Engineer Software

{ DevOps & CI/CD Automation }

Shipping changes without drama

Tested, released and reversible, so changes reach customers on a Tuesday afternoon rather than late on a Friday night.

Still working out what you need?

Tell us what is going wrong, or what you are trying to build. We will tell you what it would take, roughly what it would cost, and whether we are the right people for it.

From 1,400 DKK an hour, excluding VAT — and less per hour on a longer booking.

Get in touch See what it costs

Selected work demonstrating this service.

Turn the service into a result

Make “DevOps & CI/CD Automation” serve a business outcome. We can connect the technical work to the users, operations and decisions it needs to support. Tested, released and reversible, so changes reach customers on a Tuesday afternoon rather than late on a Friday night.

Get in touch

Examine the relevant work

Use the related work on this page to examine our approach. Ask about the tradeoffs and the conditions behind each result. Automated GIS SaaS deployment with GitHub Actions — deployment, data and reporting, all off the hands

Automated GIS SaaS deployment with GitHub Actions Get in touch →

Automated GIS SaaS deployment with GitHub Actions

Automated GIS SaaS application deployment, data processing, and reporting system using GitHub Actions CI/CD, Python, Bash, and SQL.

Automated delivery of 20 GIS data pipelines

Automated delivery of 20 GIS data pipelines and app data ETL processes, streamlining infrastructure automation and reporting.

Overhauled internal processes, saving 8,000 hours

Overhauled internal processes, saving 8,000 hours by improving software architecture, systems, and scheduling efficiency.

Infrastructure simplified beyond AWS

Reworked an international client's infrastructure by removing unused services, adjusting capacity and moving workloads away from AWS. The founder reports a 10x reduction in cloud costs with no loss of productivity.

Generated the API contract from the database out

Generated the API contract outward from the database — OpenAPI, a 44,076-line typed TypeScript client, 61 mock handlers and the limits the UI enforces — with a guard at every hop that fails on drift.

Built the entire core infrastructure from scratch

As one of the first hires, designed and built the entire core infrastructure and supporting processes from scratch for a green-energy SaaS startup, laying the foundation for rapid growth.

Built a test suite across Go, SQL and browser

Built a layered automated test suite — 981 Go tests, 543 frontend and browser specs, 494 SQL behavioural tests — with mutation testing, property-based tests and an accessibility gate.

Built a 268-check guard engine for the repo

Built the repository's guard engine — 268 registered commit checks, 277 lint rules and 15 custom ESLint rules — plus 146 tests of the guards themselves, so the build holds the standard, not review.

Built first-party error monitoring and tracing

Built first-party error monitoring and OpenTelemetry tracing rather than buying them — payload sanitising, spike and regression detection, symbolication and a synthetic heartbeat — behind 11 operator views.

Your code on your own server

Deployed the company's own git forge on Soft Serve, private by default with no web panel and its SSH port bound to loopback behind a jump host, and made the landing page in front of it a build artefact of the main site rather than a hand-kept copy.

Built a 22-linter commit gate

Built a commit gate of 22 one-line linters plus five that earn a paragraph, with no warning tier and no inline suppressions permitted, covering HTML, CSS, JavaScript, Python, YAML, Markdown, shell, links, spelling, secrets and typography.

A safety net that no longer costs half an hour

Cut the site's browser-driven quality gate from 1,636 seconds to 615 by scheduling its checks longest-first through a worker pool bounded to four lanes, after measuring that alphabetical order cost 320 seconds against 224.

Mirrored the site to Gemini and Gopher

Built Gemini and Gopher outputs from the same website source; the achievement records 777 documents in each format at that stage, with zero HTML bytes changed by adding the formats.

Fixed a sitemap with one shared timestamp

Fixed a sitemap where 172 of 176 URLs shared one modification timestamp, by taking the date from git history after establishing that the export rewrites every file on every run.

Linted the 10,242-line quality gate itself

Brought 10,242 lines of quality-gate JavaScript under a formatter and a linter after establishing it was the largest body of code in the repository and the only one nothing read, fixing 13 findings and suppressing none.

Held the generator to 981 tests and 92% coverage

Held the generator to 981 test cases at a 92% branch-coverage floor with warnings treated as failures, and asserted idempotence by running the whole build pipeline twice from an empty file and requiring the second pass to change nothing.

Enabled every linter rule as an error

Selected every rule the Python linter has as an error, working through 1,815 findings to reach zero, with each of the few exemptions carrying a written reason and two of them backed by a checker instead of a comment.

Wrote tests for the checkers themselves

Wrote tests for the checkers themselves after establishing that a checker fed only clean input will one day report clean because it read nothing — planting a misspelling to confirm the spell-check finds it, and taking an id range from the database rather than from a number in the test.

Wrote a parser verifying every FFI call

Wrote a parser that reads the real 7,308-line C header and verifies every call site, every enum constant and that every pointer-owning class is final, after a hand-written placeholder header let calls to three removed functions compile, link and crash.

Reached the unused half of the core

Reached the half of the messaging core the application had never used — backup transfer, disappearing messages, message editing and resending, verified invitations, proxies and encryption policy — driving every test against the real library with no mocks.

Looking for DevOps & CI/CD Automation?

Use the linked work to judge relevant experience. Tell us what you need to build; we can discuss the scope, delivery approach and handover.

Get in touch Full portfolio

Scope the work before commissioning it

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Talk through your requirements

Tell us what you need from “DevOps & CI/CD Automation”, including the deadline and the team that will maintain it. We can discuss a practical way forward.

Get in touch

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://software.engineer.company/services/devops-cicd/