Portworx Enterprise: Kubernetes Native Data Services for Edge Computing

Introduction

Organizations across industries — from retail and manufacturing to telecommunications and the public sector — are extending Kubernetes infrastructure to the edge. According to Gartner, 50% of enterprises will initiate proofs of concept by 2026 to phase out existing VMware-based deployments in favor of distributed hybrid infrastructure, with edge playing a central role. Spectro Cloud predicts that by the end of 2026, 75% of enterprises will be using Kubernetes at the edge.

Edge deployments bring compute closer to data sources, users, and operational systems, enabling lower latency, local autonomy, and real-time decision-making. But running stateful workloads at the edge introduces a fundamental infrastructure gap: the absence of a Kubernetes-native persistent storage platform purpose-built for resource-constrained, distributed environments.

Without a dedicated data services layer, every stateful service at the edge — databases, message queues, AI/ML model stores, configuration registries, and operational logs — relies on fragile, ad hoc storage that cannot survive node failures, cluster disruptions, or intermittent connectivity with central data centers.

This paper examines the persistent storage challenges unique to edge Kubernetes deployments and shows how Portworx Enterprise fills that gap with Kubernetes-native data services for local drive management, failure resilience, data snapshots, and data migration — all managed through the Kubernetes control plane.

1. The Edge Computing Challenge

1.1 Why Kubernetes at the Edge

Kubernetes has become the standard orchestration layer for modern applications. As organizations push workloads beyond the data center — to retail stores, manufacturing floors, remote facilities, telecommunications sites, and field operations — Kubernetes provides a consistent operational model from core to edge.

However, Kubernetes was originally designed for stateless microservices. Edge workloads are increasingly stateful: databases powering local applications, message queues coordinating distributed systems, AI model inference engines, telemetry and log aggregation, and operational data stores all require persistent volumes that survive pod restarts, node failures, and cluster disruptions.

1.2 Unique Constraints at the Edge

Edge environments — including Retail/Remote Office Branch Office (ROBO), Far Edge, and Near Edge — present challenges fundamentally different from data center or cloud deployments:

Challenge	Impact
Limited physical footprint	Often restricted to a server closet, telco cabinet, or small enclosure with constrained power, cooling, and space

Challenge	Impact
No on-site IT staff	Storage and infrastructure must be self-managing with minimal manual intervention
Intermittent connectivity	Connections to central data centers may be limited, degraded, or periodically unavailable; edge sites must operate autonomously
Cost sensitivity	Edge sites are typically high in count and must remain cost-effective per location
Hardware constraints	Commodity, off-the-shelf servers rather than enterprise-grade storage arrays
Outsized failure impact	In small clusters, a single node failure has a disproportionate effect on data resilience and application availability

1.3 The Persistent Storage Gap

Without a Kubernetes-native storage platform at the edge, organizations encounter several critical issues:

- **Fragile data persistence** — Local volumes tied to individual nodes with no replication or failover
- **Manual storage administration** — Capacity monitoring and management requiring on-site intervention
- **No standardized data protection** — Inconsistent backup, snapshot, and recovery capabilities across sites
- **Siloed data** — No automated mechanism to synchronize or migrate data between edge and central locations
- **External storage dependencies** — Reliance on SAN arrays or cloud-connected storage that cannot function during WAN outages
- **Multi-vendor complexity** — Each application team or vendor implementing its own storage solution, creating incompatible data silos that increase integration effort and fragment capacity

2. Portworx Enterprise: Kubernetes-Native Data Services

Portworx Enterprise is the industry's leading Kubernetes-native data services platform, purpose-built for persistent storage, data protection, and data mobility in containerized and virtualized environments. Portworx runs as a fully software-defined storage layer inside the Kubernetes cluster, aggregating whatever local storage is available into an application-aware data platform — eliminating external storage dependencies and enabling fully autonomous edge operations, even during WAN outages.

