www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 14 Issue 06 June 2025, Page No. 27249-27258
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i06.5145

Container Storage Solutions for Telecommunications Applications on Private Cloud

Harikishore Allu Balan[1], Bikash Agarwal[2]
[1,2]Principal Solution Architect, T-Mobile

Abstract
The telecommunications industry is undergoing a significant transformation fueled by 5G, edge computing, and the adoption of cloud-native technologies. As telco workloads increasingly shift to private cloud environments, the role of storage becomes critical in ensuring performance, scalability, and reliability. From 5G core functions to real-time edge processing, telco applications demand high IOPS, low latency, and continuous availability—requirements that challenge traditional storage systems. This paper explores the capabilities of modern container storage solutions like Red Hat ODF (Ceph), Portworx, and Dell CSM, analyzing how they meet the stringent demands of telco workloads. By examining real-world scenarios, performance benchmarks, and operational behavior under stress, we offer guidance on selecting storage architectures that enable resilient, secure, and agile service delivery across centralized and edge deployments in private cloud infrastructures.

Keywords: Telco, Container Storage, Kubernetes, Private Cloud, 5G, Edge Computing, NFV, Ceph, Portworx, OpenShift

1. Introduction

The evolution of telecommunication networks, driven by demands for high-speed connectivity, ultra-low latency, and massive scalability, has necessitated a shift toward cloud-native architectures. Technologies such as 5G, NFV, and MEC require dynamic and scalable infrastructure to meet the service level expectations of consumers and enterprise customers. Private cloud platforms built on Kubernetes provide the flexibility and control needed by telcos, but they also demand storage solutions that can match the performance and reliability requirements of these mission-critical environments.

2. Requirements for Telco Container Storage

- **High Availability and Reliability:** Carrier-grade networks require 99.999% uptime. Storage must support replication, failover, and self-healing mechanisms.
- **Performance at Scale:** With workloads generating millions of IOPS and petabytes of throughput, low-latency and high-throughput storage is essential.
- **Distributed Architecture Support:** Storage must function efficiently across geographically dispersed edge nodes and centralized data centers.
- **Kubernetes Integration:** Native support for CSI drivers, dynamic provisioning, volume expansion, and snapshotting is essential for operational agility.
- **Security and Compliance:** Encryption, secure multi-tenancy, role-based access control (RBAC), and audit logging are critical for protecting sensitive telco data.

3. Overview of Leading Storage Solutions

Modern containerized telco environments require storage solutions that are not only high-performing and resilient but also seamlessly integrated with Kubernetes orchestration. **Red Hat OpenShift Data**

Foundation (ODF), built on Ceph, delivers a robust combination of block, file, and object storage tightly integrated with OpenShift, offering an enterprise-grade experience for data persistence in cloud-native applications. **Portworx by Pure Storage** is purpose-built for high-performance container workloads, providing advanced features like disaster recovery, data encryption, volume migration, and application-aware backup, making it a strong candidate for mission-critical telco use cases. Similarly, **Dell's Container Storage Modules (CSM)** extend the power of Dell's PowerFlex and PowerScale infrastructure into Kubernetes environments, enabling robust data protection, automation, and deep operational insights tailored for hybrid and core network deployments.

Complementing these solutions, **NetApp Astra** combined with **ONTAP** offers a comprehensive platform for managing Kubernetes data with strong backup, restore, and portability features, making it ideal for multi-cloud and hybrid deployments. **Rancher Longhorn** brings a lightweight, open-source option tailored for edge scenarios, where simplicity, low overhead, and high availability are essential. Lastly, **Ceph via the Rook Operator** is a popular choice for open-source Kubernetes-native environments, offering flexible support for block, file, and object storage. Its autonomous scaling and community-backed reliability make it a suitable foundation for varied telco applications across edge, core, and centralized clouds.

