Multi-VRF ELK on Kubernetes with Everpure

Centralized Log Aggregation Across Network Boundaries

Authors: Manish Kumar, Radek Mul, Dhruv Bhatnagar

Table of Contents

Introduction

Organizations managing large-scale log aggregation in **multi-VRF** (Virtual Routing and Forwarding) network environments face a critical challenge: providing centralized services accessible across network boundaries while maintaining security isolation.

This whitepaper presents Everpure's production deployment of the **ELK** (Elasticsearch, Logstash, Kibana) stack on **Baremetal Kubernetes**, featuring an innovative multi-VRF architecture using **Cilium BGP Control Plane V2**. The solution leverages Everpure **FlashArray** for high-performance hot data storage and **FlashBlade** for cost-effective cold storage, orchestrated by Portworx Enterprise.

Key Innovations

- **Multi-VRF BGP**: Single Kubernetes cluster accessible from three isolated VRFs
- **Intelligent Data Tiering**: Hot data on FlashArray (200 TB, <1ms latency), cold data on FlashBlade (1+ PB)
- **Full Automation**: Foreman + PXE provisioning, Kubespray deployment, ArgoCD GitOps

This architecture pattern applies to any centralized service requiring cross-VRF accessibility: monitoring platforms, databases, CI/CD infrastructure, and more.

Business Challenge

Log Aggregation Requirements

Modern enterprises generate massive volumes of log data from applications, infrastructure, and security systems. At Everpure, we needed a log aggregation platform that could support:

- **Ingestion**: 12 TB uncompressed logs and metrics daily (sustained), with resilience to handle 24 TB peak loads
- **Query Performance**: Sub-5 second response times for p95 queries
- **Retention**: 1-3 months of searchable logs, 1 year of archived snapshots
- **Availability**: 99.9% uptime with no data loss
- **Multi-Tenant Access**: Secure access from Development, Production, and Management VRFs
- **Cost Efficiency**: Predictable costs without egress fees, reduced AWS infrastructure footprint

The Multi-VRF Challenge

The critical challenge was providing a single ELK service accessible from three isolated network VRFs:

- **Development VRF**: Engineering teams, development workloads
- **Production VRF**: Production applications, customer-facing services
- **Management VRF**: Infrastructure services, monitoring, logging

The Problem

Traditional Kubernetes deployments exist within a single VRF. Applications in other VRFs cannot access services without complex firewall rules, NAT configurations, or VPN tunnels.

The Requirement

Our ELK cluster needed to accept logs from all three VRFs while maintaining network security boundaries, avoiding complex firewall rules, and providing transparent access.

Solution Architecture

Architecture Overview

Our solution combines three key innovations to address the multi-VRF challenge while delivering enterprise-grade performance:

1. **Multi-VRF BGP Networking**: A single Kubernetes cluster accessible from three isolated VRFs using Cilium BGP Control Plane V2, enabling shared access while maintaining network isolation.
2. **Intelligent Storage Tiering**: Hot data (0-10 days) on Everpure FlashArray for real-time ingestion and queries, cold data (e.g. 14+ days) on FlashBlade for cost-effective long-term retention. Portworx Enterprise orchestrates container-native storage with automated provisioning and lifecycle management.
3. **Full Infrastructure Automation**: Bare metal provisioning via Foreman + PXE boot, Kubernetes deployment via Kubespray, and GitOps-based application management with ArgoCD ensure consistent, repeatable deployments.

