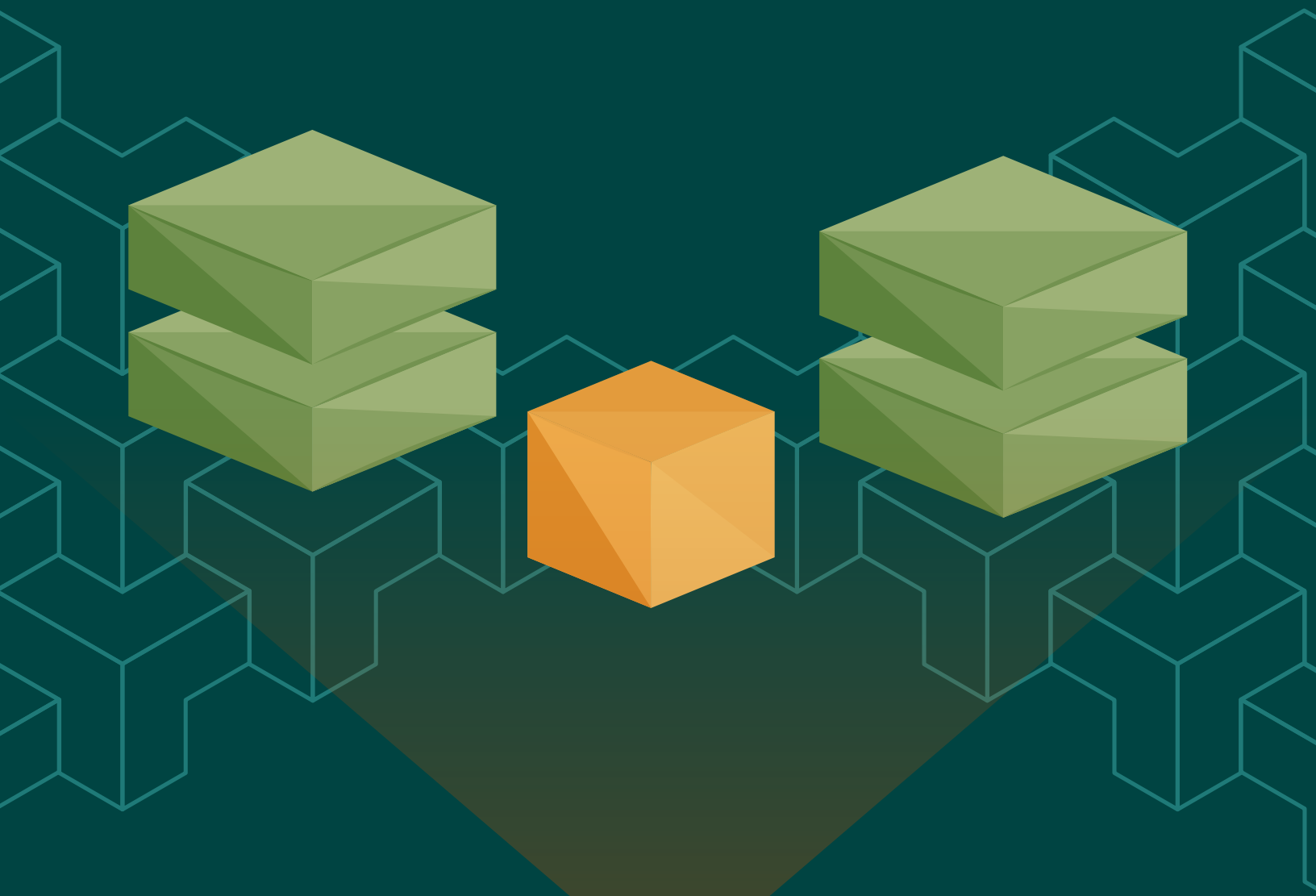
Spectro Cloud

Everpure™ formerly Pure Storage

High availability in edge Kubernetes without the third full node

How Spectro Cloud and Portworx by Everpure deliver resilient storage with a two-node + witness architecture



Executive summary

Three-node high availability is a core requirement across the tech industry for distributed systems, including Kubernetes, to maintain quorum and prevent split-brain conditions. That model works well in centralized environments, but it can add unnecessary infrastructure at edge, remote, and resource-constrained locations where you may only need the compute and storage capacity of two servers.

