aws

ユーザーガイド

Amazon Managed Grafana



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Managed Grafana: ユーザーガイド

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスはAmazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Amazon Managed Grafana とは 1
サポート対象の リージョン 1
入門 4
ユーザー認証 4
必要なアクセス許可 5
最初のワークスペースを作成する 6
セットアップ AWS 10
にサインアップする AWS アカウント 10
管理アクセスを持つユーザーを作成する 11
ワークスペースの管理 13
Grafana のバージョンの違い 14
Grafana バージョン 10 14
Grafana バージョン 9 16
ワークスペースの作成 17
ワークスペースの作成 18
ユーザー認証 24
SAML 25
IAM アイデンティティセンター 38
Grafana バージョン 41
ワークスペースの更新後のトラブルシューティング 43
Enterprise プラグイン 43
エンタープライズプラグインに対するアクセス権管理 45
Grafana Labs へのリンク 46
AWS Marketplace エンタープライズユーザー向けのよくある質問 47
ワークスペース間でコンテンツを移行する 51
Workspace へのユーザーのアクセス権 51
ユーザーまたはグループにアクセス権限を付与する 51
権限不一致エラー 54
権限不一致に関するよくある質問 55
データソースおよび通知のアクセス権 56
を使用したリソースの作成 AWS CloudFormation 57
Amazon Managed Grafana と AWS CloudFormation テンプレート 58
の詳細 AWS CloudFormation 58
ネットワークアクセスコントロール 58

ネットワークアクセス制御の設定 60
Amazon VPC のデータに接続する 62
VPC 接続の仕組み 64
VPC への接続を作成する 65
VPC のトラブルシューティング 67
ワークスペースを設定する 72
API または AWS CLIを使用した設定 74
ワークスペースの削除 76
Grafana ワークスペースを使用する 77
Grafana とは 78
メトリクスとログを調べる 78
アラート 78
注釈 78
ダッシュボード変数 79
WorkSpace に接続する 79
ユーザー、チーム、およびアクセス許可 80
[ユーザー] 81
ユーザーロール 81
チームの管理 82
許可の使用 84
最初のダッシュボード 89
ダッシュボードの作成 89
Grafana プラグイン 97
プラグインカタログ 98
プラグインの管理 100
プラグインのインストールまたは削除 101
プラグインの更新 101
AWS データソースプラグイン 102
データソース 104
特別なデータソース 105
の使用 AWS Organizations 106
ビルトイン データソース 108
Enterprise データソース 292
Grafana バージョン 10 の使用 374
ダッシュボード 375
パネルと視覚化 462

Explore ... 663
相関関係 ... 690
アラート ... 696
Grafana バージョン 9 の使用 ... 819
ダッシュボード ... 820
パネルと視覚化 ... 886
Explore ... 1033
アラート ... 1050
Grafana バージョン 8 の使用 ... 1152
パネル ... 1152
ダッシュボード ... 1246
Explore ... 1275
Linking (リンク中) ... 1284
テンプレートおよび変数 ... 1292
Grafana アラート ... 1318
設定を変更する ... 1367
Amazon Managed Grafana プロファイルを編集する ... 1368
設定を編集する ... 1368
Amazon Managed Grafana セッションを表示する ... 1369
サポートバンドル ... 1369
サポートバンドルのコンポーネント ... 1370
サポートバンドルの作成 ... 1370
従来のアラート ... 1371
アラート設定 ... 1372
クラスタリング ... 1372
通知 ... 1372
アラートの実行 ... 1372
アラート通知 ... 1373
アラートの作成 ... 1378
アラートルールの一時停止 ... 1383
既存のアラートルールの表示 ... 1384
通知テンプレートの作成 ... 1385
アラートのトラブルシューティング ... 1386
Grafana API リファレンス ... 1387
トークンを使用した認証 ... 1388
サービスアカウント ... 1388

API キー ... 1393
アラート API ... 1397
アラートの取得 ... 1397
ID によるアラート取得 ... 1399
ID によるアラート一時停止 ... 1400
アラート通知チャネル API ... 1400
すべての通知チャネルの取得 ... 1401
すべての通知チャネルの取得 (ルックアップ) ... 1402
UID によるすべての通知チャネルの取得 ... 1403
Id によるすべての通知チャネルの取得 ... 1404
通知チャネルの作成 ... 1405
UID による通知チャネルの更新 ... 1406
Id による通知チャネルの更新 ... 1407
UID による通知チャネルの削除 ... 1409
Id による通知チャネルの削除 ... 1409
テスト通知チャネル ... 1410
注釈 API ... 1411
注釈の検索 ... 1411
注釈の作成 ... 1413
グラファイト形式の注釈の作成 ... 1414
注釈の更新 ... 1415
パッチ注釈 ... 1415
Id による注釈の削除 ... 1416
認証 API ... 1417
API キーの取得 ... 1417
API キーの作成 ... 1418
API キーの削除 ... 1418
ダッシュボード API ... 1419
ダッシュボードの作成/更新 ... 1420
uid によるダッシュボードの取得 ... 1425
uid によるダッシュボードの削除 ... 1427
ホームダッシュボードの取得 ... 1427
ダッシュボードタグの取得 ... 1429
ダッシュボードアクセス許可 API ... 1429
ダッシュボードのアクセス許可の取得 ... 1430
ダッシュボードのアクセス許可の更新 ... 1432

ダッシュボードバージョン API 1433
すべてのダッシュボードバージョンの取得 1433
ダッシュボードバージョンの取得 1435
ダッシュボードの復元 1437
ダッシュボードバージョンの比較 1439
データソース API 1441
すべてのデータソースの取得 1441
id による単一データソースの取得 1442
UID による単一データソースの取得 1443
名前による単一データソースの取得 1444
名前によるデータソース Id の取得 1445
データソースを作成します。 1445
既存のデータソースの削除 1449
Id によるデータソースの削除 1450
UID によるデータソースの削除 1451
名前によるデータソースの削除 1451
データソースのプロキシコール 1452
データソースのクエリ 1452
ID によるデータソースのクエリ 1454
データソースアクセス許可 API 1457
データソースのアクセス許可の有効化 1458
データソースのアクセス許可の無効化 1458
データソースのアクセス許可の取得 1459
データソースのアクセス許可の追加 1461
データソースのアクセス許可の削除 1462
外部グループ同期 API 1463
外部グループの取得 1463
外部グループの追加 1464
外部グループの削除 1465
フォルダ API 1465
フォルダの作成 1466
フォルダの更新 1467
すべてのフォルダの取得 1469
uid によるフォルダの取得 1470
id によるフォルダの取得 1470
uid によるフォルダの削除 1472

フォルダ/ダッシュボード検索 API 1473
フォルダとダッシュボードの検索 1473
フォルダアクセス許可 API 1475
フォルダのアクセス許可の取得 1476
フォルダに対するアクセス許可の更新 1477
組織 API 1479
現在の組織の取得 1479
現在の組織内のすべてのユーザーの取得 1480
現在の組織内のすべてのユーザーの取得 (ルックアップ) 1480
指定ユーザーの更新 1481
現在の組織のユーザーの削除 1482
現在の組織の更新 1482
現在の組織へのユーザーの追加 1483
プレイリスト API 1483
プレイリストの検索 1484
プレイリストを 1 つ取得 1485
プレイリスト項目の取得 1486
プレイリストダッシュボードの取得 1486
プレイリストの作成 1487
プレイリストの更新 1488
プレイリストの削除 1489
プラグイン API 1490
プラグインのインストール 1490
プラグインのアンインストール 1491
すべてのプラグインの取得 1491
プラグインの取得 1493
プラグインバージョンの取得 1496
設定 API 1498
現在のユーザー設定の取得 1499
現在のユーザー設定の更新 1499
現在の組織設定の取得 1500
現在の組織設定の更新 1500
スナップショット API 1501
新しいシェープショットの作成 1501
スナップショットのリストの取得 1503
key によるスナップショットの取得 1504

key によるスナップショットの削除 1505
deleteKey によるスナップショットの削除 1506
チーム API 1506
ページ分割によるチーム検索 1507
Id によるチームの取得 1508
チームの追加 1509
チームの更新 1510
Id によるチームの削除 1510
チームメンバーの取得 1511
チームメンバーを追加する 1512
チームのメンバーの削除 1513
チーム設定の取得 1514
チーム設定の更新 1514
ユーザー API 1515
ユーザーがメンバーであるチームの取得 1515
スナップショットのリストの取得 1516
ダッシュボードのスター解除 1516
実際のユーザーの認証トークンの取得 1517
実際のユーザーの認証トークンの取り消し 1518
オブザーバビリティソリューション 1520
Amazon EKS のモニタリング 1520
このソリューションについて 1521
コスト 1524
前提条件 1526
このソリューションの使用 1528
追跡されるメトリクスのリスト 1533
作成済みアラートのリスト 1540
トラブルシューティング 1550
JVM アプリケーションのモニタリング 1553
このソリューションについて 1554
コスト 1554
前提条件 1556
このソリューションの使用 1557
追跡されるメトリクスのリスト 1560
トラブルシューティング 1561
Kafka アプリケーションのモニタリング 1563

このソリューションについて 1564
コスト 1565
前提条件 1566
このソリューションの使用 1567
追跡されるメトリクスのリスト 1570
トラブルシューティング 1572
Tagging 1575
ワークスペースのタグ付け 1576
セキュリティ 1582
データ保護 1583
Amazon Managed Grafana でのデータ保護 1584
Identity and Access Management 1584
対象者 1585
アイデンティティを使用した認証 1585
ポリシーを使用したアクセスの管理 1589
Amazon Managed Grafana と IAM の連携方法 1592
アイデンティティベースのポリシーの例 1598
AWS 管理ポリシー 1603
トラブルシューティング 1620
サービス間の混乱した代理の防止 1622
サービスにリンクされたロールの使用 1623
他の AWS サービスのアクセス許可とポリシー 1628
サービス管理のアクセス許可 (単一アカウント用) 1628
組織の「サービス管理」のアクセス許可 1631
「カスタマー管理」のアクセス許可 1637
IAM 許可 1638
Amazon Managed Grafana のアクセス許可 1638
コンプライアンス検証 1639
耐障害性 1641
インフラストラクチャセキュリティ 1641
CloudTrail ログ 1642
CloudTrail の Amazon Managed Grafana 管理イベント 1644
Amazon Managed Grafana イベント例 1644
Grafana API イベント例 1648
セキュリティに関するベストプラクティス 1666
有効期間の短い API キーを使用する 1666

自己管理型 Grafana からの移行 1667
インターフェイス VPC エンドポイント 1667
インターフェイス Amazon VPC エンドポイントを使った Amazon Managed Grafana の使用 1668
Amazon Managed Grafana に AWS PrivateLink 接続するための VPC エンドポイントの作成 1669
ネットワークアクセスコントロールを使用して Grafana ワークスペースへのアクセスを制限する 1669
エンドポイントポリシーを使用して Amazon Managed Grafana API VPC エンドポイントへのアクセスを制御する 1669
Service Quotas 1672
ドキュメント履歴 1677
..... mdclxxxiii

Amazon Managed Grafana とは

Amazon Managed Grafana は、フルマネージド型で安全なデータ可視化サービスであり、複数のソースからの運用メトリクス、ログ、トレースを即座に照会、関連付け、視覚化できます。Amazon Managed Grafana を使用すると、拡張可能なデータサポートに対して定評があり、広く展開されているデータ視覚化ツールである Grafana を簡単に展開、運用、スケールできます。

Amazon Managed Grafana では、ワークスペースと呼ばれる論理的に分離された Grafana サーバーを作成します。次に、Grafana ダッシュボードと視覚化を作成して、メトリクス、ログ、トレースを分析できます。Grafana サーバーを実行するハードウェアを構築、パッケージ化、デプロイする必要はありません。

Amazon Managed Grafana は、論理的な Grafana サーバーのプロビジョニング、セットアップ、スケール、メンテナンスを管理するため、これらのタスクを自分で行う必要はありません。また、Amazon Managed Grafana には、シングルサインオン、データアクセスコントロール、監査レポートなど、コーポレートガバナンス要件に準拠するためのセキュリティ機能が組み込まれています。

Amazon Managed Grafana は、Amazon CloudWatch、Amazon OpenSearch Service、、 AWS X-Ray AWS IoT SiteWise Amazon Timestream、Amazon Managed Service for Prometheus などの運用データを収集する AWS データソースと統合されています。Amazon Managed Grafana には、サポートされている AWS サービスをデータソースとして追加するためのアクセス許可プロビジョニング機能が含まれています。Amazon Managed Grafana は、多くの一般的なオープンソース、サードパーティー、その他のクラウドデータソースもサポートしています。

ユーザー認証と認可のために、Amazon Managed Grafana は SAML 2.0 をサポートする ID プロバイダー (IdP) と統合でき、 AWS IAM Identity Centerと統合することもできます。

Amazon Managed Grafana は、ワークスペース内のアクティブユーザーごとに料金が設定されます。料金に関する詳細については、「Amazon Managed Grafana の料金」を参照してください。

サポート対象の リージョン

Amazon Managed Grafana では現在、次のリージョンがサポートされています。

リージョン名	リージョン	エンドポイント	プロトコル	
米国東部 (オハイオ)	us-east-2	grafana.us-east-2.amazonaws.com grafana.us-east-2.api.aws	HTTPS HTTPS	
米国東部 (バージニア北部)	us-east-1	grafana.us-east-1.amazonaws.com grafana.us-east-1.api.aws	HTTPS HTTPS	
米国西部 (オレゴン)	us-west-2	grafana.us-west-2.amazonaws.com grafana.us-west-2.api.aws	HTTPS HTTPS	
アジアパシフィック (ソウル)	ap-northeast-2	grafana.ap-northeast-2.amazonaws.com grafana.ap-northeast-2.api.aws	HTTPS HTTPS	
アジアパシフィック (シンガポール)	ap-southeast-1	grafana.ap-southeast-1.amazonaws.com grafana.ap-southeast-1.api.aws	HTTPS HTTPS	
アジアパシフィック (シドニー)	ap-southeast-2	grafana.ap-southeast-2.amazonaws.com grafana.ap-southeast-2.api.aws	HTTPS HTTPS	
アジアパシフィック (東京)	ap-northeast-1	grafana.ap-northeast-1.amazonaws.com grafana.ap-northeast-1.api.aws	HTTPS HTTPS	
欧州 (フランクフルト)	eu-central-1	grafana.eu-central-1.amazonaws.com grafana.eu-central-1.api.aws	HTTPS HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
欧州 (アイルランド)	eu-west-1	grafana.eu-west-1.amazonaws.com grafana.eu-west-1.api.aws	HTTPS HTTPS	
欧州 (ロンドン)	eu-west-2	grafana.eu-west-2.amazonaws.com grafana.eu-west-2.api.aws	HTTPS HTTPS	

Amazon Managed Grafana リソースの作成および使用方法について説明します

このチュートリアルは Amazon Managed Grafana の使用を開始するのに役立ちます。最初のワークスペースを作成してから、そのワークスペースの Grafana コンソールに接続します。

ワークスペースは、論理 Grafana サーバーです。アカウント内の各リージョンには、最大 5 つのワークスペースを設定できます。

> **Note**
>
> がない場合は AWS アカウント、まず Amazon Managed Grafana を使用する AWS ように をセットアップする方法を学習します。

トピック

- ユーザー認証
- 必要なアクセス許可
- 最初のワークスペースを作成する
- Amazon Managed Grafana を使用する AWS ように を設定する

ユーザー認証

Amazon Managed Grafana はワークスペース内のユーザー認証で以下のオプションをサポートしています。

- ID プロバイダー (IdP) に保存されているユーザー認証情報、Security Assertion Markup Language 2.0 (SAML 2.0) による認証
- AWS IAM Identity Center

SAML

SAML を使用する場合は、ユーザーを ID プロバイダーで作成済みである必要があります。Amazon Managed Grafana は SAML 2.0 をサポートする ID プロバイダーをサポートしています。詳細については、「Amazon Managed Grafana ワークスペースで SAML を使用する」を参照してください。

AWS IAM Identity Center

ワークスペースを作成し、認証 AWS IAM Identity Center に を使用することを選択すると、Amazon Managed Grafana はアカウントで IAM Identity Center をまだ使用していない場合にアクティブ化します。IAM Identity Center の詳細については、「 とは AWS IAM Identity Center」を参照してください。

Amazon Managed Grafana で IAM Identity Center を使用するには、アカウントで も AWS Organizations アクティブ化されている必要があります。まだアクティブ化していない場合、Amazon Managed Grafana は IAM Identity Center をアクティブ化するときにそれをアクティブ化します。Amazon Managed Grafana が Organizations を有効にすると、お客様の組織も作成されます。Organizations の詳細については、「AWS Organizationsとは」を参照してください。

> **Note**
>
> 既に AWS 組織のメンバーであるアカウントにワークスペースを作成するには、組織の管理アカウントで IAM Identity Center を有効にする必要があります。お客様が 2019 年 11 月 25 日以前に管理アカウントで IAM Identity Center を有効にした場合、管理アカウントでも IAM Identity Center 統合アプリケーションを有効にする必要があります。詳細については、「IAM Identity Center 統合アプリケーション」を参照してください。

必要なアクセス許可

認証に IdP と SAML を使用するワークスペースを作成するには、AWSGrafanaAccountAdministrator ポリシーがアタッチされた IAM プリンシパルにサインインする必要があります。

認可 AWS IAM Identity Center に を使用する最初のワークスペースを作成するには、少なくとも次のポリシーがアタッチされている IAM プリンシパルにサインオンする必要があります。

- AWSGrafanaAccountAdministrator
- AWSSSOMemberAccountAdministrator
- AWSSSODirectoryAdministrator

詳細については、「Amazon Managed Grafana ワークスペースとユーザーを IAM Identity Center を使用する単一スタンドアロンアカウントで作成および管理する」を参照してください。

最初のワークスペースを作成する

最初のワークスペースを作成するには、次の手順に従います。

Amazon Managed Grafana でのワークスペースの作成方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. [ワークスペースを作成する] を選択します。
3. [WorkSpace 名] ボックスに、ワークスペースの名前を入力します。

   オプションとして、ワークスペースの説明を入力します。
4. [次へ] を選択します。
5. [認証アクセス] では、[AWS IAM Identity Center]、[Security Assertion Markup Language (SAML)]、またはその両方を選択します。

   - AWS IAM Identity Center— IAM Identity Center を選択し、ご使用のアカウントで IAM Identity Center を有効にしていない場合は、最初の IAM Identity Center ユーザーを作成すると、それを有効にするようにプロンプトが表示されます。IAM Identity Center は、Amazon Managed Grafana ワークスペースへのアクセスのユーザー管理を処理します。

     IAM Identity Center を有効にするには、以下の手順に従います。

     a. [ユーザーの作成] を選択します。

     b. ユーザーのメールアドレス、名、姓を入力して、[ユーザーの作成] を選択します。このチュートリアルでは、Amazon Managed Grafana を試すために使用するアカウントの名前とメールアドレスを使用します。メールメッセージが送信され、IAM Identity Center のために、このアカウントのパスワードを作成するようプロンプトが表示されます。

> **Important**
>
> 作成したユーザーに Amazon Managed Grafana ワークスペースーへのアクセス許可が自動的に付与されるわけではありません。後のステップのワークスペースの詳細ページで、ワークスペースへのアクセス許可をユーザーに付与します。

- SAML — SAML を選択した場合、ワークスペースの作成後に SAML セットアップを完了します。

6. [次へ] を選択します。
7. この最初のワークスペースで、[サービス管理] が [アクセス許可タイプ] で選択されていることを確認します。この選択により、Amazon Managed Grafana は、このワークスペースに使用する AWS データソースに必要なアクセス許可を自動的にプロビジョニングできます。
8. このチュートリアルでは、[現在のアカウント] を選択します。
9. (オプション) このワークスペースでクエリするデータソースを選択します。この開始方法のチュートリアルでは、データソースを選択する必要はありません。ただし、リスト内のデータソースのいずれかでこのワークスペースを使用する場合は、ここで選択します。

   データソースを選択すると、Amazon Managed Grafana はデータソースごとに AWS Identity and Access Management (IAM) ポリシーを作成して、Amazon Managed Grafana がデータを読み取るアクセス許可を持つようにします。これで、これらのサービスを Grafana ワークスペースのデータソースとして設定することが完了するわけではありません。Grafana ワークスペースのコンソール内で完了できます。
10. (オプション) このワークスペースの Grafana アラートを Amazon Simple Notification Service (Amazon SNS) 通知チャネルに送信する場合は、[Amazon SNS] を選択します。これにより、Amazon Managed Grafana は、`grafana` で始まる `TopicName` 値を使用して、ご使用のアカウントの Amazon SNS トピックに対して発行する IAM ポリシーを作成できます。これで、ワークスペースの通知チャネルとしての Amazon SNSの設定が完了するわけではありません。ワークスペースの Grafana コンソール内で完了できます。
11. [次へ] を選択します。
12. ワークスペースの詳細を確認して、[ワークスペースの作成] を選択します。

    ワークスペースの詳細ページが表示されます。

    最初は、[ステータス] が [作成中] です。

> **Important**
>
> ステータスが [アクティブ] になるまで待ってから、次のいずれかを実行します。
>
> - SAML を使用する場合は、SAML セットアップを完了します。

- IAM Identity Center を使用する場合は、IAM Identity Center ユーザーにワークスペースへのアクセス許可を割り当てます。

現在のステータスを表示するには、ブラウザの更新が必要になる場合があります。

13. IAM Identity Center を使用する場合は、以下を実行します。

    a. [認証] タブで、[新しいユーザーまたはグループを割り当て] を選択します。

    b. ワークスペースへのアクセス許可を付与するユーザーの隣のチェックボックスをオンにして、[ユーザーの割り当て] を選択します。

    c. そのユーザーの隣のチェックボックスをオンにして、[アクション] ドロップダウンリストから [管理者にする] アクションを選択します。

    > **Important**
    >
    > Grafana ワークスペースのコンソールにサインインしてワークスペースを管理するには、ワークスペースごとに少なくとも 1 人のユーザーを `Admin` として割り当てます。

14. SAML を使用する場合は、次の手順に従います。

    a. [認証] タブの [Security Assertion Markup Language (SAML)] で、[設定を完了] を選択します。

    b. [インポート方法] に、以下のいずれかを選択します。

    - [URL] を選択して、IdP メタデータの URL を入力します。
    - [アップロードまたはコピー/貼り付け] を選択します。メタデータをアップロードする場合は、[ファイルを選択] を選択してメタデータファイルを選択します。または、コピーアンドペーストを使用する場合は、メタデータを [メタデータをインポート] にコピーします。

    c. [アサーション属性ロール] に、ロール情報を抽出する SAML アサーション属性の名前を入力します。

    d. [管理者ロール値] に、Amazon Managed Grafana ワークスペースの `Admin` ロールをすべて付与する必要がある IdP のユーザーロールを入力、または [ワークスペースへの管理者の割り当てをオプトアウトする] を選択します。

> **Note**
>
> [ワークスペースへの管理者の割り当てをオプトアウトする] を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Grafana ワークスペースコンソールを使用してワークスペースを管理することはできません。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

e. (オプション) さらに SAML 設定を入力するには、[詳細設定] を選択し、以下を 1 つ以上実行します。このすべてのフィールドはオプションとなります。

- [アサーション属性名] で、SAML ユーザーの完全な「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性ログイン] で、SAML ユーザーのユーザーサインイン名に使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性電子メール] で、SAML ユーザーの電子メール名として使用する SAML アサーション内の属性の名前を指定します。
- [ログイン有効期間 (分単位)] で、それを過ぎると再度サインインする必要がある、SAML ユーザーのサインイン有効期間を指定しします。
- [アサーション属性組織] で、ユーザー組織の「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性グループ] で、ユーザーグループの「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [許可される組織] では、IdP 内の特定の組織のメンバーであるユーザーのみにユーザーアクセスを制限できます。許可する 1 つ以上の組織を、カンマで区切って入力します。
- [エディタロールの値] で、Amazon Managed Grafana ワークスペースで `Editor` ロールをすべて付与する必要がある IdP のユーザーロールを入力します。1 つ以上のロールを、カンマで区切って入力します。

> **Note**
>
> 管理者またはエディタロールを特別に割り当てられていないユーザーは、ビューワーとして割り当てられます。

f. [SAML 設定の保存] を選択します。

15. ワークスペースの詳細ページで、[Grafana ワークスペース URL] の下に表示される URL を選択します。
16. ワークスペースの URL を選択すると、Grafana ワークスペースコンソールのランディングページに移動します。次のいずれかを行います：

    - [SAML でサインイン] を選択し、名前とパスワードを入力します。
    - でサインイン AWS IAM Identity Centerを選択し、この手順の前半で作成したユーザーの E メールアドレスとパスワードを入力します。これらの認証情報は、IAM Identity Center のパスワードを作成するようプロンプトが表示される、Amazon Managed Grafana からのメールに応答した場合にのみ機能します。

    Grafana ワークスペース、または論理 Grafana サーバーに移動しました。データのクエリ、視覚化、分析に対するデータソースの追加を開始できます。詳細については、「Grafana ワークスペースを使用する」を参照してください。

Amazon Managed Grafana を使用する AWS ように を設定する

このセクションのタスクを完了して、 AWS を初めてセットアップします。が既にある場合は AWS アカウント、 から始めますAmazon Managed Grafana リソースの作成および使用方法について説明します。

にサインアップすると AWS、 は Amazon Managed Grafana を含む AWSのすべてのサービス AWS アカウント に自動的にアクセスできます。ただし、料金が発生するのは実際に使用したサービスの分だけです。

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しまたはテキストメッセージを受け取り、電話キーパッドで検証コードを入力するように求められます。

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があ

ります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、ルートユーザーアクセスが必要なタスクの実行にはルートユーザーのみを使用するようにしてください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 を保護し AWS IAM Identity Center、 を有効にして管理ユーザーを作成します。

を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、アカウント所有者AWS Management Consoleとして にサインインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドのルートユーザーとしてサインインするを参照してください。
2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、IAM ユーザーガイドの AWS アカウント「ルートユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。
2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

   を ID ソース IAM アイデンティティセンターディレクトリ として使用する方法のチュートリアルについては、AWS IAM Identity Center 「 ユーザーガイド」の「デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン「 ユーザーガイド」の AWS 「 アクセスポータルにサインインする」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」を参照してください。
2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「グループの結合」を参照してください。

Amazon Managed Grafana でワークスペース、ユーザー、ポリシーを管理する

Amazon Managed Grafana を使用するには、Grafana ワークスペースを作成します。Grafana ワークスペースは論理的な Grafana サーバーで、ここでダッシュボードや視覚的表現を作成して、メトリクス、ログ、トレースを分析できます。ユーザーを追加し、そのユーザーに対してワークスペースの管理、編集、閲覧権限を管理します。

ワークスペースを Grafana の新しいバージョンにアップグレードしたり、エンタープライズプラグインのサポートを追加して、さまざまなデータソースにアクセスできるようにすることができます。また、ワークスペースへのネットワークアクセスも管理できます。を使用して、Amazon Managed Grafana ワークスペースを作成および管理できます AWS CloudFormation。

このセクションのトピックでは、Amazon Managed Grafana でワークスペース、ユーザー、ポリシーを管理する方法について説明します。

トピック

- Grafana のバージョン間の違い
- Amazon Managed Grafana ワークスペースを作成する
- Amazon Managed Grafana ワークスペースでユーザーを認証する
- ワークスペースのバージョンを更新する
- Enterprise プラグインへのアクセス許可の管理
- Amazon Managed Grafana ワークスペース間でコンテンツを移行する
- Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する
- データソースおよび通知チャネルのアクセス権管理
- を使用した Amazon Managed Grafana リソースの作成 AWS CloudFormation
- Amazon Managed Grafana ワークスペースに対するネットワークからのアクセス制御を設定する
- Amazon Managed Grafana から Amazon VPC のデータソースまたは通知チャネルに接続する
- Amazon Managed Grafana ワークスペースを設定する
- Amazon Managed Grafana ワークスペースを削除する

Grafana のバージョン間の違い

Grafana ワークスペースを作成するときは、作成する Grafana バージョンを選択する必要があります。Grafana バージョン 8、9、10 と互換性のあるバージョンを選択できます。いずれも、以前のバージョンから機能が追加されました。以下のトピックでは、バージョン 9 および 10 の変更点について説明します。これには、バージョン 9 で使用する機能を破損する可能性のあるバージョン 10 の変更点が含まれます。

> **Note**
>
> Grafana ワークスペースを使用するためのバージョン固有のドキュメントは、「Grafana バージョン 8 での作業」、「Grafana バージョン 9 での作業」、および「Grafana バージョン 10 での作業」トピックで確認できます。

バージョン別の詳細なメモと Grafana Labs からの詳細については、「Grafana Labs ドキュメント」の「Grafana の新機能」を参照してください。

Grafana バージョン 10

Grafana バージョン 10 では、以下の機能が追加されました。

- [相関関係] – 相関関係は、あるデータソース内のデータを使用して別のデータソース内のデータにクエリを実行する方法を定義し、Explore による視覚化が、表示されたデータに関連するクエリを簡単に実行できるようにします。詳細については、「Grafana バージョン 10 の相関関係」を参照してください。
- サブフォルダ – ダッシュボードを整理するときに、サブフォルダを使用してネストされた階層を作成できるようになりました。詳細については、「ダッシュボードフォルダの作成」を参照してください。
- アラート – Grafana アラートは、サイレントアラートをサポートするようになりました。さらに、Grafana アラートは通知を 3 回送信しなくなりました。
- [アラートのアップグレードプレビュー] – 従来のダッシュボードアラートから Grafana アラートにアップグレードする前に、アラートがどのように表示されるかを確認し、移行時に適用される変更を加えることもできます。詳細については、「従来のダッシュボードアラートを Grafana アラートに移行する」を参照してください。Grafana Labs は、Grafana バージョン 11 以降では、従来のダッシュボードアラートがサポート対象外となることを発表しました。

- サポートバンドル – サポートバンドルは、Grafana ワークスペースに関する情報を収集して製品サポートと共有するための簡単な方法を提供します。移行、プラグイン、設定などのデータを含むサポートバンドルをすばやく作成できます。詳細については、「サポート用情報の収集」を参照してください。
- 新しいビジュアライゼーション — 3 つの新しいビジュアライゼーションが利用可能です。XY チャート、データグリッド、および トレンドパネルはすべて、バージョン 10 と互換性のあるワークスペースで使用できます。また、バージョン 9 のワークスペースでも、XY チャートを使用できます。
- PagerDuty – エンタープライズ向けプラグインに PagerDuty のプラグインが搭載されました。
- [トランスフォーメーションの再設計] – トランスフォーメーションタブでは、ユーザーエクスペリエンスとビジュアルデザインが改善されています。トランスフォーメーションは分類され、各トランスフォーメーションタイプには、適切なタイプを選択するのに役立つ解説があります。
- Prometheus メトリクス百科事典 – Prometheus クエリビルダーにある Prometheus メトリクスのメトリクスドロップダウンが、ページ分割された検索可能なメトリクス百科事典に置き換わりました。
- API キー UI の廃止 – サービスアカウントは、Grafana HTTP API への呼び出しを認証する推奨方法です。Grafana Labs は API キーの廃止に取り組んでおり、ワークスペースユーザーインターフェイスを通じて API キーを作成することはできません。API キーは、 AWS API を介してのみ作成できます。

  Grafana Labs による API キー廃止の詳細については、Grafana GitHub の問題リストにある「APIKeys: API キーの終了」を参照してください。

重要な変更

Grafana バージョン 10.4 リリースには、Grafana バージョン 9.5 から 10.4 への変更が含まれています。Grafana バージョン 10.0 および 10.3 には、場合によっては機能を損なう可能性のある変更がいくつかありました。新しいバージョンに更新する場合は、本番稼働用ワークスペースを更新する前に、本番稼働用以外の環境でテストすることをお勧めします。

以下の変更は、Grafana バージョン 10 に更新する一部のユーザーに影響する可能性があります。

- Angular の廃止 – Angular を使用するプラグインは、Grafana の今後のリリースではサポートされなくなります。バージョン 10 では、Angularを使用するパネルにおいて、廃止された機能を使用していることを示すバナーが表示され、将来のバージョンでは機能しないという通知が表示されます。

- CloudWatch のエイリアスを削除 – CloudWatch クエリエディタのエイリアスパターンがラベル (動的ラベル) に置き換えられました。

  エイリアスフィールドを使用する任意のダッシュボードを開き、保存してください。エイリアスはラベルに自動的に移行されます。
- 古いプラグインはアップグレードが必要 – Athena および Amazon Redshift データソースのプラグインを Grafana v10 ワークスペースで更新する必要があります。Athena データソースプラグインはバージョン 2.9.3 以降である必要があります。また、Amazon Redshift データソースプラグインはバージョン 1.8.3 以降である必要があります。

