Category	Feature	XLOUD	VMware	Nutanix	HPE VME	Openshift	KVM
Infrastructure	**Compute/Hypervisor Services**: Enterprise VM lifecycle (create, scale, live-migrate, snapshot) with scheduler policies for affinity/anti-affinity and host maintenance workflows.	✔ KVM/Nova with live migration & policies	✔ ESXi/vCenter with DRS & vMotion	✔ AHV + Prism with live migration	✔ KVM with HA & live migration	⚠ VM lifecycle supported; live migration exists but with constraints; scheduling is limited vs full hypervisors	⚠ Core VM lifecycle supported; no native HA or policy scheduler without external stack (e.g., Xloud OpenStack)
	Bare-Metal Provisioning: Provision and lifecycle-manage physical servers—exposed via the same cloud APIs as VMs.	✔ Native Bare-Metal Provisioning via Ironic	⚠ Possible via external PXE/IPMI tools	✘ Not natively supported	⚠ External PXE/IPMI integrations	⚠ Bare Metal Operator / Metal³ enables node provisioning only	✘ No native bare-metal provisioning
	GPU/Accelerator Virtualization: Attach GPUs/FPGAs to workloads via passthrough, SR-IOV or mediated devices; manage pools and profiles for AI/NFV/HPC.	✔ Passthrough/SR-IOV + Cyborg lifecycle	✔ Mature vGPU (GRID/MIG) + Bitfusion	✔ GPU passthrough and NVIDIA vGPU	⚠ Passthrough; vGPU varies by build	⚠ GPU passthrough supported; vGPU/mdev depends on vendor plugins	✔ Full PCIe passthrough; ⚠ mdev/vGPU varies by vendor support
	Memory/CPU Optimization: NUMA awareness, CPU pinning, emulator pinning, and HugePages to minimize jitter and boost throughput for NFV/HPC/databases.	✔ Full NFV-grade controls	✔ Supported with granular tuning	✔ Supported in AHV	✔ Supported on KVM	⚠ CPU pinning, NUMA awareness and hugepages via CPU/Topology Manager + KubeVirt; tuning model is Kubernetes-native rather than a dedicated VM DRS engine.	✔ NUMA, pinning, hugepages, IO/emulator thread policies supported
	High-Throughput vNICs: Multiqueue virtio, SR-IOV vNICs, and DPDK datapaths to maximize packet rates and reduce latency for data-plane workloads.	✔ Multiqueue + SR-IOV + OVS-DPDK	✔ Multiqueue + NSX datapath	⚠ Virtio multiqueue widely supported; SR-IOV and HW offload available only in newer AHV releases and specific NIC SKUs (no general OVS-DPDK datapath).	⚠ Basic multiqueue; DPDK limited	⚠ SR-IOV operator available; DPDK supported only in specific deployments	✔ SR-IOV & multiqueue supported; ⚠ DPDK requires manual configuration
	Tenant Networking (VPC semantics): Create virtual networks with VXLAN/GENEVE overlays, routed subnets, security groups, and NAT/floating IPs per project.	✔ Neutron/OVN VPC-like model	✔ NSX-T overlays + distributed routing	✔ AHV's Flow VN supports overlay VPCs (Geneve)	⚠ Overlays are integration-driven in VME, not native.	⚠ Networking optimized for pods, not true multi-tenant VM VPCs	✘ No built-in tenant networking (requires integrations)
	Persistent Storage Services: Block (for VMs/databases), File (shared POSIX/SMB), and Object (S3-compatible) with snapshots, replication, and erasure coding.	✔ Cinder/Manila/Swift + Ceph EC/replica	⚠ vSAN for block + vSAN File Services (NFS/SMB); S3-style object storage delivered via ecosystem add-ons, not core vSAN.	✔ AOS (block/file) + Objects	✘ No native Block/File/Object; attaches NFS/iSCSI/FC (array-side snapshots)	⚠ Block/File via CSI backends; object requires external services	✘ No native SDS (depends entirely on external storage)
	Replication & Disaster Recovery: Cross-cluster replication and orchestrated failover with runbooks for RTO/RPO objectives.	✔ XDRS	✔ SRM + HCX (mature DR)	✔ Async + Metro (sync) replication; Leap DR runbooks & recovery plans	⚠ Snapshots/backup; DR via external integrations	⚠VM backup/restore and DR patterns achievable via OpenShift Data Foundation and ecosystem tools; no native per-VM replication/failover engine.	✘ No native DR (requires external tooling)
