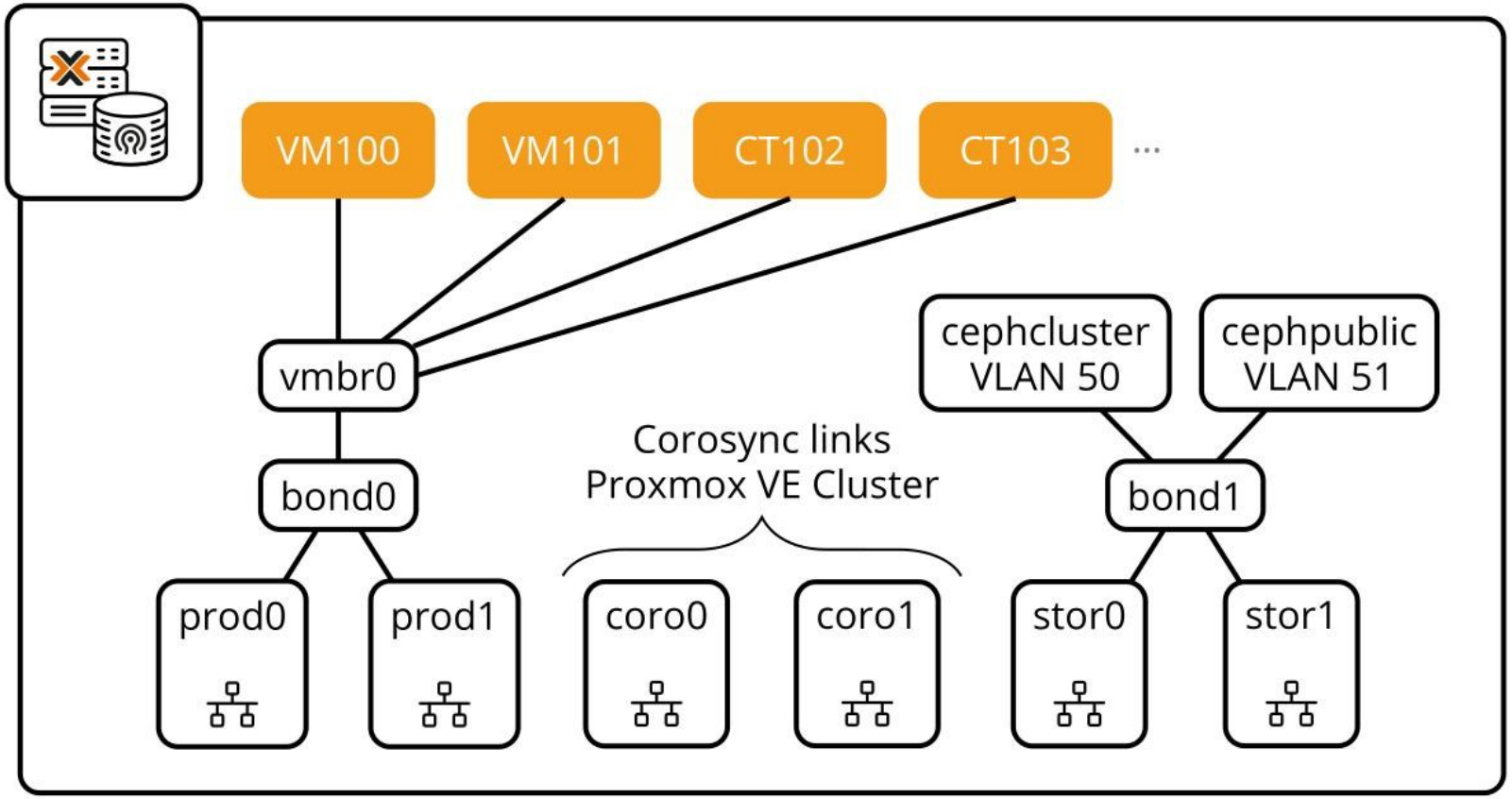


Netzwerk mit Ceph



© inett GmbH -- info@inett.de -- www.inett.de

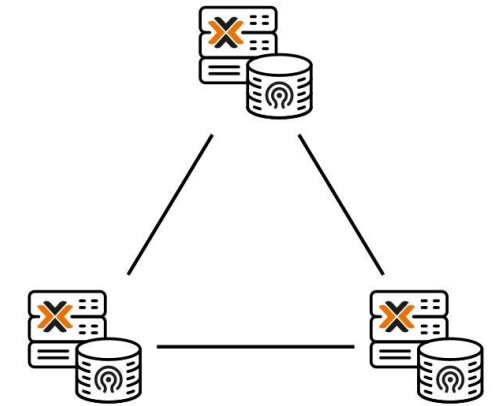
Netzwerk mit Cluster und Ceph über VLANs

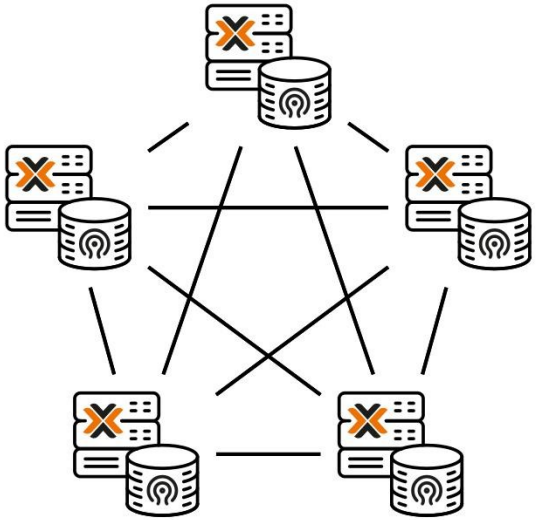


© inett GmbH -- info@inett.de -- www.inett.de

Netzwerk in Full-Mesh-Topologie

- Keine Switches
- Skaliert weniger gut (normal 3-5 Nodes)
- Gut für Cluster-interne Netzwerke (z.B. Ceph)

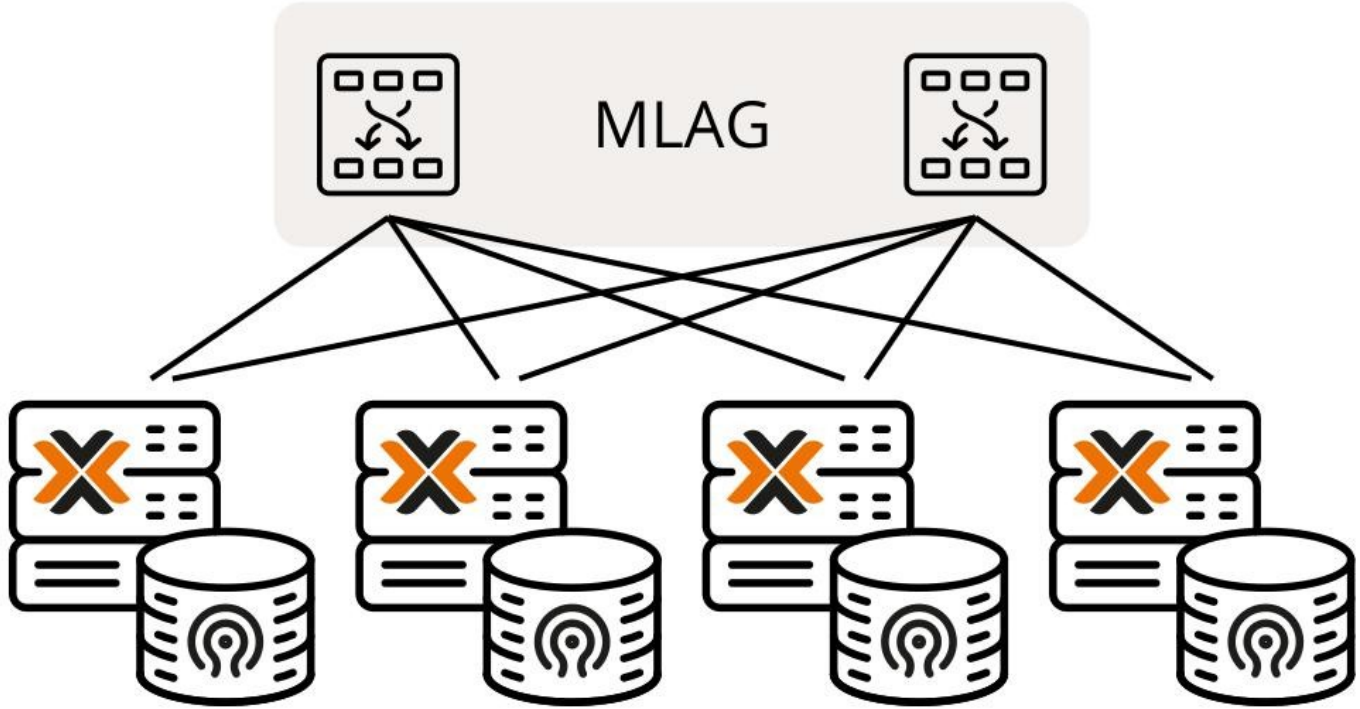




© inett GmbH -- info@inett.de -- www.inett.de

Netzwerk in Stern-Topologie

- Skaliert gut
- MLAG für Redundanz

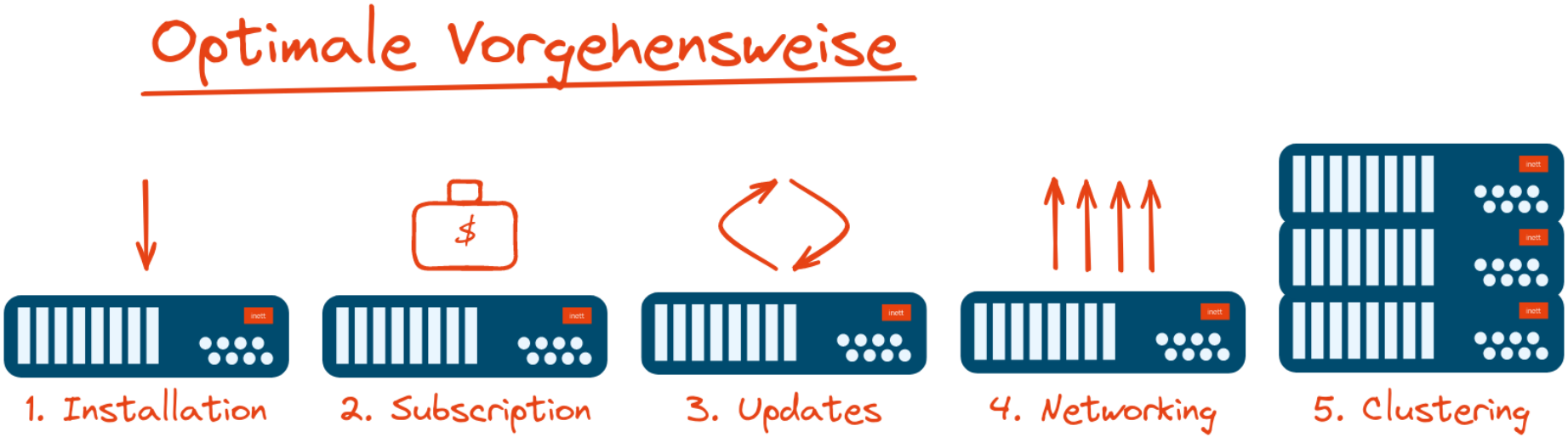


© inett GmbH -- info@inett.de -- www.inett.de

Installation der PVE Cluster Nodes



Installation der PVE Cluster Nodes



©inett GmbH -- info@inett.de -- www.inett.de

Optimale Reihenfolge

- Nodes installieren
- Subscription aktivieren
- Updates installieren bis zum aktuellen Stand
- Netzwerk konfigurieren (mehrere Links)
- Cluster erstellen
- Weitere Nodes hinzufügen

©inett GmbH -- info@inett.de -- www.inett.de

Installation der PVE Cluster Nodes

inett

Vorgehensweise (Live Task)

- Alle Proxmox VE Server updaten und auf die aktuellste Version bringen
- Reboot und GUI neu laden
- *NTP und Zeitsynchronisation auf Funktion prüfen (shell)*
- Füge alle Hostnamen der Datei /etc/hosts auf allen Proxmox VE Servern hinzu
- Erstelle einen Cluster auf dem ersten Server
    - GUI: Datacenter → Cluster → Create
    - Shell: `# pvecm create <clustername> [Options]`

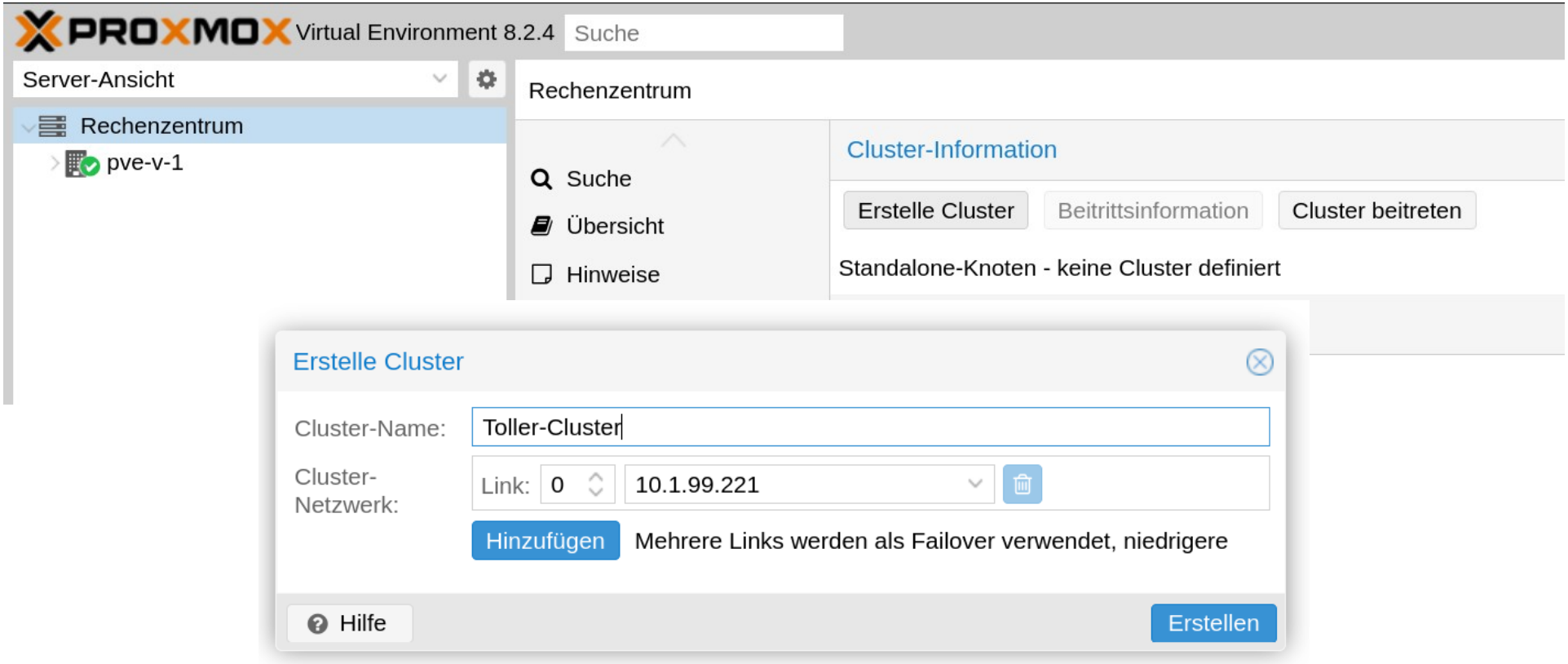
Nachträgliche Erstellung eines Clusters

- *Erstellung* eines Clusters möglich, wenn bereits VMs existieren
- *Hinzufügen* einer Node zu einem Cluster *nur dann* möglich, wenn noch keine VMs auf der Node existieren
- Workaround: Backup VMs → VMs löschen → Clusterbeitritt

  → Restore VMs

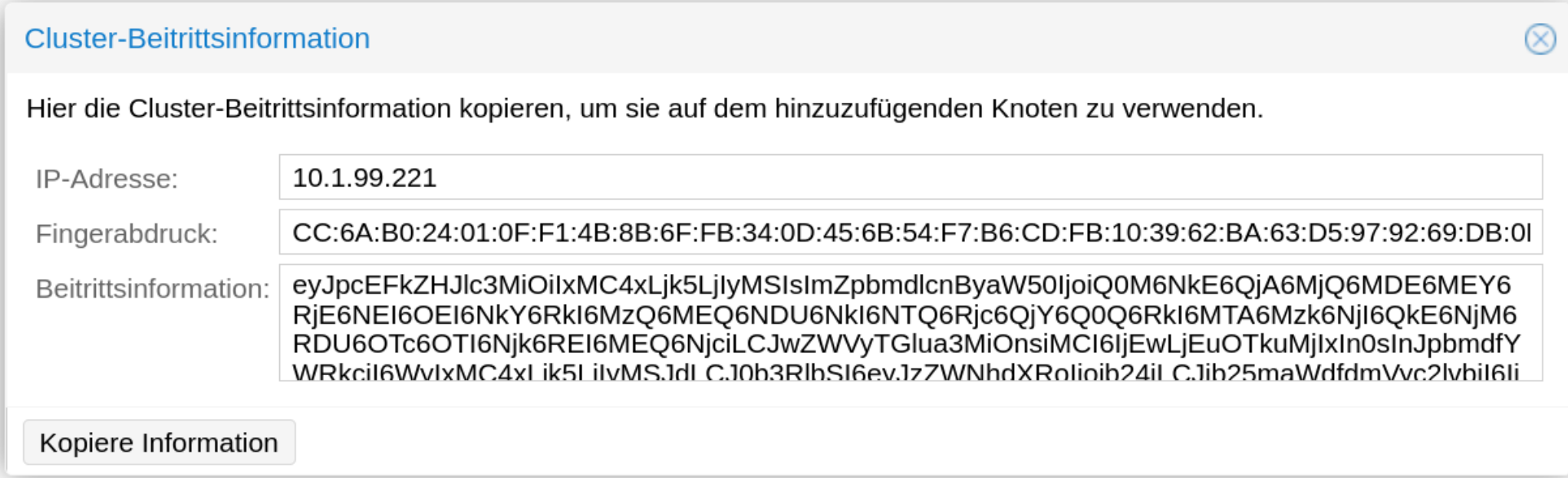
©inett GmbH -- info@inett.de -- www.inett.de

Installation der PVE Cluster Nodes



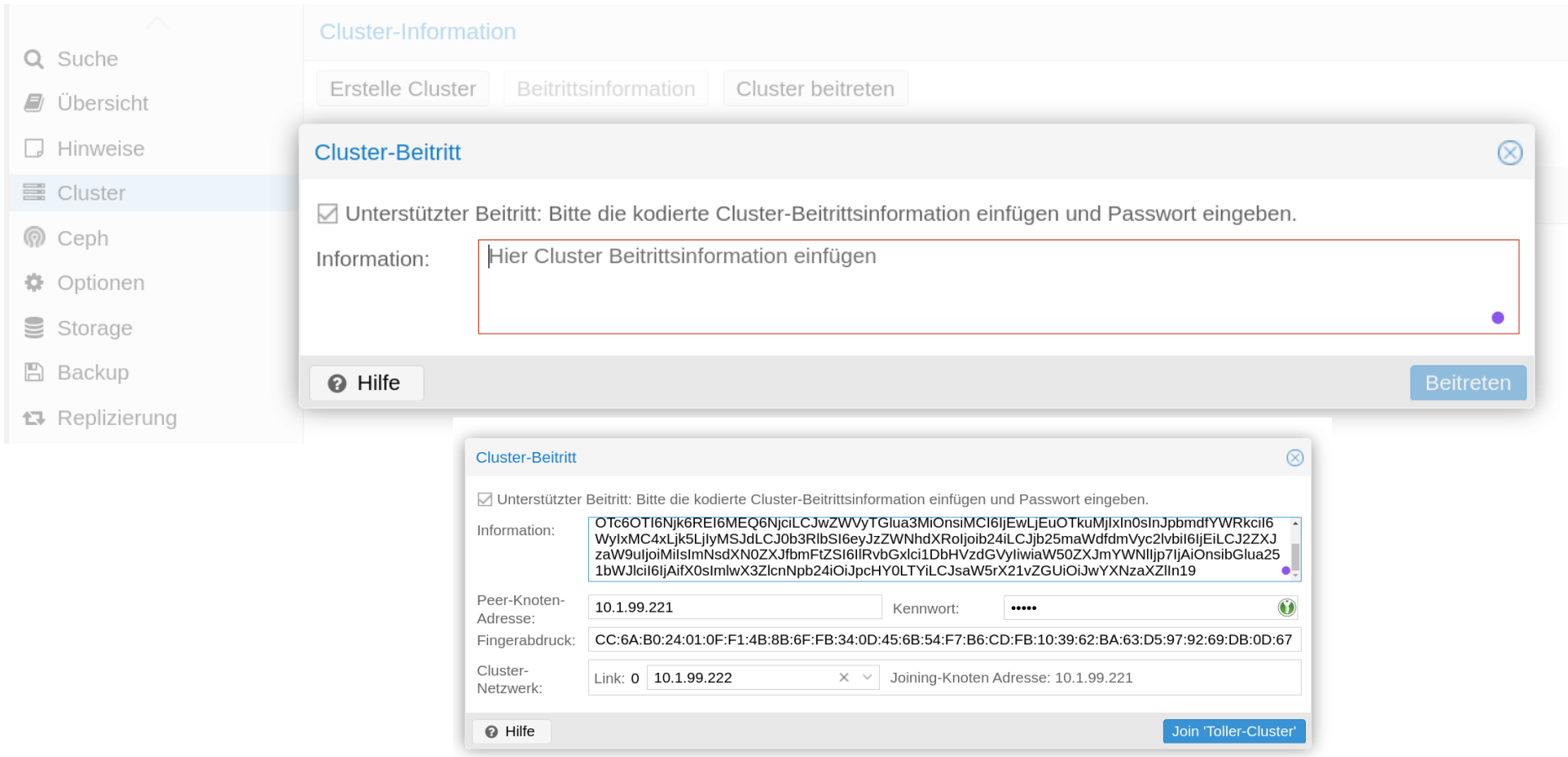
Installation der PVE Cluster Nodes

inett



- Hint: `# cat „Beitrittsinformation“ | base64 -d | jq` *ist ... interessant*

Installation der PVE Cluster Nodes



Installation der PVE Cluster Nodes

- Der dritte Node kann dann als Übung per CLI dem Cluster hinzugefügt werden:
  - # ssh pve-*X*-3
  - (von dort:) # pvecm add pve-*X*-1 (oder pve-*X*-2)

- Status des Clusters prüfen:

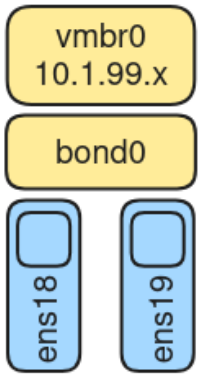
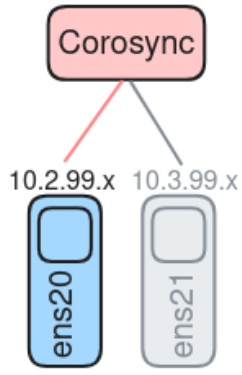
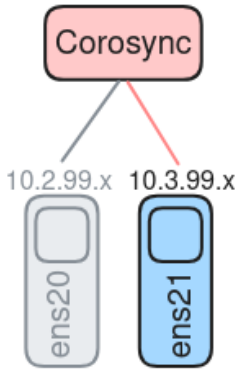
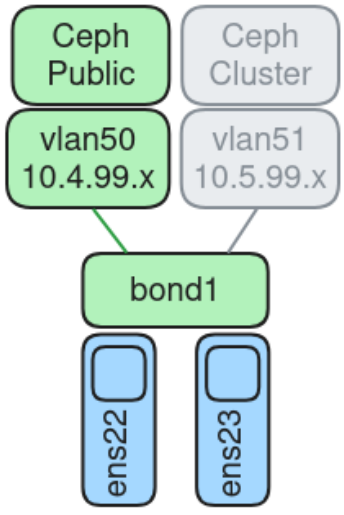
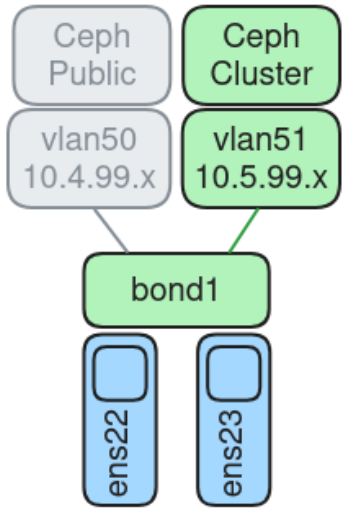
  - # pvecm status
  - # corosync-cfgtool -s
  - # iptraf-ng *(und 2. Menü-Eintrag)*

Anpassung der Cluster-Konfiguration

- Zwar funktioniert unser Cluster, aber:
- ***Corosync nutzt nur eine Netzwerkverbindung → Keine Redundanz!***
- ***Corosync nutzt den Uplink → evtl. hohe Latenz, wenn Uplink ausgelastet!***
- Da wir sechs Netzwerkkarten pro Server zur Verfügung haben, werden wir nun die Konfiguration anpassen um Corosync mit einem zweiten Link auszustatten:

    - 10.1.99.XXX = existierender Uplink auf bond0 / vmbr0
    - 10.2.99.XXX = Corosync Link 0
    - 10.3.99.XXX = Corosync Link 1
    - 10.4.99.XXX = Ceph_Public über VLAN 50, ein Bond wird hergestellt
    - 10.5.99.XXX = *Ceph_Cluster_Network* über VLAN 51

Anpassung der Cluster-Konfiguration: Beispiel

Hostname	Uplink	Corosync Link0	Corosync Link1	Ceph Public	Ceph Cluster
					
desktop-k	10.1.99.110				
pve-k-1	10.1.99.111	10.2.99.111	10.3.99.111	10.4.99.111	10.5.99.111
pve-k-2	10.1.99.112	10.2.99.112	10.3.99.112	10.4.99.112	10.5.99.112
pve-k-3	10.1.99.113	10.2.99.113	10.3.99.113	10.4.99.113	10.5.99.113
pve-k-4	10.1.99.114	10.2.99.114	10.3.99.114	10.4.99.114	10.5.99.114
pbs-k	10.1.99.115	10.2.99.115	10.3.99.115	10.4.99.115	10.5.99.115

©inett GmbH -- info@inett.de -- www.inett.de

Anpassung der Cluster-Konfiguration

(Erst Netzwerk komplett konfigurieren, dann anwenden)

- Bei vmbr0 die Karte ens18 entfernen
- Dann bond0 aus ens18 und ens19 erstellen
- Dann bond0 in vmbr0 *wieder* einfügen (sonst kein Uplink → Game Over!)
- ens20 und ens21 werden einzeln als *Corosync-Link 0* und *1* konfiguriert
- ens22 und ens23 bilden zusammen den bond1
- Interfaces VLAN50 und VLAN51 erstellen und bond1 hinzufügen, um Ceph Traffic zwischen *Ceph_Public_Network* und *Ceph_Cluster_Network* logisch zu trennen

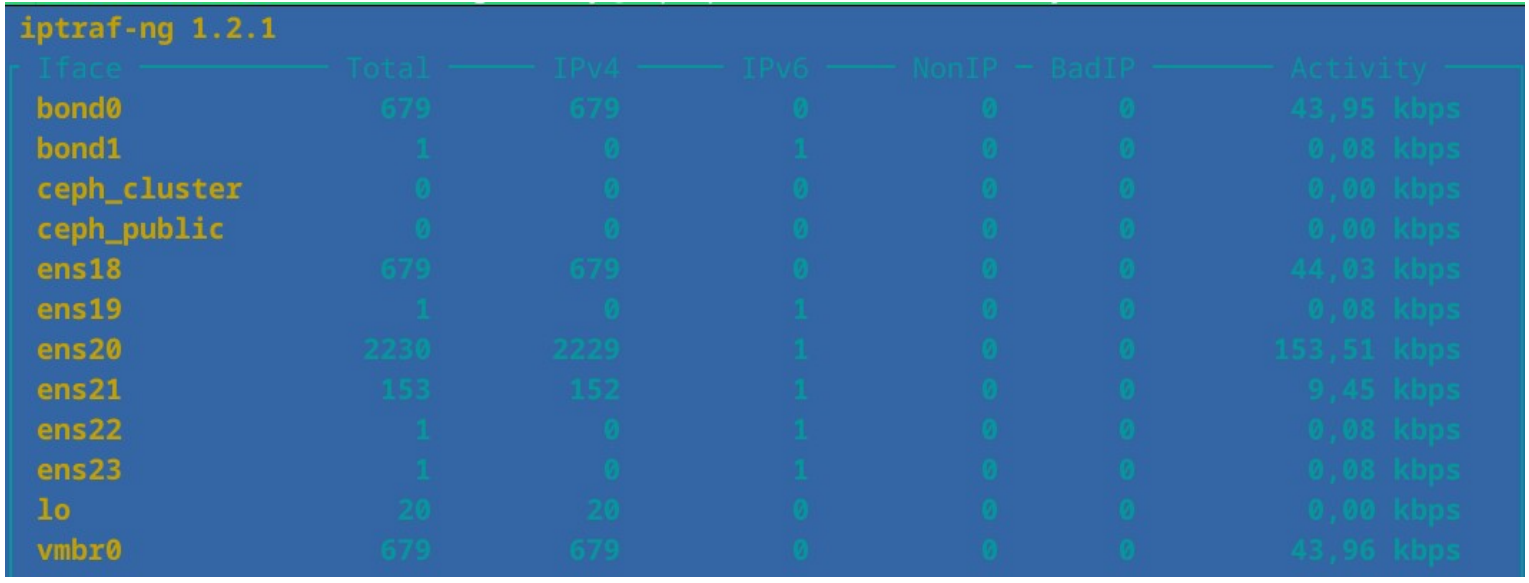
Anpassung der Cluster-Konfiguration

- Nachdem das Netzwerk angepasst wurde, muß auch Corosync konfiguriert werden, damit er die richtigen Karten auch nutzt.
- !! Auf jeden Fall eine Kopie von `/etc/pve/corosync.conf` machen, und auf diese Kopie arbeiten ! Z.B.: `# cp /etc/pve/corosync.conf /root/`
- Zuerst wird *ring1* hinzugefügt
- Syntax kontrollieren : `# corosync -t -c ~/corosync.conf`
- Erst wieder zu `/etc/pve` überschreiben,
  wenn die Syntax stimmt.
- *Dann erst ring0* mit der gleichen Methode ändern.