  プラグインのインストールまたはアップグレードに関する詳細については、「プラグインカタログによるプラグインの検索」を参照してください。
- DoiT BigQuery プラグインは今後サポート対象外 – DoiT BigQuery データソースプラグインはサポートされなくなりました。代わりに、公式の Grafana Labs BigQuery データソースプラグインをご使用ください。
- [トランスフォーメーションの変更] – Grafana バージョン 10 では、フィールド名とキーにいくつかのバグ修正変更を行いました。詳細については、Grafana Labs のドキュメントにある「変換に関する重要な変更」を参照してください。
- データソースへのアクセス許可 API – データソースの権限にアクセスするためのエンドポイントが変更されました。詳細については、Grafana Labs ドキュメントにある「データソースのアクセス許可の変更」を参照してください。

変更の中断、およびプラグインデベロッパーに影響する重要な変更の詳細については、「Grafana Labs ドキュメント」にある以下のトピックを参照してください。

- Grafana v10.0 の重要な変更
- Grafana v10.3 の重要な変更

Grafana バージョン 9

Grafana v9 では、以下の機能が追加されました。

- アラート: Grafana が管理するアラートルールがグループ名をサポートするようになりました。
- [Explore]: Explore ビュー内からダッシュボードを作成します。
- Prometheus クエリ: Prometheus クエリの新しいクエリビルダー (PromQL を使用) を使用すると、クエリの書き込みが容易になります。

- Loki クエリ: Loki クエリの新しいクエリビルダー (LogQL を使用) を使用すると、クエリの書き込みが容易になります。
- API トークン/サービスアカウント: サービスアカウントは Grafana でのマシンアクセスを簡素化し、API トークンの管理に役立ちます。
- プラグイン管理: プラグイン管理を有効にすると、ワークスペースにコミュニティプラグインをインストール、削除、更新できます。これにより、より多くのデータソースや視覚化にアクセスでき、使用する各プラグインのバージョン全体を管理できます。
- メトリクスへのトレース: トレースデータソースを設定して、クエリとタグを含むメトリクスへのリンクを追加します。
- Canvas パネル: 静的要素と動的要素を含む新しいパネル視覚化で、画像とオーバーレイテキストを含むデータ駆動型のカスタムパネルを作成します。
- インターフェイスの再編成: Grafana コンソールでのナビゲーションを容易にして UI を更新しました。
- CloudWatch: Amazon CloudWatch データソースは、 間 AWS アカウント および 間でメトリクスをモニタリングできるようになりました AWS リージョン。
- ログ: ログの詳細のインターフェイスが改善されました。
- 全般: バグ修正と全体的に改良を行いました。

Amazon Managed Grafana ワークスペースを作成する

ワークスペースは、論理 Grafana サーバーです。アカウント内の各リージョンには、最大 5 つのワークスペースを設定できます。

必要なアクセス許可

ワークスペースを作成するには、AWSGrafanaAccountAdministrator ポリシーがアタッチされている AWS Identity and Access Management (IAM) プリンシパルにサインインする必要があります。

IAM Identity Center を使用して認証を行う最初のワークスペースを作成するには、IAM プリンシパルに以下の追加ポリシー (または同等のアクセス許可) も付与されている必要があります。

- AWSSSOMemberAccountAdministrator
- AWSSSODirectoryAdministrator

詳細については、「Amazon Managed Grafana ワークスペースとユーザーを IAM Identity Center を使用する単一スタンドアロンアカウントで作成および管理する」を参照してください。

ワークスペースの作成

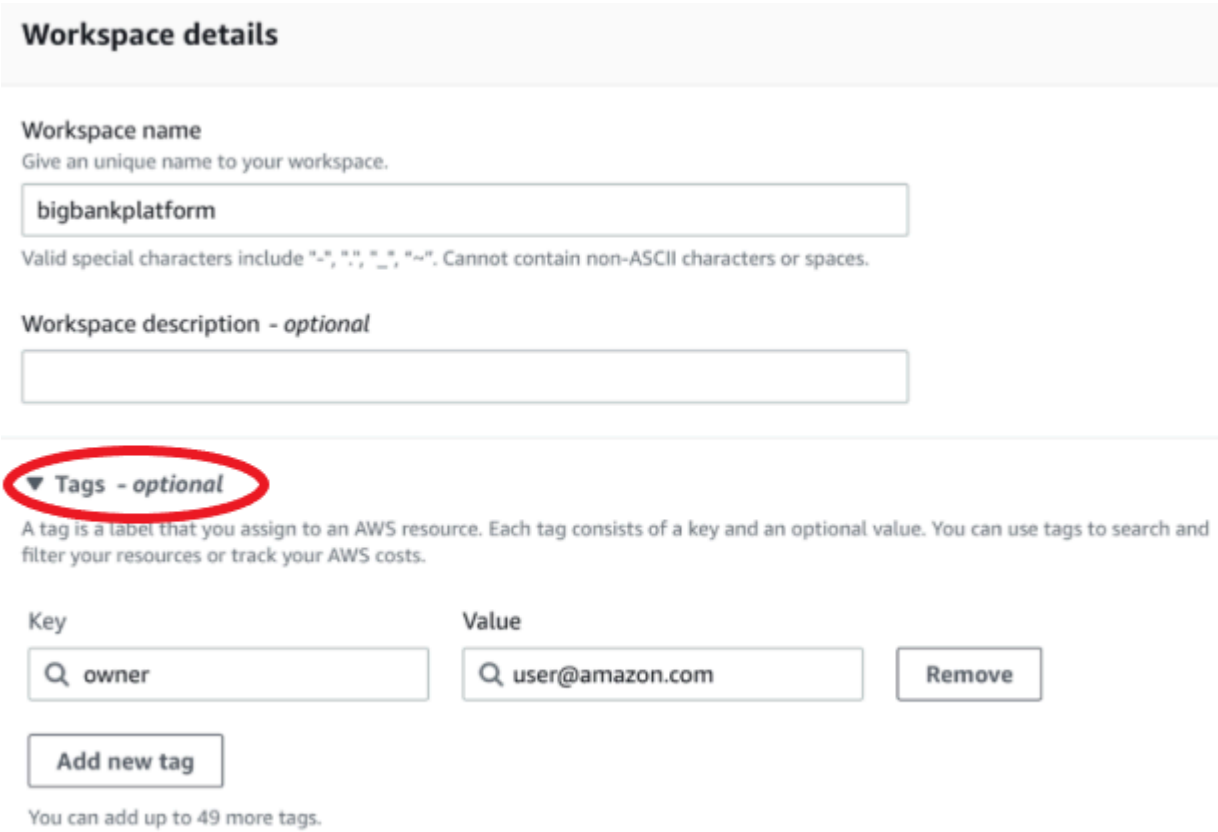
Amazon Managed Grafana ワークスペースを新規に作成する手順について説明します。

Amazon Managed Grafana でのワークスペースの作成方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. [ワークスペースを作成する] を選択します。
3. [ワークスペースの詳細] ウィンドウで、[ワークスペース名] にワークスペースの名前を入力します。

   オプションとして、ワークスペースの説明を入力します。

   必要に応じて、このワークスペースに関連付けるタグを追加します。タグはワークスペースの識別と整理に役立ち、 AWS リソースへのアクセスを制御するためにも使用できます。たとえば、ワークスペースにタグを割り当てて、そのタグを持つ限定されたグループやロールだけがワークスペースにアクセスできる権限を持つように設定することができます。タグベースのアクセスコントロールの詳細については、IAM ユーザーガイドの「タグを使用した AWS リソースへのアクセスの制御」を参照してください。



4. ワークスペースの [Grafana バージョン] を選択します。バージョン 8、9 または 10 を選択できます。バージョン間の違いについては「Grafana のバージョン間の違い」を参照してください。
5. [次へ] を選択します。
6. [認証アクセス] では、[AWS IAM Identity Center]、[Security Assertion Markup Language (SAML)]、またはその両方を選択します。詳細については、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。

   - IAM アイデンティティセンター — IAM アイデンティティセンターを選択し、アカウント AWS IAM Identity Center で をまだ有効にしていない場合は、最初の IAM アイデンティティセンターユーザーを作成して有効にするように求められます。IAM Identity Center は、Amazon Managed Grafana ワークスペースへのアクセスのユーザー管理を処理します。

     IAM Identity Center を有効にするには、以下の手順に従います。

     a. [ユーザーの作成] を選択します。
     b. ユーザーのメールアドレス、名、姓を入力して、[ユーザーの作成] を選択します。このチュートリアルでは、Amazon Managed Grafana を試すために使用するアカウントの名前とメールアドレスを使用します。IAM Identity Center 用のこのアカウントのパスワードを作成するように求めるメールが送信されます。

     > **Important**
     >
     > 作成したユーザーに Amazon Managed Grafana ワークスペースーへのアクセス許可が自動的に付与されるわけではありません。後のステップのワークスペースの詳細ページで、ワークスペースへのアクセス許可をユーザーに付与します。

   - SAML — SAML を選択した場合、ワークスペースの作成後に SAML セットアップを完了します。
7. [サービス管理] または [カスタマー管理] を選択します。

   サービスマネージドを選択した場合、Amazon Managed Grafana は自動的に IAM ロールを作成し、このワークスペースに使用するこのアカウントの AWS データソースに必要なアクセス許可をプロビジョニングします。

   これらのロールとアクセス許可を自分で管理する場合、[カスタマー管理] を選択します。

組織のメンバーアカウントにワークスペースを作成する場合、[サービス管理] を選択できるようにするには、そのメンバーアカウントが組織内の被委任管理者アカウントである必要があります。被委任管理者の追加の詳細については、「被委任管理者アカウントの登録」を参照してください。

8. (任意) Amazon 仮想プライベートクラウド (VPC) にはこのページで接続できますが、後で接続することもできます。詳細については Amazon Managed Grafana から Amazon VPC のデータソースまたは通知チャネルに接続するを参照してください。
9. (任意) このページでは、以下を含む他のワークスペース設定オプションを選択できます。

   - Grafana アラートを有効にする。Grafana アラートを使用すると、Grafana アラートと Prometheus で定義されたアラートを、Grafana ワークスペース内の単一のアラートインターフェイス内に表示することができます。

     バージョン 8 または 9 を実行しているワークスペースでは、Grafana アラートに対して複数の通知が送信されます。Grafana で定義されたアラートを使用する場合は、ワークスペースをバージョン 10.4 以降として作成することをお勧めします。
   - Grafana 管理者がこのワークスペースのプラグインを管理できるようにします。プラグイン管理を有効にしない場合、管理者はワークスペースのプラグインをインストール、アンインストール、または削除できなくなります。Amazon Managed Grafana で使用できるデータソースと視覚的表現パネルの種類が制限される場合があります。

   これらの設定は、ワークスペースの作成後に変更できます。ワークスペースの設定の詳細については、「Amazon Managed Grafana ワークスペースを設定する」を参照してください。
10. (任意) ワークスペースにネットワークアクセスコントロールを追加できます。ネットワークアクセスコントロールを追加するには、[アクセス制限] を選択します。ネットワークアクセスコントロールは、ワークスペースを作成した後で有効にすることもできます。

    ネットワークアクセスコントロールの詳細については、「Amazon Managed Grafana ワークスペースに対するネットワークからのアクセス制御を設定する」を参照してください。
11. [次へ] を選択します。
12. サービスマネージドを選択した場合は、現在のアカウントを選択して、Amazon Managed Grafana が現在のアカウントでのみ AWS データを読み取れるようにするポリシーとアクセス許可を自動的に作成します。

    管理アカウントまたは組織内の委任管理者アカウントにワークスペースを作成する場合は、組織を選択して、指定した組織単位内の他のアカウントの AWS データを読み取ることができるポ

リシーとアクセス許可を Amazon Managed Grafana が自動的に作成するようにできます。被委任管理者の追加の詳細については、「被委任管理者アカウントの登録」を参照してください。

> **Note**
>
> 組織の管理アカウントに Amazon Managed Grafana ワークスペースなどのリソースを作成することは、 AWS セキュリティのベストプラクティスに違反します。

a. Organization を選択し、 AWS CloudFormation StackSets を有効にするように求められた場合は、信頼されたアクセスを有効にするを選択します。次に、Amazon Managed Grafana がデータを読み取る組織 AWS Organizations 単位 (OUs) を追加します。これにより、Amazon Managed Grafana が、選択した各 OU 内のすべてのアカウントからデータを読み取ることができます。

b. [組織]] を選択した場合は、[データソースと通知チャネル - 任意] を選択します。

13. このワークスペースでクエリを実行する AWS データソースを選択します。データソースを選択すると、Amazon Managed Grafana がこれらのソースからデータを読み取るのに必要な IAM ロールとアクセス許可を作成できるようになります。ただし、データソースは Grafana ワークスペースコンソールで追加する必要があります。

14. (オプション) このワークスペースの Grafana アラートを Amazon Simple Notification Service (Amazon SNS) 通知チャネルに送信する場合は、[Amazon SNS] を選択します。これにより、Amazon Managed Grafana は、`grafana` で始まる `TopicName` 値を使用して、ご使用のアカウントの Amazon SNS トピックに対して発行する IAM ポリシーを作成できます。これで、ワークスペースの通知チャネルとしての Amazon SNSの設定が完了するわけではありません。ワークスペースの Grafana コンソール内で完了できます。

15. [次へ] を選択します。

16. ワークスペースの詳細を確認して、[ワークスペースの作成] を選択します。

    ワークスペースの詳細ページが表示されます。

    最初は、[ステータス] が [作成中] です。

> **Important**
>
> ステータスが [アクティブ] になるまで待ってから、次のいずれかを実行します。
>
> - SAML を使用する場合は、SAML セットアップを完了します。

- IAM Identity Center を使用する場合は、IAM Identity Center ユーザーにワークスペースへのアクセス許可を割り当てます。

現在のステータスを表示するには、ブラウザの更新が必要になる場合があります。

17. IAM Identity Center を使用する場合は、以下を実行します。

    a. [認証] タブで、[新しいユーザーまたはグループを割り当て] を選択します。

    b. ワークスペースへのアクセス許可を付与するユーザーの隣のチェックボックスをオンにして、[ユーザーの割り当て] を選択します。

    c. ユーザーの横にあるチェックボックスを選択し、[管理者にする] を選択します。

> **Important**
>
> Grafana ワークスペースのコンソールにサインインしてワークスペースを管理するには、ワークスペースごとに少なくとも 1 人のユーザーを `Admin` として割り当てます。

18. SAML を使用する場合は、次の手順に従います。

    a. [認証] タブの [Security Assertion Markup Language (SAML)] で、[設定を完了] を選択します。

    b. [インポート方法] に、以下のいずれかを選択します。

    - [URL] を選択して、IdP メタデータの URL を入力します。
    - [アップロードまたはコピー/貼り付け] を選択します。メタデータをアップロードする場合は、[ファイルを選択] を選択してメタデータファイルを選択します。または、コピーアンドペーストを使用する場合は、メタデータを [メタデータをインポート] にコピーします。

    c. [アサーション属性ロール] に、ロール情報を抽出する SAML アサーション属性の名前を入力します。

    d. [管理者ロール値] に、Amazon Managed Grafana ワークスペースの `Admin` ロールをすべて付与する必要がある IdP のユーザーロールを入力、または [ワークスペースへの管理者の割り当てをオプトアウトする] を選択します。

> **Note**
>
> [ワークスペースへの管理者の割り当てをオプトアウトする] を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、コンソールを使用してワークスペースを管理することはできません。ワークスペースに対する管理変更は、Amazon Managed Grafana API を使用してのみ実行できます。

e. (オプション) さらに SAML 設定を入力するには、[詳細設定] を選択し、以下を 1 つ以上実行します。このすべてのフィールドはオプションとなります。

- [アサーション属性名] で、SAML ユーザーの完全な「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性ログイン] で、SAML ユーザーのユーザーサインイン名に使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性電子メール] で、SAML ユーザーの電子メール名として使用する SAML アサーション内の属性の名前を指定します。
- [ログイン有効期間 (分単位)] で、それを過ぎると再度サインインする必要がある、SAML ユーザーのサインイン有効期間を指定しします。デフォルトは 1 日で、最大は 30 日間です。
- [アサーション属性組織] で、ユーザー組織の「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性グループ] で、ユーザーグループの「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [許可される組織] では、IdP 内の特定の組織のメンバーであるユーザーのみにユーザーアクセスを制限できます。許可する 1 つ以上の組織を、カンマで区切って入力します。
- [エディタロールの値] で、Amazon Managed Grafana ワークスペースで `Editor` ロールをすべて付与する必要がある IdP のユーザーロールを入力します。1 つ以上のロールを、カンマで区切って入力します。

f. [SAML 設定の保存] を選択します。

19. ワークスペースの詳細ページで、[Grafana ワークスペース URL] の下に表示される URL を選択します。
20. ワークスペースの URL を選択すると、Grafana ワークスペースコンソールのランディングページに移動します。次のいずれかを行います：

- [SAML でサインイン] を選択し、名前とパスワードを入力します。

- でサインイン AWS IAM Identity Centerを選択し、この手順の前半で作成したユーザーの E メールアドレスとパスワードを入力します。これらの認証情報は、IAM Identity Center のパスワードを作成するようプロンプトが表示される、Amazon Managed Grafana からのメールに応答した場合にのみ機能します。

  Grafana ワークスペース、または論理 Grafana サーバーに移動しました。データのクエリ、視覚化、分析に対するデータソースの追加を開始できます。詳細については、「Grafana ワークスペースを使用する」を参照してください。

詳細の参照先

> **Tip**
>
> を使用して、Amazon Managed Grafana ワークスペースの作成を自動化できます AWS CloudFormation。詳細については、「を使用した Amazon Managed Grafana リソースの作成 AWS CloudFormation」を参照してください。

Amazon Managed Grafana ワークスペースでユーザーを認証する

個々のユーザーはワークスペースにサインインし、ダッシュボードの編集や表示を行います。ワークスペースにユーザーを割り当てた後、そのユーザーにはユーザー、編集者、または管理者権限を付与することができます。最初に、ID プロバイダーを作成して (既存の ID プロバイダーがあればこれを使用して) ユーザーを認証します。

ユーザーは、IAM を使用するのではなく、組織の ID プロバイダーを使ったシングルサインオンによって、Amazon Managed Grafana ワークスペースの Grafana コンソールを使用するように認証されます。各ワークスペースでは、次の認証方法のいずれかまたは両方を使用できます。

- Security Assertion Markup Language 2.0 (SAML 2.0) をサポートする ID プロバイダー (IdP) に保存されたユーザー認証情報
- AWS IAM Identity Center. AWS Single-sign-on (AWS SSO) が IAM アイデンティティセンターにブランド変更されました。

SAML、IAM Identity Center、またはその両方を使用するかはワークスペースごとに選択できます。最初にどちらを使用するかを選択した場合も、後でその他の方法に切り替えることができます。

ワークスペース内の機能にアクセスする前に、ユーザー (またはユーザーが属するグループ) にワークスペースへのアクセス権を付与する必要があります。ユーザーに権限を付与する方法の詳細については、「Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する」を参照してください。

トピック

- Amazon Managed Grafana ワークスペースで SAML を使用する
- Amazon Managed Grafana ワークスペース AWS IAM Identity Center で を使用する

Amazon Managed Grafana ワークスペースで SAML を使用する

> **Note**
>
> 現在、Amazon Managed Grafana はワークスペースに対する IdP 開始のログインをサポートしていません。SAML アプリケーションは、Relay State を空白で設定する必要があります。

SAML 認証を使用すると、既存の ID プロバイダーを使用して Amazon Managed Grafana ワークスペースの Grafana コンソールにシングルサインオンでログインできるようになります。IAM を介して認証する代わりに Amazon Managed Grafana の SAML 認証を使用すると、サードパーティーの ID プロバイダーを使用したログイン、アクセスコントロールの管理、データの検索、視覚的表現の構築を行うことができます。Amazon Managed Grafana は、SAML 2.0 標準を使用し、Azure AD、CyberArk、Okta、OneLogin、および Ping Identity と連携したアプリケーションを構築およびテストした ID プロバイダーをサポートします。

ワークスペース作成時の SAML 認証の設定方法の詳細については、「ワークスペースの作成」を参照してください。

SAML 認証フローでは、Amazon Managed Grafana ワークスペースがサービスプロバイダー (SP) として機能し、IdP と対話してユーザー情報を取得します。SAML の詳細については、「Security Assertion Markup Language」を参照してください。

IdP のグループを Amazon Managed Grafana ワークスペースのチームにマッピングし、それらのチームにきめ細かなアクセス権限を設定することができます。また、IdP で定義されている組織ロールを Amazon Managed Grafana ワークスペースのロールにマッピングすることもできます。例えば、IdP で開発者ロールが定義されている場合、そのロールを Amazon Managed Grafana ワークスペースの Grafana 管理者ロールにマッピングすることができます。

> **Note**
>
> IdP と SAML を認証に使用する Amazon Managed Grafana ワークスペースを作成する際は、AWSGrafanaAccountAdministrator ポリシーがアタッチされている IAM プリンシパルにサインインしている必要があります。

Amazon Managed Grafana ワークスペースにサインインするには、ユーザーはワークスペースの Grafana コンソールのホームページにアクセスし、[SAML を使用してログイン] を選択します。ワークスペースは SAML 設定を読み込み、認証のためにユーザーを IdP にリダイレクトします。ユーザーは IdP ポータルにサインイン認証情報を入力し、有効なユーザーである場合、IdP は SAML アサーションを発行し、ユーザーを Amazon Managed Grafana ワークスペースにリダイレクトします。Amazon Managed Grafana は、SAML アサーションが有効であることを確認し、ユーザーはサインインしてワークスペースを使用できるようになります。

Amazon Managed Grafana は、次の SAML 2.0 バインディングをサポートしています。

- サービスプロバイダー (SP) から ID プロバイダー (IdP):
  - HTTP-POST バインディング
  - HTTP-Redirect バインディング
- ID プロバイダー (IdP) からサービスプロバイダー (SP):
  - HTTP-POST バインディング

Amazon Managed Grafana は、署名および暗号化されたアサーションをサポートしていますが、署名または暗号化されたリクエストはサポートしていません。

Amazon Managed Grafana は SP 開始のリクエストをサポートしており、IdP 開始のリクエストはサポートしていません。

アサーションマッピング

SAML 認証フロー中、Amazon Managed Grafana はアサーションコンシューマーサービス (ACS) コールバックを受け取ります。このコールバックには、認証されるユーザーに関連するすべての情報が埋め込まれた SAML レスポンスが含まれています。Amazon Managed Grafana はそのレスポンスを解析して、内部データベース内でユーザーを作成 (または更新) します。

Amazon Managed Grafana がユーザー情報をマッピングする際、アサーション内の個々の属性が表示されます。これらの属性はキーと値のペアと考えることができますが、それ以上の情報も含んでいます。

Amazon Managed Grafana は、これらの値をどのキーから取得するかを変更できる設定オプションを提供しています。

Amazon Managed Grafana コンソールを使用して、次の SAML アサーション属性を Amazon Managed Grafana の値にマッピングできます。

- [アサーション属性ロール]に、ユーザーロールとして使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性名] で、SAML ユーザーの完全な「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性ログイン] で、SAML ユーザーのユーザーサインイン名に使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性電子メール] で、SAML ユーザーの電子メール名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性組織] で、ユーザー組織の「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性グループ] で、ユーザーグループの「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [許可される組織] では、IdP 内の特定の組織のメンバーであるユーザーのみにユーザーアクセスを制限できます。
- [エディタロールの値] に、Amazon Managed Grafana ワークスペースで `Editor` ロールをすべて付与する必要がある IdP のユーザーロールを入力します。

ID プロバイダーへの接続

次の外部 ID プロバイダーは Amazon Managed Grafana でテストされており、SAML で Amazon Managed Grafana を設定するのに役立つアプリケーションをアプリケーションディレクトリまたはギャラリーで直接提供しています。

トピック

- Azure AD を使用するように Amazon Managed Grafana を設定する
- CyberArk を使用するように Amazon Managed Grafana を設定する

- Okta を使用するように Amazon Managed Grafana を設定する
- OneLogin を使用するように Amazon Managed Grafana を設定する
- Ping Identity を使用するように Amazon Managed Grafana を設定する

Azure AD を使用するように Amazon Managed Grafana を設定する

次の手順で、Azure Active Directory を ID プロバイダーとして使用するように Amazon Managed Grafana を設定します。この手順は、Amazon Managed Grafana ワークスペースを既に作成していることを前提としています。ワークスペースの ID、URL、および AWS リージョン の情報をお手元に用意してください。

ステップ 1: Azure Active Directory での作業手順

Azure Active Directory で以下手順を実行します。

Azure Active Directory を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Azure コンソールに管理者としてサインインします。
2. [Azure Active Directory] を選択します。
3. [エンタープライズアプリケーション] を選択します。
4. 「Amazon Managed Grafana SAML2.0」を検索して選択します。
5. アプリケーションを選択して [セットアップ] を選択します。
6. Azure Active Directory アプリケーション設定で、[ユーザーとグループ] を選択します。
7. 必要なユーザーとグループにアプリケーションを割り当てます。
8. [Single Sign-On] を選択します。
9. [次へ] を選択して、SAML 設定ページに移動します。
10. SAML の設定内容を指定します。
    - 識別子 (エンティティ ID)に、Amazon Managed Grafana ワークスペースからのサービスプロバイダー識別子 URL を貼り付けます。
    - 返信 URL (Assertion Consumer Service URL) に、Amazon Managed Grafana ワークスペースからのサービスプロバイダーの返信 URL を貼り付けます。
    - [Sign Assertion] が選択され、[Encrypt Assertion] が選択されていないことを確認します。
11. ユーザー属性とクレームセクションで、これらの属性がマッピングされていることを確認します。大文字と小文字が区別されます。

- mail は user.userprincipalname に設定されます。
- [表示名] は user.displayname に設定されます。
- 一意のユーザー識別子は user.userprincipalname に設定されます。
- 渡したい他の属性を追加します。アサーションマッピングで Amazon Managed Grafana に渡すことができる属性の詳細については、「アサーションマッピング」を参照してください。

12. Amazon Managed Grafana ワークスペース設定で使用する SAML メタデータ URL をコピーします。

ステップ 2: Amazon Managed Grafana での作業手順

Amazon Managed Grafana コンソールで以下手順を実行します。

Azure Active Directory を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択します。
3. [すべての WorkSpaces] を選択します。
4. ワークスペースの名前を選択します。
5. [認証]] タブで、[SAML 設定のセットアップ]] を選択します。
6. [Import the metadata (メタデータのインポート)] で [Upload or copy/paste (アップロードまたはコピー/貼り付け) を選択し、前のセクションでSAML Metadata URL (SAML メタデータ URL) からコピーした Azure Active Directory URL を貼り付けます。
7. [アサーションマッピング] で、次を行います:

   - I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) が選択されていないことを確認してください。

   > **Note**
   >
   > I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana ワークスペースコンソールを使用してワークスペースを管理することができなくなります。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

- [アサーション属性ロール] を選択した属性名に設定します。
- [管理者ロールの値] を、管理者ユーザーのロールに対応する値に設定します。
- (任意) Azure Active Directory アプリケーションでデフォルトの属性を変更した場合、[追加設定 - オプション] を展開し、新しい属性名を設定します。

  デフォルトでは、Azure の [表示名] 属性が [名前] 属性として渡され、Ping Identity の mail (メール)属性が email (eメール) 属性と[ログイン] 属性の両方に渡されます。

8. [SAML 設定の保存] を選択します。

CyberArk を使用するように Amazon Managed Grafana を設定する

次の手順で、CyberArk を ID プロバイダーとして使用するように Amazon Managed Grafana を設定します。この手順は、Amazon Managed Grafana ワークスペースを既に作成していることを前提としています。ワークスペースの ID、URL、および Region の情報をお手元に用意してください。

ステップ 1: CyberArk での作業手順

CyberArk で以下のステップを完了します。

CyberArk を Amazon Managed Grafana の ID プロバイダーとして設定する

1. CyberArk Identity 管理ポータルにサインインします。
2. [アプリ]、[Web アプリ] を選択します。
3. [Web アプリの追加] を選択します。
4. [Amazon Managed Grafana for SAML2.0] を検索して [追加]] を選択します。
5. CyberArk アプリケーション設定で、Trust (信頼) セクションに移動します。
6. Identity Provider Configurationで、[メタデータ] を選択します。
7. [Copy URL] (URL をコピー) を選択して URL を保存します (後で使用します)。
8. [サービスプロバイダーの設定]] で、[手動設定]] を選択します。
9. SAML の設定内容を指定します。

   - SP Entity IDに、Amazon Managed Grafana ワークスペースからのサービスプロバイダー識別子 URL を貼り付けます。
   - アサーション カスタマー サービス (ACS) URL に、Amazon Managed Grafana ワークスペースからのサービスプロバイダーの返信 URL を貼り付けます。

- [Sign Response Assertion] (サイン レスポンス アサーション) を [Assertion] (アサーション) に設定します。
- [NameID の形式] が [emailAddress] であることを確認します。

10. [保存] を選択します。
11. [SAML レスポンス] セクションで、Amazon Managed Grafana 属性が [アプリケーション名] にあり、CyberArk 属性が[属性値] にあることを確認します。次に、以下の属性がマッピングされていることを確認します。大文字と小文字が区別されます。

    - [表示名] は LoginUser.DisplayName に設定されます。
    - mail は LoginUser.Email に設定されます。
    - 渡したい他の属性を追加します。アサーションマッピングで Amazon Managed Grafana に渡すことができる属性の詳細については、「アサーションマッピング」を参照してください。

12. [保存] を選択します。
13. [権限] セクションで、このアプリケーションを割り当てるユーザーとグループを選択し、[保存] を選択します。

ステップ 2: Amazon Managed Grafana での作業手順

Amazon Managed Grafana コンソールで以下手順を実行します。

CyberArk を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択します。
3. [すべての WorkSpaces] を選択します。
4. ワークスペースの名前を選択します。
5. [認証]] タブで、[SAML 設定のセットアップ]] を選択します。
6. [Import the metadata (メタデータのインポート)] で[Upload or copy/paste (アップロードまたはコピー/貼り付け) を選択し、前の手順でコピーした CyberArk の URL を貼り付けます。
7. [アサーションマッピング] で、次を行います:

    - I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) が選択されていないことを確認してください。

> **Note**
>
> I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana ワークスペースコンソールを使用してワークスペースを管理することができなくなります。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

- [アサーション属性ロール] を選択した属性名に設定します。
- [管理者ロールの値] を、管理者ユーザーのロールに対応する値に設定します。
- (任意) CyberArk アプリケーションでデフォルトの属性を変更した場合、[追加設定 - オプション] を展開し、新しい属性名を設定します。

  デフォルトでは、CyberA の [表示名]属性が [名前] 属性に渡され、CyberArk の [メール] 属性が [eメール] 属性と[ログイン]属性の両方に渡されます。

8. [SAML 設定の保存] を選択します。

Okta を使用するように Amazon Managed Grafana を設定する

次の手順で、Okta を ID プロバイダーとして使用するように Amazon Managed Grafana を設定します。この手順は、Amazon Managed Grafana ワークスペースを既に作成していることを前提としています。ワークスペースの ID、URL、および Region の情報をお手元に用意してください。

ステップ 1: Okta での作業手順

Okta で以下の手順を実行します。

Okta を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Okta コンソールに管理者としてサインインします。
2. 左側のパネルで、[アプリケーション]、[アプリケーション] の順に選択します。
3. [アプリカタログを参照] を選択して [Amazon Managed Grafana] を検索します。
4. [Amazon Managed Grafana] を選択して [追加]、[完了] の順に選択します。
5. アプリケーションを選択してセットアップを開始します。
6. [Sign On] (サインオン) タブで、[編集] を選択します。

7. [詳細サインオン設定] の [名前空間] フィールドに Amazon Managed Grafana ワークスペース ID を、[リージョン] フィールドにリージョンをそれぞれ入力します。Amazon Managed Grafana ワークスペース ID とリージョンは Amazon Managed Grafana ワークスペース URL に、***workspace-id***.grafana-workspace.***Region***.amazonaws.com の形式で記載されています。
8. [保存] を選択します。
9. SAML 2.0 で、Identity Provider metadata (ID プロバイダーのメタデータ) の URL をコピーします。この URL は後で Amazon Managed Grafana コンソールで使用します。
10. [割り当て] タブで、Amazon Managed Grafana を使用する[ユーザー]と[グループ]を選択します。

