Solution Brief

Everpure and Red Hat for AI Factory Training

A scalable reference design for event-driven fine-tuning on Red Hat OpenShift AI and FlashBlade, from pilot to AI factory

Everpure™ and Red Hat have validated an AI training factory that automates the path from new data to a production-ready model. A new dataset landing in object storage triggers the fine-tuning of a 32-billion-parameter model across a distributed GPU cluster on Red Hat OpenShift AI, with Everpure FlashBlade//S™ serving every stage: the trigger inbox, the training data, a 393GB sharded checkpoint, the evaluation evidence, and an immutable record of the model that reaches production. No one launches the job, and no GPU stays allocated after it finishes.

This is a reference design, not a fixed configuration. It runs on the NVIDIA GPUs enterprises deploy for this software stack—GB300/GB200 rack-scale systems, B300/B200, RTX PRO, and L40S—with GPU count driven by the size of the model and whether the job is fine-tuning or full training. FlashBlade//S carries the design from initial pilots through NVIDIA DGX SuperPOD™ and AI factory deployments in the 1,000- to 2,000-GPU range, and FlashBlade//EXA™ extends it further. The platform stays performant, reliable, and well-orchestrated at every step of that growth curve.

GPU clusters are only as fast as their data

AI training pushes storage in two directions at once. During data loading, thousands of workers issue massively parallel small reads; at every checkpoint, the cluster turns around and writes terabytes in saturating sequential bursts. Legacy network-attached storage (NAS) cannot feed modern accelerators at these rates, and parallel file systems close the gap only by adding proprietary clients, constant tuning, and specialist administration.

The economics are unforgiving. Modern AI infrastructure targets GPU utilization above 90%, and a $5 million GPU investment can lose up to $44,000 per year in return for every percentage point it falls below that mark.[1] Idle time between runs compounds the problem: in most shops, a training job is still launched by hand, and the cluster it ran on stays allocated long after the loss curve has flattened.

Event-driven training

New data triggers fine-tuning automatically. GPUs are allocated for the run and released the moment it ends.

Checkpoints that hold up

All ranks write to one FlashBlade® namespace; SafeMode™ Snapshots and S3 keep every checkpoint recoverable and auditable.

Certified for AI factories

NVIDIA-certified storage scales from your first training cluster to 10+ TB/s AI factories on standard Ethernet and NFS.

The solution: An autonomous training pipeline

The validated implementation runs on bare-metal Red Hat OpenShift 4.18, with Red Hat OpenShift AI providing the distributed workload stack: KubeRay for cluster life cycle, Ray Train for orchestration, and Kueue for admission control. DeepSpeed ZeRO-3 shards a Qwen3-32B model across a cluster of single-GPU workers. FlashBlade//S500 provides both storage planes—S3 for objects and events and Network File System (NFS) through Portworx® FlashBlade® Direct Access for the shared checkpoint file system—so every artifact of the run lives on one array.

Nothing in the design is tied to a particular accelerator. It supports the GPUs enterprises standardize on for training and fine-tuning—NVIDIA GB300/GB200, B300/B200, RTX PRO, and L40S—and the worker count, per-worker sizing, and Kueue quotas are set by the model and job. Fine-tuning a mid-size model may need a handful of workers; full training of a large model needs many more. Larger jobs mean more ranks and larger checkpoint shards, which is precisely the traffic the FlashBlade data plane is built to absorb.