Core Technology Stack

Component	Technology	Purpose
Orchestration	Kubernetes 1.31.4	Container orchestration and workload management
Networking	Cilium 1.18.2 with BGP v2	CNI, network policies, BGP-based load balancing
Storage Orchestration	Portworx Enterprise 3.4.1	Container-native storage, dynamic provisioning
Hot Storage	Everpure FlashArray	100 TB, 500K+ IOPS, <1ms latency
Cold Storage	Everpure FlashBlade	1+ PB, S3-compatible, zero egress fees
GitOps	ArgoCD	Declarative application deployment
Centralized logging	Elasticsearch, Logstash, Kibana	Distributed search and analytics engine with UI for logs exploration and dashboard creation
Observability	Prometheus + Thanos	Metrics collection and long-term retention

Table 1. Core Tech Stack

Data Flow Architecture

Ingestion Architecture

Logs flow through a multi-stage pipeline:

1. **Log collectors** - Elastic Agents (deployed on ~10,000 hosts) send logs and metrics into ELK kubernetes clusters from other systems within the company network. This is centrally governed in Elastic Fleet to allow seamless policy upgrades.
2. **Logstash** (running in Kubernetes) parses and enriches logs. It also ensures no data is lost via the connection to a FlashArray-based Persistent Queue. The logs are further distributed into Elasticsearch hot nodes via kubernetes service.
3. **Elasticsearch** (running in Kubernetes) indexes and stores logs on Everpure FlashArray.
4. After reaching a certain time threshold, the logs are transferred to the cold storage on Everpure FlashBlade.

Multi-VRF Kubernetes with BGP

The Solution: Cilium BGP Control Plane V2

Our Multi-VRF architecture uses **Cilium BGP Control Plane V2** to advertise Kubernetes services into multiple VRFs simultaneously via BGP addressing. This is the key innovation that enables a single Kubernetes cluster to serve multiple isolated networks.

How It Works:

1. **BGP Peering**: Each Kubernetes control plane node establishes BGP peering sessions with top-of-rack (ToR) switches
2. **VRF Isolation**: Each VRF maintains its own routing table, traffic stays within security boundaries
3. **ECMP Load Balancing**: Traffic is distributed across all control plane nodes using Equal-Cost Multi-Path routing

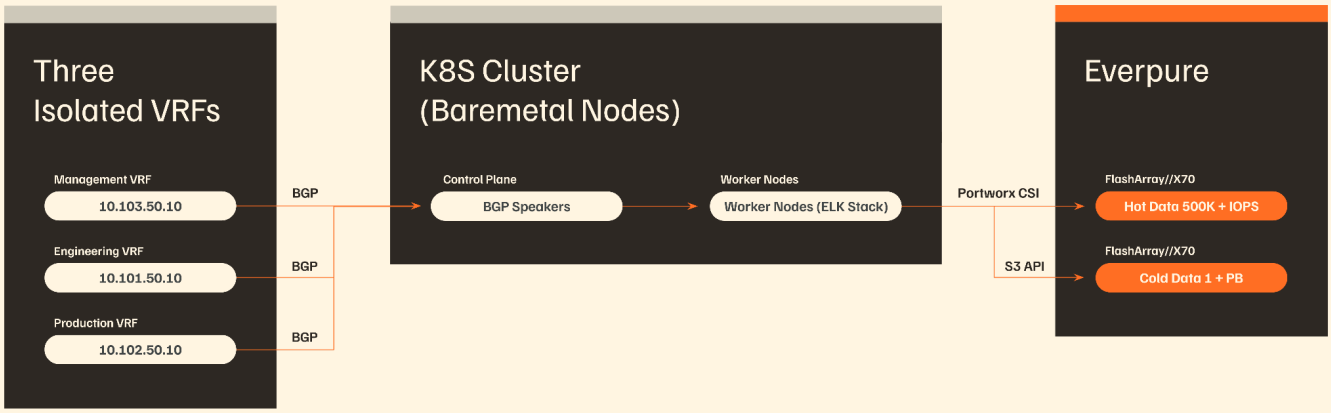

Figure 1: Multi-VRF Kubernetes Architecture with BGP

Infrastructure Specifications & Cluster Provisioning

Bare Metal Cluster Specifications

Our production ELK cluster leverages high-density bare metal infrastructure to eliminate virtualization overhead and maximize performance.

