Engineer Rådgivning

{ DevOps och CI/CD-automatisering }

Leverera ändringar utan dramatik

Testade, släppta och möjliga att backa, så att ändringar når kunderna en tisdagseftermiddag i stället för sent en fredagskväll.

Håller ni fortfarande på att reda ut vad ni behöver?

Berätta vad som inte fungerar eller vad ni försöker bygga. Vi säger vad det skulle kräva, ungefär vad det skulle kosta och om vi är rätt personer för uppgiften.

Från 1 400 DKK i timmen, exklusive moms — och mindre per timme vid en längre bokning.

Hör av er Se vad det kostar

Utvalt arbete som visar den här tjänsten.

Gör tjänsten till ett resultat

Låt ”DevOps och CI/CD-automatisering” stödja ett affärsmål. Vi kan knyta det tekniska arbetet till användarna, driften och besluten det ska stödja. Testade, släppta och möjliga att backa, så att ändringar når kunderna en tisdagseftermiddag i stället för sent en fredagskväll.

Hör av er

Granska det relevanta arbetet

Använd arbetet på sidan för att granska vårt angreppssätt. Fråga om avvägningarna och förutsättningarna bakom varje resultat. Förnyade interna processer, sparade 8 000 timmar — 8,000 sparade timmar genom en översyn av hur arbetet bedrevs

Förnyade interna processer, sparade 8 000 timmar Hör av er →

Förnyade interna processer, sparade 8 000 timmar

Gjorde en genomgripande översyn av de interna processerna och sparade 8 000 timmar genom att förbättra mjukvaruarkitekturen, systemen och effektiviteten i schemaläggningen.

Effektiviserade CI/CD och sparade 4 000 timmar

Effektiviserade CI/CD-processerna och sparade 4 000 timmar genom att införa automatisering i pipelines för mjukvaruutveckling.

Skrev 578 automationsmål i go-task

Skrev 578 automationsmål i go-task som täcker utvecklingslägena native, Docker och HTTPS samt lintning, testning, databas och driftsättning.

Satte ribban vid noll varningar i sex språk

Satte kvalitetsribban vid noll varningar i sex språk — Go, TypeScript, SQL, Python, Shell och Markdown — upprätthållen av pre-commit-hookar.

Automatiserade teamets samarbete

Automatiserade teamets samarbete, lösenordshantering samt uppgifts- och tidshantering och byggde ett halvautomatiskt system för att visa upp projekt, vilket höjde teamets produktivitet.

Byggde en testsvit för Go, SQL och webbläsare

Byggde en skiktad automatiserad testsvit — 981 Go-tester, 543 specifikationer för frontend och webbläsare, 494 beteendetester i SQL — med mutationstestning, egenskapsbaserade tester och en tillgänglighetsgrind.

Byggde repots kontrollmotor med 268 kontroller

Byggde repots kontrollmotor — 268 registrerade commit-kontroller, 277 lintregler och 15 egna ESLint-regler — plus 146 tester av själva kontrollerna, så att det är bygget som håller standarden, inte granskningen.

Drev ett löpande program för säkerhetsskanning

Etablerade ett kontinuerligt säkerhetsprogram — kodskanning, DAST, kontroller av beroenden och sårbarheter, generering av SBOM, skanning efter hemligheter och SHA-låsta actions — vid sidan av 21 skriftliga säkerhetsgranskningar.

Byggde en commit-grind med 22 linters

Byggde en commit-grind med 22 linters på en rad var plus fem som förtjänar ett eget stycke, utan varningsnivå och utan tillåtna inline-undantag, som täcker HTML, CSS, JavaScript, Python, YAML, Markdown, shell, länkar, stavning, hemligheter och typografi.

Ett skyddsnät som inte längre kostar en halvtimme

Kortade webbplatsens webbläsarstyrda kvalitetsgrind från 1 636 sekunder till 615 genom att schemalägga kontrollerna med den längsta först i en workerpool begränsad till fyra spår, efter att ha mätt att alfabetisk ordning kostade 320 sekunder mot 224.

Lintade själva kvalitetsgrinden på 10 242 rader

Förde in 10 242 rader JavaScript för kvalitetsgrinden under en formaterare och en linter efter att ha konstaterat att det var den största kodmassan i repot och den enda som inget läste, och åtgärdade 13 fynd utan att undertrycka något.

Gav generatorn 981 tester och 92 % täckning

Krävde 981 testfall och minst 92 % grentäckning av generatorn, med varningar behandlade som fel, och säkerställde idempotens genom att köra hela byggpipelinen två gånger från en tom fil och kräva att den andra körningen inte ändrade något.

Aktiverade varje lintregel på felnivå

Aktiverade varje regel i Python-lintern på felnivå och arbetade igenom 1 815 fynd ned till noll, där vart och ett av de få undantagen har ett skriftligt skäl och två av dem backas upp av en kontroll i stället för en kommentar.

Skrev tester för själva kontrollerna

Skrev tester för själva kontrollerna efter att ha konstaterat att en kontroll som bara matas med felfri indata en dag rapporterar felfritt för att den inte läste något — lade in en avsiktlig felstavning för att bekräfta att stavningskontrollen hittar den och hämtade ett id-intervall från databasen i stället för från ett tal i testet.

Upptäckte att hookarna och grinden var oense

Upptäckte att commit-hookarna och kvalitetsgrinden körde olika kontroller medan ett dokument lovade att de var desamma, genom att jämföra de två listorna i ett test — de fem som bara någonsin kördes för hand var just de som läste CV-texten.

Skrev en parser som verifierar varje FFI-anrop

Skrev en parser som läser den verkliga C-headern på 7 308 rader och verifierar varje anropsställe, varje enum-konstant och att varje klass som äger en pekare är final, efter att en handskriven platshållarheader hade låtit anrop till tre borttagna funktioner kompilera, länka och krascha.

Tog testsviten ned till 8,9 sekunder

Tog testsviten från sex tester över en gräns på sextio sekunder till 135 godkända på 8,9 sekunder genom att profilera huvudtråden och ta bort de två anrop som den befann sig i vid 3 989 av 4 017 stickprov.

Lät 4 repon följa en och samma historikstandard

Lät fyra repon följa en och samma historikstandard — konventionella commits, fria från emojier, utan attributionsrader, upprätthållen av en hook för commit-meddelanden — vid sidan av 48 instruktionsdokument som styr hur arbetet utförs.

Letar ni efter DevOps och CI/CD-automatisering?

Använd de länkade arbetena för att bedöma relevant erfarenhet. Berätta vad ni behöver bygga; vi kan diskutera omfattning, arbetssätt och överlämning.

Hör av er Hela portföljen

Avgränsa uppdraget före beställning

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Gå igenom era krav med oss

Berätta vad ni behöver inom ”DevOps och CI/CD-automatisering”, vilken tidsram ni önskar och vilket team som ska förvalta lösningen. Vi kan diskutera en praktisk väg framåt.

Hör av er

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://advisory.engineer.company/sv/services/devops-cicd/