2.1 Architecture Overview

Portworx deploys as a DaemonSet on every Kubernetes worker node, aggregating local NVMe, SSD, and HDD drives into a distributed storage pool. Applications consume storage through standard Kubernetes primitives:

- **StorageClasses** define differentiated storage tiers with declarative policies for replication, encryption, I/O profiles, and quality of service
- **PersistentVolumeClaims (PVCs)** provide self-service storage provisioning for developers and operations teams
- **CSI driver integration** ensures full compatibility with the Kubernetes storage API

This architecture means Portworx operates entirely within the Kubernetes control plane. Every storage operation — provisioning, replication, failover, snapshot, migration, encryption — is expressed as a Kubernetes-native declaration and managed through familiar kubectl workflows, CI/CD pipelines, and GitOps tooling.

2.2 Edge Storage Requirements Mapping

The following table illustrates how Portworx maps to common edge storage requirements:

Edge Storage Requirement	Portworx Capability	Operational Impact
Real-time local storage	PX-StoreV2 NVMe-optimized volumes	Sub-millisecond latency for latency-sensitive applications
Tiered storage lifecycle	Sync back to central systems running on FlashArray/FlashBlade	Data moves automatically from hot edge to enterprise data center storage
Common storage substrate	CSI StorageClasses shared across all workloads	Single storage fabric for all applications — no per-vendor silos
AI/ML platform storage	High-throughput shared volumes	Train and serve models on the same storage as operational data
Data protection	Application-consistent snapshots, encryption, immutability	Automated, policy-driven data protection without manual intervention

2.3 Core Capabilities for Edge Deployments

PX-StoreV2, the storage engine within Portworx Enterprise, provides high-performance software-defined block storage optimized for NVMe drives common in modern edge servers:

- **Automatic drive aggregation** — Portworx discovers and pools all local block devices (NVMe, SSD, HDD) on each node into a unified, shared storage cluster, eliminating manual disk provisioning
- **Heterogeneous disk support** — Unlike solutions that require identical drive types and sizes, Portworx scales using any mixture of disk types and capacities
- **StorageClass-based provisioning** — Platform teams define storage "golden paths" for different workload profiles (e.g., high-IOPS

for databases, throughput-optimized for analytics, replicated volumes for critical state) — developers simply reference the StorageClass in their PVC

- **Thin provisioning** — Efficient capacity allocation that avoids overcommitting physical storage
- **Efficient resource utilization** — Container-native architecture with flexible storage pooling across limited local disks, right-sized for small-cluster deployments

At the edge, there is no SAN and typically no storage administrator. Portworx builds resilience directly into the Kubernetes data layer:

- **Synchronous replication (RF2/RF3)** — Every write commits to multiple nodes before acknowledgment, ensuring zero data loss (Zero RPO) in the event of hardware failure. RF2 (two replicas) is typical for ROBO and Far Edge clusters; RF3 (three replicas) is available for Near Edge sites requiring higher durability
- **Automatic failover with self-healing** — When a node fails, Portworx detects the failure and makes the replicated data immediately available on surviving nodes. Kubernetes reschedules the pod, and it attaches to the existing replica with no data loss or manual intervention. Fast rebuilds and self-healing capabilities restore full redundancy automatically once a node returns or a replacement is provisioned
- **Live migration for planned maintenance** — When a node must be taken down for hardware servicing, firmware updates, or application patches, workloads live-migrate to surviving nodes and migrate back upon completion — with zero downtime
- **Quorum and split-brain protection** — For minimal-footprint edge clusters, a lightweight arbiter node (which can run on extremely low-power hardware) maintains quorum and prevents data inconsistency during node failures, without requiring a full additional storage node
- **Application I/O Control** — Per-volume I/O throttling and quality-of-service policies prevent noisy-neighbor effects when multiple workloads share limited edge infrastructure

Portworx provides Kubernetes-native snapshot and backup capabilities essential for edge data protection:

- **Application-consistent snapshots** — Point-in-time snapshots that capture the consistent state of a running application, not just raw block data. Snapshots are created through the standard Kubernetes VolumeSnapshot API
- **Scheduled snapshot policies** — Declarative policies (defined in YAML) automate periodic snapshots at configurable intervals, with retention rules that manage snapshot lifecycle without manual intervention
- **Local backups with central archival** — Container-granular backups stored locally at the edge for rapid recovery, with periodic backup to central object storage for long-term retention and compliance
- **Encryption at rest and in transit** — AES-256 encryption for data at rest and TLS 1.3 for data in transit, with cluster-wide or per-volume encryption policies defined through StorageClass parameters
- **Ransomware protection** — Automated backup policies support immutability and portability across clouds, enabling rapid response to ransomware and cyber attacks with point-in-time recovery