	Kubernetes Enablement: Provision clusters as a service, manage lifecycle (scale/upgrade), and integrate CNI/CSI with cloud networking and storage.	✔ Magnum + external (Rancher/OCP)	✔ Tanzu Kubernetes Grid	✔ NKE -full lifecycle provisioning/upgrade/scale	⚠ K8s via Morpheus (blueprints/cluster provisioning); lifecycle depth depends on chosen distro/integration.	✔ Core competency; native Kubernetes platform	✘ No Kubernetes enablement
	Multi-Tenancy & Federation: Hard isolation via projects/quotas with cross-region identity federation and policy propagation.	✔ Keystone projects/quotas + federation	⚠ Orgs/projects; RBAC/SSO; limited cross-region federation and policy propagation	⚠ Robust RBAC/projects, SSO; federation across regions is limited vs Keystone.	✔ Tenants/orgs/projects; RBAC/quotas; IdP-based cross-region federation; policy propagation across onboarded clouds	⚠ Namespaces provide soft multitenancy; limited VM tenant isolation	✘ No multitenancy
Security & Ecosystem	**SIEM/XDR:** Ability to ship host/VM logs and security telemetry into a SIEM/XDR, enforce agent policies, and derive alerts/compliance findings.	✔ Agents + syslog/Fluent Bit to SIEM; audit/API logs; dashboards supported	✔ vCenter/NSX remote syslog → SIEM; guest agents supported.	✔ Syslog forwarding (audit/Flow) → SIEM; agent integrations OK.	⚠ Logs export + Activity/Audit; agents by policy; SIEM via API/automation, not a native "pipeline"	⚠ Logs + audit streams to SIEM; limited VM telemetry	✘ No SIEM capability
	VM Image Signing & Verification (at boot): Only run VMs from cryptographically signed templates.	✔ Signed VM images verified at boot; key-managed	⚠ Secure Boot and signed ESXi/VIBs supported; no strict "only run cryptographically signed VM templates" enforcement gate.	⚠ Templates without native boot-time signature verification	✔ Depends on KVM/automation; no native enforcement	⚠ Strong container image signing/admission; no first-class 'signed VM template' mechanism for boot-time verification of VM images.	✘ No native support
	Live Patching : Apply live kernel/libc/OpenSSL patches to hosts/VMs to reduce reboots and shrink exposure windows.	✔ Host + guest live patching (kernel/libc/OpenSSL) on KVM/QEMU	⚠ Guest live patching; ESXi hosts via LCM/maintenance (no true host live patch)	⚠ Guest live patching; AHV hosts via Nutanix LCM (maintenance window)	⚠ Host + guest live patching possible on KVM; not VME-native	⚠ Kernel hotpatching via RHEL but not for VM guests	⚠ Hypervisor live patching depends on host distribution
	Vulnerability & Compliance (OS baseline + image hardening): Automated host/guest scans, policy gates, and exportable reports.	✔ Vulnerability/compliance pipeline for hosts & images; policy gating supported	⚠ Compliance baselines + 3rd-party VA integration; no native SCAP pipeline	⚠ Security/compliance dashboards + 3rd-party VA; no SCAP pipeline	⚠ Orchestrates VA via agents/tasks; no native VA engine/SCAP	⚠ Compliance Operator for cluster nodes; VM guest scanning requires external tooling	✘ No native compliance scanning
	Workload Migration/DR: Replicate VMs across platforms/clouds with runbooks and test failovers; support heterogeneous targets.	✔ Heterogeneous DR/migration (VMware/AWS/Azure/GCP); runbooks & test-failover	✔ HCX/SRM for VMware↔VMware DR/migrations; native-cloud targets require conversion/orchestrators	✔ Leap/Move for into-AHV DR/migrations; heterogeneous targets primarily inbound to Nutanix	✔ Native VMware→HVM bulk migration; DR runbooks/plans across onboarded clouds; no built-in replication engine	⚠ Cross-cluster DR/migration achievable for containers via GitOps; VMs require external DR tooling—no built-in heterogeneous VM migration engine.	✘ No native migration/DR tooling
