

Proxmox Virtual Environment

Dalle *enterprise* a casa tua

Emiliano Vavassori

Linux Day 2025 Spring Edition
Creative LAB, Lunetta (MN) — 10 maggio 2025



Agenda

- Cos'è Proxmox Virtual Environment
- Sostenibilità del progetto e modello di *business*
- Supporto della *community*
- Approccio *enterprise* e "casalingo"
- Considerazioni sullo *storage* locale
- Pro e contro delle tre modalità di configurazione dello *storage*
- Esempio avanzato di configurazione dello *storage* locale
- Considerazioni su CPU
- Considerazioni su RAM
- Meccanismi di *passthrough*



Cos'è Proxmox Virtual Environment

- Un *hypervisor* di tipo 1
- Anche una soluzione di containerizzazione
- Basata su Debian GNU/Linux (più alcune *patch* per *performance*)
- Licenza GNU AGPLv3 ⇒ Software libero!

Feature di Proxmox VE — 1

- Virtualizzazione basata su KVM/QEMU
- Containerizzazione basata su LXC
- Gestione centralizzata (*clustering*) via web e CLI
- REST API (diversi *wrapper* già sviluppati)
- Funzionalità di backup/restore integrate
- Supporto ad altri motori d'autenticazione (LDAP, Active Directory, OpenID Connect)
- *High availability* (HA) e *live migrations*
- *Software Defined Networks* e *stack* di rete con tecnologie Linux
- Firewall

Feature di Proxmox VE – 2

Storage interni:

- LVM
- ZFS
- btrfs
- **Ceph Cluster**

Storage esterni:

- LVM
- iSCSI
- NFS
- SMB/CIFS
- Ceph
- GlusterFS

Sostenibilità del progetto e modello di *business*

Proxmox Server Solutions GmbH è l'azienda che sviluppa Proxmox VE dal 2005.

Il modello di *business* è basato sulla vendita di servizi di supporto (con diversi livelli).

Gli altri prodotti:

- Proxmox Mail Gateway (è stato il primo)
- Proxmox Backup Server
- Proxmox Datacenter Manager (alpha)

Supporto della community

Il metodo principale per contattare la *community* e farsi supportare è quello del forum ufficiale: forum.proxmox.com.

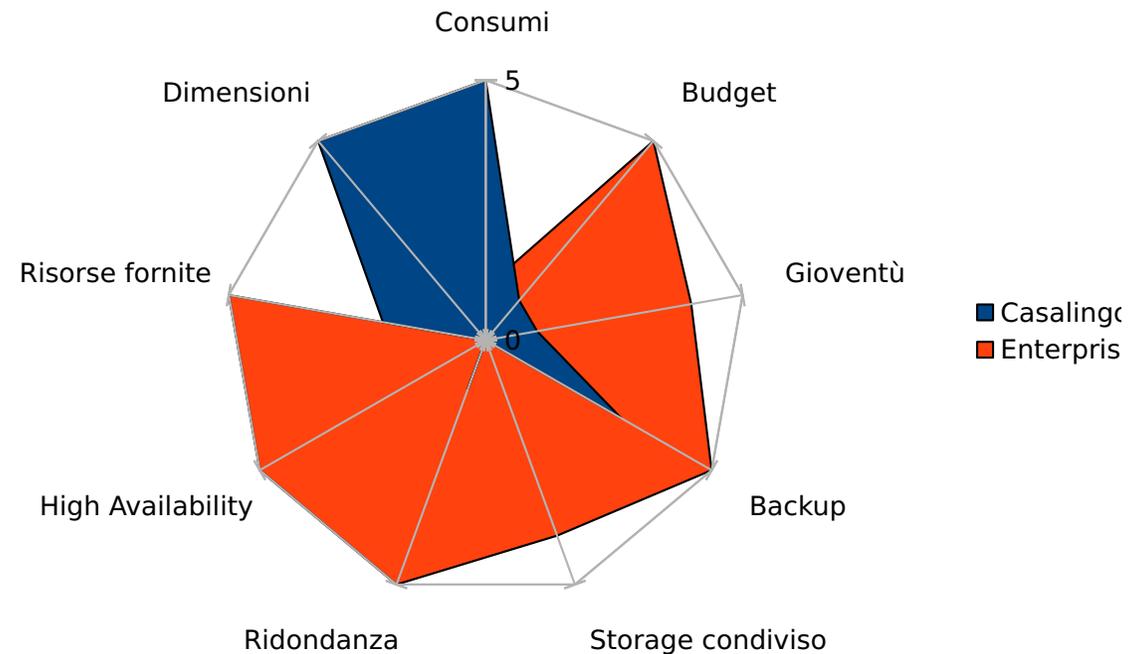
Esiste una *wiki* ricchissima di informazioni e molto aggiornata: pve.proxmox.com/wiki.