Anpassung der Cluster-Konfiguration

- *iptraf-ng* erlaubt Check von Bandbreite und Netzwerkaktivität
- Installation mit: `# apt install -y iptraf-ng`



Proxmox Cluster Filesystem (pmxcfs)



Proxmox Cluster Filesystem

- ermöglicht clusterweite, zentrale Konfiguration
- wird durch pve-cluster.service gemountet
- Limits: Gesamtgröße 128 MB, pro Datei maximal 1 MB
- Dateien gehören alle root, Gruppe ist www-data (nicht änderbar)
- Der Benutzer darf schreiben, die Gruppe darf nur lesen
  - pvedaemon läuft unter root → darf schreiben
  - pveproxy läuft als www-data → darf nur lesen
- .clusterlog ist nur lesbar → selbst root sollte die logs nicht ändern

©inett GmbH -- info@inett.de -- www.inett.de

Proxmox Cluster Filesystem Backend

- Backend ist Sqlite3 Datenbank
- /var/lib/pve-cluster/config.db enthält sämtliche Daten aus /etc/pve
- Zur Not kann `# sqlite3 config.db` bei einem Recovery helfen
- config.db ist auf allen Servern eines Clusters logisch gleich
- Wird über Corosync in Echtzeit repliziert

Cluster File System: pmxcfs

- Debug Modus:
  - Logs vorher: `# journalctl -f`
  - *dann* `# echo 1 > .debug` *→ journalctl wird gesprächig!*
  - deaktivieren mit `# echo 0 > .debug`
- Symlinks wie `qemu-server`, `lxc` oder `local` sind NICHT in der Datenbank enthalten, sondern werden von pve-cluster beim mount erzeugt

  (siehe `/etc/pve/local` in verschiedenen Servern).
- pmxcfs ist eine Sqlite Datenbank, die sich in `/var/lib/pve-cluster` befindet.
- *Zur Not kann* `# sqlite3 config.db` *helfen, Dateien aus einen gefährdeten Cluster zu retten.*

Proxmox VE HCI
Hyper-Converged Infrastructure



©inett GmbH -- info@inett.de -- www.inett.de

HCI - Hyper-Converged Infrastructure



Ceph Storage

- Ceph ist ein redundanter, über *viele* Server verteilter Objektspeicher
- Bietet verschiedene Arten von Speicher:
  - S3-/Swift-API-kompatibler Objektspeicher (RADOS Gateway)
  - Virtuelle Block Storage (RADOS Block Device / RBD)
  - Verteiltes Cluster File System (CephFS)
- Alles im gleichen Cluster
- Ab 3 Server bis zu tausenden von Servern skalierbar

HCI: Ceph Features

- Software Defined Storage, Open Source
- Läuft auf Standardhardware, kein RAID Controller
- Snapshots, Copy-On-Write, Cloning
- Thin Provisioning, Selbstheilend, Skalierbar bis Exabyte Level
- Client direkt in Linux und QEMU KVM integriert
- Caching und Auto-Tiering, Pools für HDD, SSD, NVMe
- CRUSH Algorithmus (Controlled Replication Under Scalable Hashing)
- NFS, iSCSI Gateways verfügbar
- Kommerzieller Support von Proxmox, Red Hat, Suse, Ubuntu und anderen IT Service Providern

Referenz: CERN

CERN
@CERN

Today's #PhotoOfTheWeek takes us to the CERN Data Centre.

It can store and analyse more than 600 petabytes of data (600 million gigabytes) generated by the collisions during #BeamTime of #LHCRun3.

It corresponds to over 20 000 years of 24/7 HD video recording. #ComputingatCERN

Tweet übersetzen

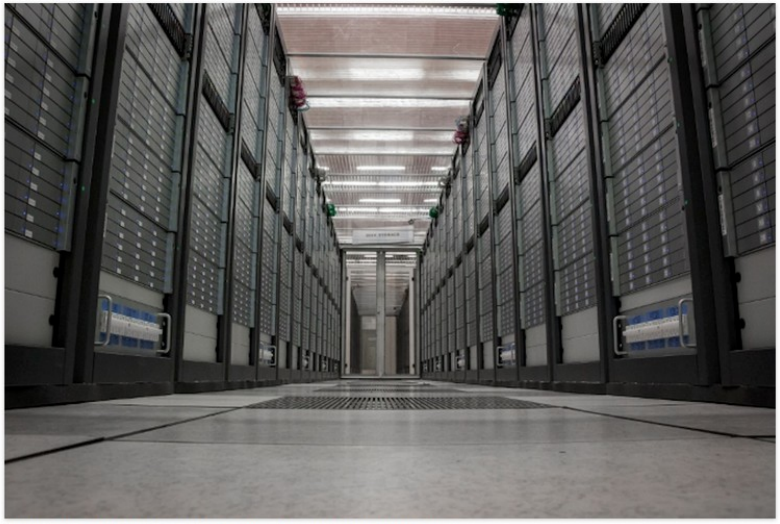


5:03 nachm. · 5. Apr. 2023 · **63.300** Mal angezeigt

An exabyte of disk storage at CERN

CERN disk storage capacity passes the threshold of one million terabytes of disk space

29 SEPTEMBER, 2023 | By Tim Smith



A fraction of the 111 000 devices that form CERN's data storage capacity. (Image: CERN)

CERN's data store has now crossed the remarkable capacity threshold of one exabyte, meaning that CERN has one million terabytes of disk space ready for data!

Ceph Integration in Proxmox VE

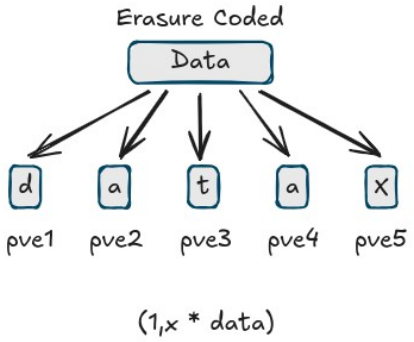
- Proxmox VE kann entweder:
  - An externe Ceph Cluster angebunden werden
  - Als Storage Komponente für einen hyperkonvergenten Betrieb von Proxmox VE (HCI) verwendet werden
- Im Falle von Hyperkonvergenz (HCI):
  - Ceph Administration und Management über die Proxmox GUI möglich
  - Ceph Dashboard in der Proxmox GUI vorhanden
  - Ceph CLI Befehle können genutzt werden

Ceph Voraussetzungen

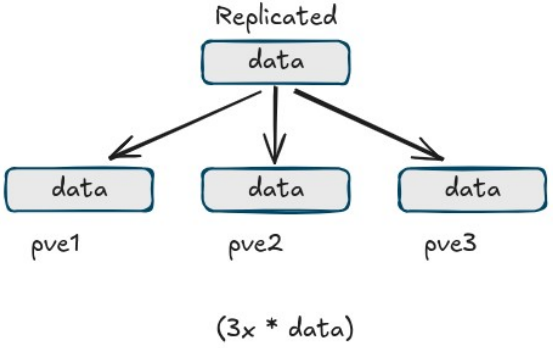
- Netzwerk 10 Gbit/s oder schneller
- Mindestens 4 Disks pro Server
- Ähnliche Disks und Server in Clustern bei weniger als ca. 10 Nodes
- Ceph benötigt RAM und CPU
  - Minimum ***pro*** Storage Device (OSD):
    - 5 GB RAM
    - 1 Core
- MON, MGR, MDS benötigen zusätzlich RAM und Rechenleistung

Ceph Pool Typen

Erasure Coded (EC) Pool: Splittet Daten in ***k*** Chunks und erzeugt zusätzlich ***m*** Coding Chunks, jeder Chunk wird auf einem separaten Server gespeichert



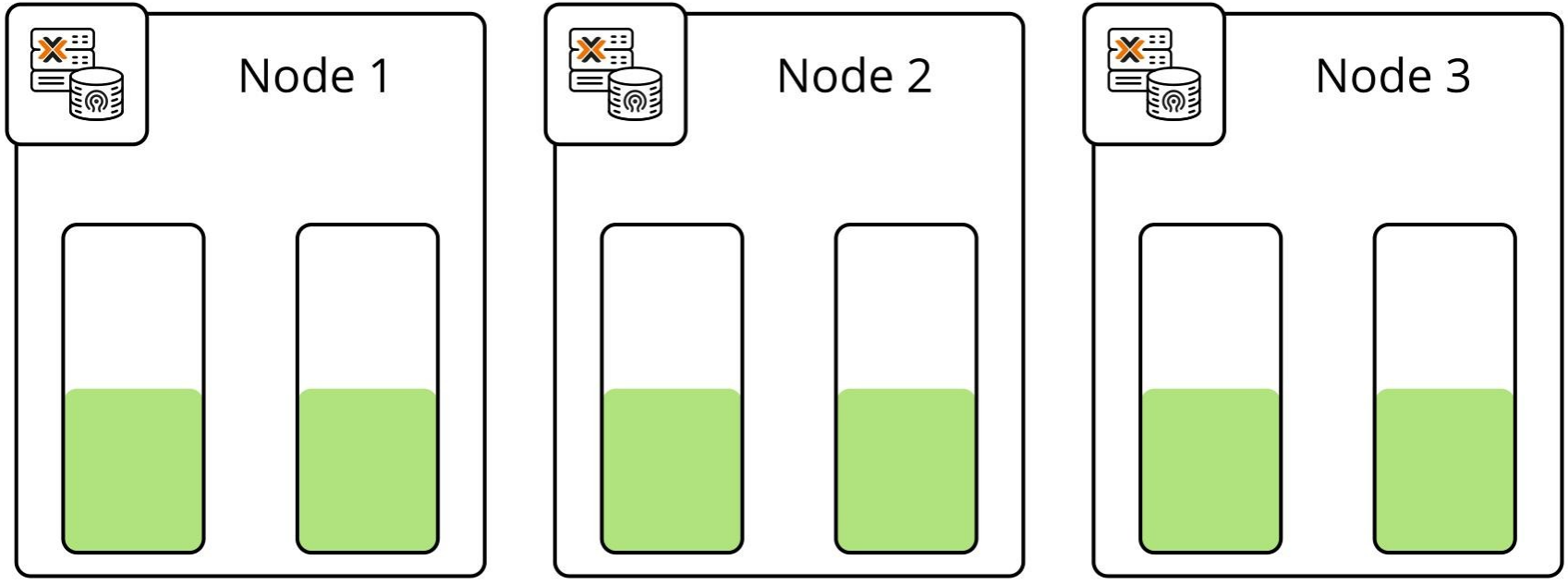
Replicated Pool: Speichert mehrere Kopien / Replika, jede Kopie wird auf einem separaten Server gespeichert (Redundanz)



©inett GmbH -- info@inett.de -- www.inett.de

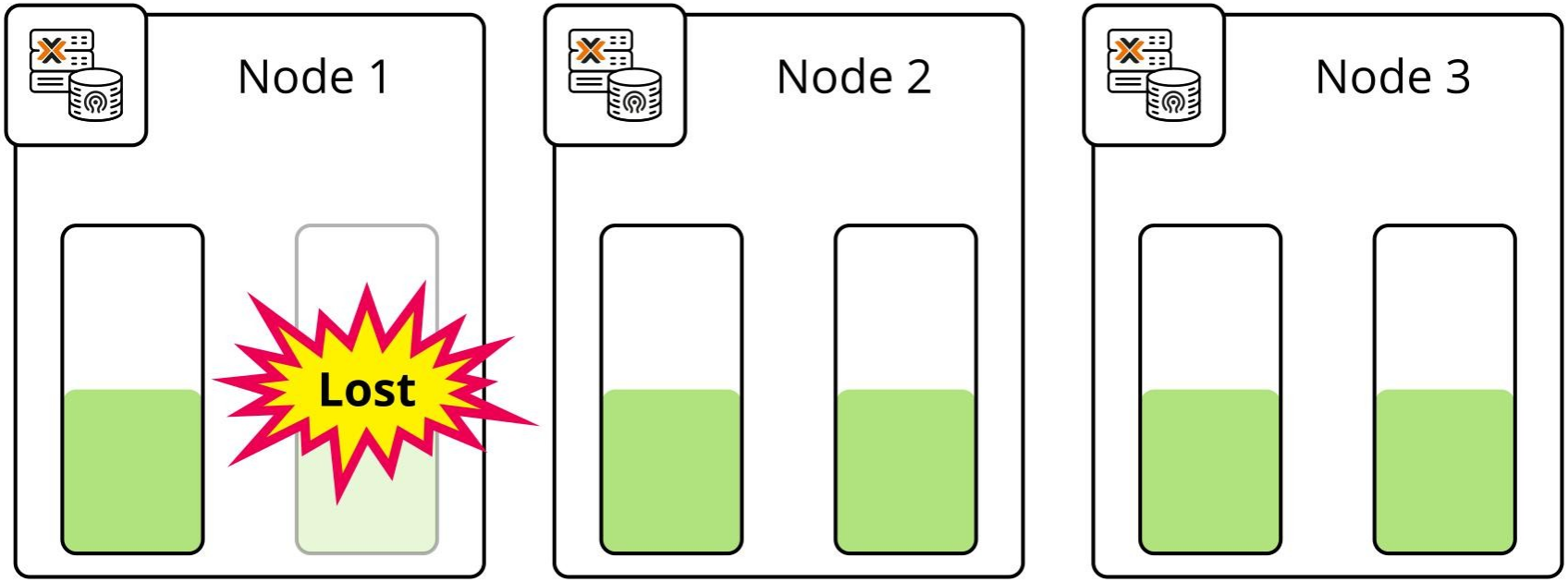
Zu wenig OSDs pro Node

- Wenn die Anzhal der Knoten der Anzhal der Replicas gleicht, müssen spezifische Maßnahmen berücksichtigt werden !



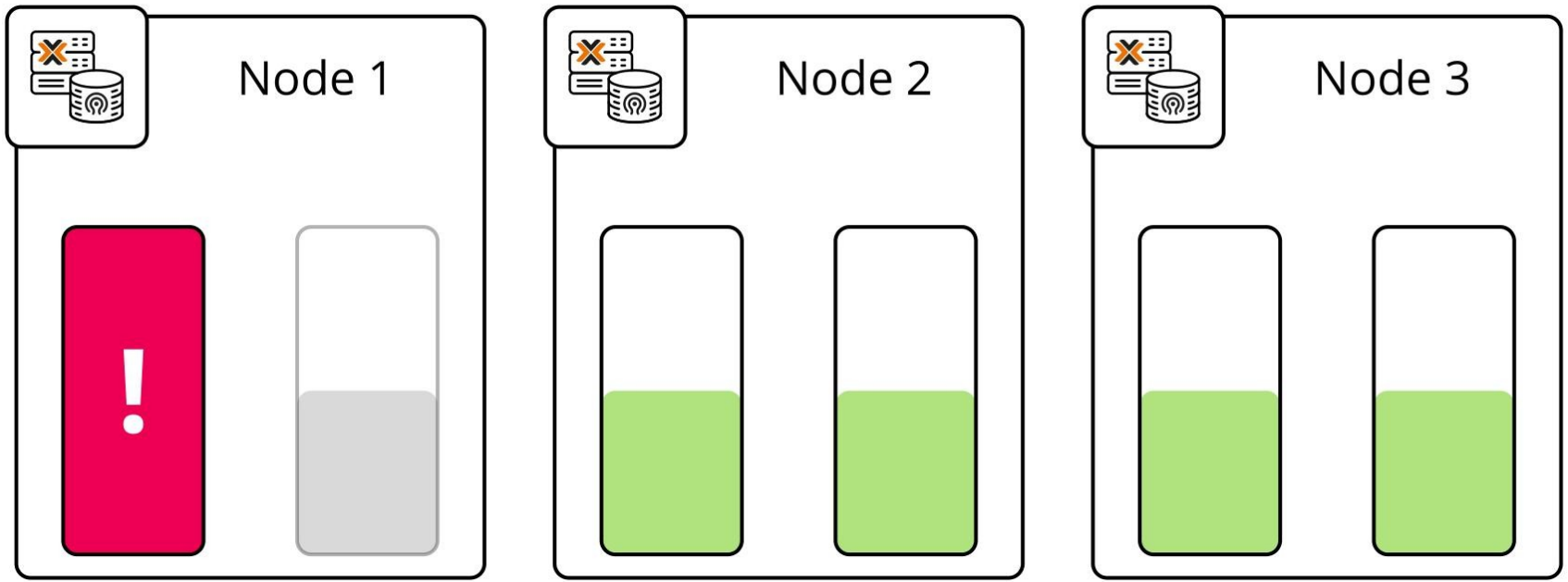
Zu wenig OSDs pro Node

- Wenn die Anzhal der Knoten der Anzhal der Replicas gleicht, müssen spezifische Maßnahmen berücksichtigt werden !



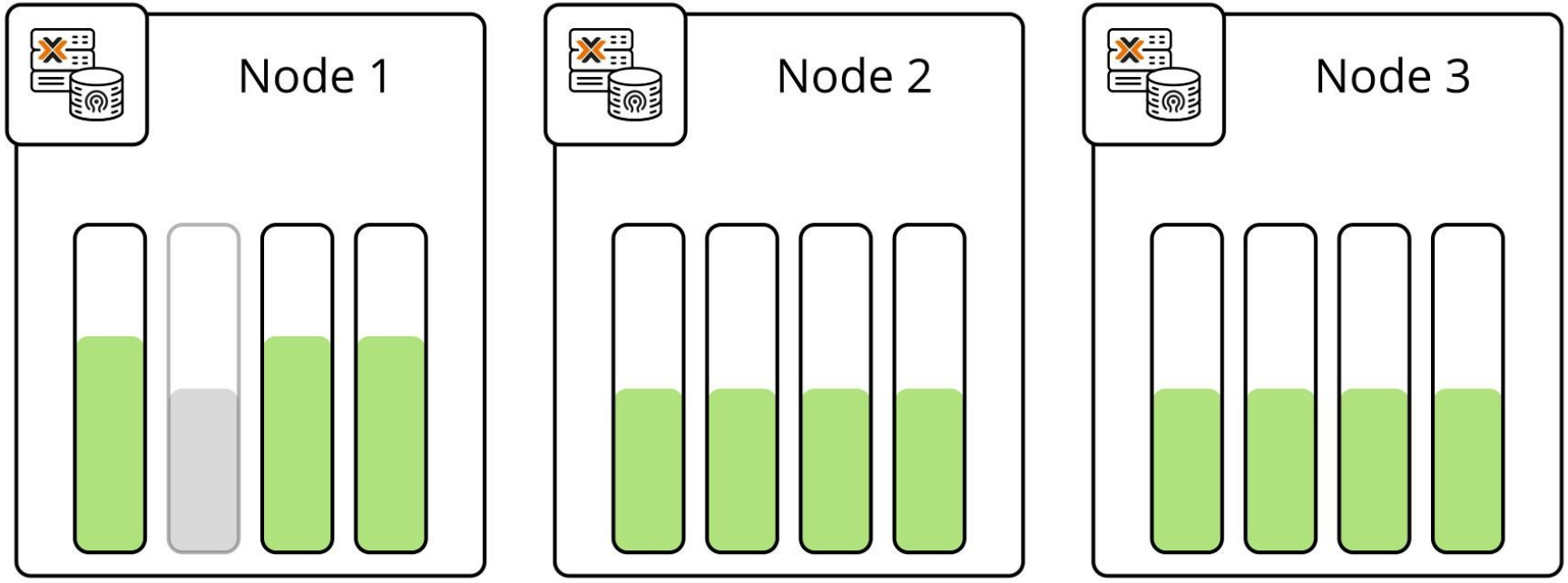
Zu wenig OSDs pro Node

- Wenn die Anzhal der Knoten der Anzhal der Replicas gleicht, müssen spezifische Maßnahmen berücksichtigt werden !



Zu wenig OSDs pro Node

- Wenn die Anzhal der Knoten der Anzhal der Replicas gleicht, müssen spezifische Maßnahmen berücksichtigt werden !



©inett GmbH -- info@inett.de -- www.inett.de

Beispiel: Kapazitätsplanung

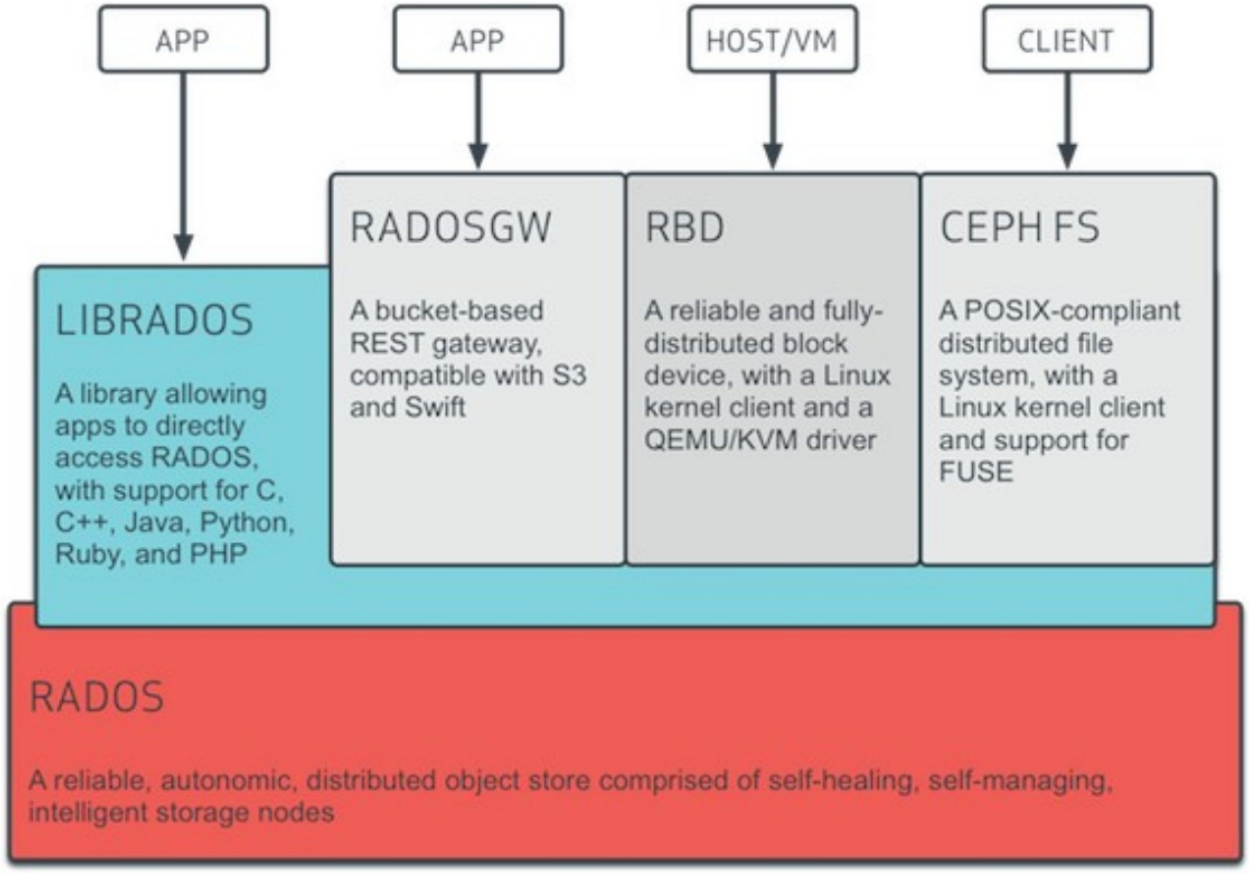
- Standard: Replicated Pool (size=3, min_size=2)
- Benötigt für VMs: 10 TB
- „Verschnitt“: 10 TB * 1,1 = 11 TB
- 3-fache Replikation: 11 TB * 3 = 33 TB
- Max. Space Usage < 85% = 33 TB / 0.85 = 38.82 TB
- Objekte sind nicht absolut gleich verteilt, Abweichung ~10%:

  38.82 TB / 0.9 = 43.13 TB
- Fällt ein Server aus, werden Objekte auf restlichen Nodes wiederhergestellt
  → Mehr Speicher nötig: 1 von 5 fällt aus = 20% → 43.13 TB / 0.8 = 54 TB

  54 TB auf 5 Servern = ~11 TB pro Server
- 11 TB = 4x 3.68 TB SSDs/NVMe pro Server

©inett GmbH -- info@inett.de -- www.inett.de

Ceph Stack


APP
APP
HOST/VM
CLIENT
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
RADOS
A reliable, autonomic, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

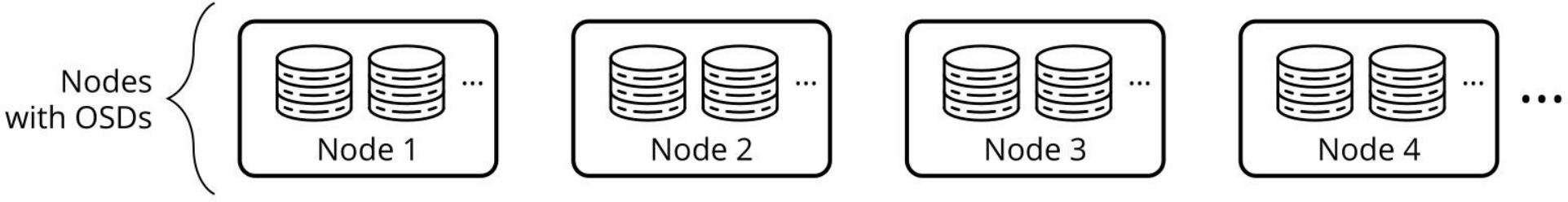
Ceph Komponenten



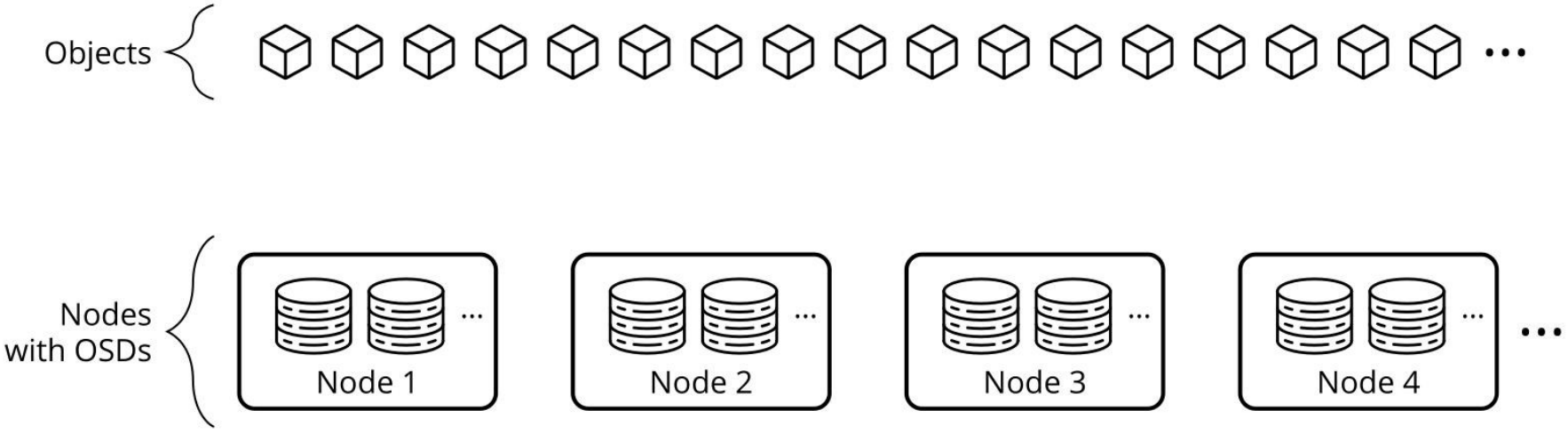
- **OSD (Object Storage Daemon/Device):** speichern Daten-Objekte, überprüfen den eigenen Status sowie den Status anderer OSDs und melden Informationen an die MONs.
- **MON (Monitor)** hält die Cluster-Map und bildet mit anderen MONs das Quorum. Authentifiziert Clients und versorgt sie mit einer Kopie der Cluster-Map.
- **MGR (Manager)** fungiert als Endpunkt für Monitoring, Orchestrierung und Plug-In-Module.
- **MDS (Metadata Server)** verwaltet die Metadaten von CephFS-Dateisystemen.

