Image provenance

- Why image provenance matters
- How Docker Hardened Images support provenance
  - Attestations
  - Code signing
- Additional resources

Image provenance refers to metadata that traces the origin, authorship, and integrity of a container image. It answers critical questions such as:

- Where did this image come from?
- Who built it?
- Has it been tampered with?

Provenance establishes a chain of custody, helping you verify that the image you're using is the result of a trusted and verifiable build process.

Why image provenance matters

Provenance is foundational to securing your software supply chain. Without it, you risk:

- Running unverified or malicious images
- Failing to meet internal or regulatory compliance requirements
- Losing visibility into the components and workflows that produce your containers

With reliable provenance, you gain:

- Trust: Know that your images are authentic and unchanged.
- Traceability: Understand the full build process and source inputs.
- Auditability: Provide verifiable evidence of compliance and build integrity.

Provenance also supports automated policy enforcement and is a key requirement for frameworks like SLSA (Supply-chain Levels for Software Artifacts).

How Docker Hardened Images support provenance

Docker Hardened Images (DHIs) are designed with built-in provenance to help you adopt secure-by-default practices and meet supply chain security standards.

Attestations

DHIs include attestations—machine-readable metadata that describe how, when, and where the image was built. These are generated using industry standards such as in-toto and align with SLSA provenance.

Attestations allow you to:

- Validate that builds followed the expected steps
- Confirm that inputs and environments meet policy
- Trace the build process across systems and stages

Code signing

Each Docker Hardened Image is cryptographically signed and stored in the registry alongside its digest. These signatures are verifiable proofs of authenticity and are compatible with tools like `cosign`, Docker Scout, and Kubernetes admission controllers.

With image signatures, you can:

- Confirm that the image was published by Docker
- Detect if an image has been modified or republished
- Enforce signature validation in CI/CD or production deployments

Additional resources

- Provenance attestations
- Image signatures
- Attestations overview

□□□ #2
□ □□□ □□□ 2026-10-31 19:15:47 UTC
□ □□□ □□□ 2026-10-31 19:15:48 UTC