Engineer

Byggde egen felövervakning och spårning

Byggde egen felövervakning och OpenTelemetry-spårning i stället för att köpa dem — sanering av nyttolaster, upptäckt av toppar och regressioner, symbolisering och ett syntetiskt hjärtslag — bakom 11 operatörsvyer.

2025

11

operatörsvyer över övervakning
som byggts i egen regi, inte köpts in

Situation. Skillnaden mellan en plattform som är uppe och en plattform som fungerar är om någon alls skulle få veta det. Ett fel som en användare stöter på klockan elva på kvällen, på en sida som ingen testar, är osynligt om inte något går och samlar in det. Det vanliga svaret är att köpa en hostad felspårningstjänst, vilket är ett bra svar — och det innebär också att plattformens egna fel, stackspår och användarkontext går vidare till en tredje part.

Uppgift. Fel och spårdata måste samlas in, grupperas och göras möjliga att åtgärda, utan att plattformens inre lämnade plattformen.

Handling. Två delar byggdes. OpenTelemetry sköter spårningen över OTLP, så att ett långsamt anrop kan följas genom frontenden, API:t och databasen i stället för att gissas fram. Bredvid den finns en egenutvecklad felpipeline — en errmon-tjänst och en ingest-tjänst — som sanerar nyttolaster före lagring, grupperar fel till återkommande problem i stället för en platt lista, upptäcker toppar och regressioner med en karenstid så att en dålig driftsättning inte larmar någon fyrtio gånger, översätter minifierade stackspår från frontenden tillbaka till läsbar kod och kör ett syntetiskt hjärtslag för att bevisa att pipelinen själv lever. Allt hamnar i databasen som felhändelser, felgrupper, en inkorg, API-latens och stackprov, och det presenteras i 11 operatörsvyer, däribland en för servicenivåmål.

Resultat. Fel blir en kö som någon kan beta av, och en regression gör sig till känna i stället för att upptäckas av en användare. Att bygga i stället för att köpa kostade verklig tid och innebär att det är ännu en sak att underhålla — en köpt felspårare hade varit igång samma eftermiddag. Det man fick i gengäld var att inget känsligt lämnar plattformen, och att larmreglerna passar den här plattformen i stället för en generisk.

Använd erfarenheten i ert projekt

Använd ”Byggde egen felövervakning och spårning” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

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 egen felövervakning och spårning — 11 operatörsvyer över övervakning som byggts i egen regi, inte köpts in

Se våra tjänster Hör av er →

Skulle du märka det om din app gick sönder i natt?

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: Site Reliability Engineer. Det bygger på Backendutveckling, DevOps, Infrastruktur, Tillförlitlighet och säkerhetskopiering, Övervakning och observerbarhet och Säkerhet. Och nu när ni är här: vi är ganska bra på Backend- och API-utveckling, DevOps och CI/CD-automatisering, Tillförlitlig drift och övervakning och Utveckling av skräddarsydda AI-agenter.

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

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Skulle du märka det om din app gick sönder i natt?

Fråga om besluten bakom ”Byggde egen felövervakning och spårning”. 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-first-party-error-monitoring-and-tracing/