Moving data between edge sites and central locations — for centralized analytics, AI model training, compliance archiving, or disaster recovery — is a critical edge requirement. Portworx provides built-in replication and migration capabilities:

- **Asynchronous replication** — Configurable RPO replication for edge-to-datacenter connectivity. Portworx transmits only changed data blocks (differential block-level synchronization), dramatically reducing bandwidth consumption on constrained network links
- **Near-synchronous replication** — Sub-second RPO between nearby clusters, suitable for sites with low-latency connectivity
- **Automated reconciliation** — When connectivity is restored after an outage, Portworx automatically reconciles changes between edge and central clusters according to configurable policy — no manual intervention required
- **PX-Migrate** — Kubernetes-aware migration of applications, volumes, and associated metadata between clusters, enabling workload mobility from edge to data center (or between edge sites) through a single declarative operation

2.4 Multi-Vendor Storage Unification

Many edge environments run applications from multiple vendors on shared infrastructure. Without a common storage abstraction layer, each vendor implements its own storage drivers, replication, and data protection — multiplying integration complexity and creating incompatible silos where cross-application data sharing requires brittle custom code, storage capacity fragments, and backup and lifecycle policies cannot be applied uniformly.

With Portworx, all applications consume storage through standard Kubernetes StorageClasses with differentiated QoS policies — high-IOPS for real-time databases, throughput-optimized for analytics and logistics, and replicated volumes for critical operational data. This composable model unifies storage management regardless of how many vendors share the platform.

2.5 Automated Capacity Management with Autopilot

At edge sites with no on-site storage administrator, storage exhaustion is a common failure mode. Portworx Autopilot monitors storage utilization in real time and takes automated action based on declarative policies:

- **Automatic volume resizing** — Volumes expand as applications consume storage, preventing out-of-space errors
- **Storage pool rebalancing** — Data is automatically redistributed across available drives to optimize capacity and performance
- **Policy-driven tiering** — Data can be automatically tiered from high-performance local storage to lower-cost or remote storage based on access patterns and retention policies

2.6 Unified Support for Containers and Virtual Machines

Many edge environments run a mix of containerized applications and legacy workloads that still require virtual machines. Portworx provides persistent storage for KubeVirt VMs with the same replication, encryption, and data services as container volumes, enabling organizations to:

- Run VMs and containers side by side on a single Kubernetes platform
- Eliminate the need for separate infrastructure and separate teams for VMs vs. containers
- Apply consistent storage policies, backup schedules, and DR configurations across both workload types
- Modernize incrementally by running legacy applications in VMs on the same Kubernetes fabric as new cloud-native services

3. Edge Deployment Architecture

3.1 Deployment Models by Edge Tier

Portworx supports a range of edge deployment sizes, each optimized for specific operational requirements:

ROBO / Far Edge (2–3 Nodes) — Ideal for retail stores, manufacturing lines, remote facilities, and far-edge deployments. This architecture features a compact footprint, RF2 replication for critical workloads, and local snapshots with periodic backup to central object storage. Minimal-footprint clusters use a lightweight arbiter node for quorum.

Near Edge (3–5 Nodes) — Ideal for regional aggregation sites, hospitals, distribution centers, and network edge facilities. Supports RF2/RF3 for higher durability and handles mixed workloads including AI inference, analytics, and databases.

3.2 Hardware Requirements

Portworx is optimized for compact, rugged edge infrastructure across leading OEM portfolios, delivering enterprise-grade storage services in space-constrained environments without dedicated server rooms.

Minimum hardware requirements per storage node:

Component	Minimum	Recommended
CPU Cores (Physical)	8	16
Memory	8 GiB	16 GiB
Network	1 GbE	10 GbE
Network latency (max between workers)	10 ms	—
Storage	312 GiB (OS + metadata + data disks)	312+ GiB

Arbiter node requirements are significantly lower (as few as 4 physical CPU cores, 8 GiB memory, and 184 GiB storage), with up to 200 ms network latency tolerance to the worker nodes — making it possible to deploy the arbiter on compact, low-power, or even remote hardware.

