Engineer Софтвер

Автоматизація та CI/CD

35 досягнень

Поєднайте частини системи

Роботи за темою «Автоматизація та CI/CD» показують одну частину дієвої системи. Ми можемо допомогти пов'язати її з рештою вашої діяльності. Цифри з'являються там, де потрібні, за розкладом, і ніхто не копіює таблицю наприкінці місяця.

Переглянути наші послуги Зв'язатися з нами →

Знайдіть схоже завдання

Відкрийте приклад, схожий на вашу ситуацію. Огляд пояснює рішення, що привели до результату, а не лише називає технологію. Один інфраструктурний план для 14 підрозділів — 14 дослідницьких підрозділів, охоплених спільним інфраструктурним планом

Один інфраструктурний план для 14 підрозділів Зв'язатися з нами →

Усунула 1 000 проблем у географічних даних

Усунула 1 000 проблем у географічних даних та часових рядах, використовуючи GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL), Bash, забезпечивши високоякісну обробку великих даних.

Спроєктувала й підтримувала 6 пайплайнів ETL/ELT

Спроєктувала, реалізувала та адмініструвала 6 пайплайнів ETL/ELT, використовуючи Google BigQuery, MSSQL, PostgreSQL, shell-скрипти, PL/pgSQL та Transact-SQL, інтегрувавши дані для ефективної обробки через Python API.

Автоматизувала розгортання GIS SaaS через CI/CD

Автоматизувала розгортання GIS SaaS-застосунку, обробку даних та систему звітності за допомогою GitHub Actions CI/CD, Python, Bash та SQL.

Автоматизувала розгортання 20 GIS-пайплайнів

Автоматизувала розгортання 20 пайплайнів GIS-даних та ETL-процесів даних застосунку, оптимізувавши автоматизацію інфраструктури та звітність.

Реорганізувала процеси, заощадивши 8 000 годин

Реорганізувала внутрішні процеси, заощадивши 8 000 годин завдяки вдосконаленню архітектури програмного забезпечення, систем та ефективності планування.

Розробила систему аналітичної звітності

Розробила систему аналітичної звітності, збільшивши квартальний дохід від програмного забезпечення на 400% завдяки PDF-звітам на базі Python.

Згенерувала контракт API із самої бази даних

Згенерувала контракт API з бази даних назовні — OpenAPI, типізований TypeScript-клієнт на 44 076 рядків, 61 mock-обробник та обмеження, які застосовує UI, — з перевіркою на кожній ланці, що падає в разі розбіжностей.

Створила англо-данську систему i18n

Створила систему інтернаціоналізації для англійської та данської мов, що містить 12 027 повідомлень на локаль у 458 файлах просторів імен, зі словником, дотримання якого забезпечував лінтер, та бюджетом у 400 рядків на файл.

Побудувала тести для Go, SQL і браузера

Побудувала багаторівневий набір автоматизованих тестів — 981 тест Go, 543 фронтендні та браузерні специфікації, 494 поведінкові тести SQL — з мутаційним тестуванням, property-based тестами та обов'язковою перевіркою доступності.

Побудувала механізм із 268 перевірок репозиторію

Побудувала механізм перевірок репозиторію — 268 зареєстрованих перевірок коміту, 277 правил лінтингу та 15 власних правил ESLint — а також 146 тестів самих перевірок, щоб стандарт тримала збірка, а не код-рев'ю.

Перенесла повільні операції в чергу завдань River

Перенесла повільну роботу зі шляху запиту в чергу завдань River — 15 модулів воркерів, 8 запланованих задач та 20 завдань pg_cron — щоб запит повертався, поки робота за ним триває.

Ваш код на вашому власному сервері

Розгорнула власний git-сервер компанії на Soft Serve, приватний за замовчуванням і без вебпанелі, з портом SSH, прив'язаним до loopback за проміжним хостом (jump host), і зробила цільову сторінку перед ним артефактом збірки основного сайту, а не копією, яку підтримують вручну.

