

From n8n to Cloudflare Workflows

A content-pipeline port built around explicit step boundaries, control and recoverability.

INPUT

154-node source workflow

OUTPUT

Native Worker and Workflow code

SCOPE

150 runtime nodes mapped

HOW THE SYSTEM FITS TOGETHER

01

Map

Account for the source workflow

02

Port

Compile behavior into typed code

03

Control

Validate ingress and side effects

04

Verify

Test paths and rehearse rollback

WHAT THE SCOPE COVERS

- A census of 154 source nodes, including 150 runtime nodes.
- 49 Code nodes compiled into testable functions.
- Signed ingress, execution controls and explicit retry boundaries.
- Test evidence, a parity method and an operating runbook.

DESIGN DECISIONS

Keep a traceable mapping

Every runtime node needs an explanation in the port; visual similarity is insufficient evidence of equivalent behavior.

Treat side effects deliberately

Signed requests, idempotency and execution gates control what a replay or retry can do.

Rehearse the operational change

The recorded canary and rollback exercise exposed an integration issue before wider use.

IMPLEMENTED AND TESTED

The evidence covers implementation, tests and a recorded canary exercise. It does not establish production parity for every external provider or workload.

Cloudflare Workers / Workflows / TypeScript / n8n / KV

[Read the full case study](#)

Scope and tradeoffs, explained in detail