Spectro Cloud and Portworx provide an alternative with a two-node plus witness architecture. Two primary nodes run application workloads and maintain synchronously replicated Portworx persistent storage. A smaller third node acts as a witness, providing the additional vote required to maintain quorum, without itself hosting application workloads or replicated application data.

If either primary node fails, the surviving primary node and witness retain quorum, allowing applications to continue operating against the surviving copy of their persistent data. If the witness fails, the two primary nodes retain quorum and continue operating.

Spectro Cloud PaletteAI manages the cluster configuration and lifecycle, **Kairos** provides the immutable operating foundation, and Portworx provides persistent storage and storage consensus.

For organizations deploying Kubernetes, virtual machines, and stateful applications across distributed locations, the architecture provides high availability without requiring a third full-sized server at every site.

The challenge with two nodes

Distributed infrastructure has always involved a tradeoff between resilience and footprint. High availability usually means more infrastructure: additional servers, storage, networking, power, and systems to operate. In a centralized data center, the incremental cost of another server may be relatively easy to absorb. At the edge or across hundreds of remote locations, that equation changes quickly.

Many organizations deploy two substantial servers to provide both the compute capacity required by their applications and the redundancy needed for site resiliency. Deploying a single, more powerful server might satisfy performance requirements, but it creates a single point of failure. Conversely, adding a third equivalently configured server purely for quorum provides little extra application value, even though traditional high-availability architectures often demand it.

The issue isn't simply having another copy of an application or its data. Distributed systems also need a quorum to safely commit changes to cluster state, ensuring that only a majority-backed set of nodes can continue writing after a failure.

With only two voting members, that becomes difficult.

If Node A loses communication with Node B, it can't determine whether Node B has failed or whether the network connection between them has failed. Node B has the same problem. If both continue independently, the cluster can enter a split-brain condition, where each side believes it is authoritative.

For stateful applications, a split brain can result in conflicting decisions about workload ownership, storage state, or which copy of the data is current.

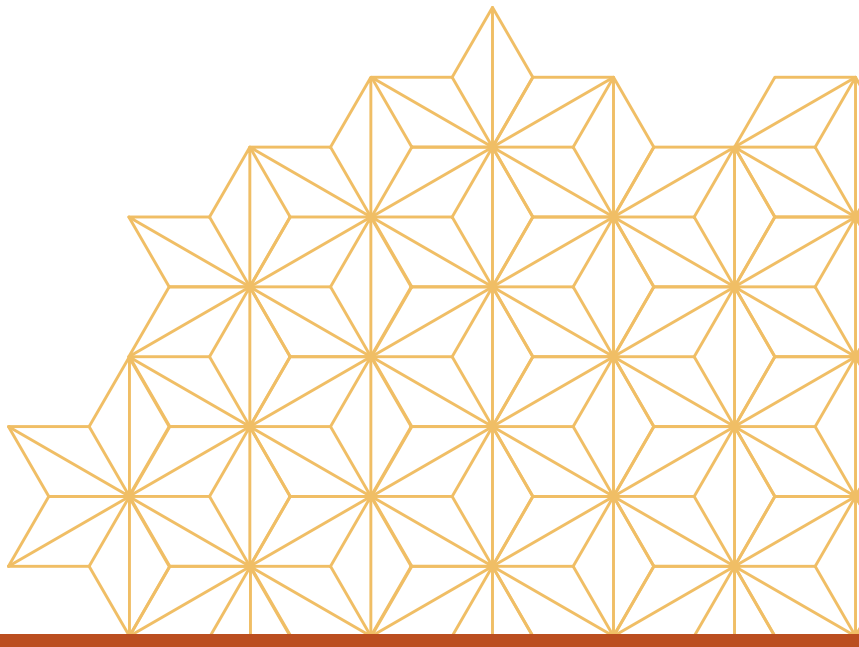
Distributed systems prevent this by requiring a majority, or quorum, before allowing a portion of the cluster to continue making authoritative changes. With two members, neither node can establish a majority if communication between them is lost.

