Engineer

Grafana

Dashboardwerkzeug für Metriken und Protokolle

Was es ist

Grafana macht aus Zahlen in anderen Systemen Dashboards. Es sendet eine Abfrage an eine Datenquelle wie Prometheus, Loki, PostgreSQL oder MySQL, zeichnet die Antwort als Zeitreihendiagramm, Tabelle, Messanzeige oder Heatmap und löst einen Alarm aus, wenn ein Wert eine festgelegte Grenze überschreitet. Die eigene Dokumentation beschreibt Software zum Abfragen, Visualisieren, Alarmieren und Erkunden von Metriken, Protokollen und Traces „unabhängig von ihrem Speicherort“. In diesen letzten vier Wörtern steckt der Kern des Entwurfs.

Die Telemetriedaten auf dem Bildschirm bleiben dort, wo sie gesammelt wurden. Jedes Panel holt beim Laden oder Aktualisieren des Dashboards seinen Ausschnitt aus der Datenquelle, über eines von mehr als 150 Datenquellen-Plugins. Grafana speichert nur seine eigene Konfiguration: Dashboards, Ordner, Nutzer, Berechtigungen und Alarmregeln. Diese liegen in einer kleinen Datenbank, standardmäßig SQLite, dessen Engine in die Binärdatei eingebettet ist. Für den Produktivbetrieb werden MySQL oder PostgreSQL empfohlen. Ein Dashboard ist ein JSON-Dokument: in Zeilen angeordnete Panels, jeweils eine Abfrage samt der gewählten Darstellungsform, sowie Variablen am oberen Rand, mit denen dasselbe Dashboard per Auswahlmenü zwischen Hosts, Umgebungen oder Diensten wechselt.

Grafana Labs entwickelt es neben Loki für Protokolle, Tempo für Traces und Mimir für die langfristige Prometheus-Speicherung. Diese Zusammenstellung heißt oft Grafana-Stack. Am 20. April 2021 wechselten Grafana, Loki und Tempo von Apache 2.0 zu AGPLv3. Grafana gibt es als Open-Source-Ausgabe, als Grafana Enterprise mit kommerzieller Lizenz und als Grafana Cloud, die Grafana Labs hostet und betreibt.

Verglichen wird es meist mit Kibana, das diese Rolle vor Elasticsearch übernimmt, mit dem in Prometheus eingebauten Ausdrucksbrowser und mit gehosteten Observability-Suiten, die Speicher und Anzeige als ein Produkt verkaufen. Business-Intelligence-Werkzeuge wie Metabase oder Power BI können dieselben Datenbanken lesen. Sie sind jedoch für Berichte gebaut, die ein Mensch studiert; Grafana für Anzeigen, die sich alle paar Sekunden erneuern.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist Grafana?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Monitoring und Alarme melden sich, solange ein Fehler noch klein ist; Kennzahlen zeigen, wie oft das gelingt.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. Das erste Observability-Dashboard aufgebaut — der erste Einblick des Unternehmens in die eigenen Systeme

Das erste Observability-Dashboard aufgebaut Kontakt aufnehmen →

Warum es wichtig ist

Die Installation ist der leichte Teil. Grafana läuft auf einem kleinen Rechner und lässt sich mit einer bereits vorhandenen Datenbank verbinden. Die Entscheidungen über seinen Wert folgen danach: Für wen ist welcher Bildschirm gedacht, wie werden Dutzende Dashboards gepflegt und wo wird Grafana betrieben?

Weil Grafana nur darstellt, verschiebt seine Einführung keine Daten. Ein Team kann Prometheus, PostgreSQL und seinen Protokollspeicher am bisherigen Ort lassen und die Anzeige später austauschen, ohne eine einzige Zeitreihe zu migrieren. Bindung und Rechnung liegen deshalb bei den Datenquellen. Suiten, die Speicher und Anzeige gemeinsam verkaufen, gehen den umgekehrten Kompromiss ein: eine Rechnung und eine Anmeldung, dafür Daten, die sich schwerer anderswohin mitnehmen lassen.

