Engineer

Driftsvarsel

en melding et system sender når noe trenger en person

Hva det er

Et driftsvarsel er et automatisk signal om at en betingelse som overvåkes, har holdt seg sann lenge nok til at det er verdt å forstyrre noen. I daglig tale heter det bare varsel eller alarm, og et varsel som er alvorlig nok til å vekke den som har vakt, kaller mange en page, etter personsøkeren det en gang gikk til.

Ordet dekker to ulike ting, og Prometheus holder dem fra hverandre på en måte som gjør skillet lett å se. En varslingsregel (alerting rule) i Prometheus lager varsler: ett for hvert unike sett med etiketter som uttrykket gir et resultat for. Varslene går videre til et eget program, Alertmanager, som gjør dem om til meldinger: en e-post, en chatmelding, et kall til en vakttjeneste. Prosjektet sier rett ut at varslingsregler alene er langt fra et fullstendig varslingssystem. Skillet betyr noe fordi tallene sprer seg. Femti varsler kan komme fram som én melding, ett varsel kan gi en påminnelse med noen timers mellomrom, og et dempet varsel når fram til ingen i det hele tatt, mens det fortsatt står i firing.

Livsløpet er kort. Så lenge uttrykket returnerer tomt, er varselet inactive. Første gang det gir et resultat, blir varselet pending, og der blir det stående så lenge regelens `for` -ledd krever. Er betingelsen fortsatt sann når ventetiden er ute, går varselet over i firing og sendes videre for levering. Når betingelsen slutter å være sann, blir varselet resolved. Et nyere ledd, `keep_firing_for` , holder et varsel i firing en stund etter at betingelsen er borte, og dokumentasjonen beskriver det som et vern mot flapping og falske meldinger om at problemet er løst. Prometheus lagrer dessuten tilstandene pending og firing som en tidsserie kalt `ALERTS` , så du kan kjøre spørringer mot selve varslene og tegne dem i grafer.

En regel som venter tjue minutter før den går av, og ti minutter til før den slipper, ser slik ut:

```
groups:
    - name: exports
      rules:
          - alert: ExportQueueStalled
            expr: exports_waiting > 0 and rate(exports_done_total[15m]) == 0
            for: 20m
            keep_firing_for: 10m
            labels:
                severity: page
                team: billing
            annotations:
                summary: "{{ $value }} exports queued and none finishing"
```

Etiketter (labels) og annotasjoner ligner på hverandre, men gjør ulike jobber. Etikettene er en del av det som gjør et varsel til akkurat dette varselet og ikke et annet, og Alertmanager leser dem for å avgjøre hvor det skal: `severity: page` kan gå til en telefon, mens `severity: ticket` havner i en sakskø. Annotasjonene bærer ordene en person trenger når meldingen kommer: en sammendragslinje, en lengre beskrivelse, en lenke til runbooken som sier hva som skal gjøres først. Annotasjoner kan bygges med maler som setter inn varselets etiketter og verdien som utløste det, og det er derfor sammendraget over kan sitere et tall.

Driftsvarsel og driftshendelse er to ord av en grunn. Et varsel melder at en betingelse holdt seg sann. Om betingelsen utgjør en driftshendelse, er en vurdering et menneske må gjøre.

Bruk begrepet i en beslutning

Bruk «Hva er et driftsvarsel?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Overvåking og varsler som sier fra mens en feil fortsatt er liten, og tallene som viser hvor ofte den holder seg liten.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Bygde selskapets første observerbarhetsdashbord — selskapets første innsyn i egne systemer

Bygde selskapets første observerbarhetsdashbord Ta kontakt →

Hvorfor det er viktig

Av alt overvåkingen produserer, er varselet det eneste som oppsøker deg av seg selv. Metrikker ligger i lagringen, et dashbord venter på en skjerm og en logg venter på et søk, mens et varsel trenger seg inn i noens kveld. Det gjør hver varslingsregel til et lite, stående krav på en persons oppmerksomhet, og kravet

er bare rimelig hvis personen kan gjøre noe med det. Prometheus-prosjektets egne råd om varsling sier det slik: unngå å tilkalle noen når det ikke finnes noe å gjøre, og hold varslingen enkel.

Den andre grunnen er regning. Én årsak utløser som regel mange regler samtidig. Når en nettverkspartisjon skiller halvparten av en tjenestes instanser fra databasen, kan hver instans sende sitt eget varsel, og dokumentasjonen til Alertmanager bruker nettopp det tilfellet til å vise hundrevis av varsler som blir til én melding. Alertmanager samler varsler som deler etikettene som er nevnt i `group_by`, og venter `group_wait` (30 sekunder som standard) så den første meldingen rommer hele puljen. Deretter sender den oppdateringer for gruppen høyst hvert `group_interval` (5 minutter), og gjentar en melding som fortsatt er i firing, etter `repeat_interval` (4 timer). Undertrykking (inhibition) går et steg lenger og holder noen varsler tilbake mens et annet er i firing, så et varsel om at en hel klynge er utilgjengelig, kan stilne varslene fra hver tjeneste inni den. Dempinger (silences) gjør varsler som matcher et sett etiketter, stumme i en fast periode, typisk under planlagt vedlikehold.