Adding a third vote solves that problem. But that third vote doesn't necessarily require another full application and storage server.

That's the role of the witness.

It should be noted that two-node high availability is one way to address the infrastructure challenge without adding a third server. Spectro Cloud, for example, supports a two-node HA architecture that allows either node to continue operating if the other fails. This provides application availability with the smallest possible hardware footprint and can be a good fit for highly constrained edge locations.

But two-node HA uses a different approach to availability rather than providing a traditional three-member quorum. For organizations that specifically require quorum-based consensus, whether because of application, storage, consistency, or operational requirements, a two-node architecture may not be viable. Thankfully two-node + witness is a very strong option in these situations.



How two-node + witness works

The Spectro Cloud and Portworx architecture consists of two primary nodes and a smaller witness node.

The primary nodes provide the compute and storage resources for the environment. They run Kubernetes workloads, including containerized applications and virtual machines, and Portworx synchronously replicates persistent data between them.

The witness participates in quorum but doesn't host normal application workloads or replicated application data. For Portworx, all three nodes participate in the internal quorum, while application data remains on the two storage nodes.

Under normal operation, all three nodes participate in quorum and both primary nodes maintain copies of persistent data.

If one primary node fails, the remaining primary node and witness still provide two of the three votes. Quorum remains intact, persistent data is available from the surviving Portworx replica, and workloads can be rescheduled onto the remaining primary node.

If the witness fails, the two primary nodes still make up a majority. Both storage replicas remain available and applications continue operating.

The witness supplies the third vote needed for consensus without having to be sized like the systems running the applications.

Why use a witness?

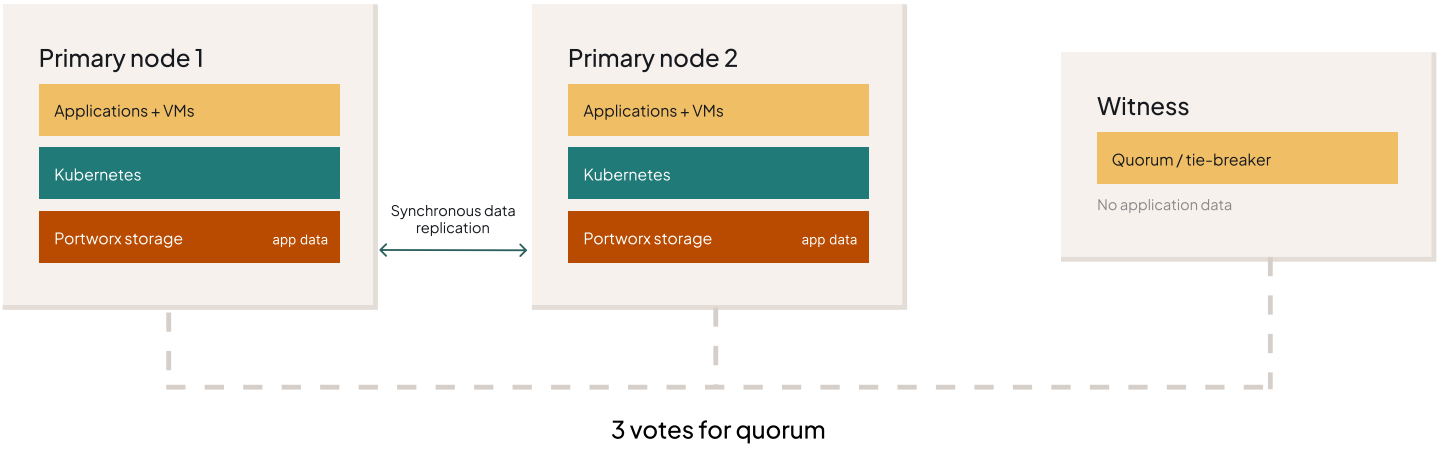
The technical reason for the witness is quorum. The practical reason is that many distributed sites don't need three full servers or do not have the space/capacity to support them.

