> Anonymisierte Portfolio-/Case-Study-Version. Produkt-/Kundenname, Domains, Hostnamen und öffentliche Adressen entfernt; Architektur-Patterns unverändert. Arbeitsprobe IT-Systemintegration.

Case-Study: Compute-Layer auf Proxmox VE (Multi-Tenant-SaaS-Plattform)

> Compute-Layer-Dokumentation für eine Multi-Tenant-SaaS-Plattform (Produkt anonymisiert). Single-Server-Virtualisierung auf einem dedizierten Bare-Metal-Server, sauber strukturiert nach VMID-Konvention, mit isoliertem Backup-Layer und Disaster-Recovery-Runbook.

1. Executive Summary

Single-Server-Setup auf einem dedizierten Bare-Metal-Server (Hetzner) mit Proxmox VE 9.1, nativ als Hypervisor installiert. **35+ Gäste in Betrieb** — eine bewusste Mischung aus VMs (qemu/KVM) und LXC-Containern, strukturiert nach einer klar definierten VMID-Konvention plus dediziertem, netzwerk-isoliertem Backup-Layer.

VMs werden dort eingesetzt, wo echte Kernel-Isolation und Hardware-Features gefragt sind (Firewalls, Datenbank, Workload-Hosts); LXC-Container für leichtgewichtige Dienste (Backup-Server, CI-Runner, Cert-/Docs-/Notify-Dienste), die keinen eigenen Kernel brauchen und ressourcenschonend laufen.

Die Plattform befindet sich im Eigenbetrieb (Pre-Launch): Die Infrastruktur läuft im Vollbetrieb, das darauf aufsetzende SaaS-Angebot ist noch nicht im Kundenbetrieb.

Quick Facts

Kennzahl	Wert
Dedicated Server	Bare-Metal bei Hetzner, deutsches Rechenzentrum (Hostname anonymisiert)
Hypervisor	Proxmox VE 9.1 (Kernel 7.0.0-3-pve), nativ
Aktive Gäste	35+ in Betrieb (29 VMs + 8 LXC)
Public IPv4-Adressen	4 gesamt: 3 am Host (je eigene Hetzner-MAC) + 1 am separaten Cloud-VPS-Shield
Storage	ZFS (lokaler Pool für VMs + Container)
Host-Firewall	firewalld (Default-Zone `public` = Default-Deny); pve-firewall NICHT in Benutzung
Backup	Proxmox Backup Server (LXC, isolierte Bridge), Off-Site, verschlüsselt, Restore-getestet
Out-of-Band-Zugriff	Robot-KVM des Anbieters + Out-of-Band-WireGuard zum Host
Automatisierung	Ansible (Infrastructure as Code)

2. Hardware-Plattform

Dedicated Server

Feld	Wert
Server-Typ	Dedizierter Bare-Metal-Server bei Hetzner (Hostname anonymisiert)
Standort	Deutsches Rechenzentrum (Standort anonymisiert)
Hypervisor-Software	Proxmox VE 9.1 (Kernel 7.0.0-3-pve), nativ als Hypervisor
CPU-Features	VT-x + Nested Virtualization aktiviert
Physisches NIC	Eine physische NIC (Bezeichnung anonymisiert)

Netz-Anbindung des Hosts

Der Host hat **eine physische NIC**. Darauf liegen zwei logische Pfade:

- eine **Haupt-Bridge für Public** (trägt die drei Host-seitigen Public-IPs über MAC-gebundene vNICs)
- ein **Hetzner vSwitch** (VLAN 4000, MTU 1400) als privates Layer-2 zur Cloud, das die Standorte/Dienste über ein providerinternes, getrenntes Segment verbindet.

Jeder Firewall-Transit erhält zusätzlich eine **eigene dedizierte Proxmox-Bridge** — es gibt bewusst **kein gemeinsames L2** zwischen den Firewall-VMs.