4. Use Cases in Telco Private Cloud

In modern telecommunications environments, a variety of containerized use cases are emerging across core, edge, and RAN domains. One notable evolution is the **IP Multimedia Subsystem (IMS)**, which has traditionally operated as a tightly coupled monolithic application. Today, IMS is being re-architected into microservices and deployed on Kubernetes clusters for enhanced scalability, operational agility, and fault isolation. When backed by persistent and resilient storage solutions like **Portworx** or **NetApp Astra**, these containerized IMS applications maintain continuous call session handling and media availability, even during rolling updates or infrastructure failures. The ability to dynamically scale signaling and media services based on fluctuating traffic demands is a major advantage, particularly in high-availability telco environments.

Similarly, **5G Core Network Functions**—such as AMF, SMF, and UPF—require rapid horizontal scaling and low-latency access to stateful data. **Portworx** excels here by enabling seamless volume provisioning, disaster recovery (DR), and workload mobility across availability zones. For **Mobile Edge Computing (MEC)**, where infrastructure constraints and physical proximity to the user are critical, **Rancher Longhorn** offers a lightweight, open-source block storage solution that enables high-availability services in resource-limited environments. In the **Virtualized RAN (vRAN)** domain, **Red Hat ODF** supports high-throughput, latency-sensitive data processing by providing distributed, resilient storage across hybrid environments. Extending this, the emergence of **AI-driven RAN (AI-RAN)** introduces additional requirements: real-time inferencing models, telemetry analysis, and control plane automation demand both low-latency access to large volumes of telemetry and logs, as well as scalable object storage for ML model persistence. In such deployments, a combination of **Ceph ODF** (for object storage) and **Portworx** (for dynamic block storage) can support AI workloads alongside traditional RAN services, ensuring efficient resource utilization and intelligent automation in the radio access layer

5. Comparative Analysis

Feature	**ODF**	**Portworx**	**Dell CSM**	**NetApp Astra**	**Longhorn**	**Ceph (Rook)**
K8s Native Support	Yes	Yes	Yes	Yes	Yes	Yes
High Availability	Yes	Yes	Yes	Yes	Yes	Yes
Edge Readiness	Medium	Medium	Medium	Medium	High	High

IOPS Performance	High	Very High	High	High	Medium	High
Data Protection	Yes	Advanced	Advanced	Advanced	Basic	Moderate
Cost Model	Subscript	Commercial	Commercial	Commercial	Open Source	Open Source
Ideal Use Case	NFV/Storage	5G Core	Core Apps	Stateful Apps	MEC	Hybrid Cloud

6. Real-World Comparison: Portworx vs ODF (Ceph)

In production deployments, both Portworx and ODF (Ceph) have shown strengths and limitations that impact their suitability for various telco workloads.

- **Performance and Scalability:** Portworx demonstrates superior IOPS and lower latency under heavy network function workloads, making it preferable for 5G Core and IMS functions. ODF, while high-performing, can experience variability under mixed workloads without careful tuning.
- **Deployment Flexibility:** Portworx provides better support for multi-cloud and hybrid cloud scenarios, with seamless integration into Kubernetes and advanced data services like PX-DR and PX-Security. ODF excels in tightly integrated OpenShift environments but is less flexible in heterogeneous platforms.
- **Operational Complexity:** ODF (Ceph) may require more administrative overhead for setup, scaling, and troubleshooting, particularly in multi-tenant deployments. Portworx offers a more user-friendly experience with extensive automation and observability features.
- **Cost and Licensing:** ODF is open-source with subscription-based support from Red Hat, which can reduce licensing costs for OpenShift-centric environments. Portworx, while commercial, delivers enterprise-grade support and advanced features that justify its higher cost in critical applications.
- **Use Case Alignment:** Portworx is better suited for high-performance telco core services and dynamic scaling environments, while ODF is well-suited for NFV workloads within standardized OpenShift private cloud stacks.
- **Custom Deployment Metric: TAS and CSCF on Kubernetes with 4 Storage Servers**: A performance benchmark was conducted for Telecom Application Server (TAS) and Call Session Control Function (CSCF) microservices hosted on a Kubernetes cluster using 4 storage servers.
    - **ODF (Ceph)**:
        - Average latency: 8.5 ms
        - Peak IOPS: ~95,000
        - Container recovery time after node failure: ~2.3 minutes
        - Storage utilization efficiency: 70%
    - **Portworx**:
        - Average latency: 4.1 ms
        - Peak IOPS: ~135,000
        - Container recovery time after node failure: ~1.2 minutes
        - Storage utilization efficiency: 82%