©inett GmbH -- info@inett.de -- www.inett.de

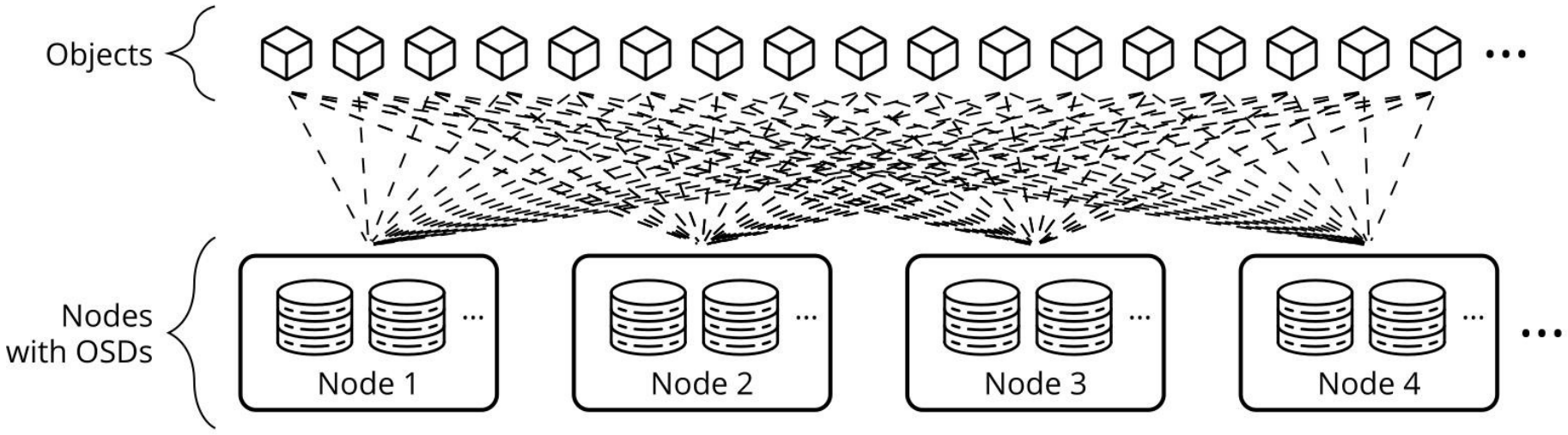
Ceph Data Organization



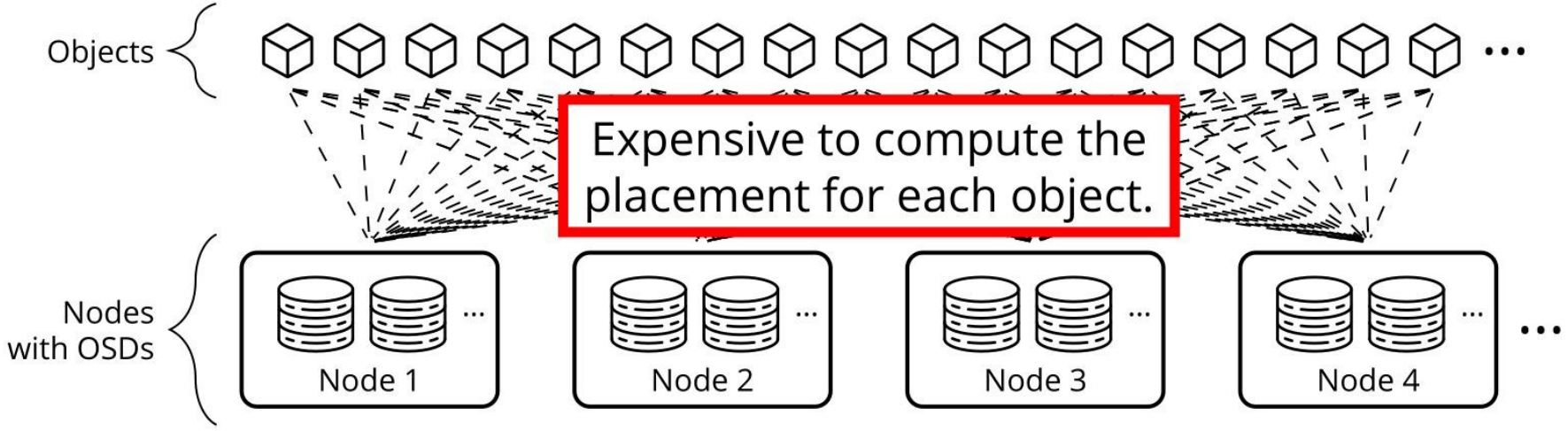
Ceph Data Organization



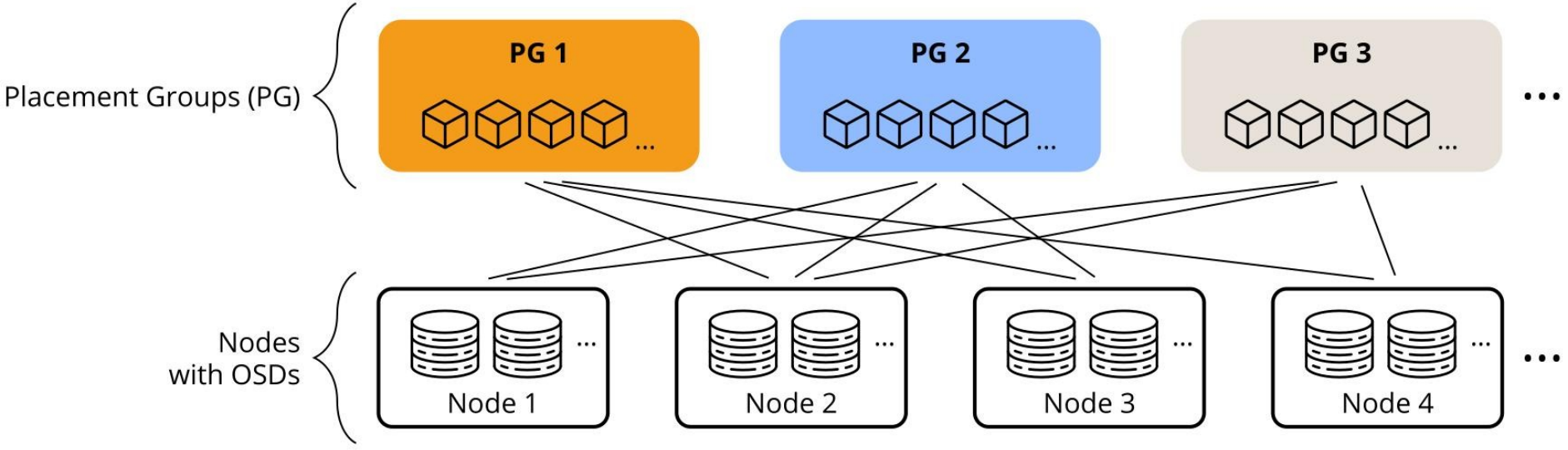
Ceph Data Organization



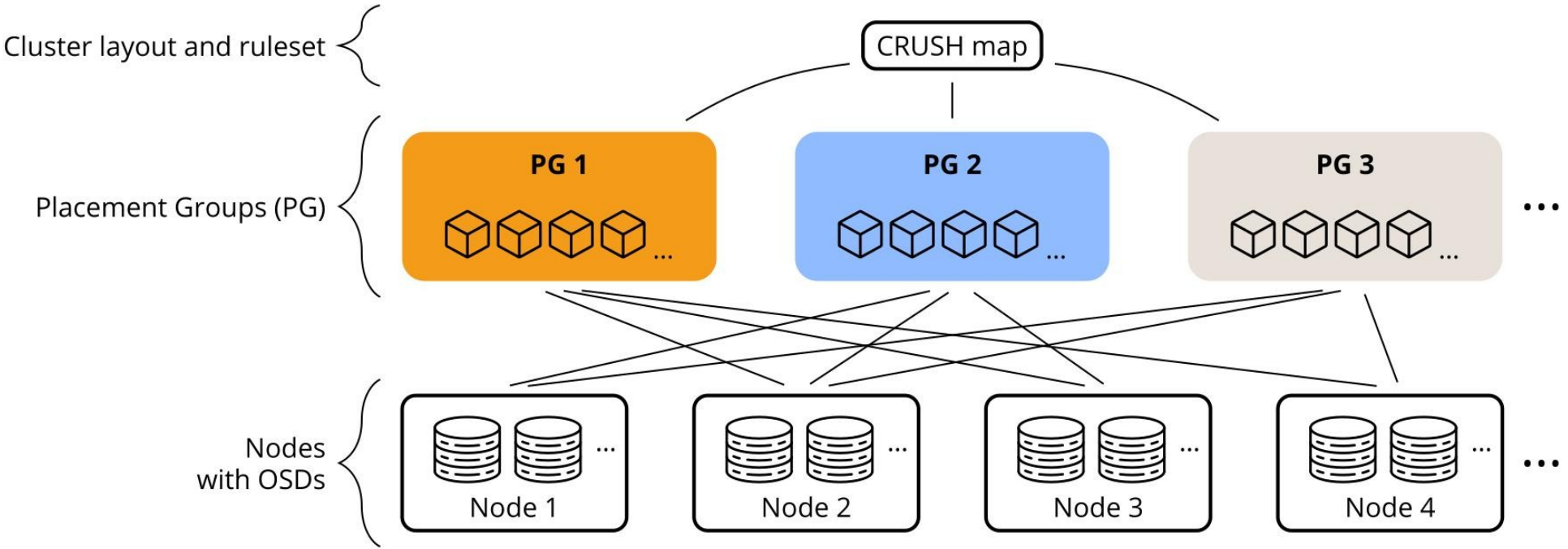
Ceph Data Organization



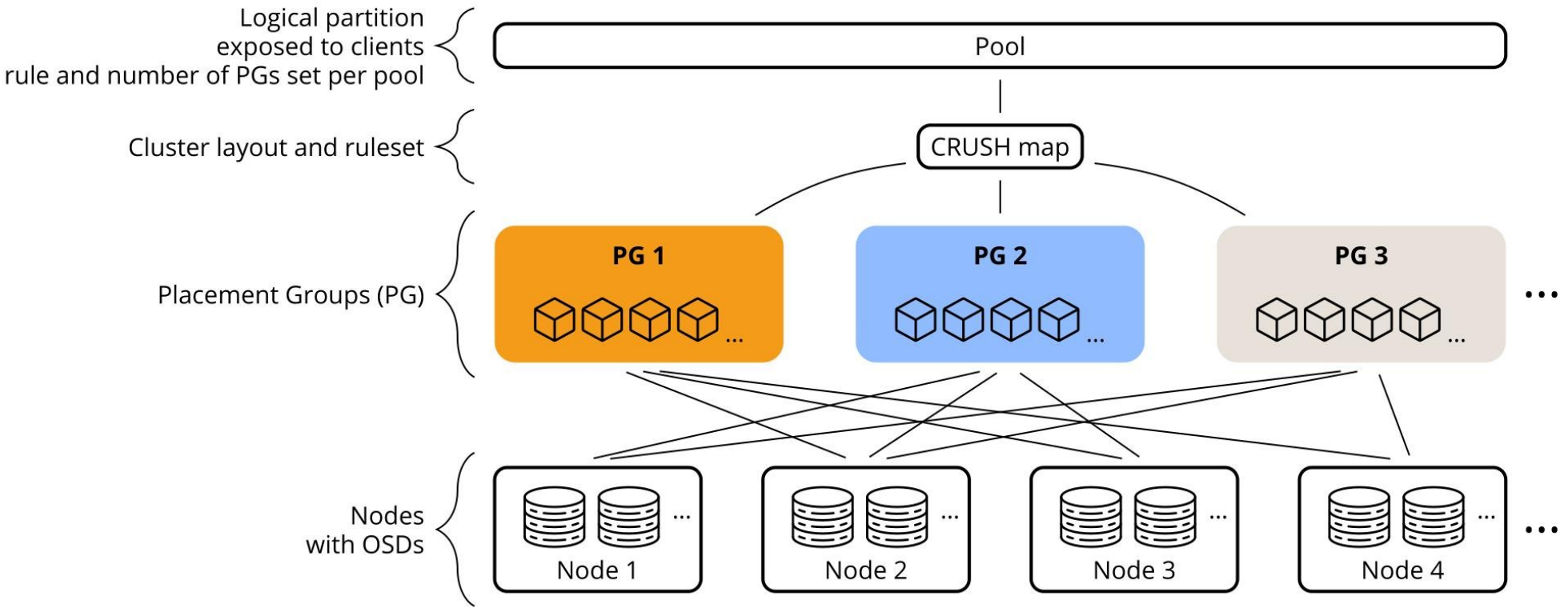
Ceph Data Organization



Ceph Data Organization

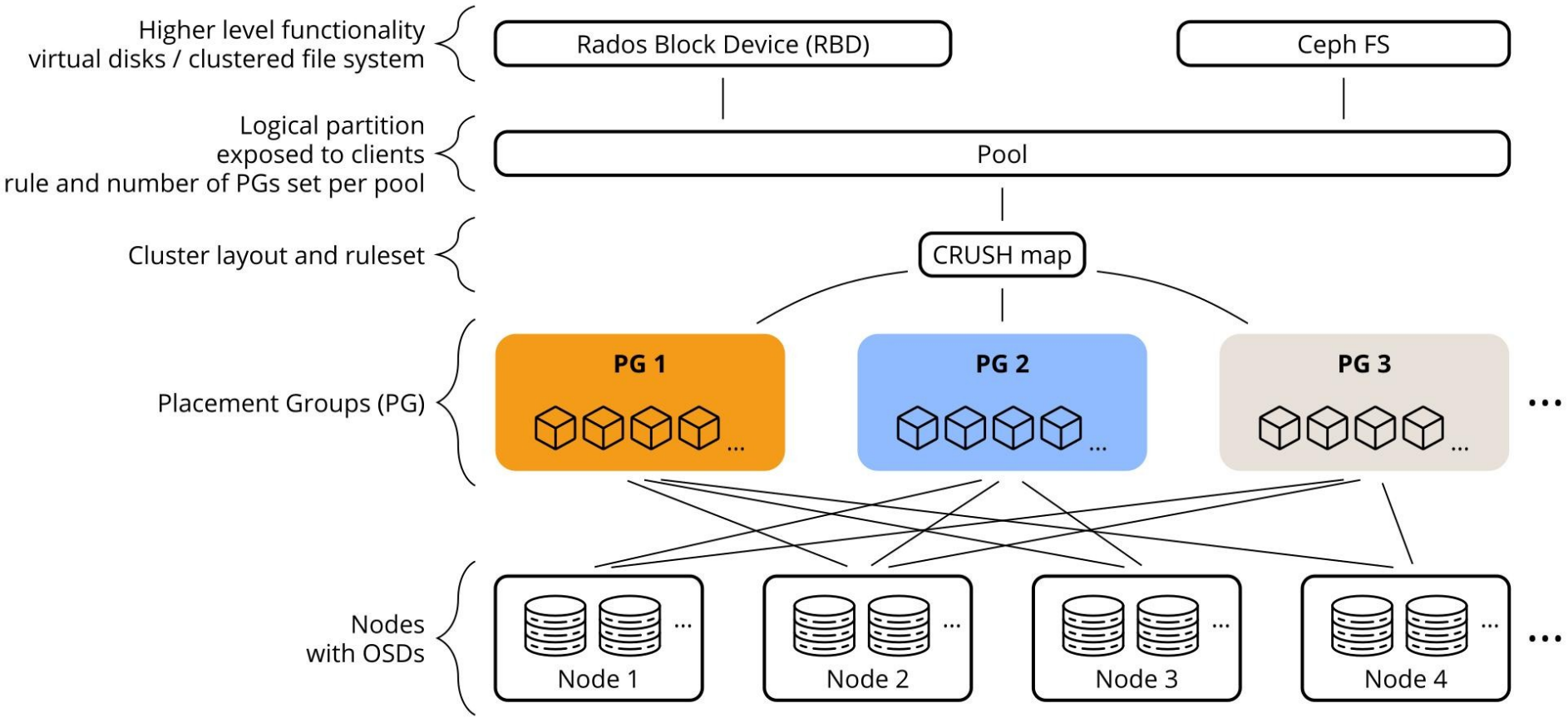


Ceph Data Organization



©inett GmbH -- info@inett.de -- www.inett.de

Ceph Data Organization

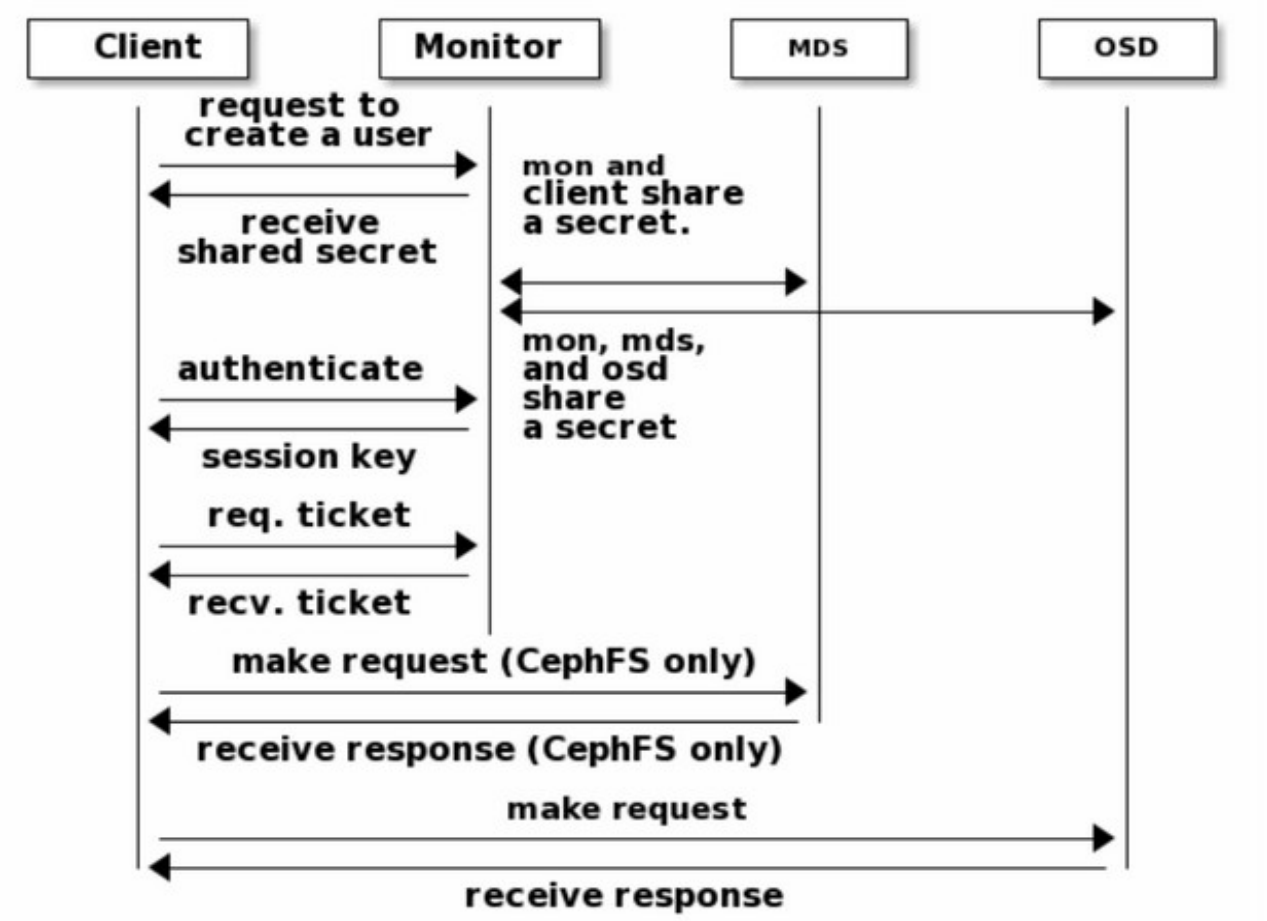


©inett GmbH -- info@inett.de -- www.inett.de

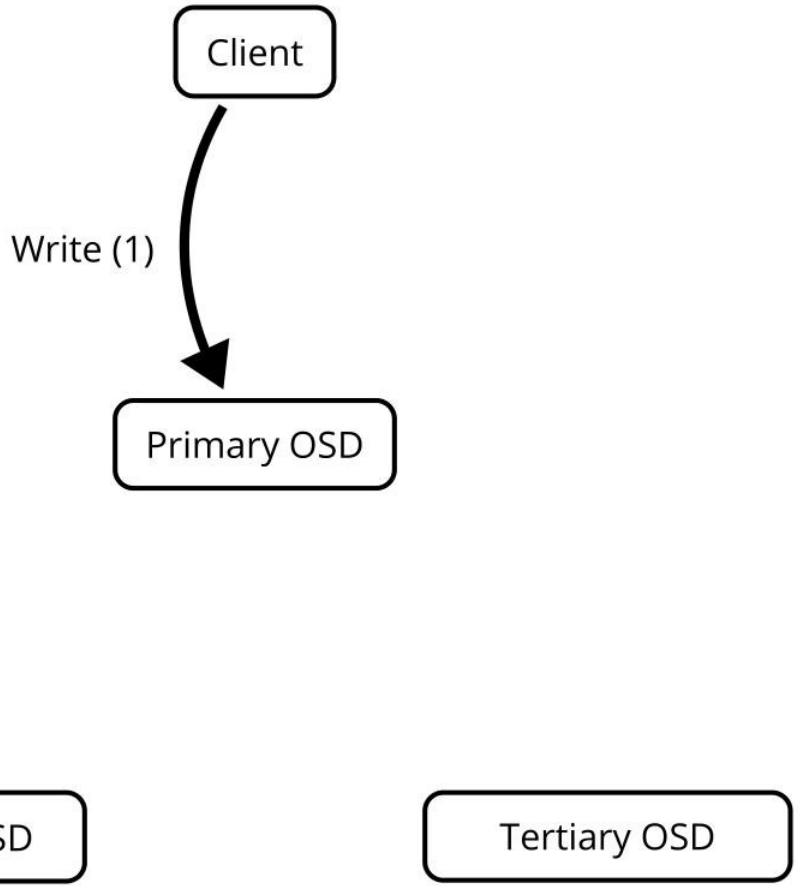
Placement Groups (PGs)

- Jeder Pool besitzt eigene Placement Groups (<poolid>.<pgid>)
- Ziel: 100 Placement Groups pro OSD (Standard=128)
- Zweck: Optimale Balance zwischen Objektredundanz und Deviceredundanz mit möglichst wenig Rechnerressourcen
- PG-Autoscaler optimiert die Anzahl der PGs im Betrieb

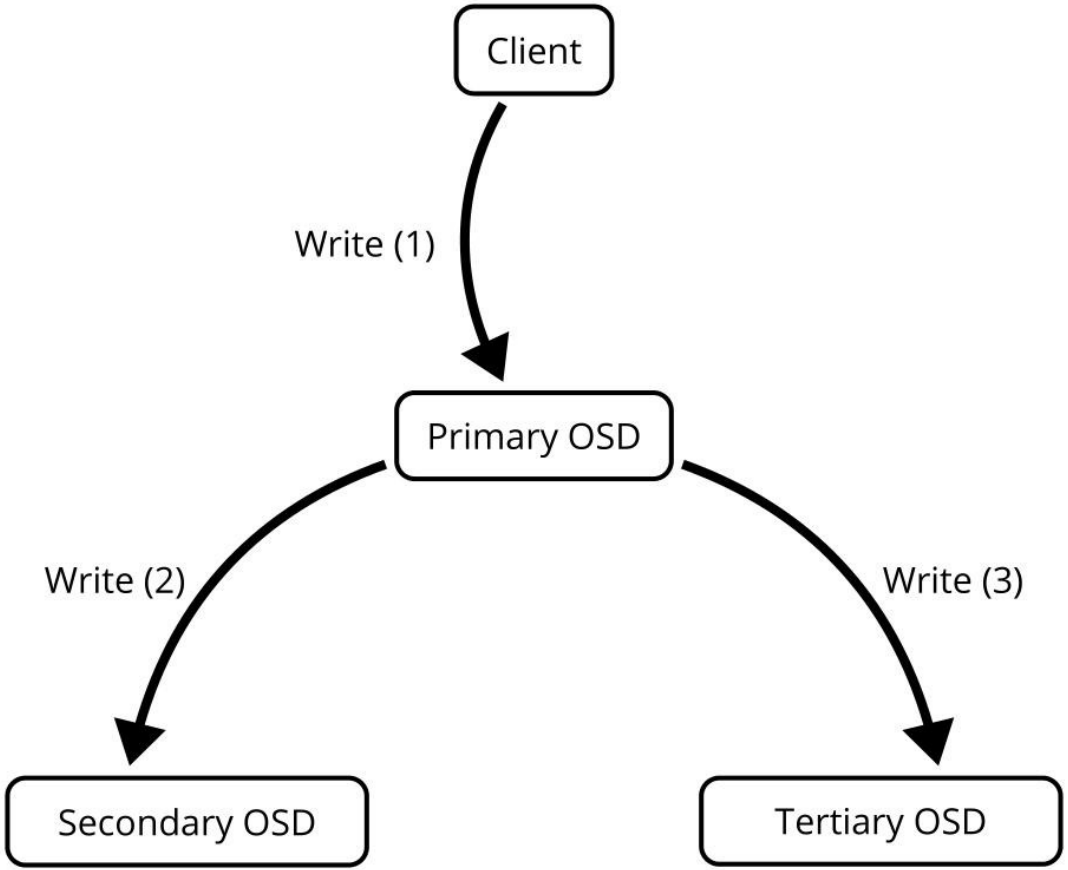
Ceph Requests



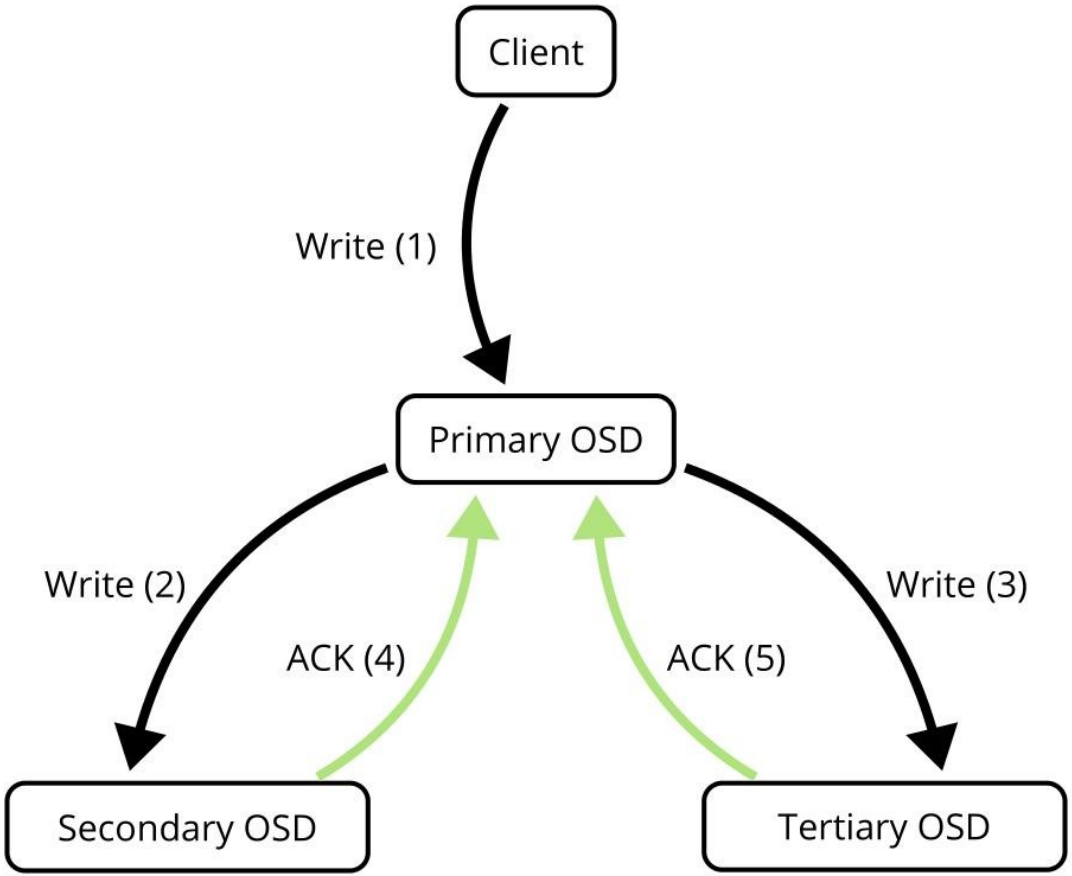
Ceph Schreibvorgang (Replicated Pool)



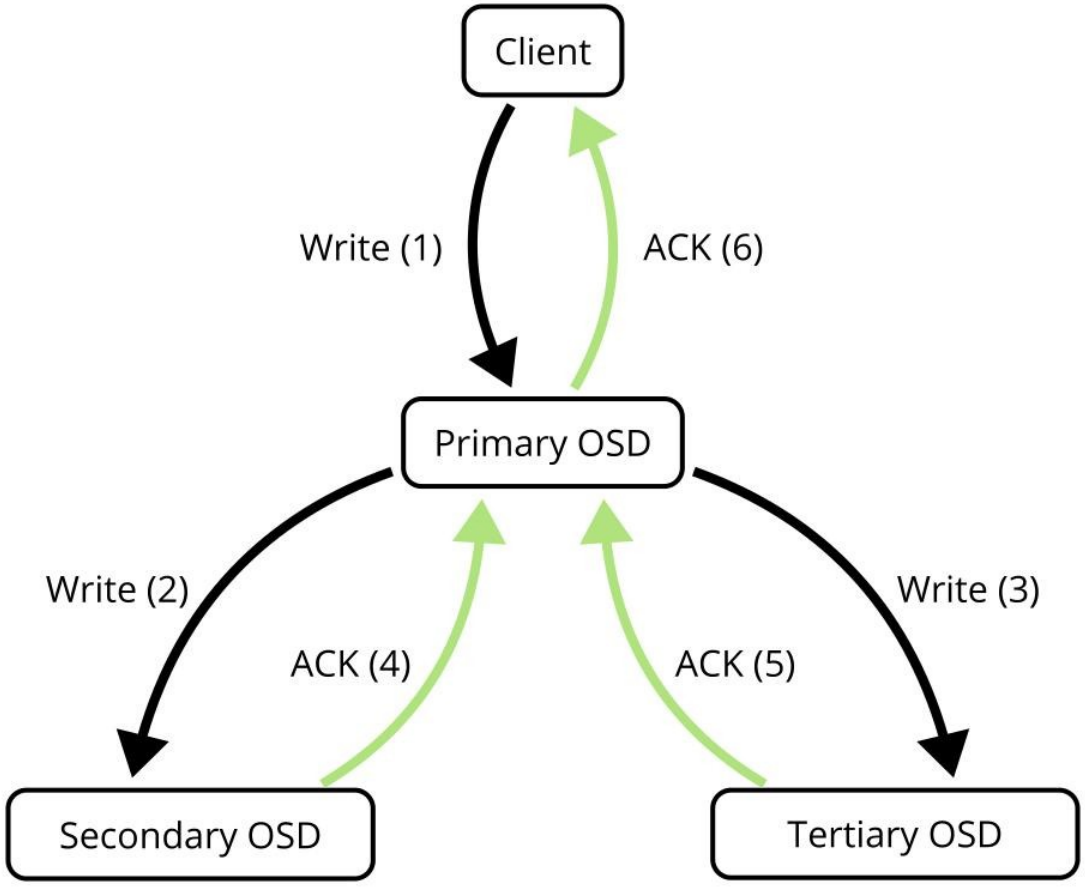
Ceph Schreibvorgang (Replicated Pool)



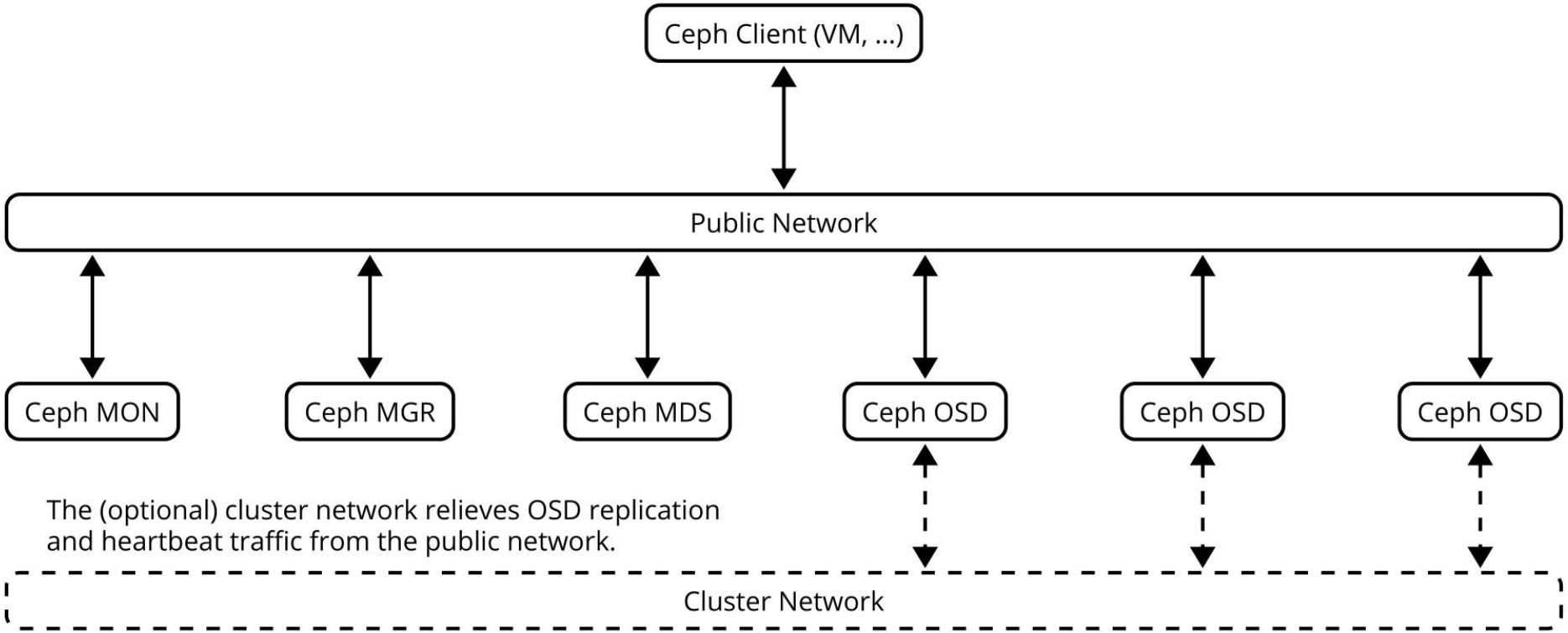
Ceph Schreibvorgang (Replicated Pool)



Ceph Schreibvorgang (Replicated Pool)

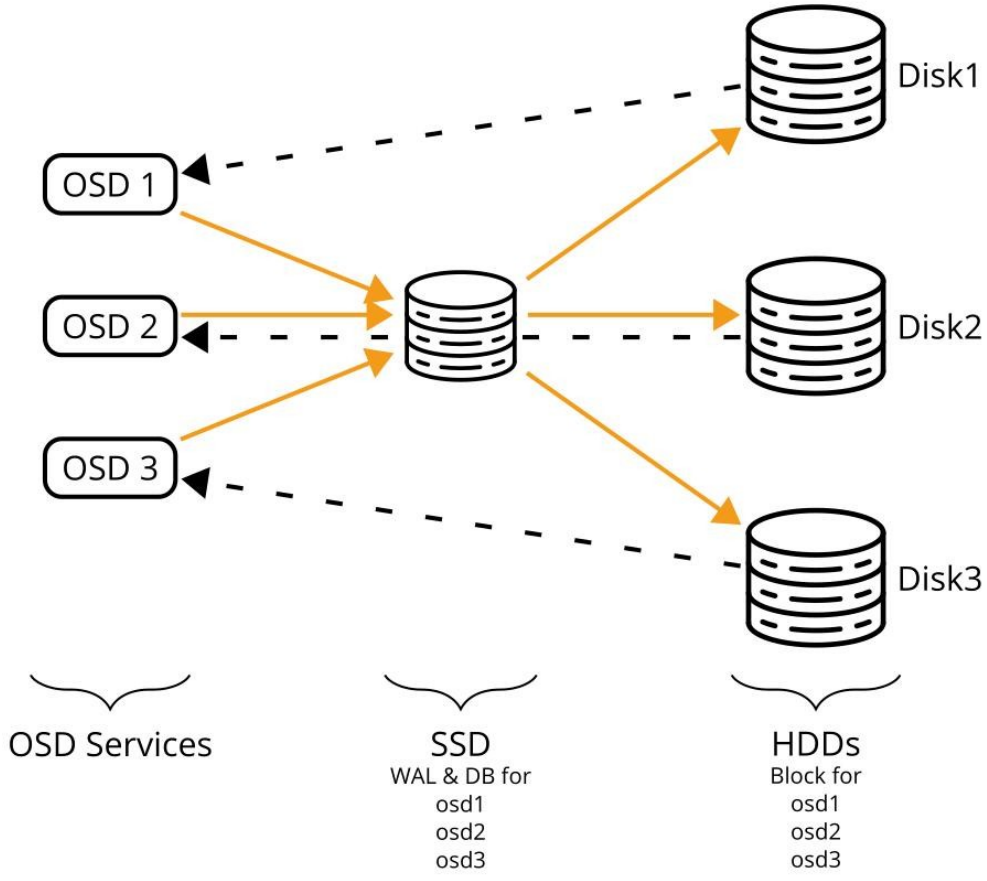


Ceph Netzwerk(e)



Ceph WAL/DB/Block auf SSD/NVMe

- **WAL:** Write Ahead Log.
- **DB:** OSD-interne Datenbank für Metadaten.
- **Block:** Wo die Daten tatsächlich gespeichert sind.
- Es ist möglich, WAL und DB verschiedener OSDs auf einen, schnelleren SSD/NVMe zu setzen.