Public-IP-Zuordnung mit MAC-Filter

Hetzner bindet jede zusätzliche Public-IP an eine **separate MAC-Adresse** ("zusätzliche IP mit eigener MAC", Anti-Spoofing auf Upstream-Router-Ebene). Beim Erstellen der VM wird die MAC am vNIC fest eingetragen. **Drei** Public-IPs liegen so am Host; eine **vierte** liegt am separaten Cloud-VPS-Shield (anderer Provider, eigene MAC dort) — zusammen **4 Public-IPv4**:

Public-IP	MAC-Zuweisung	Zugewiesen an
`<HOST-PUBLIC-IP>`	Host-eigene MAC	Proxmox VE Host — **nur Out-of-Band-Management**, isoliert vom Datenpfad
`<EDGE-LAN-PUBLIC-IP>`	Separate MAC (Provider-zugewiesen)	Edge-LAN-Firewall (VMID 100, OPNsense)
`<EDGE-DMZ-PUBLIC-IP>`	Separate MAC (Provider-zugewiesen)	Edge-DMZ-Firewall (VMID 102, OPNsense)
`<SHIELD-PUBLIC-IP>`	Separate MAC (anderer Provider)	Cloud-VPS-Shield — Reverse-Ingress zur DMZ (separater Provider-Standort)

Wichtig: Die **Host-eigene Public-IP dient ausschließlich dem Out-of-Band-Management** und ist vom produktiven Datenpfad getrennt. Der Hypervisor liegt nicht im Datenfluss der Workloads.

Off-Site-Backup-Target

Feld	Wert
Backup-Ziel	Off-Site-Storage des Anbieters (eigenes Sub-Account, eigenes Passwort)
Protokoll	CIFS/SMB-Mount über den Backup-Pfad
Verschlüsselung	Verschlüsseltes PBS-Datastore-Repository (Off-Site)
Pfad PBS-Datastore	`/<sub-account>/pbs-datastore/` (auf dem Off-Site-Storage)
Pfad PVE-Mount	`/mnt/pbs-storagebox/` (auf dem PVE-Host)
Pfad LXC-Bind-Mount	`/mnt/datastore/storagebox` (innerhalb des Backup-LXC)

Sicherheits-Argument MAC-Filtering

Das MAC-Filtering ist die erste Verteidigungsebene vor der eigenen Firewall. Ohne die zugewiesene MAC erreicht ein Paket die VM gar nicht. Das verhindert: - IP-Spoofing von anderen Kunden im selben Provider-Subnetz - MAC-Flooding im Provider-Backbone - Failover-Angriffe bei IP-Übernahmen ohne Kontroll-Wechsel

---

3. VMID-Konvention

Statt VMs in der Reihenfolge ihrer Erstellung durchzunummerieren, folgt die Plattform einer funktionalen, dependency-basierten VMID-Konvention. Die VMID verrät auf einen Blick, welche Schicht eine Maschine bedient.

Schema

```
100-199    Firewall-Zone (produktiv)
200-299    Pentest / Security-Tools
300-399    Database-Layer
400-499    Development (Dev-Hosts, Test-VMs)
500-599    Reserve
600-699    Backup / Monitoring-Layer
700-899    Reserve (Queue, Storage, Identity, wenn konkret)
900-999    Reserve / Inaktive
1000-1999  Application-Layer (HA-fähig, breit skalierbar)
2000-2999  Docker-Host-Cluster (HA-fähig)
3000+      Future / weitere Mandanten-Ranges
```

Begründung der Reihenfolge

Bottom-up nach Dependency: - Firewalls / Security = Netzwerk-Grundlage - Database = Daten-Fundament (Apps brauchen DBs) - Backup = quer zu allem, eigene Range, isoliert - Application = oben drauf, kann skalieren - Docker-Hosts = Container-Runtime, parallel zu klassischen App-VMs

Bewusste Skalierungs-Entscheidung: - 100er-Blöcke für statische Komponenten (FW/DB/Dev/Backup) - 1000er-Blöcke für Application + Docker: Apps wachsen horizontal in HA-Setups und brauchen viel Platz

---

4. Inventar (anonymisiert, gruppiert)

Auszug aus dem Inventar, repräsentativ pro Range. Service-Namen sind generisch gehalten; produkt- und hostbezogene Bezeichner wurden entfernt. Insgesamt laufen 35+ Gäste (29 VMs + 8 LXC).