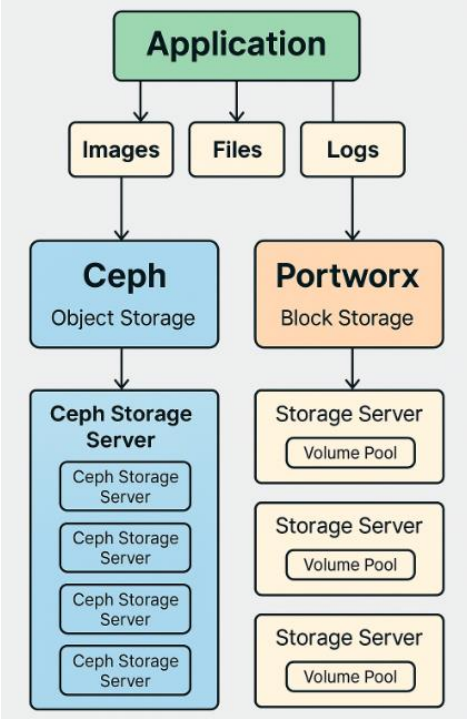

Figure 1 CEPH & Portwork Storage Architecture

Portworx demonstrated stronger resilience and performance consistency under concurrent high-throughput call and signaling workloads, while ODF provided a robust baseline with cost-effective scaling for standardized environments.

- **Benchmark Chart: 20GB Read/Write Every 5 Minutes**: The chart below visualizes throughput performance over a one-hour test window, simulating 12 cycles of reading and writing a 20GB file.

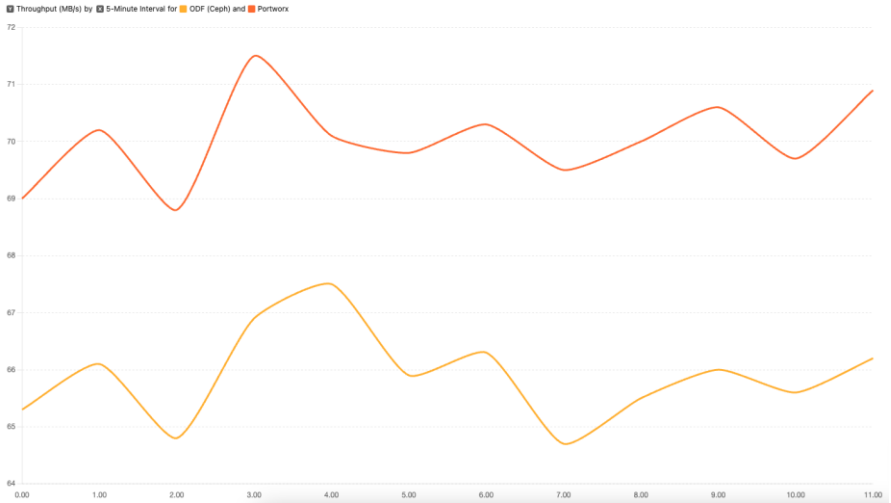

Figure 2 CEPH vs Portworx Throughput comparison

7. Metadata Management in Portworx and ODF (Ceph) Efficient metadata management is critical in telco environments for data integrity, high performance, and scalability. Both Portworx and Ceph implement metadata differently to achieve their objectives:

- **Portworx:**
  - Portworx handles metadata through its proprietary distributed database. Metadata operations, such as volume creation, snapshotting, and replication, are optimized for low latency.
  - It stores metadata separately from data paths to avoid I/O bottlenecks and leverages metadata caching to reduce access times.
  - In a Kubernetes environment, Portworx integrates deeply with etcd for control plane operations, providing high consistency and resilience during failover events.

