Engineer

Övervakning

att bevaka ett system i drift så att det säger till först

Vad det är

Övervakning är den löpande avläsningen av det som ett system i drift rapporterar om sig självt: något håller hela tiden rapporterna mot gränser som någon har dragit och avgör när en människa behöver få veta. Boken Site Reliability Engineering från Google definierar övervakning som att samla in, bearbeta, aggregera och visa kvantitativa data i realtid om ett system: hur många förfrågningar som kommer in och av vilka slag, hur många som misslyckas och hur, hur lång tid de tar och hur länge varje server har varit igång. Målet är en ordningsföljd: systemet säger att det har problem innan en användare gör det.

Data kommer från två håll. Övervakning av typen white-box läser det som ett system visar om sitt eget inre, t.ex. de systemmätvärden det publicerar, längden på en kö eller antalet omförsök, och kan därför se problem som byggs upp innan någon känner av dem. Övervakning av typen black-box står utanför och beter sig som en användare: den begär en sida, slår upp ett namn eller genomför en inloggning, och rapporterar om det fungerade och hur lång tid det tog. Den ser bara det som redan är trasigt, och det är just det som gör den till ett pålitligt skäl att väcka någon.

Det mesta som ett övervakningssystem sparar är tal som samplas över tid, eftersom tal är billiga att spara och lätta att hålla mot ett tröskelvärde. Samma kapitel i SRE-boken pekar ut fyra av dem som stället att börja för allt som betjänar användare: latens, den tid en förfrågan tar; trafik, hur mycket efterfrågan som kommer in; fel, andelen förfrågningar som misslyckas; och mättnad, hur nära den mest belastade resursen är att bli full.

Runt talen finns två sorters utdata. En dashboard är till för att titta, när någon redan misstänker något. Ett larm är till för att avbryta, när alla är upptagna med annat. I ekosystemet kring Prometheus utvärderar servern reglerna, och

Alertmanager grupperar, tar bort dubbletter och dirigerar det som utlöses.

Två närliggande begrepp är lätta att blanda ihop med övervakning. Loggning skriver ned enskilda händelser, en rad per sak som hände, och är det någon läser efter ett larm för att ta reda på varför. Observerbarhet, som OpenTelemetry-projektet ramar in det, är den bredare förmågan att ställa nya frågor till ett system utifrån. Övervakning är den smalare och äldre praktiken: frågor som bestäms i förväg, ställs hela tiden och har ett svar som kan väcka en människa.

Använd begreppet i ett beslut

Använd ”Vad är övervakning?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Övervakning och larm som säger till medan ett fel fortfarande är litet, och siffrorna som visar hur ofta det förblir så.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Byggde företagets första observerbarhetsdashboard — företagets första överblick över de egna systemen

Byggde företagets första observerbarhetsdashboard Hör av er →

Varför det spelar roll

Alternativet är att höra om problem från dem som drabbas, vid vilken timme de än märker det. Övervakning flyttar det ögonblicket tidigare och lämnar det till någon som kan agera.

Det som tar slut först är uppmärksamheten, inte verktygen. Ett larm till jouren avbryter arbete på dagen och sömn på natten, och SRE-boken säger rakt ut vad som händer när sådana larm kommer för ofta: folk skummar dem, ifrågasätter dem och slutar till sist bry sig om dem, ibland även det verkliga larmet som gömmer sig i bruset. Vägledningen från Prometheus-projektet ryms i en mening: håll larmen enkla, larma på symtom, ha bra konsoler för att hitta orsakerna och undvik larm där det inte finns något att göra. Varje regel som kan väcka någon är ett löfte om att en människa har något brådskande att göra när den utlöses.

Det gör det centrala beslutet till ett beslut om vad användarna märker. En disk som är fylld till 85 % är en orsak; en betalning som misslyckas är ett symtom. Att larma på symtom håller nere antalet regler, eftersom det finns långt färre sätt för användare att drabbas än för maskiner att krångla. Orsakerna hör fortfarande hemma på dashboards och i ärenden, där de hjälper felsökningen utan att väcka någon. Kapacitet är det ärliga undantaget: en resurs som snart är full skadar ingen i dag, men inom kort behöver den en människa.

Sedan kommer frågan om att driva det själv eller hyra det. Att driva det själv betyder ett lager för tidsserier, en router för larm och någon som håller båda friska. En hostad tjänst tar bort det arbetet och fakturerar efter mängden data som skickas, så varje ny värddator syns på en faktura. SRE-boken noterar att även på Google har ett team med 10–12 ingenjörer oftast en eller två vars huvuduppgift är övervakning. Hur som helst är det ett system med eget underhåll, som ska dimensioneras efter systemet det vakar över.

Så ser det ut i praktiken

Två av grundarens tidigare roller och företagets egen värddator visar samma idé i vitt skilda skalor.

