Cozystack vs VMware

CNCF-native · Open-source (Aenix/Cozystack)

Open-source & CNCF-native vs platform with proprietary scripting and cloudlet billing.

Why Cozystack Wins

- **One operating model:**
  Pods + KubeVirt VMs in Kubernetes (single RBAC/policy/tooling).
- **Built-in services:**
  Managed Postgres (PITR→S3), first-class S3, Velero DR; all GitOps-driven.
- **Predictable economics & white-label multi-tenancy:**
  node/preset pricing, tenant quotas, brandable portal.

Choose Cozystack if you want a single Kubernetes-native platform for VMs + containers + data (DB/S3), need multi-tenant self-service, AI/GPU sharing, and seek lower TCO with GitOps and open standards.

VMware (vSphere/VCF + Tanzu)

- **Dual-domain reality:**
  vSphere for VMs, VKS/Tanzu for K8s, plus TMC for fleet (more surfaces).
- **Product sprawl/coupling:**
  DB/S3/DR commonly external or separate SKUs/processes.
- **Licensing & cadence complexity:**
  multi-SKU planning (VCF/NSX/TMC/VKS) and version alignment.

VMware—best applied when: you are deeply standardized on vSphere/NSX, rely on SRM/NSX advanced networking, and prefer to keep VM and Kubernetes as separate domains under the VMware enterprise operating model.

Category	Cozystack advantage	Why it matters	VMware status / trade-off
No vendor lock-in	Open-source, CNCF ecosystem	Trust, transparency, and easier hiring—no single-vendor dependency.	Proprietary solution
Costing	**Predictable node/preset** model	Budgetable TCO; easy chargeback	Very high price after Broadcom M&A. Multi-SKU (VCF/NSX/TMC/VKS) complexity
Licensing model	OSS platform; CNCF project, Apache 2.0	Always remains open source	Proprietary
Install footprint	Lightweight bundles; air-gapped friendly	Faster stand-up, easier replication	Heavier suite alignment (VCF cadence)
White-label	Branded portal & catalog (K8s/VM/DB/S3/LB)	MSP/enterprise self-service out of the box	Usually multiple consoles/portals
Tenancy	**Tenants/Sub-tenants,** quotas, guard-rails	MSP/enterprise self-service out of the box	Assembled from pools/NSX/TMC policies
GPU for AI	GPUs/vGPU for Pods and VMs under one scheduler	ML teams don’t juggle stacks	VM GPU mature; Pod GPU add-ons/coordination
Observability	Grafana / Victoria Metrics, Victoria Logs out of the box	No vendor lock; fast root-cause	Aria/TMC + add-ons (more parts/licensing)
APIs	Full REST + kubectl on CRDs; Terraform/Helm modules	Cozystack is consistent OSS	vSphere/Tanzu/NSX APIs
Team skills	Pure CNCF stack (K8s, Helm, KubeVirt, Cozystack itself)	Easier hiring, lower training debt	VMware-specific skills required
DB & data	Managed Postgres (PITR→S3) + first-class S3	Built-in DBaaS/backup flows, faster RTO	DB/S3 typically external stacks/licenses
GitOps	Flux as default for clusters, apps, infra	Fewer “snowflake” changes; easy rollbacks	Possible, but across multiple products
Networking	Cilium + KubeOVN + MetalLB	Simple, open, auditable	NSX powerful but licensed & heavier
Roadmap risk	OSS primitives; no SKU churn	Stable planning horizons	Portfolio/licensing changes to track
DR & backup	**Velero** for K8s/PV; PITR for DBs	Unified, repeatable DR drills	Mix of SRM/Velero/storage tools
Catalog extensibility	Helm/Operators for new services	Add anything, versioned in Git	Marketplace, but more product coupling
App migration	Helm repack + Velero namespace moves	Low-risk blue/green, same tools	TKC→CAPI + app toolchain per product
VM migration	CDI import (qcow2/raw) → **KubeVirt**	Keep images; unify ops afterward	HCX/vMotion (great for staying in vSphere)
Operating model	**Single plane:** Pods + **KubeVirt VMs** in Kubernetes	One RBAC, one policy set, one toolchain	Dual domains: vSphere for VMs, VKS/Tanzu for K8s
Capacity mgmt	Presets & quotas across CPU/RAM/**GPU/**Storage/Net	Tenant SLO guard-rails	Pools/limits across several planes