Cozystack vs Virtuozzo (Jelastic)

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

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

Why Cozystack Wins

- **Open source and standard integration**
  Open Source components, Kubernetes API, Helm, Terraform—your engineers already know it
- **Financial control and transparency**
  Per-physical node model. Just add it to hardware price
- **Unified workloads**
  Kubernetes + VMs in one control plane (KubeVirt).

Cozystack: an open cloud-native platform (K8s + KubeVirt + Managed DB + S3) with transparent per-node/preset pricing, a white-label UI, extensibility through standard tools (API/Terraform/Helm), and personalized support.

Virtuozzo Challenges

- **Proprietary software and custom scripting (JPS)**
  Another DSL to learn; less portable skills
- **Complicated pricing**
  Non-standard unit of measuring cloudlet. Harder to forecast peaks and total cost
- **VMs as a byproduct component**
  Focus on containers/stacks; legacy workloads need more glue.

Virtuozzo Application Platform (Jelastic): a PaaS with “cloudlet” billing (a unit of CPU/RAM), a strong JCA admin panel, a large marketplace, and Cloud Scripting/JPS (a proprietary DSL on top of JSON/YAML).

AREA	COZYSTACK	VIRTUOZZO (JELASTIC)
Truly managed Kubernetes	Get fully functional Kubernetes clusters, designed the same way just like in AWS or Google cloud	---
HA and Full lifecycle management	Truly managed services, with HA and monitoring. Managed by specialized kubernetes operators.	Ready templates that do not guarantee that service will be automatically recovered after failure.
No vendor lock-in	Open-source, CNCF ecosystem	Proprietary solution
Pricing model	Per physical node; budget is knowable upfront	Non-standard unit of measuring cloudlet. Harder to forecast peaks and total cost
Billing integrations	Open approach: flexible billing API, you can customize on frontend	At some providers, billing is “inside VAP,” with WHMCS mainly invoicing what the platform computed
Minimum install	3 nodes cover control+storage+workload (replicated)	For some customers, high baseline just for the platform itself (historically: separate nodes for VAP + workload + storage)
VM	KubeVirt (VMs as first-class alongside K8s), networking/BGP	Focus on containers/stacks; VMs are not the core
Easy to extend	Add your own software and managed services quickly, Ænix team ready to help	Hardcoded apps, not easy to extend
Support	Get new features quickly, dedicated support channel with engineers, personalized roadmap	---
Backups/DR	PITR for DB (CNPG→S3), Velero/snapshots, DR runbooks	Manual DB backups scripts; scenarios vary by provider/installation
Automation/IaC	Helm/Terraform/API (de-facto standards)	JPS / Cloud Scripting (JSON/YAML + proprietary concepts)
Marketplace	Helm charts, OSS repos	Marketplace with stacks/DB/add-ons
Wite labeling	✓	✗
Docker services	Containers via K8s/presets; quick wizards	Docker as a Service (single Docker host as a first-class entity)
Ultimate bare-metal performance	Managed services are running on pure bare-metal nodes, giving you ultimate bare-metal performance	Services are limited to run on VMs that adds additional overhead