Engineer

Grafana

et dashboardværktøj til metrikker og logge

Hvad det er

Grafana gør tal, der ligger i andre systemer, til dashboards. Det sender en databaseforespørgsel til en datakilde som Prometheus, Loki, PostgreSQL eller MySQL, tegner svaret som en tidsseriegraf, en tabel, en måler eller et heatmap og sender en alarm, når en værdi krydser en grænse, som nogen har sat. Grafanas egen dokumentation beskriver det som software til at forespørge, visualisere, alarmere på og udforske metrikker, logge og traces, uanset hvor de er gemt, og netop den sidste del bærer hele designet.

Det, skærmen viser, bliver liggende der, hvor det blev indsamlet. Hvert panel henter sit udsnit fra datakilden, når dashboardet indlæses eller opdateres, gennem et af mere end 150 datakilde-plugins, og Grafana gemmer kun sin egen konfiguration: dashboards, mapper, brugere, rettigheder og alarmregler. De ligger i en lille database, som er SQLite som standard og bygget ind i selve programmet, mens MySQL eller PostgreSQL anbefales i produktion. Et dashboard er et JSON-dokument: paneler ordnet i rækker, hvor hvert panel er en forespørgsel plus et valg af visning, og variabler øverst, der lader ét dashboard skifte mellem værter, miljøer eller tjenester fra en rullemenu.

Grafana udvikles af Grafana Labs sammen med Loki til logge, Tempo til traces og Mimir til langtidslagring af Prometheus-data, en samling mange kalder Grafana-stakken. Den 20. april 2021 skiftede Grafana, Loki og Tempo fra Apache 2.0-licensen til AGPLv3. Grafana findes som open source-udgave, som Grafana Enterprise under en kommerciel licens og som Grafana Cloud, som Grafana Labs selv hoster og driver.

Det sammenlignes typisk med Kibana, der spiller samme rolle foran Elasticsearch, med den udtryksbrowser, der er bygget ind i Prometheus, og med hostede observability-suiter, der sælger lagring og skærme som ét produkt. BI-værktøjer

som Metabase eller Power BI kan læse de samme databaser, men de er lavet til rapporter, man sætter sig ned og studerer, mens Grafana er lavet til skærme, der opdateres med få sekunders mellemrum.

Brug begrebet i en beslutning

Brug »Hvad er Grafana?« 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

At installere Grafana er den nemme del. Det kører på en lille maskine og kan peges mod en database, teamet allerede har. De beslutninger, der afgør, hvad det er værd, kommer bagefter: hvem hver skærm er til, hvordan man holder styr på dashboards, når der er dusinvis af dem, og hvor det hele skal køre.

Fordi Grafana kun viser, flytter det ingen data at tage det i brug. Et team kan beholde Prometheus, PostgreSQL og sit loglager, hvor de er, og kunne skifte visningslaget ud senere uden at migrere en eneste serie, så både bindingen og regningen ligger hos datakilderne. Suiter, der sælger lagring og visning samlet, gør den modsatte handel: én regning og ét login, og data, der er sværere at tage med sig.

Publikum former et dashboard mere, end værktøjet gør. En udvikler med vagten vil have en side, der med ét blik svarer på »er noget i stykker, og hvor«, med links videre til detaljerne. Et lokale fuldt af folk med forskellige opgaver har brug for færre og større paneler, klare titler og en tærskellinje, der siger, hvornår man skal blive bekymret. Den samme Grafana kan tjene begge gennem hver sit dashboard, og en side, der skal tjene begge på én gang, ender ofte med at tjene ingen af dem.

Hvordan dashboards opbevares, er den anden beslutning. Grafana kan indlæse dem fra JSON-filer på disken og holde øje med mappen, og dokumentationen kalder det provisioning. I versionsstyring får et dashboard review, historik og en

vej tilbage, og et nyt miljø får de samme skærme, uden at nogen skal klikke dem sammen. Prisen er, at redigering i browseren holder op med at være måden at ændre ting på, og det skal hele teamet acceptere.

Licens og drift kommer til sidst. Under AGPLv3 skal den, der ændrer Grafana og lader andre bruge den ændrede version over et netværk, tilbyde dem kildekoden. Et team, der kører en uændret Grafana for sine egne medarbejdere, står et helt andet sted end en leverandør, der bygger en tilpasset udgave ind i et produkt, og leverandøren bør læse licensen sammen med en jurist. Selvhosting er ét program, indtil der skal en instans nummer to til. Fra da af deler instanserne en MySQL- eller PostgreSQL-database, og Grafana Cloud er måden at overlade det arbejde til Grafana Labs på.