Das Publikum prägt ein Dashboard stärker als das Werkzeug. Eine technische Fachkraft im Bereitschaftsdienst braucht eine Seite, die auf einen Blick „Ist etwas kaputt, und wo?“ beantwortet und zur Vertiefung verlinkt. Ein Raum mit Menschen verschiedener Aufgaben braucht weniger und größere Panels, klare Titel und eine Grenzlinie, die zeigt, wann Anlass zur Sorge besteht. Dasselbe Grafana bedient beide Gruppen über verschiedene Dashboards. Eine für beide zugleich gebaute Seite dient häufig keiner richtig.

Die Pflege ist die zweite Entscheidung. Grafana kann Dashboards aus JSON-Dateien auf der Festplatte laden und den Ordner auf Änderungen überwachen. Die Dokumentation nennt das Provisioning. In der Versionsverwaltung erhält ein Dashboard Review, Historie und einen Rückweg. Eine neue Umgebung bekommt dieselben Anzeigen, ohne dass jemand sie zusammenklickt. Der Preis: Änderungen entstehen nicht mehr durch Bearbeiten im Browser, und das ganze Team muss das mittragen.

Lizenz und Hosting kommen zuletzt. Wer Grafana verändert und anderen die geänderte Version über ein Netzwerk zur Nutzung anbietet, muss ihnen unter AGPLv3 den Quellcode dieser Version anbieten. Ein Team mit unverändertem Grafana für die eigenen Beschäftigten steht daher an einem ganz anderen Punkt als ein Anbieter, der eine gepatchte Version in einem Produkt ausliefert. Letzterer sollte die Lizenz mit juristischer Beratung lesen. Beim Selbsthosting genügt eine Binärdatei, bis eine zweite Instanz nötig wird. Ab dann teilen sich die Instanzen eine MySQL- oder PostgreSQL-Datenbank. Mit Grafana Cloud lässt sich diese Arbeit stattdessen Grafana Labs überlassen.

Wie es in der Praxis aussieht

In beiden Fällen aus der Arbeit dieses Unternehmens war Grafana der sichtbare Teil, mit Prometheus dahinter.

Ausgangspunkt der Arbeit in einer Rolle als Infrastructure Engineer von 2022 bis 2024 war ein Team, das erst nach seinen Nutzern von Problemen erfuhr, weil eine gemeinsame Sicht auf den Zustand seiner Systeme fehlte. Ein Monitoring-Dashboard mit Grafana und Prometheus schuf diese Sicht. Die wichtigen Dienste wurden instrumentiert. Metriken wurden ausgewählt, weil sie etwas aussagten, nicht weil sie nach viel Aktivität aussahen. Darauf wurden Alarmschwellen gesetzt und dort sichtbar gemacht, wo das Team sie noch rechtzeitig zum Handeln sah. Probleme wurden erkannt, bevor sie eskalierten, die Zuverlässigkeit verbesserte sich. Das Team ging von der Brandbekämpfung dazu über, Schwierigkeiten zuvorzukommen.

Der zweite Fall entstand für ein gemischtes Publikum. Ein Dashboard auf einem großen Bürofernseher sollte technische und nichttechnische Kollegen gleichermaßen erreichen. Seine zweite Version mit Grafana und Prometheus zeigte gemeinsam mit den DevOps- und Engineering-Teams ausgewählte Metriken. Datenpipelines lieferten Leistungswerte aus mehreren Systemen; beim

Überschreiten eines Grenzwerts folgten Alarme. Die Teams erkannten und behoben Vorfälle 40 % schneller. Weil das ganze Büro die Zahlen sah, übernahmen die Menschen gemeinsam Verantwortung für den Systemzustand.