Der CRUSH Algorithmus

- Ceph verwendet *keine* zentrale Lookup-Table, sondern den CRUSH Algorithmus zur Berechnung des Data-Placement
- Mit CRUSH kann jeder Client berechnen, auf welcher Disk Daten geschrieben und wo Daten gelesen werden müssen
- Bei einer size=3 enthält das reproduzierbare Ergebnis des Algorithmus' 3 OSD IDs
- Client kommuniziert mit erster OSD der Liste
- Bei Ausfall dieser OSD nimmt sich der Client die nächste OSD

CRUSH Rules (Placement-Regeln)

- Mit einer CRUSH Regel lässt sich das Placement beeinflussen
- Standardregel: 3 Replikate auf 3 verschiedenen Hosts
- Mögliche andere Regeln:
  - 3 Replikate in verschiedenen Racks
  - 2 Replikate in Raum A sowie 2 Replikate in Raum B
- Einfache Regeln per CLI mit `# ceph osd crush rule ...`
- Komplexe Regeln als Text-Config-Datei speichern: die ***crush Map***

Komplexe Regeln

- Bei komplexen Regeln ist es notwendig, die Crush Map manuell zu ändern, z.B. um eine Crush Regel anzupassen
- Vorgehensweise:
  - Crush Map aus MON als Binärdatei extrahieren
  - Binärdatei in Textdatei konvertieren
  - Konfiguration in Textdatei anpassen
  - Crush Map von Text- in Binärdatei kompilieren
  - Crush Map in MON importieren



©inett GmbH -- info@inett.de -- www.inett.de

Crush Map Änderung: Beispiel

```
# Datum in einer Variable angeben:
DATE=$(date +'%F')
# Crush Map in Binär Format dumpen:
ceph osd getcrushmap -o crushmap_original_${DATE}.bin
# Binäre Map in einer Textdatei konvertieren:
crushtool -d crushmap_original_${DATE}.bin -o crushmap_original_${DATE}.txt
# Zur Sicherheit eine Kopie vor der Bearbeitung machen  (zum vergleichen speichern):
cp crushmap_original_${DATE}.txt crushmap_neu_${DATE}.txt
# Änderung der CRUSH map:
nano crushmap_neu_${DATE}.txt
# Textdatei wieder im binären Format kompilieren:
crushtool -c crushmap_neu_${DATE}.txt -o crushmap_neu_${DATE}.bin
# Crush Map im Cluster wieder einspielen
ceph osd setcrushmap -i crushmap_neu_${DATE}.bin
```

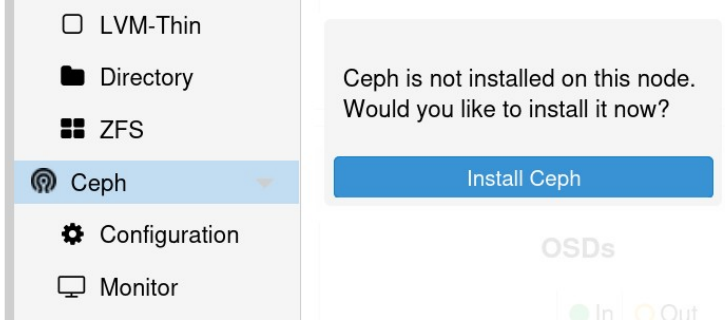
Proxmox Hyper-Converged Infrastructure (HCI) mit Ceph

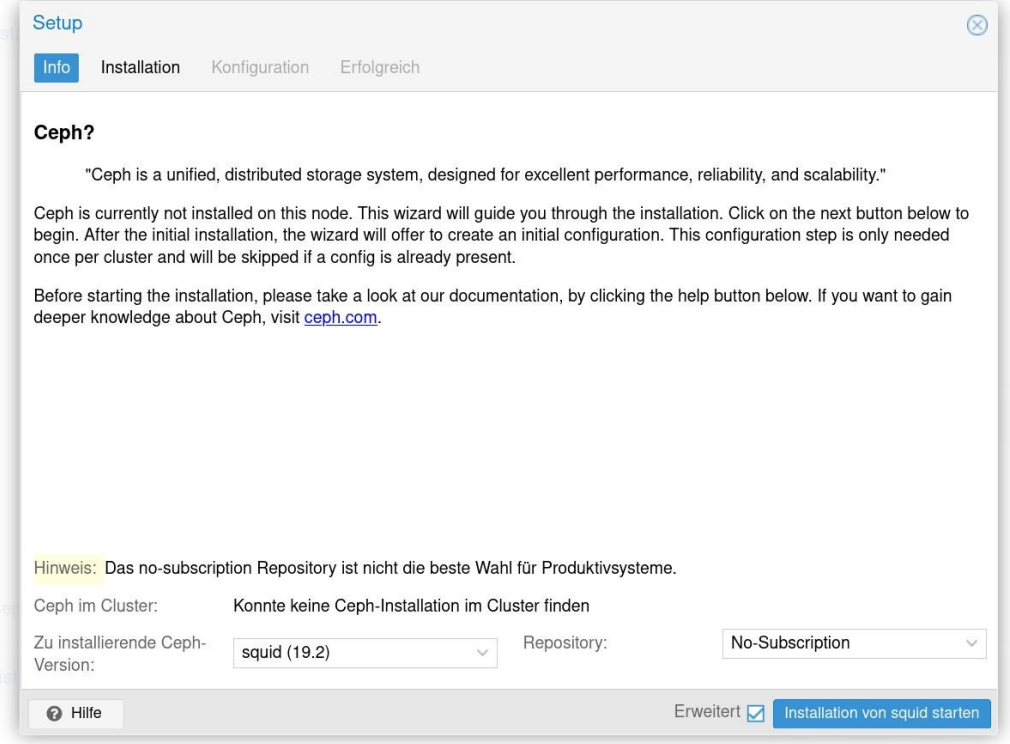


©inett GmbH -- info@inett.de -- www.inett.de

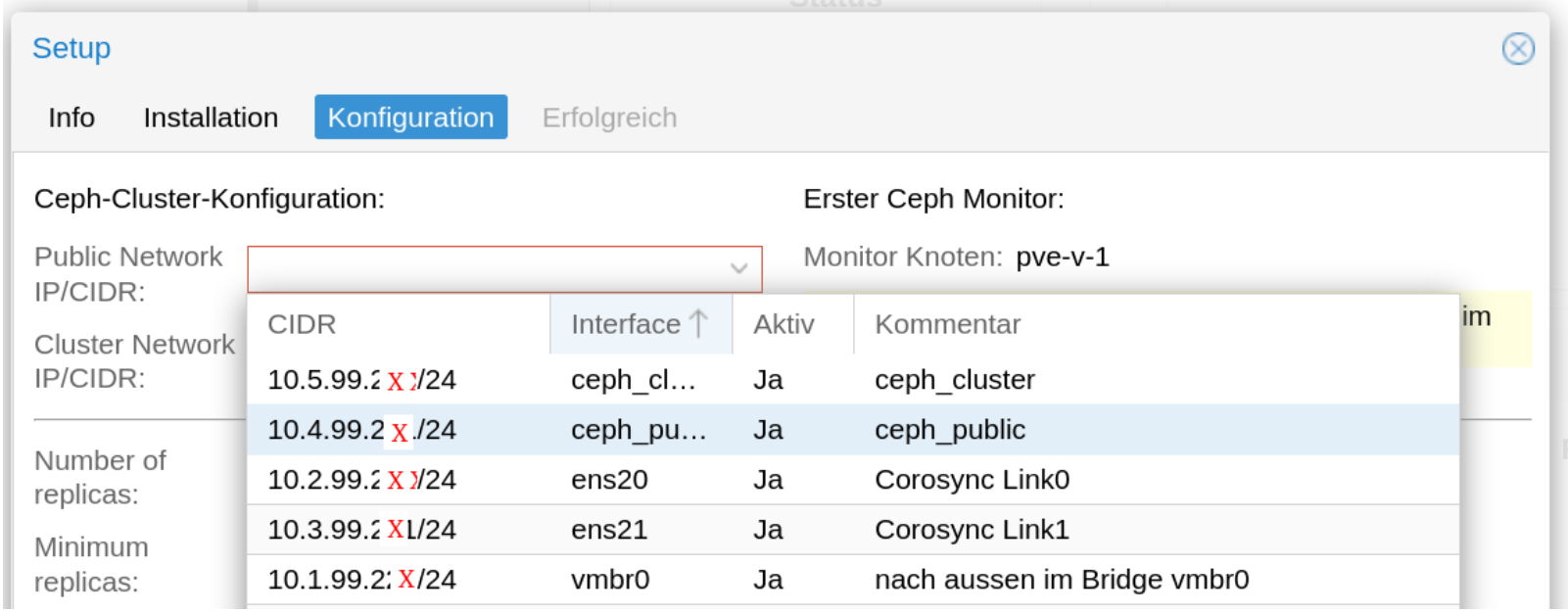
HCI mit Ceph Storage Cluster

- *Zunächst einmal muss Ceph auf jede Node installiert werden.*
- *No-subscription Repository und neueste Version wählen*





HCI mit Ceph Storage Cluster

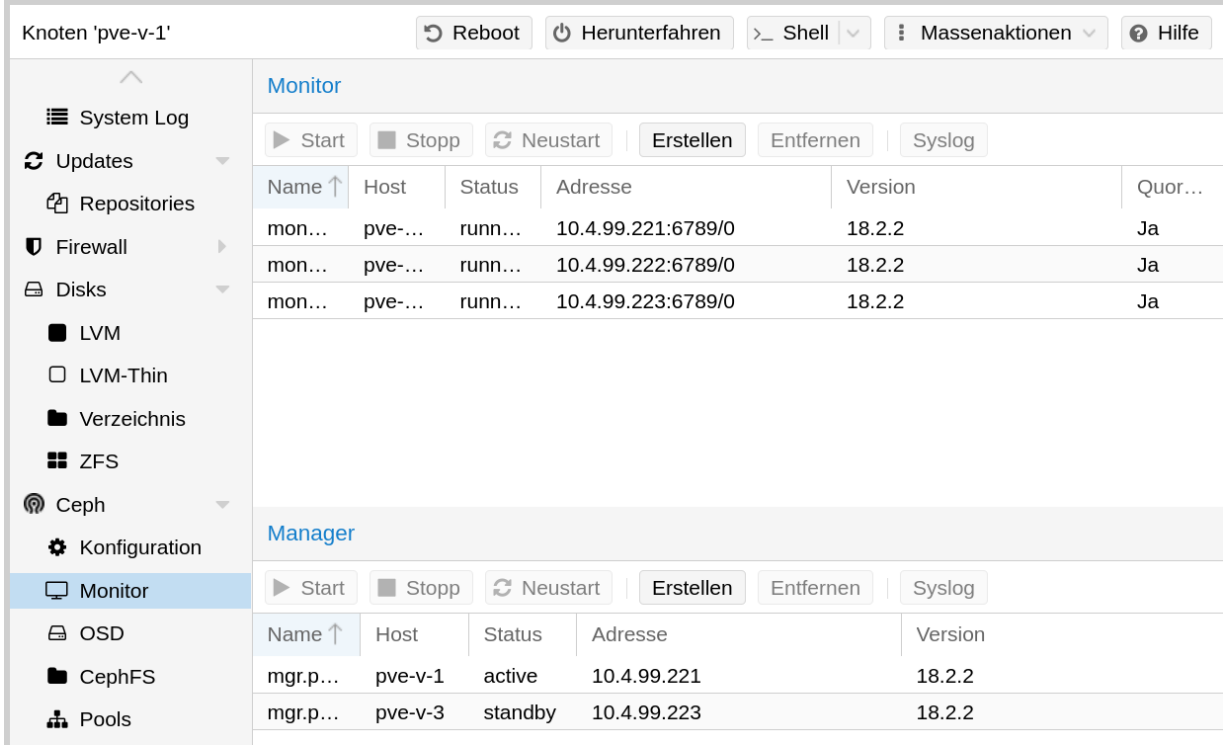


- Der erste Node wird installiert *und konfiguriert*, *dann erst* der zweite und der dritte eventuell über die CLI :

```
# pveceph install --version tentacle --repository no-subscription
```

HCI mit Ceph Storage Cluster

- 3 MONs und 2 Manager werden konfiguriert



HCI mit Ceph Storage Cluster

inett

- Dann werden *für jeden Knoten* jeweils *zwei* OSDs als *HDD* erstellt, und *zwei* als *SSD*.

Name	Klasse	OSD T…	Status	V…	weight	reweight	Ve…	Total	Apply/Co… Latency (…	PGs
default										
pve-x-3				19.2.0						
osd.11	ssd	bluestore	up / in	19.2.0	0,09769	1,00	3,18	100.00 GiB	0 / 0	21
osd.10	ssd	bluestore	up / in	19.2.0	0,09769	1,00	2,37	100.00 GiB	0 / 0	20
osd.9	hdd	bluestore	up / in	19.2.0	0,09769	1,00	16,31	100.00 GiB	0 / 0	33
osd.8	hdd	bluestore	up / in	19.2.0	0,09769	1,00	14,24	100.00 GiB	0 / 0	31
pve-x-2				19.2.0						
osd.3	ssd	bluestore	up / in	19.2.0	0,09769	1,00	3,10	100.00 GiB	0 / 0	19
osd.2	ssd	bluestore	up / in	19.2.0	0,09769	1,00	2,56	100.00 GiB	0 / 0	22
osd.1	hdd	bluestore	up / in	19.2.0	0,09769	1,00	18,44	100.00 GiB	0 / 0	35
osd.0	hdd	bluestore	up / in	19.2.0	0,09769	1,00	12,11	100.00 GiB	0 / 0	29
pve-x-1				19.2.0						
osd.7	ssd	bluestore	up / in	19.2.0	0,09769	1,00	2,70	100.00 GiB	0 / 0	21
osd.6	ssd	bluestore	up / in	19.2.0	0,09769	1,00	2,70	100.00 GiB	0 / 0	20
osd.5	hdd	bluestore	up / in	19.2.0	0,09769	1,00	12,25	100.00 GiB	0 / 0	27
osd.4	hdd	bluestore	up / in	19.2.0	0,09769	1,00	17,88	100.00 GiB	0 / 0	37

HCI mit Ceph Storage Cluster

- Nun können anhand der OSDs Pools erstellt werden.
- Wir erstellen einen *pool_HDD* und einen *pool_SSD*

Erstellen | Bearbeiten | Entfernen

Pool #	Name	Größe/min	# of Placeme…	Optimale # v…	Autoscale-Mo…	CRUSH Rule (ID)	Verwendet (%)
1	.mgr	3/2	1	1	on	replicated_rule (0)	2.27 MiB (0,00%)
2	ssd_Pool	3/2	32	8192	on	replicated_rule (0)	0 B (0,00%)
3	hdd__Pool	3/2	32	32	on	replicated_rule (0)	0 B (0,00%)
							2.27 MiB

Ceph: CLI Befehle

- `# ceph status` zeigt den Status / Zustand des Clusters allgemein an
- `# ceph health` zeigt nur die Gesundheit an
- `# ceph health detail` zeigt die Gesundheit detailliert an (und gibt evtl. Hinweise zur Fehlerbehebung)
- `# ceph report` generiert einen JSON Report über den Ceph Cluster
- `# ceph df` gibt einen interessanten Einblick über den belegten Festplattenplatz



©inett GmbH -- info@inett.de -- www.inett.de

HCI mit Ceph Storage Cluster

- Wir müssen auch Ceph konfigurieren, so daß er den *pool_ssd* für SSD-Festplatten verwendet, und den *pool_hdd* für HDD-Festplatten.
- Das geht mit sogenannten ***Crush rules***, Über die CLI.

```
# ceph osd tree
# ceph osd crush tree –show-shadow
# ceph osd crush rule ls
```

- *Ergibt die gleiche Infos als in der GUI, listet Device Classes und auch die Namen der bestehenden Regeln*

©inett GmbH -- info@inett.de -- www.inett.de

HCI mit Ceph Storage Cluster

inett

```
# ceph osd crush rule create-replicated ssd-rule default host ssd
```

1 2 3 4 5 6

- 1: Wir erstellen eine zusätzliche Regel
- 2: Die *Art* der Regel
- 3: Der *Name* dieser Regel
- 4: Was ist das *root* dieser Regel?
- 5: Welche *Failure Domain*: auf welcher Basis wollen wir unsere Daten verteilen?
- 6: Nur eine bestimmte *Device Class* wird verwendet → hier unsere SSDs.



HCI mit Ceph Storage Cluster

inett

- Eine zweite Regel wird für unsere HDDs erstellt
- Anschließend können wir die Regeln auflisten

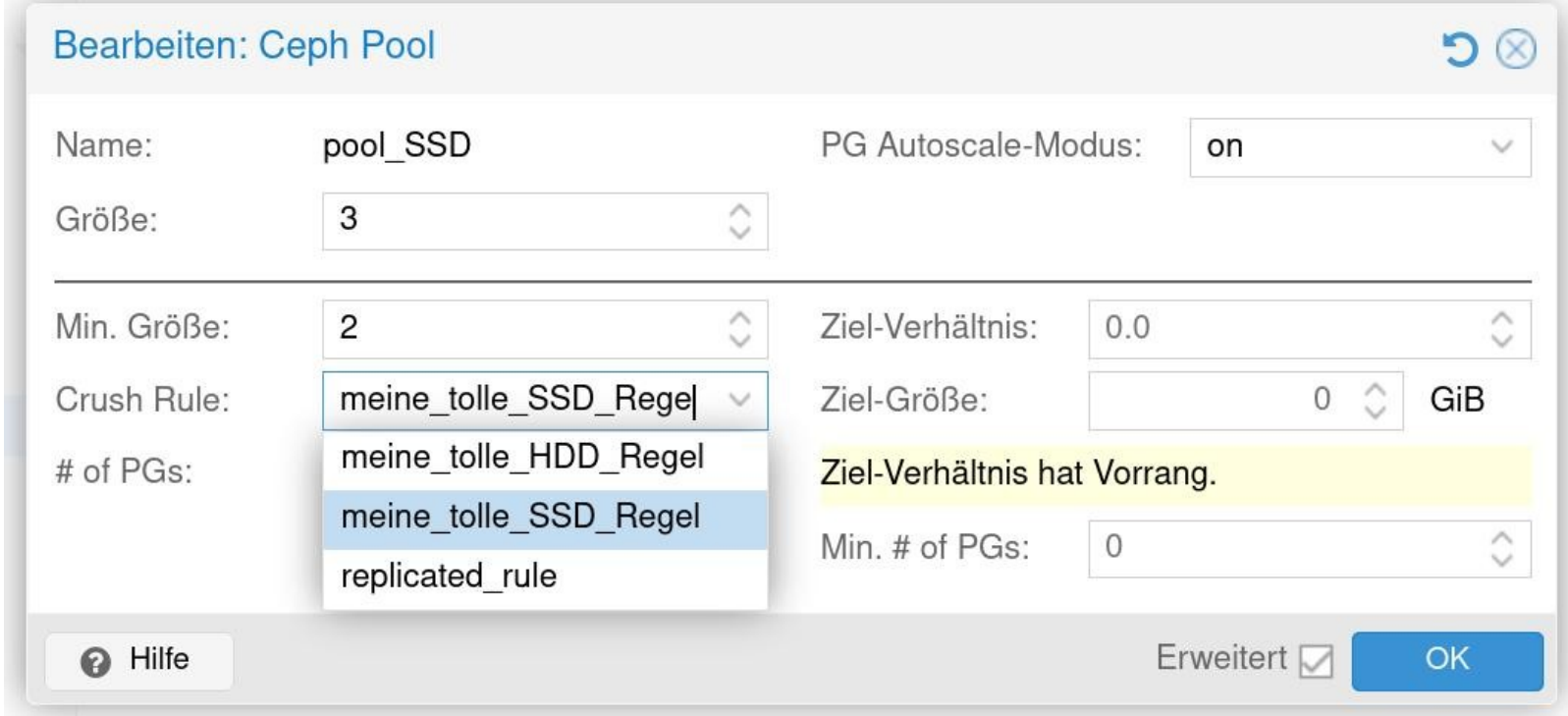
```
# ceph osd crush rule create-replicated hdd-Regel default host hdd
# ceph osd crush rule ls
```

- Die Proxmox Dokumentation ist dazu recht hilfreich:

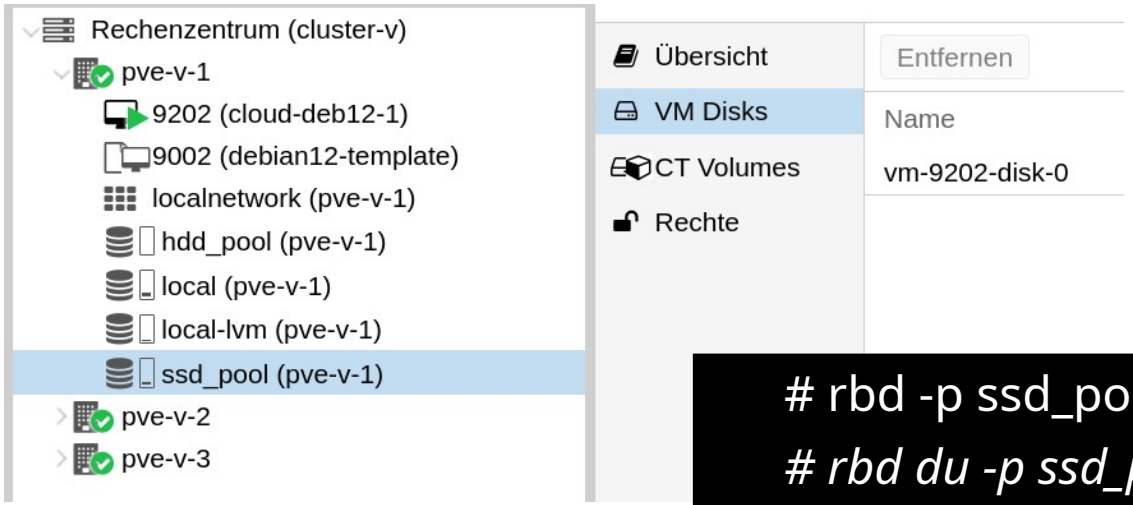
https://pve-X-1:8006/pve-docs/chapter-pveceph.html#pve_ceph_device_classes

HCI mit Ceph Storage Cluster

- Wir müssen nur noch dafür sorgen, dass die Pools die richtigen Regeln erhalten:



HCI mit Ceph Storage Cluster




```
# rbd -p ssd_pool ls
# rbd du -p ssd_pool
```

Bei der Migration meiner VM könnte *Thin Provisioning verloren gegangen sein. Um nicht verwendete Bereiche zurückzugewinnen:*

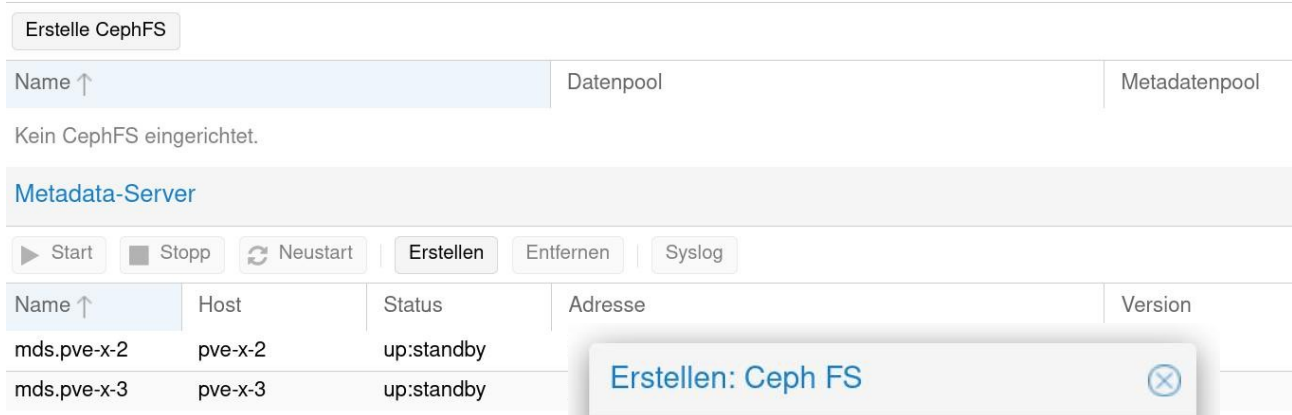
```
# rbd sparsify ssd_pool/vm9202-disk-0
```

HCI mit Ceph Storage Cluster

inett

CephFS:

- Zum Beispiel für ISO-Images, Container Templates…
- Benötigt zwei *Metadata (MDS) Server* → Active und Backup
- Dann kann unser *CephFS* erstellt werden




```
root@pve-x-1:~# df -h | grep cephfs
10.4.99.241,10.4.99.242,10.4.99.243,10.4.99.244:/  507G     0  507G   0% /mnt/pve/Mein_tolles_cephfs
root@pve-x-1:~#
```

Andere Arten der „Shared Storage“



Shared Storage mit NFS

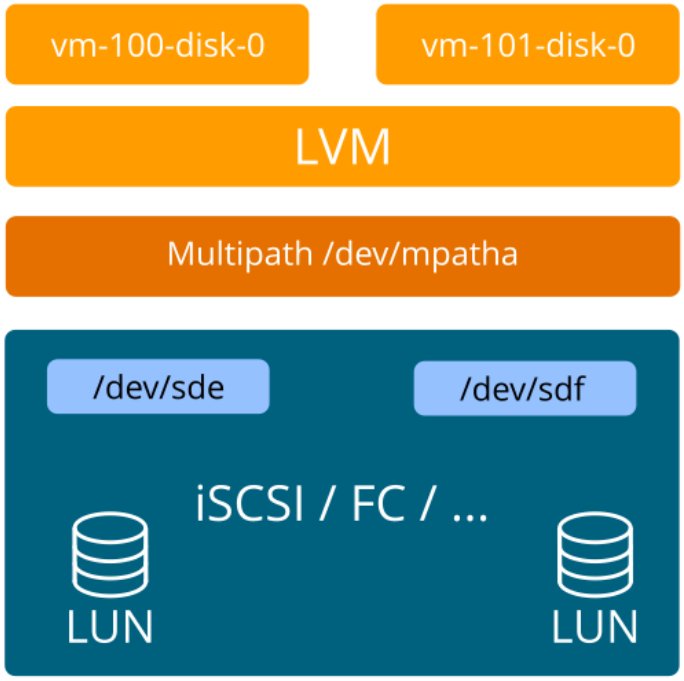
- NFS wird als filebasierte Storage eingemountet
- in der Regel etwas langsamer als iSCSI
- Dank qcow2 sind Snapshots innerhalb des Images möglich

```
root@pve-x-1:~# pvesm status
Name                   Type     Status           Total            Used       Available        %
Mein_tolles_cephfs      cephfs     active       531099648               0       531099648    0.00%
iso                        nfs     active      1030991872       386557952       591988736   37.49%
local                      dir     active        35679784         5948592        27886560   16.67%
local-lvm              lvmthin     active        44298240               0        44298240    0.00%
nfs                        nfs     active      1030991872       386557952       591988736   37.49%
pool_HDD                   rbd     active       265553344               0       265553344    0.00%
pool_SSD                   rbd     active       265551792               0       265551792    0.00%
root@pve-x-1:~#
```

Shared Storage mit iSCSI und FC

- Proxmox VE kann iSCSI und FC als Shared Block Storages anbinden
- Redundanz über Multipathing
- Hohe Geschwindigkeit
- Keine Snapshots möglich
- Snapshots ab PVE 9 als Volume-Chains möglich → aktuell Tech Preview
  - Performance Einschränkungen beachten
  - Jeder Snapshot benötigt noch mal Speicher für gesamtes Image

Multipathing



Proxmox VE Guest Migration



Proxmox VE Guest Migration

- KVM Live Migration:
  - Proxmox VE Cluster, min. 2 Nodes
  - Shared Storage wie SAN, NAS oder Distributed Storage (z.B. Ceph)
  - Local Storage möglich, gesamtes Disk Image wird dann migriert (dauert laaa...aaaange 😴)
  - Mit dem Datacenter Manager oder CLI auch über Cluster hinweg möglich
- Online und Offline Migration möglich
- Online Migration benötigt keine Downtime

