Engineer

Grafana

メトリクスやログを表示するダッシュボードツール

どのようなものか

Grafanaは、ほかのシステムに保存した数値をダッシュボードにします。Prometheus、Loki、PostgreSQL、MySQLなどのデータソースへクエリを送り、返ったものを時系列グラフ、表、ゲージ、ヒートマップで描き、設定した境界を値が超えるとアラートを出します。公式文書は「保存先がどこであっても」、メトリクス、ログ、トレースのクエリ、可視化、アラート、探索を行うソフトウェアと説明します。原文で最後の四語を占める、この保存先を問わない表現に、設計の中心があります。

画面のテレメトリーデータは、収集した場所に置いたままです。ダッシュボードの読み込みや更新のたびに、各パネルが150を超えるデータソースプラグインのいずれかを使い、必要な部分を取得します。Grafana自体が保持するのは、ダッシュボード、フォルダー、利用者、権限、アラート規則という、自身の設定だけです。小さいデータベースに保存し、既定はバイナリーに組み込まれたSQLiteで、本番にはMySQLかPostgreSQLを推奨します。ダッシュボードはJSON文書です。行に並ぶ各パネルは、クエリと描画方法の選択からなり、上部の変数によって、ドロップダウンからホスト、環境、サービスを切り替えて、一つの画面を使い回せます。

開発元のGrafana Labsは、ログ用のLoki、トレース用のTempo、Prometheusの長期保存用のMimirも開発しています。まとめてGrafanaスタックとよく呼びます。2021年4月20日、Grafana、Loki、TempoはApache 2.0ライセンスからAGPLv3へ移りました。Grafanaには、オープンソース版、商用ライセンスのGrafana Enterprise、Grafana Labsがホストして運用するGrafana Cloudがあります。

比較されるのは普通、Elasticsearchの前で同じ役割を担うKibana、Prometheusに組み込まれた式ブラウザー、そして保存と画面を一製品で売る、運用込みの可観測性スイートです。MetabaseやPower BIなどのBIツールも同じデータベース

を読めますが、人がじっくり検討するレポート向けです。Grafanaは数秒ごとに更新する画面向けに作っています。

概念を判断に生かす

「Grafanaとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 障害が小さいうちに知らせる監視とアラートを整え、安定した状態をどれだけ維持できているかを数値で示します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 社内初のオブザーバビリティダッシュボードを構築 — 社内システムの状態を初めて可視化

社内初のオブザーバビリティダッシュボードを構築 お問い合わせ →

なぜ重要なのか

導入そのものは簡単です。小さいマシンで動き、チームがすでに持つデータベースへ接続できます。その価値を左右する決定は、後にあります。画面の対象者、数十個になったダッシュボードの管理方法、そして動かす場所です。

表示するだけなので、採用してもデータを移しません。Prometheus、PostgreSQL、ログ保存先はそのままにでき、後で表示層を交換しても、一つの系列さえ移行する必要はありません。そのため、乗り換えを難しくする依存関係も請求も、データソース側にあります。保存と表示をまとめて売るスイートは逆で、請求もログインも一つになる代わりに、データをほかへ持ち出しにくくなります。

画面を形作るのは、道具よりも対象者です。当番のエンジニアは、「異常があるか、あるならどこか」が一目で分かり、詳細へ進めるページを求めます。違う仕事の人がいる部屋では、少数の大きいパネル、分かりやすい題名、気にすべき時点を示す閾値の線が必要です。同じGrafanaでも別々のダッシュボードで両方に応えられますが、一画面で兼ねようとすると、どちらにも役立たなくなりがちです。

二つ目の決定は、ダッシュボードの管理方法です。Grafanaはディスク上のJSONファイルから読み込み、フォルダーの変更を監視できます。公式文書はこれをプロビジョニングと呼びます。版管理に入れれば、レビュー、履歴、元へ戻す方法が得られ、新しい環境でも、誰かが画面をクリックして設定せずに、同じものを使えます。代償として、変更の方法はブラウザーでの編集ではなくなり、チーム全体で受け入れる必要があります。

最後は、ライセンスと実行場所です。AGPLv3では、Grafanaを変更し、その変更版をほかの人にネットワーク越しに使わせるなら、その利用者がソースを入手できるようにする必要があります。未変更のGrafanaを自社の従業員向けに動かすチームと、修正したものを製品へ組み込んで出荷するベンダーでは、立場が大きく違い、後者は法律家とライセンスを確認すべきです。自社運用は、二つ目のインスタンスが必要になるまでは単一バイナリーで済みます。その後は、各インスタンスでMySQLかPostgreSQLを共有します。この仕事をGrafana Labsへ任せる選択がGrafana Cloudです。

