

Deploy Rollback Procedure

INTERNAL — Helix Engineering / CS Operations

WHEN TO ROLL BACK

Roll back when a deploy has caused: (1) elevated error rate, (2) increased latency, (3) any data corruption, or (4) security regression. Don't try to debug forward in production during an active incident; roll back first.

STEP 1 — Identify the bad deploy

From the deploys dashboard, identify the deploy that correlates with the symptom. Helix records every deploy with its commit SHA, deployer, and timestamp.

STEP 2 — Confirm rollback is safe

Database migrations: if the bad deploy ran a migration that is not backward-compatible, rolling back the code will not roll back the schema. Stop and engage the on-call DBA.

STEP 3 — Execute rollback

Run ``helix deploy rollback`` from the deploy CLI. Specify the service name. The CLI confirms the target version (the previous successful deploy) and proceeds on confirmation.

STEP 4 — Monitor

Watch error rates and latency dashboards. The rollback should reduce error rates within 60 seconds. If not, the rollback may have failed; check pod status.

STEP 5 — Communicate

Update #incidents with the rollback action. If customers were impacted, update the status page.

STEP 6 — Block re-deploy

Lock the service in the deploy CLI: ``helix deploy lock``. This prevents re-deploys until manually unlocked. Do not unlock until root cause is fixed and verified.

POSTMORTEM

Every rollback triggers a postmortem within 5 business days. Even if the rollback resolved the issue cleanly, the postmortem captures why the bug got past CI.