RAPIDFORT

RapidFort Kimia: The Secure Successor to Kaniko

A Drop-In, Security-Enhanced Container Builder for Modern Cloud Environments

Executive Summary

With Kaniko now sunset and no longer actively maintained, organizations that depend on secure, daemonless container builds face a growing challenge: how to continue creating images safely, reproducibly, and in compliance with modern security standards. Kimia, an open-source project developed by RapidFort in collaboration with SOSi, is engineered as the natural successor to Kaniko—providing 100% backward compatibility while delivering significant advances in security, reproducibility, and enterprise readiness.

Kimia preserves everything that made Kaniko valuable—daemonless builds, Kubernetes-native workflows, and easy adoption—while introducing a modern architecture built for today's security-conscious environments. With true rootless operation, user namespace isolation, minimal Linux capabilities, reproducible builds, and real-time SBOM and provenance generation, Kimia transforms the container build pipeline from a basic image builder into a secure, verifiable, and fully auditable part of the software supply chain.

Kimia offers seamless migration for existing Kaniko users, requiring only a one-line image update and minimal Kubernetes security context updates. The result is a next-generation, actively maintained tool that provides teams with a hardened, compliant, and efficient container build process, designed for modern cloud-native infrastructure.

RF Kimia is available now on GitHub under an open-source license at:
https://github.com/rapidfort/kimia

The Problem: Kaniko Sunset and the Emerging Security Gap

Kaniko revolutionized Kubernetes-native image building by eliminating the need for privileged Docker daemons. However, its deprecation has created a significant operational and security gap. As threats increase and compliance mandates tighten, organizations cannot rely on an unmaintained builder—especially one that still runs as root and lacks modern supply chain protections.

Key challenges with Kaniko today include:

- **Root execution** (UID 0), increasing container escape risk
- **No user namespace isolation**, limiting defense-in-depth
- **No integrated SBOM or provenance**
- **Limited Dockerfile compatibility**, especially involving chown or complex ownership rules
- **Pod Security Standards: “Baseline” only**

Modern DevSecOps teams need more than image assembly—they need **secure, attested, verifiable builds** that reduce risk and meet compliance requirements without slowing down developers. Kimia was engineered precisely to meet this need.

Kimia: A Secure, Drop-In Replacement for Kaniko

Kimia provides a one-to-one functional replacement for Kaniko while introducing a hardened execution model and a more advanced security architecture. It requires only minimal changes:

Kaniko → Kimia Migration Example:

image: gcr.io/kaniko-project/executor:latest

Becomes:

image: ghcr.io/rapidfort/kimia:latest

All major Kaniko arguments work exactly the same in Kimia—100% compatibility across core flags, including **--context, --dockerfile, --destination, --build-arg, --cache, --cache-dir, --insecure, --skip-tls-verify**, and more. Even Git context builds are compatible, and features Kaniko lacked, such as reproducible output, are now native.

Kimia’s argument compatibility ensures organizations can migrate in minutes, not weeks, with no disruption to existing CI/CD pipelines.

A Modern, Hardened Security Architecture

Kimia's design reflects contemporary security requirements and the lessons learned from years of using Kaniko in production.

True Rootless Operation

Kimia runs as UID 1000, never as root. This significantly reduces the blast radius of container escapes and enables safe use in regulated and multi-tenant environments.

User Namespace Isolation

Unlike Kaniko, Kimia leverages user namespaces to map container UID 0 to host UID 1000—adding another mandatory security boundary and making privilege escalation nearly impossible.

Minimal Linux Capabilities

Kimia requires only:

- **SETUID**
- **SETGID**

All other capabilities are dropped. No privileged mode. No SYS_ADMIN. No daemon.

Pod Security Standards: "Restricted" Compliance

Kimia is fully compliant with the strictest Kubernetes security profile. This brings secure builds into alignment with modern cluster hardening requirements and government compliance frameworks.

Reproducible, Verifiable, and Compliant Builds

As supply-chain security becomes a priority across industries, reproducibility and attestations are no longer optional. Kimia includes these capabilities natively.

Reproducible Builds

Kimia supports **bit-for-bit deterministic output**, ensuring that the same source and Dockerfile produce identical images every time—a foundational requirement for SLSA compliance and forensic verification.

SBOM Generation

Kimia generates a real-time **SPDX 2.3 SBOM**, providing a full inventory of packages and dependencies. This enables automated vulnerability scanning and compliance reporting.

Provenance & Attestation

Kimia emits verifiable build metadata, including build time, environment, builder identity, and source context.

Image Signing

Native Sigstore Cosign support ensures images can be cryptographically verified before deployment.

Enterprise-Grade Dockerfile and Platform Compatibility

Kimia expands upon Kaniko's instruction support with:

- Full chown and ownership operations
- Advanced multi-stage build handling
- Support for Docker-specific instructions (HEALTHCHECK, SHELL, STOPSIGNAL) when using Docker output format
- OCI and Docker-compatible build outputs
- Multi-architecture build support (**--custom-platform=linux/arm64**, etc.)

Kimia runs seamlessly in:

- Kubernetes clusters
- Docker environments
- Local CI runners
- Any OCI-compliant registry, public or private

It also supports concurrent builds, isolated via user namespaces.

Performance, Reliability, and Operational Readiness

Kimia is engineered for production-grade workloads:

- **2–5% CPU overhead** relative to Kaniko
- **256MB–2GB RAM** depending on build complexity
- Supports caching (**--cache, --cache-dir**)
- Compatible with overlay or VFS storage drivers
- Designed for parallel execution at scale

Operational issues—such as namespace errors, permission conflicts, and cache directory restrictions—have well-documented resolutions. Kimia also supports Git-based contexts, distroless images, and regulated environment deployment requirements.

When to Choose Kimia

Kimia is the right choice when security, compliance, or complex build requirements matter:

Choose Kimia when you need:

- Defense-in-depth security for production builds
- Compliance with Kubernetes “Restricted” Pod Security Standards
- Reproducibility for supply chain verification
- Attestation, SBOM, and image signing
- Complex Dockerfile compatibility
- Deployment in regulated industries (PCI-DSS, HIPAA, SOC 2, government/defense)

Consider Kaniko only when:

- You cannot enable user namespaces
- You are on a legacy Kubernetes version
- You require zero configuration changes for quick development use cases

For all modern, security-focused environments, Kimia offers a hardened and fully verifiable path forward.

Conclusion

Kimia preserves everything users value in Kaniko—daemonless builds, CI/CD simplicity, Kubernetes-native workflows—while delivering the security, compliance, and reproducibility that modern software supply chains demand.

As the actively maintained, fully compatible successor to Kaniko, Kimia enables organizations to:

- Modernize their container build pipeline
- Adopt secure, rootless, user-namespace–isolated builds
- Generate verifiable, reproducible, and signed images
- Protect against supply chain attacks
- Meet the strictest compliance standards

With Kimia, secure container builds become the default—not an afterthought. It is the future-ready drop-in replacement organizations have been waiting for.

About us

RapidFort, Inc. is a leading software supply chain security company that provides an innovative platform designed to automatically secure container applications and accelerate compliance processes. The company's comprehensive solution addresses critical cybersecurity challenges by removing up to 95% of Common Vulnerabilities and Exposures (CVEs) from container images without requiring any code changes. Visit our website to learn more.

RAPIDFORT