A traditional three-node architecture makes sense when the application needs the compute and storage capacity of all three systems. When the third node exists primarily to satisfy availability requirements, however, you may be provisioning far more hardware than that role requires.

At a single location, the difference may not seem significant. Across hundreds or thousands of edge, retail, industrial, or branch locations, it can mean a substantial amount of additional hardware, storage, power, rack space, support, and lifecycle management.

The two primary nodes in a witness architecture remain appropriately sized for the workloads they run. The witness can be significantly smaller because it isn't an application or storage node. This can also make two-node + witness useful when you already have two-server locations in place. Instead of replacing those systems or adding another equivalently sized server solely for quorum, you can add a smaller witness where the workload and availability requirements support the architecture.

Two full nodes hold workloads and data; a small witness provides the third vote



Maintaining high availability

Portworx provides the persistent storage layer across the two primary nodes.

In the reference architecture, persistent volumes use a replication factor of two, placing a copy on each storage node. The witness doesn't contain application storage.

During normal operation, applications have access to replicated persistent data across both primary systems. If one of those systems fails, the surviving primary node still has a copy of the data. The surviving primary and witness also maintain quorum. Applications designed for failover can be rescheduled onto the remaining node and continue using their persistent storage. The cluster is operating in a degraded state at this point because only one storage replica remains. After the failed node is restored and rejoins the cluster, Portworx can restore the configured replication level.

A witness failure is less disruptive. Both primary nodes remain online, both copies of the application data are available, and the two primary nodes still constitute a majority of the three quorum members. The witness should still be restored promptly because the cluster has temporarily lost its ability to tolerate another node failure.

The architecture is designed to tolerate any single node failure while keeping the cluster operational.

Making the architecture repeatable with Spectro Cloud

Portworx handles the distributed storage and storage quorum. Spectro Cloud provides the cluster lifecycle and orchestration around it. This matters for distributed environments because manually configuring a specialized architecture at one site is manageable. Repeating it consistently across a large number of sites is much harder.

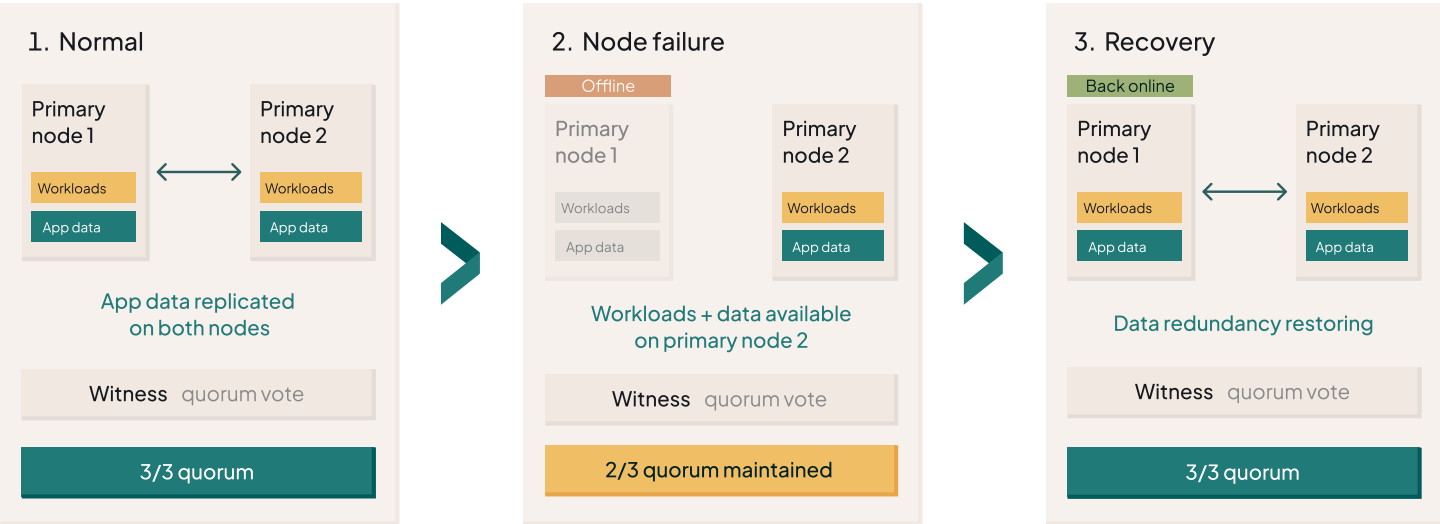
Spectro Cloud PaletteAI uses declarative Cluster Profiles to define the software and configuration used to build and manage a cluster. The two primary nodes can be configured as workload and Portworx storage nodes, while the witness receives the appropriate quorum role without becoming an application storage target.