Implementation Highlights:

Component	Bare Metal	AWS
Worker Nodes	6 (Each with 48 Physical Cores / 96 Threads and 768 GB RAM)	6× r8g.8xlarge (32 vCPU, 256 GB RAM) 6× m8g.12xlarge (48 vCPU, 192 GB RAM) 6× m8g.4xlarge (16 vCPU, 64 GB RAM)
CPU Cores	**288** (Physical) **576** (vCPU Threads)	**576** vCPU
Memory	**4.6 TB** (4,608 GB)	**3.07 TB** (3,072 GB)
Storage Type	FlashArray//X90 R4 + FlashBlade//S500 X7	EBS io2 (8K IOPS) + EBS gp3 + S3
Latency	< 1ms	< 10 ms (gp3)
Consistency	24/7	Burst credits (gp3)
Hot storage capacity	**268 TB** (raw), 600+ TB (effective with deduplication)	20TB (io2) + 60TB (gp3) = **80 TB**
Cold storage capacity	**1 PB +**	**118 TB** (gp3) + **1PB** (S3)

Table 2. Baremetal vs AWS specification

Cluster Provisioning Workflow: From Bare Metal to Kubernetes

To maintain consistent and repeatable infrastructure, we utilize a fully automated provisioning pipeline that eliminates manual overhead:

1. **OS Provisioning**: **Foreman + PXE Boot** automates the installation of the base operating system on new bare metal nodes for an identical, hardened configuration.
2. **Cluster Orchestration**: **Kubespray** (Ansible-based) deploys the Kubernetes control plane and worker nodes, ensuring a "Vanilla Kubernetes" installation with high availability and integration with Cilium and Portworx.
3. **Storage Configuration**: Local disk management is achieved by mounting a **Dedicated Runtime Disk** (100GB+) for /data/containerd/ to prevent DiskPressure evictions during heavy log ingestion.
4. **Infrastructure as Code**: **GitOps** manages all infrastructure configuration, using **ArgoCD** to deploy applications from Git repositories, ensuring declarative configuration and full reproducibility.

Bare Metal Performance Advantages

Beyond Everpure performance, bare metal infrastructure enables optimizations impossible in cloud environments:

1. **CPU Governor Optimization**: Setting the CPU governor to "performance" mode prevented frequency scaling, which reduced the average p95 read latency by **33%** (from 4.2s to 2.8s) and p99 read latency **47%** (from 8.5s → 4.5s)
   (AWS limitation: Cannot control CPU governor on EC2 instances)
2. **Kernel Parameter Tuning**: Full control allows optimizing parameters like `vm.swappiness`, `vm.max_map_count`, and network stack settings for better performance. (AWS limits kernel parameter access).
3. **Network Bandwidth**: Provides 100 Gbps per node (4x25 Gbps bonded) compared to ~12.5 Gbps on comparable AWS instances, enabling faster log ingestion (12 TB/day sustained), quicker queries with large result sets, and faster snapshots to FlashBlade.
4. **Storage performance**: Direct access to non-listed types of hardware, like Everpure FlashArray
5. **Predictable IOPS**: No latency spikes during peak hours or when cloud burst credits are exhausted.

Storage Architecture & Data Tiering

Storage performance is the foundation of any high-performance Elasticsearch deployment. Elasticsearch is fundamentally a storage-intensive application that performs random reads (searching across indices), sequential writes (ingesting logs), merge operations (background optimization), and snapshot operations (backing up to cold storage).

Our solution leverages three Everpure products (Flash Array, Flash Blade, Portworx) to deliver enterprise-grade performance and efficiency.

FlashArray for Hot Data

Elasticsearch hot and warm indices store freshly ingested data (0-14 days retention) requiring the highest performance for real-time log ingestion, interactive queries from concurrent users, background merge operations, and index optimization tasks.