Sådan ser det ud i praksis

Begge gange Grafana optræder i virksomhedens arbejde, er det den del, folk så på, med Prometheus bag sig.

I en rolle som infrastrukturingeniør fra 2022 til 2024 var udgangspunktet et team, der hørte om problemer, efter brugerne havde mærket dem, fordi der ikke fandtes ét samlet overblik over systemerne. Et overvågningsdashboard på Grafana og Prometheus gav det overblik. De vigtigste tjenester blev instrumenteret, metrikkerne blev valgt, fordi de betød noget og ikke bare så travle ud, og alarmtærskler blev sat på dem og vist der, hvor teamet faktisk ville se dem, mens der stadig var tid til at handle. Problemer blev fanget, før de voksede sig store, pålideligheden blev bedre, og teamet gik fra brandslukning til at komme problemerne i forkøbet.

Den anden sag var bygget til et blandet publikum. Et dashboard på en stor tv-skærm på kontoret var tænkt til både tekniske og ikke-tekniske kolleger. Anden version, bygget på Grafana og Prometheus, viste metrikker udvalgt sammen med DevOps- og udviklingsteamene, fodret af datapipelines, der hentede ydelsestal ind fra flere systemer, og med alarmer, når en tærskel blev overskredet. Teamene opdagede og løste hændelser 40 % hurtigere, og med tallene foran hele kontoret blev systemernes sundhed noget, man ejede i fællesskab.

En vægskærm som den bliver læst på tværs af et lokale af folk, der aldrig kommer til at redigere en forespørgsel. Den belønner få store paneler, enheder i titlerne og farver, der kun skifter, når nogen bør handle, og Grafana overlader hvert eneste af de valg til den, der bygger dashboardet.

Hvor det går galt

Et dashboard vokser, til ingen læser det. Hver hændelse tilføjer et panel, og ingen fjerner nogensinde et, så et år senere rummer siden fyrre små grafer, og den, der betyder noget, ligger langt nede, hvor ingen scroller hen. Giv hvert dashboard ét publikum og ét spørgsmål, læg svaret på »er det sundt« i øverste række, og flyt detaljerne til dashboards, der linkes til derfra. Et panel, som ingen på teamet kan forklare, er et panel, der skal slettes.

En rettelse lavet i browseren forsvinder. Dashboardet blev provisioneret fra en fil. Med `allowUiUpdates` på standardværdien false afviser Grafana at gemme og viser dialogen »Cannot save provisioned dashboard«. Står den til true, gemmes ændringen i databasen og overskrives så af filen, næste gang filen ændres. Ret filen, eller eksportér JSON fra gem-dialogen og commit den, og skriv indstillingen eksplicit hos provideren:

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

Vægskærmen gør det system, den viser, langsommere. Hver åben browser, tv'et medregnet, kører hvert panels forespørgsel igen ved hver opdatering, og fem sekunder er den korteste opdatering, Grafana tillader som standard. Tyve paneler med fem sekunders opdatering giver fire forespørgsler i sekundet fra én skærm. Opdatér i det tempo, data ændrer sig i, hvilket for Prometheus betyder scrape-intervallet eller langsommere, hæv bunden for alle, og beregn tunge udtryk på forhånd med recording rules i datakilden:

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

Den samme hændelse kalder folk op to gange. Reglen findes i Prometheus, sendt videre gennem Alertmanager, og igen i Grafana Alerting. Eller også evaluerer to Grafana-instanser hver især alle regler, fordi alarmeringen blev

glemt, da instans nummer to kom til. Giv hver regel ét hjem, og følg Grafanas trin for høj tilgængelighed i alarmeringen, før en ekstra instans går i drift.

Dashboards og brugere skifter alt efter, hvilken server der svarer. To Grafana-instanser bag en load balancer har hver sin SQLite-fil, så et dashboard gemt på den ene mangler på den anden, og en bruger findes kun det ene sted. Grafanas egen vejledning gør en delt MySQL- eller PostgreSQL-database til et krav for høj tilgængelighed.

Læs mere

- About Grafana — projektets eget overblik over, hvad Grafana gør, dets udgaver og open source-projekterne omkring det.
- Data sources — de systemer, Grafana kan forespørge, og hvordan hvert enkelt tilføjes og sættes op.
- Dashboards — paneler, rækker, variabler og den JSON-model, et dashboard gemmes som.
- Provision Grafana — datakilder og dashboards fra filer, og hvad der sker med ændringer lavet i browseren.
- Set up Grafana for high availability — den delte database, en instans nummer to kræver, og de ekstra trin for alarmering.
- Licensing — Grafana Labs' side om skiftet fra Apache 2.0 til AGPLv3 i 2021.

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 Grafana?« 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/grafana/