	Secrets & Key Management: Native key/secret storage with secure retrieval by services and APIs; KMS/HSM integrations for envelope encryption.	✔ Barbican + KMS/HSM integrations	⚠ Native Key Provider + external KMS (KMIP) for VM/disk keys; not an app secrets vault.	⚠ Built-in KMS (optional) + external KMS; no general app secrets vault.	✔ Morpheus Cypher (secrets) + external Vault/KMS	✔ Strong native secret/KMS framework (Vault/KMS/Cloud-KMS integrations) — VM secret mgmt via Kubernetes	✘ No secret/key mgmt capability

	Tenant-Scoped Volume/Image Encryption (BYOK & per-object keys): Encrypt each volume/image with project-owned keys; rotate/revoke without re-image.	✔ Per-volume/image encryption tied to tenant keys (BYOK/HSM) with policy gating	⚠ Cluster/VM-level encryption via external KMS; not tenant-scoped per-object	⚠ Cluster-level at-rest encryption via KMS; no per-tenant per-object keys	✘ No native per-object, tenant-scoped keys	⚠ Supported via CSI encryption if backend supports it; not native VM-image encryption	✘ No encryption mechanism for VM disks/images
Software Defined Networking	**CPU/Emulator Pinning & NUMA**: Bind vCPUs and emulator threads to specific physical cores and NUMA nodes to minimize jitter and improve throughput for NFV/HPC workloads.	✔ Fine-grained pinning + NUMA aware	✔ Advanced CPU/NUMA controls	✔ NUMA aware + pinning	✔ NUMA aware + pinning	⚠ Exposed via KubeVirt CPU pinning; not as granular as full hypervisors	✔ Full support (pinning, emulator threads, NUMA topology)
	HugePages (1G/2M): Back guest memory with large pages to reduce TLB misses and raise memory bandwidth efficiency.	✔ 1G/2M HugePages supported	✔ Large pages supported	✔ HugePages supported	✔ HugePages supported	✔ KubeVirt supports hugepages (1G/2M)	✔ Native hugepages support
	High-Performance Networking (vNIC Multiqueue): Enable multiple transmit/receive queues per vNIC to scale packet processing across CPU cores.	✔ Virtio multiqueue supported	✔ vNIC multiqueue supported	⚠ Virtio multiqueue widely supported; SR-IOV and HW offload available only in newer AHV releases and specific NIC SKUs (no general OVS-DPDK datapath).	⚠ Basic multiqueue support	⚠ Supported via Multus/virt-launcher tuning, but not as robust as AHV/ESXi	✔ Multiqueue supported in virtio-net
	SR-IOV for NIC/Storage: Expose virtual functions directly to VMs for near line-rate I/O while maintaining tenancy boundaries.	✔ SR-IOV widely supported	✔ SR-IOV widely supported	⚠ Partial SR-IOV support	⚠ Basic SR-IOV support	✔ SR-IOV supported via SR-IOV Network Operator	✔ SR-IOV passthrough supported
	DPDK Datapaths: Bypass kernel networking stack for user-space fast paths (e.g., OVS-DPDK) to maximize packets-per-second for data planes.	✔ OVS-DPDK and userspace vSwitch	✔ NSX datapath optimizations	⚠ Limited; OEM-dependent	⚠ Limited; integration-dependent	⚠ Supported via Performance Add-on + SR-IOV/DPDK pods; VM DPDK limited	⚠ Supported only through manual host-level DPDK configuration
	Overlay Networking (VXLAN/GENEVE): Encapsulate tenant traffic in overlays to deliver VPC-like isolation, multi-tenant routing, and mobility.	✔ VXLAN/GENEVE via OVN/OVS	✔ VXLAN/GENEVE via NSX-T	✔ VLAN-first, but overlay support (Geneve) via Nutanix Flow	⚠ VLAN-first; overlays via SDN add-ons	✔ OVN-Kubernetes uses GENEVE overlays	✘ No overlay networking unless paired with SDN (e.g., Open vSwitch + external controller)
	BGP EVPN & Routed Leaf-Spine: Integrate with data-center fabrics for scalable multi-tenant routing and optimal east-west paths.	✔ BGP EVPN agents in OVN	✔ EVPN with NSX-T	✘ AHV/Prism rely on external leaf-spine fabrics (Cisco/Arista/Mellanox) for BGP EVPN; no Nutanix-native EVPN control-plane.	⚠ Depends on external SDN	⚠ Possible via OVN-K + external integrations; not native for VMs	✘ No routing/EVPN capability
