

Change Management Policy

Owner	[Security owner role]
Approved by	[Policy approver role]
Effective date	[Effective date]
Next review	[Next review date]
Version	[Version]

1. Purpose and scope

This policy supports the SOC 2 common criteria CC8.1 by establishing a formal process for managing changes to production systems.

This policy applies to every change to application code, infrastructure as code, databases and system configuration in the systems in scope.

2. Change management

- Tracking: Changes are tracked in [Ticketing system].
- Approval: Changes require peer review and approval before deployment.
- Testing: Changes are tested in a separate environment before production release.
- Emergency changes: A process exists for emergency changes with post-implementation review.
- Deployment: Changes reach production only through [Deployment pipeline], which records who deployed what and when.

3. Types of change

Every change is one of the types below. The type decides the approval it needs and the record it leaves.

Type	Example	Approval	Record
Standard	A dependency patch or a copy change	One reviewer who is not the author	The merged pull request
Normal	A feature, a schema change or a new integration	One reviewer who is not the author, with automated tests passing	The ticket and the merged pull request

Type	Example	Approval	Record
Infrastructure	A change to networks, permissions or cloud resources	One reviewer who is not the author, with the planned change attached	The pull request with the plan output
Emergency	A fix shipped during an incident	The incident lead, then a reviewer within 2 business days	The incident record linked to the change

4. Secure development

- Code review: Code changes require review by at least one other engineer.
- Branch protection: Production branches in [Code repository] are protected against direct commits.
- Secrets management: Secrets are managed in [Secrets manager] and never hardcoded in source code. Secrets are rotated upon departure of staff with access or suspected compromise.
- Environment separation: Production, staging, and development environments are logically separated.

5. Small teams

No one approves their own change. Where only one engineer works on a system, a second named person who did not write the change, such as the [Security owner role], reviews it before release.

6. Roles and responsibilities

Role	Held by	Responsibility
Policy approver	[Policy approver role]	Approves this policy, each new version, and any risk the Company accepts instead of meeting it.
Policy owner	[Security owner role]	Maintains this policy, makes sure the controls in it operate, keeps the evidence, and reviews it at least once a year.
Reviewers	[Engineering owner role]	Review and approve changes they did not write, and confirm tests passed.
All personnel	Everyone in scope	Follow this policy and report a suspected breach of it to the [Security owner role].

7. Evidence and records

The records below show that this policy operates. Each is dated, kept for at least [Evidence retention period], and stored where an auditor can be given it.

Record	What it shows	Where it is kept
Approved policy	This document, signed by the approver, with its version history.	[Documentation system]
Change request records	Approved change tickets.	[Ticketing system]
Branch protection rules	Screenshot of repository settings.	[Code repository]
Code review evidence	Example pull request with approval.	[Code repository]
Deployment log	Every production deployment, with the person and the time.	[Deployment pipeline]
Emergency change reviews	The after-the-fact review of each emergency change.	[Ticketing system]

8. Exceptions and enforcement

A deviation from this policy needs a written exception approved by the [Security owner role] before it begins. Each exception records the rule not met, the business reason, the compensating control and an expiry date no more than 12 months away. Exceptions are reviewed at each annual review.

A breach of this policy is handled under the Company disciplinary process and may lead to action up to and including termination of employment or contract. A breach that exposes customer data is also handled as a security incident.

9. Review and approval

The [Security owner role] reviews this policy at least once a year and after any significant change to the business, the systems in scope or the risks they face. The [Policy approver role] approves each version. Personnel are told of material changes and acknowledge the current version.

Version	Date	Summary of change	Approved by
[Version]	[Effective date]	First version.	[Policy approver role]

Policy approver

Name:

Signature:

Date:

10. Fields to complete

Replace every bracketed field in this document with your own detail, then delete this section.

Field	What to enter
[Company name]	Your legal company name.
[Effective date]	The date the approver signs this version.
[Version]	Start at 1.0 and increase it each time the document changes.
[Security owner role]	The role that runs the security program day to day, for example Chief Technology Officer.
[Policy approver role]	The officer who approves policies and accepted risks, for example Chief Executive Officer.
[Documentation system]	Where policies and procedures are published, for example the company wiki.
[Ticketing system]	Where changes and incidents are tracked, for example Jira or Linear.
[Code repository]	Where source code is hosted, for example GitHub or GitLab.
[Deployment pipeline]	The system that builds and deploys code, for example GitHub Actions.

Field	What to enter
[Secrets manager]	Where keys and secrets are stored, for example a cloud key management service.
[Evidence retention period]	How long audit evidence is kept, for example three years.
[Engineering owner role]	The role that owns production systems, for example engineering lead.
[Next review date]	The date this version must be reviewed by, at most 12 months after the effective date.

Source

Source: free template from www.polaralabs.com/templates/change-management-policy (CC BY 4.0).