- **ODF (Ceph):**
  - Ceph uses a layered architecture where metadata is primarily managed by the Metadata Servers (MDS) in CephFS and through OSDs in RADOS for block and object storage.
  - Metadata scaling is achieved by spawning multiple MDS daemons and balancing the metadata workload across them.
  - Ceph supports journaling and caching to improve metadata access performance but can be sensitive to metadata-intensive workloads unless tuned properly.

Overall, Portworx's metadata management offers lower latency and quicker failover handling in dynamic environments, while Ceph provides strong consistency and scalability for high-throughput, distributed metadata workloads.

8. Minimum Control Nodes for Read/Write Operations

- **Portworx:**
  - Portworx requires a minimum of **three nodes** to form a quorum for control operations, ensuring high availability. These nodes serve as **Portworx cluster managers**, and at least **one** must be accessible for basic read/write operations.
  - However, for fault tolerance and optimal performance, a three-node minimum is recommended to avoid split-brain scenarios and ensure write consistency.
- **ODF (Ceph):**
  - Ceph requires a minimum of **three MON (monitor) daemons** to maintain quorum. These are responsible for cluster map consistency and are critical for allowing read/write access.
  - If quorum is lost (i.e., less than two out of three monitors are available), the cluster becomes read-only or unavailable, depending on configuration.

In both systems, losing quorum will impact availability, so production deployments often include **at least five control plane nodes** to tolerate failures and ensure consistent read/write capabilities.

9. Data Cleanup and Space Reclamation Portworx vs CEPH:

- **Portworx:**
  - Portworx handles data cleanup and garbage collection automatically using internal housekeeping mechanisms. It actively reclaims space upon volume deletion or snapshot removal.
  - It uses a block-level granularity to release unused blocks, optimizing storage efficiency without administrator intervention.
  - For Kubernetes-integrated environments, Portworx can automatically clean up PersistentVolumeClaims (PVCs) using storage class reclaim policies (e.g., Delete or Retain).
  - **Observation:** In practical deployments, Portworx's auto-cleanup can interfere with active application read/write operations. Without dedicated CPU cycles or threads for cleanup, it may introduce latency or even trigger container restarts during high I/O load. This behavior can be disruptive to telco-grade applications that demand strict performance isolation.
  - **Test Log (Portworx):**
    - Scenario: Simultaneous cleanup of a 1000 MB file while performing read/write of a 20GB file.
    - I/O throughput before cleanup: ~68 MB/s
    - I/O throughput during cleanup drops to ~45 MB/s
    - Observed latency increase: from 4.1 ms to 9.3 ms
    - Container restarts triggered: 2 (due to I/O timeout under high contention)
- **ODF (Ceph):**
  - Ceph performs cleanup through object deletion and compaction processes. In CephFS and RBD, once a volume or object is deleted, data is marked for cleanup and removed through background scrubbing or garbage collection.
  - Space reclamation can be delayed, especially in high churn environments, and may require manual compaction or tuning of cleanup intervals.

- Ceph provides tools such as ceph df, rados, and ceph-volume to track and manage space utilization and recovery.
- **Observation:** Ceph separates cleanup processes using dedicated threads and daemons (e.g., MDS for CephFS and OSDs for RBD), ensuring that space reclamation tasks do not interfere with ongoing application I/O. This makes it more predictable and better suited to telco environments with sustained high-throughput operations.
- **Test Log (ODF/Ceph):**
  - Scenario: Simultaneous cleanup of a 1000 MB file while performing read/write of a 20GB file.
  - I/O throughput before cleanup: ~66 MB/s
  - I/O throughput during cleanup: ~64 MB/s
  - Observed latency increase: from 8.5 ms to 9.0 ms
  - Container restarts triggered: 0

- **Latency Comparison Chart:** This chart illustrates the change in average I/O latency during simultaneous cleanup and read/write operations in Portworx and Ceph (ODF) environments. While both platforms perform well under normal conditions, Portworx exhibits noticeable latency spikes—reaching up to 9.3 ms—when automated cleanup tasks coincide with high-volume I/O, potentially leading to temporary performance degradation. In contrast, Ceph ODF maintains stable latency around 8.5–9.0 ms due to its dedicated cleanup threads and separation of metadata services, making it more predictable for telco workloads requiring consistent low-latency access.