Побудувала перевірку комітів із 22 лінтерів

Побудувала перевірку перед комітом із 22 лінтерів, кожен з яких описується одним рядком, і ще п'ятьох, що заслуговують окремого абзацу, без рівня попереджень і без дозволу на вбудоване вимкнення правил, яка охоплює HTML, CSS, JavaScript, Python, YAML, Markdown, shell, посилання, орфографію, секрети й типографіку.

Страховка, що більше не забирає пів години

Скоротила браузерну перевірку якості сайту з 1 636 секунд до 615, запускаючи перевірки від найдовших до найкоротших через пул воркерів із максимум чотирма одночасними перевірками, після того як виміряла, що алфавітний порядок коштував 320 секунд проти 224.

Замінила 27 розмірів шрифту шкалою з 6 кроків

Замінила 27 фактичних розмірів шрифту, сусідні значення яких різнилися лише на 0,6%, шестиступеневою шкалою Major Third і написала лінтер, який відхиляє двадцять восьмий.

Вади доступності, яких не бачить ваша автоматична перевірка

Знайшла й виправила 11 дефектів доступності, які браузер вважав справними: вісім посилань у підвалі, що лишалися в порядку табуляції попри вимкнену взаємодію через pointer-events, часову шкалу прокрутки (scroll timeline), яка обрізала колофон на трьох сторінках, і рушій правил, що виконував лише 70 зі своїх 105 правил.

Обмежила палітру сайту 28 кольорами

Обмежила систему кольорів 28 задокументованими кольорами й написала лінтер, який відхиляє значення, що застосоване, але не задокументоване, задокументоване, але не застосоване, хибно виміряне, переписане як літерал або розташоване в межах перцептивної відстані 0,02 від уже наявного.

Знайшла, що світлій темі бракує цілого шару

Виявила, що у світлій темі від самого її створення бракувало повносторінкового шару накладання, додавши перевірку, яка вимагає, щоб обидві теми малювали кожну багатошарову поверхню однаковою кількістю шарів.

Віддзеркалила сайт у Gemini та Gopher

Побудувала вихідні формати Gemini та Gopher з того самого джерела сайту; запис досягнення фіксує по 777 документів на тому етапі та нуль змінених байтів HTML від додавання форматів.

Замінила 963 блоки метаданих одним графом

Замінила 963 анонімні блоки структурованих даних, які 1 671 раз повторно описували компанію, одним зв'язаним графом із 16 типів, створеним на основі стабільних ідентифікаторів походження.

Скоротила пакунок стилів із 87 КБ до 31 КБ

Скоротила пакунок стилів із 87 КБ до 31 КБ, вийнявши з нього шрифт у base64, і позбулася 756 КБ у 22 файлах, які публікувалися під час кожного розгортання і на які ніщо не посилалося.

Виправила мапу сайту з однією позначкою часу

Виправила мапу сайту, де 172 з 176 URL мали одну позначку часу зміни, взявши дату з історії git, після того як з'ясувала, що експорт переписує кожен файл під час кожного запуску.

Перевірила лінтером 10 242 рядки перевірки якості

Охопила форматувальником і лінтером 10 242 рядки JavaScript у перевірках якості, з'ясувавши, що це найбільший обсяг коду в репозиторії і єдиний, який ніщо не перевіряло, — виправила 13 зауважень і не вимкнула жодного.

Побудувала генератор резюме на основі бази даних

Побудувала на Python генератор CV, рекомендацій, портфоліо та супровідних листів на основі бази даних — 41 модуль, 10 580 рядків, — що відтворює шість форматів виводу з одного джерела SQLite на 23 таблиці, зібраного ідемпотентним пайплайном із 19 кроків.

