Engineer

Alarm

en besked, et system sender, når noget kræver et menneske

Hvad det er

En alarm er et automatisk signal om, at en betingelse, nogen holder øje med, har været opfyldt så længe, at det er værd at forstyrre et menneske. I driftsteams hører man lige så tit det engelske »alert«, og en alarm, der er alvorlig nok til at vække den vagthavende, kaldes ofte en »page«.

Ordet dækker to forskellige ting, og Prometheus holder dem så tydeligt adskilt, at forskellen er let at se. En alarmregel i Prometheus skaber alarmer, én for hvert sæt labels, som udtrykket giver et resultat for. Alarmerne går videre til et separat program, Alertmanager, der gør dem til notifikationer: en mail, en chatbesked, et opkald gennem en vagtplanstjeneste. Projektet skriver selv, at alarmregler alene ikke udgør et komplet notifikationssystem. Opdelingen betyder noget, fordi tallene skilles ad. Halvtreds alarmer kan ende som én besked, én alarm kan give en påmindelse med få timers mellemrum, og en alarm, der er gjort lydløs, når frem til ingen, selv om den stadig er udløst.

Forløbet er kort. Så længe udtrykket ikke returnerer noget, er alarmen inaktiv. Første gang det giver et resultat, bliver alarmen afventende (pending), og den forbliver afventende, så længe regelens `for` -klausul kræver det. Er betingelsen stadig opfyldt, når ventetiden er gået, bliver alarmen udløst (firing) og sendt videre. Holder betingelsen op med at gælde, er alarmen løst (resolved). Den nyere klausul `keep_firing_for` holder en udløst alarm åben et stykke tid efter, at betingelsen er forsvundet, og dokumentationen beskriver den som et værn mod blafren og falske meldinger om, at alt er godt igen. Prometheus gemmer desuden de afventende og udløste tilstande som tidsserien `ALERTS` , så man kan forespørge på selve alarmerne og tegne dem som grafer.

En regel, der venter tyve minutter, før den udløses, og ti minutter mere, før den slipper, kan se sådan ud:

```
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"
```

Labels og annotationer ligner hinanden, men har hver sin opgave. Labels er en del af det, der gør en alarm til netop den alarm, og Alertmanager læser dem for at afgøre, hvor den skal hen: `severity: page` kan ende på en telefon, mens `severity: ticket` havner i en sagskø. Annotationer bærer de ord, et menneske skal bruge, når beskeden er kommet frem: en kort opsummering, en længere beskrivelse og et link til den drejebog, der fortæller, hvad man gør først. Annotationer kan bygges med skabeloner ud fra alarmens labels og den værdi, der udløste den, og det er sådan, opsummeringen ovenfor kan nævne et tal.

En alarm og en driftshændelse er to forskellige ting. Alarmen melder, at en betingelse holdt; om det, der sker, er en driftshændelse, afgør et menneske.

Brug begrebet i en beslutning

Brug »Hvad er en alarm?« 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

Alarmen er den eneste del af overvågningen, der ikke venter på at blive set. En metrik ligger i et lager, et dashboard står på en skærm, en log venter på en søgning, men en alarm trænger ind i en persons aften. Hver alarmregel er derfor et lille, stående krav på nogens opmærksomhed, og kravet er kun rimeligt,

hvis modtageren kan gøre noget ved sagen. Prometheus' egen vejledning om alarmering formulerer det som et råd om ikke at vække folk, når der intet er at gøre.

Den anden grund er ren regning. Én årsag får som regel mange regler til at slå ud på én gang. Mister halvdelen af en tjenestes instanser forbindelsen til databasen under en netværksopdeling, kan hver instans sende sin egen alarm, og Alertmanagers dokumentation bruger netop det tilfælde til at vise, hvordan hundredvis af alarmer bliver til én notifikation. Alertmanager samler alarmer, der deler de labels, der står i `group_by`, venter `group_wait` (som standard 30 sekunder), så den første besked rummer hele bunken, sender opdateringer for gruppen højst hvert `group_interval` (5 minutter) og gentager beskeden om en alarm, der stadig er udløst, efter `repeat_interval` (4 timer). Inhibering går et skridt videre og undertrykker visse alarmer, mens en anden er udløst, så en alarm om, at en hel klynge er faldet ud, kan dæmpe alarmerne fra de enkelte tjenester i den. En silence, som Alertmanager kalder det, gør alarmer med bestemte labels lydløse i et fast tidsrum, typisk under planlagt vedligehold.

Den tredje grund er slutningen. De fleste modtagere sender en besked, når alarmen er løst, men for mailmodtageren står `send_resolved` som standard til false, så et team, der får sine alarmer på mail, kan se problemet blive meldt uden nogensinde at se det slutte.