Firewalls (100-103)

VMID	Rolle	Firewall-Engine	Public-IP	Zone
100	Edge LAN	OPNsense	`<EDGE-LAN-PUBLIC-IP>`	Edge

VMID	Rolle	Firewall-Engine	Public-IP	Zone
101	Core LAN, Admin-Hub	pfSense	intern (RFC1918)	Core-LAN
102	Edge DMZ	OPNsense	`<EDGE-DMZ-PUBLIC-IP>`	Edge
103	Core DMZ	pfSense	intern (RFC1918)	Core-DMZ

Naming-Konvention nach Funktion: Edge LAN = OPNsense, Edge DMZ = OPNsense, Core LAN = pfSense, Core DMZ = pfSense. Insgesamt **4 Firewall-VMs**, die den eingehenden und internen Traffic segmentieren.

Pentest / Security-Tools (200)

VMID	Zweck
200	Pentest-VM, sitzt im DEV-VLAN

Database-Layer (300-301)

VMID	Service	Netz
300	Redis (Cache, Sessions, Queue) auf Debian, native VM statt Container	internes /29
301	PostgreSQL (Plattform + interne Apps), mit Row-Level-Security	internes /29

Hinweis zur Architektur-Entscheidung: kritische Infrastruktur (PostgreSQL, Redis) läuft bewusst als native systemd-VM, nicht als Container. So eliminiert man einen Container-Daemon-SPOF und macht VM-Snapshots im Backup trivial. PostgreSQL setzt Row-Level-Security für die Mandanten-Isolation ein.

Development (400-402)

VMID	Zweck
400	Entwicklungs-Server 1
401	Entwicklungs-Server 2 (Main-Dev)
402	Generische Test-/Sandbox-VM

Backup-Layer (600)

CT/VMID	Service	Netz / Bridge
CT 600	Proxmox Backup Server (LXC, unprivileged, Debian), Datastore auf Off-Site-Storage, Backup-Target für alle PVE-Gäste	internes Backup-/29 (isolierte Bridge)

Architektur-Detail: - Container-Type: **Unprivileged** (sauberer als privileged für Backup-Workload) - Features: nesting, FUSE, keyctl - Protection: aktiviert (kein versehentliches Löschen) - Ressourcen: 2 vCPU, 2 GB RAM, 10 GB Rootdisk auf dem ZFS-Pool - Mountpoint: `mp0` = Bind-Mount Host `/mnt/pbs-storagebox/pbs-datastore` zu LXC `/mnt/datastore/storagebox` - Web-UI: nur über die isolierte Backup-Bridge erreichbar, Zugriff via SSH-Tunnel oder Out-of-Band-WireGuard - SSH: aktiv mit Key-Auth (LXC-Zugang via PVE-Host oder VPN)

LXC wurde hier bewusst gewählt: ein Backup-Dienst braucht keinen eigenen Kernel, läuft mit 2 GB RAM ressourcenschonend und ist in Minuten neu deploybar.

Reserve (900-901)

VMID	Status
900	Windows-Server mit Hyper-V-Rolle, aktuell inaktiv (provisioniert für mögliche spätere Windows-Tests). Hinweis: beim ersten PBS-Backup erschien eine Microsoft-UEFI-2011-Cert-Warnung (läuft Mitte 2026 aus). Bei Aktivierung `qm enroll-efi-keys 900` ausführen.
901	Test-Firewall (experimentell, für Konfig-Tests vor Produktiv-Übernahme)

Application-Layer (1000-1003)

VMID	Service	VLAN-Zuordnung
1000	Backend (NestJS API)	APP_API (internes /29)
1001	Core-Service (Steuerungs-/Koordinationsschicht)	APP_CORE (internes /29)
1002	Frontend / Landing	APP_WEB
1003	Office-Collaboration (Collabora Online)	OFFICE-WORKSPACE

Docker-Host-Cluster (2000)

VMID	Service	Container-Zahl
2000	Docker-Host (Microservices via macvlan)	mehrere Dutzend Container (Reverse-Proxy, Mail-Stack, SIEM, Log-Aggregation, Monitoring, Messaging, Tunnel-Daemon u.a.)