3.3 Integration with Kubernetes Distributions

Portworx integrates with all major Kubernetes distributions deployed at the edge:

- **Red Hat OpenShift**
- **Spectro Cloud Palette**
- **SUSE Rancher**
- **Amazon EKS Anywhere**
- **Azure AKS** (hybrid)
- **Google Anthos**

This distribution-agnostic approach ensures that organizations can standardize on Portworx data services regardless of their chosen Kubernetes platform.

3.4 Centralized Management at Scale

Managing hundreds or thousands of edge clusters individually is impractical. Portworx integrates with centralized management platforms such as Red Hat Advanced Cluster Management (ACM), enabling:

- **Single pane of glass** monitoring and administration across all edge clusters
- **Consistent policy enforcement** for storage, replication, encryption, and backup across the fleet
- **Remote troubleshooting** through telemetry and observability integrations (Prometheus metrics, Grafana dashboards), as well as telemetry to Pure1 for AI-assisted diagnostics and proactive cluster health monitoring
- **Zero-touch provisioning** of Portworx storage on new edge clusters through GitOps workflows
- **Simplified field serviceability** — In the event of a hardware failure, any field technician with minimal training can perform break-fix operations to service edge hardware, without requiring specialized storage expertise

4. Integration Architecture: Edge to Enterprise

Portworx sits at the intersection of the infrastructure layer and the data layer in a modern edge architecture, providing the persistent storage fabric that all applications depend on for stateful operations.

Infrastructure Layer Integration:

- Deploys as a DaemonSet on every Kubernetes worker node, aggregating local NVMe/SSD/HDD into a distributed storage pool
- CSI driver provides standard PersistentVolume and StorageClass APIs — any application receives storage through native Kubernetes primitives
- Integrates with stack monitoring via Prometheus metrics and Grafana dashboards

Data Layer Integration:

- Provides a vendor-neutral persistent substrate for all data services — a robust common storage layer
- Powers data synchronization through built-in replication without additional middleware
- Supports stateful services by ensuring databases (PostgreSQL, Redis, etcd, etc.) have persistent, replicated storage

Application Layer Support:

- Operational databases run on Portworx persistent volumes with synchronous replication
- AI/ML model stores persist training data and model artifacts on high-throughput volumes
- Message queues and event streams maintain durability across pod restarts and node failures

- Legacy applications run as KubeVirt VMs on the same Portworx storage fabric as cloud-native containers

4.1 Connecting Edge to Enterprise

The typical edge computing pattern is: collect and process data locally at the edge, operate autonomously during periods of limited connectivity, and synchronize with central data centers when bandwidth is available. Portworx Enterprise delivers exactly this pattern:

- **During normal connectivity** — Differential asynchronous replication minimizes bandwidth consumption while keeping central systems current with edge data
- **During disconnection** — The local Portworx cluster maintains full data availability and protection through synchronous replication across local nodes, with policy-driven provisioning and lifecycle management continuing to operate without any external dependency
- **On reconnection** — Automated reconciliation synchronizes accumulated changes according to configurable policy, without manual intervention

At the data center tier, Portworx integrates natively with Everpure FlashArray (block) and FlashBlade (file/object) for medium and long-term storage. Data tiers automatically from edge PX-Store volumes to enterprise storage arrays via asynchronous replication, enabling centralized AI training on datasets collected at the edge, long-term archival and compliance retention, and disaster recovery across geographically distributed sites. Everpure Evergreen//One provides a consumption-based storage model for organizations that prefer OpEx-aligned pricing at the data center tier.