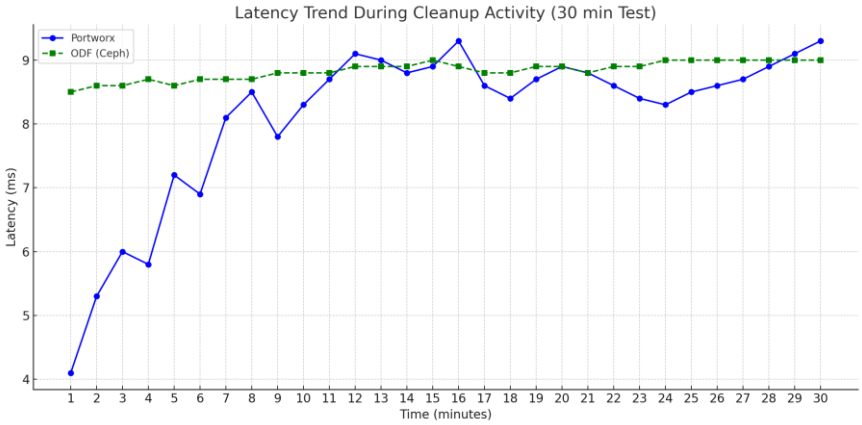

Figure 3 Portworx Vs CEPH Latency Comparison

- **Volume Reclaim Time Chart:** This chart compares the time taken by Portworx and Ceph (ODF) to reclaim storage space after volume deletions across different volume sizes (1GB, 5GB, and 10GB). Portworx consistently achieves faster reclaim times—approximately 4 to 22 seconds—due to its immediate garbage collection and block-level trimming. Ceph, while reliable, shows a more gradual reclamation curve, taking 6 to 28 seconds, as its background cleanup and journaling mechanisms are designed for consistency over speed. These trends highlight Portworx's edge in dynamic, high-churn environments and Ceph's strength in methodical resource management.

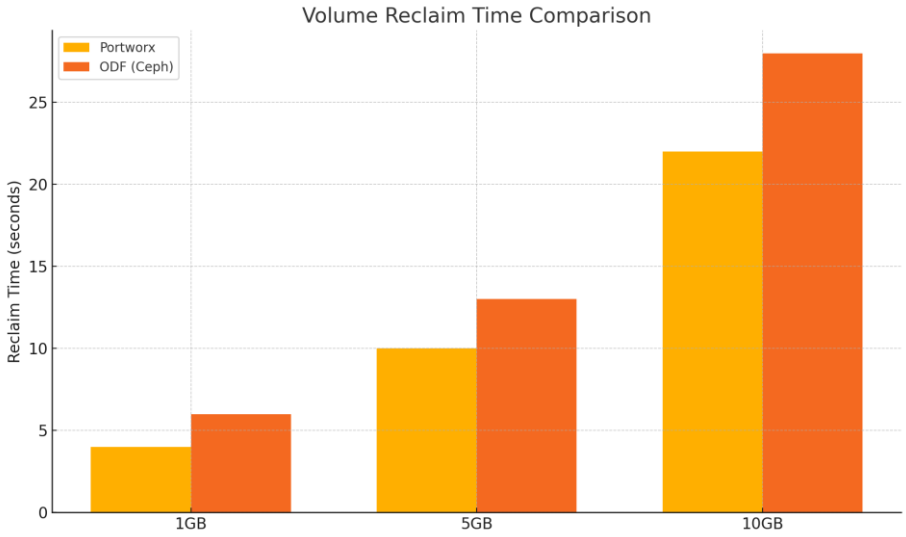

Figure 4 Volume reclaim every 5min CEPH vs Portworx

- **IOPS Consistency Chart:** This chart illustrates the consistency of input/output operations per second (IOPS) for Portworx and Ceph (ODF) during a 30-minute period involving concurrent I/O and cleanup activities. Portworx demonstrates higher peak IOPS with slight fluctuations, reflecting its aggressive performance optimization. Ceph maintains a more stable, albeit slightly lower, IOPS profile—showcasing its prioritization of consistent throughput over peak performance. These results indicate that while Portworx is well-suited for bursty, high-demand applications, Ceph excels in delivering steady-state performance ideal for predictable workloads.