LXC-Dienste (leichtgewichtig)

Neben dem Backup-LXC laufen weitere LXC-Container für leichtgewichtige Dienste, die keinen eigenen Kernel benötigen: **CI-Runner, Cert-Dienst, Docs-Dienst und Notify-Dienst**. Diese Trennung (8 LXC neben 29 VMs) hält das Setup ressourcenschonend, ohne die Isolation der kritischen Workloads aufzugeben.

---

5. Proxmox-Bridge-Layout (Überblick)

Kein Shared-Netz zwischen Firewalls. Jeder Firewall-zu-Firewall-Link ist eine eigene Proxmox-Bridge. Zusätzlich gibt es dedizierte Admin-Bridges für eine sternförmige Management-Topologie um die Core-LAN-Firewall sowie eine isolierte Backup-Bridge für PBS.

WAN-Anbindung + physisch

Bridge	Typ	VLAN-aware	Zweck
(physische NIC)	Physical NIC	n/a	Provider-Uplink (eine physische NIC)
`vmbr0`	Linux Bridge	Yes	Haupt-Bridge Public, trägt die drei Host-Public-IPs über MAC-Filter
Hetzner vSwitch	VLAN-Interface	VLAN 4000	Privates L2 zur Cloud, MTU 1400

Backup-/Management-Bridge (isoliert von Produktiv-VLANs)

Bridge	Typ	VLAN-aware	Zweck
`pbs01`	Linux Bridge	No	Internal-only Bridge für den PBS-Backup-Pfad (internes Backup-/29). Outbound via firewalld-MASQUERADE über `vmbr0` (zum Off-Site-Storage). Kein L2-Kontakt zu Produktiv-VLANs.

Sicherheits-Begründung: PBS sitzt auf demselben Isolations-Level wie PVE selbst (Management-Tool, nicht Produktiv-VM). Eine Kompromittierung des PBS-LXC führt nicht zu L2-Zugang ins Produktiv-Netz, da `pbs01` nur einen Layer-3-Pfad über den PVE-Host (firewalld) hat.

Rescue-/Wartungs-Bridges

Jede Firewall hat eine dedizierte Rescue-Bridge, an der im Notfall eine Wartungs-VM direkt angeschlossen werden kann. Analog zum physischen Setup, wo man bei einem Netzwerk-Problem einen Laptop direkt an Switch/Router hängt.

Bridge	Zweck
`edge_lan`	Rescue-Port für Edge-LAN-Firewall
`edge_dmz`	Rescue-Port für Edge-DMZ-Firewall
`core_lan`	Rescue-Port für Core-LAN-Firewall
`core_dmz`	Rescue-Port für Core-DMZ-Firewall

Haupt-Transits + VLAN-Trunks

Bridge	Mitglieder	Zweck	Transit-Net
`br_lan01`	Edge-LAN + Core-LAN	Transit Edge-LAN zu Core-LAN	dediziertes /30
`br_dmz01`	Edge-DMZ + Core-DMZ	Transit Edge-DMZ zu Core-DMZ	dediziertes /30
`vmbr2_VLAN`	LAN-Service-VMs	802.1Q-Trunk, Core-LAN routet, Service-VMs mit VLAN-Tag	n/a
`dmz_vlan01`	DMZ-Service-VMs	802.1Q-Trunk, Core-DMZ routet	n/a

Cross-Zone + Admin-Bridges (sternförmig um die Core-LAN-Firewall)

Bridge	Mitglieder	Zweck
`inter_fw`	Core-LAN zu Core-DMZ	Cross-Zone-Bridge für Admin-Pfad + Mail-Stream
`lan_opn`	Core-LAN zu Edge-LAN	Admin-Bridge: Core-LAN administriert Edge-LAN GUI/SSH

Bridge	Mitglieder	Zweck
`dmz_opn`	Core-LAN zu Edge-DMZ	Admin-Bridge: Core-LAN administriert Edge-DMZ GUI/SSH