In italiano c'è un canale dedicato su Telegram: [Proxmox Italia](#).

Un'interessante risorsa per tutti è [Proxmox VE Helper-Scripts](#), che è una raccolta di script da poter far girare direttamente su Proxmox VE per ottenere diverse applicazioni già pronte.

Approccio *enterprise* o "casalingo"?

I due approcci sono quasi antitetici e tengono in considerazione esigenze differenti



Configurazione dello *storage* locale

- Utilizzo di un *hardware RAID controller*: non è sempre il meglio!
- Modalità di configurazione:
 - LVM-Thin
 - ZFS
 - btrfs
- Parametri avanzati differenti per ciascuna modalità
- ZFS e btrfs sono quelli che necessitano di più RAM per poter lavorare
⇒ tenerne conto nel dimensionamento!
- I dischi a sistema vengono usati *tutti* e per *tutta la loro dimensione* (ma: configurazione avanzata)

Caratteristiche comuni degli *storage*

- Snapshot (permette di estendere la funzionalità alle VM)
- sottovolumi
- Gestione logica dei dischi (fino a un certo punto)

Storage locale LVM-Thin: pro e contro

- Non supporta meccanismi di RAID (LVM non è ridondanza!)
 - RAID Controller hardware
 - Installazione Debian con `mdadm`, poi aggiornamento a Proxmox VE
- I dischi delle VM sono effettivamente dei *logical volume* di LVM
- Per via dell'estensione LVM-Thin, viene utilizzato solo lo spazio effettivamente occupato dai dati all'interno del disco
- A seconda dei casi d'uso, l'installer finisce per dedicare troppo spazio al sistema

Storage locale ZFS: pro e contro

- In Proxmox VE è stato portato OpenZFS/ZFS on Linux (versione attuale: 2.3.2)
- Supporta internamente l'uso di più dischi in ridondanza (RAID)
 - *Just a Bunch Of Disks* (JBOD)
 - con controller in modalità *Host Bus Adapter* (HBA)
- Uso della RAM considerevole (*ARC cache*, di default la metà della RAM)
- *Cache* opzionali, di lettura o scrittura, meglio se su dischi *flash*:
 - Cache scrittura: *ZFS Intent Log* (ZIL) ⇒ Log
 - Cache lettura: *Level 2 ARC* (L2ARC) ⇒ cache
- Sicuramente il metodo più flessibile e potente
- Supporto via interfaccia web non completo

Storage locale btrfs: pro e contro

- Supporterebbe internamente RAID (ma limitatamente a RAID 1 «*mirroring*», RAID 5 e 6: [no comment](#))
- Richiede molte risorse sul sistema (RAM ma anche CPU)
- Meno utilizzato
- A parità di macchina, spesso si preferisce ZFS (per stabilità e *feature set*)

Esempio di configurazione avanzata con ZFS — 1

- 2x dischi NVMe 120GB
- 3x dischi HDD 8TB
- RAM: almeno 26-32 GB dedicati a ZFS, resto per le VM

