ICANN86 · SEVILLE · 2026

Proxmox VE for ccTLD Operations

Open Source Virtualization

Virtualization · Automation · Monitoring

Luis Diego Espinoza

Technical Advisor · NA-NIC

Context & Motivation

- A small ccTLD needs to operate: **authoritative DNS**, RDAP/WHOIS, registry portal, database, monitoring — on a **limited budget** with a small team
- Typical setup: **2–3 physical servers**, 2–4 engineers, must ensure uptime 24/7
- Proprietary hypervisors (VMware, Hyper-V) = per-socket licensing costs not justified for a small ccTLD operation
- Anycast DNS service offloads authoritative DNS — **Proxmox handles backend services**
- **Proxmox VE** = zero licensing cost, enterprise features, Debian-based, fully open source

> **Goal:** evaluate Proxmox VE as a practical, open-source alternative for small ccTLD infrastructure — resilient, automatable, and operable with a small team.

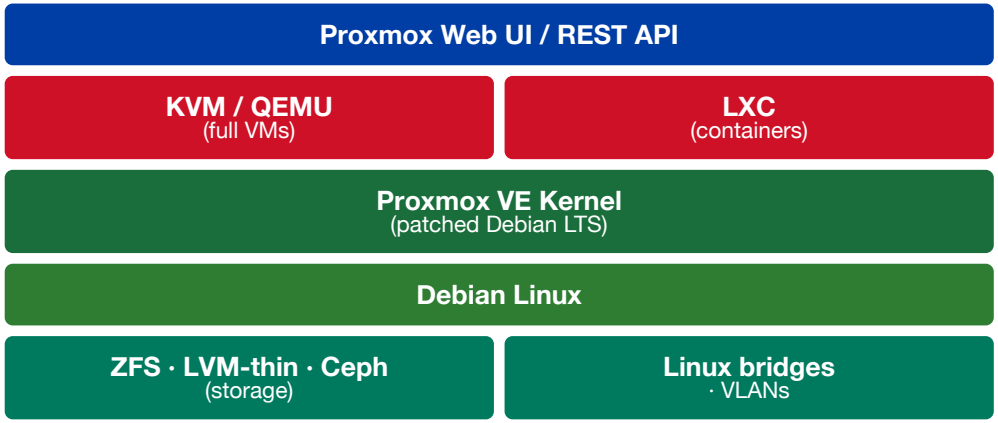
SECTION 1 OF 6

Architecture

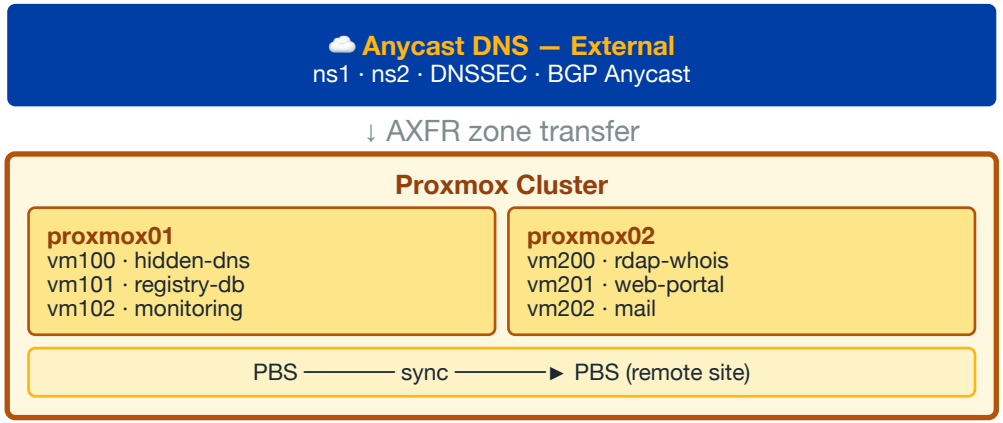
Proxmox stack · ccTLD topology · Anycast DNS

Architecture: Stack & ccTLD Topology

Proxmox VE · Software Stack



ccTLD · Deployment Topology



SECTION 2 OF 6

Features, Benefits & Security

Why Proxmox?