ZFS Replizierung

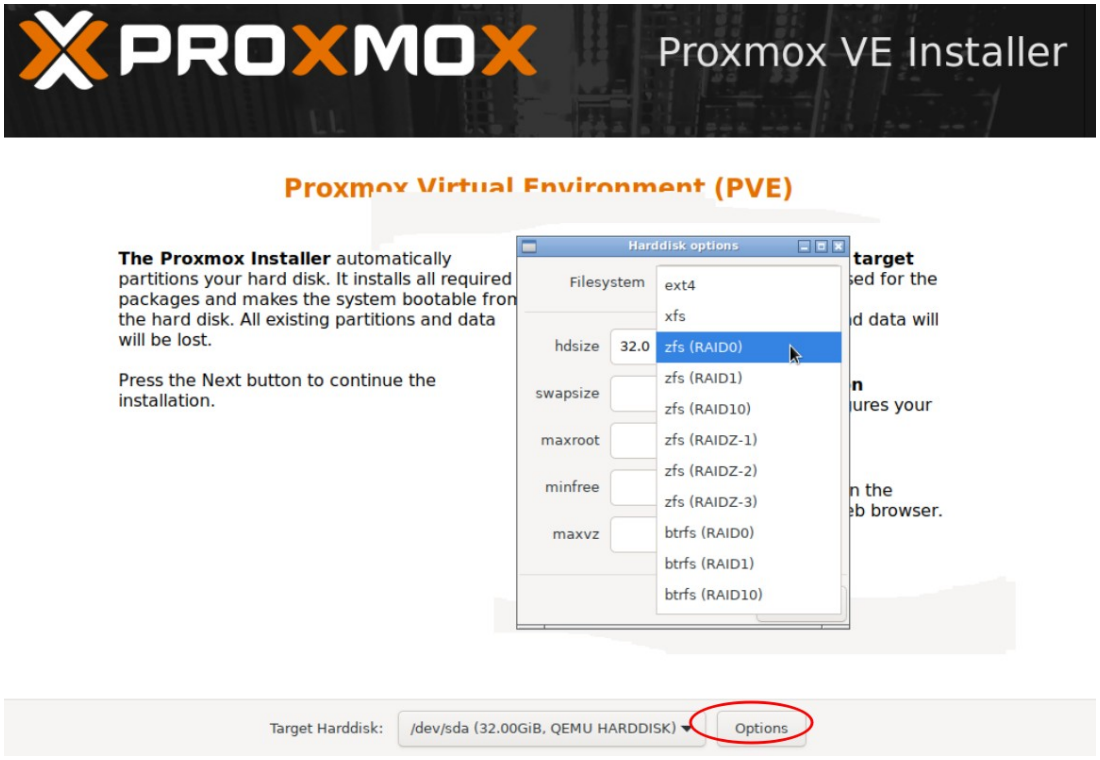


ZFS Storage



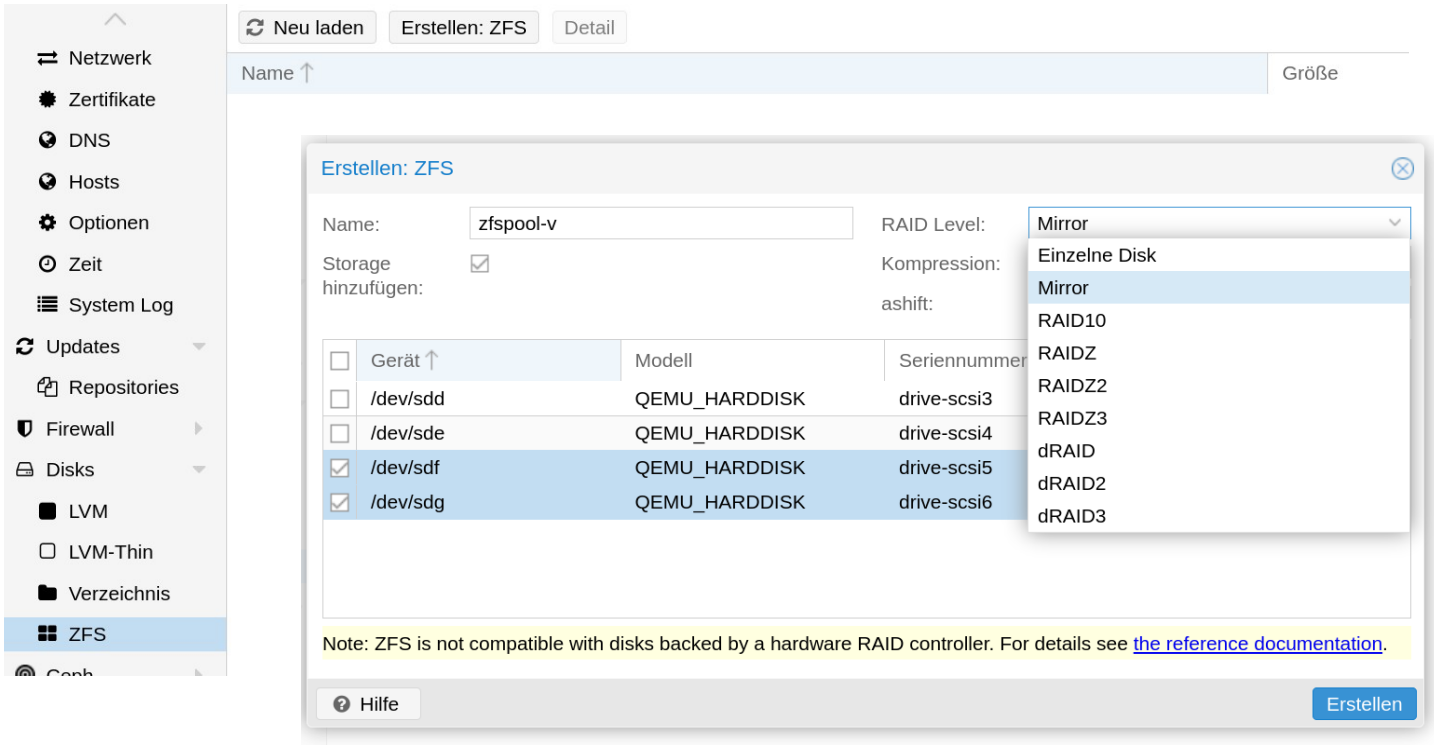
- Modernes 128 Bit Storage System
- *Umfasst* Device-, Pool-, Raid-, Volume- und POSIX Filesystem-Management
- Für *große* Datenvolumina/Server geeignet
- Permanente Integritätsprüfung und Selbstheilung
- Features wie Copy-On-Write, Snapshots, Metadaten auf SSDs, Komprimierung, Deduplizierung, asynchrone Replikation
- *Nicht kompatibel* mit HW RAID-Controller → nur HBAs nutzen

ZFS unter Proxmox VE



Proxmox VE 3.4 war die erste Linux-Distribution, die eine *native* Installation des root-FS über ZFS bietete.

ZFS unter Proxmox VE



Selbst wenn das System nicht auf ZFS installiert wurde, steht es als zusätzliche Storage für VMs und Container zur Verfügung

PVE ZFS Filesystem

- Über die CLI :
- ZFS Pool erstellen:

  # zpool create -o ashift=12 <pool_name> <device>

- „ashift“ sorgt für die richtige Ausrichtung (alignment) bei 4k Festplatten und SSDs.
- Komprimierung aktivieren / anpassen:

  # zfs set compression=lz4 <pool_name>

PVE ZFS Filesystem

- Raid0 (performant, aber unsicher)

  # zpool create <pool_name> <device1> <device2> ...

- Raid1 (Mirroring, 2 redundante Disks)

  # zpool create <pool_name> mirror <device1> <device2>

- Raid10 (4 Disks, redundant *und* performant)

  # zpool create <pool_name> mirror <device1> <device2>
  mirror <device3> <device4>

©inett GmbH -- info@inett.de -- www.inett.de

PVE ZFS Filesystem

- Raidz1 (3 Disks, redundant, optimierte Kapazität)

  # zpool create <pool_name> raidz1 <device1> <device2> <device3>

- Raidz2 (4 Disks, 2-Fach redundant)

  # zpool create <pool_name> raidz2 <device1> <device2> <device3>
  <device4>

- Raidz3 (5 Disks, 3-Fach redundant)

  # zpool create <pool_name> raidz3 <device1> <device2> <device3>
  <device4> <device5>

PVE ZFS Pool vergrössern

- zpool add <devices> zu einem bestehenden Pool:

```
  pool: rpool
 state: ONLINE
  scan: scrub repaired 0B in 0h35m with 0 errors on Sun Mar 11 00:59:39 2018
config:

        NAME                                            STATE     READ WRITE CKSUM
        rpool                                           ONLINE       0     0     0
          mirror-0                                      ONLINE       0     0     0
            scsi-3600508b1001c1bc00dd554d42a70a9fa      ONLINE       0     0     0
            scsi-3600508b1001cf286081f2e92715cfea7      ONLINE       0     0     0
```

- Zum Beispiel: zpool add rpool mirror /dev/sdg /dev/sdh

```
  pool: rpool
 state: ONLINE
  scan: scrub repaired 0B in 0h35m with 0 errors on Sun Mar 11 00:59:39 2018
config:

        NAME                                            STATE     READ WRITE CKSUM
        rpool                                           ONLINE       0     0     0
          mirror-0                                      ONLINE       0     0     0
            scsi-3600508b1001c1bc00dd554d42a70a9fa      ONLINE       0     0     0
            scsi-3600508b1001cf286081f2e92715cfea7      ONLINE       0     0     0
          mirror-1                                      ONLINE       0     0     0
            scsi-3600508b1001c3a216505dd5c5507a4bf      ONLINE       0     0     0
            scsi-3600508b1001c4f71ff646acecdd2aec0      ONLINE       0     0     0
```

PVE ZFS Cache & Log devices

- SSD Cache hinzufügen (beschleunigt lesen):

  # zpool add -f <pool_name> cache <device>

- SSD Log hinzufügen (beschleunigt schreiben):

  # zpool add -f <pool_name> log <device>

©inett GmbH -- info@inett.de -- www.inett.de

ZFS Special Devices

- Lagert Metadaten, und optional kleinere Dateien eines ZFS HDD Pools auf schnellere SSDs/NVMes aus
- Gleiche Redundanz wie Pool wird benötigt
- Kann in bestimmten Konfigurationen nicht mehr entfernt werden
- Bei Ausfall fällt *der gesamte Pool aus*
- Einsatz im Proxmox Backup Server um HDD Zugriffe zu beschleunigen

ZFS Special Devices

- ZFS Pool mit Special Device erzeugen:

  # zpool create -f <pool_name> mirror <device1> <device2> special mirror <device3> <device4>

- ZFS Pool mit Special Device erweitern:

  # zpool add <pool_name> special mirror <device1> <device2>

- Dateien bis 4K direkt aufs Special Device schreiben:

  # zfs set pool special_small_blocks=4K <pool_name>

ZFS Fehlerbehandlung

- Ausfall der Cache SSD ist unkritisch: ZFS liest dann (langsamer) von den normalen Festplatten.
- Ausfall der Log SSD ist unkritisch, sofern nicht gleichzeitig RAM defekt ist.
- Ein fehlerhaftes Device austauschen:

```
# zpool replace -f <pool_name> <old_device> <new_device>
```

ZFS Tipps n’ Tricks: zfs-auto-snapshot

inett

- Erstellt automatisch und regelmäßig Snapshots
- Löscht alte, abgelaufene snapshots wieder
- Zusätzlichen Speicherbedarf berücksichtigen

  ```
  # apt install zfs-auto-snapshot
  ```

- Erstellt viertelstündlich einen Snapshot aller Volumen, behält die 4 neuesten, verwirft ältere, labelt sie “PVE” :

  ```
  # crontab -e
  */15 * * * *  zfs-auto-snapshot –-quiet –-syslog –-label=PVE --keep=4
  ```

ZFS Tipps n’ Tricks: zfs-auto-snapshot

```
# zfs set com.sun:auto-snapshot=false rpool/swap
```

- Exkludiert rpool/swap beim Erstellen der automatischen Snapshots
- Rollback eines Snapshots:

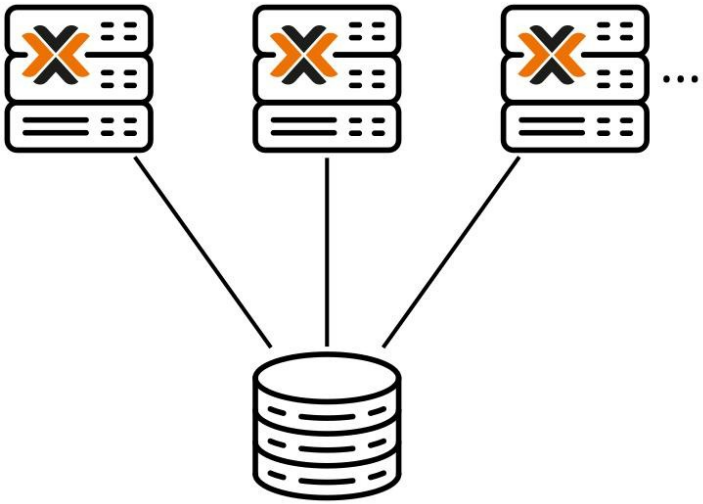
```
# zfs rollback rpool/data/vm-1337-disk-0@zfs-auto-snap_daily-2026-02-31-6996 [-r]
```

- -r wird benötigt, wenn der Snapshop nicht der Allerletzte ist – die zwischenzeitlich Erstellten werden dann gelöscht.

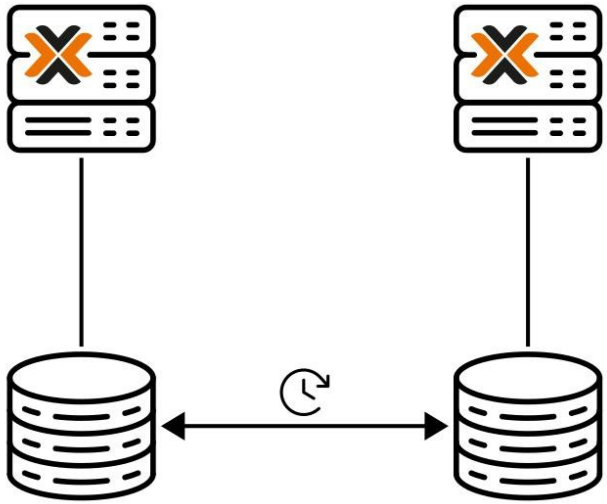
Proxmox VE
Storage Replication



PVE Storage Replication



- Alle Knoten sehen den gleichen ‚shared' Storage

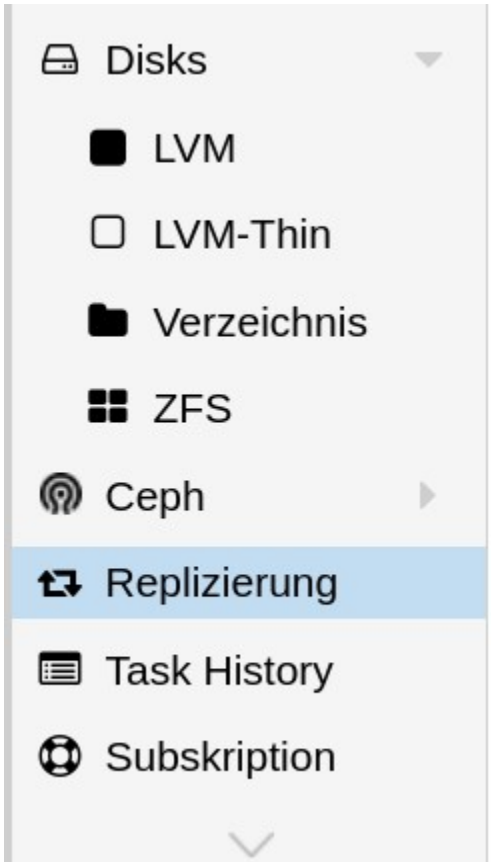


- Knoten sehen einen lokalen Storage, zyklisch repliziert (möglicherweise ältere Daten)

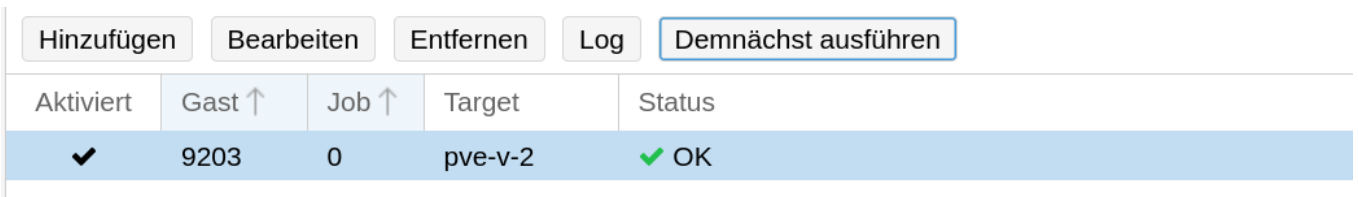
PVE Storage Replication

- Gut geeignet für ‚kleine‘ Setups aus zwei Servern
- Es liegt immer eine Kopie einer VM auf dem zweiten Server, und sie kann dort nach Verlust des ersten Server gestartet werden.
- Geht in beide Richtungen.
- ZFS muß auf beiden Hosts verwendet werden
- ZFS Storage / Poll *muß* den gleichen Namen tragen
- Sendet nach dem initialen Transfer nur noch Snapshots

PVE Storage Replication

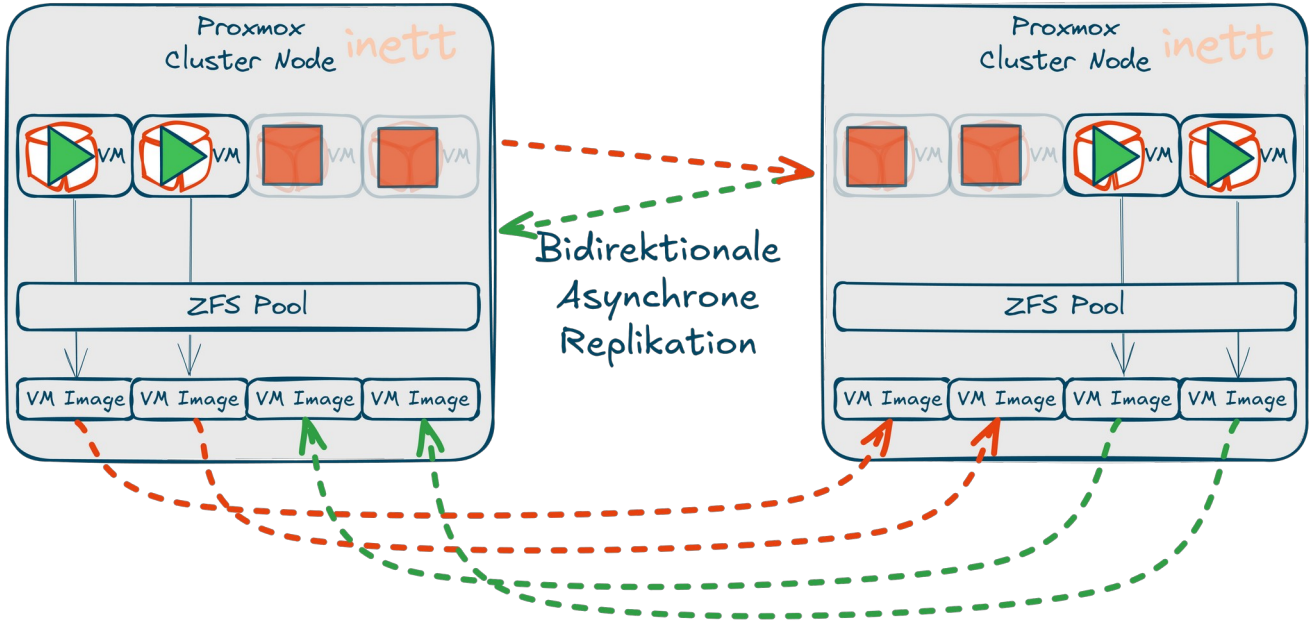






©inett GmbH -- info@inett.de -- www.inett.de

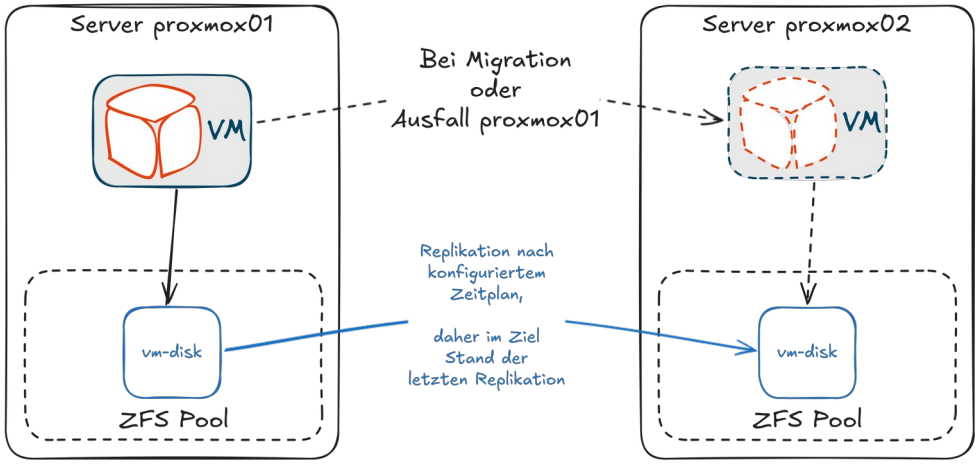
Bi-/Multidirektionale Replikation



> Asynchron nach flexiblem Zeitplan getaktet - Bsp: Alle Minute, Stunde, Tag, 15 Minuten tagsüber, Samstags
> Repliziertes Image daher nicht auf gleichem Stand (Recovery Point Objective/RPO > 0 und je nach Zeitplan)
> Für HA Anwendungen daher nur bedingt geeignet, z.B. bei relativ statischen Daten
> Erste Replikation voll, danach nur noch Änderungen
> Keine engen Voraussetzungen bei Latenz oder Bandbreite

PVE Storage Replication und HA

- PVE Storage Replication lässt sich mit HA verwenden
- Im HA Fall fehlen jedoch die Daten seit der letzten Replikation
- Potenziell droht damit Datenverlust



©inett GmbH -- info@inett.de -- www.inett.de

PVE High Availability (HA)



PVE High Availability

Was bedeutet High Availability?

- Definition von SLAs
- 99% Verfügbarkeit ist *normal*
- HA beginnt hinter dem Komma (99,9%)
- Definition von Verfügbarkeitsklassen
  → siehe auch Wikipedia :

https://de.wikipedia.org/wiki/Hochverf%C3%BCgbarkeit

Quelle: Strategic Research Corp.

SYSTEM AVAILABILITY METRICS

System Type	Unavailability (min./year)	System Availability (% of 100% uptime)	Availability Class
Unmanaged	50,000	90.0%	1
Managed	5,000	99.0%	2
Well Managed	500	99.9%	3
Fault-Tolerant	50	99.99%	4
High Availability	5	99.999%	5
Very High Availability	0.5	99.9999%	6
Ultra-Availability	0.05	99.99999%	7

Strategic Research Corp.

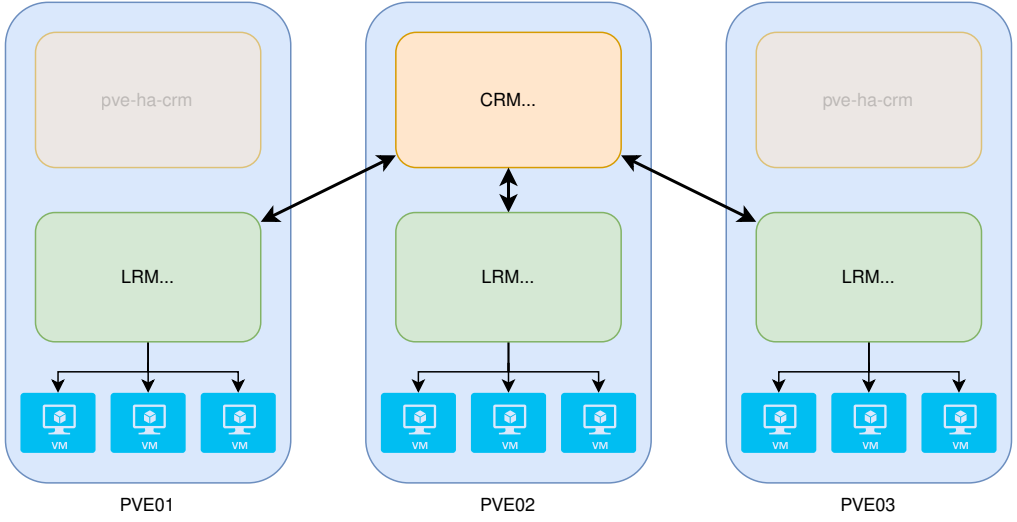
HA Cluster Services: PVE HA Manager

- **crm-manager (pve-ha-crm)**
  - CRM = Cluster Ressource Manager
  - kennt alle Ressourcen des Clusters und ihren Status
  - steuert alle Local Ressource Manager
  - einmal im Cluster aktiv
  - fällt die Node mit dem Master-CRM aus, wird eine neue Master-CRM Node von den restlichen Nodes bestimmt
- **lrm-manager (pve-ha-lrm)**
  - LRM = Local Ressource Manager
  - steuert lokale Ressourcen einer Node
  - meldet Status an CRM

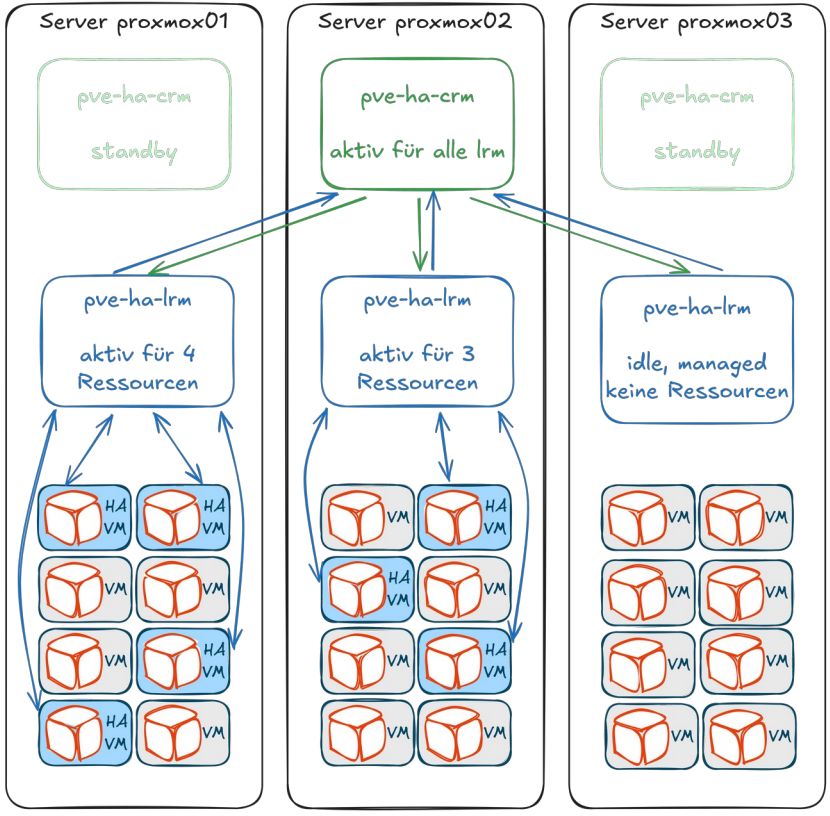
©inett GmbH -- info@inett.de -- www.inett.de

Cluster Services: PVE HA Manager

- **crm-manager** (pve-ha-crm) **steuert** lrm-manager
- **lrm-manager** (pve-ha-lrm) **führt aus und meldet** an crm-manager



Beispiel: Aktive CRM/LRM



PVE High Availability: Einführung in HA

1) Ist eine VM als HA konfiguriert
2) und der physische Server fällt aus
3) wird die VM auf eine andere Node migriert
4) und erneut gestartet

ZIEL: Verringerung einer Downtime durch *eigenständiges* Handeln des Cluster Stacks

©inett GmbH -- info@inett.de -- www.inett.de

HA Voraussetzungen

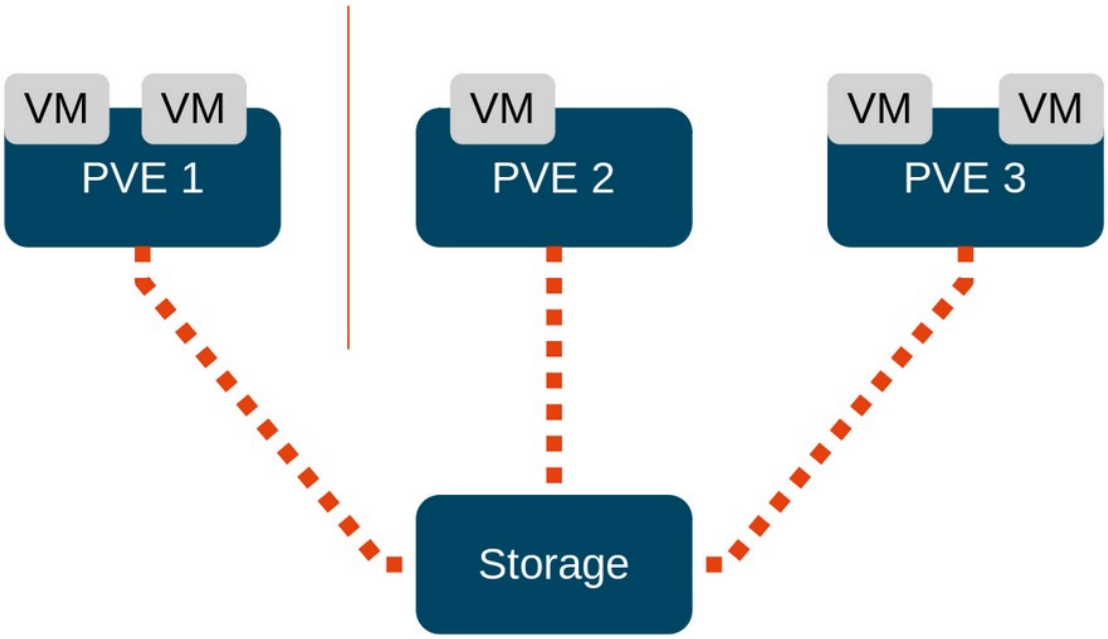
- Server Hardware ohne SPOF, z.B. durch die Anwendung von:
  - RAID oder ZFS mirroring
  - Mehrere NICs
  - Redundante Stromversorgung
  - USV
- 3 Nodes (2 Nodes + QDevice auch möglich)
- NAS, SAN oder verteiltes Storage wie z.B. Ceph RBD

HA Voraussetzungen

- Zuverlässige Netzwerkverbindung zur Clusterkommunikation
- Gleiches Datum/Uhrzeit auf allen Nodes (ntp)
- Latenz zwischen Nodes unter 5 ms

©inett GmbH -- info@inett.de -- www.inett.de

HA Relocation & Restart



©inett GmbH -- info@inett.de -- www.inett.de

Fencing

- Vermeiden einer "*Split Brain*“ Situation und Sicherstellen von Datenintegrität (HA VM/CT/Service darf nur einmal laufen)
- Verhindert, daß zwei Nodes gleichzeitig auf gleiche Daten zugreifen, und Daten zerstören

Fencing

- Proxmox VE HA Manager verwendet standardmäßig “Self Fencing” durch Hardware oder Kernel Watchdog
- Fencing garantiert Datenintegrität unter allen Umständen
- Fencing ist *essentiell* in HA Umgebungen
- Ohne Fencing funktioniert HA *nicht zuverlässig*

