EcomLabs

TEMPLATE AND CHECKLIST

A rollback plan before you deploy

The rollback plan template in seven lines, the three expand and contract steps that keep the old code working, what to verify after the deploy and the final checklist. The SQL and the verification script are tested in the article.

FOR	Teams deploying a web app with a PostgreSQL database
TESTED WITH	Node 22.22.3 · PGlite 0.5.10 (PostgreSQL 18.6) · Windows 11
UPDATED	November 12, 2026
DOCS	railway.com/guides/roll-back-bad-deploy
ARTICLE	ecomlabs.dev/g/rollback-en
AUTHOR	Luis Morales · @luismoralesdev

Contents

Before you start	What this guide covers
Step 01	The rollback plan template
Step 02	Migrating without breaking the way back
Step 03	Verify after the deploy
When you finish	Post-deploy checklist

BEFORE YOU START

What this guide covers

A rollback plan is the note you write before you deploy so that going back is a decided step, not an argument. Here is the template, how to migrate without breaking the old code, what to verify afterward and the final checklist. We tested the SQL on PGlite 0.5.10 (PostgreSQL 18.6 in WebAssembly) and the verification script against a local test server, with Node 22.22.3. The platform steps come from Railway's documentation and were not run on a real deployment.

STEP 01

The rollback plan template

Fill it in once per service, before the deploy. The examples are illustrative.

FIELD	WHAT TO WRITE	ILLUSTRATIVE EXAMPLE
New version	The identifier `/healthz` returns	`a1b2c3d`
Version you return to	The last identifier that worked, not "the previous one"	`9f8e7d6`
How you return	Exact steps and who has access	Deployments, deployment `9f8e7d6`, three-dot menu, Rollback
Window	How long the platform keeps the previous image	72 h (Hobby plan)
Database	Last migration and whether it has been contracted	Expanded, not contracted
Criterion	What result triggers the rollback	The verification script exits with code 1
Who decides	A name, with a phone number	Technical owner
Who is told	Sales, support, the agency	Team channel

Per Railway's guide, a rollback restores the Docker image and custom variables as a new deployment, with no rebuild. Image retention is 24 hours on Free/Trial, 72 on Hobby, 120 on Pro and 360 on Enterprise; after that you use Redeploy, which rebuilds from the original source. That guide says nothing about databases or migrations.

STEP 02

Migrating without breaking the way back

A one-step `RENAME COLUMN name TO full_name` broke the old code in our test (`column "name" does not exist`). With expand and contract, every step works with both versions:

STEP	WHEN	WHAT YOU DO	WAY BACK TO THE OLD VERSION?
1 · Expand	Before the deploy	`ADD COLUMN full_name text`, a trigger that syncs with `name` and a backfill `UPDATE`	Yes
2 · Migrate	In the deploy	The new version reads and writes only `full_name`	Yes
3 · Contract	After the rollback window	`SET NOT NULL`, `DROP TRIGGER`, `DROP FUNCTION`, `DROP COLUMN name`	No

> **CAUTION**
> The full trigger SQL and the test are in the article. Contract closes the way back, so write it in the plan. On a large table, run the backfill in batches and test the trigger with your own writers.

STEP 03

Verify after the deploy

The platform's healthcheck runs only at deploy time and does not watch afterward. Verify yourself, with a script that exits non-zero if anything fails (the full script is in the article). At minimum, check:

- `/healthz` returns the release you just deployed (it catches the old version still serving).
- The home page returns 200 and has its content.
- A product page returns 200 and contains the add to cart button.
- A path that does not exist returns 404 (it catches a server that answers 200 to everything).

Locally, against a test server on 127.0.0.1: healthy new version, exit 0; product returning 500, exit 1; old version still serving, exit 1; nothing listening, exit 1.

WHEN YOU FINISH

Post-deploy checklist

- [ ] `/healthz` returns the release you just deployed.
- [ ] The verification script exits with code 0.
- [ ] You walked product, cart and checkout by hand, or ran the smoke test.
- [ ] The logs from the first minutes show no new errors.
- [ ] If the plan's criterion was met, you went back first and are investigating afterward.
- [ ] This deploy's migration only expands; contract goes in a later one, after the window.
- [ ] You wrote down what happened and what you change in the plan.