Spectro Cloud's edge architecture also uses Kairos-based immutable operating system images. Host dependencies required by Portworx can be incorporated into the image, and storage can be configured according to each node's role during provisioning. The primary nodes receive the storage required for Portworx data and metadata, while the witness only needs what's required for its quorum responsibilities.

Together, PaletteAI and Kairos let platform teams define and deploy the two-node + witness architecture consistently rather than building and maintaining each site individually.

What happens when a node fails

The surviving primary node and the witness keep quorum, so applications keep running



Choosing the right high-availability architecture

Two-node + witness is one of several ways to provide high availability, and it is not intended to replace either Spectro Cloud's two-node HA or a traditional three-node architecture in every environment. Each approach addresses a different set of infrastructure requirements.

1. Spectro Cloud two-node HA: This provides high availability using only two full nodes, making it a good fit where minimizing the physical infrastructure footprint is a priority. The tradeoff is that it is a fixed architecture: a cluster deployed as two-node remains a two-node cluster and cannot later be expanded.

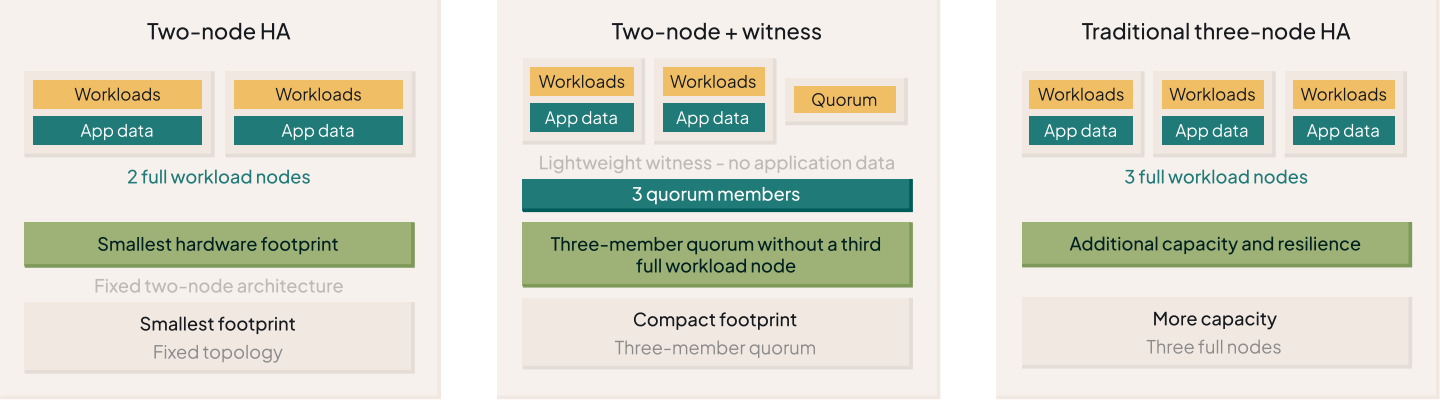
2. Traditional three-node architecture: This provides three full workload-bearing nodes. This makes sense when the environment can use the additional compute and storage capacity, or when three full nodes are appropriate for application availability and placement requirements. The tradeoff is the additional hardware footprint at each location.