ステップ 2: Amazon Managed Grafana での作業手順

Amazon Managed Grafana コンソールで以下手順を実行します。

Okta を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択します。
3. [すべての WorkSpaces] を選択します。
4. ワークスペースの名前を選択します。
5. [認証] タブで、[セットアップの完了] を選択します。
6. [メタデータのインポート] で [Upload or copy/paste] (アップロードまたはコピー/貼り付け) を選択し、前の手順でコピーした Okta の URL を貼り付けます。
7. [アサーションマッピング] で、次を行います:

   - I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) が選択されていないことを確認してください。

> **Note**
>
> I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana ワークスペースコンソールを使用してワークスペースを管理することができなくなります。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

- [アサーション属性ロール] を選択した属性名に設定します。
- [管理者ロールの値] を、管理者ユーザーのロールに対応する値に設定します。
- (任意) Okta アプリケーションでデフォルトの属性を変更した場合、[追加設定 - オプション] を展開し、新しい属性名を設定します。

  デフォルトでは、Okta の [表示名] 属性が [名前] 属性に渡され、Okta の [メール] 属性が [e メール] 属性と[ログイン] 属性の両方に渡されます。

8. [SAML 設定の保存] を選択します。

OneLogin を使用するように Amazon Managed Grafana を設定する

次の手順で、OneLogin を ID プロバイダーとして使用するように Amazon Managed Grafana を設定します。この手順は、Amazon Managed Grafana ワークスペースを既に作成していることを前提としています。ワークスペースの ID、URL、および Region の情報をお手元に用意してください。

ステップ 1: OneLogin での作業手順

OneLogin で次のステップを完了します。

OneLogin を Amazon Managed Grafana の ID プロバイダーとして設定する

1. ステップ 1: 管理者として OneLogin ポータルにサインインする。
2. [アプリケーション]、[アプリケーション]、[アプリの追加] の順に選択します。
3. [Amazon Managed Service for Grafana] を検索します。
4. 任意の表示名を設定して [保存] を選択します。
5. [設定] に移動し、[名前空間] に Amazon Managed Grafana ワークスペース ID を入力し、Amazon Managed Grafana ワークスペースのリージョンを入力します。
6. [設定] タブで、Amazon Managed Grafana ワークスペースの URL を入力します。
7. [adminRole] パラメータはデフォルトの [No Default] のままで構いません。必要に応じて[ルール] タブから Amazon Managed Grafana に対応する値を設定できます。この例では、Amazon Managed Grafana の [アサーション属性ロール] が [adminRole] に設定され、値は真になります。この値は、テナント内の任意の属性に設定できます。+ をクリックして、組織の要件を満たすようにパラメータを追加および設定します。
8. [ルール] タブで [ルールの追加] を選択してルールに名前を付けます。[条件] フィールド (if 文) にメール ([email address] を含む) を追加します。[アクション] フィールド (then 文) で [Amazon Managed Service で AdminRole を設定] を選択し、[Set adminRole] ドロップダウンで [マクロ]

を選択し、値に [真] を設定します。異なるルールを選択して、さまざまなユースケースに対応できます。

9. [保存] を選択します。[その他のアクション] へ移動して、[エンタイトルメントマッピングの再適用] を選択します。ルールを作成または更新した都度、マッピングを再適用する必要があります。
10. 発行者 URL を控えます。この URL は Amazon Managed Grafana コンソールで行う設定 (後述) で使用します。次に、[保存] を選択します。
11. [アクセス] タブを選択して、Amazon Managed Grafana にアクセスする OneLogin のロールを割り当て、アプリケーションセキュリティポリシーを選択します。

ステップ 2: Amazon Managed Grafana での作業手順

Amazon Managed Grafana コンソールで以下手順を実行します。

OneLogin を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択します。
3. [すべての WorkSpaces] を選択します。
4. ワークスペースの名前を選択します。
5. [認証]] タブで、[SAML 設定のセットアップ]] を選択します。
6. [メタデータのインポート]] で [アップロードまたはコピー/貼り付け] を選択し、前の手順で OneLogin コンソールからコピーした OneLogin の 発行者 URL を貼り付けます。
7. [アサーションマッピング] で、次を行います:

   - I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) が選択されていないことを確認してください。

> **Note**
>
> I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana ワークスペースコンソールを使用してワークスペースを管理することができなくなります。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

- [アサーション属性ロール] を選択した属性名に設定します。OneLogin のデフォルト値は [adminRole]です。
- [管理者ロールの値] を、管理者ユーザーのロールに対応する値に設定します。
- (任意) OneLogin アプリケーションでデフォルトの属性を変更した場合、[追加設定 - オプション] を展開し、新しい属性名を設定します。

  デフォルトでは、OneLogin の [表示名] 属性が [名前] 属性に渡され、OneLogin の [メール] 属性が [eメール] 属性と[ログイン] 属性の両方に渡されます。

8. [SAML 設定の保存] を選択します。

Ping Identity を使用するように Amazon Managed Grafana を設定する

次の手順で、Ping Identity を ID プロバイダーとして使用するように Amazon Managed Grafana を設定します。この手順は、Amazon Managed Grafana ワークスペースを既に作成していることを前提としています。ワークスペースの ID、URL、および Region の情報をお手元に用意してください。

ステップ 1: Ping Identity での作業手順

Ping Identity で以下手順を実行します。

Ping Identity を Amazon Managed Grafana の ID プロバイダーとして設定する

1. 管理者として Ping Identity コンソールにサインインします。
2. [Applications] (アプリケーション) を選択します。
3. [アプリケーションの追加]、[アプリケーションカタログの検索] の順に選択します。
4. [Amazon Managed Grafana for SAML] アプリケーションを検索して選択し、[セットアップ] を選択します。
5. Ping Identity アプリケーションで、[次へ] を選択して SAML 設定ページに移動します。続いて以下の SAML 設定を行います。

   - アサーション カスタマー サービス に、Amazon Managed Grafana ワークスペースからのサービスプロバイダーの返信 URL を貼り付けます。
   - Entity IDに、Amazon Managed Grafana ワークスペースからのサービスプロバイダー識別子を貼り付けます。
   - [Sign Assertion] が選択され、[Encrypt Assertion] が選択されていないことを確認します。

6. [Continue to Next Step] (次のステップに進む) を選択します。

7. [SSO 属性マッピング]で、Amazon Managed Grafana 属性が [アプリケーション属性] に、Ping Identity 属性が [Identity Bridge 属性] に設定されていることを確認します。続いて以下の設定を行います。

   - [mail] は [Email (Work)] である必要があります。
   - [displayName] は [表示名] である必要があります。
   - [SAML_SUBJECT] は [Email (Work)] である必要があります。次に、この属性で、[アドバンスト] を選択し、[SP に送信する名前 ID 形式] を urn:oasis:names:tc:SAML:2.0:nameid-format:transient に設定し、[保存] を選択します。
   - 渡したい他の属性を追加します。
   - 渡したい他の属性を追加します。アサーションマッピングで Amazon Managed Grafana に渡すことができる属性の詳細については、「アサーションマッピング」を参照してください。

8. [Continue to Next Step] (次のステップに進む) を選択します。
9. [グループアクセス] で、このアプリケーションを割り当てるグループを選択します。
10. [Continue to Next Step] (次のステップに進む) を選択します。
11. `https://admin- api.pingone.com/latest/metadata/` で始まる SAML メタデータ URL をコピーします。これは、後の設定で使用します。
12. [Finish] を選択してください。

ステップ 2: Amazon Managed Grafana での作業手順

Amazon Managed Grafana コンソールで以下手順を実行します。

Ping Identity を Amazon Managed Grafana の ID プロバイダーとして設定する

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択します。
3. [すべての WorkSpaces] を選択します。
4. ワークスペースの名前を選択します。
5. [認証]] タブで、[SAML 設定のセットアップ]] を選択します。
6. [Import the metadata (メタデータのインポート)] で [Upload or copy/paste (アップロードまたはコピー/貼り付け) を選択し、前の手順でコピーした Ping の URL を貼り付けます。
7. [アサーションマッピング] で、次を行います:

- I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) が選択されていないことを確認してください。

> **Note**
>
> I want to opt-out of assigning admins to my workspace (ワークスペースへの管理者の割り当てをオプトアウトする) を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana ワークスペースコンソールを使用してワークスペースを管理することができなくなります。ワークスペースに対する管理変更は、Grafana API を使用してのみ実行できます。

- [アサーション属性ロール] を選択した属性名に設定します。
- [管理者ロールの値] を、管理者ユーザーのロールに対応する値に設定します。
- (任意) Ping Identity アプリケーションでデフォルトの属性を変更した場合、[追加設定 - オプション] を展開し、新しい属性名を設定します。

  デフォルトでは、Ping Identity の [表示名] 属性が [名前] 属性に渡され、Ping Identity の [メール] 属性が [eメール] 属性と[ログイン] 属性の両方に渡されます。

8. [SAML 設定の保存] を選択します。

Amazon Managed Grafana ワークスペース AWS IAM Identity Center で を使用する

Amazon Managed Grafana は と統合 AWS IAM Identity Center して、ワークフォースに ID フェデレーションを提供します。Amazon Managed Grafana と IAM Identity Center を使用した場合、ユーザーはまず、自社のディレクトリにリダイレクトされ、既存の認証情報でサインインします。その後、Amazon Managed Grafana ワークスペースにシームレスにサインインされます。これにより、パスワードポリシーや 2 要素認証などのセキュリティ設定が適用されます。IAM Identity Center を使用した場合も、既存の IAM 設定には影響しません。

既存のユーザーディレクトリがない場合、またはフェデレーションを希望しない方のために、IAM Identity Center は Amazon Managed Grafana のユーザーとグループを作成するために使用できる統合ユーザーディレクトリを提供しています。Amazon Managed Grafana では、IAM ユーザーとロールを使用して Amazon Managed Grafana ワークスペース内のアクセス権限を割り当てることはできません。

IAM Identity Center の詳細については、「とは AWS IAM Identity Center」を参照してください。IAM Identity Center を始めるための詳細については、「開始方法」を参照してください。

IAM Identity Center を使用するには、アカウントで も AWS Organizations アクティブ化されている必要があります。必要に応じて、Amazon Managed Grafana は、IAM Identity Center を使用するように設定された最初のワークスペースを作成するときに Organizations をアクティブ化することもできます。

IAM Identity Center を使用するために必要なアクセス権限

このセクションでは、Amazon Managed Grafana で IAM Identity Center を使用するために必要なポリシーについて説明します。Amazon Managed Grafana を管理するために必要なポリシーは、 AWS アカウントが組織に参加しているかどうかによって異なります。

AWS Organizations アカウントで Grafana 管理者を作成します。

組織内で Amazon Managed Grafana ワークスペースを作成および管理するためのアクセス許可を付与し、 などの依存関係を許可するには AWS IAM Identity Center、ロールに次のポリシーを割り当てます。

- AWSGrafanaAccountAdministrator IAM ポリシー: Amazon Managed Grafana ワークスペースの管理を許可します。
- AWSSSODirectoryAdministrator: Amazon Managed Grafana ワークスペースの設定時に IAM Identity Center の使用を許可します。
- AWSSSOMasterAccountAdministrator IAM ポリシー: 組織全体の Amazon Managed Grafana ワークスペースの作成と管理を許可する場合、これを付与します。AWSSSOMemberAccountAdministrator IAM ポリシー: 組織の単一のメンバーアカウント内のワークスペースの作成と管理を許可する場合、これを付与します。
- AWSMarketplaceManageSubscriptions IAM ポリシー (または同等のアクセス許可): Amazon Managed Grafana ワークスペースを Grafana エンタープライズにアップグレードすることを許可します (任意)。

Amazon Managed Grafana ワークスペースの作成時にサービス管理のアクセス許可を使用する場合、ワークスペースを作成するロールに `iam:CreateRole`、`iam:CreatePolicy`、および `iam:AttachRolePolicy` 権限も必要です。これらの権限は、 AWS CloudFormation StackSets を使用して、組織のアカウント内のデータソースを読み取ることを許可するポリシーを展開するために必要です。

> **Important**
>
> ユーザーに、`iam:CreateRole`、`iam:CreatePolicy`、および `iam:AttachRolePolicy` アクセス許可を付与すると、そのユーザーには、 AWS アカウントへの完全な管理アクセス許可が与えられます。たとえば、これらのアクセス許可を持つユーザーは、すべてのリソースに対する完全なアクセス許可を持つポリシーを作成し、そのポリシーを任意のロールにアタッチできます。これらのアクセス許可を付与するユーザーには十分注意してください。

AWSGrafanaAccountAdministrator に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaAccountAdministrator」を参照してください。

Amazon Managed Grafana ワークスペースとユーザーを単一のスタンドアロンアカウントで作成および管理する

スタンドアロン AWS アカウントは、組織のメンバーではないアカウントです。詳細については AWS Organizations、「 とは」を参照してください AWS Organizations。

スタンドアロンアカウントで Amazon Managed Grafana ワークスペースとユーザーを作成および管理するための権限を付与するには、次の IAM ポリシーをロールに割り当てます。

- AWSGrafanaAccountAdministrator
- AWSSSOMasterAccountAdministrator
- AWSOrganizationsFullAccess
- AWSSSODirectoryAdministrator

> **Important**
>
> ロールに AWSOrganizationsFullAccess ポリシーを付与すると、そのロールに AWS アカウントへの完全な管理アクセス権が付与されます。これらのアクセス許可を付与するユーザーには十分注意してください。

AWSGrafanaAccountAdministrator に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaAccountAdministrator」を参照してください。

ワークスペースのバージョンを更新する

Amazon Managed Grafana コンソールで Amazon Managed Grafana ワークスペースを Grafana の新しいバージョンに更新する方法は 2 つあります。

> **Note**
>
> バージョンは Grafana の新しいバージョンにのみ更新が可能で、以前にリリースされたバージョンの Grafana にダウングレードすることはできません。
> Grafana のバージョンを更新しても、ワークスペースにインストールされているプラグインは更新されません。新しいバージョンの Grafana と互換性のないプラグインは、個別に更新する必要がある場合があります。プラグインの表示と管理の詳細については、「プラグインカタログによるプラグインの検索」を参照してください。各バージョンの変更点の一覧については、「Grafana のバージョン間の違い」を参照してください。

オプション 1 - ワークスペースのリストからバージョンを更新する

1. https://console.aws.amazon.com/grafana にアクセスして Amazon Managed Grafana コンソールを開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 更新を希望するワークスペースの詳細を含む行で、[バージョンの更新] を選択します。更新可能なワークスペースにのみ、このオプションが表示されます。

   > **Warning**
   >
   > 更新プロセスは元に戻すことができません。一時停止やキャンセルもできません。本番稼働用ワークスペースを更新する前に、本番稼働用以外の環境で新しいバージョンをテストすることをお勧めします。更新中はワークスペースに変更を加えることはできません。

5. バージョンの更新画面のドロップダウンからバージョン番号を選択し、[更新] をクリックして確認します。
6. [ワークスペース] タブで更新のステータスを定期的に確認します。この更新プロセスには最大 10 分かかることがあります。このプロセス中、ワークスペースは「読み取り専用」モードにな

ります。ワークスペースの更新が成功したか失敗したかを示すバナーが表示されます。更新に失敗した場合は、バナーに表示される内容に従って、再度実行してください。

オプション 2 - ワークスペースの概要ページからバージョンを更新する

1. https://console.aws.amazon.com/grafana にアクセスして Amazon Managed Grafana コンソールを開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 更新を希望するワークスペース名のハイパーリンクを選択します。更新可能なワークスペースにのみ、このオプションが表示されます。
5. 概要ブロックでバージョンの更新プロンプトを選択します。

> **Warning**
>
> 更新プロセスは元に戻すことができません。一時停止やキャンセルもできません。本番稼働用ワークスペースを更新する前に、本番稼働用以外の環境で新しいバージョンをテストすることをお勧めします。更新中はワークスペースに変更を加えることはできません。

6. バージョンの更新画面のドロップダウンからバージョン番号を選択し、[更新] をクリックして確認します。
7. [ワークスペース] タブで更新のステータスを定期的に確認します。この更新プロセスには最大 10 分かかることがあります。このプロセス中、ワークスペースは「読み取り専用」モードになります。ワークスペースの更新が成功したか失敗したかを示すバナーが表示されます。更新に失敗した場合は、バナーに表示される内容に従って、再度実行してください。

> **Note**
>
> バージョンの更新は、Amazon Managed Grafana API の UpdateWorkspaceConfiguration を使用して行うこともできます。

更新後のワークスペースで問題が発生した場合は、「ワークスペースの更新後のトラブルシューティング」を参照してください。

ワークスペースの更新後のトラブルシューティング

更新後も、ワークスペースは引き続き機能します。このセクションでは、更新後に発生する可能性のある問題を追跡するのに役立ちます。

- バージョンの違い。

  一部の機能はバージョンによって変更があります。

  - 機能に問題を与える可能性のある変更など、バージョン間の主要な変更のリストについては、「Grafana のバージョン間の違い」を参照してください。
  - バージョン 9 に固有の機能に関するドキュメントについては、「Grafana バージョン 9 での作業」を参照してください。バージョン 10 については、「Grafana バージョン 10 での作業」を参照してください。

- [PostgreSQL TLS の問題]

  [TLS/SSL モード]がバージョン 8 で `require` に設定されていて、ルート証明書のみを使用していた場合、更新後に PostgreSQL データソースで TLS または証明書の問題が発生する可能性があります。その場合、PostgreSQL データソースの TLS 設定を変更します (Grafana ワークスペースサイドメニューで、[設定]アイコンを選択し、[データソース]を選択します) 。

  - TLS/SSL モードを `verify-ca` に変更します。
  - TLS/SSL メソッドを `Certificate content` に設定します。
  - [ルート証明書]を PostgreSQL データベースサーバーのルート証明書に設定します。これは、証明書を入力する唯一のフィールドです。

Enterprise プラグインへのアクセス許可の管理

Amazon Managed Grafana コンソールを使用してワークスペースを管理し、Enterprise プラグインにアクセスできます。アップグレードすると、以下のリストを含む、サードパーティーのさまざまな独立系ソフトウェアベンダー (ISV) からのデータソースをサポートするエンタープライズプラグインにアクセスできます。

エンタープライズライセンスを使用すると、Grafana Labs のコンサルティングおよびサポートサービスにもアクセスできます。

Amazon Managed Grafana Enterprise プラグインで使用できる Enterprise データソースには、以下が含まれます。

- AppDynamics
- Databricks
- Datadog
- Dynatrace
- GitLab
- Honeycomb
- Jira
- MongoDB
- New Relic
- Oracle Database
- Salesforce
- SAP/HANA
- ServiceNow
- Snowflake
- Splunk
- Splunk Infrastructure Monitoring (旧 SignalFx)
- Wavefront

アップグレード時に利用可能な Enterprise データソースプラグインの詳細については、「Enterprise データソースに接続する」を参照してください。いつでも新しいプラグインを追加できます。完全かつ最新のリストについては、Amazon Managed Grafana ワークスペース内のプラグインカタログを使用できます。

ワークスペースを作成する場合は、デフォルトでは Enterprise プラグインにアクセスできませんが、いつでもアップグレードできます。Enterprise プラグインで複数の Amazon Managed Grafana ワークスペースを使用する場合は、それぞれをアップグレードする必要があります。

[Amazon Managed Grafana Enterprise の管理] ページから、アクセスの追加や削除など、Enterprise プラグインライセンスを管理できます。

Amazon Managed Grafana Enterprise プラグインへのアクセスを管理するプロセスは変更されました。以前に を使用したことがある場合は AWS Marketplace、 AWS Marketplace エンタープライズユーザー向けのよくある質問トピックに関心があるかもしれません。

トピック

- Amazon Managed Grafana Enterprise プラグインに対するアクセス権管理
- アカウントを Grafana Labs にリンクする
- AWS Marketplace エンタープライズユーザー向けのよくある質問

Amazon Managed Grafana Enterprise プラグインに対するアクセス権管理

エンタープライズプラグインに対するアクセス権を管理する方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana) を開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。

   ワークスペースのリストが表示されます。各ワークスペースの [エンタープライズライセンス] 列に、ワークスペースが持つライセンスの種類 (「ライセンスなし」または「エンタープライズプラグイン」ライセンスのいずれか) が表示されます。
4. ライセンスを管理するワークスペースの名前を選択します。選択すると、そのワークスペースの詳細ページが開きます。
5. サマリーの [エンタープライズライセンス] で、管理または Amazon Managed Grafana Enterprise にアップグレード を選択します (エンタープライズライセンスの現在の状況に応じて選択肢が 1 つのみ表示されます)。

   選択すると、Amazon Managed Grafana Enterprise の管理ページが開きます。2 つのオプションから選択できます。現在有効になっているオプションは (現在) が表示されます。

   - [なし] – Amazon Managed Grafana Enterprise ライセンスを削除するか、持っていない場合に選択します。(現在エンタープライズライセンスを保有している場合) ワークスペースにこのオプションを選択すると、保存時にエンタープライズプラグインへのアクセスが直ちに削除されます。
   - エンタープライズプラグイン – これを選択した場合、任意のエンタープライズプラグインをワークスペースにインストールできるほか、Grafana Labs のコンサルティングおよびサポートサービスにアクセスできるようになります。ワークスペースにエンタープライズプラグインをインストールすると、追加のデータソース にアクセスできます。

     このオプションを初めて選択するときは、 AWS アカウント を Grafana Labs のトークンにリンクする必要があり、リンクするように求められます。詳細については、次のセクション「アカウントを Grafana Labs にリンクする」を参照してください。

Amazon Managed Grafana Enterprise プラグインへのアクセスには、Amazon Managed Grafana の料金に加えて、ユーザー料金が含まれます。料金の詳細については、「Amazon Managed Grafana の料金」ページを参照してください。

6. 選択後、[保存] を選択して続行します。

アカウントを Grafana Labs にリンクする

Amazon Managed Grafana Enterprise プラグインにアップグレードされたワークスペースは、Grafana Labs のサポートとコンサルティングを利用することができます。この機能にアクセスするには、 を Grafana Labs アカウントトークンにリンク AWS アカウント する必要があります。エンタープライズライセンスにアップグレード AWS するときに、新規または既存の Grafana Labs アカウントを に登録します。 ???

> **Note**
>
> Grafana Labs アカウントトークンは、リージョンごとに 1 回に限り登録が求められます。アカウントが以前にリンクされている場合 (例えば、リージョン内の別のワークスペースをアップグレードしてエンタープライズプラグインにアクセスする場合)、再度リンクを求められることはありません。

リンクは、Amazon Managed Grafana で使用される Grafana Labs アカウントからトークンを取得してアカウントを登録することで構成されます。Grafana Labs に新しいアカウントを作成するか、既存のアカウントを使用することができます。

後で使用するために、Grafana Labs トークンをコピーして安全で便利な場所に保存することをお勧めします。

Grafana Labs アカウントをリンクする方法

1. 「Amazon Managed Grafana Enterprise プラグインに対するアクセス権管理」の手順に従って、エンタープライズプラグインにアクセスしてアカウントをアップグレードします。アップグレードの過程で、トークンを追加してアカウントをリンクするように求められます。
2. すでにトークンをお持ちの場合は、それを直接入力できます。トークンをお持ちでない場合は、「トークンの取得」を選択します。選択すると、新しいブラウザタブで Grafana Labs のウェブサイトが開きます。

Grafana Labs のウェブサイトから、Grafana Labs アカウントにサインイン (または新しいアカウントを作成) し、トークンを取得します。

3. トークンをコピーしたら、Amazon Managed Grafana ブラウザタブまたはウィンドウに戻り、Grafana Labs トークンセクションにトークンを入力します。
4. [保存] を選択してアップグレードを完了します。

トークンを他のワークスペースで再利用する

以前に Grafana Labs アカウントを登録していて、Grafana Labs トークンを求められた場合 (別のリージョンのワークスペースをアップグレードする場合など)、その都度同じトークンを使用して登録できるため、新しい Grafana Labs アカウントを作成する必要はありません。トークンを保存していない場合は、次のいずれかの方法でトークンを取得できます。

- https://grafana.com/partners/amg/support にアクセスし、My Account を選択して、Grafana Labs アカウントでトークンを検索することでトークンを取得できます。
- DescribeWorkspaceを使用してトークンを取得することで、既既にリンクされている既存のワークスペースからトークンを取得できます。
- いずれの方法でもトークンが取得できない場合、Grafana Labs サポート までお問い合わせください。

AWS Marketplace エンタープライズユーザー向けのよくある質問

以前は、 を通じて Grafana Enterprise のライセンスを購入していたかもしれません AWS Marketplace。で新しいライセンスを購入することはできなくなり AWS Marketplace、以前に で購入したライセンスを更新することもできなくなります AWS Marketplace。 AWS Marketplace ライセンスの状態によっては、次のよくある質問が役立つ場合があります。

AWS Marketplaceから 30 日間の無料トライアルに登録しましたが、まだワークスペースに関連付けていません。今から適用することはできますか？

いいえ。無料トライアルは Amazon Managed Grafana ではサポートされなくなりました。

AWS Marketplaceから登録した 30 日間の無料トライアルが、すでにワークスペースに関連付けられています。トライアルはどうなりますか？

無料トライアルは、有効期限が切れるまで継続されます。エンタープライズプラグインを使用するためにアップグレードしたい場合、前のセクションで説明したように、Amazon Managed Grafana コンソールを使用してアップグレードできます。

まだ有効期限が切れていない AWS Marketplace 有料ライセンスがありますが、Amazon Managed Grafana マネージドエンタープライズプラグインを使用したいと考えています。どうすればよいですか？

現在の AWS Marketplace ライセンスがある限り、そのライセンスはワークスペースにのみ関連付けることができます。Amazon Managed Grafana コンソールでアップグレードできるのは、 AWS Marketplace ライセンスの有効期限が切れた後のみです (または、ライセンスをキャンセルします AWS Marketplace) 。

詳細については、以下の質問と回答を参照してください。

から完全な Grafana Enterprise ライセンスを購入 AWS Marketplace し、1 つ以上のワークスペースに関連付けました。これらはどうなりますか。

ライセンスの有効期限が切れると (自動更新が有効になっていない限り、30 日後) 、ワークスペースで使用しているエンタープライズデータソースは機能しなくなります。Enterprise データソースを引き続き使用する場合、Amazon Managed Grafana コンソールから直接 Enterprise プラグインを使用するようにアップグレードすることができます。

ライセンスの有効期限が切れるとワークスペースがエンタープライズプラグインにアクセスできなくなり、ダウンタイムが発生する可能性があります。これを回避するにはどうすればよいですか？

ライセンスが期限切れになると、新しいエンタープライズプラグインライセンスに切り替える際に、いくつかのダウンタイムが発生します。ただし、これを最小限に抑えることができます。

> **Note**
>
> ダウンタイムを最小限に抑えるには、次の手順を正確に実行する必要があります。開始する前によくお読みになることをお勧めします。

新しい料金を取得するには、 AWS Marketplace ライセンスを引き続き使用するのではなく、Amazon Managed Grafana Enterprise プラグインにアップグレードすることをお勧めします。

ダウンタイムを最小限に抑えながら、 AWS Marketplace エンタープライズライセンスから Amazon Managed Grafana Enterprise プラグインに切り替える。

1. 準備として、まず Grafana Labs のウェブサイトにアクセスし、アカウントにサインインします (または新しいアカウントを作成します)。Grafana Labs トークンを取得します (プロセスの後半で使用します)。

   プロセスのこの部分の詳細については、「アカウントを Grafana Labs にリンクする」を参照してください。
2. AWS Marketplace コンソール にサインインし、左側のメニューから [サブスクリプションの管理] を選択します。
3. 切り替えるサブスクリプションを検索して [管理] を選択します。選択すると、サブスクリプションの詳細が表示されます。

   > **Note**
   >
   > 表示されたページには、サービス終了日が表示されます。現在のサブスクリプションを最大限に活用いただくため、ここですぐにキャンセルせずに、期限間近になってから進めていただいても構いません。

4. [Actions] (アクション)、[Cancel subscription] (サブスクリプションをキャンセル) の順に選択します。

   AWS Marketplaceのサブスクリプションがキャンセルされます。ただし、ライセンスは Amazon Managed Grafana によってその日の日末 (ワークスペースの現地時間) に自動的に削除されるため、それまでの間はエンタープライズデータソースを引き続き使用できます。

   でのサブスクリプションのキャンセルの詳細については AWS Marketplace、「 AWS Marketplace 購入者ガイド」の「製品サブスクリプションのキャンセル」を参照してください。
5. サブスクリプションがキャンセルされたら AWS Marketplace、Amazon Managed Grafana でキャンセルします。

   1. Amazon Managed Grafana コンソールにサインインします。

2. 左側のメニューから、[すべてのワークスペース] を選択します。
3. 切り替えるワークスペースの名前を選択します。
4. [エンタープライズライセンス] で、[管理] を選択します。
5. [なし] を選択し、 [保存] を選択します。これにより、Amazon Managed Grafana から AWS Marketplace ライセンスが削除されます。

   エンタープライズライセンスの削除後は、ワークスペースのエンタープライズプラグインにアクセスできなくなります。

6. Amazon Managed Grafana コンソールでアップグレードできるようになりました。最初のステップで作成した Grafana Labs トークンを使用して、Amazon Managed Grafana Enterprise プラグインに対するアクセス権管理 トピックの指示に従います。

> **Note**
>
> ライセンスを Amazon Managed Grafana でキャンセルした時点から、エンタープライズプラグインにアクセスできるようにアップグレードするまでの間、ワークスペースはエンタープライズデータソースにアクセスできなくなります。通常約 10～15 分で完了しますが、作業速度によっては時間がかかる場合があります。Grafana Labs トークンを準備しておくことで、この時間を最小限に抑えることができます。

自動更新の AWS Marketplace ライセンスがあります。これは継続されますか？

はい。 AWS Marketplace サブスクリプションは廃止され、手動で更新することはできませんが、自動更新を設定した場合は、無効にするまで続行されます。無効にした後は、これまでの回答の手順に従ってアップグレードできます。

新しい料金を取得するには、 AWS Marketplace ライセンスを引き続き使用するのではなく、Amazon Managed Grafana Enterprise プラグインにアップグレードすることをお勧めします。

ワークスペースにまだ関連付けていない AWS Marketplace ライセンスがありますが、使用できますか？

はい。その AWS Marketplace ライセンスを関連付けて、有効期限が切れるまで使用できます。有効期限は、自動更新を有効にしない限り、30 日以内に失効します。詳細については、これまでの質問と回答を参照してください。

Amazon Managed Grafana ワークスペース間でコンテンツを移行する

コンテンツ (データソース、ダッシュボード、フォルダ、アラートルールを含む) をあるワークスペースから別のワークスペースに移行したいと考える場合があります。例えば、オンプレミスの Grafana インスタンスから Amazon Managed Grafana ワークスペースに移行する際、既存のコンテンツを新しいワークスペースに移行したいと考える場合があります。

Amazon Managed Grafana ではワークスペース間の直接的なコンテンツの移行をサポートしていませんが、 AWS はワークスペースまたは Grafana インスタンス内で、エクスポートおよびインポート機能を備えたこのシナリオを処理できるオープンソースの移行ユーティリティを提供しています。このユーティリティは、Amazon Managed Grafana Migrator と呼ばれます。

詳細については、GitHub の「Amazon Managed Grafana Migrator」を参照してください。

Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する

Amazon Managed Grafana ワークスペースには、ID プロバイダー (IdP) または AWS IAM Identity Centerで設定されたユーザーを使用してアクセスします。これらのユーザー (またはユーザーが属するグループ) にワークスペースへのアクセス権限を付与する必要があります。付与する権限は、`User`、`Editor`、または `Admin` から選択します。

ユーザーまたはグループにアクセス権限を付与する

前提条件

- ユーザーまたはユーザーグループに Amazon Managed Grafana ワークスペースに対するアクセス権を付与するには、まずユーザーまたはグループが ID プロバイダー (IdP) または AWS IAM Identity Centerに設定されている必要があります。詳細については、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。
- ユーザーとグループのアクセスを管理するには、 AWS Identity and Access Management (IAM) ポリシー AWSGrafanaWorkspacePermissionManagementV2、または同等のアクセス許可を持つユーザーとしてサインインする必要があります。IAM Identity Center でユーザーを管理する場合、AWSSSOMemberAccountAdministrator および AWSSSODirectoryReadOnly の IAM ポリシー、または同等の権限も必要です。詳細については、「Amazon Managed Grafana に対するアクセス許可のユーザーへの割り当ておよび割り当て解除」を参照してください。