Menschen, die nie eine Abfrage bearbeiten werden, lesen einen solchen Wandbildschirm von der anderen Seite des Raums aus. Er profitiert von wenigen großen Panels, Einheiten in den Titeln und Farben, die nur bei Handlungsbedarf wechseln. Grafana überlässt jede dieser Entscheidungen demjenigen, der das Dashboard baut.

Wo es schiefgeht

Ein Dashboard wächst, bis es niemand mehr liest. Jeder Vorfall fügt ein Panel hinzu, keines verschwindet. Ein Jahr später enthält die Seite vierzig kleine Diagramme, und das entscheidende steht unterhalb des sichtbaren Bereichs. Geben Sie jedem Dashboard eine Zielgruppe und eine Frage. Platzieren Sie die Antwort auf „Ist das System in Ordnung?“ in der obersten Reihe und verschieben Sie Details auf verlinkte Dashboards. Ein Panel, das niemand im Team erklären kann, sollte gelöscht werden.

Eine Korrektur im Browser verschwindet. Das Dashboard wurde aus einer Datei provisioniert. Mit dem Standardwert false für `allowUiUpdates` verweigert Grafana das Speichern und zeigt „Cannot save provisioned dashboard“. Bei true speichert es die Änderung in seiner Datenbank und überschreibt sie bei der nächsten Dateiänderung mit dem Dateiinhalt. Bearbeiten Sie die Datei oder exportieren Sie das JSON aus dem Speicherdialog und committen Sie es. Halten Sie die Providereinstellung ausdrücklich fest:

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

Der Wandbildschirm bremst das überwachte System. Jeder geöffnete Browser, auch der auf dem Fernseher, führt bei jeder Aktualisierung jede Panelabfrage erneut aus. Fünf Sekunden sind das kürzeste standardmäßig zulässige Intervall. Zwanzig Panels mit fünf Sekunden Intervall erzeugen vier Abfragen pro Sekunde

von einem Bildschirm. Aktualisieren Sie im Takt der Datenänderungen. Für Prometheus heißt das: im Scrape-Intervall oder langsamer. Heben Sie die Untergrenze für alle an und berechnen Sie aufwendige Ausdrücke mit Recording Rules in der Datenquelle vor:

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

Derselbe Vorfall alarmiert Menschen doppelt. Die Regel liegt in Prometheus mit Weiterleitung über Alertmanager und nochmals in Grafana Alerting. Oder zwei Grafana-Instanzen werten jede Regel aus, weil die Alarmierung beim Hinzufügen der zweiten vergessen wurde. Geben Sie jeder Regel einen einzigen Ort. Folgen Sie Grafanas Schritten für hochverfügbare Alarmierung, bevor eine zweite Instanz produktiv geht.

Dashboards und Nutzer hängen vom antwortenden Server ab. Zwei Grafana-Instanzen hinter einem Load Balancer hielten jeweils ihre eigene SQLite-Datei. Ein auf einer Instanz gespeichertes Dashboard fehlt daher auf der anderen, ein Nutzer existiert nur auf einer. Grafanas eigene Anleitung verlangt für Hochverfügbarkeit eine gemeinsame MySQL- oder PostgreSQL-Datenbank.

Weiterführende Lektüre

- Über Grafana — Der Projektüberblick über Aufgaben, Ausgaben und benachbarte Open-Source-Projekte.
- Datenquellen — Die von Grafana abfragbaren Systeme und ihre Einrichtung und Konfiguration.
- Dashboards — Panels, Zeilen, Variablen und das JSON-Modell der gespeicherten Dashboards.
- Grafana provisionieren — Datenquellen und Dashboards aus Dateien und die Folgen von Browseränderungen.
- Grafana hochverfügbar einrichten — Die gemeinsame Datenbank einer zweiten Instanz und die zusätzlichen Schritte für die Alarmierung.
- Lizenzierung — Die Seite von Grafana Labs zum Wechsel von Apache 2.0 auf AGPLv3 im Jahr 2021.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist Grafana?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/grafana/