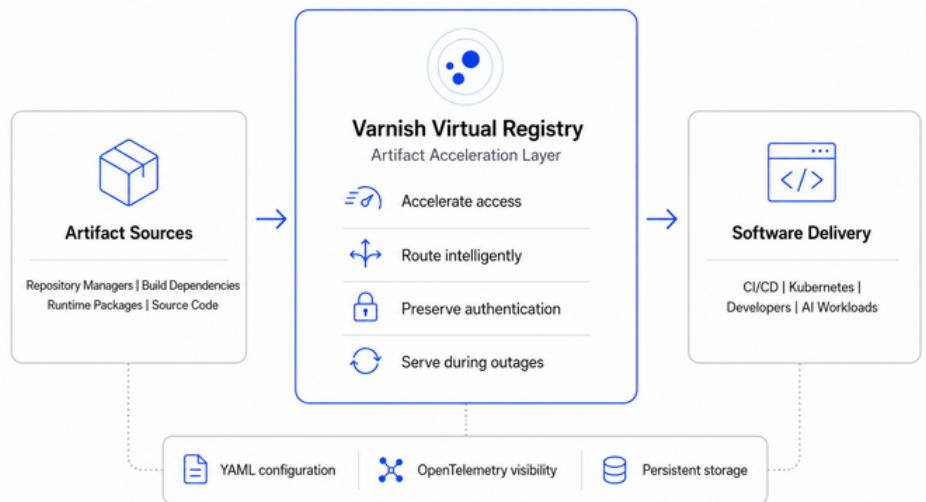


Varnish Virtual Registry

Artifact Delivery Acceleration for Modern Software Infrastructure

Varnish Virtual Registry, also known as Orca, is a high-performance caching and routing layer that accelerates access to containers, packages, Git LFS objects, models, and other software artifacts across CI/CD, Kubernetes, developer, and AI workloads. It preserves existing authentication and access controls while **reducing build delays, infrastructure costs, egress, and dependency on overloaded and expensive repository platforms.**



Artifact Delivery for Machine-to-Machine Scale

Modern software delivery no longer runs at human speed. CI/CD pipelines, Kubernetes clusters, AI agents, GPU workloads, build farms, and distributed developer environments continuously pull dependencies, containers, source artifacts, and model assets from registries and object stores, creating latency, backend saturation, cloud egress costs, and new points of failure. Varnish Virtual Registry moves artifact delivery closer to where artifacts are consumed, reducing dependency on overloaded repositories and upstream platforms.

- **Faster Artifact Delivery:** Caches containers, packages, Git LFS objects, models, and other artifacts closer to developers, CI/CD runners, Kubernetes clusters, and AI workloads.
- **Lower Infrastructure Cost:** Reduces repeated requests to Artifactory, Nexus, cloud registries, and S3-backed stores. Customer environments have seen **70%–99% fewer backend requests**, **40%–60% lower origin CPU**, and **50%–80% network or egress savings**.
- **Resilient, Vendor-Neutral Access:** Keeps serving cached artifacts when registries, object stores, or networks are slow, overloaded, unavailable, or being migrated, without forcing teams to replace existing repositories or center artifact delivery on one vendor.

Key Features & Capabilities

1. Artifact-Aware Acceleration

Varnish Virtual Registry is built for software artifact traffic, not generic web caching. It caches immutable assets such as container layers, package files, binaries, Git LFS objects, and model artifacts while handling metadata, manifests, indexes, and package listings with freshness-aware revalidation.

- **Request Coalescing:** Deduplicates concurrent requests so many pipelines can be served from a single upstream fetch.
- **Persistent Caching:** Keeps frequently used artifacts available across restarts and infrastructure events.

2. Multi-Registry Routing and Virtualization

Varnish Virtual Registry creates a single access layer across existing repositories, registries, object stores, and cloud artifact services.

- **Unified Artifact Access:** Presents multiple artifact sources through one logical endpoint.
- **Intelligent Routing:** Routes requests across registries, mirrors, remotes, and regions using declarative configuration.
- **Failover and Origin Shielding:** Continues serving cached artifacts during origin slowdowns, outages, migrations, or upstream instability.

3. Authentication-Preserving Private Artifact Caching

Varnish Virtual Registry accelerates private artifact traffic without bypassing registry security.

- **Preserves Access Controls:** Validates requests against upstream authentication before serving private cached artifacts.
- **Shared Cache Efficiency:** Allows approved users to benefit from cached private artifacts without duplicating heavy binaries per user.
- **No Pipeline Changes Required:** Works with existing tools, typically a DNS or registry-mirror cutover.

4. Operational Visibility and Deployment Flexibility

Varnish Virtual Registry is designed for enterprise platform teams operating distributed software delivery infrastructure.

- **OpenTelemetry Support:** Exports logs, metrics, and traces for visibility into cache hit rates, latency, traffic volume, and backend load.
- **Simple Configuration:** Uses YAML-based configuration to define registries, routes, remotes, and operating behavior.
- **Flexible Deployment:** Runs as a container, Kubernetes deployment with Helm, or Linux package, and scales horizontally across clusters and regions without shared state or a central database.

Deployment & Ecosystem Support

Varnish Virtual Registry supports Docker/OCI images, Helm charts, npm, PyPI, Maven, NuGet, Cargo, Go modules, Ansible, Conan, Debian, RPM, Git, Git LFS, model artifacts, and other HTTP-delivered immutable assets. It works with public and private sources including JFrog Artifactory, Sonatype Nexus, Cloudsmith, GitHub Packages, GitLab Package Registry, Docker Hub, npm Registry, PyPI, Maven Central, Amazon ECR, Google Artifact Registry, Azure Container Registry, PHP, and S3-backed artifact stores. Additional formats and model artifacts on near-term roadmap.

It can also be paired with Varnish Artifact Firewall so the same artifact access layer can accelerate delivery while enforcing package policy at request time.