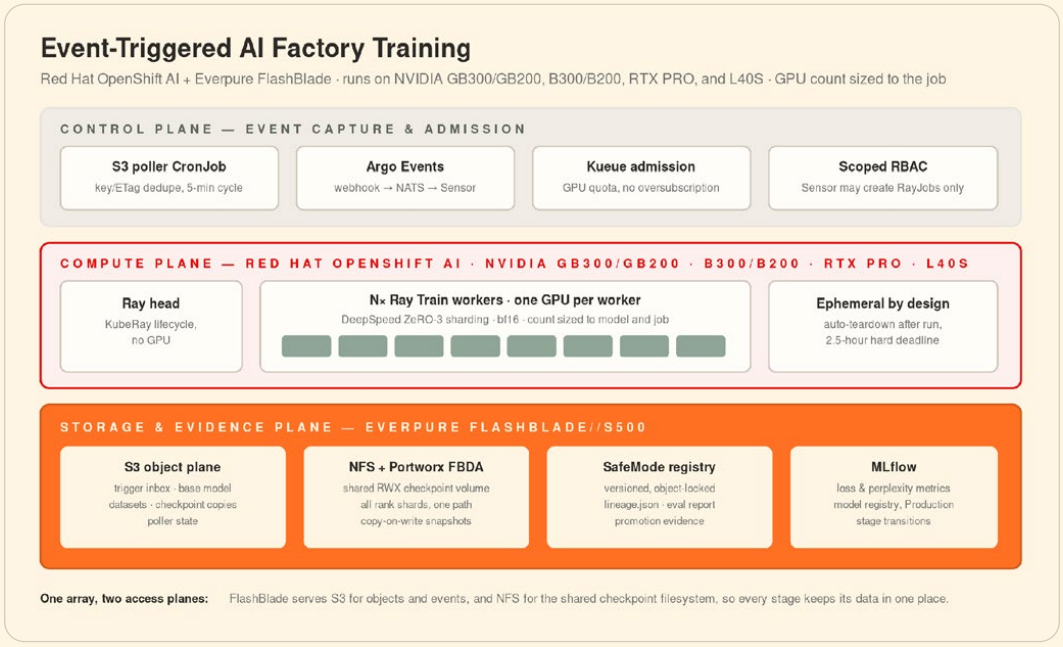


FIGURE 1 The three planes of the validated architecture: Argo Events and Kueue control when GPUs are consumed, Red Hat OpenShift AI runs the distributed training, and FlashBlade holds the data and evidence.

Key capabilities

- **Event-driven triggering**: A new object in the S3 inbox starts the pipeline; a poller deduplicates by key and ETag and posts to Argo Events.
- **Quota-aware admission**: Kueue admits a job only when its full GPU quota is free, so autonomous triggers queue instead of oversubscribing the cluster.
- **Ephemeral distributed training**: Ray Train and DeepSpeed ZeRO-3 fine-tune the 32-billion-parameter model in BF16 with CPU offload; the cluster tears itself down when the job ends.
- **Shared checkpointing**: Every rank writes to one FlashBlade-backed ReadWriteMany (RWX) volume, and then the complete tree is copied to S3.
- **Evidence-based promotion**: A held-out evaluation and a GPU-free threshold gate decide whether the model reaches the registry.
- **Immutable lineage**: SafeMode™-enabled S3 retains the trigger identity, metrics, and promotion evidence under Object Lock.

Deployment and integration

- Runs on the Red Hat OpenShift AI DataScienceCluster; the KubeRay and Kueue integration is the supported path for distributed workloads
- Uses standard Ethernet, NFS, and S3 throughout, with no proprietary storage clients or parallel file system to tune
- Exposes FlashBlade NFS as Kubernetes-native RWX volumes with Container Storage Interface (CSI) snapshots through Portworx FlashBlade Direct Access
- Leverages FlashBlade, NVIDIA-certified enterprise storage, from first training cluster to DGX SuperPOD-scale AI factories
- Delivers SLA-backed storage as a service through Evergreen//One™, sized to the bandwidth of your GPUs

How it works: From object to production model

The pipeline begins with no GPUs allocated and ends the same way. Between those two states, it moves through five phases, each with an observable outcome.



FIGURE 2 The pipeline life cycle. GPUs are claimed at admission and released at teardown; the evaluation, gate, and registration steps run as separate jobs.

Phase	What happens
Trigger	A file lands in s3://.../incoming/ on FlashBlade. The poller compares keys and ETags against the state kept in the same bucket and posts new objects to Argo Events.
Admit	The sensor creates a RayJob with the trigger identity in its annotations. Kueue checks the job's full GPU, CPU, and memory request against quota and queues it if capacity is busy.
Train	KubeRay provisions a head pod and the requested set of one-GPU workers. Each rank stages the base model from S3, initializes ZeRO-3 sharding, and trains in BF16 with CPU offload. Rank 0 streams loss to MLflow.
Checkpoint	Every rank writes its ~49GB shard into the shared FlashBlade Direct Access volume; rank 0 uploads the complete ~393GB tree to S3. A copy-on-write snapshot taken mid-run provides the rollback path. The RayCluster is then deleted automatically.
Evaluate, gate, and register	A separate job measures perplexity on 64 held-out samples. A GPU-free gate compares it to a threshold and writes a promotion marker only on pass. Registration stores lineage in SafeMode-enabled S3 and promotes the MLflow model version to production.

TABLE 1 The five operational phases of the validated pipeline.

Note: The implementation is a validated proof of concept at pilot scale, not a general-purpose orchestration platform. Evaluation, gating, and registration run as separate jobs rather than a declarative, multi-stage directed acyclic graph.

Validated results

The capstone run produced real numbers, not projections. Both branches of the promotion gate were exercised against the same measured metric.