Sådan ser det ud i praksis

Virksomhedens egen vært bruger en alarm, der reagerer på et udeblevet signal. Dens 464 MB hukommelse betjener en webserver og en forge, og mangel på hukommelse og en fuld disk er to væsentlige fejlrisici. En timer på værten pinger kun en ekstern tjeneste, så længe dens tærskelkontroller for tilgængelig hukommelse og ledig diskplads består, hvert kvarter med lidt jitter, og den eksterne tjeneste slår alarm, når de forventede ping udebliver. Fejl, der forhindrer scriptet i at køre eller nå tjenesten, herunder et tabt netværk eller en mislykket opstart, giver et udeblevet ping. Et modtaget signal bekræfter, at scriptet kørte, og at dets tærskelkontroller for tilgængelig hukommelse og ledig diskplads blev bestået; det dokumenterer ikke værtens generelle tilstand eller, at sitet svarer. Overvågning kan afsløre advarselstegn, men tidspunktet afhænger af belastningen og de forhold, der observeres.

Den samme omvending kan beskytte planlagte job. I backupimplementeringen har sikkerhedskopiering og gendannelse hvert sit successignal. Scripts og systemd-units understøtter krypterede snapshots uden for værten, integritetstjek og en månedlig prøve, hvor en kendt fil gendannes; et successignal sendes først, når det pågældende forløb er lykkedes. Det er alarmvejens design, ikke bevis for, at den var aktiv. Implementeringsregistreringen dateret 2026-10-27 siger, at timerne var deaktiveret uden et konfigureret lager, og registrerer ingen gennemført gendannelse under reelle driftsforhold. Tavshed kan udløse en ekstern alarm, når forløbet og dets overvågning er aktiveret; konfiguration alene viser ikke, at de kører.

Omvendte alarmer har deres egen fejltilstand, og den viste sig, da platformen skulle vokse fra én vært til to. Havde begge værter delt én switch, ville det have svækket netop den alarmering, den skulle udvide: en enkelt vært, der stadig pinger, holder tjekket grønt for dem begge, også mens den anden ligger død. Hver switch blev en værdi pr. vært, og den eksisterende værts switch-felter blev med vilje stående blanke, så konfigurationen siger, at forbindelsen er lagt, og at værdien mangler.

Hvor det går galt

En målt værdi havner i en label. Fordi labels identificerer en alarm, gør en label med det aktuelle tal hver ændring af tallet til en ny alarm og løser den forrige. Modtageren ser en strøm af udløst-og-løst-par for ét problem. Brug labels til identitet og routing, og læg tallene i annotationerne.

Alarmen blafrer. En betingelse, der svæver omkring tærsklen, bliver udløst, løst og udløst igen med få minutters mellemrum, og hver besked om, at den er løst, fortæller læseren noget forkert. Forlæng `for`, så et kort udsving er overstået, før nogen hører om det, og brug `keep_firing_for`, så et kort dyk under tærsklen ikke afslutter alarmen.

En silence lever længere end sin grund. Nogen gør en alarm lydløs under vedligehold med en bred betingelse og et rundhåndet sluttidspunkt, og næste uges ægte fejl kommer frem uden en lyd. Afgræns en silence så snævert, som arbejdet tillader, lad den udløbe tæt på det tidspunkt, hvor arbejdet burde være færdigt, og skriv i kommentarfeltet, hvem der oprettede den og hvorfor.

En bred rute sluger de specifikke. Alertmanager gennemgår rutetræet i rækkefølge og stopper ved det første underpunkt, der passer, medmindre ruten har `continue: true`. En opsamlingsrute, der står over et teams egen rute, tager teamets alarmer med sig. Sæt de snævreste ruter først, og prøv træet af med eksempler på labels, før I stoler på det.

Alt er akut. En disk på 70 %, et certifikat, der udløber næste måned, og en betalingstjeneste, der svarer med fejl, ringer alle til den samme telefon, og telefonen bliver hurtigt overhørt. Væk kun folk for forhold, der kræver handling nu, send resten til en sagskø eller en daglig opsummering, og slet en regel, der har slået ud i månedsvis, uden at nogen har handlet på den.

Læs mere

- Alerting overview — hvordan alarmregler i Prometheus og Alertmanager deler arbejdet, og trinene til at sætte de to op.
- Alerting rules — syntaksen for regler, herunder `for`, `keep_firing_for`, labels, annotationer og skabeloner.
- Alertmanager — gruppering, inhibering, silences og drift af flere Alertmanager-instanser for høj tilgængelighed.
- Alertmanager configuration — rutetræet, grupperingens timere, inhiberingsregler og hver modtagers indstillinger, blandt dem `send_resolved`.
- Alerting practices — projektets egne råd om, hvad der er værd at vække nogen for, og hvad der hører hjemme andre steder.

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 en alarm?« 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/alert/