Esempio di configurazione avanzata con ZFS — 2

- All'installer si indica di usare solo i due dischi NVMe, si chiede di configurare in modalità `mirror` e si lascia 60GB di spazio (diminuire `hdspace` a 60GB)
- Si partiziona lo spazio libero con due partizioni: una da 8GB e una da 52GB circa (CLI)

Esempio di configurazione avanzata con ZFS — 3

- Viene creato un nuovo zpool `datastore` in modalità `raidz` (RAID 5) con i dischi rotativi; circa 16TB di spazio per i dischi delle VM e dei Container (via web)
- A questo zpool vengono agganciate le due *cache* (CLI)
 - *Cache* di lettura: in modalità *striping* sulle due partizioni da 52GB circa
 - *Cache* di scrittura: in modalità `mirror` sulle due partizioni da 8GB
- Si riconfigura `zfs_arc_max` per usare solo 26-32GB di RAM (CLI)
 - 2-8 GB per la sola presenza di ZFS
 - 1GB di RAM per ogni TB di storage "grezzo" (non considerando RAID)

Uno storage particolare: Ceph

Feature:

- File system condiviso (⇒ infrastruttura iperconvergente!)
- Supporto agli Snapshot
- *self-healing*
- Scalabile all'Exabyte
- Replica dei dati sui nodi ⇒ *fault tolerance*
- "Commodity hardware" 🤖
- Open source

Requisiti (e qui casca l'asino!):

- Sugeriti almeno 3 nodi identici
- CPU: più IOPS, più core richiesti
 - 1 core/proc/host
 - 100+ kIOPS ⇒ 4-6 core/proc/host
- RAM: 3-5 GB di RAM per il processo OSD + 1 GB per TB di storage
- Rete: richiesto 10Gbps

Considerazioni sulla CPU – 1

Caratteristiche notevoli della CPU:

- Numero di Core
- Numero di Thread - *Hyperthreading*
- Set di istruzioni (es. AVX2)
- Velocità di clock
- Consumo (*Thermal Design Point*, TDP)

Considerazioni sulla CPU – 2

In ambienti Enterprise si scelgono CPU *multicore* e *hyperthreading* e, possibilmente, si installano più processori (*socket*).

In ambienti casalinghi, magari è preferibile utilizzare CPU appena meno performanti ma più parsimoniose:

Modello	Clock (GHz)	Core	Thread	TDP (W)
Intel Core i7 6700	3.40	4	8	65
Intel Core i7 6700T	2.80	4	8	35

Considerazioni sulla RAM

- Dimensionamento anche sulla base del filesystem che si vuole utilizzare!
- Condivisione della RAM su più macchine
- *Ballooning*: recupero RAM inutilizzata da quella assegnata a una VM
- *Overprovisioning*: assegnare più RAM di quanta è fisicamente disponibile sul sistema

Dispositivi in *passthrough*

Mettere in passthrough un dispositivo hardware dell'host vuol dire **dedicarlo alla VM/CT** a cui viene associato.

Utile e ampiamente utilizzato per associare GPU a macchine virtuali specifiche:

- disegno CAD/3D, rendering, altre applicazioni grafiche
- *mining* di Cryptovalute
- Intelligenze Artificiali Generative

Necessari

- Supporto hardware dell'host (IOMMU: Intel VT-d, AMD AMD-Vi)
- Modifiche alla configurazione di Proxmox VE (== non abilitato di default!)

Riferimenti personali e link per approfondimento

Queste slides sono disponibili a questo indirizzo:

sys42.cc/ld25se-pve

Se avete domande o volete chiedermi qualcosa:

 [syntaxerrormmm\(at\)gmail.com](mailto:syntaxerrormmm(at)gmail.com)

 [@syntaxerrormmm](https://t.me/syntaxerrormmm)