	NAT/SNAT/DNAT & Floating IPs: Provide north-south connectivity with NAT policies and public address mapping per tenant/project.	✔ Native SNAT/DNAT + Floating IPs	✔ NAT policies; routed edges	⚠ NAT/SNAT via AHV virtual networking and Flow; no CSP-style elastic floating-IP abstraction across sites/regions.	⚠ NAT via HAProxy/NGINX/edges	⚠ Kubernetes Services/NAT available; Floating-IP style semantics for VMs not natively supported	✘ No built-in NAT/floating-IP functionality
	L7 Policies (SSL, WAF, Health): Terminate TLS, enforce HTTP routing rules, perform health checks, and integrate WAF for application resilience.	✔ Octavia L7 policies - TLS termination, HTTP L7 routing (host/path/header/cookie), ordered policies, health checks.	✔ TLS offload, advanced HTTP policy sets, health checks, analytics-driven traffic steering.	⚠ Basic LB; limited L7 - primarily L4 with selective HTTP rules and health checks,deeper WAF/advanced L7 typically via partner VNFs.	⚠ Basic LB; TLS termination and simple HTTP rules/health checks; deeper features are integration-dependent.	⚠ Ingress/Route layer supports TLS termination & health checks; WAF is external only	✘ No L7 network features
CMP	**Billing & Metering:** Ability to rate usage for compute (vCPU-hour), memory (GB-hour), storage (GB-month), IOPS, egress, and auxiliaries like load balancers; supports chargeback/showback and cost allocation models.	✔ Native rating & chargeback; granular meters	⚠ vSphere/Aria/Cloud Director provide strong usage/showback and chargeback; commercial billing and payment gateways typically handled by external BSS/billing platforms.	⚠ Core Prism provides capacity/usage reports; richer cost/showback/chargeback via optional NCM Cost Governance. Billing and payment gateways typically implemented via external billing/BSS systems.	✔ Chargeback/costing native; gateways via integrations	✘ No native billing/metering engine for VMs	✘ No billing capability
	Payment Gateways & Invoicing: Native integration to payment processors (e.g., Stripe/Razorpay/PayU) to generate invoices, collect taxes, and reconcile payments for monetized clouds or MSPs.	✔ Direct PG integrations + invoicing	✘ No native PG; rely on 3rd-party	✘ Not provided natively	✔ Via Morpheus integrations/plug-ins	✘ Not available	✘ Not available
	Image & Application Catalog: Curated golden images and multi-tier app blueprints (Terraform/Ansible) with policy-backed self-service deployments and versioned releases.	✔ Images + blueprints + policy guardrails	✔ Catalog + blueprints (Aria/vRA)	✔ Calm blueprints; app stacks	✔ Rich multi-cloud catalog (Morpheus)	⚠ VM templates and OperatorHub provide a basic catalog; not a full IaaS-style VM marketplace with multi-tenant metering and governance.	✘ No catalog capability
	White-Label & Multi-Brand: Ability to rebrand the portal (logo, theme, domain) and segment experiences for multiple brands, partners, or tenants.	✔ Full white-label & multi-brand	⚠ Limited theming	⚠ Basic customization	✔ White-label supported	✘ No white-label functionality	✘ Not applicable
	Identity, MFA & RBAC: Enterprise SSO (OIDC/SAML/LDAP/SCIM), MFA enforcement, and hierarchical RBAC (org → project → role) with delegated administration.	✔ OIDC/SAML/SCIM/MFA; deep RBAC	✔ SSO + MFA + roles	✔ SAML + RBAC	✔ SSO/MFA + org/project RBAC	✔ Strong SSO/MFA/RBAC support via OAuth, LDAP, SAML, OIDC	✘ No identity or RBAC layer
	Governance & Approvals: Guardrails, quota policies, and workflow approvals that prevent misconfiguration and ensure compliant provisioning across teams.	✔ Policy-as-code + approvals + quotas	✔ Policy + approvals (Aria/vRA)	✔ Policies + approvals + quotas & guardrails	✔ Policies + approvals + quotas	⚠ Quotas exist; no approval workflows	✘ Not supported
	Observability, Audit & FinOps: Cost analytics, usage dashboards, API/audit logs, and exportable evidence packs for compliance and chargeback reconciliation.	✔ Cost & audit analytics; exports	✔ Aria Ops/Audit strong	⚠ Core metrics; limited FinOps	✔ Cost analytics + audit trails	⚠ Observability/logs/audit available; cost/FinOps not available	✘ No observability stack (only host-level logs)