Validation area	Observed result
Model and hardware	Qwen3-32B fine-tuned across a distributed multi-GPU cluster, with one DeepSpeed rank per GPU.
Training signal	Loss fell from 2.03 to 0.67, a reduction of roughly 66%.
Checkpoint	Eight ZeRO-3 shards of ~49GB each, ~393GB total, were written to shared FlashBlade storage and copied to S3.
Held-out evaluation	Perplexity measured 2.0032 on 64 unseen samples, which was recorded in eval-report.json and MLflow.
Promotion gate	The model passed at a 2.5 threshold but failed and blocked registration at a 1.5 threshold against the same metric.
Registry	qwen3-fbda-finetune version 1 was promoted to production with immutable S3 lineage.
Idle state	Zero training GPUs were allocated between runs; poller and queues stayed active.

TABLE 2 Capstone evidence from the validated run.

The engineering lessons behind these results are documented as well, including one worth highlighting: an early rank-0-only checkpoint upload silently omitted every shard but its own, and the fix was collection on shared storage before upload. That is a storage lesson as much as a software one.

Why FlashBlade is the data plane for AI training

FlashBlade provides the data services that keep workers fed, checkpoints consistent, and model lineage intact.

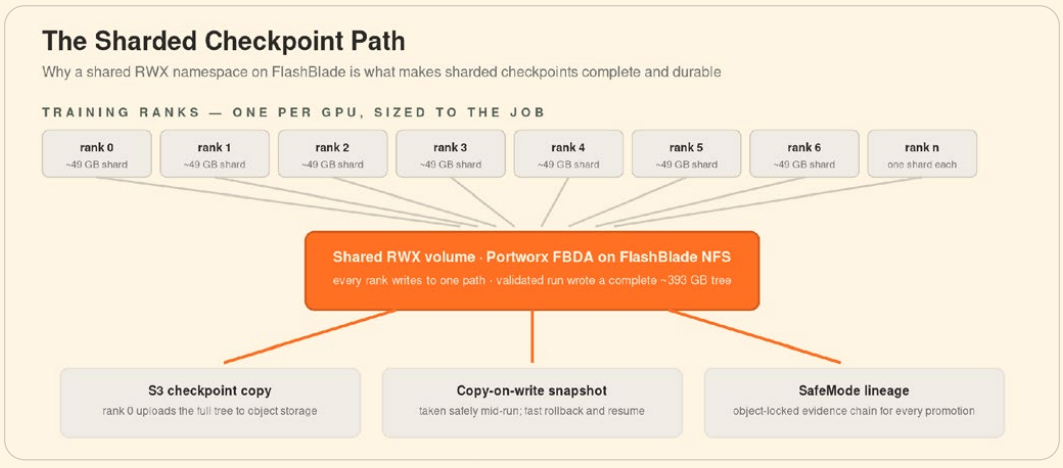


FIGURE 3 The checkpoint path. A shared namespace is what makes a rank-sharded checkpoint complete; SafeMode Snapshots make it durable and auditable.

- **One array, two access planes**: FlashBlade serves S3 for the trigger inbox, base model, datasets, and lineage, and NFS for the shared checkpoint file system. There are no data copies between separate object and file systems.
- **Massively parallel data loading**: A distributed metadata core sustains the small-file and metadata rates of modern training datasets while workers load in parallel.
- **Line-rate checkpointing**: Writes go directly to NVMe data nodes, so sharded checkpoints complete at full storage bandwidth and training loops stay short.
- **A shared RWX namespace that makes checkpoints whole**: Every rank writes into one directory tree. The validated run proved the failure mode this prevents: without a common namespace, the uploaded checkpoint was silently incomplete.
- **Snapshots that are safe under load**: A copy-on-write snapshot was taken while every rank was actively writing, giving a proven rollback and resume path without pausing training.
- **Governance built into the storage**: SafeMode versioning and Object Lock preserve lineage, evaluation, and promotion evidence even if the tracking service is lost.
- **Energy where it belongs**: FlashBlade delivers up to 90% more performance per watt than competing solutions, allowing more data center power to be dedicated to GPUs rather than storage.

A reference design that scales

The pattern validated at pilot scale is the same one Everpure deploys at AI factory scale because none of its load-bearing pieces care about cluster size. Kueue quotas grow with the estate. Ephemeral RayClusters keep accelerators released between runs whether the job requests eight accelerators or 800. And the storage traffic that grows fastest with scale—parallel data loading and rank-sharded checkpoint bursts—is what the FlashBlade family is certified against, including NVIDIA DGX SuperPOD with GB200 and GB300 systems.

