Engineer

Grafana

et dashbordverktøy for metrikker og logger

Hva det er

Grafana gjør tall som ligger i andre systemer, om til dashbord. Det sender en spørring til en datakilde som Prometheus, Loki, PostgreSQL eller MySQL, tegner det som kommer tilbake, som en tidsseriegraf, en tabell, en måler eller et varmekart, og sender et driftsvarsel når en verdi krysser en linje noen har satt. Dokumentasjonen beskriver Grafana som programvare for å spørre etter, visualisere, varsle på og utforske metrikker, logger og spor uansett hvor de er lagret, og den siste delen bærer hele utformingen.

Telemetrien på skjermen blir liggende der den ble samlet inn. Hvert panel henter sin bit fra datakilden når dashbordet lastes eller oppdateres, gjennom ett av over 150 programtillegg for datakilder, og Grafana tar bare vare på sin egen konfigurasjon: dashbord, mapper, brukere, tilganger og varslingsregler. Den ligger i en liten database som er SQLite som standard, bygd inn i binærfilen, mens MySQL eller PostgreSQL anbefales i produksjon. Et dashbord er et JSON-dokument: paneler ordnet i rader, der hvert panel er en spørring pluss et valg av hvordan svaret skal tegnes, og variabler øverst som lar ett og samme dashbord bytte mellom verter, miljøer eller tjenester fra en nedtrekksliste.

Grafana Labs utvikler det, sammen med Loki for logger, Tempo for spor og Mimir for langtidslagring av Prometheus-data, et sett mange kaller Grafana-stakken. Den 20. april 2021 gikk Grafana, Loki og Tempo over fra lisensen Apache 2.0 til AGPLv3. Grafana finnes som utgaven med åpen kildekode, som Grafana Enterprise under en kommersiell lisens og som Grafana Cloud, som Grafana Labs hoster og drifter selv.

Det blir gjerne veid opp mot Kibana, som spiller den samme rollen foran Elasticsearch, mot uttrykksleseren som er bygd inn i Prometheus, og mot hostede pakker for observerbarhet som selger lagring og skjermer som ett

produkt. Verktøy for forretningsanalyse som Metabase eller Power BI kan lese de samme databasene, men de er laget for rapporter en person sitter og studerer, mens Grafana er laget for skjermer som oppdateres med noen sekunders mellomrom.

Bruk begrepet i en beslutning

Bruk «Hva er Grafana?» 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

Å installere Grafana er den enkle delen. Det kjører på en liten maskin og kan pekes mot en database teamet allerede har. Avgjørelsene som bestemmer hva det er verdt, kommer etterpå: hvem hver skjerm er til for, hvordan dashbordene holdes i orden når det finnes dusinvis av dem, og hvor det skal kjøre.

Fordi Grafana bare viser, flytter det ingen data å ta det i bruk. Et team kan la Prometheus, PostgreSQL og logglageret bli der de er, og kunne byttet ut visningslaget senere uten å migrere en eneste tidsserie, så både innlåsingen og regningen ligger hos datakildene. Pakker som selger lagring og visning sammen, gjør den motsatte byttehandelen: én regning og én innlogging, og data som er vanskeligere å ta med seg et annet sted.

Publikummet former et dashbord mer enn verktøyet gjør. En ingeniør med vakt vil ha en side som svarer på «er noe ødelagt, og hvor» med ett blikk, med lenker videre til detaljene. Et rom fullt av folk med ulike jobber trenger færre og større paneler, enkle titler og en terskellinje som sier når det er grunn til uro. Den samme Grafana-instansen betjener begge gjennom ulike dashbord, og en side som er laget for begge på én gang, ender ofte med å tjene ingen av dem.

Hvordan dashbordene holdes ved like, er den andre avgjørelsen. Grafana kan laste dem fra JSON-filer på disk og følge med på mappen etter endringer, noe dokumentasjonen kaller provisioning. I versjonskontroll får et dashbord gjennomgang, historikk og en vei tilbake, og et nytt miljø får de samme

skjermene uten at noen må klikke seg gjennom dem. Prisen er at redigering i nettleseren slutter å være måten endringer gjøres på, og det må hele teamet godta.

Lisens og drift kommer sist. Under AGPLv3 må den som endrer Grafana og lar andre bruke den endrede versjonen over et nettverk, tilby dem kildekoden. Et team som kjører en uendret Grafana for sine egne ansatte, står et helt annet sted enn en leverandør som leverer en patchet utgave inne i et produkt, og leverandøren bør lese lisensen sammen med en jurist. Å drifte selv er én binærfil helt til det trengs en andre instans. Fra da av deler instansene en MySQL- eller PostgreSQL-database, og Grafana Cloud er måten å overlate det arbeidet til Grafana Labs på.

Slik ser det ut i praksis

Begge gangene Grafana dukker opp i selskapets arbeid, er det den delen folk så på, med Prometheus bak.

