Immutable infrastructure

- Why immutability matters
- How Docker Hardened Images support immutability
- Immutable patterns in practice

Immutable infrastructure is a security and operations model where components such as servers, containers, and images are never modified after deployment. Instead of patching or reconfiguring live systems, you replace them entirely with new versions.

When using Docker Hardened Images, immutability is a best practice that reinforces the security posture of your software supply chain.

Why immutability matters

Mutable systems are harder to secure and audit. Live patching or manual updates introduce risks such as:

- Configuration drift
- Untracked changes
- Inconsistent environments
- Increased attack surface

Immutable infrastructure solves this by making changes only through controlled, repeatable builds and deployments.

How Docker Hardened Images support immutability

Docker Hardened Images are built to be minimal, locked-down, and non-interactive, which discourages in-place modification. For example:

- Many DHI images exclude shells, package managers, and debugging tools
- DHI images are designed to be scanned and signed before deployment
- DHI users are encouraged to rebuild and redeploy images rather than patch running containers

This design aligns with immutable practices and ensures that:

- Updates go through the CI/CD pipeline
- All changes are versioned and auditable
- Systems can be rolled back or reproduced consistently

Immutable patterns in practice

Some common patterns that leverage immutability include:

- Container replacement: Instead of logging into a container to fix a bug or apply a patch, rebuild the image and redeploy it.
- Infrastructure as Code (IaC): Define your infrastructure and image configurations in version-controlled files.
- Blue/Green or Canary deployments: Roll out new images alongside old ones and gradually shift traffic to the new version.

By combining immutable infrastructure principles with hardened images, you create a predictable and secure deployment workflow that resists tampering and minimizes long-term risk.

□□□ #2
□ □□□ □□□ 2026-10-31 19:15:44 UTC
□ □□□ □□□ 2026-10-31 19:15:46 UTC