Automation	**Policy-as-Code Provisioning:** Encode quotas, network/security policy, and placement into flavors/blueprints for deterministic outcomes.	✔ Flavor/image extra_specs + Keystone/Neutron/Nova policies; deterministic, tenant-aware.	✔ Aria policies work, but some constraints live outside blueprints.	⚠ Calm policies are simpler; fewer low-level controls.	⚠ Morpheus policies strong, but enforcement depends on underlying KVM/SDN wiring.	✔ Supported via GitOps & Kubernetes-native policy frameworks	✖ No policy engine
	Automation Evidence & Change Journal: Exportable artifacts to prove "who/what/when" for audits.	✔ API/audit logs + playbook logs + metrics exportable; easy to script evidence packs.	✔ Aria Ops/Logs produce reports; productized but license-coupled.	✔ Prism exports are solid; fewer levers for deep pipeline evidence.	⚠ Morpheus audit + external ELK/Prom; assembly required.	⚠ Kubernetes/API and GitOps (ArgoCD) provide audit trails and change history; no single consolidated 'cloud automation journal' across all layers.	✖ No change tracking
	Lease/TTL with Auto-Teardown: Time-boxed environments that **auto shut down & clean up** on expiry (no manual ticket).	✔ Blazar leases; enforce start/expiry → auto delete or park.	✔ Achievable via Aria lease policies; not native to vCenter itself.	⚠ No first-class project leases; requires custom runbooks to emulate.	✔ Morpheus supports expiring apps/instances with scheduled teardown.	⚠ TTL controllers and automation can implement time-boxed environments; no native VM/project 'lease' primitive with policy-driven teardown.	✖ Not supported
	Metric-based VM Autoscaling: Automatically add/remove VMs when CPU/queue/lag thresholds are crossed (no human in loop)	✔ Heat AutoScalingGroup + alarms (Prometheus webhooks) to scale a stack.	✔ Via Aria policies/workflows	⚠ Calm runbooks support basic triggers.	⚠ Requires wiring monitors and defining runbooks.	✖ Not available for VMs (only for pods)	✖ Not supported
Monitoring	**Unified Cloud Health (control-plane & services):** End-to-end heartbeat for APIs, schedulers, agents, and data paths.	✔ XAVS Health dashboards + Prometheus exporters validate Nova/Neutron/Cinder/OVN and data plane liveness.	✔ vCenter health + Aria Operations status views for ESXi/NSX/edges.	✔ Prism "Health" for cluster/VM/storage networking.	⚠ Morpheus monitors for KVM/guests; depth depends on integrated collectors.	⚠ Control-plane health built-in; VM health limited	✖ No health management framework
	Tenant & Project Utilization Views: Real-time CPU/RAM/IO/network per project with quota burn-down.	✔ Per-project dashboards	✔ Multi-tenant dashboards, budgeting, and quota analytics across vSphere/NSX estates.	⚠ Project dashboards + quota utilization exist; time-series 'burn-down' and budgeting usually come from NCM Cost Governance.	✔ Provides tenant/project utilization and cost-style reports with exportable views.	⚠ Per-project metrics available via console/Prometheus; no turnkey 'VM cloud tenant utilization' dashboard comparable to IaaS CMPs.	✖ No multi-tenant utilization capability
	Log Analytics & Event Correlation: Centralized logs with search, context, and drill-downs to failing components.	✔ ELK stacks with XAVS parsers for Nova/Neutron/OVN; link alerts → logs.	✔ Aria Operations for Logs (vRealize Log Insight) for ESXi/NSX/vCenter events.	⚠ Prism eventing + basic logs; deep correlation typically via external ELK.	⚠ Needs integration for full-text and dashboards.	✔ Cluster logging stack available; VM logs limited	✖ No log analytics beyond syslog