Features, Benefits & Security

Operational & Cost

- **Zero licensing cost** — community repo, fully functional
- VMs (full OS isolation) + LXC containers on same host
- **REST API** — everything scriptable: bash, Terraform, Ansible
- **RBAC** — delegate tasks without granting root access

Stability

- **Debian Stable** base — LTS lifecycle, predictable patching
- Production deployments: universities, ISPs, hosting companies

Security

- **KVM hardware isolation** — breach in one VM cannot escape
  - CPU virtualization (Intel VT-x / AMD-V) enforces hard boundaries — unlike Docker containers
- Per-VM **stateful firewall** rules built-in
- **2FA** — TOTP / WebAuthn, no extra software required
- **Audit log** — every action recorded with user + timestamp
- TLS on API, Web UI, and all cluster traffic

> All services isolated from each other on **the same physical hardware** — cost efficient and secure

Service Layout on Proxmox

```
proxmox01  (Physical Server 1)
├── vm100  hidden-dns
│          NSD — hidden master DNS
├── vm101  registry-db
│          PostgreSQL — registry backend
└── vm102  monitoring
           Grafana + Prometheus + Monit

proxmox02  (Physical Server 2)
├── vm200  rdap-whois  (RDAP + WHOIS)
├── vm201  web-portal  (registrar portal)
└── vm202  mail        (Postfix)

[Cloud] Anycast provider — ns1, ns2, DNSSEC
```

- Hidden master pushes signed zones to **Anycast provider** via AXFR
- RDAP/WHOIS is the **public query interface** for domain registrations
- **Web portal** = registrar-facing EPP client + self-service
- **Ceph RBD** shared storage across both nodes — lose a node, VMs keep running
- **PBS** backs up to a remote site — scheduled, deduplicated, off-site DR
- **Scale out** later: add a 3rd node for full HA with fencing

> **Anycast DNS** handles what matters most for uptime — the backend services can tolerate brief maintenance windows.

SECTION 3 OF 6

Fault Tolerance

Snapshots · Storage Backends · HA Clustering

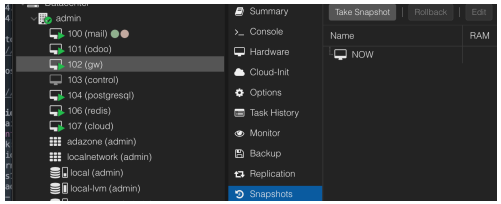
Snapshots & Cloning

- **Instant snapshot** before any upgrade — disk + memory state
- **Rollback in seconds** if the upgrade fails
- **Linked clone** → test environment from a snapshot in under 10 s
- **Full clone** → independent production-ready replica

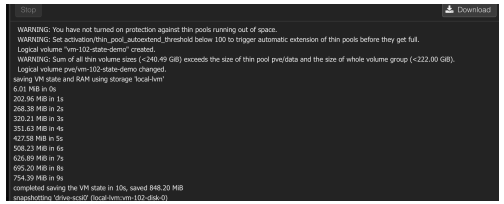
```
# Snapshot before upgrade
qm snapshot 100 pre-upgrade --vmstate

# Rollback if needed
qm rollback 100 pre-upgrade

# Full clone: vm100 → vm200
qm clone 100 200 --name rdap-replica --full
```

- **PBS backup** — incremental, deduplicated, runs on a schedule
  - `proxmox-backup-client backup vm/100/...`
  - Only changed blocks transferred — efficient over WAN
- **Remote PBS** — sync job ships backups to a second site automatically
  - PBS → PBS via TLS port 8007; only the delta each run

Ceph + PBS = no storage single point of failure: VMs run on any node — backups survive any single node loss.



Snapshot panel — vm102



Task log — 848 MB in 10 s

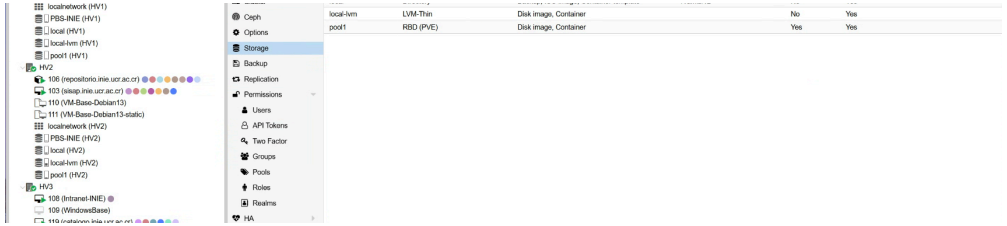
Storage Backends

Four backends — same Web UI

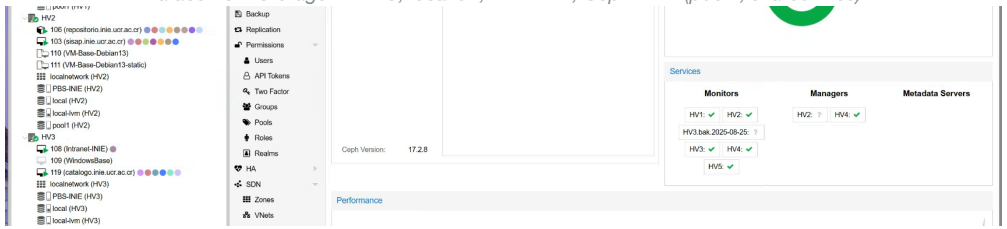
- **local** (Directory) — ISO images, container templates, PBS backups
- **local-lvm** (LVM-Thin) — VM disks on a single node; thin-provisioned
- **ZFS** — VM disks + instant snapshots + `zfs send` replication to peer
- **Ceph RBD** (`pool1`) — **shared across all nodes**, enables live migration and HA without NFS or DRBD

