Engineer Software

{ DevOps- und CI/CD-Automatisierung }

Änderungen ohne Aufregung ausliefern

Geprüft, veröffentlicht und rückgängig zu machen: Änderungen erreichen Ihre Kunden am Dienstagnachmittag statt spät am Freitagabend.

Sie klären noch, was Sie benötigen?

Erzählen Sie uns, was nicht funktioniert oder was Sie entwickeln möchten. Wir sagen Ihnen, was dafür nötig wäre, was es ungefähr kosten würde und ob wir die Richtigen dafür sind.

Ab 1.400 DKK pro Stunde, zuzüglich Mehrwertsteuer — und weniger pro Stunde bei längerer Buchung.

Kontakt aufnehmen Preise ansehen

Ausgewählte Arbeiten, die diese Leistung zeigen.

Die Leistung in ein Ergebnis überführen

Machen Sie „DevOps- und CI/CD-Automatisierung“ für ein Geschäftsergebnis nutzbar. Wir können die technische Arbeit mit den Nutzern, Betriebsabläufen und Entscheidungen verbinden, die sie unterstützen soll. Geprüft, veröffentlicht und rückgängig zu machen: Änderungen erreichen Ihre Kunden am Dienstagnachmittag statt spät am Freitagabend.

Kontakt aufnehmen

Die passenden Arbeiten prüfen

Prüfen Sie unser Vorgehen anhand der zugehörigen Arbeiten auf dieser Seite. Fragen Sie nach den Abwägungen und den Bedingungen hinter jedem Ergebnis. GIS-SaaS-Deployments mit GitHub automatisiert — Bereitstellung, Daten und Berichte ohne Handarbeit

GIS-SaaS-Deployments mit GitHub automatisiert Kontakt aufnehmen →

GIS-SaaS-Deployments mit GitHub automatisiert

Bereitstellung, Datenverarbeitung und Berichtswesen einer GIS-SaaS-Anwendung mit GitHub Actions CI/CD, Python, Bash und SQL automatisiert.

20 GIS-Datenpipelines automatisiert ausgeliefert

Die Bereitstellung von 20 GIS-Datenpipelines und ETL-Prozessen für Anwendungsdaten automatisiert und dadurch Infrastrukturautomatisierung und Berichterstattung vereinfacht.

Abläufe überarbeitet und 8.000 Stunden eingespart

Interne Abläufe überarbeitet und durch bessere Softwarearchitektur, Systeme und Terminplanung 8.000 Stunden eingespart.

Infrastruktur vereinfacht und AWS verlassen

Die Infrastruktur eines internationalen Kunden überarbeitet, ungenutzte Dienste entfernt, Kapazitäten angepasst und Workloads von AWS weg verlagert. Die Gründerin berichtet von einer Senkung der Cloudkosten um den Faktor 10 ohne Produktivitätsverlust.

Den API-Vertrag aus der Datenbank generiert

Den API-Vertrag von der Datenbank aus nach außen generiert — OpenAPI, einen typisierten TypeScript-Client mit 44.076 Zeilen, 61 Mock-Handler und die von der UI durchgesetzten Grenzen — mit einer Prüfung an jedem Übergang, die Abweichungen zurückweist.

Die gesamte Kerninfrastruktur neu aufgebaut

Als eine der ersten Mitarbeiterinnen die gesamte Kerninfrastruktur und die unterstützenden Prozesse eines Green-Energy-SaaS-Start-ups von Grund auf entworfen und aufgebaut und damit die Grundlage für schnelles Wachstum gelegt.

Testsuite für Go, SQL und Browser aufgebaut

Eine mehrschichtige automatisierte Testsuite aufgebaut — 981 Go-Tests, 543 Frontend- und Browserspezifikationen, 494 verhaltensprüfende SQL-Tests — mit Mutationstests, eigenschaftsbasierten Tests und einer verbindlichen Barrierefreiheitsprüfung.

Repository-Prüfengine mit 268 Checks entwickelt

Die Prüfengine des Repositorys entwickelt — 268 registrierte Commit-Prüfungen, 277 Lint-Regeln und 15 eigene ESLint-Regeln — sowie 146 Tests der Prüfungen selbst, damit der Build den Standard durchsetzt und nicht erst das Review.

Eigene Fehlerüberwachung und Tracing entwickelt

Eigene Fehlerüberwachung und OpenTelemetry-Tracing entwickelt statt eingekauft — Bereinigung von Nutzdaten, Erkennung von Fehlerspitzen und Regressionen, Symbolauflösung und synthetisches Lebenszeichen — mit 11 Betriebsansichten.