Self Fencing

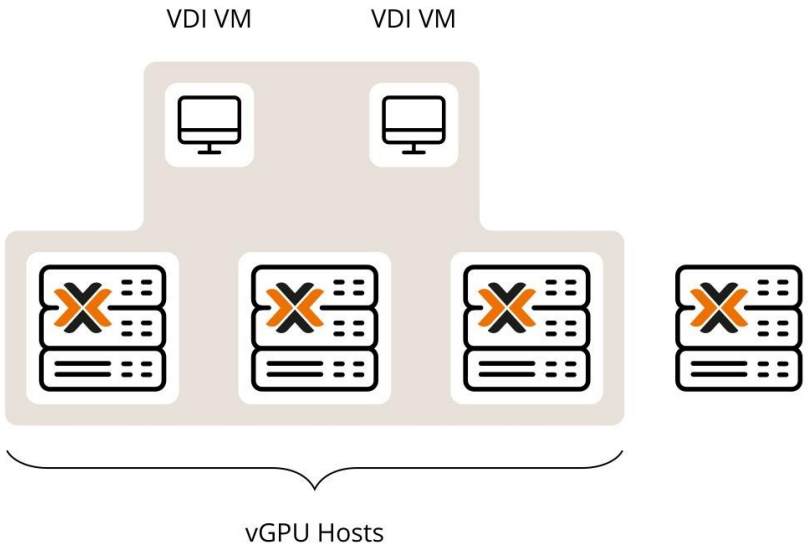
- Self Fencing funktioniert "*out-of-the-box*"
- Vor Produktionsbetrieb testen
- Load Tests, Backups, Live Migration, Software Update Tests, etc.
- "*If you think you are finished, do final tests*"
- HA verstehen hilft, im Fehlerfall ruhig zu bleiben

HA Cluster Affinitäten

- Bis PVE 8.x: HA-Gruppen
- Ab PVE 9.x: HA-Affinitätsregeln

  (Affinity Rules)

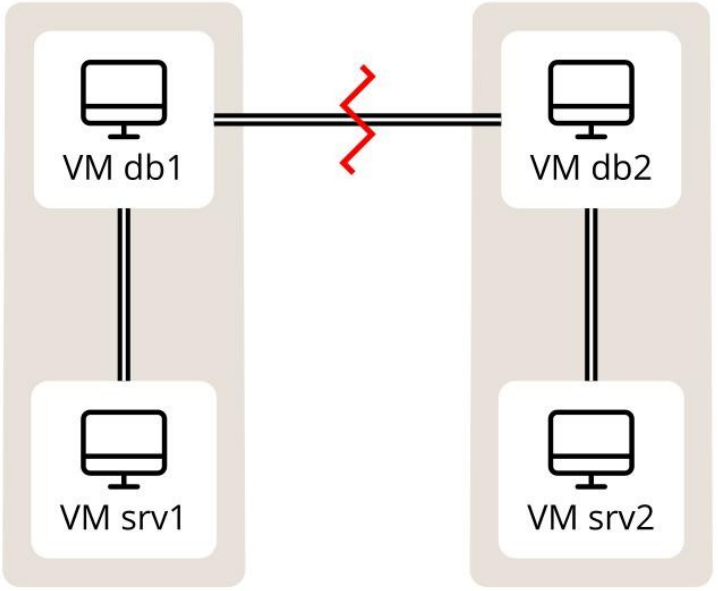
  - HA *Node* Affinitätsregeln ersetzen HA-Gruppen und werden bei einem Upgrade automatisch konvertiert
  - Damit werden bestimmte VMs an bestimmte Nodes angebunden, z.B. wenn bestimmte Ressourcen nicht clusterweit verfügbar sind (hier vGPU)



©inett GmbH -- info@inett.de -- www.inett.de

HA Cluster Affinitäten

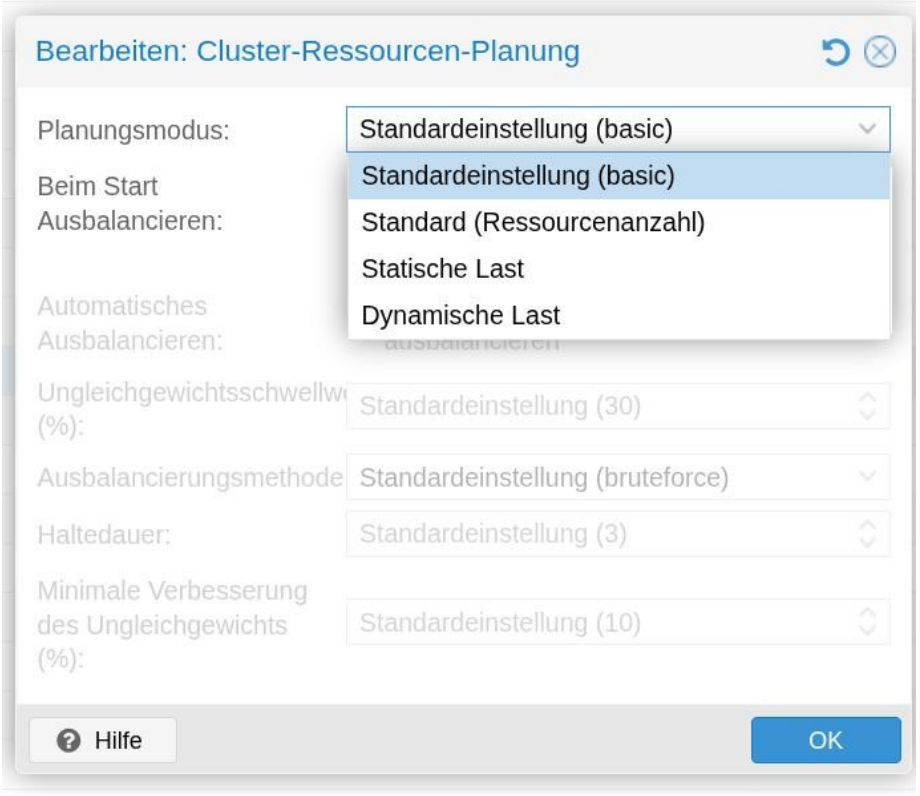
- HA *Resource* Affinitätsregeln ermöglichen Affinitäten zwischen Vms
- Positive Affinität: VMs auf *gleicher* Node halten
- Negative Affinität: VMs auf *unterschiedliche* Nodes verteilen



©inett GmbH -- info@inett.de -- www.inett.de

HA Cluster Lastverteilung

- Unter Rechenzentrum → Options → Cluster-Ressourcen-Planung kann die Lastverteilung konfiguriert werden
- Ressourcenanzahl ist die Standard Einstellung: je nach *Anzahl* der VM pro Node werden VM migriert und möglichst gleichmäßig im Cluster verteilt
- Die Einstellung muß bei Jeder HA-Ressource eingehakt worden sein.



PVE Benutzerverwaltung



PVE Benutzerverwaltung

- Benutzer:
  - Besteht aus zwei Teilen: ‘<userid>@<realm>‘
  - Login Screen zeigt separate Felder an
  - Intern nur zusammen verwendet und gespeichert (/etc/pve/user.cfg)
- Administrator:
  - Der Superuser/Administrator ist 'root@pam‘
  - System Mails gehen an hinterlegte Mailadresse

©inett GmbH -- info@inett.de -- www.inett.de

PVE Benutzerverwaltung

Gruppen

- Benutzer kann mehreren Gruppen angehören
- Berechtigungen können und *sollten* an Gruppen vergeben werden
- Kürzere Berechtigungslisten
- Einfachere Administration

©inett GmbH -- info@inett.de -- www.inett.de

Proxmox VE Benutzerverwaltung

- In /etc/pve/user.cfg :

```
root@proxmox01-        :~# cat /etc/pve/user.cfg
user:checkmk@pve:1:0:Checkmk:::::
user:f    l@inett.de:1:0:Fabian     ::f    l@inett.de:::
user:gdelanoy@inett.de:1:0:Guillaume Delanoy::gdelanoy@inett.de:::
user:mab    @inett.de:1:0:Marcel B    ::mab    @inett.de:::
user:mgabriel@inett.de:1:0:Marco Gabriel::mgabriel@inett.de:::
user:mgabriel@pve:1:0::::::
user:ph    @inett.de:1:0:Pascal H    ::ph    @inett.de:::
user:pt       @inett.de:1:0:Peter T       ::pt       @inett.de:::
user:root@pam:1:0:::admin@inett.de:::
user:ts      @inett.de:1:0:Tobias S          ::ts      @inett.de:::

group:Admins:f    @inett.de,gdelanoy@inett.de,mab    @inett.de,mgabriel@inett.de,

pool:prod::112,113,118,701,702,703,901,902,903,905,908,909,910,911,913,914,971,97


acl:1:/:@Admins:Administrator:
acl:1:/:checkmk@pve:PVEAuditor:
```

Proxmox VE Benutzerverwaltung

- Passwörter werden hier *nicht* gespeichert
- Ausnahmen :
  - @pam → in der Systemdatei /etc/shadow
  - @pve → in /etc/pve/priv/shadow.cfg

- Benutzer sind stattdessen mit den *Authentifizierungsrealms* verknüpft.

- Daher wird ein Benutzer oft intern durch ihren Benutzernamen und Realm in Form von <*BenutzerID*>@<*Realm*> identifiziert.

PVE Benutzerverwaltung

Authentication Realms

- Unterstützung mehrerer Authentifizierungsquellen:
  - Linux PAM
  - Integrierter Proxmox VE Authentication Server „@pve“
  - Active Directory
  - LDAP
  - OpenID Connect für Single Sign On

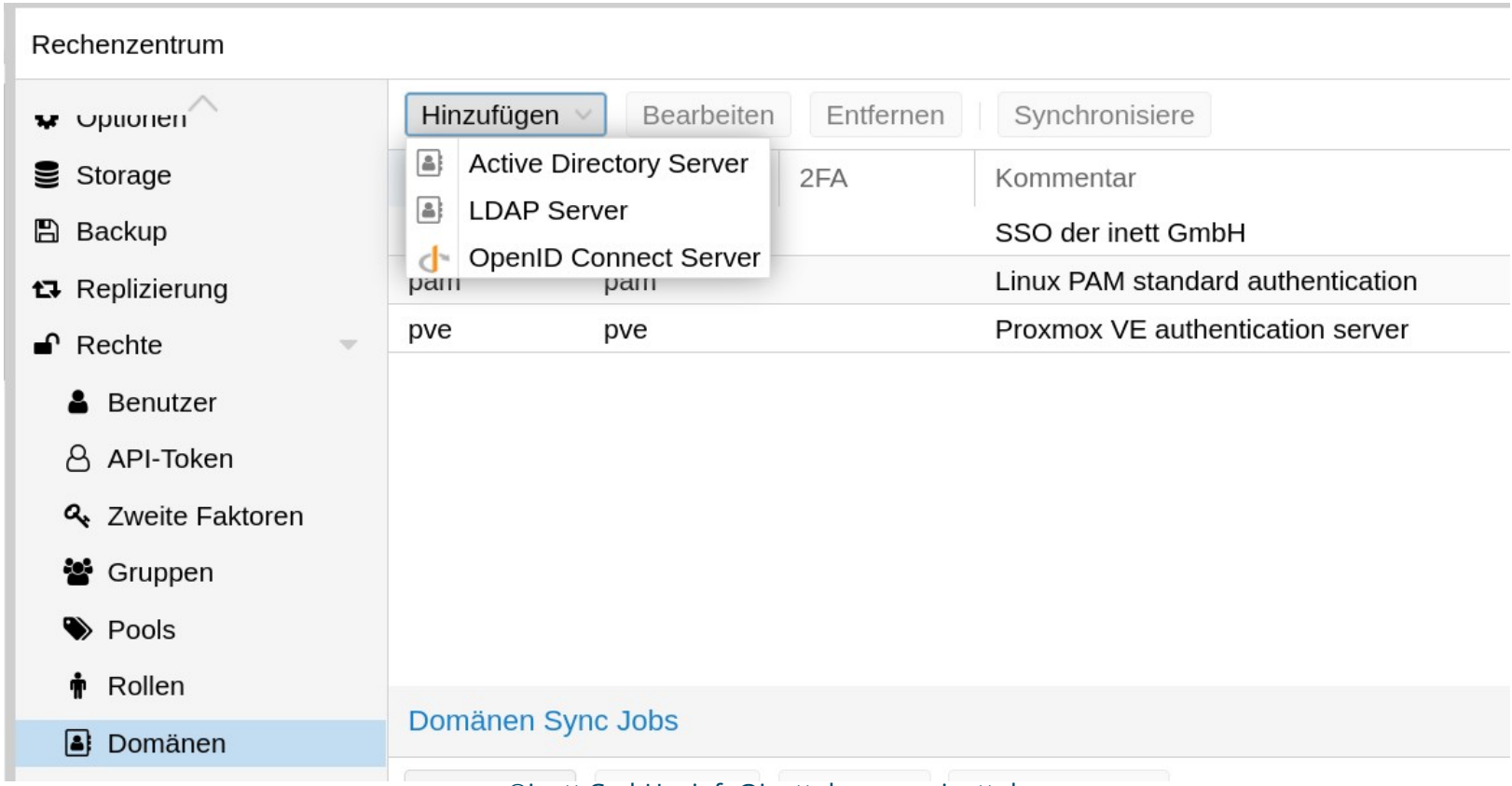
©inett GmbH -- info@inett.de -- www.inett.de

PVE Benutzerverwaltung

Authentication Realms

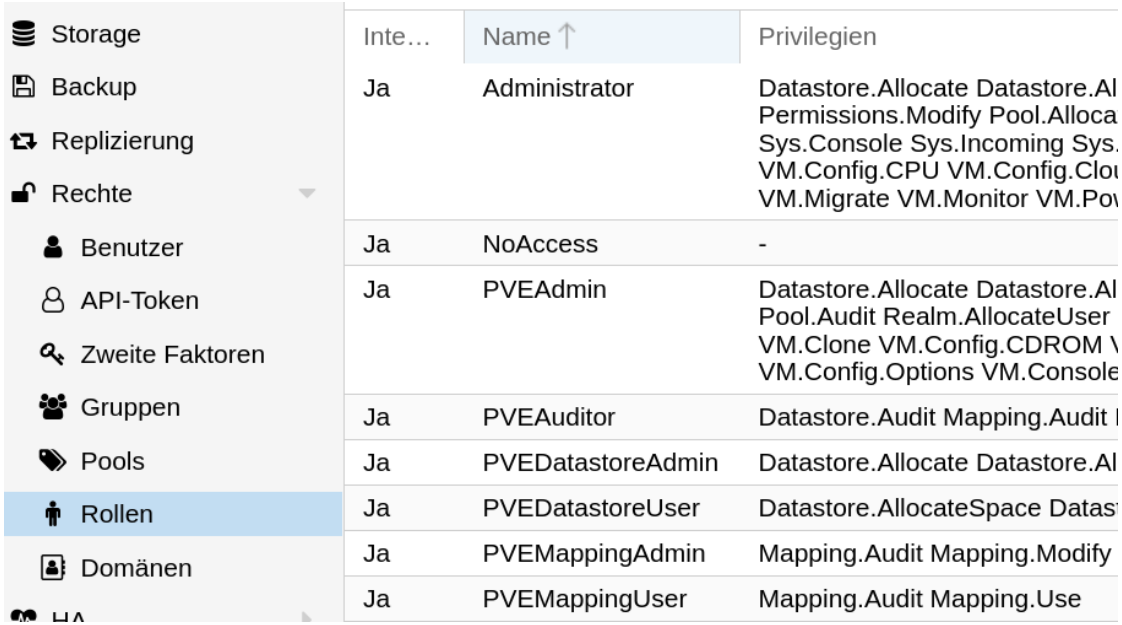
- “@pve”-Benutzerattribute werden in /etc/pve/users.cfg abgelegt
- Für jeden Benutzer muss ein Eintrag existieren
- Passwörter werden hier nicht gespeichert, diese werden direkt gegen die Backends geprüft
- Passwörter für „@pve“ → /etc/pve/priv/shadow.cfg

Benutzerverwaltung: Authentifizierungsrealms



Berechtigungsverwaltung

- Damit ein Benutzer eine Aktion ausführen kann (zum Beispiel das Auflisten, Ändern oder Löschen von Teilen der Konfiguration einer VM), benötigt der Benutzer die entsprechenden Berechtigungen.



Inte…	Name ↑	Privilegien
Ja	Administrator	Datastore.Allocate Datastore.Al Permissions.Modify Pool.Alloca Sys.Console Sys.Incoming Sys. VM.Config.CPU VM.Config.Clou VM.Migrate VM.Monitor VM.Pov
Ja	NoAccess	-
Ja	PVEAdmin	Datastore.Allocate Datastore.Al Pool.Audit Realm.AllocateUser VM.Clone VM.Config.CDROM V VM.Config.Options VM.Console
Ja	PVEAuditor	Datastore.Audit Mapping.Audit I
Ja	PVEDatastoreAdmin	Datastore.Allocate Datastore.Al
Ja	PVEDatastoreUser	Datastore.AllocateSpace Datas
Ja	PVEMappingAdmin	Mapping.Audit Mapping.Modify
Ja	PVEMappingUser	Mapping.Audit Mapping.Use

Berechtigungsverwaltung: Rollen

- *Eine Berechtigung ist das Recht, eine bestimmte Aktion auszuführen.*
- *Zur Vereinfachung des Managements werden Listen von Berechtigungen in Rollen gruppiert, die dann in der Berechtigungstabelle verwendet werden können.*
- *Beachten Sie, daß Berechtigungen nicht direkt Benutzern und Pfaden zugewiesen werden können, ohne Teil einer Rolle zu sein.*
- *Eine Rolle ist einfach eine Liste von Berechtigungen.*
- *Proxmox VE wird mit einer Reihe vordefinierter Rollen geliefert, die die meisten Anforderungen erfüllen.*

  - Administrator: Alle Privilegien
  - NoAccess: Keine Privilegien (Wird verwendet um Zugriff zu verbieten)
  - PVEAdmin: Fast alle Privilegien außer System Settings (Sys.PowerMgmt, Sys.Modify, Realm.Allocate)
  - PVEAuditor: Hat nur lesenden Zugriff → Monitoring

©inett GmbH -- info@inett.de -- www.inett.de

Benutzerverwaltung: Pools

- *Pools können verwendet werden, um eine Gruppe von virtuellen Maschinen und Datenspeichern zu gruppieren.*
- *Sie können dann einfach Berechtigungen auf Pools (/pool/{poolid}) setzen, die dann von allen Poolmitgliedern geerbt werden.*
- *Dies ist eine großartige Möglichkeit, den Zugriff zu vereinfachen.*

PVE Benutzerverwaltung

Objekte und Pfade

- Zugriffsberechtigungen sind an Objekte gebunden, z.B. :
  - Virtuelle Maschinen (/vms/{vmid})
  - Storage (/storage/{storeid})
  - ...

- PVE benutzt Pfade ähnlich einem Filesystem, um Objekte zu adressieren
- Pfade folgen die Baumstruktur der API,
  und Berechtigungen werden vererbt

©inett GmbH -- info@inett.de -- www.inett.de

PVE Benutzerverwaltung

Privilegien

- Privileg bezeichnet das Recht, einen bestimmten Vorgang ausführen zu dürfen
- Privilegien werden in Rollen *gruppiert*
- Rollen können Benutzern oder Gruppen zugewiesen werden
- Vereinfachte Administration mit Gruppen

- Technisch gesehen ist eine Berechtigung eine Zusammenfassung von <Pfad,Benutzer,Rolle>
- Auch bekannt als “Access Control List” = ACL
- Jede Berechtigung spezifiziert ein Subjekt (Benutzer/Gruppe), eine Rolle (Gruppe von Privilegien) und einen Pfad zu einem Objekt

inett

Vererbung von Berechtigungen

- Vererbungsregeln:
  - Berechtigungen für individuelle Benutzer überschreiben Gruppenberechtigungen
  - Berechtigungen einer Gruppe greifen, wenn ein Benutzer der Gruppe angehört
  - Berechtigungen einer tieferen Ebene überschreiben vererbte Berechtigungen

©inett GmbH -- info@inett.de -- www.inett.de

PVE Benutzerverwaltung

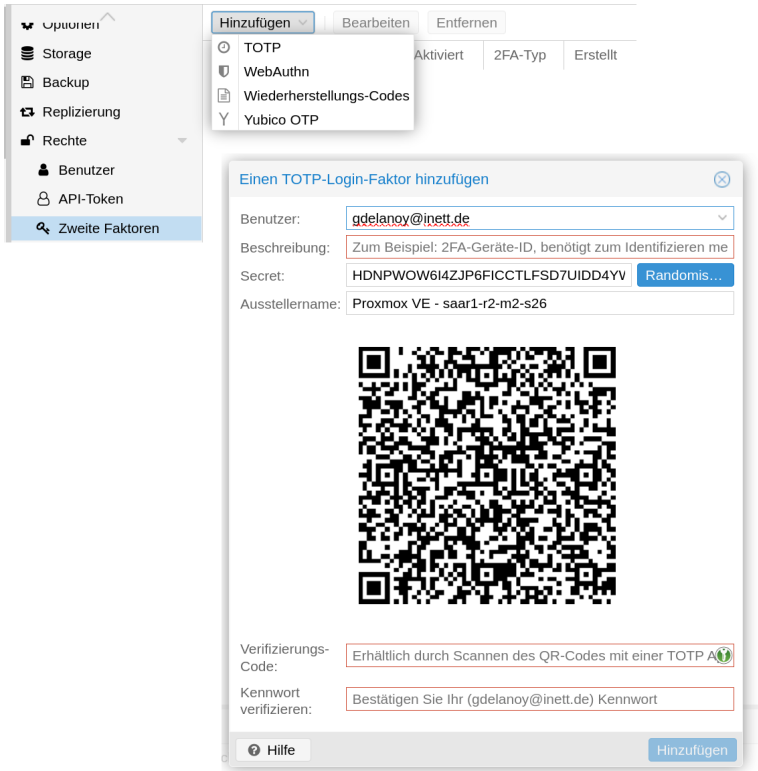
Beispiele für VM bezogene Privilegien:

VM.Allocate: Erstellen/löschen einer neuen VM zum Server Inventar
VM.Migrate: Migrieren einer VM auf eine andere Cluster Node
VM.PowerMgmt: Power Management (Start, Stopp, Reset, Shutdown, ...)
VM.Console: Konsolenzugriff auf eine VM
VM.Monitor: Zugriff zum VM Monitor (KVM/QEMU)
... (mehr im Proxmox VE Wiki)

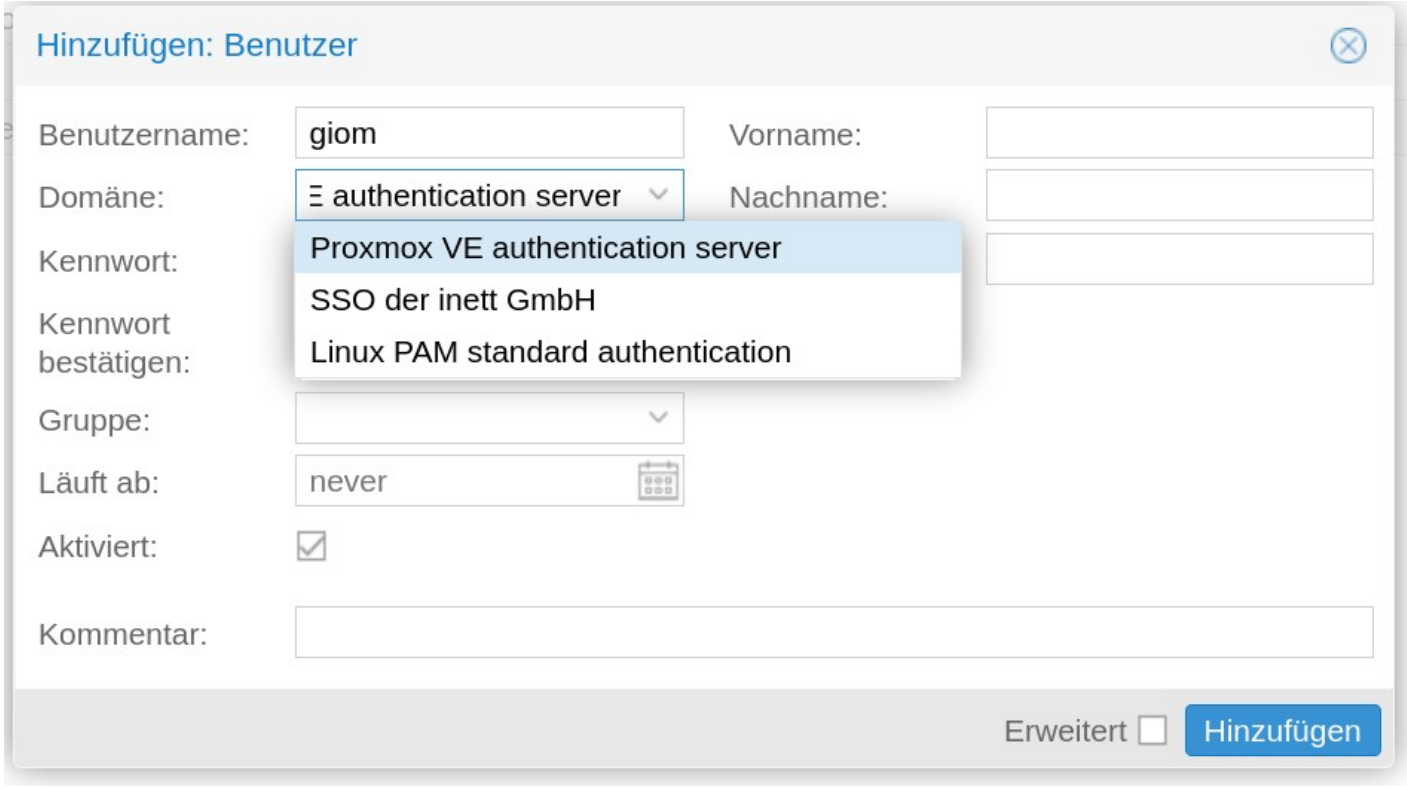
©inett GmbH -- info@inett.de -- www.inett.de

Benutzerverwaltung: 2FA

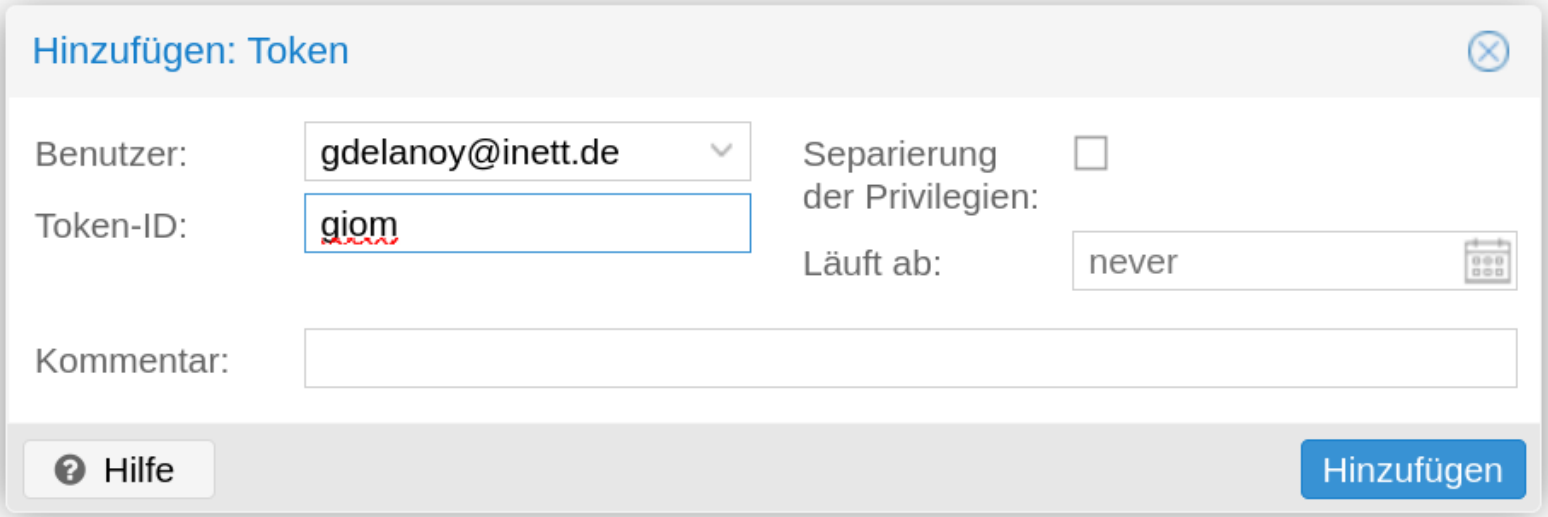
- Es gibt zwei Möglichkeiten, die Zwei-Faktor-Authentifizierung zu verwenden:
- Sie kann vom Authentifizierungsrealm erforderlich sein, entweder über TOTP (zeitbasiertes Einmalpasswort) oder YubiKey OT
- In diesem Fall muss einem neu erstellten Benutzer sofort ihre Schlüssel hinzugefügt werden, da es keine Möglichkeit gibt, sich ohne den zweiten Faktor anzumelden.

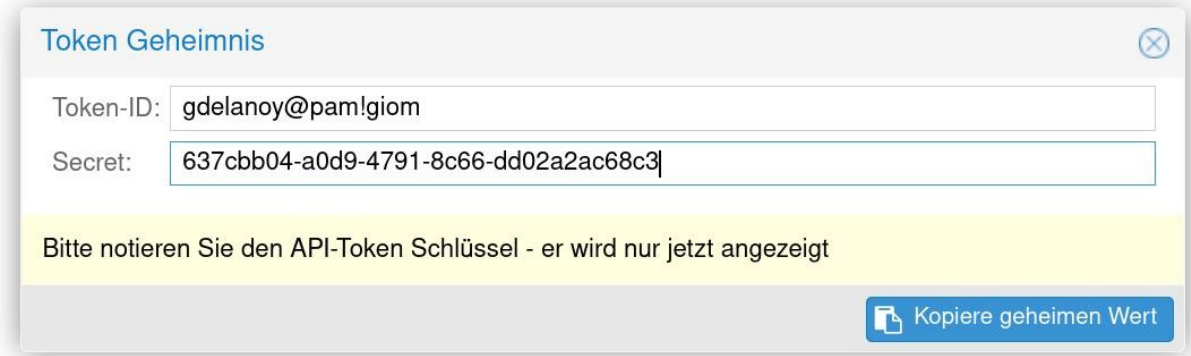


Benutzerverwaltung: API Tokens



Benutzerverwaltung: API Tokens





Benutzerverwaltung: API Tokens

- Es gibt zwei grundlegende Arten:

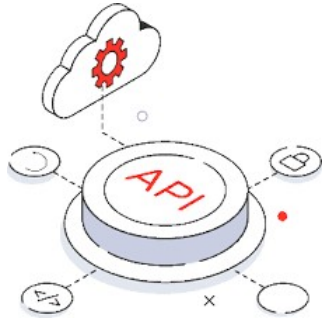
- *Getrennte* Berechtigungen:
  Das Token muss expliziten Zugriff mit ACLs erhalten. Seine effektiven Berechtigungen werden durch den Schnitt von Benutzer- und Tokenberechtigungen berechnet.

- *Vollständige* Berechtigungen:
  Die Berechtigungen des Tokens sind identisch mit denen des zugehörigen Benutzers.