Amazon Managed Grafana コンソールを使用して Grafana ワークスペースに対するユーザーのアクセス権を管理する方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースの名前を選択します。
5. [認証] タブを選択します。
6. このワークスペースで IAM Identity Center を使用している場合は、[ユーザーとユーザーグループの設定] を選択し、次のいずれか、または複数の操作を行います。

   - ユーザーに Amazon Managed Grafana ワークスペースへのアクセス権を付与するには、該当するユーザーのチェックボックスを選択し、[ユーザーを割り当て]] を選択します。
   - ユーザーをワークスペースの `Admin` にするには、[管理者にする] を選択します。
   - ユーザーのワークスペースに対するアクセス権を削除するには、[ユーザーの割り当てを解除]] を選択します。
   - ユーザーのグループ (LDAP グループなど) を追加するには、[割り当てられたユーザーグループ]] タブを選択します。次に、以下のいずれかを行います。
     - グループのメンバー全員に Amazon Managed Grafana ワークスペースに対するアクセス権を付与するには、グループの横にあるチェックボックスをオンにしてから、[グループの割り当て] を選択します。
     - グループのメンバー全員にワークスペース内の `Admin` ロールを付与するには、[管理者にする] を選択します。
     - グループのメンバー全員のワークスペースに対するアクセス権を削除するには、[グループの割り当てを解除] を選択します。

> **Note**
>
> IAM Identity Center を使用してユーザーを管理する場合、新しいユーザーやグループのプロビジョニングのみ IAM Identity Center コンソールで行います。Grafana ワークスペースに対するアクセス権の付与または削除は、Amazon Managed Grafana コンソールまたは API を使用して行います。

IAM Identity Center と Amazon Managed Grafana の間で同期が取れなくなった場合、競合を解決するためのオプションが表示されます。詳細については、以下の [ユーザーおよびグループの設定時の権限不一致エラー] を参照してください。

7. このワークスペースで SAML を使用している場合、[SAML 設定] を選択し、次のいずれかを実行します。

   - [インポート方法] に、以下のいずれかを選択します。
     - [URL] を選択して、IdP メタデータの URL を入力します。
     - [アップロードまたはコピー/貼り付け] を選択します。メタデータをアップロードする場合は、[ファイルを選択] を選択してメタデータファイルを選択します。または、コピーアンドペーストを使用する場合は、メタデータを [メタデータをインポート] にコピーします。
   - [アサーション属性ロール] に、ロール情報を抽出する SAML アサーション属性の名前を入力します。
   - [管理者ロール値] に、Amazon Managed Grafana ワークスペースの `Admin` ロールをすべて付与する必要がある IdP のユーザーロールを入力、または [ワークスペースへの管理者の割り当てをオプトアウトする] を選択します。

     > **Note**
     >
     > [ワークスペースへの管理者の割り当てをオプトアウトする] を選択すると、データソース、ユーザー、ダッシュボードアクセス許可の管理などのタスクを含め、Amazon Managed Grafana コンソールを使用してワークスペースを管理することはできません。ワークスペースに対する管理変更は、Amazon Managed Grafana API を使用してのみ実行できます。

   - (任意) さらに SAML 設定を入力するには、[追加設定] を選択し、以下の 1 つ以上の操作を行い、[SAML 設定の保存]を選択します。このすべてのフィールドはオプションとなります。
     - [アサーション属性名] で、SAML ユーザーの完全な「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
     - [アサーション属性ログイン] で、SAML ユーザーのユーザーサインイン名に使用する SAML アサーション内の属性の名前を指定します。
     - [アサーション属性電子メール] で、SAML ユーザーの電子メール名として使用する SAML アサーション内の属性の名前を指定します。
     - [ログイン有効期間 (分単位)] で、それを過ぎると再度サインインする必要がある、SAML ユーザーのサインイン有効期間を指定しします。

- [アサーション属性組織] で、ユーザー組織の「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [アサーション属性グループ] で、ユーザーグループの「わかりやすい」名として使用する SAML アサーション内の属性の名前を指定します。
- [許可される組織] では、IdP 内の特定の組織のメンバーであるユーザーのみにユーザーアクセスを制限できます。許可する 1 つ以上の組織を、カンマで区切って入力します。
- [エディタロールの値] で、Amazon Managed Grafana ワークスペースで `Editor` ロールをすべて付与する必要がある IdP のユーザーロールを入力します。1 つ以上のロールを、カンマで区切って入力します。

8. または、ユーザーのグループ (LDAP グループなど) を追加するには、[ユーザーグループ] タブを選択します。次に、以下のいずれかを行います。

   - グループのメンバー全員に Amazon Managed Grafana ワークスペースに対するアクセス権を付与するには、グループの横にあるチェックボックスをオンにしてから、[グループの割り当て] を選択します。
   - グループのメンバー全員にワークスペース内の `Admin` ロールを付与するには、[管理者にする] を選択します。
   - グループのメンバー全員のワークスペースに対するアクセス権を削除するには、[グループの割り当てを解除] を選択します。

ユーザーおよびグループの設定時の権限不一致エラー

Amazon Managed Grafana コンソールでユーザーやグループを設定する際に、権限不一致エラーが発生することがあります。これは、Amazon Managed Grafana と IAM Identity Center が同期していないことを示します。この場合、Amazon Managed Grafana から警告と不一致を解決するための選択肢が表示されます。[解決]] を選択すると、Amazon Managed Grafana から同期されていないアクセス権限を持つユーザーが記載されたダイアログが表示されます。

IAM Identity Center から削除されたユーザーは、ダイアログで `Unknown user` として表示され、数値 ID が表示されます。これらのユーザーに対する不一致の修正方法は、[解決]] を選択してアクセス権を削除する以外にありません。

IAM Identity Center にまだ存在するが、以前に持っていたアクセス権を持つグループに属していないユーザーは、解決リストにそのユーザー名が表示されます。この問題を解決する方法は 2 つあります。[解決]] ダイアログを使用してアクセス権を削除または制限するか、前のセクションの手順に従ってアクセス権を付与することです。

アクセス許可の不一致に関するよくある質問

Amazon Managed Grafana コンソールの「ユーザーとグループの設定」セクションで、アクセス許可の不一致を示すエラーが表示される理由は何ですか？

このメッセージが表示される理由は、IAM Identity Center のユーザーとグループの関連付けと、ワークスペースの Amazon Managed Grafana のアクセス権限に不一致が見つかったためです。Grafana ワークスペースへのユーザーの追加または削除は、Amazon Managed Grafana コンソール ([ユーザーとグループの設定] タブ) または IAM Identity Center コンソール (アプリケーションの割り当てページ) から行うことができます。ただし、Grafana ユーザーの権限設定は、(Amazon Managed Grafana コンソールまたは API を使用して) ユーザーまたはグループに [閲覧者]、[編集者]、または [管理者] 権限を割り当てることによって、Amazon Managed Grafana からのみ行うことができます。ユーザーは、異なる権限を持つ複数のグループに属することができます。この場合、ユーザーの権限は、そのユーザーが属するすべてのグループと権限の中で最も高いアクセスレベルが適用されます。

不一致の記録が発生する原因は以下が考えられます:

- IAM Identity Center からユーザーまたはグループが削除されたが、Amazon Managed Grafana では削除されていない場合。この場合、Amazon Managed Grafana コンソールで [不明なユーザー] として表示されます。
- IAM Identity Center ([アプリケーションの割り当て] の下) でユーザーまたはグループの Grafana との関連付けが削除されたが、Amazon Managed Grafana では削除されていない場合。
- ユーザーの権限が以前、Grafana ワークスペースから直接更新された場合。Amazon Managed Grafana では Grafana ワークスペースからの更新はサポートされていません。

これらの不一致を回避するために、Amazon Managed Grafana コンソールまたは Amazon Managed Grafana API を使用して、ワークスペースのユーザーおよびグループの権限を管理することをお勧めします。

以前、Grafana ワークスペースから一部のチームメンバーのアクセスレベルを更新しましたが、現在、アクセスレベルが古いアクセスレベルに戻ってしまいました。このような現象が発生する理由と解決方法は？

これは、IAM Identity Center のユーザーとグループの関連付けと、ワークスペースの Amazon Managed Grafana の権限レコードの間で不一致が確認されたことが原因である可能性が高いと考えられます。チームメンバーに異なるアクセスレベルが割り当てられていて、お客様または Amazon

Managed Grafana の管理者が Amazon Managed Grafana コンソールから不一致を解決するために不一致レコードを削除した可能性があります。Amazon Managed Grafana コンソールまたは API から必要なアクセスレベルを再度割り当てることで、必要な権限を復元することができます。

> **Note**
>
> Grafana ワークスペースからのユーザーアクセス管理はサポートされていません。ユーザーやグループの権限は、Amazon Managed Grafana コンソールまたは API を使用して割り当ててください。

アクセスレベルが変更されることがありますがその理由は何ですか？ 例えば、以前は管理者権限がありましたが、現在は編集者権限になっています。

ワークスペースの管理者によって権限が変更された可能性があります。これは、IAM Identity Center のユーザーとグループの関連付けと、Amazon Managed Grafana の権限に不一致が生じた場合に起こることがあります。この場合、不一致を解消する過程で、より高い権限が削除された可能性があります。Amazon Managed Grafana コンソールから必要なアクセスレベルを再度割り当てるよう、管理者に依頼してください。

データソースおよび通知チャネルのアクセス権管理

Amazon Managed Grafana ワークスペースには、アラートのメトリクスと通知チャネルの AWS データソースにアクセスするためのアクセス許可が必要です。Amazon Managed Grafana コンソールを使用して、Amazon Managed Grafana ワークスペースで使用する AWS データソースと通知チャネルの AWS Identity and Access Management (IAM) ポリシーとアクセス許可を Amazon Managed Grafana に自動的に作成できます。

データソースおよび通知チャネルのアクセス権とポリシーの管理方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースの名前を選択します。
5. アクセス権限の管理をサービス管理とカスタマー管理の間で切り替えるには、IAM ロールの編集アイコンを選択して選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。

アクセス権限の管理をサービス管理からカスタマー管理に変更しても、現在のアカウントで Amazon Managed Grafana が作成したロールとポリシーは削除されません。ただし、組織単位でアクセス権限の管理にサービス管理を使用していた場合、組織内の他のアカウントのロールとポリシーは削除されます。

6. [データソース] タブを選択します。
7. アクセス権限の管理にサービス管理を使用している場合、IAM アクセス許可設定の横にある [編集] を選択して、サービス管理のアクセス権限を現在のアカウントのみに適用するか、組織全体に適用するかを変更することができます。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。

   データソースで、このワークスペースでクエリする AWS データソースを選択します。データソースを選択することで、Amazon Managed Grafana がこれらのソースからデータを読み取るのに必要な IAM ロールとアクセス許可を作成できるようになります。ただし、データソースは Grafana ワークスペースコンソールで追加する必要があります。

   通知チャネルとして使用できる AWS サービスを管理するには、通知チャネルを選択します。

   このワークスペースで使用する AWS 通知チャネルを選択します。通知チャネルを選択することで、Amazon Managed Grafana がこれらのサービスを利用するのに必要な IAM ロールとアクセス許可を作成できるようになります。ただし、通知チャネルは Grafana ワークスペースコンソールで追加する必要があります。

   > **Note**
   >
   > 通知の使用方法の詳細については、「アラート通知の管理」を参照してください。

を使用した Amazon Managed Grafana リソースの作成 AWS CloudFormation

Amazon Managed Grafana は と統合されています。これは AWS CloudFormation、 AWS リソースとインフラストラクチャの作成と管理に費やす時間を短縮できるように、リソースのモデル化とセットアップを支援するサービスです。必要なすべての AWS リソース (ワークスペースなど) を記述するテンプレートを作成すると、 はそれらのリソースを AWS CloudFormation プロビジョニングして設定します。

を使用すると AWS CloudFormation、テンプレートを再利用して Amazon Managed Grafana リソースを一貫して繰り返しセットアップできます。リソースを 1 回記述し、同じリソースを複数の AWS アカウント およびリージョンで何度もプロビジョニングします。

Amazon Managed Grafana と AWS CloudFormation テンプレート

Amazon Managed Grafana および関連サービスのリソースをプロビジョニングして設定するには、AWS CloudFormation テンプレートを理解しておく必要があります。テンプレートは、JSON や YAML でフォーマットされたテキストファイルです。これらのテンプレートは、 AWS CloudFormation スタックにプロビジョニングするリソースを記述します。JSON または YAML に慣れていない場合は、 AWS CloudFormation デザイナーを使用して AWS CloudFormation テンプレートの使用を開始できます。詳細については、「 AWS CloudFormation ユーザーガイド」のAWS CloudFormation 「デザイナーとは」を参照してください。

Amazon Managed Grafana は、 AWS CloudFormationでのワークスペースの作成をサポートしています。ワークスペースの JSON テンプレートと YAML テンプレートの例を含む詳細情報については、「AWS CloudFormation ユーザーガイド」の「Amazon Managed Grafana リソースタイプのリファレンス」を参照してください。

の詳細 AWS CloudFormation

詳細については AWS CloudFormation、以下のリソースを参照してください。

- AWS CloudFormation
- AWS CloudFormation ユーザーガイド
- AWS CloudFormation API リファレンス
- AWS CloudFormation コマンドラインインターフェイスユーザーガイド

Amazon Managed Grafana ワークスペースに対するネットワークからのアクセス制御を設定する

ユーザーとホストに対して Grafana ワークスペースへのアクセス権を設定することができます。

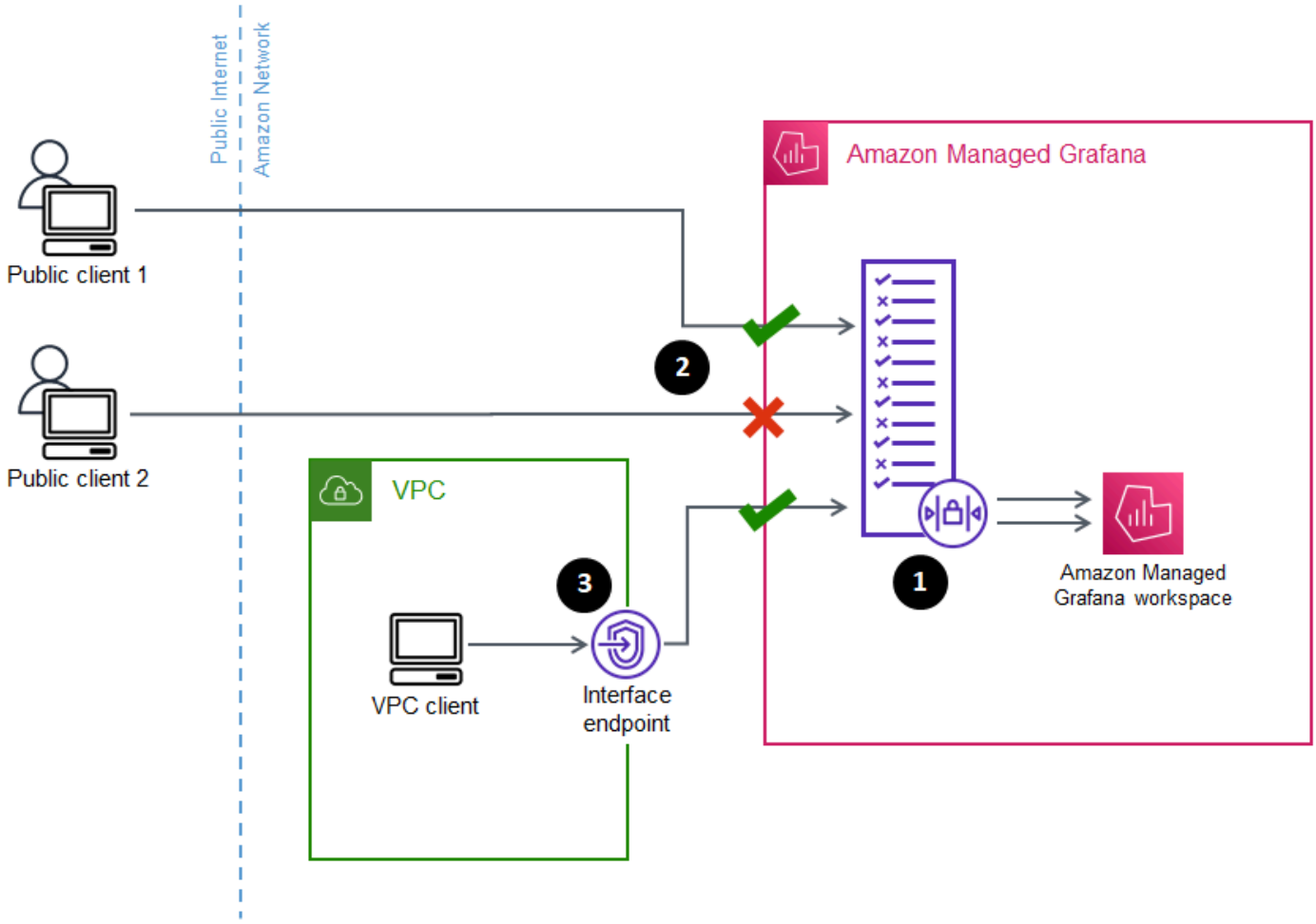
Grafana では、すべてのユーザーが認証された上で権限を与えられている必要があります。ただし、デフォルトでは、Amazon Managed Grafana ワークスペースはすべてのネットワークトラフィックに対して解放された状態です。ワークスペースに対するネットワークアクセス制御を設定して、ワークスペースに到達できるネットワークトラフィックを制御することができます。

ワークスペースに対するトラフィックは、2 つの方法で制御できます。

- IP アドレス (プレフィックスリスト) – ワークスペースへのアクセスを許可する IP 範囲を指定したマネージドプレフィックスリストを作成することができます。Amazon Managed Grafana では、ネットワークアクセス制御に使用できるのはパブリック IPv4 アドレスのみとなります。
- VPC エンドポイント – 特定のワークスペースにアクセスできる VPC エンドポイントのリストを作成して、そのワークスペースへのアクセスを許可することができます。

ネットワークアクセスコントロールを設定する際は、少なくとも 1 つのプレフィックスリストまたは VPC エンドポイントを設定する必要があります。

Amazon Managed Grafana は、プレフィックスリストと VPC エンドポイントを使用して、どのリクエストが Grafana ワークスペースに対する接続を許可されるかを決定します。以下の図に、このフィルタリングの様子を示します。



Amazon Managed Grafana ワークスペースに対するネットワークアクセスコントロール (1) の設定で、どのリクエストのアクセスを許可するかを指定します。ネットワークアクセスコントロールでは、IPアドレス (2) または使用されているインターフェイスエンドポイント (3) によってトラフィックを許可またはブロックできます。

次のセクションでは、ネットワークアクセスコントロールを設定する方法について説明します。

ネットワークアクセス制御の設定

既存のワークスペースにネットワークアクセスコントロールを追加するか、ワークスペースを最初に作成する際の作業の一環で設定します。

前提条件

ネットワークアクセスコントロールを設定するには、まずワークスペースのインターフェイス VPC エンドポイント、または許可する IP アドレスの IP プレフィックスリストを少なくとも 1 つ作成する必要があります。両方、または両方を複数作成することもできます。

- VPC エンドポイント – すべてのワークスペースへのアクセスを許可するインターフェイス VPC エンドポイントを作成できます。エンドポイントを作成した後は、許可する各エンドポイントの VPC エンドポイント ID が必要になります。VPC エンドポイント ID の形式は `vpce-1a2b3c4d` です。

  Grafana ワークスペースの VPC エンドポイントの作成については、「インターフェイス VPC エンドポイント」を参照してください。ワークスペース専用の VPC エンドポイントを作成するには、`com.amazonaws.region.grafana-workspace` エンドポイント名を使用します。

  ワークスペースへのアクセスを許可する VPC エンドポイントに対してエンドポイントのセキュリティグループを設定することで、より詳細なアクセス制限が可能になります。詳細については、「Amazon VPC ドキュメント」の「セキュリティグループの関連付け」と「セキュリティグループのルール」を参照してください。
- マネージドプレフィックスリスト (IP アドレス範囲の場合) – IP アドレスを許可するには、Amazon VPC で 1 つ以上のプレフィックスリストを作成し、許可する IP 範囲のリストを指定する必要があります。Amazon Managed Grafana で使用するプレフィックスリストにはいくつかの制限があります。
  - 各プレフィックスリストに含めることができる IP アドレス範囲の上限は最大 100 個までとなっています。
  - プライベート IP アドレス範囲 (例: `10.0.0.0/16`) は無視されます。プレフィックスリストにプライベート IP アドレス範囲を含めることはできますが、Amazon Managed Grafana はワークス

ペースへのトラフィックをフィルタリングするときにそれらを無視します。これらのホストがワークスペースにアクセスできるようにするには、ワークスペース用の VPC エンドポイントを作成してアクセスを許可する必要があります。

- Amazon Managed Grafana は、プレフィックスリストで IPv4 アドレスのみをサポートします (IPv6 はサポート外)。IPv6 アドレスは無視されます。

マネージドプレフィックスリストは、Amazon VPC コンソールを使用して作成します。プレフィックスリストを作成した後で、Amazon Managed Grafana で許可するリストごとにプレフィックスリスト ID が必要になります。プレフィックスリスト ID の形式は `pl-1a2b3c4d` です。

プレフィックスリストの作成に関する詳細については、「Amazon 仮想プライベートクラウドユーザーガイド」の「マネージドプレフィックスリストを使用して CIDR ブロックをグループ化する」を参照してください。

- Amazon Managed Grafana ワークスペースを設定または作成するには、特定のアクセス許可が必要です。たとえば、 AWS 管理ポリシー を使用できます`AWSGrafanaAccountAdministrator`。

ワークスペースへのアクセスを許可するプレフィックスリストまたは VPC エンドポイントの ID のリストを取得したら、ネットワークアクセスコントロール設定の作成準備完了です。

> **Note**
>
> ネットワークアクセスコントロールを有効にしても、設定にプレフィックスリストを追加しない場合、許可された VPC エンドポイント経由を除き、ワークスペースへのアクセスは許可されません。
> 同様に、ネットワークアクセスコントロールを有効にし、設定に VPC エンドポイントを追加しない場合、許可された IP アドレス経由を除き、ワークスペースへのアクセスは許可されません。
> ネットワークアクセスコントロール設定には、少なくとも 1 つのプレフィックスリストまたは VPC エンドポイントを登録する必要があり、どちらの登録もない場合、どこからもワークスペースへのアクセスはできなくなります。

ワークスペースに対するネットワークアクセスコントロールの設定方法

1. Amazon Managed Grafana コンソールを開きます。

2. 左のナビゲーションペインの [すべてのワークスペース] を選択します。
3. ネットワークアクセスコントロールを設定するワークスペースの名前を選択します。
4. [ネットワークアクセスコントロール] タブの [ネットワークアクセスコントロール]で、[制限付きアクセス] を選択してネットワークアクセスコントロールを設定します。

> **Note**
>
> この内容は、ワークスペースの作成時のオプションにも登場します。

5. ドロップダウンから、[プレフィックスリスト] または [VPC エンドポイント] から追加する方を選択します。
6. 追加する VPC エンドポイントまたはプレフィックスリスト ID を選択します (または、使用する ID を入力することもできます)。少なくとも 1 つを選択する必要があります。
7. エンドポイントまたはリストをさらに追加するには、追加するリソースごとに [新しいリソースを追加] を選択します。

> **Note**
>
> 最大 5 つのプレフィックスリストと 5 つの VPC エンドポイントを追加できます。

8. 設定を完了するには、[変更の保存] を選択します。

> **Warning**
>
> ワークスペースを使用する既存のユーザーがいる場合、そのユーザーの IP 範囲または VPC エンドポイントを設定に含めてください。含めない場合、そのユーザーは `403 Forbidden` エラーでアクセスできなくなります。ネットワークアクセスコントロールの設定または変更後に、既存のアクセスポイントをテストすることをお勧めします。

Amazon Managed Grafana から Amazon VPC のデータソースまたは通知チャネルに接続する

デフォルトでは、Amazon Managed Grafana ワークスペースからデータソースまたは通知チャネルへのトラフィックは、パブリックインターネット経由で流れます。このため、Amazon Managed

Grafana ワークスペースがアクセスできるのは、パブリックにアクセス可能なサービスに限定されます。

> **Note**
>
> プライベート VPC を設定しておらず、Amazon Managed Grafana がパブリックにアクセス可能なデータソースに接続している場合、 を介して同じリージョンの一部 AWS のサービスに接続します AWS PrivateLink。CloudWatch 、Amazon Managed Service for Prometheus、AWS X-Rayなどのサービスがこれに該当します。これらのサービスへのトラフィックは、パブリックインターネットに流れることはありません。

VPC内のプライベートなデータソースに接続する場合 (トラフィックを VPC にローカルに保持したい場合) は、Amazon Managed Grafana ワークスペースをこれらのデータソースが配置されている Amazon Virtual Private Cloud (Amazon VPC) に接続することができます。この VPC データソース接続を設定すると、すべてのトラフィックが VPC 経由で流れるようになります。

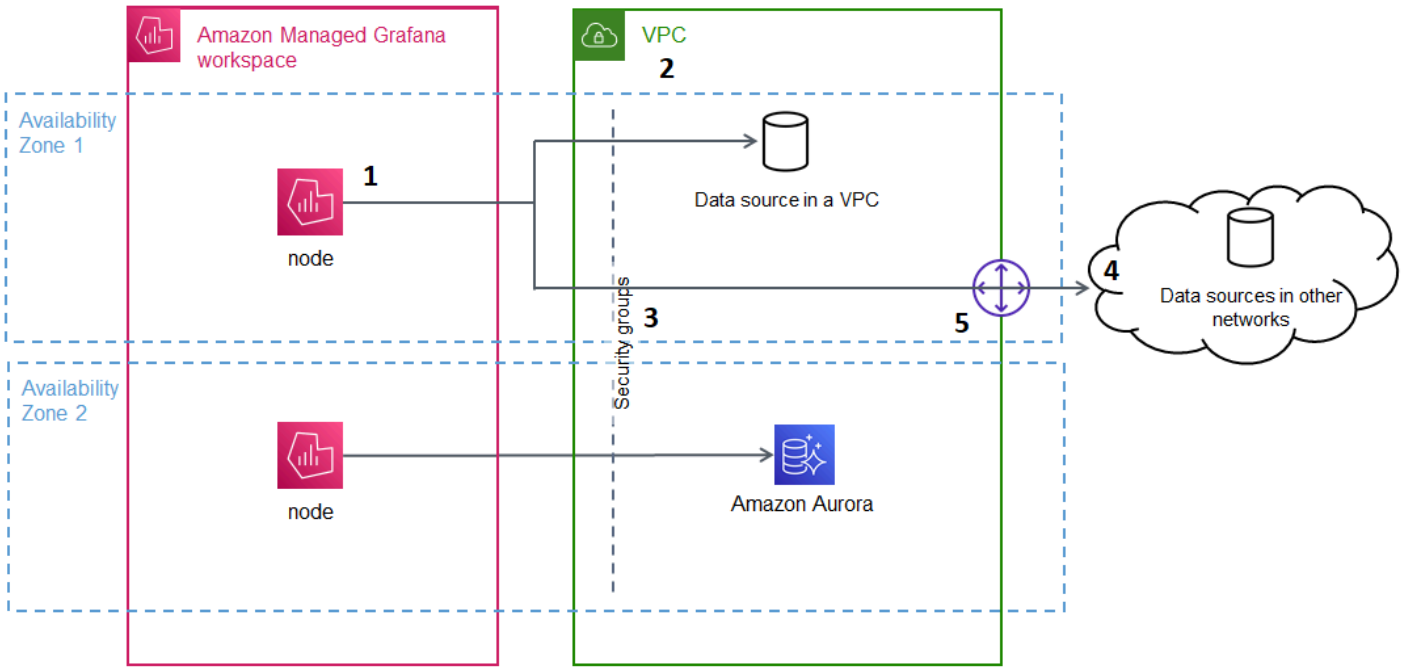
Virtual Private Cloud (VPC) は、 専用の仮想ネットワークです AWS アカウント。これは、他の仮想ネットワーク (他の VPC やパブリックインターネットを含む) から論理的に分離されます。Amazon VPC を使用して、 AWS クラウドで VPC を作成および管理します。Amazon VPC では、リソースの配置、接続、セキュリティなど、仮想ネットワーク環境を完全に制御できます。Amazon Managed Grafana データソースやその他のリソースを、VPC 内に作成できます。Amazon VPC の詳細については、「Amazon 仮想プライベートクラウドユーザーガイド」の「Amazon VPC とは?」を参照してください。

> **Note**
>
> Amazon Managed Grafana ワークスペースを VPC の外部、別のネットワーク、またはパブリックインターネットのデータに接続する場合は、他のネットワークへのルーティングを追加する必要があります。VPC を別のネットワークに接続する方法については、「Amazon 仮想プライベートクラウドユーザーガイド」の「VPC を他のネットワークに接続する」を参照してください。

VPC 接続の仕組み

Amazon VPC では、環境を自由に管理でき、アプリケーションを接続するためのパブリックおよびプライベートサブネットの作成や、サービスやリソースへのアクセス権を管理するためのセキュリティグループを作成することができます。

VPC 内のリソースで Amazon Managed Grafana を使用するには、Amazon Managed Grafana ワークスペースのその VPC への接続を作成する必要があります。接続を設定すると、Amazon Managed Grafana はワークスペースをその VPC の各アベイラビリティーゾーン内の各提供されたサブネットに接続し、Amazon Managed Grafana ワークスペースとの間で送受信されるすべてのトラフィックは VPC を経由します。以下の図に、この接続の論理構成を視覚的に示します。



Amazon Managed Grafana は、VPC (2) に接続するための接続 (1) をサブネット (Elastic Network Interface、ENI) を使用) ごとに作成します。Amazon Managed Grafana VPC 接続は、VPC と Amazon Managed Grafana ワークスペース間のトラフィックを制御する一連のセキュリティグループ (3) に関連付けられています。アラートの送信先やデータソースの接続など、すべてのトラフィックは設定された VPC 経由でルーティングされます。別の VPC やインターネットにあるデータソースやアラート先 (4) に接続するには、他のネットワークと VPC の間にゲートウェイ (5) を作成します。

VPC への接続を作成する

このセクションでは、既存の Amazon Managed Grafana ワークスペースから VPC に接続する手順について説明します。ワークスペースを作成する際にも、この手順と同じ内容を実行します。ワークスペース作成の詳細については、「Amazon Managed Grafana ワークスペースを作成する」を参照してください。

前提条件

以下は、既存の Amazon Managed Grafana ワークスペースから VPC への接続を確立するために必要な前提条件です。

- Amazon Managed Grafana ワークスペースを設定または作成するには、特定のアクセス許可が必要です。たとえば、 AWS 管理ポリシー を使用できますAWSGrafanaAccountAdministrator。
- アカウントには、それぞれに 1 つのプライベートサブネットが設定された少なくとも 2 つのアベイラビリティーゾーンを持つ VPC 設定が必要です。VPC のサブネットとセキュリティグループの情報を把握しておく必要があります。

> **Note**
>
> Local Zones と Wavelength Zones はサポートされていません。
> Tenancy が Dedicated に設定された VPC はサポートされません。

> **Important**
>
> Amazon Managed Grafana ワークスペースに接続されている各サブネットには、少なくとも 15 個の使用可能な IP アドレスが必要です。VPC サブネットの IP 使用率をモニタリングするようにアラームを設定することを強くお勧めします。サブネットで使用可能な IP アドレスの数が 15 を下回ると、次の問題が発生する可能性があります。
>
> - 追加の IP アドレスを解放するか、追加の IP アドレスでサブネットをアタッチするまで、ワークスペースの設定を変更できない
> - ワークスペースがセキュリティ更新プログラムやパッチを受信できない
> - まれに、ワークスペースの可用性が完全に失われ、アラートが機能しなくなり、ダッシュボードにアクセスできなくなることがあります。

- データソースが設定されている既存の Amazon Managed Grafana ワークスペースを接続する場合は、Amazon Managed Grafana を VPC に接続する前に、それらのデータソースに接続するように VPC を設定しておくことをお勧めします。これには、 を介して接続される CloudWatch などのサービスが含まれます AWS PrivateLink。これを行っていない場合、これらのデータソースへの接続が失われます。
- VPC に他のネットワークへの複数のゲートウェイがすでにある場合は、複数のゲートウェイ間で DNS 解決を設定する必要があります。詳細については、「ルート 53 リゾルバ」を参照してください。