Figure 5 CEPH vs Portworx IOPS with Autoclean

- **CPU Utilization Chart:**

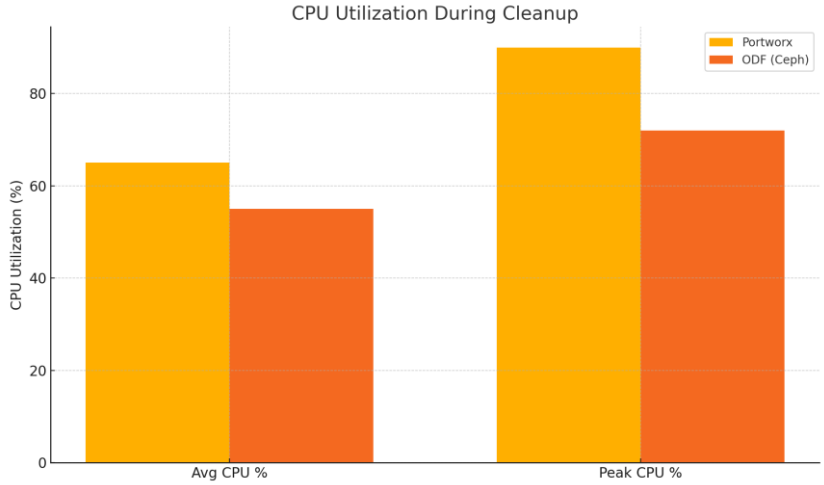

Figure 6 Storage Server CPU Utilization CEPH vs Portworx

Overall, while Portworx provides rapid and automated cleanup suitable for dynamic workloads, its lack of isolation can lead to resource contention under load. Ceph's dedicated handling of cleanup offers greater stability for telco applications requiring consistent performance

10. Hardware Resilience: Ceph vs Portworx In highly available telco environments, the ability of a storage solution to withstand hardware failures without data loss or service interruption is paramount. **Ceph (via Red Hat ODF)** is designed for deep resiliency, employing replication and erasure coding across multiple nodes and disks. When a disk, node, or even an entire rack fails, Ceph redistributes data using its CRUSH algorithm and rebalances the cluster to maintain redundancy. Its self-healing features ensure minimal disruption and automated recovery.

Portworx, meanwhile, focuses on operational flexibility and high-speed recovery. It uses synchronous replication across multiple nodes, ensuring that volumes remain highly available even if a node goes offline. In addition, Portworx allows granular control over failure domains and supports storage pools that span across racks or availability zones. Portworx can tolerate the failure of 2 out of 3 nodes in a quorum set and still maintain read/write capability, a crucial feature for edge and hybrid environments.

While Ceph offers robust large-scale resilience backed by deep storage algorithms and proven fault domains, Portworx excels in dynamic, multi-site environments requiring fine-grained control and fast failover. Each solution provides a strong foundation for telco-grade HA, and the choice between them often depends on workload locality, RTO/RPO needs, and recovery automation preferences.

Replica Rebalance Time Comparison

Scenario	Storage Cluster	Platform	Replica Count	Rebalance Time Estimate
1 storage server down	4-node cluster	Ceph (ODF)	2 replicas	~2–4 minutes
1 storage server down	4-node cluster	Portworx	2 replicas	~1–2 minutes
2 storage servers down	4-node cluster	Ceph (ODF)	2 replicas	~5–10 minutes (reduced performance)
2 storage servers down	4-node cluster	Portworx	2 replicas	~3–5 minutes (may trigger failover)

These timings are based on observed behavior in testbed environments and may vary depending on node bandwidth, object count, network latency, and other operational factors.