5. Policy-as-Code and Compliance

In regulated industries and government or public sector environments, storage infrastructure must meet strict security and compliance requirements. Portworx storage policies are defined entirely as Kubernetes-native YAML specifications, enabling:

- **Declarative security** — Replication factor, encryption settings, IOPS limits, and access policies are encoded in StorageClasses and deployed through standard CI/CD pipelines
- **Encryption** — FIPS-compliant AES-256 encryption at rest and TLS 1.3 in transit, configurable at the cluster-wide or per-volume level
- **Role-based access control** — Kubernetes-native RBAC integration for fine-grained control over storage operations
- **Audit and compliance** — All storage operations are logged and observable through standard Kubernetes monitoring and auditing mechanisms
- **Secure boot support** — Portworx Enterprise supports secure boot for enhanced platform integrity

6. Edge Use Cases

Portworx supports stateful Kubernetes workloads across a wide range of edge verticals:

Vertical	Example Workloads
Retail	Point-of-sale systems, inventory management, video analytics

Vertical	Example Workloads
Manufacturing	MES systems, IoT data processing, quality control
Telecommunications	5G network functions, MEC applications, vRAN
Healthcare	Medical imaging, edge analytics, patient data processing
Energy / Utilities	Remote monitoring and control systems, SCADA
Public Sector	Field operations, distributed C2 applications, logistics

In each case, Portworx provides the persistent, resilient, and self-managing data layer that allows these workloads to run reliably on small Kubernetes clusters without dedicated storage infrastructure or on-site IT expertise.

7. Portworx Edge SKU

For cost-sensitive edge deployments, Portworx offers an Edge-specific SKU designed for small Kubernetes clusters (up to 5 nodes) deployed outside of traditional data centers:

Feature	Portworx Enterprise — Edge
Workloads	Containers and VMs
Max cluster size	5 nodes
Volume attachments per node	256
Autopilot	Included
Encryption	Cluster-wide
Backing storage	Local NVMe, SSD, HDD

This SKU provides the essential Portworx data services — persistent storage, local synchronous replication, failover, Autopilot capacity management, and support for both containers and KubeVirt VMs — at a price point designed for high-volume edge deployments, optimized for cost efficiency when scaling to hundreds of locations.

Organizations requiring cross-cluster disaster recovery (asynchronous or synchronous replication to a central data center, as described in Section 4) can deploy Portworx Enterprise with the DR add-on for full edge-to-enterprise data mobility.

Customers deploying Red Hat OpenShift with Portworx at the edge can anticipate up to a **65% reduction in licensing costs** compared to legacy VMware-based alternatives.

8. Summary

Kubernetes at the edge requires more than container orchestration. It requires a persistent, self-managing data layer that can handle the realities of distributed infrastructure: limited hardware, no on-site IT, intermittent connectivity, and the need for data to flow between edge and enterprise.

Portworx Enterprise delivers this data layer as a fully Kubernetes-native platform:

- **Local drive management** — Automatic aggregation of heterogeneous local drives into a unified storage pool
- **Failure resilience** — Synchronous replication (RF2/RF3), automatic failover with self-healing, live migration, and quorum-based split-brain protection
- **Snapshots and data protection** — Application-consistent snapshots, scheduled backup policies, encryption, and ransomware protection
- **Data migration** — Differential block-level replication from edge to data center with automated reconnection and reconciliation
- **Multi-vendor unification** — A single Kubernetes-native storage fabric for all applications, eliminating per-vendor silos
- **Autonomous operations** — Fully software-defined with no external storage dependencies, operating independently during WAN outages
- **Kubernetes control plane integration** — Every capability is exposed through native Kubernetes constructs (StorageClasses, PVCs, CRDs) and managed through standard DevOps tooling

By extending enterprise-grade data services to edge Kubernetes clusters, Portworx enables organizations to run stateful workloads at the edge with confidence — knowing their data is protected, available, and connected to the broader enterprise.

Learn More

- **Portworx Enterprise Documentation:** https://docs.portworx.com
- **Red Hat OpenShift with Portworx Validated Design:** https://www.purestorage.com/docs.html?item=/type/pdf/subtype/doc/path/content/dam/pdf/en/validated-design-guides/vd-red-hat-openshift-with-portworx.pdf
- **Spectro Cloud VMO with Portworx Enterprise Reference Architecture:** https://www.spectrocloud.com/resources/collateral/vmo-architecture-pdf
- **Portworx by Everpure:** https://portworx.com