CHANGE MANAGEMENT POLICY

Version Number:	3.1
Version Date:	10/08/2025
Owned by:	Security, GRC
Approved by:	Nate Fitch, Sr. V.P., Chief of Staff
Confidentiality Level:	Internal

Table of Contents

1. Purpose

This policy establishes standardized, secure, and auditable procedures for managing changes to Docker's technology systems, software, infrastructure, and configurations. It ensures all changes follow a structured lifecycle including planning, classification, documentation, peer review, and testing in non-production environments before implementation.

The policy defines:

- Risk-based change assessment and approval workflows with appropriate stakeholder involvement
- Strict access controls and segregation of duties to prevent unauthorized modifications
- Emergency change procedures with mandatory post-implementation review
- Comprehensive logging to support audits, post-incident analysis, and operational transparency

This framework minimizes risks of disruptions, security incidents, and operational errors while enabling continuous improvement and innovation. It supports Docker's compliance with ISO/IEC 27001 (A.8.32), SOC 2 Trust Criteria (CC8.1, CC6), and secure DevOps best practices.

2. Scope

This policy applies to all changes that could impact the integrity, availability, or confidentiality of Docker’s systems, services, and environments, including but not limited to:

- Production
- Internal tooling, CI/CD pipelines, and developer infrastructure
- Software-as-a-Service (SaaS) platforms and integrations
- Code repositories, infrastructure-as-code, and configurations
- Identity and access controls (IAM)
- Employees, contractors, or vendors initiating or approving technical or operational changes

This policy covers both automated and manual changes and applies across engineering, DevOps, security, IT, and vendor-provided services.

3. Roles and Responsibilities

Role	**Responsibilities**
Developer	Submit, document, and resolve change requests with appropriate detail and justification.
Code Reviewer	Assess changes for technical accuracy, completeness, and impact.
Security Engineer	Review and approve changes that affect security posture or involve sensitive data.
DevOps / Infrastructure Team	Validate deployment tooling, enforce CI/CD gating rules, and monitor change execution.

Change Advisory Board (CAB) / VP	Provide oversight and sign-off on high-risk, emergency, or business-critical changes.
Compliance Team	Audit change records, verify process adherence, and generate reports for external and internal audits.

4. Definitions

Segregation of Duties - The security principle requiring that no individual can initiate, approve, and implement the same change without oversight, ensuring accountability and preventing unauthorized modifications.

Emergency Change - A time-sensitive modification required to resolve active incidents, critical vulnerabilities, or operational outages that may bypass standard approval procedures but requires mandatory post-implementation review.

Post-Implementation Review (PIR) - A formal assessment conducted within five business days of high-risk or failed changes to evaluate impact, identify lessons learned, and define corrective actions.

Code Owners - Designated individuals or teams with authority and responsibility to review and approve changes within specific repositories, modules, or system components based on technical expertise and business ownership.

5. Change Classification

All changes must be categorized to determine approval, testing, and documentation requirements:

Type	**Description**
Standard	Low-risk, routine changes that follow pre-approved runbooks or templates (e.g., nightly backups, patching).
Normal	Medium- to high-risk changes requiring full lifecycle adherence, including review, testing, and approval.
Emergency	Time-sensitive changes required to resolve incidents, critical vulnerabilities, or operational outages.
Technical	Code changes, system configurations, infrastructure updates, IAM modifications.
Operational	Business process changes, workflow updates, or onboarding/offboarding of SaaS platforms and vendors.

6. Change Lifecycle Stages

6.1 Planning and Initiation

- Changes must be initiated via a Pull Request (PR) or Change Request (CR) with a clear summary of intent, scope, and business justification.
- PR templates must include:
  - Purpose and scope of change
  - Linked ticket or incident (e.g., JIRA)
  - Risk level (low, medium, high)
  - Rollback plan and test strategy
  - Dependencies or sequencing considerations

6.2 Review and Approval

- All changes must undergo peer review by qualified team members.
- Code Owners must be defined for each repository and approve changes within their domain.
- High-risk or security-impacting changes require:
  - Security review and approval
  - Secondary technical approval
  - Final sign-off from a VP-level approver, as applicable

6.3 Testing

- Changes must be tested in a non-production environment using applicable unit, integration, and user acceptance tests.
- Test results must be linked to the PR or CR for traceability.
- Emergency changes must be tested retrospectively and documented accordingly.

6.4 Implementation

- All production changes must be deployed via approved CI/CD automation tools (e.g., GitHub Actions, Jenkins, ArgoCD).
- Manual deployment to production must be minimized and require dual authorization.
- Merge and deploy permissions to production are restricted to CI service accounts or designated automation roles.
- Rollback instructions must be validated and included for all changes.

6.5 Logging and Auditability

- Change events must be logged with timestamps, approvers, commit hashes, and implementation details.
- GitHub audit logs, CI/CD deployment logs, and change tracking systems must retain records for a minimum of 12 months.
- All system changes are subject to continuous monitoring through automated compliance tooling.

6.6 Segregation of Duties

- No individual may initiate, approve, and implement the same change without oversight.
- Any required bypass of this principle for emergencies must be fully documented and reviewed after implementation.

7. Emergency Change Procedure

Emergency changes may bypass standard procedures if required to address:

- Active incidents or outages
- Critical vulnerabilities (e.g., P1 CVEs)
- Compliance failures or urgent legal requirements

Emergency changes must:

- Be logged in GitHub or CR tracking system.
- Be approved.
- Include justification and rollback plan.
- Be reviewed within the target SLAs of our incident response plan.
- Be added to the change tracking register and subject to post-review

It is acceptable if some of these steps are conducted post-change-execution.

9. Enforcement and Exceptions

All Docker personnel are expected to comply with this policy in full. Failure to adhere may result in disciplinary action, up to and including termination of employment or contract. In cases involving willful misconduct, malicious activity, or violations of law, Docker may initiate legal proceedings. The organization reserves the right to monitor, audit, and inspect user activity on its systems to ensure compliance.

Any exceptions to this policy must be formally requested, documented, and approved by Docker's GRC or Legal team. Exception requests must clearly outline the business justification and associated risks, and should be submitted to Docker's Risk Submission Portal.

10. Change History

Date	Version	Responsible Party	Description of Change
10/08/2025	3.1	Nate Fitch, Sr. V.P., Chief of Staff	Revised policy approved
9/29/2025	3.0	Chad Fryer, Sr. Security Engineer, GRC	Policy annual review and update
12/03/2024	2.1	Tushar Jain, Executive Vice President, Engineering	Policy Approved

4/27/2024	2.0	Rachel Taylor, Director, Security, Risk & Trust	Draft Updated - Scope of Policy Expanded
2/23/2023	1.2	Jean-Laurent de Morlhon, SVP of Engineering	Policy Approved
2/6/2023	1.1	Rachel Taylor, Senior Manager, Security, Risk & Trust	Draft Submitted for Review
12/19/2022	1.0	Rachel Taylor, Senior Manager, Security, Risk & Trust	Basic Document Outline