既存の Amazon Managed Grafana ワークスペースから VPC に接続する

以下の手順では、既存の Amazon Managed Grafana ワークスペースに Amazon VPC データソース接続を追加する方法について説明します。

> **Note**
>
> Amazon VPC への接続を設定すると、IAM ロールが作成されます。Amazon Managed Grafana は VPC への接続を作成する際に、このロールを使用します。IAM ロールは、サービスリンクロールポリシー「`AmazonGrafanaServiceLinkedRolePolicy`」を使用します。サービスリンクロールの詳細については、「Amazon Managed Grafana 向けのサービスリンクロールのアクセス許可」を参照してください。

既存の Amazon Managed Grafana ワークスペースから VPC に接続する方法

1. Amazon Managed Grafana コンソールを開きます。
2. 左のナビゲーションペインの [すべてのワークスペース] を選択します。
3. VPC データソース接続を追加するワークスペースの名前を選択します。
4. [ネットワークアクセス設定] タブで、アウトバウンド VPC 接続の横にある [編集] を選択して VPC 接続を作成します。
5. 接続する VPC を選択します。
6. マッピングで、使用するアベイラビリティーゾーンを選択します。少なくとも 2 つ選択する必要があります。
7. 各アベイラビリティーゾーンで少なくとも 1 つのプライベートサブネットを選択します。サブネットは IPv4 形式である必要があります。

8. VPC に少なくとも 1 つのセキュリティグループを選択します。最大 5 つのセキュリティグループを指定できます。または、この接続に適用するセキュリティグループを作成することもできます。
9. 設定を完了するには、[変更の保存] を選択します。

VPC 接続の設定が完了したので、その VPC から Amazon Managed Grafana ワークスペースにアクセス可能な データソースに接続する を追加できます。

アウトバウンド VPC 設定内容の変更

設定内容を変更するには、ワークスペース設定の [ネットワークアクセス設定] タブに戻るか、UpdateWorkspace API を使用します。

> **Important**
>
> VPC 設定の管理は、Amazon Managed Grafana が行います。これらの VPC 設定の編集は、Amazon EC2 コンソールまたは API からは行わないでください。設定が同期されなくなります。

Amazon Managed Grafana での VPC の使用に関するトラブルシューティング

Amazon Managed Grafana での Amazon Virtual Private Cloud (Amazon VPC) の使用に関する一般的な質問への回答。

Amazon Managed Grafana で VPC を設定する必要があるのはどのような場合ですか？

プライベート VPC でのみ使用可能な (インターネットからアクセスできない) データソースに接続しようとする場合は、Amazon Managed Grafana で VPC を設定する必要があります。

データソースがインターネットに公開されている場合や、公開されているエンドポイントを持つ場合は、VPCの設定は不要です。

Amazon CloudWatch、Amazon Managed Service for Prometheus、または に接続する場合は AWS X-Ray、VPC を設定する必要はありません。これらのデータソースは、デフォルトで AWS PrivateLink を介して Amazon Managed Grafana に接続されています。

Amazon Managed Grafana ワークスペースで VPC を設定した後、既存のデータソースが接続できなくなったのはなぜですか？

既存のデータソースはおそらくインターネット経由でアクセス可能であり、Amazon VPC の設定ではパブリックネットワークへのアクセスが許可されていない可能性があります。Amazon Managed Grafana ワークスペースで VPC 接続を設定した後、すべてのトラフィックがその VPC を経由する必要があります。これには、その VPC 内でホストされるプライベートデータソース、別の VPC のデータソース、VPC で利用できない AWS 管理のサービス、インターネットに公開されているデータソースが含まれます。

この問題を解決するには、設定した VPC に他のデータソースを接続する必要があります。

- インターネットに接続するデータソースの場合、その VPC をインターネットに接続します。たとえば、NAT デバイスを使用してインターネットや他のネットワークに接続することができます（詳細は「Amazon 仮想プライベートクラウドユーザーガイド」を参照してください）。
- 他の VPC のデータソースの場合、2 つの VPC 間のピアリング接続を作成します。詳細については、「Amazon 仮想プライベートクラウドユーザーガイド」の「VPC ピアリング接続を使用して VPCを接続する」を参照してください。
- CloudWatch、X-Ray、Amazon AWS Managed Service for Prometheus など、VPC でアクセスできないマネージドサービスの場合は、VPC でそのサービスのインターフェイス VPC エンドポイントを作成する必要があります。詳細については、「 AWS PrivateLink ガイド」の「インターフェイス VPC エンドポイントを使用して AWS サービスにアクセスする」を参照してください。

専有テナンシーで VPC を使用できますか？

いいえ。 が VPCs `Dedicated`はサポートされていません。 `Tenancy`

Managed AWS Services (Amazon Managed Service for Prometheus、CloudWatch、X-Ray など) とプライベートデータソース (Amazon Redshift を含む) の両方を同じ Amazon Managed Grafana ワークスペースに接続できますか？

はい。プライベートデータソースと同じ VPC 内の AWS マネージドサービスへの接続を設定し (インターフェイス VPC エンドポイントや NAT ゲートウェイなどを使用)、Amazon Managed Grafana ワークスペースを同じ VPC に接続するように設定する必要があります。

Amazon Managed Grafana ワークスペースで VPC を設定した後にデータソースに接続しようとすると、**502 Bad Gateway Error** を取得するのはなぜですか？

以下はデータソース接続で 502 エラーが返される主な 3 つの原因です。

- セキュリティグループエラー — Amazon Managed Grafana で VPC を設定する際に選択したセキュリティグループには、データソースへの接続を許可するためのインバウンドおよびアウトバウンドのルールが適切に設定されている必要があります。

  この問題を解決するには、データソースセキュリティグループと Amazon Managed Grafana セキュリティグループの両方のルールでこの接続が許可されていることを確認してください。
- ユーザーアクセス許可エラー — 割り当てられたワークスペースユーザーにデータソースをクエリするための適切な権限がない可能性があります。

  この問題を解決するには、ユーザーにワークスペースを編集するために必要な IAM アクセス許可があり、ホスティングサービスからデータにアクセスしてクエリを実行するための適切なデータソースポリシーが設定されていることを確認します。アクセス許可は、https://console.aws.amazon.com/iam/ の AWS Identity and Access Management (IAM) コンソールで利用できます。
- 提供された接続の詳細が正しくありません — 提供された接続情報が間違っているため、Amazon Managed Grafana ワークスペースがデータソースに接続できません。

  この問題を解決するには、データソースの接続情報 (データソース認証とエンドポイント URL など) を確認し、接続を再試行してください。

同じ Amazon Managed Grafana ワークスペースから複数の VPC に接続できますか？

Amazon Managed Grafana ワークスペースには 1 つの VPC のみを設定できます。異なる VPC またはリージョン間でデータソースにアクセスするには、次の質問を参照してください。

異なる VPC のデータソースに接続するにはどうすればよいですか？ 別の AWS リージョン または にある VPC からデータソースに接続するにはどうすればよいですか AWS アカウント？

VPC ピアリングまたは を使用してクロスリージョンまたはクロスアカウント VPCs AWS Transit Gatewayを接続し、Amazon Managed Grafana ワークスペースと同じ AWS アカウント および リージョンにある VPC を接続します。Amazon Managed Grafana は、VPC 内の他の接続と同様に外部データソースに接続します。

> **Note**
>
> VPC ピアリングを選択できない場合、ユースケースをアカウントマネージャーと共有するか、aws-grafana-feedback@amazon.com に E メールを送信してください。

Amazon Managed Grafana ワークスペースが VPC に接続されている場合でも、他のパブリックデータソースに接続できますか？

あり。VPC とパブリックデータソースの両方のデータソースを同じ Amazon Managed Grafana ワークスペースに同時に接続することができます。パブリックデータソースには、NAT ゲートウェイ ゲートウェイ または他の VPC 接続 を介して VPC 接続を設定する必要があります。パブリックデータソースへのリクエストは VPC を通過するため、これらのリクエストに対するより強力な可視化と制御を得ることができます。

IP アドレスが不十分なために Amazon Managed Grafana ワークスペースを更新できない場合はどうすればよいですか？

Amazon Managed Grafana ワークスペース設定を変更するときに、次のエラーが発生することがあります。VPC 設定のすべてのサブネットには、少なくとも 15 個の使用可能な IP アドレスが必要です。

このエラーは、ワークスペースに接続されている 1 つ以上のサブネットが最小 IP 要件を満たしていない場合に表示されます。ワークスペースに接続されている各サブネットには、少なくとも 15 個の使用可能な IP アドレスが必要です。サブネットで使用可能な IP アドレスの数が 15 を下回ると、次の問題が発生する可能性があります。

- 追加の IP アドレスを解放するか、追加の IP アドレスを持つサブネットをアタッチするまで、ワークスペースの設定を変更できない
- ワークスペースがセキュリティ更新プログラムやパッチを受信できない
- まれに、ワークスペースの可用性が完全に低下し、アラートが機能せず、ダッシュボードにアクセスできなくなることがあります。

IP の枯渇を軽減

1. サブネットの使用可能な IP アドレスが 15 未満の場合は、インスタンスに関連付けられた IP アドレスを解放するか、未使用のネットワークインターフェイスを削除して IP 容量を解放します。

2. 既存のサブネットで IP アドレスを解放できない場合は、サブネットを少なくとも 15 個の使用可能な IP アドレスを持つサブネットに置き換える必要があります。Amazon Managed Grafana には専用サブネットを使用することをお勧めします。

サブネットを置き換える

1. Amazon Managed Grafana コンソールを開きます。
2. 左側のナビゲーションペインで、すべてのワークスペースを選択し、ワークスペースの名前を選択します。
3. ネットワークアクセスコントロールタブで、アウトバウンド VPC 接続の横にある編集を選択します。
4. マッピングで、IP アドレスが不十分なサブネットを含むアベイラビリティーゾーンを選択します。
5. ドロップダウンで、IP アドレスが不十分なサブネットの選択を解除し、使用可能な IP アドレスが 15 個以上のサブネットを選択します。必要に応じて、VPC に新しいサブネットを作成します。詳細については、「Amazon VPC ユーザーガイド」の「サブネットの作成」を参照してください。
6. 設定を完了するには、[変更の保存] を選択します。

VPC 接続を設定する前は、Grafana アラートが PagerDuty や Slack などのダウンストリームサービスに正常に送信されていました。VPC を設定した後、Grafana アラートがこれらの通知先に配信されないのはなぜですか？

Amazon Managed Grafana ワークスペースの VPC 接続を設定すると、ワークスペース内のデータソースへのすべてのトラフィックは、設定された VPC を経由します。VPC に、これらのアラート通知サービスに到達するルートがあることを確認してください。例えば、サードパーティーがホストするアラートの通知先をインターネットに接続する必要がある場合があります。データソースと同様に、外部接続先へのインターネット接続、 AWS Transit Gateway接続、またはその他の VPC 接続を設定します。

VPC は手動で編集できますか？ セキュリティグループまたはサブネットを変更すると、Amazon Managed Grafana ワークスペースが使用できなくなるのはなぜですか？

Amazon Managed Grafana の VPC 接続は、VPC と Amazon Managed Grafana ワークスペースの間の通信の制御にセキュリティグループとサブネットを使用しています。セキュリティグループまたはサブネットが Amazon Managed Grafana コンソール以外 (VPC コンソールなど) から変更ま

たは削除されると、Amazon Managed Grafana ワークスペースの VPC 接続はワークスペースのセキュリティの保護を停止し、ワークスペースにアクセスできなくなります。この問題を修正するには、Amazon Managed Grafana コンソールで Amazon Managed Grafana ワークスペースに設定されているセキュリティグループを更新してください。ワークスペースを表示するときは、[ネットワークアクセスコントロール] タブで [アウトバウンド VPC 接続] を選択し、VPC 接続に関連付けられたサブネットまたはセキュリティグループを変更します。

Amazon Managed Grafana ワークスペースを設定する

Amazon Managed Grafana の設定については、「Amazon Managed Grafana の認証とアクセス許可の設定」と、「Grafana ワークスペースの設定」に分けて説明することができます。このセクションでは、「Grafana ワークスペースの設定」について説明します。

「Amazon Managed Grafana の認証とアクセス許可の設定」の詳細については、以下のトピックを参照してください。

- Amazon Managed Grafana ワークスペースでユーザーを認証する
- Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する
- ユーザー、チーム、およびアクセス許可

Grafana ワークスペースの設定内容の変更は、ワークスペースのプロパティを表示する際にある [Workspace 設定オプション] タブの Amazon Managed Grafana 内で行うことができます。

Grafana インスタンスの設定を変更すると、新しい設定を反映するためにインスタンスが再起動することがあります。ブラウザで Grafana ワークスペースを表示している場合、設定の変更後、ブラウザページを更新する必要がある場合があります。

> **Note**
>
> ワークスペースを初めて作成するときにも、同じオプションを使用できます。

Amazon Managed Grafana コンソールを使用した Grafana ワークスペースの設定の変更方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。

4. 設定するワークスペースの名前を選択します。これにより、そのワークスペースの詳細が開きます。
5. [Workspace 設定オプション] タブを選択して、インスタンスの「インスタンス設定オプション」を表示します。
6. Grafana アラートまたはプラグイン管理の横にある[編集] を選択します。

   - Grafana アラート

     Grafana アラートを有効にします。Grafana ワークスペースで Prometheus アラートを表示するには、チェックボックスをオンにして Grafana アラートをオンにします。バージョン 8 または 9 を実行しているワークスペースでは、Grafana アラートに対して複数の通知が送信されます。Grafana で定義されたアラートを使用する場合は、ワークスペースをバージョン 10.4 以降に更新することをお勧めします。

     従来の Grafana アラートを使用する場合は、Grafana アラートを有効にするの横にあるチェックボックスをオフにして、従来のダッシュボードアラートを有効にします。Grafana アラートが有効でない場合も、既存の Grafana アラートは評価されます。

     > **Note**
     >
     > 従来のダッシュボードアラートは、バージョン 11 で削除される予定です。Grafana バージョン 10 ワークスペースでは、Grafana アラート機能を確認できます。詳細については、「従来のダッシュボードアラートを Grafana アラートに移行する」を参照してください。

   - プラグイン管理

     プラグイン管理を有効にするには、[プラグイン管理を有効にする] チェックボックスをオンにします。プラグイン管理を有効にすると、Amazon Managed Grafana ワークスペースの管理者は、Grafana プラグインカタログを使用してプラグインをインストール、更新、または削除できまるようになります。このオプションは、Grafana バージョン 9 以降をサポートするワークスペースでのみ使用できます。

> **Note**
>
> Grafana アラートをオフにすると、Grafana アラートがオンになっている間にアラート設定に加えられたすべての変更が失われます。作成した新しいアラートルールも削除されます。

Grafana アラートの使用方法と、アラートのオン/オフの効果の詳細については、「Grafana バージョン 10 のアラート」を参照してください。

次のセクションでは、Amazon Managed Grafana API または AWS CLIを使用して Grafana インスタンス設定を変更する方法について紹介します。

API または AWS CLIを使用した設定

Grafana ワークスペースの設定は、Amazon Managed Grafana API または AWS CLIを使用して行うことができます。

> **Note**
>
> `configuration` は、後で追加される将来の構成設定を可能にする JSON 文字列です。

AWS CLI

AWS CLIを使用した Amazon Managed Grafana インスタンス設定の更新方法

以下のコマンドを実行して、インスタンスの Grafana アラート機能とプラグイン管理機能を有効にします。*<region>* および *<workspace-id>* 文字列をご自身のインスタンスに適した値に置き換えてください。

```
aws grafana update-workspace-configuration \
    --region region \
    --workspace-id <workspace-id> \
    --configuration '{"plugins": {"pluginAdminEnabled": true}, "unifiedAlerting":
 {"enabled": true}}'
```

現在、設定では以下のオプションがサポートされています。これを実行すると、Grafana アラートまたはプラグイン管理をオンまたはオフにすることができます。

- Grafana アラートを有効にするには、以下の設定オプションを使用します。

  ```
  --configuration '{"unifiedAlerting": { "enabled": true }}'
  ```

- プラグイン管理を有効にするには、以下の設定オプションを使用します。

```
--configuration '{"plugins": {"pluginAdminEnabled": true }}'
```

このオプションは、Grafana バージョン 9 以降をサポートするワークスペースでのみ使用できます。

Amazon Managed Grafana API

API を使用した Amazon Managed Grafana インスタンス設定の更新方法

以下のアクションを使用して、インスタンスの Grafana アラート機能とプラグイン管理機能を有効にします。*<workspace-id>* 文字列をご自身のインスタンスに適した値に置き換えてください。

```
PUT /workspaces/<workspace-id>/configuration HTTP/1.1
Content-type: application/json

{
   "configuration": "{ \"unifiedAlerting\": { \"enabled\": true }, \"plugins\":
 { \"pluginAdminEnabled\": true }}"
}
```

現在、設定では以下のオプションがサポートされています。これを実行すると、Grafana アラートまたはプラグイン管理をオンまたはオフにすることができます。

- Grafana アラートを有効にするには、以下の設定オプションを使用します。

  ```
  "configuration": "{\"unifiedAlerting\": { \"enabled\": true }}"
  ```

- プラグイン管理を有効にするには、以下のオプションを使用します。

  ```
  "plugins": "{\"pluginAdminEnabled\": true }"
  ```

  このオプションは、Grafana バージョン 9 以降をサポートするワークスペースでのみ使用できます。

Amazon Managed Grafana ワークスペースを削除する

Amazon Managed Grafana ワークスペースを削除すると、そのワークスペースに対して設定された情報。(ダッシュボード、データソース設定、アラート、スナップショットなど) も削除されます。

Amazon Managed Grafana ワークスペースの削除方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側のナビゲーションペインで、メニューアイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 削除するワークスペースの名前を選択します。
5. [削除] を選択します。
6. 削除を確認するには、ワークスペース名を入力して [削除] を選択します。

> **Note**
>
> この手順で削除されるのはワークスペースで、他のリソースは削除されません。例えば、ワークスペースで使用されていた IAM ロールは削除されません (使用されなくなった場合はロック解除される可能性があります)。

Grafana ワークスペースを使用する

ワークスペースは、メトリクスを視覚化し、データを分析するための作業場所です。メトリクス用のデータソースを設定し、クエリを実行します。ダッシュボード内にパネルを作成し、メトリクスを表示します。データを探索します。メトリクスにアラームを作成します。

このセクションのトピックでは、Amazon Managed Grafana のワークスペースの使用方法について説明します。

> **Note**
>
> 一部、ワークスペースの Grafana のバージョンによって異なる場合があります。各バージョンごとのドキュメントについては、Grafana バージョン 10 での作業、Grafana バージョン 9 での作業およびGrafana バージョン 8 での作業を参照してください。ワークスペースをあるバージョンから別のバージョンにアップグレードする方法については、「ワークスペースのバージョンを更新する」を参照してください。

トピック

- Grafana とは
- WorkSpace に接続する
- ユーザー、チーム、およびアクセス許可
- 最初のダッシュボードの作成
- プラグインを使用したワークスペースの拡張
- データソースに接続する
- Grafana バージョン 10 での作業
- Grafana バージョン 9 での作業
- Grafana バージョン 8 での作業
- 設定を変更する
- サポート用情報の収集
- 従来のダッシュボードアラート

Grafana とは

Grafana はオープンソースの視覚的表現および分析ソフトウェアです。これを使用、メトリクスの保存場所に関係なく、メトリクスのクエリ、視覚化、アラート、調査を行うことができます。

例えば、アプリケーションのメトリクス、ログ、トレースデータを表示する場合は、ダッシュボードを作成することができます。企業の管理者として複数のチームの Grafana を管理する場合は、プロビジョニングと認証を設定する必要があります。

以下のセクションでは、Grafanaデータベースで実行できる操作の概要と、詳細を学べるリンクを掲載しています。

メトリクスとログを調べる

1 回限りのクエリやアドホッククエリを使ってデータを動的に深掘りすることができます。画面を分割し、異なる時間範囲やクエリ、データソースを並べて比較することも可能です。

詳細については、「Grafana バージョン 10 で探索する」を参照してください。

アラート

Grafana アラートを使用している場合、アラートは次のようなさまざまなアラート通知機能を介して送信できます。

- Amazon SNS
- PagerDuty
- VictorOps
- OpsGenie
- Slack

詳細については、「Grafana バージョン 10 のアラート」を参照してください。

注釈

さまざまなデータソースからの重要なイベントをグラフに注釈として追加することができます。イベントの上にカーソルを置くと、イベントの詳細なメタデータやタグが表示されます。

この機能は Grafana のグラフマーカーとして表示され、問題が発生した場合にデータの関連性を把握するのに役立ちます。注釈を手動で作成するには、Ctrl キーを押しながらグラフを選択し、テキストを入力します。または、任意のデータソースからデータを取得することもできます。

詳細については、「視覚化に注釈を付ける」を参照してください。

ダッシュボード変数

テンプレート変数を使用して、さまざまな用途に再利用できるダッシュボードを作成することができます。これらのテンプレートでは、値はハードコードされないため、作成したダッシュボードは複数のサーバーで使用することができます。例えば、本番稼働用サーバーとテストサーバーがある場合は、両方に同じダッシュボードを使用できます。

テンプレート化により、データをより深く分析しやすくなります。例えば、全体のデータから北米のデータ、さらにテキサスのデータといった具合に、段階的に掘り下げることができます。これらのダッシュボードは、組織内のチーム間で共有することもできます。一般的なデータソースに最適なダッシュボードテンプレートを作成した場合、コミュニティ全体に提供して、使用する側がカスタマイズして使用することもできます。

詳細については、「変数」を参照してください。

WorkSpace に接続する

Amazon Managed Grafana ワークスペースを使用する前に、設定した ID プロバイダーでサインインして、ワークスペースに接続する必要があります。一部の ID プロバイダー経由で認証方法を設定していない場合は、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。

> **Note**
>
> プログラムからワークスペースに接続する場合は、API トークンを使用する必要があります。詳細については、「トークンを使用した認証」を参照してください。

Grafana ワークスペースにサインインする方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開いてサインインします。

2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. サインインするワークスペースの名前を選択します。
4. ワークスペースの詳細ページで、[Grafana ワークスペース URL] の下に表示される URL を選択します。
5. ワークスペースの URL を選択すると、Grafana ワークスペースコンソールのランディングページに移動します。でサインイン AWS IAM Identity Centerを選択し、E メールアドレスとパスワードを入力します。

> **Note**
>
> ID プロバイダーで認証を設定した場合、サインインボタンのテキストと要件は異なります。

ユーザー、チーム、およびアクセス許可

Amazon Managed Grafana のアクセス許可は、Amazon Managed Grafana コンソールを使用して、ワークスペース内で直接管理されます。

- ユーザー - ユーザーは、IAM Identity Center、または Amazon Managed Grafana コンソールの SAML を使用して設定した ID プロバイダーで認証されます。
- ロールアクセス — Amazon Managed Grafana コンソールを使用して、`User`、`Editor`、または `Admin` のロールでユーザーまたはグループにアクセス許可を付与し、ワークスペースへのデフォルト権限を付与できます。
- グループ、またはチーム - ユーザーのグループを作成して、2 つの方法でアクセス許可を付与できます。ID プロバイダー (または IAM Identity Center) でグループを作成できます。次に、Amazon Managed Grafana コンソールで、ユーザーと同様にこれらのグループにアクセス許可を付与できます。または、Grafana ワークスペースにチームを作成して、必要なロールをチームに付与できます。
- 特定のアクセス許可 – 特定のダッシュボード、フォルダ、またはデータソースのロールで付与されたアクセス許可を上書きする場合は、デフォルトのアクセス許可を削除して、特定のユーザーまたはチームにアクセス許可を割り当てることができます。これは Grafana ワークスペース内で実行します。

このセクションでは、Grafana ワークスペース内でアクセス許可の管理を実行する方法について説明します。

トピック

- [ユーザー]
- ユーザーロール
- チームの管理
- 許可の使用

[ユーザー]

Amazon Managed Grafana では、Grafana ワークスペースでユーザーを追加しません。代わりに、IAM Identity Center または ID プロバイダーを使用してユーザーを認証し、Amazon Managed Grafana コンソール内から Amazon Managed Grafana ワークスペースへのアクセス許可をユーザーに付与します。詳細については、「Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する」を参照してください。

ユーザーロール

Amazon Managed Grafana では、Amazon Managed Grafana ワークスペースの使用が有効な各ユーザーは、Amazon Managed Grafana コンソールの 3 つのロールのいずれかに割り当てられます。

- 管理者ロール — 管理者ロールを持つユーザーは以下を実行できます。
  - データソースを追加、編集、削除できます。
  - ユーザーとチームを追加、編集できます。
  - ダッシュボードが含まれるフォルダを追加、編集、削除できます。
  - エディタロールで許可されるすべての操作を実行できます。
- エディタロール — エディタロールを持つユーザーは以下を実行できます。
  - アクセス許可を持つダッシュボードで、ダッシュボード、パネル、アラートルールを表示、追加、編集できます。特定のフォルダとダッシュボードでこれを無効にできます。
  - プレイリストを作成、更新、または削除できます。
  - Explore にアクセスできます。
  - 通知チャネルを追加、編集、削除できます。
  - データソースは追加、編集、または削除できません。
  - ビューワーロールで許可される操作すべてを実行できます。
- ビューワーロール — ビューワーロールを持つユーザーは以下を実行できます。

- アクセス許可を持つダッシュボードを表示できます。特定のフォルダとダッシュボードでこれを無効にできます。
- プレイリストは作成、更新、または削除できません。
- Explore にはアクセスできません。
- 通知チャネルは追加、編集、削除できません。
- データソースは追加、編集、または削除できません。
- ダッシュボードやパネルは追加、編集、または削除できません。
- 他のユーザーやチームは管理できません。

Grafana ワークスペースからのユーザー割り当ておよびユーザーアクセス管理は、Amazon Managed Grafana でサポートされていません。ユーザーとグループのアクセス許可を管理する方法は、認証に IAM Identity Center と SAML のどちらを使用するかによって異なります。

- ワークスペースが認証に IAM Identity Center を使用している場合は、Amazon Managed Grafana コンソールまたは API を使用してロールを割り当てることができます。詳細については、「Amazon Managed Grafana ワークスペースに対するユーザーとグループのアクセス権を管理する」を参照してください。
- ワークスペースが認証に SAML を使用している場合、ユーザーロールはアサーション属性でのみ定義されます。詳細については、「アサーションマッピング」を参照してください。

チームの管理

チームを使用すると、ユーザーのグループに対し同時にアクセス許可を付与できます。チーム同期を設定して、Grafana ワークスペースと認証プロバイダー間のチームメンバーシップを自動的に同期することもできます。

チームの作成または削除

チームを作成してユーザーをグループで管理します。

チームの作成方法

1. サイドバーで、[設定] (歯車) アイコンを選択し、[チーム] を選択します。
2. [新しいチーム] を選択します。
3. [名前] に新しいチームの名前を入力し、[作成] を選択します。

チームの削除

1. サイドバーで、[設定] (歯車) アイコンを選択し、[チーム] を選択します。
2. チーム名の右側で、[X] を選択します。
3. 確定するため、[削除] を選択します。

チームに対するユーザーの追加または削除

以下の手順を使用して、チームにユーザーを追加、またはチームから削除します。

チームにユーザーを追加する方法

1. サイドバーで、[設定] (歯車) アイコンを選択し、[チーム] を選択します。
2. ユーザーを追加するチームを選択します。
3. [メンバーの追加] を選択します。
4. [チームメンバーを追加する] ボックスで、チームに追加するユーザーを選択してから、[チームに追加] を選択します。

チームからユーザーを削除する方法

1. サイドバーで、[設定] (歯車) アイコンを選択し、[チーム] を選択します。
2. ユーザーを削除するチームを選択します。
3. ユーザー名の右側で、[X] を選択します。
4. 確定するため、[削除] を選択します。

チーム同期の使用

チーム同期を使用すると、Grafana で認証プロバイダーのグループとチーム間の同期を設定できます。現在サポートされている認証プロバイダーは、IAM Identity Center および SAML です。

Grafana チームを外部グループと同期する方法は以下のとおりです。

1. Grafana コンソールで、[設定]、[チーム] に移動します。
2. IAM Identity Center のグループと同期するには、IAM Identity Center のグループ ID を入力します。SAML ベースの ID プロバイダーのグループと同期するには、Amazon Managed Grafana

ワークスペース設定ページの SAML 設定セクションの [アサーション属性グループ] フィールドに入力された属性名の値を入力します。

3. [グループの追加] を選択します。

許可の使用

Amazon Managed Grafana の Grafana ワークスペースでできることは、ユーザーに関連付けられたアクセス許可で定義されます。

Amazon Managed Grafana は次の 3 種類のアクセス許可を使用します。

- Grafana 管理者として付与されるアクセス許可
- チームのメンバーシップと関連付けられたアクセス許可
- 特定のフォルダまたはダッシュボードに付与されたアクセス許可

管理者ステータス、ユーザーに割り当てられたダッシュボードまたはフォルダのアクセス許可、データソースのアクセス許可に基づき、アクセス許可を付与できます。

ダッシュボードおよびフォルダのアクセス許可の概要

ダッシュボードおよびフォルダのアクセス許可を使用すると、エディタとビューワーのデフォルトロールベースのアクセス許可を削除できます。次に、特定のユーザーとチームにアクセス許可を割り当てられます。詳細については、「ダッシュボードおよびフォルダのアクセス権限」を参照してください。

データソースアクセス許可の概要

デフォルトでは、データソースは任意のユーザーがクエリできます。例えば、 `Viewer`ロールを持つユーザーは、アクセス許可を持つダッシュボードにあるクエリだけでなく、データソースにクエリを発行できます。

データソースのアクセス許可を使用すると、データソースのデフォルトのアクセス許可を変更して、クエリのアクセス許可を特定の [ユーザー] と [チーム] に限定できます。詳細については、「データソースの権限機能」を参照してください。

ダッシュボードおよびフォルダのアクセス権限

ダッシュボードおよびフォルダに付与された編集者と閲覧者のデフォルトロールベースのアクセス権限は、[権限] ページから削除することができます。このページを使用して、特定のユーザーとチームにアクセス権限を追加および割り当てることができます。

Amazon Managed Grafana には、次の権限レベルが用意されています。アクセス権限は、ワークスペースがサポートする Grafana のバージョンによって異なります。

バージョン 8 をサポートするワークスペースの場合:

- Admin: ダッシュボードの編集と作成、およびアクセス権限の編集が可能。フォルダの追加、編集、削除も可能です。
- Edit: ダッシュボードの編集および作成が可能。フォルダやダッシュボードのアクセス権限の編集、フォルダの追加、編集、削除はできません。
- View: 既存のダッシュボードおよびフォルダの表示のみが可能。

バージョン 9 以降をサポートするワークスペースの場合:

- Admin: ダッシュボードの作成、編集、削除が可能。フォルダの追加、編集、削除、フォルダへのダッシュボードとサブフォルダの作成が可能。管理者は、ダッシュボードおよびフォルダのアクセス権限の変更も可能です。
- Edit: ダッシュボードの作成、編集、削除が可能。フォルダの編集、削除、フォルダへのダッシュボードとサブフォルダの作成が可能。編集者はフォルダまたはダッシュボードのアクセス権限の変更はできません。
- View: 既存のダッシュボードおよびフォルダの表示のみが可能。

フォルダのアクセス権限の付与

フォルダのアクセス権限を付与する方法

1. サイドバーで、ダッシュボード (正方形) アイコンにカーソルを合わせて [管理] を選択します。
2. フォルダにカーソルを合わせ、[フォルダに移動] を選択します。
3. [権限] タブで、[権限を追加する] を選択します。
4. [権限追加対象] ダイアログボックスで、[ユーザ]ー、[チーム]、または「ロールオプション」のいずれかを選択します。ワークスペースで Grafana バージョン 10 以降を使用している場合は、ユーザー、チーム、サービスアカウント、またはロールを選択します。

5. 2 番目のボックスで、権限を追加するユーザー、チーム、サービスアカウント、またはロールを選択します。ワークスペースが Grafana バージョン 9 以前を使用していて、前のステップで「ロール」を選択した場合は、このステップをスキップしてください。
6. 3 番目のボックスで、追加する権限を選択します。
7. [保存] を選択します。

ダッシュボードのアクセス権限の付与

ダッシュボードのアクセス許可を付与する方法

1. ダッシュボードの右上隅にある歯車アイコンを選択して、[ダッシュボード設定] に移動します。
2. [権限] タブで、[権限を追加する] を選択します。
3. [権限追加対象] ダイアログボックスで、[ユーザ]ー、[チーム]、または「ロールオプション」のいずれかを選択します。ワークスペースで Grafana バージョン 10 以降を使用している場合は、ユーザー、チーム、サービスアカウント、またはロールを選択します。
4. 2 番目のボックスで、権限を追加するユーザー、チーム、サービスアカウント、またはロールを選択します。ワークスペースが Grafana バージョン 9 以前を使用していて、前のステップで「ロール」を選択した場合は、このステップをスキップしてください。
5. 3 番目のボックスで、追加する権限を選択します。
6. [保存] を選択します。