	L2/L3 Auto-Discovery & Topology Map: Discover switches/routers via SNMP + LLDP/CDP, auto-draw links, and correlate VLAN/VNI to tenant networks.	✔ Full fabric discovery and live link maps; easy pivot from tenant (VLAN/VNI) to physical ports/paths.	✔ Strong logical (NSX) topology; physical L2/L3 discovery is limited natively—often needs a separate NPM tool for full fabric view.	⚠ VM/host connectivity views exist; end-to-end switch/router discovery and link mapping typically require an external NPM.	✖ No native physical discovery/topology; depends on an external network/fabric monitoring stack.	✖ Not available	✖ Not available
Orchestration	**Multi-tier Blueprints & Dependency Ordering:** Define web/app/db stacks with dependencies, waits, and outputs for repeatable deploys.	✔ Template-driven stacks with explicit resource ordering and outputs.	✔ Strong blueprints and dependency graphs.	✔ Calm blueprints cover common multi-tier patterns.	⚠ Blueprints supported; complex dependencies often need extra runbook logic.	⚠ Multi-tier apps supported via Helm/Operators for containers; multi-VM blueprints and dependency ordering require external tooling (Terraform/Ansible, etc.).	✖ Not available
	Change Preview ("Plan") & Pre-checks: Show what will change before execution; block on policy/quotas/image rules.	✔ Plan/diff previews plus admission-style checks are doable.	⚠ Validation exists; granular infra diffs are less transparent than IaC plans.	⚠ Basic validation; limited change-diff visibility.	⚠ Preview via workflow outputs; deeper pre-checks require custom validators.	⚠ Manifest-level change preview via GitOps (ArgoCD) and diff tooling; no dedicated 'VM change plan' layer with quota/image rule simulation.	✖ Not supported
	Cross-Environment Promotion: Promote the same blueprint across dev→stage→prod using parameter sets and scoped secrets.	✔ Environment vars/secret injection make promotions deterministic.	✔ Environment profiles and approvals enable controlled promotion.	✔ Projects/Calm variables support environment moves.	⚠ Environments supported; strict secret/parameter segregation depends on your policy wiring	⚠ Strong cross-environment promotion for Kubernetes resources (including KubeVirt VMs) via GitOps; no separate VM release pipeline abstraction.	✖ No concept of environment promotion
	Drift Detect & Converge: Detect config drift and re-apply the desired state automatically.	✔ Supports drift checks and re-converges stack state to the declared blueprint.	⚠ Detects drift; auto-converge typically requires added workflows/policies.	⚠ Flags deviations; full auto-converge is limited without custom runbooks.	⚠ Drift alerts achievable; enforcement back to desired state is integration-driven.	✔ Supported at Kubernetes level (GitOps/ArgoCD), ⚠ limited for VMs	✖ No drift detection
DRS (Optimization)	Policy-Driven Rebalance (affinity/anti-affinity, license & AZ constraints): Continuously place/move VMs to respect rules while smoothing host utilization.	✔ Constraint-aware rebalancing; honors AZ/host aggregates, affinity sets, quotas.	✔ Mature rule handling; strong anti-affinity/host rules.	✔ Affinity/anti-affinity supported; fewer niche constraints.	⚠ Basic rules possible; depth depends on how policies are modeled.	✖ No proactive VM rebalance engine	✖ Not available
	Maintenance-Aware Evacuation: Drain hosts for patching with SLA gates (noisy-neighbor checks, health probes) and staged re-entry.	✔ Evacuation plans with health gates and staged admission.	✔ Maintenance mode + automated evacuations; policy-driven.	✔ Guided maintenance flows; simpler gating.	⚠ Procedure-backed; relies on your playbooks.	⚠ Live migration works; no intelligent SLO-aware evacuation	✖ No maintenance evacuation
	Action-Plan Preview (dry-run before execute): Show proposed moves and impact before committing.	✔ Generates reviewable action plans/dry-runs.	✔ Change simulations/views available.	⚠ Basic previews; less granular diffs.	⚠ Preview via workflow outputs; depth varies.	✖ Not available	✖ Not available