Four Key Advantages of FlashArray//X90 R4:

- **Sub-millisecond Latency**: <1ms average read/write latency with consistent performance under load, critical for interactive query response times. Unlike EBS, FlashArray latency remains consistent - no spikes during peak hours or burst credit exhaustion.
- **Massive IOPS Capacity**: 500K+ IOPS sustained with no burst credit limitations, handling concurrent ingestion, queries, and merges simultaneously.
- **No Performance Throttling**: EBS gp3 uses burst credits that deplete under sustained load. FlashArray delivers consistent 24/7 performance with no degradation during peak hours, ensuring predictable query times.
- **Storage Efficiency**: The Elasticsearch already compresses the data inherently, FlashArray adds another layer of inline deduplication and compression. It delivers another **23%** data reduction on log data with no performance impact, reducing physical storage requirements, power consumption, and rack space.

FlashBlade for Cold Storage

For compliance and long-term retention of up to one year in backup storage, Elasticsearch cold and frozen indices (logs older than 10 days) are stored on Everpure FlashBlade. While this data is accessed less frequently, it must be quickly restorable.

S3-Compatible Snapshot Repository: FlashBlade provides native S3 compatibility for Elasticsearch snapshots with:

- Native Elasticsearch S3 integration
- Fast snapshots and restores
- Zero egress fees ($0 to read archived data vs. AWS S3 egress charges)

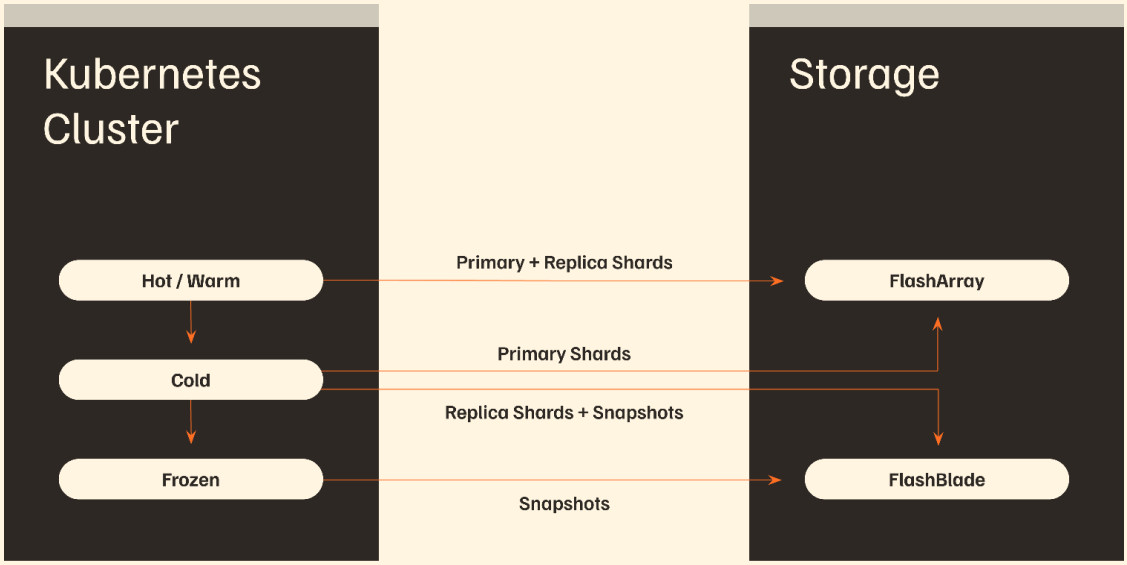

Figure 2. Three-tier storage architecture: Hot, warm, cold and frozen tiers in ELK

Portworx Storage Orchestration

Managing storage for Kubernetes workloads requires dynamic volume provisioning, automated failover and recovery, storage class abstraction, and integration with Everpure arrays.

Portworx Integration with Everpure:

Portworx acts as the Kubernetes CSI (Container Storage Interface) driver, providing:

1. **Dynamic Provisioning**: Elasticsearch pods request storage via PersistentVolumeClaims. Portworx automatically provisions volumes on FlashArray using **FADA** (FlashArray Direct Access) with no manual LUN creation or mapping required.
2. **Automated Tiering**: Elasticsearch ILM policies trigger storage class changes. Portworx migrates data between storage tiers transparently to applications.