アクセスの制限

最も高い権限が常に優先されます。

- Admin ロールを持つユーザーの権限を上書きすることはできません。管理者は常にすべてにアクセスできます。
- より高い権限レベルを持つ一般的なルールが存在する場合、より低い権限レベルの具体的な権限は効果を持ちません。一般的なルールの権限レベルを削除するか、下げる必要があります。

Amazon Managed Grafana が複数の権限を解決する方法 – 例

次の例に、複数の権限がどのように解決されるかを紹介します。

例 1: **user1** に **Editor** ロールがある

ダッシュボードのアクセス権限:

- Editor ロールを持つすべてのユーザーが編集可能。
- user1 に表示権限がある。

結果: 最も高い権限が常に優先されるため、user1 には編集権限があります。

例 2: **user1** に閲覧者ロールが付与されており、**team1** のメンバーである

ダッシュボードのアクセス権限:

- Viewer ロールを持つすべてのユーザーが閲覧可能。
- user1 に Editor ロールが付与されており、編集が可能。
- team1 に Admin ロールが付与されている。

結果: 最も高い権限が常に優先されるため、user1 には管理者権限があります。

例 3: **user1** に異なるレベルで複数の権限が付与されている

ダッシュボードのアクセス権限:

- user1 に Admin ロール (親フォルダから継承) が付与されている。
- user1 に Editor ロールが付与されており、編集が可能。

結果: より低い権限を優先させることはできません。最も高い権限が常に優先されるため、user1 には管理者権限があります。

概要

- 表示: 既存のダッシュボードまたはフォルダのみを表示できます。
- より高い権限レベルを持つ一般的なルールが存在する場合、より低い権限レベルの具体的な権限は効果を持ちません。

データソースの権限機能

デフォルトでは、データソースは任意のユーザーがクエリできます。例えば、Viewer ロールを持つユーザーは、アクセス権を持つダッシュボードにあるクエリだけでなく、データソースにもクエリを発行できます。

データソースの権限機能を使用することで、ユーザーがデータソースをクエリするためのアクセス権限を制限できます。各データソースに権限ページがあり、特定のユーザーとチームに対してクエリ権限を付与または制限することができます。

データソースの権限機能の有効化

データソースの権限機能が有効になっている場合、そのデータソースへの管理者アクセス権限およびクエリアクセス権限は、デフォルトで管理者ユーザーに制限されます。特定のユーザーやチームに対して、選択的にアクセス権を付与することができます。

データソースの権限機能を有効にする方法

1. [設定]、[データソース] の順に移動します。Grafana バージョン 10 をサポートするワークスペースの場合は、[接続]、[データソース] の順に移動します。
2. 権限機能を有効にするデータソースを選択します。
3. [権限] タブで、[有効] を選択します。

> **Warning**
>
> デフォルトのデータソースの権限機能を有効にすると、権限リストにないユーザーはクエリを呼び出すことができません。デフォルトのデータソースを使用するパネルは、それらのユーザーに対して `Access denied to data source` エラーを返します。

ユーザーとチームにデータソースのクエリを許可する

データソースの権限機能を有効にした後は、デフォルトでは管理者のみがそのデータソースにアクセスできます。ユーザーまたはチームにクエリ権限を割り当てることができます。クエリ権限を付与することで、データソースをクエリするためのアクセス権が付与されます。

ユーザーとチームにクエリ権限を割り当てる方法

1. [設定]、[データソース] の順に移動します。Grafana バージョン 10 をサポートするワークスペースの場合は、[接続]、[データソース] の順に移動します。
2. クエリ権限を付与するデータソースを選択します。
3. [権限] タブで、[権限を追加する] を選択します。
4. [チーム] または [ユーザー] を選択します。Grafana バージョン 10 以降をサポートするワークスペースでは、[サービスアカウント] または [ロール] を選択することもできます。

5. クエリ権限を付与するチーム、ユーザー、サービスアカウント、またはロールを選択し、[保存] を選択します。

データソースの権限機能の無効化

データソースの権限機能を有効にしていて、データソースの権限機能をデフォルトに戻す場合は、次の手順に従います。

> **Note**
>
> データソース用に作成された既存の権限はすべて削除されます。

データソースに対する権限の編集方法

1. [設定]、[データソース] の順に移動します。Grafana バージョン 10 をサポートするワークスペースの場合は、[接続]、[データソース] の順に移動します。
2. 権限機能を無効にするデータソースを選択します。
3. [権限] タブで、[権限機能の無効化] を選択します。

最初のダッシュボードの作成

ダッシュボードの作成

Grafana コンソールでダッシュボードを作成するには、以下の手順に従います。

最初のダッシュボードを作成する方法

1. 左パネルの + アイコンを選択して [ダッシュボードの作成] を選択し、[新しいパネルを追加] を選択します。
2. [新しいダッシュボード/パネルの編集] ビューで、[クエリ] タブを選択します。
3. クエリするデータソースを選択してクエリを設定します。例えば、データソースとして [TestDB] を追加する場合、ランダムウォークダッシュボードと呼ばれるサンプルダッシュボードが生成されます。

時系列の概要

1 日を通じた外気温の変化について知りたいとします。1 時間に 1 回温度計を確認して、現在の温度と時刻を書き留めます。しばらくすると、次のようなデータが集まります。

時間	値
09:00	24°C
10:00	26°C
11:00	27°C

このような温度データは時系列の一例で、時間順の一連の測定です。テーブルの各行は、特定の時間の 1 つの測定値を表します。

テーブルは個々の測定値の識別に役立ちますが、全体像の確認が難しい場合があります。より一般的な時系列の視覚化はグラフで、各測定値が時間軸に沿って配置されます。グラフなどの視覚的表現を使用すると、確認するのが難しいデータのパターンや特徴を簡単に発見できます。

以下のものは時系列のその他の例です。

- CPU およびメモリの使用量
- センサーデータ
- 株式市場指数

これらの例は時系列順に測定された数列ですが、以下の属性も共有しています。

- 新しいデータは、09:00、10:00、11:00 などのように、一定間隔で最後に追加されます。
- 測定値が追加後に更新されることはほとんどありません。例えば、昨日の温度は変更されません。

時系列には説得力があります。いつでもシステムの状態を分析できるため、過去について理解するのに役立ちます。時系列を見れば、空きディスクスペースが 0 になった直後にサーバーがクラッシュしたことがわかります。

時系列を使用するとデータの傾向が明らかになり、未来の予測にも役立ちます。例えば、過去数か月間に登録されたユーザー数が毎月 4% 増加している場合、年末にユーザーベースがどの程度大きくなるかを予測できます。

一部の時系列には、既知の期間全体で繰り返すパターンがあります。例えば、通常の場合気温は日中高くなってから夜間に低下します。これらの周期的、または季節的な時系列を特定すると、自信をもって次の期間を予測できます。システム負荷が毎晩 6 時頃にピークに達することがわかっている場合、直前に機器を追加できます。

時系列の集計

測定内容が異なると、データが大きく異なる場合があります。測定間隔よりも長い期間を比較する場合はどうしますか? 温度を 1 時間に 1 回測定する場合、1 日あたり 24 のデータポイントが集まります。長年にわたる 8 月の気温を比較するには、31 x 24 のデータポイントを 1 つに結合する必要があります。

測定値の集合を組み合わることを集計と呼びます。時系列データの集計方法はいくつかあります。以下は一般的な方法です。

- 平均は、すべての値の合計を値の数で割ったものを返します。
- 最小と最大は、コレクション内の最小値と最大値を返します。
- 合計はコレクションのすべての値を返します。
- カウントはコレクション内の値の数を返します。

例えば、1 か月のデータを集計すると、2017 年 8 月が平均で前年より温かかったと判断できます。最も高い温度の月を確認する場合は、各月の最高温度を比較します。

時系列データの集計方法の決定は重要であり、データを使用して伝える目的によって異なります。一般的には、異なる集計を使用して、同じ時系列データを異なる方法で視覚化します。

時系列およびモニタリング

IT 業界では、インフラストラクチャ、ハードウェア、アプリケーションなどの各イベントを監視するために時系列データが頻繁に収集されます。機械生成された時系列データは通常短い間隔で収集されるため、予期しない変更への対応や、発生直後の対応が可能です。データは速いペースで蓄積されるため、データを効率的に保存してクエリする方法を持つことが必要となります。その結果、時系列データ用に最適化されたデータベースの人気が近年高まっています。

時系列データベース

時系列データベース (TSDB) は、時系列データのために明示的に設計されたデータベースです。通常のデータベースを使用して測定値を保存することもできますが、TSDB にはいくつかの便利な最適化が用意されています。

最新の TSDB では、測定値が追加されるのが 1 回のみで、更新または削除されることかせほとんどないという事実を活用しています。例えば、各測定のタイムスタンプは時間が経過してもほとんど変化せず、冗長データが保存されます。

次の例は Unix タイムスタンプの数列を示しています。

```
1572524345, 1572524375, 1572524404, 1572524434, 1572524464
```

これらのタイムスタンプを見ると、すべて 1572524 で始まり、ディスクスペースの使用が中途半端になります。代わりに、次の例のように、後続の各タイムスタンプを最初のタイムスタンプとの差、つまり差分として保存できます。

```
1572524345, +30, +29, +30, +30
```

次の例のように、これらの差分の差分を計算することもできます。

```
1572524345, +30, -1, +1, +0
```

測定が周期的に行われる場合、これらの差分の差分の大半は 0 になります。このように最適化することにより、TSDB が使用するスペースは他のデータベースよりも大幅に少なくなります。

TSDB のもう 1 つの特徴はタグを使用して測定値をフィルタリングする機能です。各データポイントには、測定場所などのコンテキスト情報を追加するタグが用意されています。

Grafana では次の TSDB がサポートされています。

- Graphite
- InfluxDB
- Prometheus

```
weather,location=us-midwest temperature=82 1465839830100400200
  |    -------------------- --------------  |
  |             |             |             |
  |             |             |             |
+-----------+--------+-+---------+-+---------+
|measurement|,tag_set| |field_set| |timestamp|
+-----------+--------+-+---------+-+---------+
```

時系列データの収集

時系列の保存場所ができましたが、測定値はどのように収集しますか? 時系列データを収集するには、通常、モニタリングするデバイス、機器、またはインスタンスにコレクターをインストールします。一部のコレクターは特定のデータベースを念頭に置いて作成されており、一部のコレクターでは異なる出力先をサポートしています。

以下のものはコレクターの例です

- collectd
- statsd
- Prometheus exporters
- Telegraf

コレクターは、データベースにデータをプッシュするか、データベースにコレクターのデータをプルさせます。各アプローチにはそれぞれ長所と短所があります。

	長所	短所
プッシュ	複数の送信先にデータをレプリケーションする方が簡単です。	TSDB は送信データの量を制御することはできません。
プル	データインジェストの量とデータの信頼性をより細かく制御できます。	ファイアウォール、VPN またはロードバランサーでは、エージェントへのアクセスが難しくなる場合があります。

すべての測定値をデータベースに書き込むのは非効率なので、コレクターはデータを事前に集約して TSDB に周期的に書き込みます。

時系列のディメンション

時系列データでは、多くの場合データは複数の時系列セットです。多くの Grafana データソースでは、このタイプのデータをサポートしています。

よくある例では、ディメンションとして複数の追加プロパティを使用する測定に対して単一のクエリを発行します。例えば、ロケーションプロパティを使用して温度測定値をクエリします。この場合、その単一のクエリから複数のシリーズが返され、シリーズごとにディメンションとして一意の場所があります。

一連の時系列で一意のシリーズを識別するために、Grafana はラベルにディメンションを保存します。

ラベル

Grafana の各時系列にはオプションのラベルがあります。ラベルはディメンション を識別するための、一組のキーと値のペアです。ラベルの例は {location=us} または {country=us,state=ma,city=boston} です。一連の時系列内では、名前とラベルの組み合わせによって各シリーズが識別されます。例えば、temperature {country=us,state=ma,city=boston}。

時系列データのさまざまなソースには、ネイティブに保存されるディメンションがあるか、データをディメンションに抽出するための一般的なストレージパターンがあります。

通常、TSDB はディメンションをネイティブにサポートします。Prometheus ではラベルでディメンションを保存します。Graphite や OpenTSDB などの TSDB では、代わりにタグという用語を使用しています。

SQL などのテーブルデータベースの場合、これらのディメンションは一般的にクエリの GROUP BY パラメータです。

テーブル形式による複数のディメンション

テーブルレスポンスを返す SQL または SQL に類似したデータベースでは、追加のディメンションは一般的にクエリレスポンステーブルの列です。

単一のディメンション

例えば、次のようなクエリを考えてみます。

```
SELECT BUCKET(StartTime, 1h), AVG(Temperature) AS Temp, Location FROM T
  GROUP BY BUCKET(StartTime, 1h), Location
  ORDER BY time asc
```

そのクエリが 3 つの列を持つテーブルを返すとします。

StartTime	Temp	場所
09:00	24	LGA
09:00	20	BOS

StartTime	Temp	場所
10:00	26	LGA
10:00	22	BOS

テーブル形式はロング形式の時系列で、トールとも呼ばれます。ロケーションには繰り返しタイムスタンプと繰り返し値があります。この場合、セット内の 2 つの時系列は `Temp {Location=LGA}` および `Temp {Location=BOS}` として識別されます。

セット内の個々の時系列は次のディメンションを使用して抽出されます。

- 時系列の時間インデックス `StartTime` の時間型列
- シリーズ名 `Temp` の数値型列
- Location=LGA などのラベルを構築する、文字列型 `Location` 列の名前と値

複数ディメンション

複数の文字列の列 (`GROUP BY BUCKET(StartTime, 1h), Location, Sensor` など) ごとに選択およびグループ化するようにクエリを更新すると、さらにディメンションが追加されます。

StartTime	Temp	場所	センサー
09:00	24	LGA	A
09:00	24.1	LGA	B
09:00	20	BOS	A
09:00	20.2	BOS	B
10:00	26	LGA	A
10:00	26.1	LGA	B
10:00	22	BOS	A
10:00	22.2	BOS	B

この場合、ディメンションを表すラベルには、2 つの文字列型列に基づく 2 つのキー Location と Sensor があります。データは以下の 4 つのシリーズになります。

- `Temp {Location=LGA,Sensor=A}`
- `Temp {Location=LGA,Sensor=B}`
- `Temp {Location=BOS,Sensor=A}`
- `Temp {Location=BOS,Sensor=B}`

> **Note**
>
> 注: Grafana で複数のアラートにマッピングする手法では、複数のディメンションをサポートしていません。代わりに、単一アラートに対する複数条件として扱われます。

複数値

SQL のようなデータソースでは、ディメンションとして使用する追加文字列の列の有無にかかわらず、`AVG(Temperature) AS AvgTemp, MAX(Temperature) AS MaxTemp` のように、複数の数値列を選択できます。これにより、複数のディメンションと組み合わせると、多数のシリーズが発生する場合があります。今のところ複数値の選択は視覚化でのみ使用できます。

ヒストグラムおよびヒートマップの概要

ヒストグラムは数値データの分布をグラフィカルに表現したものです。値をグループ化してバケット (ビンとも呼ばれる) にします。次に、各バケットに含まれる値の数を数えます。

ヒストグラムでは、実際の値をグラフ化する代わりにバケットをグラフ化します。各バーはバケットを表し、バーの高さはバケットの間隔に含まれる値の頻度 (カウントなど) を表します。

ヒストグラムは特定の時間範囲の値分布のみを検討します。ヒストグラムの問題は、時間が経過しても、その分布に傾向や変化が見られないことです。そこではヒートマップが役に立ちます。

ヒートマップ

ヒートマップは経時的ヒストグラムのようなもので、各タイムスライスがそのヒストグラムを表します。頻度表現としてバーの高さの代わりにセルを使用し、バケット内の値数に比例してセルを色付けします。

バケット化済みデータ

多くのデータソースは、次のような経時的ヒストグラムをサポートしています。

- Amazon OpenSearch Service (ヒストグラムバケットの集計を使用)
- Prometheus (ヒストグラムのメトリクスタイプと 形式 オプションを [ヒートマップ] に設定)

通常は、バケットバウンドを表す名前のシリーズを返すデータソースや、バウンドで昇順にソートされたシリーズを返すデータソースを使用できます。

未加工データ対集計データ

ヒートマップを通常の (バケット化されていない) 時系列データで使用する場合は、データが時系列バックエンドによって集計済みであることが多いことに注意してください。大抵の時系列クエリは未加工サンプルデータを返しません。代わりに、時間間隔または maxDataPoints 制限を集計関数 (通常は平均) と組み合わせたグループが含まれます。

クエリの時間範囲に応じて異なります。重要なのは、Grafana が実行するヒストグラムバケット処理が、既に集計され平均化済みのデータに対して実行される場合があることを理解することです。より正確なヒートマップを得るには、メトリクス収集中にバケット処理を行うか、OpenSearch にデータを保存するか、未加工データに対してヒストグラムバケット処理の実行をサポートする他のデータソースに保存することをお勧めします。

より多くのデータポイントを返すために、クエリ内のグループを削除または時間によって減らす (または maxDataPoints を増やす) と、ヒートマップはより正確になります。ただし CPU とメモリに負荷がかかる場合もあります。データポイントの数をむやみに多くすると、フリーズやクラッシュする場合があります。

プラグインを使用したワークスペースの拡張

Grafana プラグインは、新しいデータソースに接続する機能を追加、またはワークスペースに視覚化やその他の機能を追加します。大きく分けて、プラグインには 3 つのタイプがあります。

- パネルプラグイン – パネルプラグインはダッシュボードで使用できる新しい視覚化タイプを追加します。これらはフロントエンドのデータのレンダリングを定義します。
- データソースプラグイン – データソースプラグインは外部データソースと通信して、Grafana が使用できる形式でデータを返します。

- アプリプラグイン – アプリケーション、アプリプラグインとも呼ばれます。これにはバンドルデータソースとパネルが含まれ、Grafana ワークスペース内でまとまりのあるエクスペリエンスを提供できます。

> **Note**
>
> Amazon は、プラグインが失敗することが多い、またはメンテナンスされていないと判断した場合、コンソールで使用可能なプラグインのリストからプラグインを削除することがあります。

バージョン 9 以降をサポートする Amazon Managed Grafana ワークスペースでは、プラグイン管理を有効化できます。これにより、ワークスペース管理者はプラグインカタログからプラグインをインストールまたはアンインストールできます。

プラグインカタログによるプラグインの検索

Amazon Managed Grafana ワークスペースには、インストール済みのすべてのプラグインと、ワークスペースにインストールできるすべてのプラグインのリストを示すページが含まれています。このページはプラグインカタログです。デフォルトでインストールされるプラグインに加えて、最大 50 個のプラグインをインストールできます。

使用可能なプラグインは次のカテゴリに大別されます。

- AWS データソース – これは、Amazon Managed Grafana が提供するアプリケーションプラグインで、アカウント内のリソースを簡単に検出 AWS できます。これはデフォルトでインストール済みです。詳細については、「AWS データソースプラグインを使用して AWS データを検索する」を参照してください。
- コアプラグイン – これらのプラグインは Grafana でデフォルトで提供されます。これには一般的なデータソースとパネルの視覚化が含まれます。プラグインカタログでは [コア] としてタグ付けされます。これらはデフォルトでインストールされ、削除できません。
- エンタープライズプラグイン – これらのプラグインは、エンタープライズライセンスを持つ Grafana ワークスペースで利用可能です。これらはデフォルトではインストールされません。プラグインカタログでは [エンタープライズ] としてタグ付けされます。有効なエンタープライズライセンスがある場合にのみインストールできます。ワークスペースをエンタープライズライセンスにアップグレードする方法については、「Amazon Managed Grafana Enterprise プラグインに対するアクセス権管理」を参照してください。

- コミュニティプラグイン – これらのプラグインは、Grafana Labs など、さまざまなソースの Grafana ワークスペース用に提供されています AWS。バージョン 9 以降をサポートする Grafana ワークスペースでは、デフォルトでインストールされません (以前のワークスペースにはこれらの一部が自動的にインストールされます)。これらは通常オープンソースプラグインです。これらのプラグインはインストールまたは削除できます。

> **Note**
>
> コミュニティプラグインの使用はユーザーが自由に決定できます。お客様と AWS間の責任共有モデルの一環として、これらのサードパーティープラグインに関して、何をワークスペースにインストールするか理解する必要があります。また、お客様はセキュリティニーズを満たすプラグインについても責任を負います。

プラグインサポート

プラグインはさまざまなソースから提供され、それによってサポートが異なります。

- AWS データソースプラグイン – このプラグインは によって提供され、 によってサポートされています AWS。
- エンタープライズプラグイン – エンタープライズプラグインは AWS と Grafana Labs の両方でサポートされています。どちらのサポートチームでも問題を送信できます。
- コアプラグイン – AWS または Grafana Labs が提供するコアプラグインおよびその他のプラグインは、 によって Amazon Managed Grafana でサポートされています AWS。GitHub でバグ修正や機能強化の問題を送信するか、 AWS または Grafana Labs でチケットを作成できます。
- コミュニティプラグイン – AWS または Grafana Labs によって作成されていないコミュニティプラグインは通常、GitHub の問題やその他のフォーラムを通じてサポートされます。このような場合のサポート情報は、プラグインカタログのプラグインの詳細情報に含まれています。

Amazon Managed Grafana または Grafana の GitHub フォーラムを通じてプラグインの問題を送信することもできます。

プラグインバージョン

ほとんどのプラグインは定期的に更新されます。Amazon Managed Grafana ワークスペースのプラグインカタログにはプラグインの最新バージョンが表示され、インストールするバージョンを選択できます。プラグインに既知のセキュリティ問題が残された古いバージョンがある場合、その古いバージョンは利用できません。

インストール済みのプラグインを更新することもできます。

> **Note**
>
> インストール済みのプラグインのセキュリティ問題が修正された、新しいバージョンのプラグインが使用可能になる場合があります。重大な問題の場合、Amazon Managed Grafana がワークスペースのプラグインを修正済みバージョンに自動更新する場合があります。

プラグインカタログによるプラグインの管理

Amazon Managed Grafana ワークスペースのプラグインはプラグインカタログから管理します。ワークスペース内のプラグインカタログに一覧表示されているプラグインのみをインストールできます。

以下に、プラグインカタログを使用するための前提条件と、プラグインカタログの検索方法について説明します。

前提条件

- バージョン 9 をサポートする Amazon Managed Grafana ワークスペース、およびそのワークスペースにログインできるアカウントが必要です。
- ワークスペースではプラグイン管理を有効にすることが必要です。
- ユーザーアカウントは Amazon Managed Grafana ワークスペースの管理者である必要があります。
- エンタープライズプラグインをインストールして使用するには、まず エンタープライズライセンスにアップグレードする必要があります。

プラグインカタログの表示方法

1. Amazon Managed Grafana ワークスペースにサインインします。
2. 左側のメニューから、[管理]、[プラグイン] の順に選択します。プラグインカタログが開きます。
3. デフォルトでは、プラグインカタログにはインストール済みのプラグインが一覧表示されます。使用可能なすべてのプラグインを表示するには、カタログの上部にある [状態] フィルターで [すべて] を選択します。インストールされたプラグインには [インストール済み] というタグが含まれます。

プラグインのインストールまたは削除

> **Note**
>
> 以前のセクションの前提条件を満たす必要があります。そうしないとプラグインを変更するアクセス許可を得られません。

Grafana プラグインのインストールまたは削除方法

1. プラグインカタログに移動します。
2. デフォルトでは、プラグインカタログにはインストール済みのプラグインのみが一覧表示されます。使用可能なすべてのプラグインを表示するには、カタログの上部にある [状態] フィルターで [すべて] を選択します。インストールされたプラグインには [インストール済み] というタグが含まれます。
3. インストールまたはアンインストールするプラグインを選択します。例えば、Datadog データソースを削除する場合は、[Datadog] プラグインを選択します。
4. プラグインの詳細ページで、アンインストールまたはインストールのオプションを選択します。
5. プラグインのインストール後、ワークスペースのすべての部分で変更が同期されるまでに最大で数分かかることがあります。新しいプラグインを使用する前に、数分待つことをお勧めします。

> **Note**
>
> ワークスペースには (デフォルトのコアプラグインを超える) 50 個のプラグインをインストールできます。

プラグインの更新

既存の Grafana プラグインの更新方法

1. Amazon Managed Grafana ワークスペースにサインインします。
2. 左側のメニューから、[管理]、[プラグイン] の順に選択します。プラグインカタログが開き、インストール済みのプラグインのみが一覧表示されます。
3. 更新するプラグインを選択します。

4. プラグインの詳細ページで、利用可能な更新があるかどうかを確認します。ある場合は、プラグインを更新するオプションを選択して更新するバージョンを選択します。

> **Note**
>
> プラグインを変更するためのアクセス許可がないという注記が表示された場合は、ワークスペースでプラグイン管理が有効になっていることを確認します。また、Amazon Managed Grafana ワークスペースの管理者である必要もあります。

AWS データソースプラグインを使用して AWS データを検索する

AWS には、Amazon Managed Grafana ワークスペースのデータソースとして リソースを簡単に検出して使用 AWS できるようにするアプリケーションプラグインが用意されています。AWS [データソース] プラグインは、新しいワークスペースにデフォルトでインストールされます。

AWS データソースプラグインには、検出のために リソースにアクセスするためのアクセス許可が必要です。詳細については、「必要なアクセス許可」を参照してください。

AWS データソースプラグインを開く

AWS データソースプラグインを開くには

1. Amazon Managed Grafana ワークスペースにサインインします。
2. 左上のメニューから、[アプリ]、[AWS データソース] の順に選択します。

AWS データソースプラグインインターフェイスに、リソースを検索できる AWS サービスが一覧表示されます。

リソースの検出

から リソースを検出するには AWS アカウント

1. AWS データソースプラグインを開きます。
2. AWS サービスのリストから、データソースとして使用するリソースを検索するサービスを選択します。この例では、[Amazon Managed Service for Prometheus] を選択します。選択すると、そのサービスが選択され状態でデータソースタブが開きます。

3. リソースを検索する AWS リージョンを選択します。例えば、US East (N. Virginia) を選択します。

> **Note**
>
> リソースを検索するには、プラグインに、そのリージョン内のサービスにアクセスするための適切なアクセス許可が必要です。

4. 一部のサービスは、1 つのリージョンに複数のリソースを持つことができます。リージョンに複数のリソースがある場合、 AWS データソースプラグインは選択できるリストを提供します。

   リソース (この例では Amazon Managed Service for Prometheus) のリストから、データソースとして使用するリソースを選択します。この例の場合、Amazon Managed Service for Prometheus ワークスペースを選択すると、そのリソースがデータソースとして設定されます。データソースは、Amazon Managed Grafana でダッシュボードまたはモニタリングで使用できる状態になります。

5. プロビジョニングしたサービスおよびリージョンのリソースは、ページの下部に表示されます。

   (任意) [設定に移動] を選択すると、そのデータソースの設定を表示および編集することができます。

> **Note**
>
> AWS データソースプラグインは、ワークスペースにインストールされている個々のデータソースプラグインによって異なります。例えば AWS X-Ray 機能を使用する場合、プラグインカタログから X-Ray データソースプラグインをインストールする必要があります。

プラグインのバージョンと更新

AWS データソースプラグインは定期的に更新されます。新しいワークスペースにインストールされているバージョンは通常、最新バージョンではありません。新しいバージョンでは、ワークスペースにインストールされているバージョンよりも多くの機能を使用できます。例えば、新しいバージョンでは、データソースとして追加の AWS サービスをサポートする場合があります。

AWS データソースプラグインの各バージョンの変更を確認するには、「変更ログ」を参照してください。

プラグインを新しいバージョンに更新するには、プラグインの更新 の標準手順に従ってください。

> **Note**
>
> AWS データソースプラグインの新しいバージョンに更新する場合は、Amazon Managed Grafana によって管理されていない新しいデータソースに追加のアクセス許可を提供する必要があります。

必要なアクセス許可

AWS データソースプラグインには、 AWS リソースにアクセスするためのアクセス許可が必要です。これを行う最も簡単な方法は、Amazon Managed Grafana がアクセス権限を管理できるようにすることです。データソースへの権限をサービス管理で行えるよう設定する方法については、「データソースおよび通知チャネルのアクセス権管理」を参照してください。Amazon Managed Grafana は、デフォルトで AWS データソースプラグインに含まれる AWS リソースのアクセス許可を管理できます。

AWS データソースプラグインをワークスペースにデフォルトで含まれているバージョンよりも新しいバージョンに更新すると、Amazon Managed Grafana によって自動的にアクセス許可が管理されない AWS リソースのサポートが追加される可能性があります。この場合、権限を自分で追加する必要があります。例えば、 AWS IoT TwinMaker が最新バージョンのプラグイン (バージョン 1.9.0) に追加され、これらのアクセス許可は Amazon Managed Grafana によって管理されません。

特定のデータソースに対する権限の詳細については、 データソースに接続する セクションで提供されているデータソースの詳細を参照してください。例えば、 an AWS IoT TwinMaker データソースに接続するセクションには、Amazon Managed Grafana に access AWS IoT TwinMaker へのアクセス許可を付与する方法の詳細が含まれています。

データソースに接続する

Amazon Managed Grafana は、さまざまなデータソース をサポートしています。データソースとは、Grafana でダッシュボードを作成する際にクエリを実行できるストレージバックエンドのことです。各データソースには、それぞれの機能や特性に合わせた専用のクエリ工ディタがあります。

データソースごとにクエリ言語や機能は異なります。複数のデータソースからのデータを 1 つのダッシュボードに結合できます。

Amazon Managed Grafana を使用するすべての AWS アカウントには、多くのデータソースを作成または設定するためのアクセス権があります。一部のデータソースでは、特定のプラグインを追加でインストールする必要があります。ワークスペースを Amazon Managed Grafana Enterprise プラグインにアップグレードする場合、Enterprise データソース用のプラグインのインストールも必要になる場合があります。以下のセクションでは、利用可能なデータソースの多くの詳細について説明しますが、Grafana コミュニティによって、ワークスペースのプラグインカタログで使用できる新しいデータソースが追加されることがあります。

> **Note**
>
> アカウント内の AWS リソースを検出し、それらをクエリするためのデータソースを設定するために、Amazon Managed Grafana は を提供しますAWS データソースプラグインを使用して AWS データを検索する。

特別なデータソース

Amazon Managed Grafana には、次の 3 つの特別なデータソースがあります。

- Grafana (以前のバージョンの Grafana での呼称は TestDB でした) – ランダムウォークデータの生成や、ファイルの一覧表示ができるビルトイン データソース。視覚的表現のテストや実験の実行に役立ちます。
- [混合] – 同じパネル内で複数のデータソースをクエリするために使用します。このデータソースを使用する場合は、追加する新しいクエリごとにデータソースを指定できます。ただし、最初のクエリは [混合]] を選択する前に指定したデータソースを使用します。

  既存のクエリを混合データソースを使用するように変更することはできません。
- [ダッシュボード] – 同じダッシュボード内の別のパネルからの結果セットを使用するために使用します。

> **Important**
>
> Amazon Managed Grafana にはデータソースのタイムアウト制限があり、データソースに設定されているタイムアウト制限が上書きされる可能性があります。2 つの制限のうち低いほうが他の制限よりも優先されます。Amazon Managed Grafana データソースのタイムアウト

制限の詳細については、の「Amazon Managed Grafana サービスクォータ」を参照してくださいAWS 全般のリファレンス。

トピック

- Amazon Managed Grafana が AWS データソースアクセス AWS Organizations のために と連携する方法
- ビルトイン データソースに接続する
- Enterprise データソースに接続する

Amazon Managed Grafana が AWS データソースアクセス AWS Organizations のために と連携する方法

を使用すると AWS Organizations、複数の AWS アカウントのデータソース設定とアクセス許可設定を一元管理できます。Amazon Managed Grafana ワークスペース AWS アカウント を使用する では、他の組織単位を指定して、 AWS データソースをプライマリアカウントで表示できるようにします。

例えば、組織内の 1 つのアカウントに組織内の他のアカウントのデータソースへのアクセス権を付与して、これを Amazon Managed Grafana の管理アカウントとして使用することができます。管理アカウントで、管理アカウントでアクセスする AWS データソースを持つすべての組織単位を一覧表示します。これらのデータソースの設定に必要なロールとアクセス許可ポリシーが自動的に作成され、Amazon Managed Grafana ワークスペースの Grafana コンソールに表示されます。

