Engineer

Overvågning

at holde øje med et kørende system, så det siger til først

Hvad det er

Overvågning, også kaldet monitorering, er løbende at læse, hvad et system i drift rapporterer om sig selv: noget holder hele tiden rapporterne op mod grænser, som nogen har trukket, og afgør, hvornår et menneske skal have besked. Googles bog om Site Reliability Engineering definerer overvågning som at indsamle, behandle, samle og vise kvantitative data om et system i realtid: hvor mange forespørgsler der kommer ind og af hvilken slags, hvor mange der fejler og hvordan, hvor lang tid de tager, og hvor længe hver server har kørt. Målet er en bestemt rækkefølge: systemet siger til, at det er i knibe, før en bruger gør.

Data kommer fra to retninger. White-box-overvågning læser det, et system selv viser om sit indre, som et metrics-endpoint, længden af en kø eller et antal genforsøg, og kan derfor se problemer vokse, før nogen mærker dem. Black-box-overvågning står udenfor og opfører sig som en bruger: den henter en side, slår et navn op eller gennemfører et login og melder, om det lykkedes, og hvor lang tid det tog. Den ser kun det, der allerede er gået i stykker, og netop derfor er den en troværdig grund til at vække nogen.

Det meste, et overvågningssystem gemmer, er tal målt over tid, fordi de er billige at gemme og nemme at holde op mod en tærskel. SRE-bogen udpeger fire af dem som stedet at begynde for alt, der betjener brugere: latenstid, den tid en forespørgsel tager; trafik, hvor meget efterspørgsel der kommer ind; fejl, andelen af forespørgsler, der fejler; og mætning, hvor tæt den travleste ressource er på at være fuld.

Omkring tallene sidder to slags output. Et dashboard er til at kigge på, når nogen allerede har en mistanke. En alarm er til at afbryde, når alle har travlt med noget andet. I Prometheus-økosystemet evaluerer serveren reglerne, og Alertmanager grupperer, fjerner dubletter og sender det videre, der udløses.

To naboer er lette at forveksle med den. Logning skriver enkelte hændelser ned, én linje for hver, og er det, man læser efter en alarm for at finde ud af hvorfor. Observability, som OpenTelemetry-projektet beskriver det, er den bredere evne til at stille et system nye spørgsmål udefra. Overvågning er den smallere og ældre praksis: spørgsmål, der er besluttet på forhånd, stillet løbende, med et svar, der kan alarmere en person.

Brug begrebet i en beslutning

Brug »Hvad er overvågning?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Overvågning og alarmer, der siger til, mens fejlen stadig er lille, og tallene, der viser, hvor tit den bliver ved med at være det.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har bygget det første observerbarhedsdashboard — virksomhedens første overblik over sine egne systemer

Har bygget det første observerbarhedsdashboard Kontakt os →

Hvorfor det betyder noget

Alternativet er at høre om problemer fra dem, de rammer, på hvilket tidspunkt de end opdager dem. Overvågning flytter det øjeblik frem og lægger det hos en, der kan handle.

Den knappe ressource er opmærksomhed snarere end software. En alarmbesked afbryder arbejde om dagen og søvn om natten, og SRE-bogen er klar: kommer de for tit, skimmer folk dem, tvivler på dem og ignorerer dem til sidst, nogle gange også den reelle alarm, der gemmer sig i støjen. Prometheus-projektets råd kan stå i én sætning: hold alarmeringen enkel, alarmér på symptomer, hav gode konsoller til at finde årsagerne, og undgå alarmbeskeder, der ikke kræver handling.

Det gør den centrale beslutning til et spørgsmål om, hvad brugerne mærker. En disk på 85 % er en årsag; en betaling, der fejler, er et symptom. At alarmere på symptomer holder antallet af regler nede, fordi der er langt færre måder, brugere kan komme til skade på, end måder, maskiner kan opføre sig forkert på. Årsager hører stadig hjemme på dashboards og i tickets, hvor de hjælper med at finde fejlen uden at vække nogen. Kapacitet er den ærlige undtagelse: en ressource tæt på fuld gør ingen ondt i dag, men får snart brug for en person.

Så kommer spørgsmålet om at drive det selv eller leje det. At drive det selv betyder et tidsserielager, en alarmrouter og en person, der holder begge sunde. En hostet tjeneste tager det arbejde væk og fakturerer efter den mængde, der sendes, så hver ny vært dukker op på en faktura. SRE-bogen nævner, at selv hos Google har et team på 10–12 ingeniører typisk en eller to, hvis hovedopgave er overvågning. Uanset hvad har det sin egen vedligeholdelse og skal dimensioneres efter det system, det holder øje med.

Sådan ser det ud i praksis

To af grundlæggerens tidligere stillinger og virksomhedens egen vært viser den samme idé i vidt forskellig skala.