Operational Benefits:

- **Self-service storage:** Developers request storage via Kubernetes manifests
- **Automated operations:** No manual storage provisioning
- **Consistent interface:** Same API for all storage tiers
- **Cloud-native:** Integrates with Kubernetes ecosystem (Helm, operators, ECK)

ELK Stack Implementation & GitOps

ELK Stack Deployment via ECK

The Elastic Stack is deployed using **ECK** (Elastic Cloud on Kubernetes), the official Kubernetes operator from Elastic. ECK provides declarative management of Elasticsearch, Logstash, and Kibana, ensuring portability and consistency across environments.

GitOps with ArgoCD

ArgoCD is used to manage both the storage layer (Portworx) and the application layer (ELK stack) via GitOps, bridging bare metal infrastructure and application agility.

- **Deployment**: Deployed in HA mode on bare metal, with the UI exposed by Cilium Ingress/LB Controllers. All applications are managed via declarative Git repositories.
- **Hierarchical Management**: Infrastructure (Portworx addon, managed by Platform Admin) and applications (ELK Stack, managed by Service teams) are treated as separate "Addons" for clear separation of concerns.
- **Key Benefits**: Ensures **declarative configuration**, **automated deployment** (Git changes trigger auto-deployments), and **full reproducibility** of the entire stack.

Performance Results & Business Benefits

Production Performance Metrics

The bare metal ELK deployment on Everpure infrastructure delivers exceptional performance across three key areas:

- **Ingestion Performance**: Achieves 12 TB/day sustained log ingestion (24 TB peak), with 500K+ IOPS sustained and <1ms average latency. This represents up to a 166x speed advantage over comparable AWS EBS storage.
- **Query Performance**: Provides Sub-5 second p95 query response times for interactive searches and consistent 24/7 performance with no cloud throttling. Optimization of the CPU governor led to a 33% reduction in query latency.
- **Storage Efficiency**: Utilizes 100 TB of hot storage on FlashArray (0-14 days retention) and 1+ PB of cold storage on FlashBlade (1+ year retention), offering zero egress fees for archived data access and inline deduplication/compression.

Business Benefits

- **Cost Efficiency**: Eliminated cloud egress fees (significant for 12 TB/day ingestion) and leveraged existing CapEx investments (FlashArray and FlashBlade) for predictable operational costs.
- **Performance & Reliability**: Achieved a **99.9% uptime SLA** with 3-way synchronous replication and **consistent sub-millisecond latency**.
- **Operational Excellence**: Ensured full infrastructure automation (Foreman + Kubespray + ArgoCD), GitOps-based deployment for reproducibility, and self-service storage via Portworx CSI, alongside comprehensive observability (Prometheus + Thanos + Grafana).
- **Security & Control**: Maintained log security within trusted data center boundaries, enforced **Multi-VRF isolation** while providing shared access, and retained full control over the network stack and kernel parameters.

Cost

The Cost analysis demonstrates the significant financial advantage of the bare metal deployment. By leveraging existing infrastructure and Everpure arrays, the solution provides **over 3 times** the hot storage capacity and comparable long-term cold storage capacity.

Bare Metal	AWS	Return In
$45k FA + $100k FB + $72k Compute (one time)	**$70k monthly**	4 Months

Table 3. Baremetal vs AWS Cost Comparison

Furthermore, the total Bare Metal CAPEX investment of approximately $217,000 is recouped in just 4 months when measured against the previous cloud OPEX cost of **$70,000 monthly**, offering a rapid and substantial return on investment.

Performance

The speed of data recovery is paramount during a critical incident or for rapid data-set provisioning. The combination of Elasticsearch snapshot capabilities and FlashBlade's massively parallel S3 architecture provides a significant advantage in restoring speed.