実際の活用例

自社の仕事にGrafanaが登場する二つの事例は、どちらも、Prometheusを背後に置き、人が見る部分をGrafanaが担います。

2022年から2024年のインフラエンジニアとしての仕事では、システムの状態をまとめて見る場所がなく、利用者より後に問題を知るチームが出発点でした。GrafanaとPrometheusの監視ダッシュボードで、その状況を見られるようにしました。主要サービスに計測を組み込み、動きが多く見えるかどうかではなく、意味があるかどうかでメトリクスを選びました。閾値を設定し、対処の時間がまだあるうちに、チームの目に入る場所でアラートを出しました。問題が大きくなる前に見つかるようになり、信頼性が向上し、後追いの火消しから先回りの対処へ移りました。

もう一つは、さまざまな職種の人向けでした。オフィスの大型テレビに表示するダッシュボードは、技術職にも、それ以外の同僚にも向けたものです。GrafanaとPrometheusによる第二版では、DevOpsとエンジニアリングのチームと選んだメトリクスを表示しました。複数システムの性能値を取り込むデータパイプラインからデータを供給し、閾値を超えるとアラートを出します。チームはインシデントを40%速く検出、解決できるようになり、オフィス全体で数字を目にすることで、システムの健全性を皆で担うようになりました。

このような壁面画面は、クエリを編集しない人たちが、部屋の反対側から読みます。少数の大きいパネル、題名の中の単位、誰かが動くべきときだけ変わる色が有効です。どの選択も、Grafanaはダッシュボードの作り手に委ねます。

問題が起きるところ

ダッシュボードに情報が増えすぎて、誰も読みません。 インシデントごとにパネルが増え、減ることはなく、一年後には小さいグラフが四十個になり、重要な一つはスクロールしないと見えません。各画面の対象者と答える問いを一つに決

め、「正常か」への答えを最上段に置き、詳細はリンク先の別のダッシュボードへ移します。チームの誰も説明できないパネルは、消すべきパネルです。

ブラウザーで直した内容が消えます。 ダッシュボードをファイルからプロビジョニングしたためです。`allowUiUpdates` が既定のfalseなら、「Cannot save provisioned dashboard」というダイアログで保存を拒否します。trueにするとデータベースへ保存しますが、次にファイルが変わると、その内容で上書きします。ファイルを編集するか、保存ダイアログからJSONをエクスポートしてコミットし、プロバイダーの設定を明示します。

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

壁面画面が、監視先のシステムを遅くします。 テレビを含む、開いているすべてのブラウザーが、更新のたびに全パネルのクエリを再実行します。初期状態で許す最短の更新間隔は五秒です。二十パネルを五秒ごとに更新すると、一画面だけで一秒に四クエリになります。データが変わる速さで更新します。Prometheusなら収集間隔と同じか、それより遅くします。全員に適用する最小間隔を引き上げ、重い式はデータソースの記録規則で事前に計算します。

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

同じインシデントで、担当者が二度呼び出されます。 規則をPrometheusに置いてアラートをAlertmanager経由で送る一方、Grafana Alertingにも同じ規則があります。または、二つ目のインスタンスを追加するときにアラートの構成を忘れ、両方のGrafanaがすべての規則を評価します。各規則の置き場を一つにし、二つ目のインスタンスを稼働させる前に、Grafanaのアラートの高可用性手順に従います。

応答するサーバーによって、ダッシュボードや利用者が変わります。 ロードバランサーの後ろの二つのGrafanaが、別々のSQLiteファイルを持っているためです。一方で保存した画面が他方にはなく、利用者も片方にしかいません。公式ガ

イドは、高可用性には共有のMySQLかPostgreSQLデータベースが必要だとしています。

参考資料

- Grafanaについて — 機能、エディション、周辺のオープンソースプロジェクトについての、開発元の概要です。
- データソース — クエリを送れるシステムと、それぞれの追加、設定方法です。
- ダッシュボード — パネル、行、変数、保存に使うJSONモデルです。
- Grafanaのプロビジョニング — ファイルからのデータソースやダッシュボードの設定と、ブラウザーでの編集に何が起きるかを説明します。
- Grafanaの高可用性構成 — 二つ目のインスタンスに必要な共有データベースと、アラート用の追加手順です。
- ライセンス — 2021年のApache 2.0からAGPLv3への移行についての、Grafana Labsのページです。

用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Grafanaとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/grafana/