6. vNIC-Zuordnung der Firewall-VMs

VM	vNICs / Bridges
100 Edge LAN	`vmbr0` (WAN via MAC), `br_lan01` (Transit), `lan_opn` (Admin), `edge_lan` (Rescue, inaktiv)
101 Core LAN	`br_lan01` (Transit), `vmbr2_VLAN` (LAN-Trunk), `inter_fw` (Cross-Zone), `lan_opn` + `dmz_opn` (Admin), `core_lan` (Rescue, inaktiv)
102 Edge DMZ	`vmbr0` (WAN via MAC), `br_dmz01` (Transit), `dmz_opn` (Admin), `edge_dmz` (Rescue, inaktiv)
103 Core DMZ	`br_dmz01` (Transit), `dmz_vlan01` (DMZ-Trunk), `inter_fw` (Cross-Zone), `core_dmz` (Rescue, inaktiv)
CT 600 (PBS)	`pbs01` only (internes Backup-/29)

7. Storage / ZFS

Der Host nutzt **ZFS** als Storage-Backend für VMs und Container — mit Snapshots, Replication-Fähigkeit und Datenintegrität auf Block-Ebene.

Pool	Typ	Zweck
`local`	Directory	Templates, lokale Snippets, Dump-Verzeichnis
`local-zfs`	ZFS	ZFS-Storage-Pool für VMs + Container, primär für Produktiv-Gäste
`pbs-storagebox`	Proxmox Backup Server	Aktives Backup-Target (chunk-basiertes Dedup, inkrementell, Off-Site, verschlüsselt)

Empfehlungen

- **Primär für VMs/CTs:** der ZFS-Pool (Snapshots, Replication-fähig)
- **Backups:** `pbs-storagebox` (PBS auf CT 600), produktiv und Restore-getestet

8. Backup / Disaster-Recovery (Proxmox Backup Server)

Aktueller Stand (produktiv)

Primär-Backup-Stack: - **CT 600:** Proxmox Backup Server als unprivileged LXC - **Datastore:** chunk-basiertes, verschlüsseltes Dedup-Repository auf dem Off-Site-Storage - **PVE-Storage `pbs-storagebox`:** API-Token-basierter Push von PVE zu PBS (Token mit Role `DatastoreBackup`)

Architektur:

```
[PVE-Host] --pbs01 (isolierte Backup-Bridge)-- [CT 600 PBS]
                                                    |
                                                    v (CIFS via vmbr0 + firewalld-MASQ)
                                          [Off-Site-Storage]
                                                    |
                                                    v
                                          /pbs-datastore/.chunks/  (verschlüsselt)
```

Es gibt **keinen L2-Pfad** vom Backup-LXC zu den Produktiv-VLANs — der Backup-Layer ist physisch und logisch vom Datenpfad getrennt.

Schedules:

Job	Schedule	Lauf	Was
Backup	`02:00` daily	PVE-Datacenter-Job	vzdump aller produktiven VMs/CTs zu PBS
Prune	`*-*-* 03:00` daily	PBS Datastore-intern	alte Snapshots nach Retention löschen
Garbage Collection	`sat 04:00` weekly	PBS Datastore-intern	verwaiste Chunks aufräumen

Retention (auf PBS-Datastore-Ebene konfiguriert, nicht im PVE-Backup-Job): - `keep-daily: 7` - `keep-weekly: 4` - `keep-monthly: 6` - `keep-yearly: 1` (optional)

Backup-Job-Selektion: - Alle Firewalls (100-103) - Datenbanken (300, 301) - Dev-VMs (400-402) - Application-Layer (1000-1003) - Docker-Host (2000) - Optional: Pentest-VM (200), Reserve-VM (900) - **Nicht** der PBS-LXC selbst: Self-Backup ist sinnlos, der Datastore liegt ohnehin extern und der LXC ist in

etwa 10 Minuten neu deploybar.