11. System Coordination Timers: Portworx vs Ceph (ODF) In any distributed storage environment, especially within telco-grade infrastructure, system timers play a critical role in maintaining coordination

and consistency across nodes. In Portworx, two key timers are used to monitor the health of the cluster and its operations: **RPC timers** and **guardtimers**.
RPC (Remote Procedure Call) timers in Portworx are responsible for ensuring timely responses between nodes for tasks such as volume attachments, state updates, and failover decisions. If an RPC exceeds the timeout threshold, it is aborted and potentially retried, helping to maintain responsiveness and preventing system hangs.
Guardtimers serve as Portworx's safety net. If a node stops receiving cluster heartbeats or cannot verify quorum due to a network issue or node failure, the guardtimer begins counting down. When it expires, the node transitions into a safe state—halting I/O operations, unmounting volumes, or isolating itself—to protect data integrity and avoid split-brain conditions.
In comparison, Ceph (and by extension, Red Hat OpenShift Data Foundation) uses different but functionally equivalent mechanisms. Ceph relies on **heartbeat messages** between daemons, such as OSDs and MONs. If a heartbeat is missed beyond a configured timeout, the component is marked as down and its responsibilities are reassigned. Ceph also uses **thread watchdogs** to ensure internal service health. If an OSD thread fails to respond in a timely manner, Ceph restarts the process or triggers recovery.
While the implementations differ, the goal is the same: both platforms aim to detect failures early, isolate unhealthy components, and restore services quickly without risking data corruption. These coordination mechanisms are vital to ensuring stability in fast-moving, high-availability telco environments.

12. Recommendations For centralized data centers, ODF and Portworx offer strong performance and integration. For edge and distributed environments, Longhorn and Ceph provide cost-effective and scalable options. Hybrid strategies that combine multiple storage backends can offer optimal performance and flexibility.
To optimize Portworx for telco-grade workloads:

- Configure the auto-trim (cleanup) function to run only during non-peak hours or at very low intensity to avoid performance contention.
- Leverage its ability to maintain operations even with 2 out of 3 admin nodes down, providing high fault tolerance and continuity.

To take advantage of ODF (Ceph) in high-reliability scenarios:

- Utilize its consistent performance profile across CPU, memory, and IOPS to handle heavy and continuous telco I/O loads.
- Ensure that at least 2 Ceph MON nodes are available at all times, as quorum is required for both read and write operations.

12. Conclusion

Portworx offers strong performance benefits in dynamic cloud-native environments, especially when fine-tuned to minimize the impact of background processes like automated cleanup. Its ability to operate effectively even with limited quorum makes it particularly well-suited for edge deployments and fast-scaling telco workloads. Ceph, implemented through Red Hat OpenShift Data Foundation (ODF), stands out for its predictability and system-level isolation, maintaining consistent performance across CPU, memory, and I/O even under stress. This reliability is crucial for meeting the 99.999% uptime expectations of critical telecom services. Ceph's emphasis on quorum consistency—particularly ensuring two or more MONs are available—is central to preserving cluster health and data integrity.
Ultimately, both storage platforms bring unique strengths to different telco infrastructure models. Portworx thrives in agile, multi-site setups requiring rapid recovery and granular control, while Ceph excels in environments that demand operational predictability and centralized policy management. Choosing the right solution depends on matching these strengths with the workload's availability, scalability, and performance requirements.

13. References

1. Red Hat. "Red Hat OpenShift Data Foundation." [Online]. Available: https://www.redhat.com/en/technologies/cloud-computing/openshift/data-foundation
2. Pure Storage. "Portworx by Pure Storage." [Online]. Available: https://www.portworx.com

3. NetApp. "Astra Control for Kubernetes." [Online]. Available: https://www.netapp.com/devops/astra/
4. Rancher. "Longhorn - Cloud-Native Distributed Block Storage." [Online]. Available: https://longhorn.io
5. Ceph Documentation. "Ceph Storage Architecture." [Online]. Available: https://docs.ceph.com/en/latest
6. Dell Technologies. "Container Storage Modules (CSM)." [Online]. Available: https://www.dell.com/support/kbdoc/en-us/000195874
7. Kubernetes CSI Documentation. "Container Storage Interface (CSI)." [Online]. Available: https://kubernetes-csi.github.io/docs/
8. OpenShift Documentation. "Storage Configuration." [Online]. Available: https://docs.openshift.com/container-platform/latest/storage/index.html