Забезпечила генератору 981 тест і 92% покриття

Покрила генератор 981 тестовим випадком із мінімальним порогом покриття гілок 92%, де попередження вважаються помилками, і перевірила ідемпотентність, двічі запустивши весь пайплайн збірки з порожнього файлу та вимагаючи, щоб другий прохід нічого не змінював.

PDF/UA-1: пройдено 106 зі 106 перевірок профілю

Зафіксувала перевірку PDF/UA-1 для 9 документів: успішно пройдено 106 зі 106 налаштованих перевірок профілю доступності, тоді як архівний варіант не пройшов одне зі 146 правил.

Увімкнула всі правила лінтера Python як помилки

Увімкнула всі правила лінтера Python як помилки й довела 1 815 зауважень до нуля; кожен із небагатьох винятків має письмове обґрунтування, а два з них підкріплені перевіркою, а не коментарем.

Написала тести для самих перевірок

Написала тести для самих перевірок, з'ясувавши, що перевірка, яка отримує лише чисті вхідні дані, одного дня звітуватиме про чистоту, бо нічого не прочитала: підклала помилку правопису, щоб підтвердити, що перевірка орфографії її знаходить, і взяла діапазон ідентифікаторів із бази даних, а не з числа в тесті.

Побудувала міжмовну перевірку чисел

Побудувала міжмовну перевірку контенту, яка падає, коли переклад губить число, вказане англійською, або коли мова вживає не притаманну їй нотацію, — і знайшла два данські описи без метрики та шістнадцять українських фрагментів, що цитували на англійський лад.

Написала парсер, що перевіряє кожен виклик FFI

Написала парсер, який читає справжній C-заголовок на 7 308 рядків і перевіряє кожне місце виклику, кожну константу переліку й те, що кожен клас-власник вказівника є final, після того як через написаний вручну заголовок-заглушку виклики трьох вилучених функцій компілювалися, лінкувалися й аварійно завершувалися.

Назвала всі 17 піктограм для зчитувачів екрана

Дала назви для зчитувачів екрана всім елементам керування, що складаються лише з піктограми, після того як виявила, що кнопка надсилання оголошувалася як «arrow up circle, button», і написала лінтер, який вимагає підпису в межах восьми рядків від піктограми.

Побудувала дизайн-систему macOS із 48 токенів

Побудувала дизайн-систему з 48 токенів на основі власного скляного матеріалу платформи й написала лінтер, який відхиляє магічне число, жорстко заданий розмір шрифту чи анімацію, що ігнорує налаштування зменшеного руху.

Задіяла невикористану половину ядра

Задіяла ту половину ядра обміну повідомленнями, якою застосунок ніколи не користувався, — передавання резервної копії, зникомі повідомлення, редагування та повторне надсилання, підтверджені запрошення, проксі та політику шифрування, — проганяючи кожен тест на справжній бібліотеці без заглушок.

Створила айдентику з 2 малюнків

Створила набори піктограм і фавіконів компанії на основі двох намальованих вручну зображень; кожен похідний ресурс перегенеровується скриптом, а перевірка падає, якщо похідний файл закомічено раніше за його джерело.

Змусила шість збирачів звітувати про пропущене

Змусила шість збирачів звітів повідомляти ту частину свого входу, яку вони так і не прочитали, — файли, що лишилися закритими, невстановлені зонди, нерозпізнані рядки журналу, — і підігнала пороги до 116 виміряних зразків, з'ясувавши, що сліпий збір і справний хост дають однакову сторінку.

Спрямуйте інженерні зусилля

Надайте репрезентативний набір даних, запитання до нього та поточне вузьке місце. Узгодимо вимірювання якості, пропускної здатності та відновлення.

Дивитися, скільки це коштує Зв'язатися з нами →

Розгляньте тему у своєму контексті

Розкажіть, як «Автоматизація та CI/CD» стосується вашої системи. Ми допоможемо обрати відповідну послугу та визначити обсяг роботи.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://software.engineer.company/uk/categories/automation/