PVE REST API

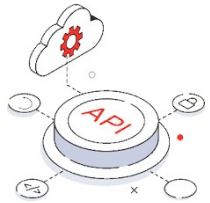


PVE Rest API: Wofür?



- Automatisierung
- Wiederkehrende Aufgaben
- Monitoring (auch performance)
- Entwicklung
- RZ-Steuerung

PVE Rest API

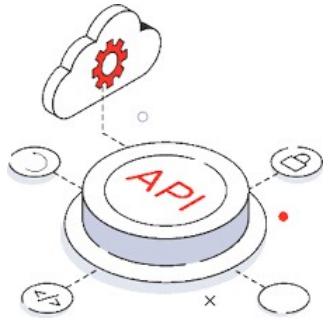


- In einem *Proxmox* Server ermöglicht die ***API*** (*Application Programming Interface*) die Steuerung des Servers über programmatische Methoden.
- Die ***API*** ermöglicht es Entwicklern und Systemadministratoren, Proxmox-Funktionalitäten zu *automatisieren* und in ihre eigenen Anwendungen zu integrieren.
- Die *Proxmox* API ist *RESTful* und basiert auf HTTP(*S*)-Anfragen.

  Sie wird z.B mit *curl* auf dem Port **8006** aufgerufen, auf dem auch die GUI mit einem Browser zu sehen ist.

©inett GmbH -- info@inett.de -- www.inett.de

PVE Rest API



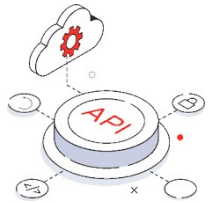
- JSON als Datenformat
  - Einfaches Datenformat (Optimal für Web)
  - Automatische Parametervalidierung
  - Automatische API Dokumentation
  - Einfacher Weg um Command Line Tools zu erstellen
- Zugriff über Browser, CLI, Software von Drittanbietern, etc.

- http://pve.proxmox.com/wiki/Proxmox_VE_API

©inett GmbH -- info@inett.de -- www.inett.de

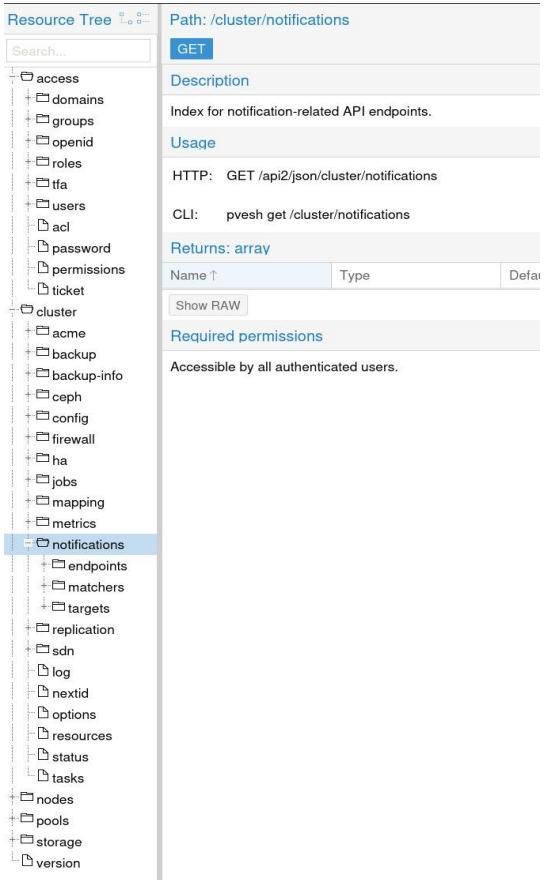
PVE Rest API

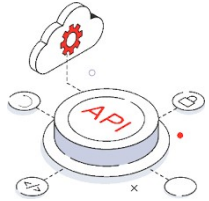
- Abrufen von Informationen mit HTTP **GET**
- Erstellen von Objekten mit HTTP **POST**
- Ändern mit HTTP **PUT**
- Löschen mit HTTP **DELETE**
- *Authentifizierungs-Token* wird benötigt
- und das Token benötigt ensprechende Rechte




```
$ curl -k -X GET "https://<Proxmox-Server:8006>/api2/json/nodes" -H 'Authorization:
PVEAPIToken=<Benutzername@Realm>!<Token_ID>=<Secret>' | jq
```

Proxmox VE API : Befehle

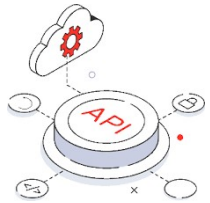




- API Dokumentation ist lokal auf Proxmox VE Servern verfügbar
- Auch hier online erhältlich.

©inett GmbH -- info@inett.de -- www.inett.de

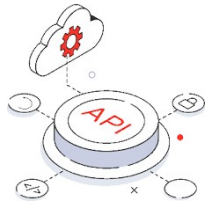
PVE API mit pvesh



- Die API kann auch lokal in der CLI abgerufen werden
- **Pvesh** (*Proxmox VE Shell*) ist ein Befehlszeilenwerkzeug, das die *Proxmox **API*** verwendet, um mit dem Proxmox-Server zu interagieren
- Es bietet eine einfache Möglichkeit, API-Anfragen direkt von der Befehlszeile des Systems
- Die Syntax der Anfragen zur API bleibt gleich

©inett GmbH -- info@inett.de -- www.inett.de

PVE API mit pvesh



- Die Liste aller virtuellen Maschinen abrufen:

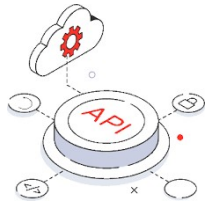
```
pvesh get /nodes/{node}/qemu
```

- Eine VM starten / stoppen:

```
pvesh create /nodes/{node}/qemu/{vmid}/status/start
pvesh create /nodes/{node}/qemu/{vmid}/status/stop | shutdown
```

©inett GmbH -- info@inett.de -- www.inett.de

PVE API mit pvesh



- Infos zu einer VM abrufen:

```
pvesh get /nodes/{node}/qemu/{vmid}/config
```

- Die Ressourcennutzung einer VM überprüfen:

```
pvesh get /nodes/{node}/qemu/{vmid}/status/current
```

-

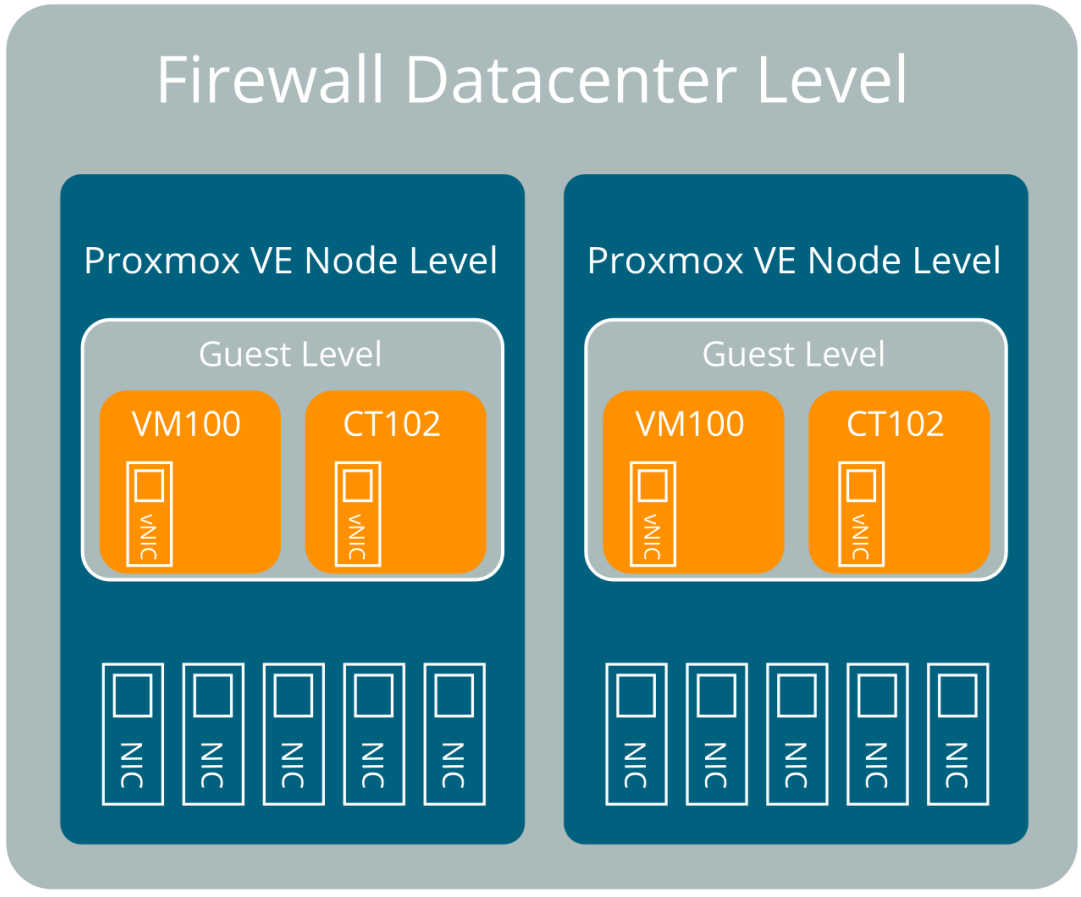
```
pvesh get /nodes/<node>/lxc
pvesh get /nodes/<node>/lxc/<id>/config
pvesh get /nodes/<node>/lxc/<id>/snapshot
pvesh create /nodes/<node>/lxc/<id>/snapshot -vmstate <snapshot_name>
```

©inett GmbH -- info@inett.de -- www.inett.de

Proxmox VE Firewall



Firewall Ebenen (Level)



©inett GmbH -- info@inett.de -- www.inett.de

Proxmox VE Firewall

- Einfach einzurichten (Frontent für Kernels *netfilter*)
- IPv4 und IPv6
- Clusterweite Konfiguration möglich
- 3 Ebenen der Konfiguration (Datacenter, Host, VM/CT)
  - Die Firewall muß *auf allen Ebenen* aktiviert werden
- Komplett anpassbar, erlaubt komplexe Szenarien, IDS/IPS einbindbar
- Default Policies (anpassbar)
  - Input → DROP
  - Output → ACCEPT
- 8 Log Levels
- High Performance, kein SPOF

Proxmox VE Firewall

- Regeln steuern das Verhalten
- Regeln werden vererbt (Nur Cluster → Node)
- Lokale Regeln übersteuern vererbte Regeln
- Alias: Ermöglicht Aliase für IP Adressen
- IPSet: Gruppen von IP-Adressen / Aliases
- Aliase und IPSets können als Source/Destination Parameter dienen
- Security Groups: Vordefinierbare Regelsätze
- local_network hat Ausnahmen gesetzt (8006/tcp, 22/tcp, 5405/udp, 3128/tcp, 5900-99/tcp)

Proxmox VE Firewall: Optionen

- Datacenter (Level)
  - Firewall aktivieren
  - Input Policy (default: DROP)
  - Output Policy (default: ACCEPT)
- Node:
  - Firewall bereits aktiviert, aber erst tätig wenn sie auf Datacenterebene hochgefahren ist
  - SMURFS Filter (Broadcastadresse als Quelle)
  - TCP Flags Filter
  - nf_conntrack_max (Default: 256k)
  - nf_conntrack_tcp_timeout_established
  - 8 Log Level Optionen (input, output, tcp flags und smurfs)
    → nolog, info, err, warning, crit, alert, emerg, debug

Proxmox VE Firewall: Optionen

- VM/CT
  - Firewall aktivieren
  - DHCP aktivieren
  - MAC Filter
  - 8 Log Level Optionen (input, output, tcp flags und smurfs)
    → nolog, info, err, warning, crit, alert, emerg, debug
  - Input Policy (default: DROP)
  - Output Policy (default: ACCEPT)

Proxmox VE Firewall aktivieren

- **Achtung**: Default Input Policy = DROP bei allen eingehenden Paketen → Erst konfigurieren !
- Vor Aktivierung: Ausnahmen auf local_network prüfen und ggf. anpassen („pve-firewall localnet“)
- **Wichtig**: Firewall muss *auf allen Ebenen* aktiviert werden
  1. Firewall auf Host Ebene ist bereits aktiviert
  2. Firewall auf Datacenter Ebene muss noch aktiviert werden
     → Nun ist die Firewall für alle Nodes aktiv
  3. Um die Firewall für VMs und Container zu aktivieren, muss sie über die VM/CT Network Device Level aktiviert und Regeln definiert werden
     → Nun ist die Firewall auf VM/CT Ebene aktiv

©inett GmbH -- info@inett.de -- www.inett.de

Proxmox VE Firewall Regeln

- Regeln können über GUI oder CLI hinzugefügt werden
- Nicht angegebene Parameter werden wie *Wildcards* behandelt
- Arbeitet mit IPv4 und IPv6
- Verschiedene Protokolle werden unterstützt
- Vordefinierte Makros für schnelles Setup
- Jede einzelne Regel kann separat aktiviert/deaktiviert werden
- Aliase und IPSets können als Source/Destination verwendet werden

©inett GmbH -- info@inett.de -- www.inett.de

inett

Custom Certificate Management



Custom Certificate Management

- Es gibt zwei Anwendungskategorien, in einem PVE Server Zertifikate zu verwenden:
  - Die erste betrifft der Zugang mit Spice,
  - und die zweite sichert die Kommunikation innerhalb eines Clusters, für Funktionalitäten wie Cluster Management oder VM Migration.

- Meistens werden zu diesen Zwecken selbstsignierte Zertifikate werwendet.

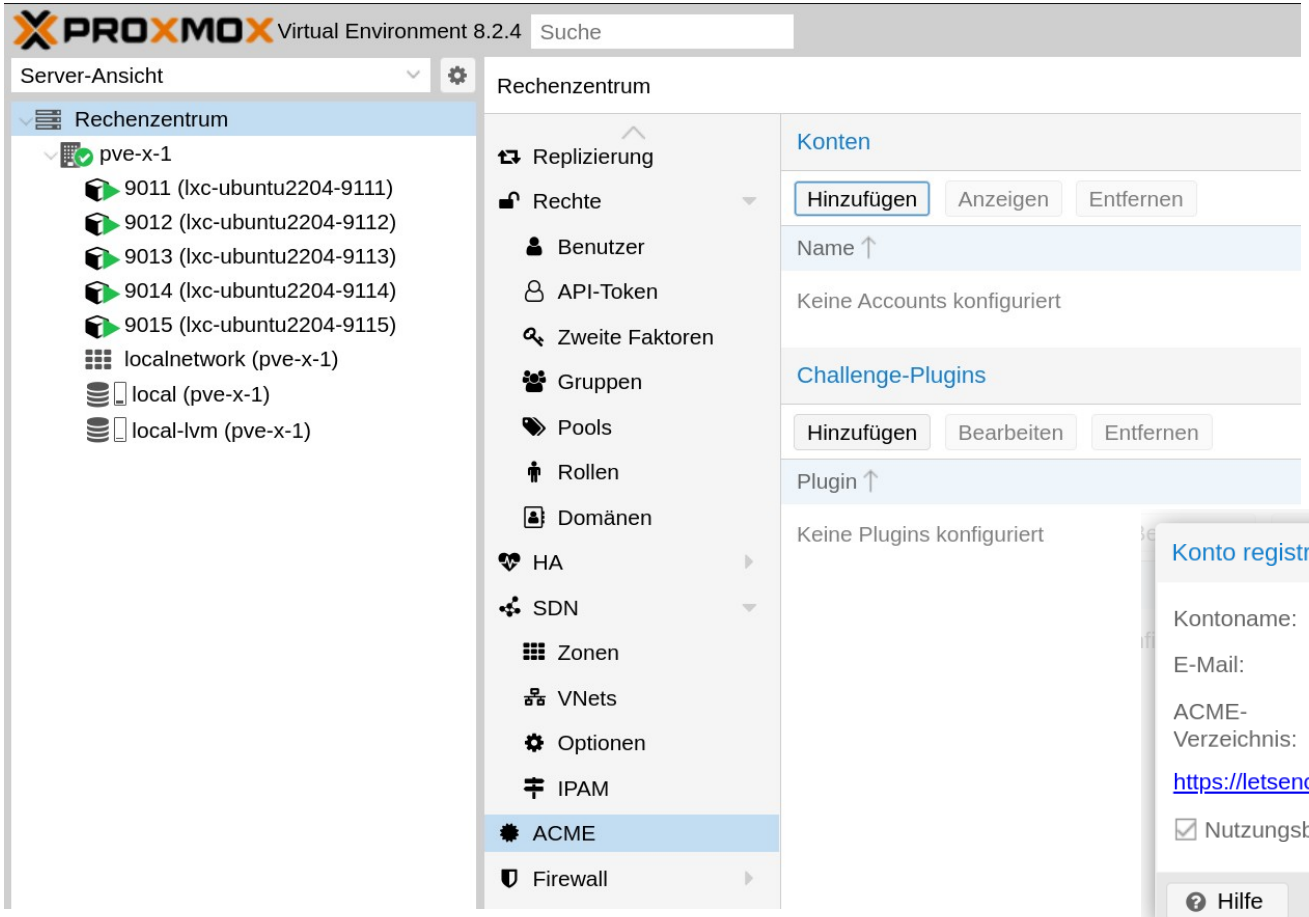
©inett GmbH -- info@inett.de -- www.inett.de

TLS-Verschlüsselung mit SPICE

- Dafür muß libvirtd-Konfiguration angepasst werden
- Das Zertifikat wird in /etc/libvirt/qemu.conf festgelegt.
- Dazu:

```
# Folgende Linien werden auskommentiert :
spice_listen=”0.0.0.0”
spice_tls=1
spice_tls_x509_cert_dir=”/etc/pki/libvirt-spice”
```

Custom Certificate Management



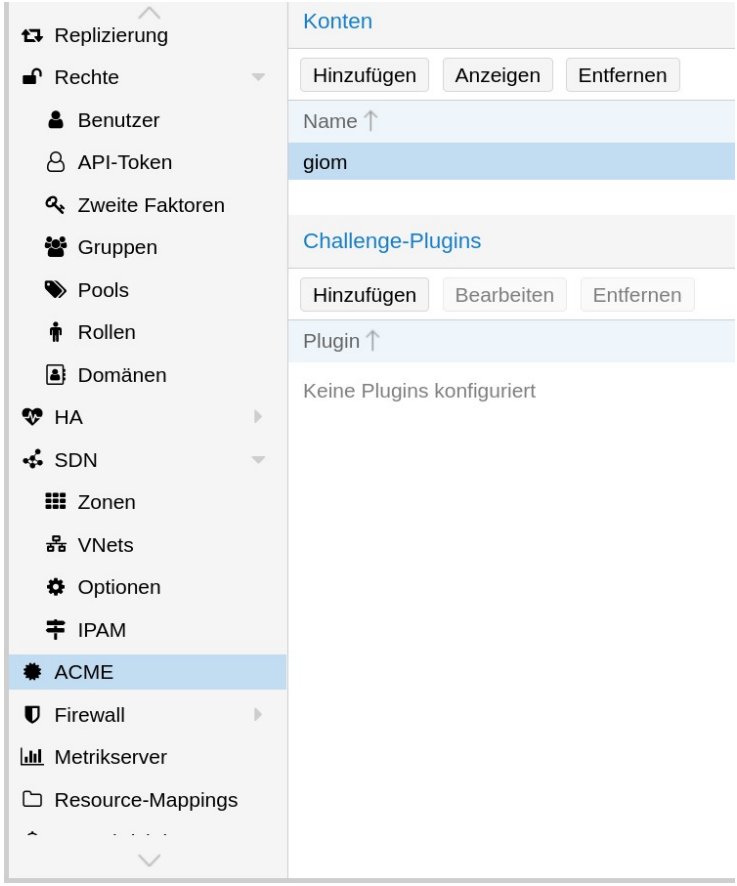


Rechenzentrum:

→ ACME

→ Konten

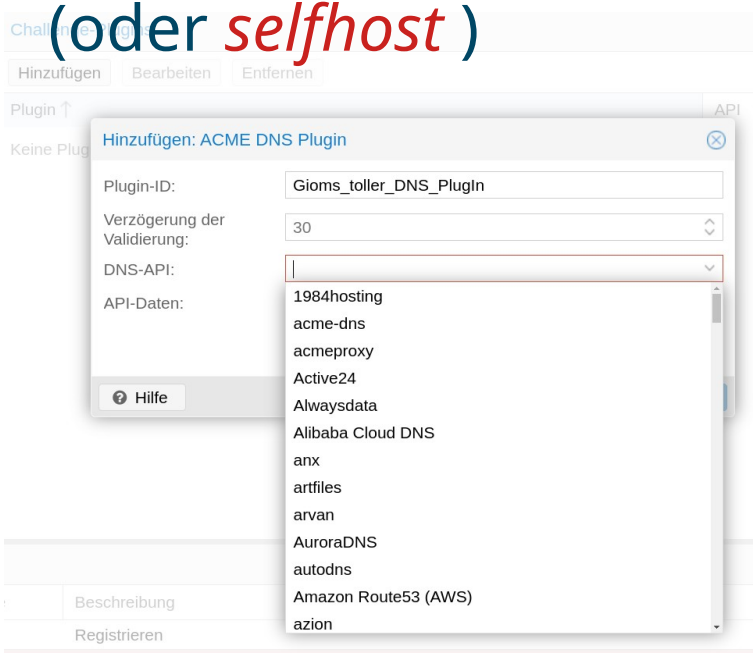
Custom Certificate Management



Challenge Plug-Ins (DNS) :

→ Hinzufügen

→ Die API Ihres Providers (oder *selfhost*)

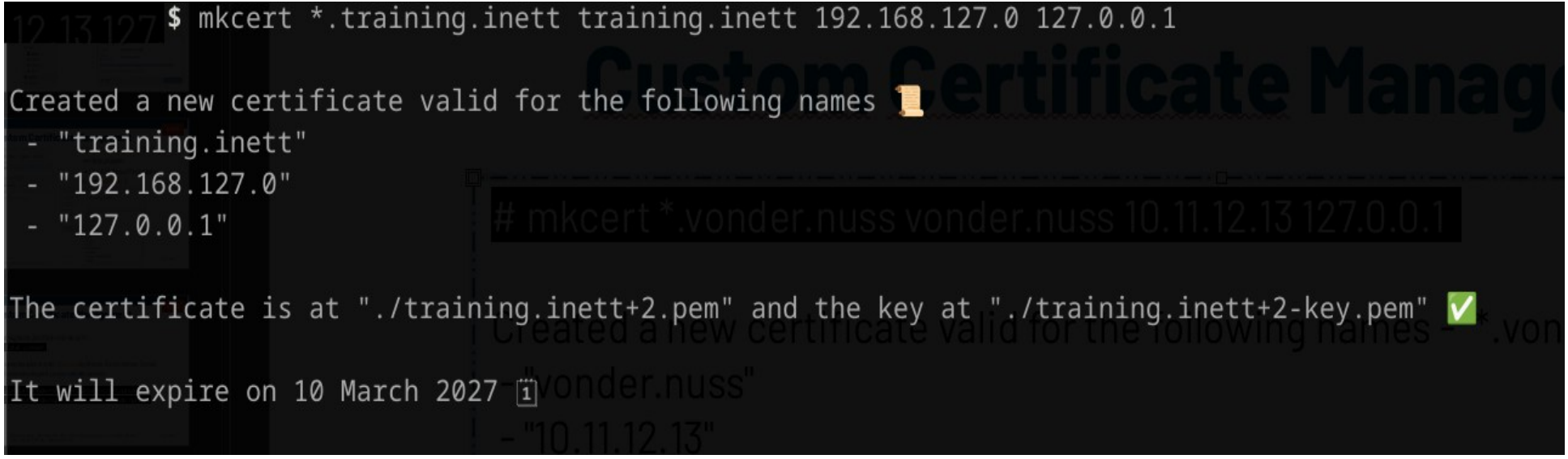


©inett GmbH -- info@inett.de -- www.inett.de

Custom Certificate Management

- Selbst-signierte Zertifikate mit mkcerts :
  ```
  # apt install -y mkcert
  ```
- Folgendes Beispiel erstellt *Wildcard* Zertifikate für ein *fiktives* Domän
- Sie können dann kopiert und verwendet werden

Custom Certificate Management



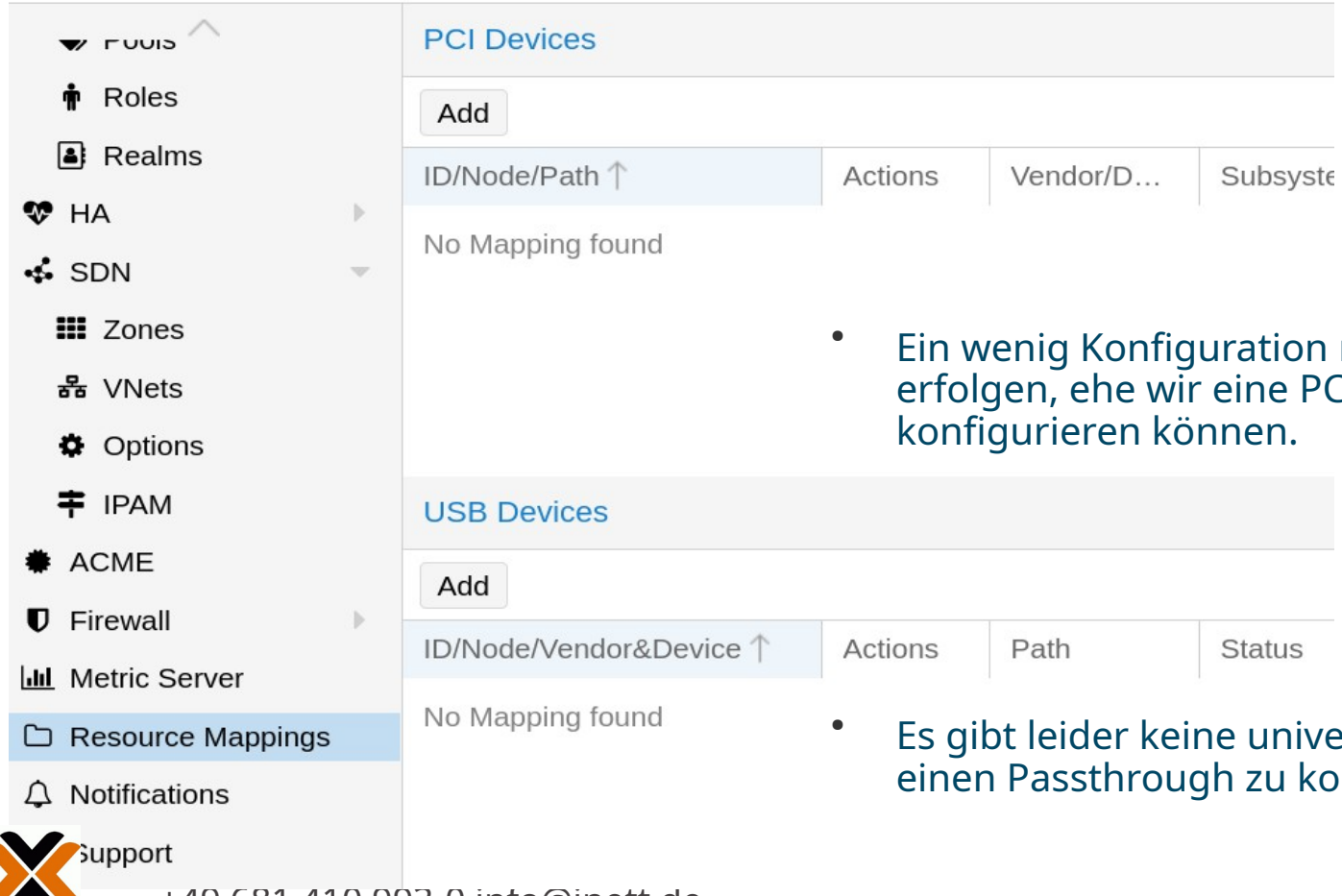
inett

PCI(e) Passthrough



PCI(e) Passthrough

Datacenter



- Ein wenig Konfiguration muß im System erfolgen, ehe wir eine PCI Karte in der GUI konfigurieren können.
- Es gibt leider keine universelle Methode, um einen Passthrough zu konfigurieren.

+49 681 410 993-0 info@inett.de

PCI(e) Passthrough: PCI

- IOMMU im BIOS / UEFI:

  - Auf Intel Hardware heisst es "VT-d" (Virtualization Technology for Directed I/O).

  - Auf AMD Hardware werden Sie eher „AMD-Vi“ sehen.

- Stellen Sie sicher, ob sie using GRUB oder systemd verwenden:

- `# efibootmgr -v` ← Sollten Sie eine solche Antwort erhalten:

  "File(\EFI\SYSTEMD\SYSTEMD-BOOTX64.EFI)", dann verwenden Sie Systemd

  und nicht GRUB.

- Verwenden Sie GRUB, werden Sie seine Konfigurationsdateien anpassen möchten



PCI(e) Passthrough: with GRUB

```
# nano /etc/default/ grub
```

- Für Intel-CPUs fügen Sie hinzu:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"
```

- Für AMD-CPUs fügen Sie hinzu:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet amd_iommu=on"
```

- Stellen Sie dann sicher, dass Grub seine neue Konfiguration abliest:

```
# update-grub
```



PCI(e) Passthrough: mit SYSTEMD

inett

- Sollten Sie Systemd verwenden, wird er dann so angepasst:

```
# nano /etc/kernel/cmdline
```

- Für Intel-CPUs fügen Sie hinzu:

```
quiet intel_iommu=on
```

- Für AMD-CPUs fügen Sie hinzu:

```
quiet amd_iommu=on
```

- Und der Boot-Tool muß aufgefrischt werden:

```
# proxmox-boot-tool refresh
```



PCI(e) Passthrough for both Grub and Systemd : add a kernel module

```
# nano /etc/modules
```

- Fügen Sie diese Zeile hinzu:

```
vfio vfio_iommu_type1 vfio_pci vfio_virqfd
```

- Und starten Sie den Server neu.
- Prüfen Sie dann mit:

```
dmesg | grep -e DMAR -e IOMMU
```



PCI(e) Passthrough: PCI

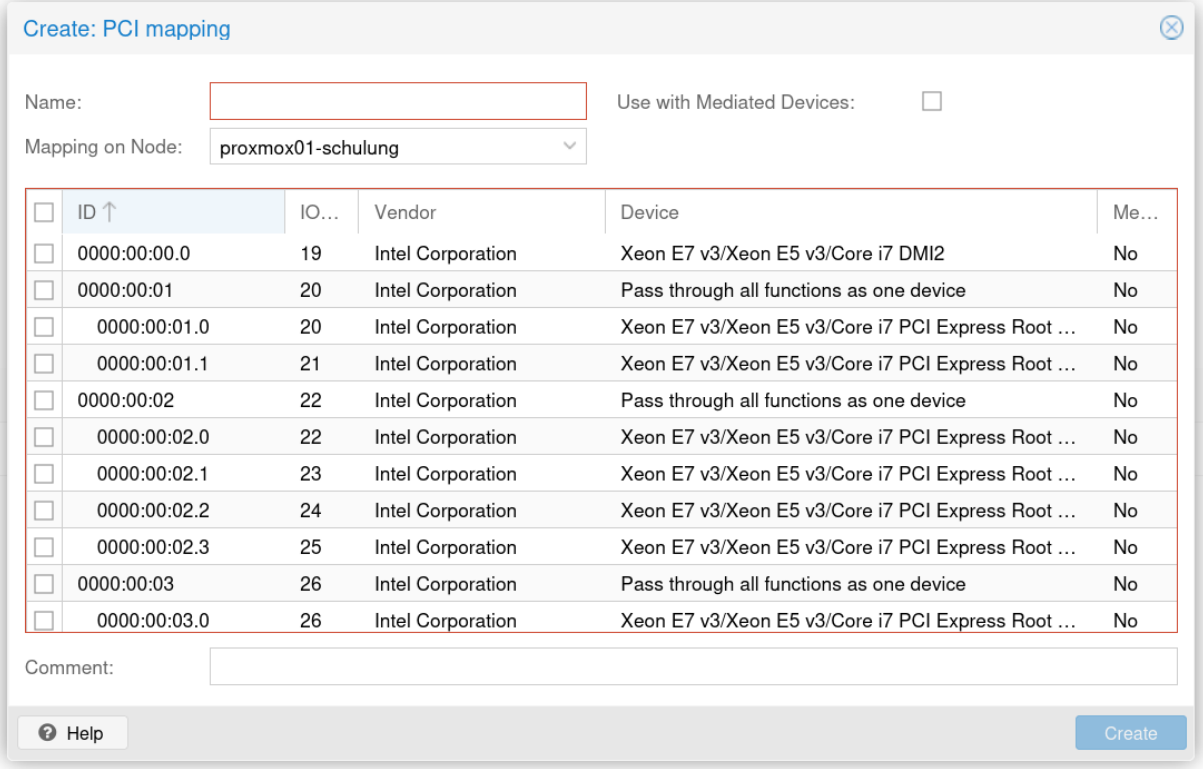
Proxmox wird den PCIe device ID sowohl als den Namen des Herstellers anzeigen.