3. Two-node + witness: This third option provides two full-sized nodes for the compute and **Portworx** storage capacity and a smaller third node for the additional quorum member. This provides a three-member quorum without requiring the compute, storage, power, and physical footprint of a third full-sized workload node.

The right architecture depends on what a site actually needs. Two-node HA prioritizes the smallest possible infrastructure footprint. Two-node + witness provides three-member quorum while keeping the third node lightweight. Traditional three-node HA provides additional workload capacity and resilience across three full nodes.

	Spectro Cloud two-node HA	Two-node + witness	Traditional three-node HA
Physical nodes	2	2 primary + 1 lightweight witness	3
Full workload nodes	2	2	3
Portworx storage nodes	Variable	2	3
Third quorum member	No separate witness (non-quorum replacement for etcd)	Lightweight witness	Full node
Application data on third node	N/A	No	Yes
Cluster expansion	Fixed at two nodes	Not constrained by native two-node HA model	Standard multi-node architecture
Hardware footprint	Lowest	Lower than three full nodes	Highest
Primary benefit	Maximum availability with the smallest possible hardware footprint	Three-member quorum without a third full workload node	Additional capacity and resilience across three full nodes

Choose the HA architecture that fits the site



Spectro Cloud and Portworx

The two-node + witness architecture builds on an existing technical integration between Spectro Cloud and Portworx by Everpure.

Spectro Cloud has supported Portworx integrations in PaletteAI across cloud, virtualized, bare-metal, and edge environments. The integration also supports persistent storage for both containerized workloads and virtual machines running on Kubernetes, including storage requirements associated with VM mobility and migration.

For the two-node + witness architecture, Portworx provides the persistent storage and storage quorum capabilities while Spectro Cloud manages the Kubernetes environment, operating configuration, and cluster lifecycle.

Conclusion

For many edge and distributed sites, three full servers is more infrastructure than the applications actually require. Two servers may provide sufficient compute and storage capacity, while the third system is needed primarily to establish quorum. A two-node + witness architecture addresses that situation directly.

Portworx replicates persistent data across the two primary nodes and uses the witness as the third quorum member. Spectro Cloud PaletteAI and Kairos provide a consistent way to deploy and manage the architecture across distributed environments. The result is a smaller infrastructure footprint that still provides the quorum and storage behavior required for high availability.

If you're considering the architecture, evaluate your workload capacity, storage, failure-domain, and recovery requirements to determine whether two-node + witness or a traditional three-node topology is the better fit.

About Portworx by Everpure

Portworx by **Everpure** is a cloud-native storage and data management platform for Kubernetes, built to run stateful applications reliably in production on any infrastructure - on-premises, cloud, or hybrid. It provides container-granular, high-performance storage plus capabilities like snapshots, encryption, and policy-based automation. Beyond basic persistence, Portworx delivers zero-RPO and low-RTO disaster recovery (PX-DR), automated capacity management with Autopilot, and intelligent scheduling and data locality via Stork to keep applications close to their data and maintain performance at scale. It also supports modern virtualization by providing persistent storage and live migration for KubeVirt virtual machines, so teams can run VMs and containers side by side on Kubernetes with a common data layer.

→ **Learn more about Everpure at www.everpuredata.com**

About Spectro Cloud PaletteAI

Spectro Cloud PaletteAI is a modern infrastructure management platform for deploying and managing applications, virtual machines, and AI workloads across data centers, clouds, and edge locations. Its declarative Cluster Profiles enable platform teams to define complete infrastructure stacks once and consistently deploy them across distributed environments. PaletteAI provides full lifecycle management, from initial provisioning through ongoing operations, helping organizations standardize infrastructure while supporting diverse hardware, operating systems, storage, networking, and application requirements at scale.

→ **Learn more about Spectro Cloud at www.spectrocloud.com**

Spectro Cloud

Everpure™
formerly Pure Storage