Pro-Service-Backups (zusätzlich, Defense-in-Depth)

Komponente	Tool	Frequenz	Aufbewahrung
PostgreSQL	pg_dump + WAL-Archiving	stündlich / kontinuierlich	RPO 15 min
Redis	AOF-Snapshots	kontinuierlich	RPO < 1 min
Docker-Volumes	Restic	täglich	14 daily + 6 weekly
Mail-Stack	Native + WORM-Archiv	täglich + Echtzeit	10 Jahre (GoBD)

Off-Site-Status

Aktuell: Das Off-Site-Storage-Ziel steht in einem physisch getrennten Rechenzentrum des Anbieters; die Backups sind verschlüsselt und Restore-getestet. Das deckt das 3-2-1-Muster für das aktuelle Single-Server-Setup ab. Echte Provider-Redundanz (zweiter, unabhängiger Anbieter) ist noch nicht etabliert und als Ausbaustufe vorgemerkt (z.B. zweites PBS-Target via S3-Endpoint oder rclone-Sync; alternativ ein eigenes PBS an einem zweiten physischen Standort, angebunden via WireGuard-Tunnel).

Disaster-Recovery-Runbook

Szenario A: PVE-Host komplett verloren, Off-Site-Storage intakt 1. Neuen PVE-Host aufsetzen (Robot-Rescue zu Reinstall) 2. Bridge `pbs01` nachbauen (siehe Section 5) 3. CT 600 neu deployen 4. CIFS-Mount + Bind-Mount wie im Original-Setup 5. PBS findet die existierenden Chunks auf dem Off-Site-Storage, Datastore-Re-Import via `proxmox-backup-manager datastore add ...` 6. Restore der VMs aus PBS

Szenario B: PBS-LXC kaputt, Off-Site-Storage intakt - Schritte 3-5 oben. Die Datastore-Chunks bleiben auf dem Off-Site-Storage erhalten.

Szenario C: Off-Site-Storage verloren - Aktuell der schmerzhafte Fall (noch kein zweites, unabhängiges Off-Site). Genau deshalb steht die Provider-Redundanz auf der Ausbau-Liste.

RTO / RPO

Kategorie	RTO	RPO
Full-System (alle VMs, vom PBS)	< 4 Stunden	24 Stunden
Single-VM-Restore (vom PBS)	< 30 Minuten	24 Stunden
File-Level-Restore aus VM-Backup (PBS-Feature)	< 10 Minuten	24 Stunden
PostgreSQL (Point-in-Time)	< 1 Stunde	15 Minuten
Mail-Stack	< 2 Stunden	24 Stunden
Firewall-Configuration (XML-Export)	< 15 Minuten	vor jeder Änderung (manueller Export)

---

9. Host-Firewall + Out-of-Band-Management (Einbahn-Vertrauen)

Host schützt sich selbst — unabhängig von den Firewall-VMs

Der Host **hostet zwar 4 Firewall-VMs (OPNsense/pfSense)**, verlässt sich für seinen eigenen Schutz aber **nicht** auf sie. Das ist ein bewusstes **Einbahn-Vertrauen**:

- **Host-Firewall: firewalld**, Default-Zone `public` = **Default-Deny**. Öffentlich erreichbar ist nur **WireGuard-UDP**.
- **Management ausschließlich über den WireGuard-Tunnel** (Zone `trusted`) — Out-of-Band, unabhängig von den produktiven Firewall-VMs.
- `pve-firewall` **ist NICHT in Benutzung** — die Host-Absicherung läuft vollständig über firewalld, eine bewusste Reduktion auf einen einzigen, klar auditierbaren Firewall-Layer am Host.

So bleibt der Hypervisor erreichbar und geschützt, selbst wenn eine Firewall-VM ausfällt oder kompromittiert wird. Der Schutz des Hosts hängt nicht an den VMs, die er hostet.

Drei separate Management-Pfade

Pfad	Aufgabe	Wann nutzen
Robot-KVM des Anbieters	Hardware-Konsole, Rescue-System, BIOS	Server-Crash, Boot-Probleme, Disk-Recovery
PVE Web-UI (nur über WireGuard)	Hypervisor-Lifecycle, VM-Console	Normaler Admin-Zugriff
Out-of-Band-WireGuard	Verschlüsselter Remote-Zugriff zu PVE-Web-UI und PBS-Web-UI	Unterwegs-Admin-Zugriff

PBS-Zugriff (via VPN oder SSH-Tunnel)

Die PBS-Web-UI liegt auf der isolierten Backup-Bridge und ist nicht direkt aus dem Internet erreichbar. Zwei Zugriffswege:

Option A, über Out-of-Band-WireGuard (`wg0`): - WireGuard-Client mit Route zum internen Backup-/29 über das Host-VPN-Gateway konfigurieren - Browser: `https://<pbs-internal-ip>:8007` - Voraussetzung: firewalld `wg0` in der trusted-Zone

Option B, SSH-Tunnel-Forwarding (ad-hoc):

```
ssh -L 8007:<pbs-internal-ip>:8007 root@<pve-host>
# Browser: https://localhost:8007
```

Sicherheits-Grundsatz

PVE hat keinen Netzwerk-Pfad zu der Core-LAN-Firewall oder zu den Produktiv-VMs. Es kann VMs lifecycle-steuern (Start/Stop/Migrate), aber nicht in deren Produktiv-Traffic eingreifen. Bei einem PVE-Kompromiss sind VMs abschaltbar, die Produktivdaten aber nicht direkt zugreifbar. Dasselbe gilt für den PBS-LXC: er liegt auf der separaten Bridge `pbs01` und hat keinen L2-Pfad zu den Produktiv-VMs.

---

10. Nested Virtualization (Reserve-VM 900)

VMID 900, aktuell inaktiv, provisioniert für mögliche spätere Nutzung (z.B. Windows-Agent-Tests, Domain-Join-Tests, Windows-spezifische Software-Tests).

Voraussetzungen / TODOs bei Aktivierung: - PVE CPU-Flag `+vmx` oder `host` für Intel VT-x Passthrough (aktiv) - Windows Server mit Hyper-V-Rolle (Lizenz nötig) - Nested-Virtualization-Setup in der PVE-VM-Config - EFI-Cert-Update vor Aktivierung: `qm enroll-efi-keys 900` (Microsoft-UEFI-2011-Cert läuft Mitte 2026 aus, neue Cert nötig) - EFI/TPM-Disks tragen noch alte Disk-Bezeichnungen aus der VMID-Migration, bei Aktivierung optional via `zfs rename` normalisieren (kosmetisch)

---

11. VMID-Migration-Pattern

Im Zuge der Einführung der neuen VMID-Konvention wurden mehrere VMs migriert, bottom-up nach Dependency (Firewall, dann DB, dann Dev, dann App, dann Docker-Host). Die Migration lief skriptgesteuert auf dem PVE-Host.

Standard-Pattern pro VM:

```
# 1. VM herunterfahren
qm shutdown <alte-id>

# 2. Config-Datei umbenennen
mv /etc/pve/qemu-server/<alte-id>.conf /etc/pve/qemu-server/<neue-id>.conf

# 3. ZFS-Disk-Volumes umbenennen
zfs rename rpool/data/vm-<alte-id>-disk-0 rpool/data/vm-<neue-id>-disk-0

# 4. Disk-Referenzen in der Config aktualisieren
sed -i "s/vm-<alte-id>-disk/vm-<neue-id>-disk/g" /etc/pve/qemu-server/<neue-id>.conf

# 5. VM starten + Verifikation
qm start <neue-id>
qm status <neue-id>
ping <vm-ip>
```

IPs blieben unverändert, daher mussten keine Anwendungs-Connection-Strings angepasst werden. Geändert wurde nur die PVE-interne Verwaltung.

---

12. PVE-CLI Cheatsheet (Auszug)

VM- und LXC-Management

```
qm list                                 # Status aller VMs
qm start <vmid> / qm shutdown <vmid>    # Start / ACPI-Shutdown
qm clone <vmid> <new-vmid> --name <name> --full
qm resize <vmid> <disk> +<size>G        # Disk vergrößern

pct list                                # Status aller LXC
pct enter <ctid>                        # In LXC einsteigen
pct exec <ctid> -- <command>            # Befehl im LXC ohne Login
pct set <ctid> -protection 1            # Anti-Delete an
```

Storage / ZFS

```
pvesm status                            # Storage-Pools auflisten
zfs list                                # ZFS-Datasets
```

Backup (PBS-basiert)

```
# Backup manuell an PBS schicken
vzdump <vmid> --storage pbs-storagebox --mode snapshot --compress zstd

# PBS-Datastore-Status (vom PVE-Host aus, in den LXC delegiert)
pct exec 600 -- proxmox-backup-manager datastore show <datastore>
```

```
# Restore einer VM aus PBS via PVE-CLI
qmrestore <pbs-storage>:<backup-id> <new-vmid>
```