I en rolle som infrastrukturingeniør fra 2022 til 2024 var utgangspunktet et team som fikk høre om problemer etter brukerne, fordi det manglet én samlet oversikt over hvordan systemene hadde det. Et overvåkingsdashbord med Grafana og Prometheus ga teamet den oversikten. De sentrale tjenestene ble instrumentert, metrikkene ble valgt fordi de betydde noe og ikke fordi de så travle ut, og varslingsterskler ble satt på dem og gjort synlige der teamet ville se dem mens det fortsatt var tid til å handle. Problemer begynte å bli fanget opp før de eskalerte, påliteligheten ble bedre, og teamet gikk fra brannslukking til å ligge i forkant.

Det andre tilfellet var bygd for et blandet publikum. Et dashbord på en stor TV-skjerm på kontoret var ment for både tekniske og ikke-tekniske kolleger. Den andre versjonen, på Grafana og Prometheus, viste metrikker som var plukket ut sammen med DevOps- og utviklingsteamene, matet av datapipelines som hentet inn ytelsestall fra flere systemer, med varsler når en terskel ble overskredet. Teamene oppdaget og løste driftshendelser 40 % raskere, og med tallene foran hele kontoret ble systemenes helse noe folk eide sammen.

En veggskjerm av det slaget leses fra andre siden av rommet av folk som aldri kommer til å redigere en spørring. Den belønner noen få store paneler, enheter i titlene og farger som bare skifter når noen bør handle, og Grafana overlater hvert eneste av de valgene til den som bygger dashbordet.

Hvor det går galt

Et dashbord vokser til ingen leser det. Hver driftshendelse legger til et panel, og ingen tar et bort, så et år senere rommer siden førti små grafer, og den ene som betyr noe, ligger nedenfor den synlige delen av skjermen. Gi hvert dashbord ett publikum og ett spørsmål, legg svaret på «er alt friskt» i øverste rad, og flytt detaljene til dashbord som lenkes derfra. Et panel som ingen på teamet kan forklare, er et panel som skal slettes.

En retting gjort i nettleseren forsvinner. Dashbordet ble provisjonert fra en fil. Med `allowUiUpdates` på standardverdien false nekter Grafana å lagre og viser dialogen «Cannot save provisioned dashboard». Settes den til true, lagrer Grafana endringen i databasen sin og overskriver den så med filen neste gang filen endres. Rediger filen, eller eksporter JSON-en fra lagringsdialogen og commit den, og skriv innstillingen eksplisitt i provider-oppføringen:

```
apiVersion: 1
providers:
    - name: operations
      type: file
      allowUiUpdates: false
      options:
          path: /var/lib/grafana/dashboards
```

Veggskjermen bremser systemet den overvåker. Hver åpen nettleser, TV-en medregnet, kjører spørringen til hvert panel på nytt ved hver oppdatering, og fem sekunder er det korteste oppdateringsintervallet Grafana tillater i standardoppsettet. Tjue paneler som oppdateres hvert femte sekund, gir fire spørringer i sekundet fra én enkelt skjerm. Oppdater i det tempoet dataene endrer seg i, som for Prometheus betyr skrapeintervallet eller sjeldnere, hev gulvet for alle, og regn ut tunge uttrykk på forhånd med recording rules i datakilden:

```
[dashboards]
min_refresh_interval = 30s
```

Den samme driftshendelsen varsler folk to ganger. Regelen ligger i Prometheus, rutet gjennom Alertmanager, og en gang til i Grafana Alerting. Eller så evaluerer to Grafana-instanser hver eneste regel hver for seg, fordi varslingen

ble glemt da den andre instansen kom til. Gi hver regel ett eneste hjem, og følg Grafanas fremgangsmåte for høy tilgjengelighet i varslingen før en andre instans går i drift.

Dashbord og brukere endrer seg etter hvilken server som svarer. To Grafana-instanser bak en lastbalanserer hadde hver sin SQLite-fil, så et dashbord som lagres på den ene, mangler på den andre, og en bruker finnes bare på én av dem. Grafanas egen veiledning gjør en delt MySQL- eller PostgreSQL-database til et krav for høy tilgjengelighet.

Les mer

- About Grafana — prosjektets egen oversikt over hva Grafana gjør, utgavene og prosjektene med åpen kildekode rundt det.
- Data sources — systemene Grafana kan spørre, og hvordan hvert av dem legges til og konfigureres.
- Dashboards — paneler, rader, variabler og JSON-modellen et dashbord lagres som.
- Provision Grafana — datakilder og dashbord fra filer, og hva som skjer med endringer gjort i nettleseren.
- Set up Grafana for high availability — den delte databasen en andre instans trenger, og de ekstra stegene for varsling.
- Licensing — siden der Grafana Labs forklarer overgangen fra Apache 2.0 til AGPLv3 i 2021.

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 Grafana?» 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/grafana/