Engineer

Grafana

outil de tableaux de bord pour les métriques et journaux

Ce que c’est

Grafana transforme les nombres conservés dans d’autres systèmes en tableaux de bord. Il interroge une source comme Prometheus, Loki, PostgreSQL ou MySQL, dessine la réponse sous forme de graphique de série temporelle, de tableau, de jauge ou de carte de chaleur et déclenche une alerte lorsqu’une valeur franchit une limite définie. Sa documentation le décrit comme un logiciel permettant d’interroger, de visualiser et d’explorer les métriques, journaux et traces, ainsi que de créer des alertes à leur sujet, « où qu’ils soient stockés » ; ces quatre derniers mots expriment le principe de conception.

La télémétrie visible reste là où elle a été recueillie. Chaque panneau récupère sa part auprès de la source au chargement ou au rafraîchissement du tableau, par l’un de plus de 150 connecteurs de sources de données. Grafana ne garde que sa configuration : tableaux de bord, dossiers, utilisateurs, droits et règles d’alerte. Ils résident dans une petite base de données SQLite, sauf si un autre moteur est configuré. Le moteur SQLite est intégré au binaire ; MySQL ou PostgreSQL sont recommandés en production. Un tableau de bord est un document JSON : des panneaux rangés en lignes, chacun avec une requête et un mode de représentation, et des variables en haut pour passer d’un hôte, environnement ou service à l’autre dans une liste déroulante.

Grafana Labs le développe aux côtés de Loki pour les journaux, Tempo pour les traces et Mimir pour le stockage Prometheus à long terme : un ensemble souvent appelé stack Grafana. Le 20 avril 2021, Grafana, Loki et Tempo sont passés de la licence Apache 2.0 à AGPLv3. Grafana existe en édition open source, en Grafana Enterprise sous licence commerciale et en Grafana Cloud, hébergé et exploité par Grafana Labs.

On le compare généralement à Kibana, qui joue le même rôle devant Elasticsearch, au navigateur d'expressions de Prometheus et aux suites d'observabilité hébergées qui vendent stockage et écrans dans un produit unique. Metabase et Power BI, outils d'informatique décisionnelle, peuvent lire les mêmes bases, mais visent des rapports qu'une personne étudie ; Grafana vise des écrans actualisés toutes les quelques secondes.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que Grafana ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Une supervision et des alertes qui signalent les incidents tant qu'ils restent mineurs, avec les chiffres qui montrent à quelle fréquence ils le restent.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Premier tableau de bord d'observabilité — première vue de l'entreprise sur ses propres systèmes

Premier tableau de bord d'observabilité Contactez-nous →

Pourquoi c'est important

Installer Grafana est la partie facile. Il fonctionne sur une petite machine et peut pointer sur une base que l'équipe possède déjà. Les décisions déterminant sa valeur viennent après : à qui s'adresse chaque écran, comment gérer les tableaux de bord lorsqu'ils se comptent par dizaines et où il s'exécute.

Comme il ne fait qu'afficher, son adoption ne déplace aucune donnée. Une équipe peut garder Prometheus, PostgreSQL et son stockage de journaux à leur place et changer plus tard la couche d'affichage sans migrer une seule série. La dépendance au fournisseur et la facture restent donc du côté des sources. Les suites associant stockage et affichage font le compromis inverse : une facture et une connexion, mais des données plus difficiles à emporter ailleurs.

Le public façonne le tableau plus que l'outil. Une personne de l'ingénierie en astreinte veut une page répondant d'un coup d'œil à « est-ce en panne et où ? », avec des liens vers le détail. Une salle de personnes aux métiers variés demande moins de panneaux, plus grands, des titres simples et une ligne de seuil indiquant quand s'inquiéter. Le même Grafana sert les deux publics avec des tableaux différents ; une page conçue pour les deux à la fois ne sert souvent aucun des deux.

La gestion des tableaux de bord est le deuxième choix. Grafana peut les charger depuis des fichiers JSON sur disque et surveiller le dossier pour détecter les modifications, ce que sa documentation appelle le provisionnement. En gestion de versions, un tableau de bord bénéficie d'une revue, d'un historique et d'un moyen de revenir en arrière ; un nouvel environnement retrouve les mêmes écrans sans configuration par clics. Le prix est que modifier dans le navigateur cesse d'être la façon de changer les tableaux, et toute l'équipe doit l'accepter.

Licence et hébergement viennent en dernier. Sous AGPLv3, qui modifie Grafana et laisse utiliser cette version par d'autres sur un réseau doit leur proposer son code source. Une équipe exploitant un Grafana non modifié pour son personnel est dans une situation très différente d'un fournisseur livrant une version modifiée dans un produit ; ce dernier devrait lire la licence avec l'aide d'un juriste. L'auto-hébergement se résume à un binaire jusqu'au besoin d'une deuxième instance. Les instances partagent alors une base MySQL ou PostgreSQL ; Grafana Cloud permet de déléguer ce travail à Grafana Labs.

