Engineer

Byggde CI/CD-pipelines i GitHub Actions

Byggde CI/CD-pipelines i GitHub Actions med en distroless-image för frontenden i produktion och uppflyttning mellan flera miljöer.

2025

en pipeline, tre miljöer, en distroless-image

Situation. Att leverera borde inte hänga på att någon kommer ihåg stegen, och det som till slut körs i produktion borde inte vara en fet allmän container som bär på ett skal och en pakethanterare som den aldrig kommer att använda — det är bara attackyta som ligger där utan anledning.

Uppgift. Vägen från commit till körning i Azure skulle bli automatisk, och produktionsavbildningarna skulle hållas så små och låsta som varje arbetslast tillåter.

Handling. Pipelinen är GitHub Actions. Separata arbetsflöden hanterar kodkvalitetsgrinden, testerna och driftsättningarna per miljö, med CodeQL, beroendegranskning och ett SBOM-steg vid sidan av, så att ingenting når en miljö utan att först ha passerat kontrollerna. Avbildningarna byggs i flera steg, och basen för varje del valdes efter sina egna meriter i stället för som ett enda generellt val: frontenden levereras på en distroless-avbildning (gcr.io/distroless/cc-debian13 — inget skal, ingen pakethanterare), Go-API:et på en slimmad Alpine och databasavbildningen på postgres-slim. Ett bygge förs vidare genom miljöerna längs en fastställd väg i stället för manuellt.

Resultat. Releaser slutade vara en försiktig manuell ritual och blev en rutinmässig, tråkig händelse, vilket är precis vad man vill ha av releaser. Frontenden i produktion körs på ungefär så lite som det går att ge den, kontrollerna fångar problem innan de landar, och ”driftsättning” är något som pipelinen gör snarare än något som någon svettas sig igenom.

Använd erfarenheten i ert projekt

Använd ”Byggde CI/CD-pipelines i GitHub Actions” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Testade, släppta och möjliga att backa, så att ändringar når kunderna en tisdagseftermiddag i stället för sent en fredagskväll.

Se våra tjänster Hör av er →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Byggde CI/CD-pipelines i GitHub Actions — en pipeline, tre miljöer, en distroless-image

Se våra tjänster Hör av er →

Behöver du en ordentlig CI/CD-pipeline?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: DevOps-ingenjör. Det bygger på DevOps, Molntjänster, Containrar (Docker/Kubernetes), Automatisering och CI/CD, Tillförlitlighet och säkerhetskopiering, Säkerhet och Test och kvalitetssäkring. Och nu när ni är här: vi är ganska bra på DevOps och CI/CD-automatisering och Containerisering och orkestrering.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

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 →

Behöver du en ordentlig CI/CD-pipeline?

Fråga om besluten bakom ”Byggde CI/CD-pipelines i GitHub Actions”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

Se våra tjänster

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://engineer.company/sv/portfolio/built-github-actions-ci-cd-pipelines/