Deployment scale	Storage platform	What it looks like
Pilot and first clusters	FlashBlade//S	The validated design in this document: one array for both storage planes, standard Ethernet and NFS/S3
Enterprise AI factory and SuperPOD (up to ~2,000 GPUs)	FlashBlade//S	Scale-out chassis growth keeps pace with GPU expansion; same namespace and operating model, with no rearchitecture
Frontier-scale AI factory (beyond 2,000 GPUs)	FlashBlade//EXA	1TB/s to 10+ TB/s in a single namespace, with S3 over Remote Direct Memory Access (RDMA) to eliminate copies between tiers

TABLE 3 One design across three scales. Teams size the GPU count to the model and the job; the orchestration pattern, data layout, and governance model carry through unchanged.

Competitive advantage for enterprise AI

The same architecture can scale from a first cluster to an AI factory, but alternative approaches introduce different trade-offs as environments grow. Table 4 highlights where common approaches fall short for enterprise training.

Alternative	Where it falls short for training
Parallel file systems	Close the performance gap only with proprietary clients, constant tuning, and specialist administrators. FlashBlade delivers comparable feeding rates on standard Ethernet and NFS.
Legacy and scale-out NAS	Cannot sustain massively parallel small reads and saturating checkpoint bursts at the same time; GPUs wait on both ends of the loop.
Local NVMe/hyperconverged infrastructure	Per-host capacity is stranded, and rank-sharded checkpoints have no common namespace—the exact failure mode this implementation had to engineer around.
Cloud object-only pipelines	No POSIX RWX path for distributed checkpoint collection, and egress plus per-token economics penalize steady, repeated training runs.

TABLE 4 How the common alternatives compare for enterprise training workloads.

The performance claims have public reference points. FlashBlade set an industry record on the SPECstorage Solution 2020_ai_image benchmark with 6,300 AI jobs, 26% above the previous high. In internal MLPerf Storage v2.0 testing (not verified by MLCommons), FlashBlade//EXA delivered roughly twice the performance of the nearest competitor on 3D U-Net while sustaining GPU utilization above 90% on large NVIDIA Hopper™ clusters. FlashBlade is certified for NVIDIA DGX SuperPOD—including GB200 and GB300 systems—NVIDIA DGX BasePOD™, and NVIDIA OVX™. In addition, Everpure is aligning FlashBlade//EXA with the NVIDIA Vera Rubin reference architecture for large-scale AI factories.

Business outcomes

Speed and performance

Training starts when data arrives, not when someone remembers to launch it. The five-minute detection cycle, quota-aware admission, and automatic teardown compress the gap between new data and a registered model to hours, while line-rate checkpointing keeps the training loop itself short.

Cost efficiency

GPUs are allocated for the run and released at the end of it—the validated steady state is zero training GPUs. Kueue prevents autonomous triggers from oversubscribing capacity, and with Evergreen//One, the storage is consumed as an SLA-backed service sized to GPU bandwidth.

Reduced complexity

One storage platform carries the pipeline end to end: trigger, data, checkpoints, snapshots, and audit evidence. There is no parallel file system to administer, no separate object store to reconcile, and the Evergreen® Architecture upgrades the array without pausing the pipelines that depend on it.

Critical capabilities

The solution is defined by five core characteristics:

- **Autonomous**: Event-triggered training with automatic GPU teardown, validated end to end
- **Governed**: Quota admission, scoped role-based access control, a real promotion gate, and object-locked lineage
- **Performant**: Parallel data loading and line-rate checkpointing on standard Ethernet and NFS
- **Durable**: Shared RWX checkpoints, snapshots proven safe under load, and S3 copies
- **Scalable**: One reference design from a pilot cluster to ~2,000 GPUs on FlashBlade//S, and more on FlashBlade//EXA

Together, these capabilities transform experimental AI projects into a scalable, governed AI training factory, from initial pilots to full AI factory scale.

Additional resources

- Learn about Red Hat OpenShift AI.
- Learn about Everpure and NVIDIA.
- Learn more about FlashBlade.
- Learn more about Evergreen//One.

Explore Everpure AI Solutions

1 | Based on an assessment conducted by Everpure

Visit Our Website **800.379.PURE**

©2026 Everpure, the Everpure P Logo, Pure Storage, Evergreen, Evergreen//One, FlashBlade, FlashBlade//EXA, FlashBlade//S, Portworx, SafeMode Snapshots, and the marks in the Everpure Trademark List are trademarks or registered trademarks of Everpure, Inc. or its licensed subsidiaries in the U.S. and/or other countries. The Trademark List can be found at everpuredata.com/trademarks. Other names may be trademarks of their respective owners.

ID54908-01-en-09/26