À quoi cela ressemble en pratique

Dans les deux cas où Grafana apparaît dans le travail de cette entreprise, il en est la partie visible, avec Prometheus derrière.

Dans une fonction d'ingénierie d'infrastructure de 2022 à 2024, le point de départ était une équipe informée des problèmes après les utilisateurs, faute de vue d'ensemble sur l'état des systèmes. Un tableau de supervision Grafana et Prometheus lui a donné cette vue. Les services clés ont été instrumentés, les métriques choisies pour leur pertinence et non pour leur apparence d'activité. Des seuils d'alerte ont été définis et rendus visibles là où l'équipe les verrait encore à temps pour agir. Les problèmes ont commencé à être détectés avant de s'aggraver, la fiabilité s'est améliorée et l'équipe est passée de l'extinction des incendies à l'anticipation.

L'autre cas visait un public mixte. Un tableau de bord sur un grand téléviseur dans les bureaux s'adressait aux collègues techniques comme non techniques. Sa deuxième version, sur Grafana et Prometheus, montrait des métriques choisies avec les équipes DevOps et d'ingénierie. Des pipelines les alimentaient en chiffres de performances de plusieurs systèmes, avec des alertes au franchissement d'un seuil. Les équipes détectaient et résolvaient les incidents 40 % plus vite et, devant les chiffres affichés à tout le bureau, la santé des systèmes est devenue une responsabilité collective.

Un tel écran mural se lit depuis l'autre bout d'une pièce par des personnes qui ne modifieront jamais une requête. Il gagne à présenter peu de grands panneaux, des unités dans les titres et des couleurs changeant seulement lorsqu'il faut agir. Grafana laisse chaque décision à la personne qui construit le tableau.

Où les problèmes surviennent

Un tableau grandit jusqu'à ne plus être lu. Chaque incident ajoute un panneau et rien n'en retire : un an plus tard, la page porte quarante petits graphiques et celui qui compte se trouve hors de la zone visible sans défilement. Donnez à chaque tableau un public et une question, placez la réponse à « le système va-t-il bien ? » dans la première ligne et déplacez les détails vers des tableaux de bord accessibles par des liens depuis cet écran. Un panneau que personne dans l'équipe ne sait expliquer est à supprimer.

Une correction dans le navigateur disparaît. Le tableau a été provisionné depuis un fichier. Avec `allowUiUpdates` à sa valeur prédéfinie false, Grafana refuse l'enregistrement avec le dialogue « Cannot save provisioned dashboard ». À true, il enregistre la modification dans sa base, puis l'écrase avec le fichier à la prochaine modification de celui-ci. Modifiez le fichier ou exportez le JSON depuis le dialogue d'enregistrement et enregistrez-le dans un commit ; gardez explicite le réglage du fournisseur de tableaux de bord :

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

L'écran mural ralentit le système qu'il observe. Chaque navigateur ouvert, téléviseur inclus, réexécute la requête de chaque panneau à chaque rafraîchissement. Dans sa configuration initiale, Grafana autorise un intervalle minimal de cinq secondes. Vingt panneaux rafraîchis toutes les cinq secondes produisent quatre requêtes par seconde depuis un seul écran. Rafraîchissez au rythme des changements des données, soit, pour Prometheus, avec un intervalle au moins égal à celui de collecte, relevez le minimum pour tous et précalculez les expressions lourdes avec des règles d'enregistrement dans la source :

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

Le même incident avertit les personnes deux fois. La règle est définie dans Prometheus, avec des alertes acheminées par Alertmanager, et à nouveau dans Grafana Alerting. Ou deux instances Grafana évaluent chaque règle, parce que la configuration des alertes a été oubliée à l'ajout de la seconde. Conservez chaque règle en un seul endroit et suivez les étapes de haute disponibilité des alertes Grafana avant la mise en service d'une deuxième instance.

Tableaux et utilisateurs changent selon le serveur qui répond. Deux instances Grafana derrière un répartiteur gardent chacune leur fichier SQLite : un tableau enregistré sur l'une manque sur l'autre et un utilisateur n'existe que sur une seule. Le guide de Grafana exige une base MySQL ou PostgreSQL commune pour la haute disponibilité.

Pour aller plus loin

- À propos de Grafana — La présentation du projet, de ses fonctions, éditions et projets open source associés.
- Sources de données — Les systèmes interrogeables et l'ajout et la configuration de chacun.
- Tableaux de bord — Panneaux, lignes, variables et modèle JSON de stockage des tableaux.
- Provisionner Grafana — Les sources et tableaux fournis par fichiers, et le sort des modifications faites dans le navigateur.
- Configurer Grafana en haute disponibilité — La base commune nécessaire à une deuxième instance et les étapes supplémentaires pour les alertes.
- Licences — La page de Grafana Labs sur le passage d'Apache 2.0 à AGPLv3 en 2021.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce que Grafana ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/grafana/