Backup: Proxmox Backup Server (PBS)

- **Incremental + deduplicated** — only changed blocks sent each run
- Retention policies: daily / weekly / monthly, automatic pruning
- **File-level restore** — recover a single file without restoring full VM



Datacenter › Storage — PBS, local dir, LVM-thin, Ceph RBD (pool1, shared=Yes)



Built-in Ceph dashboard — HEALTH_OK, shared VM disk pool across all nodes

HA Clustering & Live Migration

- **Live migration** — move a running VM to the other node with zero downtime
  - Use for: hardware maintenance, firmware updates

```
qm migrate 100 proxmox02 --online
```

- **Corosync cluster** — VMs auto-restart on node failure
- **HA groups** — set which VMs must always be running
- **Fencing** via IPMI/iDRAC prevents split-brain

> DNS is covered by Anycast. Backend VMs (RDAP, portal, DB) restart automatically on the surviving node in under 60 seconds.

```
HA failover timeline:
  proxmox01 loses power
    ↓  ~10 s  HA detects failure
  Fencing verifies node is down
    ↓  ~30 s  HA manager acts
  vm101 (registry-db) restarts
  vm200 (rdap-whois) restarts
    ↓  < 60 s  services restored
```

SECTION 4 OF 6

Scripting & Automation

pvesh · cloud-init · Terraform · Ansible

pvesh — Proxmox REST API from the CLI

- Everything the Web UI does, `pvesh` can do — same underlying REST API

```
# List all VMs as JSON
pvesh get /nodes/proxmox01/qemu --output-format json

# Create a VM from scratch
pvesh create /nodes/proxmox01/qemu \
  --vmid 201 --name "web-portal" --memory 2048 \
  --cores 2 --net0 virtio,bridge=vmbr0 \
  --scsi0 local-zfs:20 --ostype l26

# Start · stop · query
pvesh create /nodes/proxmox01/qemu/201/status/start
pvesh get    /nodes/proxmox01/qemu/201/status/current
```

> `pvesh` works **locally** without a token. Same REST API is reachable from any external script via `curl` + an API token scoped to specific operations.

Provisioning a VM — cloud-init Script

```
#!/bin/bash
VMID=201; NAME="web-portal"
IP="10.0.1.201"; GW="10.0.1.1"
TEMPLATE=9000  # cloud-init Debian base

qm clone $TEMPLATE $VMID \
    --name $NAME --full

qm set $VMID \
    --ipconfig0 ip=$IP/24,gw=$GW \
    --sshkeys   /root/.ssh/id_rsa.pub \
    --ciuser    admin \
    --memory    2048 --cores 2

qm start $VMID
until ssh admin@$IP true 2>/dev/null
do sleep 3; done
```

- Template 9000 = cloud-init Debian image, prepared **once**
- Sets IP, hostname, SSH key **at first boot** automatically — no manual console
- Provision 10 new VMs: loop over a config file, run in < 2 minutes

Infrastructure as Code

- **Terraform** — `bpg/proxmox` provider for full VM lifecycle as HCL
- **Ansible** — `community.general.proxmox` module
- Both use the same Proxmox REST API under the hood

SECTION 5 OF 6

Inventory & Monitoring

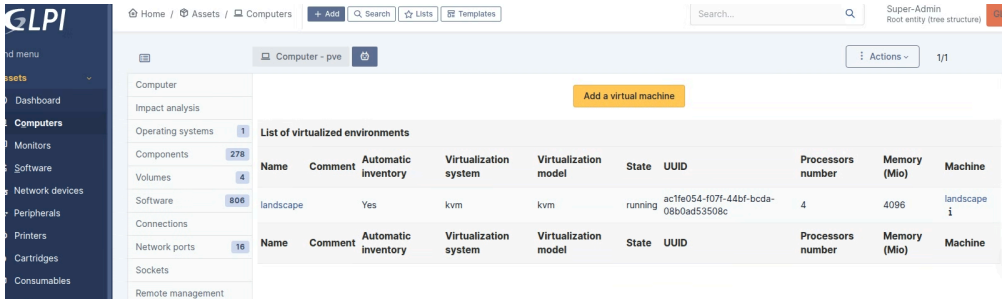
GLPI-Agent · Grafana · Prometheus · Monit