Ihr Code auf Ihrem eigenen Server

Die eigene Git-Forge auf Soft Serve bereitgestellt, standardmäßig privat, ohne Weboberfläche und mit an Loopback gebundenem SSH-Port hinter einem Sprunghost; die vorgeschaltete Startseite als Build-Artefakt der Hauptwebsite statt als manuell gepflegte Kopie umgesetzt.

Commit-Prüfung mit 22 Lintern aufgebaut

Eine verbindliche Commit-Prüfung aus 22 einzeiligen Lintern und fünf ausführlicheren Prüfungen aufgebaut, ohne Warnungsstufe und ohne Inline-Unterdrückungen, für HTML, CSS, JavaScript, Python, YAML, Markdown, Shell, Links, Rechtschreibung, Geheimnisse und Typografie.

Ein Sicherheitsnetz, das keine halbe Stunde mehr kostet

Die browsergestützte Qualitätsprüfung der Website von 1.636 auf 615 Sekunden verkürzt, indem die Prüfungen in einem auf vier parallele Ausführungen begrenzten Worker-Pool nach längster Laufzeit zuerst eingeplant werden, nachdem die alphabetische Reihenfolge 320 statt 224 Sekunden benötigt hatte.

Website nach Gemini und Gopher gespiegelt

Gemini- und Gopher-Ausgaben aus derselben Website-Quelle entwickelt; die Leistung dokumentiert für diesen Stand 777 Dokumente je Format, wobei durch das Hinzufügen der Formate null HTML-Bytes geändert wurden.

Sitemap mit einheitlichem Zeitstempel korrigiert

Eine Sitemap korrigiert, in der 172 von 176 URLs denselben Änderungszeitpunkt trugen, indem das Datum nach der Feststellung aus der Git-Historie gelesen wurde, dass der Export jede Datei bei jedem Lauf neu schreibt.

Qualitätsprüfung mit 10.242 Zeilen selbst geprüft

10.242 Zeilen JavaScript der Qualitätsprüfung unter Formatter und Linter gebracht, nachdem sie sich als größter und zugleich ungeprüfter Codebestand des Repositorys erwiesen hatten; 13 Befunde behoben und keinen unterdrückt.

Generator mit 981 Tests und 92 % Abdeckung

Den Generator mit 981 Testfällen und mindestens 92 % Zweigabdeckung geprüft, Warnungen als Fehler behandelt und Idempotenz durch zwei vollständige Pipeline-Durchläufe aus einer leeren Datei nachgewiesen, wobei der zweite nichts ändern darf.

Jede Lint-Regel als Fehler aktiviert

Sämtliche Regeln des Python-Linters als Fehler aktiviert und 1.815 Befunde bis auf null abgearbeitet; jede der wenigen Ausnahmen schriftlich begründet und zwei davon durch eine Prüfung statt nur einen Kommentar abgesichert.

Die Prüfwerkzeuge selbst mit Tests geprüft

Tests für die Prüfwerkzeuge selbst geschrieben, nachdem klar war, dass ein nur mit fehlerfreien Eingaben getestetes Werkzeug irgendwann auch nach dem Lesen von nichts Erfolg meldet — einen Tippfehler eingebaut, um die Rechtschreibprüfung zu prüfen, und einen ID-Bereich aus der Datenbank statt einer Testkonstante übernommen.

Parser zur Prüfung aller FFI-Aufrufe geschrieben

Einen Parser geschrieben, der den echten C-Header mit 7.308 Zeilen liest und jede Aufrufstelle, jede Enum-Konstante sowie final für jede pointerbesitzende Klasse prüft, nachdem ein handgeschriebener Platzhalterheader Aufrufe dreier entfernter Funktionen kompilieren und linken ließ, die zur Laufzeit abstürzten.

Die ungenutzte Hälfte des Kerns erschlossen

Die zuvor ungenutzte Hälfte des Messaging-Kerns erschlossen — Sicherungsübertragung, verschwindende Nachrichten, Nachrichtenbearbeitung und erneutes Senden, verifizierte Einladungen, Proxys und Verschlüsselungsrichtlinien — und jeden Test ohne Mocks gegen die echte Bibliothek ausgeführt.

Sie suchen DevOps- und CI/CD-Automatisierung?

Beurteilen Sie anhand der verlinkten Arbeiten die einschlägige Erfahrung. Erzählen Sie uns, was Sie entwickeln möchten; wir können Umfang, Vorgehen bei der Umsetzung und Übergabe besprechen.

Kontakt aufnehmen Gesamtes Portfolio

Den Umfang vor der Beauftragung festlegen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Ihre Anforderungen besprechen

Erzählen Sie uns, was Sie von „DevOps- und CI/CD-Automatisierung“ benötigen, einschließlich des Termins und des Teams, das die Wartung übernehmen wird. Wir können einen praktikablen Weg besprechen.

Kontakt aufnehmen

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://software.engineer.company/de/services/devops-cicd/