Fra 2015 til 2022, i en rolle som infrastrukturingeniør, var opgaven 24/7-support til omkring tusind servere på en IPTV/OTT-streamingplatform, plus et pænt antal kundeejede systemer, for kunder i Kina, USA og Tyskland. Med kunder i tre tidszoner fandtes der ingen stille time, og et problem klokken 3 om natten lokalt var primetime et andet sted. Meget af arbejdet var at bemærke noget, der drev af, før det blev et nedbrud, og problemerne blev håndteret, før de nåede en seers skærm. Her ser succes ud som ingenting, der sker, og overvågning lader et team vide det frem for at håbe det.

I en rolle som databaseingeniør fra 2022 til 2024, på tværs af 100 databaser i PostgreSQL, MS SQL og BigQuery, reducerede automatiseret overvågning nedetiden med 50 %.

At overvåge virksomhedens egen vært til offentligt tilgængelige tjenester betyder at arbejde i den anden ende af skalaen. Én maskine med 512 MB driver de offentligt tilgængelige tjenester, hvor omkring 464 MB af hukommelsen kan bruges, og hukommelsen er det argument, der afgør hver designbeslutning på den. Rapportering er her et pull frem for en agent, en pipeline og et dashboard: syv kommandoer læser værten og gengiver Markdown, ændrer intet og kører kun, når nogen beder om det. Beslutningen om onion-tjenesten viser metoden: da spørgsmålet kom op, blev der taget stikprøver på værten i et døgn, og aflæsningerne afgjorde det. I de registrerede stikprøver var lavpunktet for tilgængelig hukommelse omkring 310 MB af 464, swap lå på 2,6 %, og processoren var inaktiv 99,7 % af tiden. I den størrelse er overvågning en måling, der tages i det øjeblik, en beslutning har brug for den.

Hvor det går galt

Alarmerne kommer så tit, at folk holder op med at læse dem. Hver driftshændelse efterlader en regel, mange af dem udløses af årsager, der går over af sig selv, og efter nogle måneder bliver den reelle alarm skimmet forbi sammen med resten. Send kun alarmbeskeder om det, brugerne ville mærke, ét sted i stakken, og send alt andet til en ticketkø eller et dashboard. Lad derefter betingelsen holde et stykke tid, før den udløses, så et blip, der går over på et minut, lader alle sove:

```
groups:
    - name: checkout
      rules:
          - alert: CheckoutErrorsHigh
            expr: job:checkout_errors:ratio5m > 0.02
            for: 10m
            labels:
                severity: page
```

Overvågningen stopper, og dens tavshed bliver læst som gode nyheder. En regel, der aldrig udløses, ligner præcis et system, der aldrig går i stykker. Går Prometheus-serveren, Alertmanager eller mailrelæet ned, stopper alle alarmer på én gang. Prometheus' eget råd er at alarmere på selve overvågningen og helst at teste hele vejen, så ét tjek beviser, at en alarm kan nå fra regel til indbakke. Planlagte jobs falder i samme fælde: en backup, der holder op med at køre, giver ingen fejl, den holder bare op med at efterlade spor. Send en alarmbesked, når den seneste succes er for gammel, og giv plads til mindst to hele kørsler; for et job, der kører hver 4. time og tager en time, foreslår projektet 10 timer:

```
time() - backup_last_success_timestamp_seconds > 10 * 3600
```

Gennemsnittet siger, at alt er godt, mens nogle brugere venter. Gennemsnit er nemme at tegne, og de skjuler halen. SRE-bogens eksempel er en tjeneste med et gennemsnit på 100 ms ved 1.000 forespørgsler i sekundet, hvor 1 % af forespørgslerne sagtens kan tage 5 sekunder; når en side afhænger af flere sådanne tjenester, kan én backends 99. percentil blive frontendens median. Hurtige fejl gør det værre ved at trække gennemsnittet ned. Registrér latenstid i buckets, alarmér på en høj percentil, og hold fejlede forespørgsler adskilt fra de vellykkede.

Alle interne tal er grønne, mens brugerne ikke kan nå tjenesten. White-box-metrikker kommer indefra systemet og deler derfor dets blinde vinkler: et udløbet certifikat, en ødelagt DNS-post eller en ændring i firewallen kan lade hver eneste af dem se sund ud. Et black-box-tjek fra en maskine uden for systemets netværk fanger det, der ellers er usynligt, og virker stadig, når den interne overvågning går ned med alt andet.

Læs mere

- Prometheus: Overview — hvad et metrikbaseret overvågningssystem indsamler, hvilke dele det består af, og hvor præcision for hver enkelt hændelse kræver noget andet.
- Alerting — Prometheus-projektets råd om, hvad der skal udløses alarmer for, fra onlinetjenester og batchjobs til overvågning af overvågningen.
- Alertmanager — hvordan alarmer renses for dubletter, grupperes, gøres tavse, hæmmes og sendes videre til en modtager.
- Monitoring Distributed Systems — kapitel 6 i Googles bog om Site Reliability Engineering: symptomer og årsager, de fire gyldne signaler og halen.
- Observability primer — OpenTelemetrys gennemgang af signaler, SLI'er og SLO'er, og hvor observability rækker videre end overvågning.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er overvågning?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/monitoring/