Engineer Logiciels

DevOps

16 réalisations

Reliez les parties de votre système

Les réalisations regroupées sous « DevOps » montrent une partie d'un système opérationnel. Nous pouvons vous aider à relier cette partie au reste de votre activité. Tout ce qui se situe quelque part — navires, véhicules, parcelles, capteurs — sur une carte où chercher, mesurer et partager.

Découvrir nos prestations Contactez-nous →

Trouvez un défi comparable

Suivez un exemple qui ressemble à votre situation. Son analyse décrit les décisions qui ont conduit au résultat, au-delà du nom de la technologie. Pipeline géographique accéléré de 50 fois — 50x d'accélération des pipelines de données géographiques

Pipeline géographique accéléré de 50 fois Contactez-nous →

Déploiement SaaS GIS automatisé : GitHub Actions

Automatisé le déploiement d'une application SaaS GIS, le traitement des données et le système de reporting avec GitHub Actions CI/CD, Python, Bash et SQL.

Livraison de 20 pipelines GIS automatisée

Automatisé la livraison de 20 pipelines de données GIS et processus ETL de données applicatives, en simplifiant l'automatisation de l'infrastructure et le reporting.

Application GIS : revenus multipliés par 10

Dirigé le développement logiciel d'une application cartographique GIS, multipliant le chiffre d'affaires par 10 et faisant du produit un actif de données central.

Infrastructure simplifiée au-delà d'AWS

Refondu l'infrastructure d'un client international en supprimant les services inutilisés, en ajustant la capacité et en déplaçant les charges de travail hors d'AWS. La fondatrice fait état de coûts cloud divisés par 10, sans perte de productivité.

Contrat d'API généré à partir de la base

Généré le contrat d'API à partir de la base de données — OpenAPI, client TypeScript typé de 44 076 lignes, 61 gestionnaires de simulation et limites appliquées par l'interface — avec un contrôle à chaque étape qui échoue en cas de divergence.

Choix techniques fondateurs de la plateforme

Défini les choix fondateurs de la plateforme dans les semaines suivant l'ouverture du dépôt en novembre 2025 — architecture en couches, accès aux données centré sur la base et zéro avertissement — toujours en vigueur neuf mois plus tard.

Infrastructure centrale construite de zéro

Parmi les premières personnes recrutées, conçu et construit de zéro toute l'infrastructure centrale et les processus associés d'une startup SaaS d'énergie verte, posant les bases d'une croissance rapide.

Suite de tests Go, SQL et navigateur

Conçu une suite de tests automatisés en couches — 981 tests Go, 543 spécifications frontend et navigateur, 494 tests de comportement SQL — avec tests par mutation, tests fondés sur les propriétés et contrôle obligatoire de l'accessibilité.

Moteur de 268 contrôles pour le dépôt

Conçu le moteur de garde-fous du dépôt — 268 contrôles de commit enregistrés, 277 règles de linting et 15 règles ESLint personnalisées — ainsi que 146 tests de ces garde-fous, pour que les exigences soient appliquées par la compilation et non par la seule revue.

Surveillance des erreurs et traçage internes

Conçu en interne la surveillance des erreurs et le traçage OpenTelemetry plutôt que de les acheter — assainissement des charges utiles, détection des pics et régressions, symbolication et signal de vie synthétique — accessibles dans 11 vues d'exploitation.

Votre code sur votre propre serveur

Déployé la forge git de l'entreprise avec Soft Serve, privée par défaut, sans interface web et avec un port SSH lié à l'interface de bouclage derrière un hôte relais, puis fait de sa page d'accueil un artefact de compilation du site principal plutôt qu'une copie maintenue à la main.

Contrôle avant commit avec 22 linters

Conçu un contrôle obligatoire avant commit composé de 22 linters tenant sur une ligne et de cinq nécessitant un paragraphe, sans niveau d'avertissement ni suppression locale autorisée, couvrant HTML, CSS, JavaScript, Python, YAML, Markdown, shell, liens, orthographe, secrets et typographie.

Des vérifications qui ne prennent plus une demi-heure

Réduit les vérifications qualité du site pilotées par navigateur de 1 636 à 615 secondes, en ordonnançant les contrôles du plus long au plus court dans un pool de quatre processus, après avoir mesuré un coût de 320 secondes en ordre alphabétique contre 224.

Contrôle qualité : 10 242 lignes vérifiées

Soumis 10 242 lignes de JavaScript des contrôles qualité à un formateur et à un linter, après avoir établi qu'il s'agissait du plus gros ensemble de code du dépôt et du seul que rien ne contrôlait, en corrigeant 13 constats sans en masquer aucun.

981 tests et 92 % de couverture

Vérifié le générateur avec 981 cas de test et un seuil de couverture des branches de 92 %, en traitant les avertissements comme des échecs, et vérifié l'idempotence en exécutant deux fois tout le pipeline depuis un fichier vide, sans changement autorisé au second passage.

Chaque règle du linter traitée comme une erreur

Activé comme erreurs toutes les règles du linter Python, corrigé 1 815 constats pour atteindre zéro et documenté la justification de chaque rare exemption, dont deux soutenues par un contrôle plutôt que par un commentaire.

Ciblez l'effort d'ingénierie

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Replacez le sujet dans votre contexte

Dites-nous comment « DevOps » se manifeste dans votre propre système. Nous pouvons vous aider à choisir la prestation pertinente et à définir les travaux correspondants.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://software.engineer.company/fr/categories/devops/