Organizations の詳細については、「AWS Organizationsとは」を参照してください。

Amazon Managed Grafana は、 AWS CloudFormation StackSets を使用して、Amazon Managed Grafana が AWS 組織全体のデータソースに接続するために必要な AWS Identity and Access Management (IAM) ロールを自動的に作成します。Amazon Managed Grafana が IAM ポリシーを管理して組織全体のデータソースにアクセスする前に、組織の管理アカウントで AWS CloudFormation StackSets を有効にする必要があります。Amazon Managed Grafana は、初めて必要になったときにこれを自動的に有効にします。

AWS IAM Identity Center および Organizations との統合のためのデプロイシナリオ

AWS IAM Identity Center と Organizations の両方で Amazon Managed Grafana を使用している場合は、次の 3 つのシナリオのいずれかを使用して、組織内に Amazon Managed Grafana ワークスペー

スを作成することをお勧めします。シナリオごとに、十分なアクセス許可を持つアカウントにサインインする必要があります。詳細については、「Amazon Managed Grafana のサンプルポリシー」を参照してください。

スタンドアロンアカウント

スタンドアロンアカウントは、Organizations の組織のメンバーではない AWS アカウントです。これは、初めて試す場合 AWS のシナリオです。

このシナリオでは、AWSGrafanaAccountAdministrator、AWSSSOMemberAccountAdministrator、AWS IAM Identity Center および AWSSSODirectoryAdministrator ポリシーを持つアカウントにサインインすると、Amazon Managed Grafana が自動的に と Organizations を有効にします。AWSGrafanaAccountAdministrator 詳細については、「Amazon Managed Grafana ワークスペースとユーザーを IAM Identity Center を使用する単一スタンドアロンアカウントで作成および管理する」を参照してください。

IAM Identity Center が既に設定されている既存の組織のメンバーアカウント

メンバーアカウントでワークスペースを作成するには、AWSGrafanaAccountAdministrator、AWSSSOMemberAccountAdministrator、およびAWSSSODirectoryAdministrator ポリシーを持つアカウントでサインインする必要があります。詳細については、「IAM Identity Center を使用するメンバーアカウントの Grafana 管理者」を参照してください。

メンバーアカウントにワークスペースを作成し、そのワークスペースから組織内の他の AWS アカウントのリソースにアクセスできるようにするには、ワークスペースでカスタマー管理アクセス許可を使用する必要があります。詳細については、「「カスタマー管理」のアクセス許可」を参照してください。

サービスマネージド型のアクセス許可を使用して、ワークスペースが組織内の他の AWS アカウントからリソースにアクセスできるようにするには、組織の管理アカウントにワークスペースを作成する必要があります。ただし、組織の管理アカウントに Amazon Managed Grafana ワークスペースやその他のリソースを作成することはベストプラクティスではありません。Organizations のベストプラクティスの詳細については、「管理アカウントのベストプラクティス」を参照してください。

> **Note**
>
> 2019 年 11 月 25 日より前に管理アカウント AWS IAM Identity Center で を有効にした場合は、管理アカウントで IAM Identity Center 統合アプリケーションも有効にする必要があります。必要に応じて、管理アカウントで IAM Identity Center 統合アプリケーションを有効に

した後、メンバーアカウントでも有効にすることができます。これらのアプリケーションを有効にするには、IAM Identity Center の [設定] ページの IAM Identity Center 統合アプリケーションセクションで [アクセスを有効にする] を選択します。詳細については、「IAM Identity Center 統合アプリケーションの有効化」を参照してください。

IAM Identity Center がまだデプロイされていない既存の組織のメンバーアカウント

このシナリオでは、まず組織管理者としてサインインし、組織内で IAM Identity Center を有効にします。次に、組織内のメンバーアカウントに Amazon Managed Grafana ワークスペースを作成します。

おなたが組織の管理者でない場合、Organizations の管理者に連絡して、IAM Identity Center を有効にするように依頼する必要があります。IAM Identity Center が有効になったら、メンバーアカウントにワークスペースを作成できます。

メンバーアカウントにワークスペースを作成し、そのワークスペースから組織内の他の AWS アカウントのリソースにアクセスできるようにするには、ワークスペースでカスタマー管理アクセス許可を使用する必要があります。詳細については、「「カスタマー管理」のアクセス許可」を参照してください。

メンバーアカウントでワークスペースを作成するには、AWSGrafanaAccountAdministrator、AWSSSOMemberAccountAdministrator、およびAWSSSODirectoryAdministrator ポリシーを持つアカウントでサインインする必要があります。詳細については、「IAM Identity Center を使用するメンバーアカウントの Grafana 管理者」を参照してください。

ビルトイン データソースに接続する

すべての Amazon Managed Grafana ワークスペースで、次のデータソースがサポートされています。

バージョン 9 以降をサポートするワークスペースでは、これらのデータソースの一部がデフォルトでインストールされない場合があります。コアデータソースはすべてのインストールで使用できますが、コアセットに含まれていないデータソースには、対応する Grafana プラグインのインストールが必要です。リストにないデータソースを追加で使用する場合は、プラグインをインストールすることで対応できます。プラグインの管理の詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

トピック

- アラートマネージャーデータソースに接続する
- Amazon CloudWatch データソースへの接続
- Amazon OpenSearch Service データソースへの接続
- AWS IoT SiteWise データソースに接続する
- an AWS IoT TwinMaker データソースに接続する
- Amazon Managed Service for Prometheus およびオープンソースの Prometheus データソースへの接続
- Amazon Timestream データソースへの接続
- Amazon Athena データソースに接続する
- Amazon Redshift データソースに接続する
- AWS X-Ray データソースに接続する
- Azure Monitor データソースへの接続
- Graphite データソースに接続する
- Google Cloud Monitoring データソースへの接続
- InfluxDB ソースデータベースに接続するには
- Jaeger データソースへの接続
- Loki データソースに接続するには
- Microsoft SQL Server データソースに接続する
- MySQL データソースに接続する
- OpenSearch データソースに接続する
- OpenTSDB データソースに接続する
- PostgreSQL データソースに接続する
- Tempo データソースへの接続
- テスト用 TestData データソースの設定
- Zipkin データソースへの接続

アラートマネージャーデータソースに接続する

Grafana にはビルトインで Prometheus アラートマネージャーのサポートが含まれています。Grafana アラートの設定完了後、Grafana アラート UI を使用して、サイレンス、連絡先、通知ポリシーを管理できます。これらのページのドロップダウンから、Grafana と設定済みのアラートマネージャーデータソースのどちらを使用するかを切り替えることができます。

アラートマネージャーの実装

アラートマネージャーの Prometheus、Cortex、および Grafana Mimir 実装がサポートされています。実装は、データソース設定ページで指定できます。Prometheus のコンタクトポイント (通知先) と通知ポリシーは、HTTP API を介した設定の更新をサポートしていないため、Grafana アラート UI では読み取り専用になります。

アラートマネージャーデータソースの設定

Grafana アラートで使用するアラートマネージャーデータソースを設定できます。

前提条件

アラートマネージャーを設定するには、次の前提条件を満たしている必要があります。

- Prometheus インスタンスに取り込まれたメトリクスがあり、少なくとも 1 つのアラートまたは記録ルールが設定されている。ワークスペースの URL が必要です。
- Amazon Managed Grafana に、アラートマネージャ実装からのアラート、アラートグループ、サイレンス、およびコンタクトポイントの読み取り権限が定義された権限が付与されている。

アラートマネージャーデータソースの設定方法

1. Grafana コンソールの Grafana メニューの [設定] にある [データソース] ページを選択します。
2. [データソースの追加] を選択し、データソースタイプのリストから [アラートマネージャー] を選択します。
3. 新規データソースに次の情報を入力します:
   - 名前: データソースの名前を入力します。
   - 実装: Prometheus 、Mimir、Cortex のいずれかのアラートマネージャー実装を選択します。
   - HTTP: URL に、アラートマネージャーの URL を指定します。Prometheus の場合、この URL は ワークスペース URL で、 `alertmanager` が追加されます。例えば、`https://myprometheus/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager`。
   - 認証: アラートマネージャー実装にアクセスするために必要な認証の詳細を設定します。
4. [保存とテスト] を選択して、データソースのセットアップを完了します。

   データソースが正しく設定されている場合、ヘルスチェックに合格したことを示すメッセージが表示されます。

Amazon CloudWatch データソースへの接続

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース設定オプションを使用してAmazon CloudWatch を AWS データソースとして追加できます。この機能は、既存の CloudWatch アカウントを検出して、CloudWatch をデータソースとして追加する作業を簡素化し、CloudWatch へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定すると、データソースとして CloudWatch を追加できます。または、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

トピック

- AWS データソース設定を使用して CloudWatch をデータソースとして追加する
- CloudWatch をデータソースとして手動で追加する
- クエリエディタの使用
- 精選されたダッシュボード
- テンプレート化されたクエリ
- ec2_instance_attribute の使用例
- JSON 形式のテンプレート変数の使用
- 料金
- Service Quotas
- クロス アカウント オブザーバビリティ

AWS データソース設定を使用して CloudWatch をデータソースとして追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の CloudWatch リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして CloudWatch を追加します。

AWS データソース設定を使用して CloudWatch をデータソースとして追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ナビゲーションペインで、メニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。

4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
5. [データソース] タブを選択します。
6. Amazon CloudWatch のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
7. 再度[データソース] タブを選択します。
8. Amazon CloudWatch 行で Grafana で設定 を選択します。
9. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
10. Grafana ワークスペースコンソールのナビゲーションバーで AWS アイコンを選択し、AWS [サービス]、[CloudWatch] を選択します。
11. CloudWatch データソースがクエリを実行するデフォルトのリージョンを選択します。
12. 目的のアカウントを選択し、[データソースを追加]を選択します。

CloudWatch をデータソースとして手動で追加する

CloudWatch データソースの手動での追加方法

1. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンにカーソルを合わせてデータソースを選択します。
2. [データソースの追加] を選択します。
3. CloudWatch データソースを選択します。必要に応じて、検索ボックスに **`CloudWatch`** の入力を開始すると検索を補助することができます。

CloudWatch の設定

次の CloudWatch 設定が適用されます。

名前	説明
Name	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
Default	新しいパネルで事前に選択されるデータソースを指定します。
Default Region	クエリエディタでリージョンを設定します。クエリごとに変更できます。
Namespaces of Custom Metrics	カスタムメトリクスの CloudWatch 名前空間を指定します。名前空間が複数ある場合、カンマで区切って指定します。
Auth Provider	認証情報を取得するプロバイダーを指定します。
Assume Role Arn	引き受けるロールの Amazon リソースネーム (ARN) を指定します。
External ID	(オプション) 外部 ID を指定します。外部 ID で AWS アカウント 作成された別の でロールを引き受ける場合は、 を使用します。
Timeout	CloudWatch Logs クエリ専用のタイムアウトを設定します。
X-Ray trace links	ログに @xrayTraceId フィールドが含まれている場合にログ内にリンクを自動的に追加するには、データソース設定の [X-Ray トレースリンク] セクションで X-Ray データソースをリンクします。X-Ray データソースが既に設定されている必要があります。

認証

Amazon Managed Grafana と CloudWatch 間の認証を有効にする場合、Amazon Managed Grafana コンソールを使用することで必要なポリシーとアクセス許可を素早く作成できます。または、セルフマネージド Grafana サーバーで実行する際と同じ方法の一部を使用して認証を手動で設定することもできます。

Amazon Managed Grafana データソース設定を使用して素早くポリシーを設定する場合、「AWS データソース設定を使用して CloudWatch をデータソースとして追加する」のステップに従います。

アクセス許可を手動で設定する場合、次のセクションに記載のいずれかの方法を使用します。

AWS 認証情報

認証には 3 つの方法があります。

- AWS SDK デフォルト — ワークスペースにアタッチされているロールで定義されたアクセス許可を使用します。詳細については、「「カスタマー管理」のアクセス許可」を参照してください。
- アクセスキーとシークレットキー — AWS SDK for Go `StaticProvider` に対応します。指定されたアクセスキー ID とシークレットキーを使用して認証します。この方法にはフォールバックがないため、指定されたキーのペアが機能しない場合は失敗します。

IAM ロール

現在、CloudWatch へのすべてのアクセスは、公式 AWS SDK を使用して Grafana バックエンドによってサーバー側で行われます。AWS SDK デフォルト認証方法を選択し、Grafana サーバーが実行されている場合は AWS、IAM ロールを使用して認証を自動的に処理できます。

詳細については、「IAM ロール」を参照してください。

IAM ポリシー

Grafana が CloudWatch メトリクスや EC2 のタグ、インスタンス、リージョンを読み取るためには、IAM での権限付与が必要です。これらのアクセス許可を IAM ロールに付与し、Grafana の組み込み機能を使用してロールを引き受けることができます。

次のコードは、最小限のポリシーのコードの例です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowReadingMetricsFromCloudWatch",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
```

```
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:GetMetricData",
        "cloudwatch:GetInsightRuleReport"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowReadingLogsFromCloudWatch",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups",
        "logs:GetLogGroupFields",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:GetQueryResults",
        "logs:GetLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowReadingTagsInstancesRegionsFromEC2",
      "Effect": "Allow",
      "Action": ["ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeRegions"],
      "Resource": "*"
    },
    {
      "Sid": "AllowReadingResourcesForTags",
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Sid": "AllowReadingAcrossAccounts",
      "Effect": "Allow",
      "Action": [
        "oam:ListSinks",
        "oam:ListAttachedLinks"
      ],
      "Resource": "*"
    }
  ]
}
```

ロールを割り当てる

引き受ける IAM ロールがある場合、この `Assume Role ARN` フィールドに指定します。このフィールドを空白のままにした場合、提供された認証情報が直接使用されます。また、関連するロールまたはユーザーに必要なアクセス許可が必要です。このフィールドが空白でない場合、提供された認証情報を使用して `sts:AssumeRole` 呼び出しが実行されます。

クエリエディタの使用

Amazon Managed Grafana の CloudWatch データソースは、CloudWatch にデータを送信するさまざまな AWS サービスからメトリクスとログを取得して分析できる強力なクエリエディタを提供します。クエリエディタは、メトリクス検索と CloudWatch Logs の 2 つの異なるクエリモードをサポートしています。

メトリクス用のクエリモードは、CloudWatch API を使用して CloudWatch にアップロードされたメトリクスを検索します。ログ用のモードでは、CloudWatch Logs APIs を使用してログレコードを検索します。各モードには、専用のクエリエディタがあります。エディタの上部にあるクエリモードスイッチを使用して、クエリを実行する API を選択します。

トピック

- メトリクスクエリエディタの使用
- Amazon CloudWatch Logs クエリエディタの使用

メトリクスクエリエディタの使用

メトリクスクエリエディタでは、「メトリクス検索」と「メトリクスクエリ」の 2 種類のクエリを構築できます。[メトリクスクエリ] は、データのクエリに CloudWatch Metrics Insights を使用します。

一般的なクエリエディタのフィールド構成

次の 3 つのフィールドは「メトリクス検索」モードと「メトリクスクエリ」モードの両方に共通して存在します。

共通フィールド

ID

`GetMetricData` API では、すべてのクエリに一意の ID が必要です。このフィールドに ID を指定します。ID には、数字、文字、アンダースコアを使用できます (小文字で始める必要があ

ります)。ID が指定されていない場合、Amazon Managed Grafana は `query[refId of the current query row]` のパターンの ID を生成します。例えば、queryA はパネルエディタの最初のクエリ行を表します。

期間

期間とは、特定の CloudWatch 統計に関連付けられた時間長です。期間は秒単位で定義されます。値には 1、5、10、30、の他に 60 の倍数を使用できます。期間フィールドを空白のままにするか、auto に設定すると、時間範囲と CloudWatch 保持期間ポリシーを基に自動的に計算されます。使用される式は `time range in seconds / 2000` で、保持期間を基に期間を除外した後、事前に定められた期間 [60、300、900、3600、21600、86400] の中から次に大きな値に移動します。Amazon Managed Grafana が使用している期間を確認するには、クエリエディタで [クエリプレビューを表示] を選択します。

エイリアス

次のエイリアスパターンが適用されます。

エイリアスパターン	説明	結果の例
`{{region}}`	リージョンを返します。	`us-east-1`
`{{period}}`	期間を返します。	`3000`
`{{metric}}`	メトリクスを返します。	`CPUUtilization`
`{{label}}`	API オペレーションによって返されたラベルを返します (メトリクス検索のみ)。	`i-01343`
`{{namespace}}`	名前空間を返します (メトリクス検索のみ)。	`AWS/EC2`
`{{stat}}`	統計を返します (メトリクス検索のみ)。	`Average`
`{{[dimension name]}}`	ディメンション名を返します (メトリクス検索のみ)。	`i-01343`

メトリクス検索オプションの使用

メトリクス検索で有効なクエリを作成するには、名前空間、メトリクス名、および少なくとも 1 つの統計を指定する必要があります。[完全一致] が有効になっている場合、クエリするメトリクスのすべてのディメンションも指定する必要があります。メトリクススキーマは完全に一致する必要があります。詳細については、「CloudWatch 検索式構文」を参照してください。

[完全一致] がオフになっている場合、フィルタリングするディメンションの指定数に制限はありません。フィルター条件に一致するメトリクスは最大 100 個まで返されます。

ディメンションワイルドカード文字を使用した動的クエリ

1 つ以上のディメンション値にアスタリスク (*) ワイルドカード文字を使用して、メトリクスの動的リストをモニタリングできます。

これにより、EC2 インスタンスやコンテナなどの AWS リソースのメトリクスをモニタリングできます。立ち上げ、Auto Scaling イベントの一部として新しいインスタンスが作成されると、グラフに自動で表示されるため、新しいインスタンス ID を追跡する必要はありません。この機能によるメトリクスの取得は現在、最大 100 個に制限されています。[クエリプレビューを表示] を選択すると、ワイルドカード文字をサポートするために自動的に構築された検索式を確認することができます。

デフォルトでは、検索式は、クエリ対象のメトリクスが定義されたディメンション名と正確に一致するように定義されています。つまり、この例では、名前が `InstanceId` のディメンションが 1 つだけのメトリクスのみが返されます。

他のディメンションが定義されているメトリクスも含めるには、[完全一致] をオフにします。[完全一致] をオフにすると、ワイルドカード文字を使用しなくても検索式が作成されます。Grafana は、少なくとも名前空間、メトリクス名、および定義されたすべてのディメンションに一致するメトリクスを検索します。

複数値のテンプレート変数

複数値のテンプレート変数からディメンション値を定義する場合、検索式を使用して一致するメトリクスをクエリします。これにより、1 つのクエリで複数のテンプレート変数を使用できます。[完全一致] オプションがオフになっているクエリには、テンプレート変数を使用することもできます。

検索式は現在 1024 文字に制限されているため、値のリストが長い場合、クエリが失敗することがあります。特定のディメンション名の値を持つすべてのメトリクスをクエリする場合は、`All` オプションの代わりにアスタリスク (*) ワイルドカード文字を使用することをお勧めします。

複数値のテンプレート変数の使用は、ディメンション値でのみサポートされます。Region、Namespace、または Metric Name の複数値テンプレート変数の使用はサポートされていません。

メトリクス数式を使用する

CloudWatch のメトリクスに対して数学的な関数を使って操作を行うことで、新しい時系列メトリクスを作成できます。算術演算子、単項減算、その他の関数がサポートされており、CloudWatch メトリクスに適用することができます。メトリクス数式の詳細については、「メトリクス数式の使用」を参照してください。

例えば、メトリクスに算術演算を適用するには、未加工のメトリクスに ID (一意の文字列) を指定します。その後、この ID を使用して、新しいメトリクスの Expression フィールドで算術演算を適用できます。

Expression フィールドを使用して queryA * 2 などの別のクエリを参照する場合、そのクエリを基にアラートルールを作成することはできません。

期間

期間とは、特定の Amazon CloudWatch 統計に関連付けられた時間長です。期間は秒単位で定義されます。値には 1、5、10、30、の他に 60 の倍数を使用できます。

期間フィールドを空白のままにするか、[自動] に設定すると、時間範囲を基に自動的に計算されます。使用される式は time range in seconds / 2000 で、事前定義された期間の配列 [60, 300, 900, 3600, 21600, 86400] 内の次に高い値に進みます。Amazon Managed Grafana が使用している期間を確認するには、クエリエディタで [クエリプレビューを表示] を選択します。

Grafana パネルから CloudWatch コンソールへのディープリンク

パネルで時系列を選択するとコンテキストメニューが表示され、その中に [CloudWatch コンソールで表示] のリンクが表示されます。リンクを選択すると新しいタブが開き、そのクエリのすべてのメトリクスが表示された CloudWatch コンソールに移動します。リンククリック時に CloudWatch コンソールにサインインしていない場合、サインインページに転送されます。提供されたリンクは に有効です AWS アカウント が、Grafana で選択したデータソース AWS アカウント に対応する にサインインしている場合にのみ、正しいメトリクスが表示されます。

この機能は、メトリクス数式に基づくメトリクスでは使用できません。

メトリクスクエリを使用して CloudWatch Metrics Insights データをクエリする

> **Note**
>
> Amazon CloudWatch Metrics Insights はプレビュー中です。CloudWatch Metrics Insights の機能は、すべての AWS アカウントが利用できます。機能は変更される可能性があります。

CloudWatch Metrics Insights のデータは、[メトリクスクエリエディタ]で `metric query` モードを選択することでクエリできます。

CloudWatch Metrics Insights は、高性能な SQL クエリエンジンであり、これにより、メトリクスに対し大量のクエリを実行できます。これは、すべての CloudWatch メトリクス内の傾向とパターンをリアルタイムで特定するために使用できる、高速で柔軟な SQL ベースのクエリエンジンです。SQL の方言を使用します。Metrics Insights のクエリ構文の詳細については、「クエリ構文とキーワード」を参照してください。

クエリ構文とキーワード

CloudWatch Metrics Insights では SQL の方言を使用しています。以下にクエリの構文例を示します。

```
SELECT FUNCTION(metricName)
FROM namespace | [ SCHEMA(namespace[, labelKey [, ...] ]) ]
     [ WHERE labelKey OPERATOR labelValue [AND|OR|([...])*] [, ...] ]
[ GROUP BY labelKey [, ...]]
[ ORDER BY FUNCTION() [DESC | ASC] ]
[ LIMIT number]
```

キーワードでは大文字と小文字は区別されませんが、識別子では大文字と小文字は区別されます。識別子には、メトリクスの名前、名前空間、ディメンションが含まれます。

次の表に、クエリのキーワードとその説明を示します。

キーワード	説明
FUNCTI(	必須。使用する集計関数とクエリするメトリクスの名前を指定します。有効な値は、AVG、COUNT、MAX、MIN、SUM です。

キーワード	説明
MetricName	必須。例えば、CPUUtilization と指定します。
FROM	必須。メトリクスのソースを指定します。クエリするメトリクスを含むメトリクス名前空間、または、SCHEMA テーブル関数のどちらかを指定します。名前空間の例は、AWS/EC2、AWS/Lambda などです。
SCHEMA	(任意) クエリ結果をフィルタリングして、完全に一致するメトリクス、または一致しないメトリクスのみを表示します。
WHERE	(任意) 結果をフィルタリングして、指定した式に一致するメトリクスのみを表示します。例えば、WHERE InstanceType != 'c3.4xlarge' と指定します。
GROUP BY	(任意) クエリ結果を複数の時系列にグループ化します。例えば、GROUP BY ServiceName と指定します。
ORDER BY	(任意) 返される時系列の順序を指定します。ASC と DESC を使用できます。
LIMIT	(任意) 返される時系列の数を制限します。

次に例をいくつか示します。

- ```
  SELECT AVG(CPUUtilization) FROM "AWS/EC2"
  ```

  ディメンションを無視して AWS/EC2 名前空間内のすべての CPUUtilization メトリクスを照合し、単一の集計された時系列を返します。

- ```
  SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2")
  ```

  ディメンションが定義されていない AWS/EC2 名前空間内の CPUUtilization メトリクスのみを一致させます。

- ```
  SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2", InstanceId)
  ```
  ```

CloudWatch に報告された 1 つのディメンション (InstanceId) を持つ CPUUtilization メトリクスのみを一致させます。

- ```
  SELECT SUM(RequestCount) FROM SCHEMA("AWS/ApplicationELB", LoadBalancer,
  AvailabilityZone)
  ```

  CloudWatch に AWS/ApplicationELB から報告された 2 つのディメンション (LoadBalancer と AvailabilityZone) を持つ RequestCount メトリクスのみを一致させます。

ラベル値は一重引用符で囲む必要があります。

エスケープ文字

クエリで指定するラベル値は、常に一重引用符で囲む必要があります。 例えば、SELECT MAX(CPUUtilization) FROM "AWS/EC2" WHERE AutoScalingGroupName = 'my-production-fleet' と指定します。

文字、数字、アンダースコア (_) 以外の文字を含むメトリクスの名前空間、メトリクス名、ラベルキーは、二重引用符で囲む必要があります。例えば、SELECT MAX("My.Metric") と指定します。これらのいずれかに Bytes"Input" のように二重引用符自体が含まれている場合は、SELECT AVG("Bytes\"Input\"") のように、その二重引用符をバックスラッシュで工スケープする必要があります。メトリクス名前空間、メトリクス名、またはラベルキーに、Metrics Insights での予約キーワードが含まれている場合は、これらも二重引用符で囲む必要があります。例えば、LIMIT という名前のメトリクスを指定するのであれば、SELECT AVG("LIMIT") のようになります。また、予約キーワードを含まない場合に、名前空間、メトリクス名、またはラベルを二重引用符で囲んだとしても、エラーとはなりません。

ビルダーモードとコードモード

クエリを作成するモードには、Builder モードと Code モードがあります。

Builder モードでのクエリ作成方法

1. 前の表の情報を参考に、メトリクスの名前空間、メトリクス名、フィルター、グループ、順序オプションを参照して選択します。
2. これらの各オプションについて、利用可能な選択肢のリストから選択します。
  ```

Code モードでのクエリ作成方法

1. コードエディタにクエリを記述します。
2. クエリを実行するには、コードエディタで [クエリを実行する] を選択します。

builder モードでのクエリ作成方法:

- 前の表の情報を参考に、メトリクスの名前空間、メトリクス名、フィルター、グループ、順序オプションを参照して選択します。
- これらの各オプションについて、利用可能な選択肢のリストから選択します。

Grafana によって、選択した内容を基に SQL クエリが自動的に構築されます。

code モードでのクエリ作成方法:

- コードエディタにクエリを記述します。
- クエリを実行するには、コードエディタで [クエリを実行する] を選択します。

コードエディタには、キーワード、集計、名前空間、メトリクス、ラベル、ラベル値に対して候補を提示する自動補完機能が組み込まれています。スペース、カンマ、またはドル記号を入力すると、候補が表示されます。キーボードの組み合わせ (CTRL+Space) を使用することもできます。

コードエディタはクエリを自動補完します。ただし、コードエディタでテンプレート変数を使用した場合、自動補完が妨げられる可能性があります。

CloudWatch Metrics Insights の例

> **Note**
>
> CloudWatch Metrics Insights はオープンプレビュー中です。プレビューはすべての AWS アカウントで開かれており、アクセスをリクエストする必要はありません。一般提供を発表する前に、機能が追加または変更されることがあります。

このセクションでは、コピーして直接使用することも、クエリエディタで編集して使用することもできる、有用な CloudWatch Metrics Insights クエリの例を紹介します。これらの例の内のいくつかは、コンソールで既に利用可能な状態であり、[メトリクス] ビューで [クエリを追加] をクリックしてアクセスすることができます。

EC2 の例

インスタンスあたりの CPU 使用率のメトリクスを表示

```
SELECT AVG(CPUUtilization)
FROM "AWS/EC2"
GROUP BY InstanceId
```

フリート全体での平均 CPU 使用率を表示

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

平均 CPU 使用率が最も高い 10 個のインスタンスを表示

```
SELECT MAX(CPUUtilization)
FROM "AWS/EC2"
GROUP BY InstanceId
LIMIT 10
```

CPU 使用率が最も高い 10 個のインスタンスを降順で表示

```
SELECT AVG(CPUUtilization)
FROM "AWS/EC2"
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

このケースでは、CloudWatch エージェントはアプリケーションごとに CPU 使用率のメトリクスを収集しています。このクエリは、特定のアプリケーション名について、このメトリクスの平均によるフィルタリングを行います。

```
SELECT AVG(CPUUtilization)
```

```
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
SELECT AVG(ConcurrentExecutions)
FROM "AWS/Lambda"
```

Lambda 関数の平均実行時間上位 10 個を降順で表示

```
SELECT AVG(Duration)
FROM "AWS/Lambda"
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

Lambda 関数の実行時間の最大値、平均値、最小値を表示

```
SELECT MAX(Duration)
FROM "AWS/Lambda"
```

Application Load Balancer での例

ディメンション LoadBalancer と AvailabilityZone を持つメトリクスを表示

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

アクティブな同時 TCP 接続の数を含むメトリクスを表示

```
SELECT AVG(ActiveConnectionCount)
FROM "AWS/ApplicationELB"
```

Amazon EBS の例

ボリュームあたりの平均書き込みバイト数の上位 10 個を降順で表示

```
SELECT AVG(VolumeWriteBytes)
FROM "AWS/EBS"
GROUP BY VolumeId
ORDER BY MAX() DESC
```

```
LIMIT 10
```

Amazon EBS ボリュームでの平均書き込み時間を表示

```
SELECT AVG(VolumeTotalWriteTime)
FROM "AWS/EBS"
```

Amazon EBS ボリュームでの平均アイドル時間を表示

```
SELECT AVG(VolumeIdleTime)
FROM "AWS/EBS"
View average burst balance per volume
SELECT AVG(BurstBalance)
FROM "AWS/EBS"
GROUP BY VolumeId
View average read bytes across Amazon EBS volumes
SELECT AVG(VolumeReadBytes)
FROM "AWS/EBS"
```

Amazon EBS ボリュームでの平均書き込みバイト数を表示

```
SELECT AVG(VolumeWriteBytes)
FROM "AWS/EBS"
```

Amazon Simple Storage Service での例

平均レイテンシーグループ (バケット名別) を表示

```

SELECT AVG(TotalRequestLatency)
FROM "AWS/S3"
GROUP BY BucketName
```

すべての Amazon S3 バケットのバケットあたりのオブジェクトの平均数を表示

```

SELECT AVG(NumberOfObjects)
FROM "AWS/S3"
GROUP BY BucketName
```

Amazon Simple Notification Service での例

Amazon-Simple-Notification-Service-での例

```
SELECT AVG(NumberOfMessagesPublished)
FROM "AWS/SNS"
```

失敗したメッセージの平均数 (トピック名別) を表示

```
SELECT AVG(NumberOfNotificationsFailed)
FROM "AWS/SNS"
GROUP BY TopicName
```

AWS API の使用例

アカウントの呼び出し数で上位 20 AWS APIs を表示する