For a direct performance comparison, the testing utilized a 0.5 TB data set, spread across 10 Elasticsearch shards. The environment was standardized on 2 x 64 GB memory Elasticsearch nodes with no CPU throttling, ensuring an equivalent baseline for both the Bare Metal and AWS metrics.

Metric	Bare Metal avg restore time	AWS avg restore time	Improvement
0.5 TB snapshot restoration	39 minutes	51 minutes	23%

Table 4. Snapshot restore speed comparison

A high-performance log aggregation platform requires top-tier performance across two critical vectors: ultra-low-latency read/write access for real-time data, and rapid data recovery speed for business continuity. The following comparison quantifies the performance advantages of our Bare Metal deployment—leveraging the consistent I/O of FlashArray.

Metric	Bare Metal server latency	AWS server latency	Times Faster
Average on 0.5 TB data	676.1 ms	2060.0 ms	3.05x
P70 on 0.5 TB data	818.0 ms	3090.0 ms	3.75x
P80 on 0.5 TB data	1334.0 ms	3661.0 ms	2.74x
P90 on 0.5 TB data	1646.4 ms	4492.0 ms	2.73x

Table 5. Read speed comparison

The read speed is measured on a full-text word match with aggregation using warm nodes, which are most often utilized for searches. The measurement was taken right after the restart, with caches cleared after each run.

Business Continuity and Disaster Recovery (BCDR)

The solution establishes a comprehensive Business Continuity and Disaster Recovery foundation through Everpure's enterprise-grade data protection capabilities combined with Elasticsearch's native snapshot functionality. This multi-layered approach ensures log aggregation services remain resilient against data loss, corruption, and site-level failures.

The BCDR strategy leverages:

- **Everpure FlashArray ActiveCluster**: Provides 3-way synchronous replication across availability zones with automatic failover for metro-area disaster recovery, delivering 99.99% availability with zero data loss (RPO=0). Array-level snapshots complete in seconds with no performance impact, enabling rapid recovery scenarios.
- **Elasticsearch Native Snapshots**: Application-consistent backups of indices, cluster state, and configuration are stored on FlashBlade S3-compatible storage. Snapshots are incremental and space-efficient, enabling restoration to the same or different clusters for disaster recovery scenarios.
- **FlashBlade Cold Storage**: Long-term retention (up to 1 year) with fast restore capabilities and no egress fees provides cost-effective compliance and historical data recovery.

This architecture supports multiple recovery scenarios including index-level restoration, full cluster recovery, site-level failover via FlashArray ActiveCluster, and point-in-time recovery for data corruption events.

Conclusion

This whitepaper presented Everpure's production deployment of the ELK stack on bare metal Kubernetes, showcasing three key innovations:

1. **Multi-VRF BGP Networking**
2. **Intelligent Storage Tiering**
3. **Full Infrastructure Automation**

The deployment of ELK on bare metal Kubernetes, powered by Everpure, demonstrates that enterprises do not need to choose between the flexibility of the cloud and the performance of on-premises hardware. By combining **Cilium BGP** for networking agility with **FlashArray** and **Portworx** for storage performance, organizations can build centralized services that are faster, more secure, and more cost-efficient than their public cloud counterparts.

References

[1] Cilium BGP Control Plane V2 Documentation:
https://docs.cilium.io/en/stable/network/bgp-control-plane/bgp-control-plane/

[2] Elasticsearch Index Lifecycle Management (ILM):
https://www.elastic.co/guide/en/elasticsearch/reference/current/index-lifecycle-management.html

[3] AWS EBS Pricing:
https://aws.amazon.com/ebs/pricing/ *(accessed January 2026).*

[4] AWS S3 Data Transfer Pricing:
https://aws.amazon.com/s3/pricing/ *(accessed January 2026).*

[5] Portworx Enterprise with Everpure FlashArray:
https://docs.portworx.com/portworx-enterprise/platform

[6] Elastic Cloud on Kubernetes (ECK):
https://www.elastic.co/guide/en/cloud-on-k8s/current/index.html