Den tredje grunnen er slutten på historien. De fleste mottakere sender en melding om at varselet er resolved når det går over, men e-postmottakerens innstilling `send_resolved` står som standard til false. Et team som varsler på e-post, kan dermed se problemet meldt og aldri få vite at det tok slutt.

Slik ser det ut i praksis

Selskapets egen vert bruker en alarm som reagerer på et manglende signal. De 464 MB med minne betjener en webserver og en forge, og minnemangel og en full disk er to viktige feilrisikoer. En timer på verten pinger en ekstern tjeneste bare så lenge terskelkontrollene for tilgjengelig minne og ledig diskplass består, hvert femtende minutt med litt jitter, og den eksterne tjenesten slår alarm når de forventede pingene uteblir. Feil som hindrer skriptet i å kjøre eller nå tjenesten, blant annet et tapt nettverk eller en mislykket oppstart, gir et manglende ping. Et mottatt signal bekrefter at skriptet kjørte, og at terskelkontrollene for tilgjengelig minne og ledig diskplass ble bestått; det dokumenterer ikke vertens generelle tilstand eller at nettstedet svarer. Overvåking kan avdekke varselsignaler, men tidspunktet avhenger av belastningen og forholdene som observeres.

Den samme omvendingen kan beskytte planlagte jobber. I backupimplementeringen har sikkerhetskopiering og gjenoppretting hvert sitt suksessignal. Skriptene og systemd-enhetene støtter krypterte øyeblikksbilder utenfor verten, integritetskontroller og en månedlig test som gjenoppretter en kjent fil; et suksessignal sendes først når den aktuelle arbeidsflyten lykkes. Det er utformingen av varslingsveien, ikke bevis for at den var aktiv. Implementasjonsregistreringen datert 2026-10-27 sier at tidsurene var deaktivert uten et konfigurert lager, og dokumenterer ingen fullført gjenoppretting under reelle driftsforhold. Stillhet kan utløse et eksternt varsel når arbeidsflyten og overvåkingen er aktivert; konfigurasjonen alene viser ikke at de kjører.

Omvendte varsler har en egen feiltilstand, og den dukket opp da plattformen skulle vokse fra én vert til to. Å la begge vertene dele én dødmannsknapp ville ha svekket nettopp den varslingen den skulle utvide: så lenge én av vertene fortsetter å pinge, holder sjekken seg grønn mens den andre er død. Hver knapp ble en verdi per vert, og knappene til den eksisterende verten ble med vilje stående tomme. Da sier konfigurasjonen at koblingen finnes, og at verdien fortsatt må fylles inn.

Hvor det går galt

En målt verdi havner i en etikett. Fordi etikettene identifiserer et varsel, gjør en etikett med det aktuelle tallet hver endring av tallet til et nytt varsel, og det forrige blir resolved. Mottakeren ser en strøm av par med firing og resolved for ett og samme problem. Bruk etiketter til identitet og ruting, og legg tall i annotasjonene.

Varselet flapper. En betingelse som svever rundt terskelen, går av, blir resolved og går av igjen med noen minutters mellomrom, og hver melding om at det er over, forteller leseren noe som er usant. Gjør `for` lengre så et blipp forsvinner før noen hører om det, og bruk `keep_firing_for` så varselet holder seg i firing gjennom en kort dukkert under terskelen.

En demping lever lenger enn grunnen til den. Noen demper et varsel for vedlikehold med en bred matcher og en romslig sluttid, og neste ukes ekte feil kommer inn dempet. Gjør dempingen så smal som arbeidet tillater, gi den en sluttid nær det tidspunktet arbeidet skal være ferdig, og skriv hvem som satte den og hvorfor i kommentarfeltet Alertmanager har til det.

En bred rute sluker de spesifikke. Alertmanager går gjennom rutingtreet i rekkefølge og stopper ved det første barnet som matcher, med mindre ruten setter `continue: true`. En oppsamlingsrute plassert over et teams egen rute tar med seg det teamets varsler. Legg de smaleste rutene først, og test treet med eksempeletiketter før du stoler på det.

Alt haster. Disk på 70 %, et sertifikat som går ut neste måned, og en betalingstjeneste som returnerer feil, tilkaller alle den samme telefonen, og snart blir telefonen ignorert. Tilkall bare for betingelser en person må handle på nå, send resten til en sakskø eller et daglig sammendrag, og slett en regel som har gått av i månedsvis uten at noen har gjort noe med den.

Les mer

- Alerting overview — hvordan varslingsreglene i Prometheus og Alertmanager deler arbeidet, og stegene for å sette opp begge.
- Alerting rules — regelsyntaksen, med `for`, `keep_firing_for`, etiketter, annotasjoner og maler.
- Alertmanager — gruppering, undertrykking, dempinger og drift av flere Alertmanager-instanser for høy tilgjengelighet.
- Alertmanager configuration — rutingtreet, grupperingstidene, undertrykkingsreglene og innstillingene for hver mottaker, bl.a. `send_resolved`.
- Alerting practices — prosjektets egne råd om hva som fortjener å tilkalle noen, og hva som hører hjemme et annet sted.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et driftsvarsel?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/alert/