Dies vereinfacht die Erkennung der Netzwerkkarten im System.

Die physische Anordnung der eingebetteten NICs wird so angezeigt, daß MAC-Addressen und PCIe Ids *sequenziell* angezeigt werden.

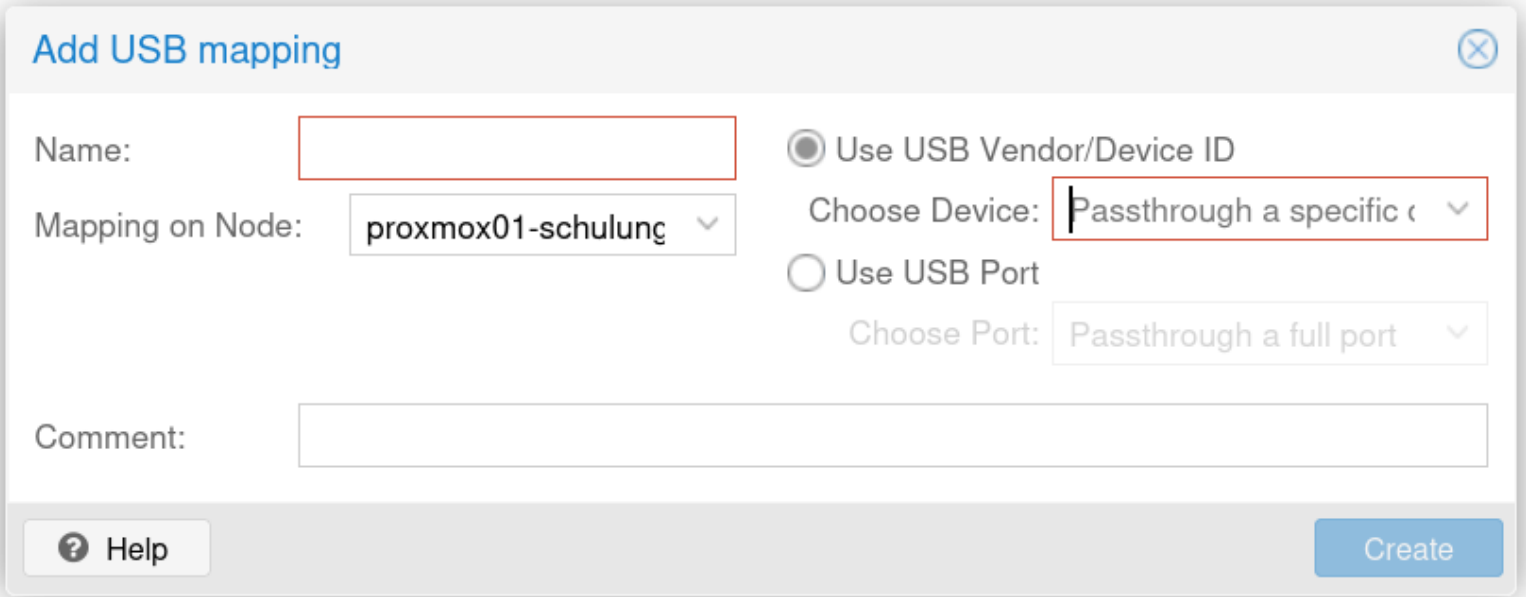
In diesem Beispiel ist 0000:01:00.0 die erste Netzwerkkarte(eth0).

0000:02:00.0 ist dann die zweite, und so weiter.



PCI(e) Passthrough: USB

- USB Hardware durchreichen ist etwas einfacher
- `# lsusb -v` wird helfen



Optionale Themen zu PVE Advanced



Optionale Themen:
PVE Performance Benchmarking



inett

Storage Benchmarks

- In vielen Virtualisierungsumgebungen ist Storage das Bottleneck
- Einfache Tools für Benchmarks
  - pveperf /var/lib/vz
  - fio, dbench, bonnie++, ...
  - LVM: dd and pv
    - dd if=/dev/mapper/pve-swap bs=256k | pv -bretp | dd of=/dev/null bs=256k
- Phoronix Test Suite (Linux)
- Innerhalb Windows VM: CrystalDiskMark

inett

Andere Benchmarks

- Netzwerk
  - Best Practice: iperf3
    (verfügbar für viele Plattformen, inklusive Windows)
- Phoronix Test Suite:
  - Läuft auf Proxmox VE Host und in KVM VM und LXC Container
  - Beispiel:
    - phoronix-test-suite benchmark build-linux-kernel

©inett GmbH -- info@inett.de -- www.inett.de

Optionale Themen: Troubleshooting



PVE Troubleshooting

- Tools
  - journalctl
  - dmesg
  - corosync-quorumtool
  - omping

- Orte
  - /var/log/pve-firewall.log
  - /var/log/lxc/*
  - /var/log/pveproxy/access.log
  - /var/log/vzdump/*
  - /var/log/syslog

Software-Defined-Networking (SDN) unter Proxmox VE



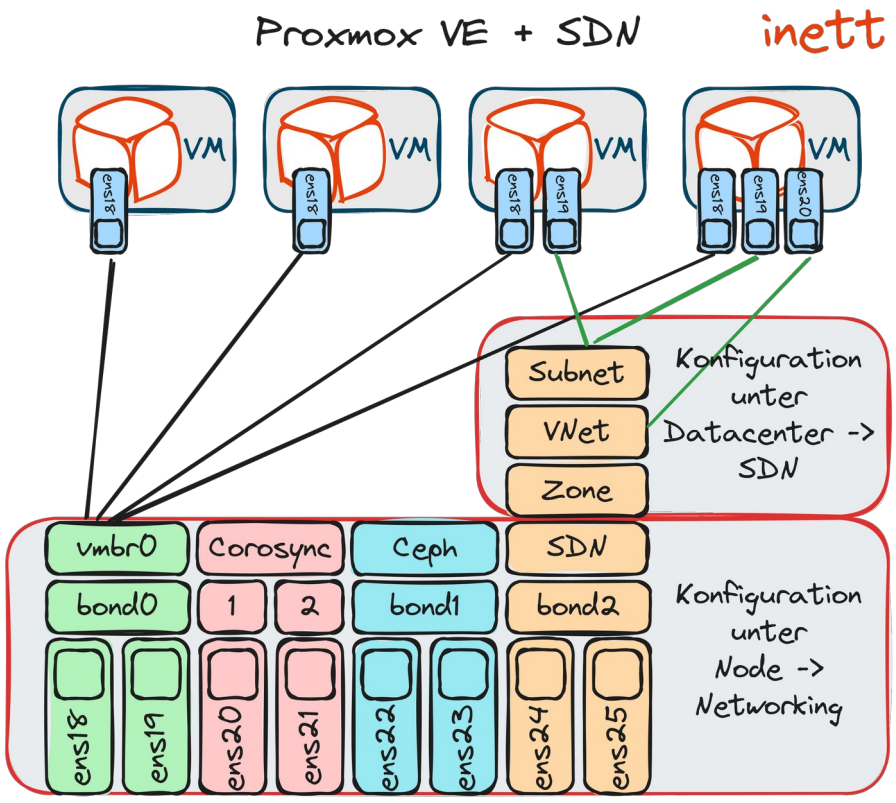
SDN Features

- Zentrale Konfiguration/Verwaltung/Steuerung
- Konfiguration via API (bei PVE auch ohne SDN schon möglich)
- Automatisierung und Flexibilität
- Virtualisierung von Netzwerken
- Im Virtualisierungsumfeld sinnvoll

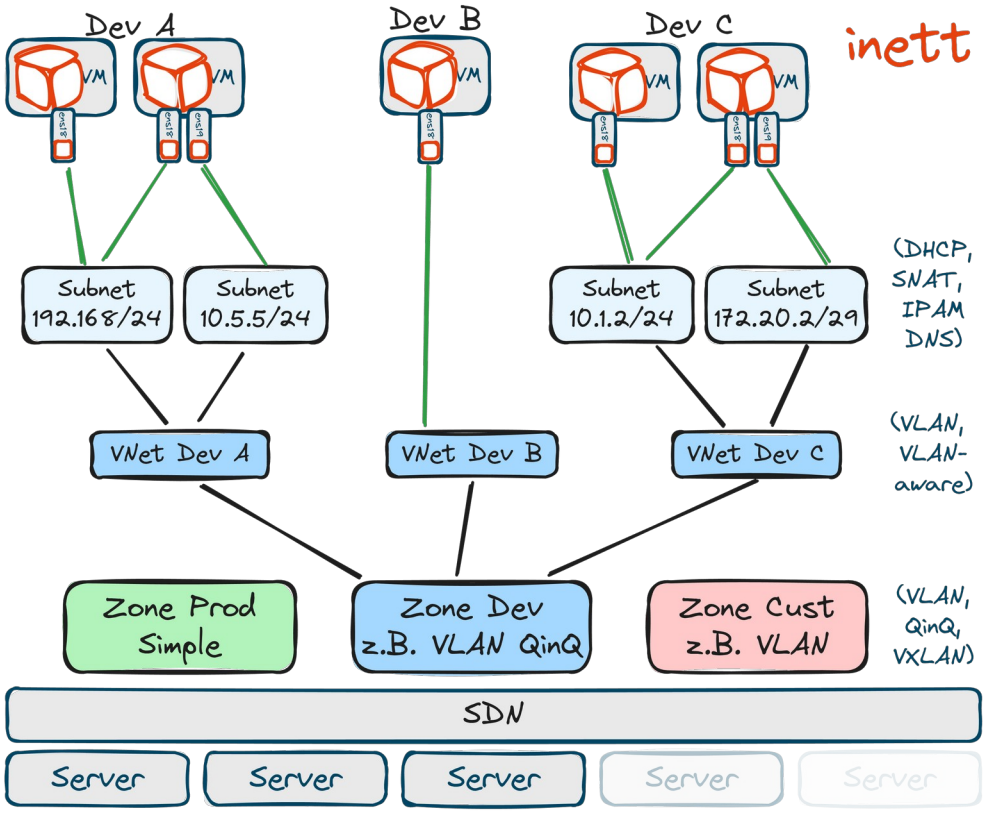
SDN Komponenten

- Zone
  - Zone bestimmt Netzwerkseitige Konfiguration (Simple, VLAN, VXLAN, Routed...)
  - Separate Netzwerkzone
  - Beispiele: Pro Zone ein Kunde, eine DMZ, Produktionsnetz, Testnetz, DevNetz
- VNet
  - Virtuelle Netzwerkbridge (vmbr)
  - Anbindung an VMs/Container
- Subnet
  - IP-Subnetzkonfiguration inklusive Services
  - DHCP, DNS, IPAM möglich

Node vs SDN Konfiguration



Beispiel für SDN im Einsatz

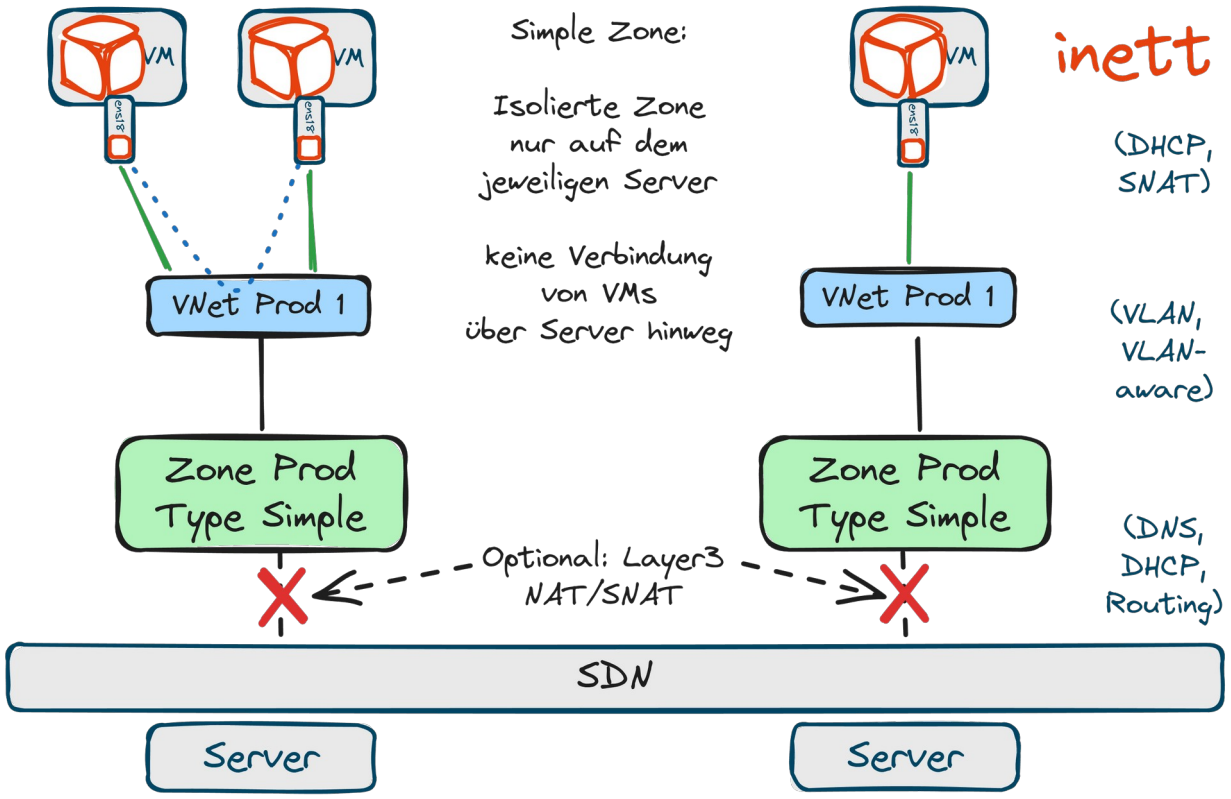


©inett GmbH -- info@inett.de -- www.inett.de

SDN Simple Zone

- Isolierte Bridge
- Komplette Isolation, SNAT oder Layer3 Routing optional
- Gäste müssen auf dem selben Server sein
- Keine Kommunikation über Server hinweg
- Alle VMs auf dem gleichen Server können kommunizieren

SDN Simple Zone

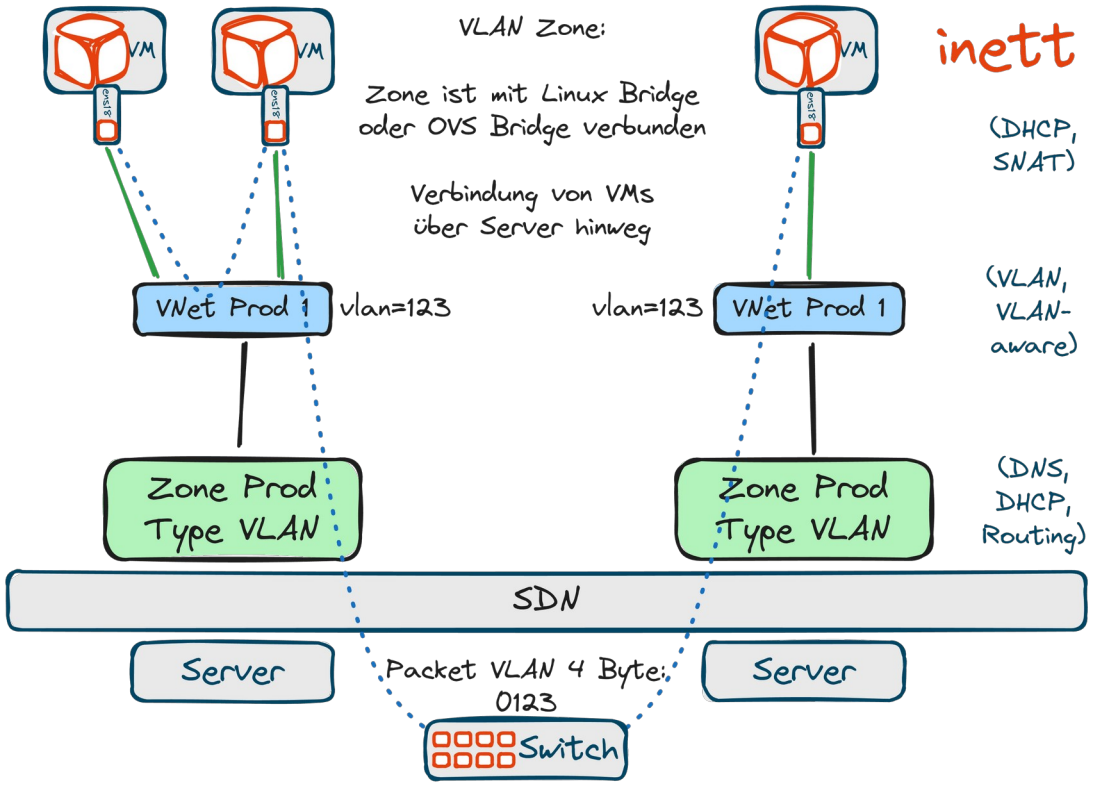


SDN VLAN Zone

- Gängige Methode um LANs zu segmentieren
- Herstellerunabhängig
- im Ethernet-Header bereits vorgesehen
- Bis zu 4094 unterschiedliche VLANs möglich
- Externe Switche müssen konfiguriert werden
- VMs im gleichen VLAN sehen sich serverübergreifend

©inett GmbH -- info@inett.de -- www.inett.de

SDN VLAN Zone

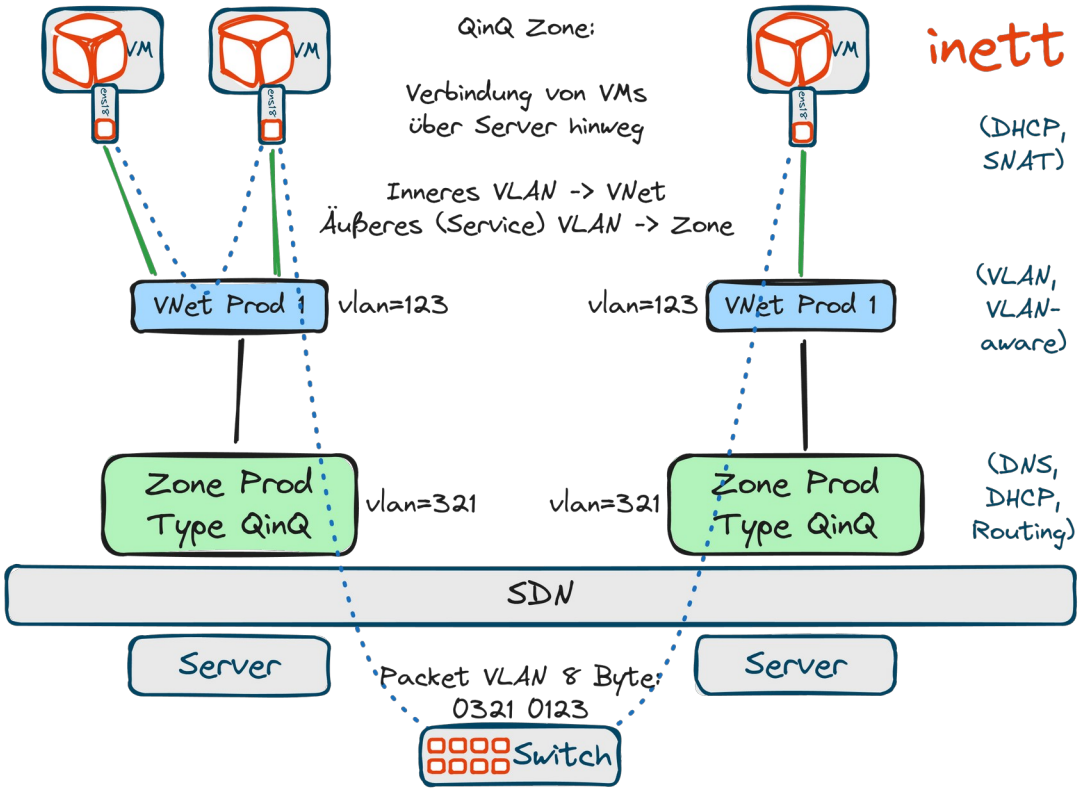


SDN QinQ Zone

- Stacked VLAN oder auch „VLAN in VLAN“
- Herstellerunabhängig 802.1ad (alt) oder 802.1q (neu)
- 16.760.836 (4094*4094) getrennte Segmente möglich
- Switch Support benötigt
- 4 Byte zusätzlich im Ethernet Frame benötigt (MTU anpassen!)

©inett GmbH -- info@inett.de -- www.inett.de

SDN QinQ Zone

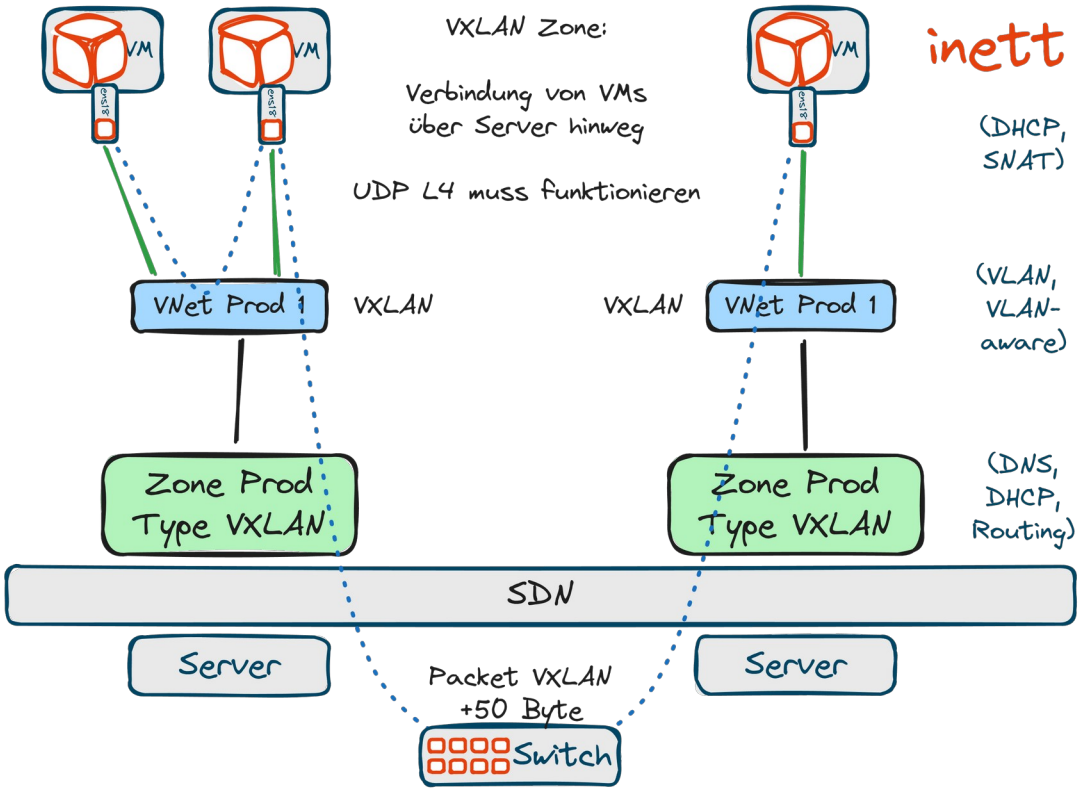


SDN VXLAN Zone

- VXLAN bildet einen „Overlay“-Tunnel auf einem „Underlay“-Netzwerk
- Layer2 Pakete werden in Layer4 UDP Paketen transportiert
- Underlay muß selbst konfiguriert werden (physisch, UDP, Target Port 4789)
- Über öffentliches Internet möglich, VMs sehen L2-Verbindung untereinander
- Kein Switch Support benötigt (Standard Layer 4 UDP Pakete)
- VXLAN Tunnel selbst ist NICHT verschlüsselt
- VXLAN Kapselung benötigt 50 Byte extra (MTU anpassen!)

©inett GmbH -- info@inett.de -- www.inett.de

SDN VXLAN Zone



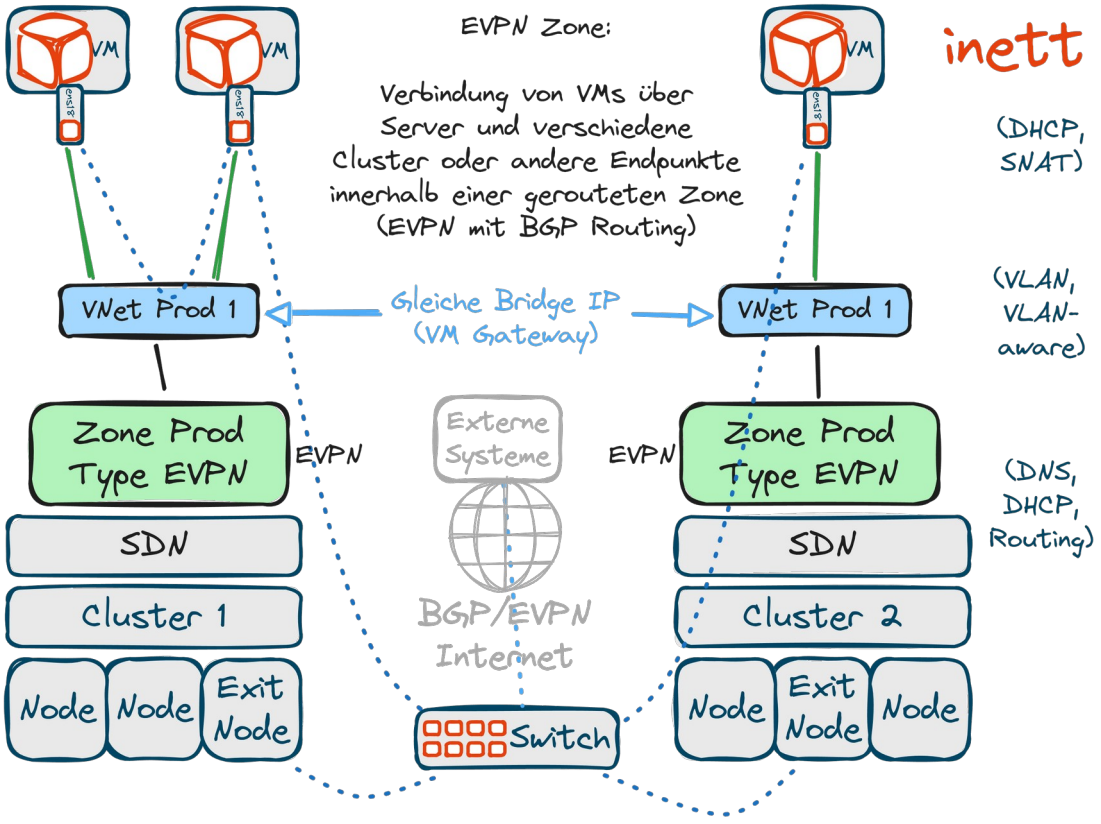
©inett GmbH -- info@inett.de -- www.inett.de

SDN EVPN(-BGP) Zone

- EVPN bildet ein routingfähiges Layer3 VPN
- Verwendet BGP als Routingprotokoll
- Kann über mehrere Zonen oder gar mehrere Cluster aufgespannt werden
- Kann externe Systeme anbinden (OPNsense Firewall z.B.)
- Verwendet Exit Nodes für Übergang VPN → „Extern“
- Benötigt zwingend Controller Instanz (Aktuell: evpn, bgp, isis)

©inett GmbH -- info@inett.de -- www.inett.de

SDN EVPN(-BGP) Zone



inett

Zusammenfassung der Zonentypen

Zone Type	Verbindung über mehrere Nodes	Verbindung über mehrere Cluster	Switch Support	Supported (Stand PVE 8.2)
Simple	❌	❌	N/A	✅
VLAN	✅	(✅)	VLAN	✅
QinQ	✅	(✅)	802.1ad / 802.1q	✅
VXLAN	✅	✅	✅nicht benötigt	❌
EVPN	✅	✅	✅nicht benötigt	❌

Status und Support

- Seit 2019 / 6.2 experimentell verfügbar
- Ab PVE 8.1 standardmäßig installiert
- Core-SDN inkl. VNet Management ab Proxmox VE 8.1 supported
- IPAM und DHCP/DNS in Tech Preview
- Routing via FRRouting und Controller Integration in Tech Preview

Empfehlungen

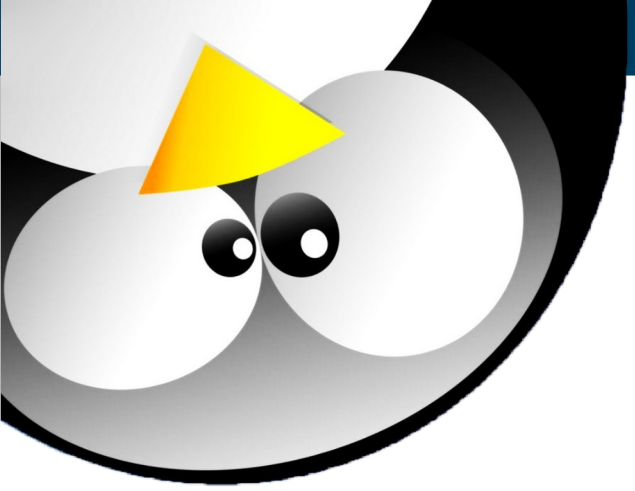
- In Produktion nur unterstützte Komponenten und Features verwenden.
- SDN verwenden, wenn es hilfreich ist.
- SDN erhöht die Komplexität - bei „normalem“ Networking bleiben, wenn kein SDN benötigt wird.
- Vereinfacht Administration / Delegation vor allem bei größerer Anzahl Nodes oder größeren Umgebungen.
- Proxmox VE Release Notes regelmäßig prüfen, da die Entwicklung noch im Gange ist.

inett

Fragen?

und Antworten...





inett

Vielen Dank !

Und bitte das Feedback nicht vergessen...