Netzwerk / Bridges / Host-Firewall

```
bridge vlan show                        # Bridge-VLAN-Filter prüfen
firewall-cmd --get-active-zones         # firewalld-Zonen (public / trusted)
firewall-cmd --zone=public --list-all   # Default-Deny + erlaubte Dienste (WireGuard-UDP)
firewall-cmd --zone=public --query-masquerade
```

CIFS-Mount-Diagnose (für Off-Site-Storage)

```
mount | grep cifs                       # Mount-Status
umount /mnt/pbs-storagebox && mount -a   # Re-Mount nach Wartung
ls -la /mnt/pbs-storagebox/pbs-datastore/ # UID/GID-Mapping verifizieren
```

---

13. Compliance-Referenzen (BSI IT-Grundschutz)

Compute-Layer-relevante Bausteine:

BSI-Baustein	Beschreibung	Implementierung
SYS.1.5	Virtualisierung	Proxmox VE 9.1 nativ, dedizierte Bridges pro Transit, MAC-Bindung, isolierte Backup-Bridge
SYS.1.3	Server unter Linux	Debian auf allen VMs/LXC
NET.1.1	Netz-Segmentierung	firewalld Default-Deny am Host, kein gemeinsames L2 zwischen Firewalls, Out-of-Band-WireGuard-Management
OPS.1.1.2	Backup/Restore	PBS produktiv (CT 600), verschlüsselt + Off-Site + Restore-getestet, DR-Drill quartalsweise (geplant)
OPS.1.1.3	Naming-Convention	VMID-Schema (Bottom-up by Function) + funktionale Namen
CON.3	Datensicherung	Multi-Tier-Backup (PBS Off-Site verschlüsselt + Pro-Service-Tools + WORM für Mail), Provider-Redundanz als Ausbaustufe vorgemerkt

---

14. Eingesetzte Kompetenzen (Zusammenfassung)

- **Proxmox VE 9.1** als nativer Hypervisor: VM- und LXC-Lifecycle, ZFS-Storage, Bridge-Layout, Nested Virtualization. Bewusster VM/LXC-Split (29 VMs + 8 LXC = 35+ Gäste) nach Isolations- und Ressourcen-Bedarf.
- **Netzwerk-Segmentierung auf Hypervisor-Ebene:** dedizierte Bridges pro Firewall-Transit (kein gemeinsames L2), 802.1Q-VLAN-Trunks, Hetzner vSwitch (VLAN 4000, MTU 1400) als privates L2, isolierte Backup-Bridge ohne L2-Pfad ins Produktivnetz, sternförmige Admin-Topologie.
- **Host-Härtung mit Einbahn-Vertrauen:** firewalld (Default-Deny, public; pve-firewall bewusst nicht genutzt), Management ausschließlich über Out-of-Band-WireGuard — der Host schützt sich unabhängig von den 4 Firewall-VMs, die er hostet.
- **Backup / DR mit Proxmox Backup Server:** unprivileged LXC, chunk-basiertes Dedup, verschlüsselter Off-Site-Datastore, Retention-Policy, getestetes Disaster-Recovery-Runbook mit definierten RTO/RPO-Zielen.
- **Infrastructure as Code:** Ansible für reproduzierbare, automatisierte Provisionierung und Konfiguration.
- **Strukturiertes Inventar-Management:** dependency-basierte VMID-Konvention, skriptgesteuerte Migration ohne Änderung der Anwendungs-Connection-Strings.
- **Out-of-Band-Management:** drei getrennte Management-Pfade, VPN-only-Zugriff auf sensible UIs.
- **Compliance-Bewusstsein:** Mapping auf BSI-IT-Grundschutz-Bausteine, GoBD-konforme Mail-Aufbewahrung.

Anonymisierte Arbeitsprobe IT-Systemintegration. Produkt-, Kunden-, Host- und Provider-spezifische Bezeichner sowie öffentliche und interne Adressen wurden entfernt; die Architektur-Patterns sind unverändert.