	Custom Optimization Goals (extensible strategies): Add your own objectives beyond built-ins.	✔ Supports **custom goal creation** and plug-in strategies (requires engineering effort).	⚠ Rich presets; true custom logic needs external tooling.	✖ No native plug-in goal model.	⚠ Achievable via scripts/runbooks; no native goal plug-ins.	✖ Not supported	✖ Not supported
	Rolling Control-Plane Upgrades: Update services without broad outages; pause/resume and batch safely.	✔ Rolling upgrades for Nova/Neutron/Cinder/etc. with playbooked steps and checkpoints	✔ vLCM/NSX-LCM automate rolling updates.	✔ One-click LCM for AOS/PC; service restarts orchestrated.	⚠ Manager/appliance upgrades via online/offline packages; no rolling multi-node control-plane.	✔ Strong native rolling updates for control-plane components	✖ No control-plane concept

Lifecycle Management	**Kernel & Userspace Live-Patching (no reboot)**: Apply security fixes to kernel/glibc/openssl without downtime.	✔ Supported on XOS (kernel + key userspace like glibc/OpenSSL).	⚠ New vSphere Live Patch reduces or avoids host evacuation for specific ESXi components (e.g., VMX) but is not a general Linux-style kernel live-patch of ESXi; many updates still need maintenance windows.	✘ Hypervisor updates require reboot; no native live-patch.	✘ No native live-patch; would require OS-level integration outside the product.	✔ Supported on host OS (RHEL) with kpatch/livepatch	⚠ Depends on host distribution's live patching (not native to KVM)
	Air-Gapped Updates (offline content): Run LCM with private registries/repos—no internet.	✔ Pull images/packages from private registries/mirrors.	✔ Offline bundles/catalogs supported.	✔ Dark-site bundles and direct upload flows.	⚠ Feasible with mirrored repos/registries; operator-run workflows.	✔ Fully supported (disconnected installation + mirrored registry)	✔ Host-level offline updates possible; VM platform has no LCM
	Config as Code & Idempotent Reapply: Reconcile drift by re-running the desired state.	✔ Declarative configs; re-apply converges to desired state.	✔ vSphere Configuration Profiles = desired state + compliance + remediate drift.	⚠ Prism Central Runbooks/Calm/LCM standardize changes (procedural); no declarative host/cluster desired-state with automatic drift fix.	✘ No native declarative host/cluster desired-state controller or continuous drift remediation.	✔ Fully supported via GitOps and Kubernetes native declarative model	✘ Not available
Software Defined Storage	**Erasure Coding & Replication per workload (policy/pool level):** Choose EC or 3x/2x replication per class of service.	✔XSDS pools support EC or replication; XAVS can pair EC data pools with replicated metadata.	✔ vSAN SPBM policies (RAID-5/6 EC or RAID-1 mirror) per-VM/VMDK.	✔ EC-X within DSF; policy-driven.	✘ No native SDS; relies on attached storage (capability varies by backend).	✘ No native SDS; relies on attached storage (capability varies by backend).	✘ No native SDS; relies on attached storage (capability varies by backend).
	Failure-Domain–Aware Placement (host/rack/site): Enforce durable placement across failure domains.	✔ CRUSH rules model host/rack/DC; pool-specific rulesets.	✔ vSAN Fault Domains + Stretched Clusters for rack/site resilience.	✔ Rack/Block availability domains; rack-aware placement.			
	Unified Block/File/Object from one SDS fabric.	✔ RBD (block), CephFS (POSIX), RGW (S3/Swift) in one cluster.	⚠ vSAN datastore + vSAN File Services (NFS/SMB); no native S3 object.	✔ Volumes (iSCSI), Files (NFS/SMB), Objects (S3) natively.			
	Instant Snapshots & Clones (COW) for rapid provisioning.	✔ RBD snapshots + COW clones; boots via Glance/Cinder.	✔ vSphere/vSAN snapshots & fast clones via SPBM.	✔ Native snapshots/clones across VMs/Volumes.			
	Encryption at Rest (software) with optional external KMS.	✔ OSD-level dm-crypt/LUKS; cluster-wide at-rest encryption.	✔ vSAN Data-at-Rest Encryption (cluster toggle).	✔ DaRE (software) and SED options; KMS/KMIP.			