Automated Inventory with GLPI-Agent

- Agent installed **inside each VM**, runs daily via cron
- Auto-collects: CPU, RAM, disk, OS, network, **installed packages**
- Sends to GLPI server over HTTP — no manual spreadsheets ever

```
# /etc/cron.daily/glpi-agent
#!/bin/sh
glpi-agent \
    --server http://glpi.internal/
```

Asset database always current: **compliance audits**, change tracking, software inventory — automatic



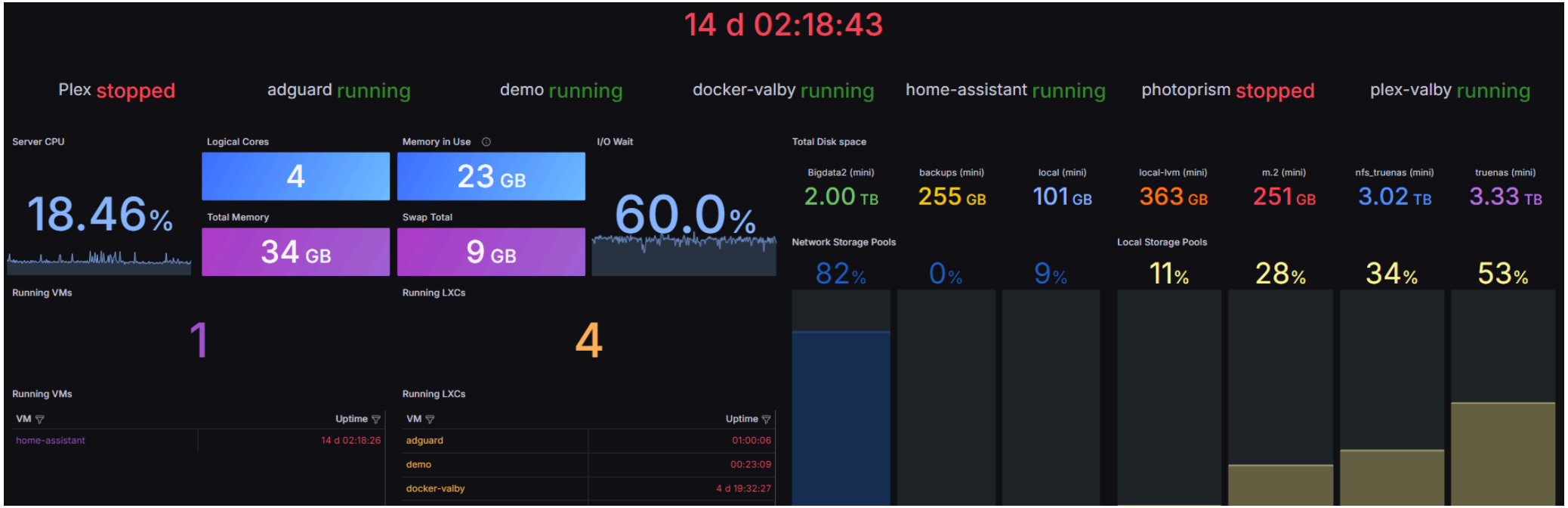
GLPI — virtualized environments on the Proxmox host: KVM VMs auto-discovered with UUID, CPU count, and RAM

Monitoring: Grafana + Prometheus + Monit

```
Proxmox node
  (built-in metrics exporter)
        │  Prometheus scrape
        ▼
   Prometheus ──► Grafana
                  alerts → email
                  (dashboard ID 10048)

Inside each VM — Monit watchdog:
  ✓ nsd / named process alive?
  ✓ PostgreSQL responding?
  ✓ disk < 85% ?
  → auto-restart or page on-call
```

- Proxmox metrics: node CPU, RAM, I/O, network, **per-VM** stats
- Grafana dashboard **10048** works out of the box with Proxmox
- **Monit** inside each VM = lightweight process watchdog
  - Restarts services automatically if they crash
  - Alerts before disk fills — before it becomes a crisis

Grafana — Proxmox Node & VM Dashboard



Grafana dashboard 20890 — per-node CPU, RAM, disk I/O and per-VM metrics from Proxmox built-in exporter

Key Takeaways

- **Zero licensing cost** — enterprise features, fully open source
- **Debian LTS base** — stable, familiar, long-term supported
- **Anycast DNS offloads** the most critical service to proven providers
- **REST API** — bash, Terraform, Ansible automation from day one
- **Ceph + PBS**: shared storage, snapshots, remote off-site backups
- **GLPI + Grafana + Monit** = complete operational picture

> **Right-sized for a small ccTLD:**
> 2 physical servers + Anycast DNS = a resilient, maintainable architecture for a small team.

- **Start small:** one Proxmox node, add the second for HA
- Community: forums.proxmox.com, active mailing lists

Questions?

forums.proxmox.com · pve.proxmox.com/wiki