

Deploy process

This is an example deployment checklist for the fictional handbook team. Adapt commands and rollback steps to your own service before using it operationally.

Before you deploy

1. Link the change to an issue that explains the user-visible behavior.
2. Get a code review and pass the relevant automated checks.
3. Verify the change in staging, including an error path and a small-screen view when the interface changes.
4. Check migrations for compatibility with the currently running version. Prepare a separate plan for irreversible data changes.
5. Record the release version, previous working version and person watching the release.

Release and observe

Deploy during staffed hours. Announce the affected service in the operations channel, then release through the approved pipeline. Check the health endpoint and complete the user journey changed by the release. Watch errors, latency and failed jobs for 15 minutes against the pre-release baseline.

If something fails

Pause further releases and assign an incident coordinator. Roll back to the recorded working version when the database remains compatible. If rollback could lose data, stop and follow the migration recovery plan. Do not repeatedly redeploy without understanding the failure.

Close the loop

Record the result and link the monitoring evidence. Create follow-up issues with owners for anything still unresolved. A release is complete when users can perform the intended task, not merely when the build is green.

Owner: Engineering. Review after a deployment incident.

Created 2026-09-09 00:12:40 UTC by Admin
Updated 2026-09-09 00:12:40 UTC by Admin