Som infrastrukturingenjör 2015–2022 bestod jobbet i support dygnet runt för ungefär 1 000 strömningsservrar för IPTV/OTT, plus ett stort antal kundägda system, för kunder i Kina, USA och Tyskland. Med kunder i tre tidszoner fanns det aldrig en lugn timme, och ett problem kl. 3 på natten lokal tid var bästa sändningstid någon annanstans. En stor del av arbetet bestod i att märka när något började glida innan det blev ett avbrott, och problemen hanterades innan de nådde en tittares skärm. För en sådan tjänst ser framgång ut som att ingenting händer, och övervakningen är det som låter ett team veta det i stället för att hoppas.

I rollen som databasingenjör 2022–2024, för 100 databaser i PostgreSQL, MS SQL och BigQuery, minskade automatiserad övervakning driftstoppen med 50 %.

Att övervaka företagets egen värddator för publika tjänster innebär att arbeta i andra änden av skalan. En enda maskin med 512 MB driver de publika tjänsterna, med ungefär 464 MB användbart minne, och minnet är det argument som avgör varje designbeslut på den. Rapporteringen är pull-baserad i stället för en agent, en pipeline och en dashboard: sju kommandon som läser värddatorn och renderar Markdown, inte ändrar någonting och bara körs när någon ber om det. Beslutet om onion-tjänsten visar metoden: när frågan kom upp samlades stickprov från värddatorn under ett dygn, och mätvärdena avgjorde den. I de dokumenterade stickproven var den lägsta nivån för tillgängligt minne ungefär 310 MB av 464, swapanvändningen låg på 2,6 % och processorn var i vila 99,7 % av tiden. I den storleken är övervakning en mätning som görs i det ögonblick ett beslut behöver en.

Var det går fel

Jourlarmet går så ofta att folk slutar läsa det. Varje incident lämnar en regel efter sig, många av dem utlöses av orsaker som går över av sig själva, och inom några månader skummas det verkliga larmet förbi tillsammans med resten. Larma bara på det som användarna skulle märka, på ett ställe i stacken, och skicka allt annat till en ärendekö eller en dashboard. Låt sedan villkoret hålla i sig en stund innan det utlöses, så att en kort topp som försvinner inom en minut inte väcker någon:

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

Övervakningen stannar, och tystnaden läses som goda nyheter. En regel som aldrig utlöses ser exakt ut som ett system som aldrig går sönder. Om Prometheus-servern, Alertmanager eller e-postreläet går ned, upphör varje larm på en gång. Prometheus rekommenderar att man larmar på övervakningen i sig, och ännu hellre testar hela vägen från ände till ände, så att en kontroll bevisar att ett larm kan ta sig från regel till inkorg. Schemalagda jobb går i samma fälla: en säkerhetskopiering som slutar köras ger inget fel, den slutar bara lämna spår. Larma när den senaste lyckade körningen är för gammal, med marginal för minst två hela körningar; för ett jobb som körs var fjärde timme och tar en timme föreslår projektet 10 timmar:

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

Medelvärdet säger att allt är bra medan vissa användare väntar. Medelvärden är lätta att rita som grafer, och de döljer svansen. Exemplet i SRE-boken är en tjänst med ett medelvärde på 100 ms vid 1 000 förfrågningar i sekunden, där 1 % av förfrågningarna lätt kan ta 5 sekunder; när en sida är beroende av flera sådana tjänster kan den 99:e percentilen för en enda backend bli medianen för frontenden. Snabba fel gör det värre, eftersom ett fel som kommer fort drar ned medelvärdet. Spara latensen i intervall, larma på en hög percentil och håll isär misslyckade förfrågningar från lyckade.

Varje internt tal är grönt medan användarna inte når tjänsten. Mätvärden av typen white-box kommer inifrån systemet och delar därför dess blinda fläckar: ett utgånget certifikat, en trasig DNS-post eller en ändring i en brandvägg kan lämna vartenda ett av dem friskt. En black-box-kontroll från en maskin utanför systemets nätverk fångar det som annars är osynligt, och fortsätter att fungera när den interna övervakningen går ned tillsammans med allt annat.

Läs mer

- Prometheus: Overview — vad ett övervakningssystem byggt på mätvärden samlar in, vilka delar det består av och när precision per händelse kräver något annat.
- Alerting — vägledningen från Prometheus-projektet om vad som ska larma, från onlinetjänster och batchjobb till övervakningen av övervakningen.
- Alertmanager — hur larm dedupliceras, grupperas, tystas, undertrycks och dirigeras till en mottagare.
- Monitoring Distributed Systems — kapitel 6 i boken Site Reliability Engineering från Google: symtom och orsaker, de fyra gyllene signalerna och svansen.
- Observability primer — hur OpenTelemetry beskriver signaler, SLI och SLO, och var observerbarhet når längre än övervakning.

Ordlista

Välj utifrån era faktiska förutsättningar

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 →

Gör begreppen praktiska

Fråga hur ”Vad är övervakning?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

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/glossary/monitoring/