```
SELECT COUNT(CallCount)
FROM "AWS/Usage"
WHERE "Type" = 'API'
GROUP BY "Service", "Resource"
ORDER BY SUM() DESC
LIMIT 20
```

CloudWatch Metrics Insights の制限

CloudWatch Metrics Insights には現在、次の制限があります。

- 直近 3 時間分のデータにのみクエリを実行できます。
- 1 つのクエリで処理できるメトリクスは 10,000 個以下です。つまり、SELECT 句、FROM 句、および WHERE 句が 10,000 を超えるメトリクスと一致した場合、見つかったこれらのメトリクスのうち最初の 10,000 のみがクエリによって処理されます。
- 1 つのクエリで 500 を超える時系列を返すことはできません。これは、クエリが 500 個を超えるメトリクスを処理している場合でも、クエリ結果にすべてのメトリクスが返されるわけではないことを意味します。ORDER BY 句を使用すると、処理されているすべてのメトリクスがソートされます。その中から (ORDER BY 句に応じ) 最大値または最小値を持つ 500 個が返されます。ORDER BY 句を含めない場合は、一致したメトリクスの中から、どの 500 個が選択され返されるのかを制御することはできません。

- 個別の GetMetricData オペレーションに含めることができるクエリは 1 つだけですが、ダッシュボードには、それぞれにクエリを 1 つ含んだ複数のウィジェットを用意することができます。

Amazon CloudWatch Logs クエリエディタの使用

CloudWatch Logs をクエリするには、クエリするリージョンと最大 20 個のロググループを選択します。メイン入力エリアにクエリを記述します。詳細については、「CloudWatch Logs Insights クエリ構文」を参照してください。

CloudWatch Logs Insights の stats コマンドを使用して、時系列データを返すクエリを記述することもできます。Explore で stats クエリを実行するときは、Metrics Explore モードになっていることを確認してください。

クエリ入力フィールドの右側には CloudWatch Logs Insights のリンクがあり、これをクリックすると、クエリが記述された CloudWatch Logs Insights コンソールが表示されます。必要に応じて、そこで探索を続けることができます。

テンプレート変数の使用

他のいくつかのデータソースと同様に、CloudWatch データソースはクエリでのテンプレート変数の使用をサポートしています。詳細については、「テンプレートおよび変数」を参照してください。

Grafana パネルから CloudWatch Logs コンソールへのディープリンク

CloudWatch Logs Insights コンソールでクエリを表示するには、クエリエディタの横にある [CloudWatch Logs Insights] ボタンを選択します。リンククリック時に CloudWatch コンソールにサインインしていない場合、サインインページに転送されます。提供されたリンクは に有効です AWS アカウント が、Grafana で選択したデータソース AWS アカウント に対応する にサインインしている場合にのみ、正しいメトリクスが表示されます。

[アラート]

CloudWatch Logs クエリは、例えば stats コマンドを使用すると数値データを返す可能性があるため、アラート機能に対応しています。詳細については、「Grafana アラート」を参照してください。

精選されたダッシュボード

更新された CloudWatch データソースには、最も人気のある 5 つの AWS サービス用に事前設定されたダッシュボードが付属しています。

- Amazon EC2

- Amazon Elastic Block Store
- AWS Lambda
- Amazon CloudWatch Logs
- Amazon Relational Database Service

事前設定済みのダッシュボードをインポートする方法: CloudWatch データソースの設定ページに移動し、[ダッシュボード]タブを選択し、使用するダッシュボードの [インポート] を選択します。ダッシュボードをカスタマイズする場合、カスタマイズするダッシュボードを別の名前で保存することをお勧めします。別の名前で保存しない場合、ダッシュボードの新しいバージョンがリリースされた際にダッシュボードが上書きされてしまいます。

テンプレート化されたクエリ

メトリクスクエリにサーバー、アプリケーション、センサー名などの詳細をハードコーディングする代わりに、変数を使用することができます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とさまざまな種類のテンプレート変数の概要については、「テンプレート」を参照してください。

クエリ変数

CloudWatch データソースでは、以下のクエリを変数編集ビューのクエリフィールドで指定することができます。これらを使用して、変数のオプションリストに region、namespaces、metric names、dimension keys/values などの項目を埋めることができます。

クエリのデータソースに設定されたデフォルトのリージョンを使用するように、region の代わりに、defaultを指定することができます。

名前	説明
`regions()`	すべての AWS リージョンのリストを返します。
`namespaces()`	CloudWatch がサポートする名前空間のリストを返します。
`metrics(namespace, [region])`	名前空間内のメトリクスのリストを返します。(カスタムメトリクスにリージョンを指定するか "default" を使用します。)

名前	説明
`dimension_keys(namespace)`	名前空間のディメンションキーのリストを返します。
`dimension_values(region, namespace, metric, dimension_key, [filters])`	指定された region、namespace 、metric または dimension_key に一致するディメンション値のリストを返します。または、ディメンション filters を使用してより具体的な結果を取得することもできます。
`ebs_volume_ids(region, instance_id)`	指定された region、instance_id に一致するボリューム ID のリストを返します。
`ec2_instance_attribute(region, attribute_name, filters)`	指定された region、attribute_name 、filters に一致する属性のリストを返します。
`resource_arns(region, resource_type, tags)`	指定された region、resource_type 、tags に一致する ARN のリストを返します。
`statistics()`	すべての標準統計のリストを返します。

CloudWatch が提供するメトリクスの詳細については、「CloudWatch メトリクスを発行するAWS サービス」を参照してください。

テンプレート化されたクエリの例

次の表は、個々の AWS サービスのリソースのリストを返すディメンションクエリの例を示しています。

クエリ	サービス
`dimension_values(us-east-1,AWS/ELB,RequestCount,LoadBalancerName)`	エラスティックロー

クエリ	サービス
	ドバランシング
`dimension_values(us-east-1,AWS/ElastiCache,CPUUtilization,CacheClusterId)`	Amazon ElastiCache
`dimension_values(us-east-1,AWS/Redshift,CPUUtilization,ClusterIdentifier)`	Amazon Redshift
`dimension_values(us-east-1,AWS/RDS,CPUUtilization,DBInstanceIdentifier)`	Amazon RDS
`dimension_values(us-east-1,AWS/S3,BucketSizeBytes,BucketName)`	Amazon Simple Storage Service (Amazon S3)
`dimension_values(us-east-1,CWAgent,disk_used_percent,device,{"InstanceId":"$instance_id"})`	CloudWatch エージェント
`resource_arns(eu-west-1,elasticloadbalancing:loadbalancer,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})`	エラスティックロードバランシング
`resource_arns(eu-west-1,ec2:instance,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})`	Amazon EC2

ec2_instance_attribute の使用例

JSON フィルター

ec2_instance_attribute クエリは filters をJSON 形式で受け取ります。ec2:DescribeInstances の事前定義フィルターを指定できます。実際のフィルタリングは AWS Grafana ではなく で行われます。

次の例は、フィルターの構文を示しています。

```
{ filter_name1: [ filter_value1 ], filter_name2: [ filter_value2 ] }
```

以下に、ec2_instance_attribute() クエリの例を示します。

```
ec2_instance_attribute(us - east - 1, InstanceId, { 'tag:Environment':
 ['production'] });
```

属性の選択

インスタンスごとに返される属性は 1 つだけです。選択可能な属性は、単一の値を持ち、オブジェクトや配列でないフラットな属性のみです。以下のフラット属性を使用できます。

- AmiLaunchIndex
- Architecture
- ClientToken
- EbsOptimized
- EnaSupport
- Hypervisor
- IamInstanceProfile
- ImageId
- InstanceId
- InstanceLifecycle
- InstanceType
- KernelId
- KeyName

- LaunchTime
- Platform
- PrivateDnsName
- PrivateIpAddress
- PublicDnsName
- PublicIpAddress
- RamdiskId
- RootDeviceName
- RootDeviceType
- SourceDestCheck
- SpotInstanceRequestId
- SriovNetSupport
- SubnetId
- VirtualizationType
- VpcId

タグは、タグ名のプレフィックスに Tags を付けることで選択できます。

以下に、ec2_instance_attribute() クエリの例を示します。

```
ec2_instance_attribute(us - east - 1, Tags.Name, { 'tag:Team': ['sysops'] });
```

JSON 形式のテンプレート変数の使用

一部のクエリは JSON 形式のフィルターを受け付けており、Grafana はテンプレート変数の JSON への変換をサポートしています。

env = 'production', 'staging' の場合、次のクエリは、Environment タグが production または staging である EC2 インスタンスの ARN を返します。

```
resource_arns(us-east-1, ec2:instance, {"Environment":${env:json}})
```

料金

Grafana の Amazon CloudWatch データソースは、`ListMetrics` および `GetMetricData` CloudWatch API コールを使用してメトリクスを一覧表示および取得します。CloudWatch Logs の料金は、CloudWatch Logs Insights クエリを通じて取得、アーカイブ、分析に利用されたデータ量を基に決定されます。詳細については、「Amazon CloudWatch 料金表」をご覧ください。

クエリエディタでディメンションを選択するたびに、Grafana から `ListMetrics` リクエストが発行されます。クエリエディタでクエリを変更するたびに、1 件の新しいリクエストが GetMetricData に発行されます。

データサンプルを取得する API リクエストは、 `GetMetricData` を使用します。これにより、CloudWatch メトリクス数式に対するサポートが向上します。また、ワイルドカード文字を使用したり [完全一致] オプションをオフにした場合の検索式の自動生成もサポートします。`GetMetricData` の使用には料金が発生します。詳細については、Amazon CloudWatch 料金表をご覧ください。

Service Quotas

AWS は、 AWS アカウントのリソース、オペレーション、および項目のクォータまたは制限を定義します。ダッシュボード内のクエリの数とダッシュボードにアクセスするユーザーの数によっては、さまざまな CloudWatch および CloudWatch Logs リソースの使用制限に達する場合があります。クォータはアカウントごとおよび AWS リージョンごとに定義されることに注意してください。複数のリージョンを使用している場合、または複数のアカウントに対してクエリを実行するために複数の CloudWatch データソースを設定している場合は、制限に達した各アカウントと各リージョンについてクォータの引き上げをリクエストする必要があります。

詳細については、「CloudWatch サービスのクォータ」を参照してください。

クロス アカウント オブザーバビリティ

> **Warning**
> この機能を使用するには、Grafana ワークスペースがバージョン 9 以降である必要があります。

CloudWatch プラグインを使用すると、複数のリージョンアカウント全体のアプリケーションの監視とトラブルシューティングを行うことができます。クロス アカウント オブザーバビリティを使用す

ると、アカウントの境界を気にすることなく、メトリクスとログをシームレスに検索、視覚化、分析できます。

クロス アカウント オブザーバビリティを有効にするには、まず CloudWatch で有効にし、その後プラグインを実行しているロール/ユーザーに適切な IAM アクションを追加します。Amazon Managed Grafana ワークスペースが VPC 内で実行されている場合は、インターネットアクセスをサポートする NAT ゲートウェイも必要です。

- この機能を有効にする方法については、Amazon CloudWatch ユーザーガイドの「CloudWatch クロス アカウントオブザーバビリティ」を参照してください。
- プラグインを実行しているロール/ユーザーに追加する適切な IAM アクションを次に示します。

```
{
"Sid":  "AllowReadingAcrossAccounts",
"Effect":  "Allow",
"Action": [
  "oam:ListSinks",
  "oam:ListAttachedLinks"
],
"Resource":  "*"
}
```

- CloudWatch データソースのクロス アカウント オブザーバビリティは、Amazon CloudWatch Observability Access Manager に依存しています。Observability Access Manager は VPC エンドポイントをサポートしていません。Amazon Managed Grafana ワークスペースが VPC 内で実行されている場合、ワークスペースがインターネット上の API を呼び出すことを可能にする NAT ゲートウェイも必要です。

> **Note**
>
> また、対象のアカウントの CloudWatch データを読み取るための IAM 権限も必要です。

Amazon OpenSearch Service データソースへの接続

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース設定オプションを使用して、Amazon OpenSearch Service を AWS データソースとして追加できます。このデータソースは、OpenSearch クラスターと従来の Elasticsearch クラスターを実行する、OpenSearch Service ドメインをサポートしています。

AWS データソース設定オプションは、既存の OpenSearch Service アカウントを検出することで、OpenSearch Service をデータソースとして簡単に追加し、OpenSearch へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定し、データソースとして OpenSearch Service を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

OpenSearch Service データソースはパイプ処理言語 (PPL) をサポートしています。PPL の詳細については、「パイプ処理言語を使用した Amazon OpenSearch Service データのクエリ」を参照してください。

OpenSearch Service データソースを使用すると、OpenSearch に保存されているログやメトリクスを視覚化するために、単純または複雑なさまざまな種類の OpenSearch クエリを実行できます。OpenSearch に保存されているログイベントを使用してグラフに注釈を付けることもできます。

トピック

- AWS データソース設定を使用して OpenSearch Service をデータソースとして追加する
- Amazon OpenSearch Service をデータソースとして手動で追加する
- OpenSearch Service の設定
- Amazon OpenSearch Service データソースを使用する
- Amazon OpenSearch Service Serverless
- トレースの補助機能

AWS データソース設定を使用して OpenSearch Service をデータソースとして追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の OpenSearch Service リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして OpenSearch Service を追加します。

AWS データソース設定を使用して OpenSearch Service をデータソースとして追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。
4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
5. [データソース] タブを選択します。次に、[Amazon OpenSearch Service] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする]を選択します。
6. [データソース] タブを再度選択し、Amazon OpenSearch Service 行で [Grafana で設定] を選択します。
7. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
8. Grafana ワークスペースコンソールの左側のナビゲーションバーで、 AWS アイコンを選択し、AWS サービス Amazon OpenSearch Service を選択します。
9. Amazon Managed Grafana で検索して OpenSearch Service リソースを検出するリージョン、追加するアカウントと OpenSearch Service ドメインの順に選択し、インデックス設定を設定し、[データソースの追加] を選択します。

Amazon OpenSearch Service をデータソースとして手動で追加する

Amazon OpenSearch Service データソースを手動で追加する方法

1. Grafana コンソールのサイドメニューで、 AWS アイコンを選択し、[データソース] を選択します。
2. [Amazon OpenSearch Service] データソースを選択します。必要に応じて、検索ボックスに **`OpenSearch`** の入力を開始すると検索を補助することができます。
3. データを検索するリージョンを選択します。
4. [データソースの追加] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

OpenSearch Service の設定

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Defa`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Url`	OpenSearch Service ドメインのエンドポイント。エンドポイントは https://search-my-domain.us-east-1.es.amazonaws.com. の形式になります。
`Acce`	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。ブラウザ = URL はブラウザからアクセス可能である必要があります。

アクセスモードは、データソースへのリクエストの処理方法を制御します。他に何も記述されていない場合は、サーバーが優先されます。

サーバーアクセスモード (デフォルト)

すべてのリクエストはブラウザから Grafana バックエンドまたはサーバーに行われ、リクエストはデータソースに転送されるため、クロスオリジンリソース共有 (CORS) 要件を回避できます。このアクセスモードを選択する場合、URL は Grafana バックエンドまたはサーバーからアクセス可能である必要があります。

ブラウザ (直接) アクセス

Amazon Managed Grafana はブラウザ直接アクセスをサポートしていません。

インデックス設定

ここでは、`time field` のデフォルトを指定し、OpenSearch インデックスの名前を指定できます。インデックス名またはワイルドカード文字に時間パターンを使用できます。

OpenSearch/Elasticsear バージョン

バージョンドロップダウンメニューで OpenSearch またはレガシー Elasticsearch バージョンを指定します。バージョンごとにクエリの構成方法が異なるため、バージョンは重要です。現在、Grafana は OpenSearch 1.0.x をサポートしています。サポートされている Elasticsearch のバージョンは、`2.0+`、`5.0+`、`5.6+`、`6.0+`、`7.0+` です。値 `5.6+` はバージョン 5.6 以降、6.0 未満を意味します。値 `6.0+` はバージョン 6.0 以降、7.0 未満を意味します。最後に、`7.0+` はバージョン 7.0 以降、8.0 未満を意味します。

最小時間間隔

時間間隔による自動グループの下限。データを 1 分ごとに書き込む場合は `1m` など、書き込み頻度に設定することをお勧めします。このオプションは、[Data Source] (データソース) オプションのダッシュボードパネルで上書き/設定することもできます。この値は、`1m` (1 分) や `30s` (30 秒) など、有効な時間識別子が続く数値としてフォーマットする必要があります。以下の時間識別子がサポートされています。

識別子	説明
`y`	年
`M`	月
`w`	週

識別子	説明
d	日
h	時間
m	分
s	秒
ms	ミリ秒

ログ

2 つのパラメータである `Message field name` と `Level field name` は、オプションで、Explore のログを視覚化するときにログメッセージとログレベルに使用されるフィールドを決定するデータソース設定ページから設定できます。

例えば、OpenSearch Service へのログ配送に Filebeat のデフォルト設定を使用する場合、次の設定が機能します。

- メッセージフィールド名: メッセージ
- レベルフィールド名: fields.level

データリンク

データリンクは、Explore のログビューでアクセスできる指定されたフィールドからリンクを作成します。

各データリンク設定は、以下で構成されます。

- フィールド – データリンクで使用されるフィールドの名前。
- URL/クエリ – リンクが外部である場合は、完全なリンク URL を入力します。リンクが内部リンクの場合、この入力はターゲットデータソースのクエリとして機能します。どちらの場合も、`${__value.raw }` マクロを使用してフィールドから値を補間できます。
- 内部リンク – リンクが内部または外部である場合に選択します。リンクが内部の場合、データソースセレクタを使用すると、ターゲットデータソースを選択できます。データソースのトレースのみがサポートされています。

Amazon OpenSearch Service データソースを使用する

メトリクスクエリエディタ

OpenSearch クエリエディタを使用すると、複数のメトリクスを選択し、複数の用語またはフィルターでグループ化できます。右側のプラスアイコンとマイナスアイコンを使用して、メトリクスまたはグループを句で追加/削除します。一部のメトリクスと group by 句にはオプションがあります。オプションテキストを選択して行を展開し、オプション別にメトリクスまたはグループを表示および編集します。

パイプ処理言語 (PPL) の使用

Amazon OpenSearch Service データソースは、パイプ処理言語 (PPL) をサポートしています。PPL を使用すると、OpenSearch のクエリと視覚的表現を簡素化しながら強力に機能させることができます。PPL を使用することで、データを探索するために、長い OpenSearch Domain Specific Language (DSL) 文を作成したり、JSON オブジェクトを使用したクエリを記述する必要が無くなります。PPL では UNIX でのパイプのように、クエリをパイプで区切られた一連のコマンドとして記述できます。

以下のサンプル DSL クエリを例に使用します。

```
GET opensearch_sample_data_logs/_search{"from":0,"size":0,"timeout":"1m","query":
{"bool":{"should":[{"term":{"response.keyword":{"value":"404","boost":1}}},{"term":
{"response.keyword":
{"value":"503","boost":1}}}],"adjust_pure_negative":true,"boost":1}},"sort":
[{"_doc":{"order":"asc"}}],"aggregations":{"composite_buckets":{"composite":
{"size":1000,"sources":[{"host":{"terms":
{"field":"host.keyword","missing_bucket":true,"order":"asc"}}},{"response":{"terms":
{"field":"response.keyword","missing_bucket":true,"order":"asc"}}}]},"aggregations":
{"request_count":{"value_count":{"field":"request.keyword"}},"sales_bucket_sort":
{"bucket_sort":{"sort":[{"request_count":{"order":"desc"}}],"size":10}}}}}}>
```

上記の DSL クエリは、簡潔で人間が読み取れる次の PPL コマンドに置き換えることができます。

```
source = opensearch_sample_data_logs | where response='404' or response='503' | stats
 count(request) as request_count by host, response | sort -request_count
```

PPL の詳細については、「パイプ処理言語を使用した Amazon OpenSearch Service データのクエリ」を参照してください。

シリーズの名前付けとエイリアスパターン

Alias 入力フィールドを使用して時系列の名前を制御できます。

パターン	説明
{{term fieldname}}	Group By という用語の値に置き換え。
{{metric}}	メトリクス名に置き換え (例：平均、最小、最大) 。
{{field}}	メトリクスフィールド名に置き換え。

パイプラインメトリクス

例えば、移動平均や派生など、一部のメトリクス集約はパイプライン集約と呼ばれます。OpenSearch パイプラインメトリクスは、別のメトリクスに基づく必要があります。メトリクスの横にある目のアイコンを使用して、メトリクスがグラフに表示されないようにします。これは、パイプラインメトリクスで使用するクエリにのみあるメトリクスに役立ちます。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

OpenSearch Service データソースは、[クエリ]変数の[クエリ]フィールドで使用できる 2 種類のクエリをサポートしています。クエリは、カスタム JSON 文字列を使用して記述します。

クエリ	説明
{"find": "fields", "type": "keyword"}	インデックスタイプ keyword のフィールド名のリストを返します。

クエリ	説明
`{"find": "terms", "field": "@hostname", "size": 1000}`	用語集約を使用してフィールドの値のリストを返します。クエリは、クエリの時間範囲として現在のダッシュボードの時間範囲を使用します。
`{"find": "terms", "field": "@hostname", "query": '<lucene query>'}`	用語集約と指定された Lucene クエリフィルターを使用して、フィールドの値のリストを返します。クエリは、クエリの時間範囲として現在のダッシュボードの時間範囲を使用します。

用語クエリにはデフォルトのサイズ制限 500 があります。カスタム制限を設定するには、クエリのサイズプロパティを設定します。クエリでは、その他の変数を使用できます。次のコード例は、$host という名前の変数のクエリ定義を示しています。

```
{"find": "terms", "field": "@hostname", "query": "@source:$source"}
```

前の例では、クエリ定義内で $source という名前の別の変数を使用します。$source 変数の現在の値をドロップダウンリストで変更すると、$host 変数の更新が開始されます。更新後、$host 変数にはホスト名のみが含まれます。この場合、@source ドキュメントプロパティによってフィルタリングされます。

デフォルトでは、これらのクエリは用語の順序で返されます (その後、任意の変数と同様にアルファベット順または数値順にソートできます) 。ドキュメント数でソートされた用語のリスト (上位 N の値リスト) を作成するには、doc_count の orderBy プロパティを追加します。これにより、降順ソートが自動的に選択されます。doc_count (降順 N リスト) で asc を使用するには、order: "asc" を設定しますが、ドキュメントカウントのエラーが増加するため、推奨されません。ドキュメントカウントの順序で用語を保持するには、変数の [ソート] ドロップダウンリストを[無効]に設定します。または、[アルファベット]順を使用して再ソートすることもできます。

```
{"find": "terms", "field": "@hostname", "orderBy": "doc_count"}
```

クエリでの変数の使用

2 つの構文があります。

- $<varname> 例: @hostname:$hostname
- [[varname]] 例: @hostname:[[hostname]]

なぜ 2 つの方法があるのですか？ 最初の構文は読み書きが容易ですが、単語の途中で変数を使用することは許可されません。[複数値]または[すべての値を含める]オプションが有効になっている場合、Grafana はラベルをプレーンテキストから Lucene 互換の条件に変換します。

前の例では、$hostname という名前の変数を使用して @hostname プロパティに基づいてドキュメントをフィルタリングする Lucene クエリがあります。また、フィールド入力ボックス別の用語グループで変数を使用しています。これにより、変数を使用してデータのグループ化方法をすばやく変更できます。

注釈

注釈を使用すると、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニューまたは注釈ビューを使用して注釈クエリを追加します。Grafana は、注釈イベントの OpenSearch インデックスをクエリできます。詳細については、「注釈」を参照してください。

名前	説明
Que	検索クエリは空白のままにするか、Lucene クエリを指定できます。
Tim	時間フィールドの名前。 日付フィールドである必要があります。
Tim End	終了時刻フィールドのオプション名は日付フィールドである必要があります。設定されている場合、注釈は時間と終了時間の間のリージョンとしてマークされます。
Tex	イベントの説明フィールド。
Tag	イベントタグに使用するオプションのフィールド名 (配列または CSV 文字列) 。

ログのクエリ

OpenSearch からのログデータのクエリと表示は、Explore で確認できます。ログを表示するには、OpenSearch Service データソースを選択し、オプションで Lucene クエリを入力します。詳細については、「Explore」を参照してください。

ログクエリ

結果が返されると、ログパネルにログ行のリストと棒グラフが表示されます。x 軸は時間、y 軸は頻度またはカウントを示します。

ログメッセージのフィルタリング

必要に応じて、クエリフィールドに Lucene クエリを入力して、ログメッセージをフィルタリングできます。例えば、デフォルトの Filebeat 設定を使用すると、`fields.level:error` を使用してエラーメッセージのみを表示できます。

Amazon OpenSearch Service Serverless

> **Note**
>
> OpenSearch Service Serverless は、Grafana バージョン 9.4 以降を実行している Grafana ワークスペースでのみ使用することができます。

Amazon Managed Grafana では、OpenSearch Service データソースを使用して、Amazon OpenSearch Service Serverless のデータにアクセスできます。データへのアクセス権は、データアクセスポリシーによって制御されます。以下に、ユーザーが特定のコレクションとインデックスをクエリできるようにするポリシーの例を紹介します。*collection_name*、*index_name*、*principal_arn* を実際の使用に合わせた値に置き換えてください。

```
[
  {
    "Rules": [
      {
        "Resource": ["collection/{collection_name}"],
        "Permission": ["aoss:DescribeCollectionItems"],
        "ResourceType": "collection"
      },
      {
        "Resource": ["index/{collection_name}/{index_name}"],
        "Permission": ["aoss:DescribeIndex", "aoss:ReadDocument"],
        "ResourceType": "index"
      }
    ],
    "Principal": ["principal_arn"],
    "Description": "read-access"
```

```
  }
]
```

トレースの補助機能

OpenSearch プラグインでは、トレースのリストをテーブル形式で表示したり、単一のトレースを [トレースの表示]] で表示して、トレーススパンのタイムラインを確認することができます。

> **Note**
>
> OpenSearch トレースのクエリは、Lucene クエリでのみ実行できます。
> トレースの補助機能は、バージョン 9.4 以降をサポートする Grafana ワークスペースでのみ使用できます。

すべてのトレースを表示するクエリを作成するには、空白のクエリで Lucene クエリタイプ `Traces` を使用します。必要に応じて、テーブルの視覚的表現タイプを選択します。

テーブルでトレース ID を選択すると、トレースビューでそのトレースが開きます。

単一のトレースを示すクエリを作成するには、クエリ `traceid: {traceId}` を使用し、必要に応じてトレースの視覚的表現タイプを選択します。

AWS IoT SiteWise データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Amazon Managed Grafana では、Grafana ワークスペースコンソールの AWS データソース設定オプションを使用して、データソース AWS IoT SiteWise として を追加できます。この機能は、既存の AWS IoT SiteWise アカウントを検出してデータソース AWS IoT SiteWise として を追加するのを簡素化し、アクセスに必要な認証情報の設定を管理します AWS IoT SiteWise。この方法を使用して認証を設定し、データソースとして AWS IoT SiteWise を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

トピック

- AWS データソース設定を使用してデータソース AWS IoT SiteWise として を追加する
- AWS IoT SiteWise データソースの手動追加
- AWS IoT SiteWise 設定
- AWS IoT SiteWise データソースの使用

AWS データソース設定を使用してデータソース AWS IoT SiteWise として を追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の AWS IoT SiteWise リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして AWS IoT SiteWise を追加します。

AWS データソース設定を使用してデータソース AWS IoT SiteWise として を追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。
4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
5. [データソース] タブを選択します。次に、[AWS IoT SiteWise] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
6. [データソース] タブを再び選択してから [AWS IoT SiteWise] 行で [Grafana で設定] を選択します。
7. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
8. Grafana ワークスペースコンソールの左側のナビゲーションバーで、 AWS アイコンを選択し、AWS サービス IoT SiteWise を選択します。
9. AWS IoT SiteWise データソースにクエリを実行するデフォルトのリージョンを選択し、アカウントを選択してから、データソースの追加を選択します。

AWS IoT SiteWise データソースの手動追加

AWS IoT SiteWise データソースを手動で追加するには

1. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンで一時停止して [データソース] を選択します。
2. [データソースの追加] を選択します。
3. AWS IoT SiteWise のデータソースを選択します。必要に応じて、検索ボックスに **SiteWise** の入力を開始すると検索を補助することができます。

AWS IoT SiteWise 設定

名前	説明
名前	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
認証プロバイダー	認証情報を取得するプロバイダーを指定します。
デフォルトのリージョン	クエリエディタでリージョンを設定するために使用します (クエリごとに変更できます)。
認証情報プロファイル名	使用するプロファイルの名前を指定します (`~/.aws/credentials` ファイルを使用する場合)。デフォルトでは空白のままにします。
引き受けるロールの Arn	引き受けるロールの ARN を指定します。
エンドポイント (オプション)	代替サービスエンドポイントを指定する必要がある場合に指定します。

AWS IoT SiteWise データソースの使用

AWS IoT SiteWise データソースの使用方法については、Github のAWS IoT SiteWise 「データソース」を参照してください。

an AWS IoT TwinMaker データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Amazon Managed Grafana を使用すると、強力な産業データ分析サービスである AWS IoT TwinMaker を Grafana ワークスペースのアプリケーションおよびデータソースとして追加できます。 AWS IoT TwinMaker を使用すると、エンドユーザーの 3D デジタルツインアプリケーションを作成して、産業オペレーションをモニタリングできます。 AWS IoT TwinMaker は、デベロッパーが現実世界のシステムのデジタルレプリカをより迅速に作成できるようにするサービスです。これにより、より多くのお客様がデジタルツインが運用を最適化する可能性を実現できます。 AWS IoT TwinMaker for Grafana は、カスタムパネル、ダッシュボードテンプレート、およびデジタルツインデータに接続するためのデータソースを提供します。

AWS IoT TwinMaker データソースの手動追加

前提条件

開始する前に、 AWS アカウントから AWS IoT [TwinMaker] にアクセスできることを確認してください。

ワークスペース IAM ロールに アクセス許可を追加して AWS IoT TwinMaker にアクセスする方法については、「」を参照してくださいワークスペースユーザーロールに for AWS IoT TwinMaker のアクセス許可を追加する。

AWS IoT TwinMaker データソースを追加するには：

1. ユーザーロールが管理者または編集者であることを確認します。
2. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンにカーソルを合わせて [データソース] を選択します。

3. [データソースの追加] を選択します。
4. [AWS IoT TwinMaker] データソースを選択します。必要に応じて、検索ボックスに **`TwinMaker`** の入力を開始すると検索を補助することができます。
5. これにより、[接続の詳細] ページが開きます。「AWS IoT TwinMaker 接続の詳細設定」の手順にそってを設定します。

ワークスペースユーザーロールに for AWS IoT TwinMaker のアクセス許可を追加する

ワークスペースユーザーロールに AWS IoT TwinMaker のアクセス許可を追加するには、Amazon Managed Grafana ワークスペースと TwinMaker ダッシュボードロールの間でロールのアクセス許可を引き受けます。

1. https://console.aws.amazon.com/iam/ に移動します。
2. ダッシュボードロールを手動で作成します。ダッシュボードロールの作成に関する詳細については、「Grafana AWS IoT TwinMaker ダッシュボードロールを手動で作成するには」を参照してください。

AWS IoT TwinMaker 接続の詳細設定

[接続の詳細] 設定の構成

1. [接続の詳細] メニューで、認証プロバイダーを選択します (推奨: Workspace IAM ロール)。
2. クエリする [デフォルトリージョン] を選択します。
3. TwinMaker 設定で、 AWS IoT TwinMaker ワークスペース名を入力します。

AWS IoT TwinMaker データソースの使用

AWS IoT TwinMaker データソースの使用方法については、GitHub の AWS IoT TwinMaker データソースを参照してください。

Grafana AWS IoT TwinMaker ダッシュボードロールを手動で作成するには

Grafana AWS IoT TwinMaker ダッシュボードロールを手動で作成するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) にサインインします。
2. サマリーで Amazon Managed Grafana ワークスペースロールを見つけます。次のように表示されます。

```
AmazonGrafanaServiceRole-random_ID
```

3. 以下のインラインポリシーをロールに追加します。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "sts:AssumeRole",
        "Resource": "< TwinMaker Dashboard Role ARN >"
    }
}
```

4. ダッシュボードロールごとに新しいインラインポリシーを追加します。または、Resource (リソース) 行にロール Amazon リソースネーム (ARN) のリストを追加します。
5. IAM コンソールでダッシュボードロールを検索します。SceneViewer ポリシー (オプションで VideoPlayer ポリシー) が必要です。
6. [Trust Relationships] (信頼関係) タブを選択します。
7. [Edit trust relationship (信頼関係の編集)] を選択します。
8. 次のポリシーを入力します。*AMGWorkspaceRoleArn* をご自身のアカウントの Arn に置き換えてください。

```
  {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "AMGWorkspaceRoleARN"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

an AWS IoT TwinMaker ポリシーの例

以下は、ダッシュボードロールにアタッチできる minimal AWS IoT TwinMaker ポリシーです。独自のリソースに基づいて、 AWS IoT TwinMaker ワークスペース ARN と ID、および Amazon S3 バケット ARN の値を置き換える必要があります。

```
{
   "Version": "2012-10-17",
   "Statement": [
     {
       "Action": [
         "iottwinmaker:ListWorkspaces"
       ],
       "Resource": [
         "*"
       ],
       "Effect": "Allow"
     },
     {
       "Action": [
         "iottwinmaker:Get*",
         "iottwinmaker:List*"
       ],
       "Resource": [
         "IoTWorkspaceArn",
         "IoTWorkspaceArn/*"
       ],
       "Effect": "Allow"
     },
     {
       "Action": [
         "kinesisvideo:Describe*",
         "kinesisvideo:Get*",
         "kinesisvideo:List*"
       ],
       "Resource": "*",
       "Effect": "Allow"
     },
     {
       "Action": [
         "iotsitewise:Describe*",
         "iotsitewise:List*",
```

```
          "iotsitewise:Get*"
        ],
        "Resource": "*",
        "Effect": "Allow"
      },
      {
        "Action": "iotsitewise:BatchPutAssetPropertyValue",
        "Resource": "*",
        "Effect": "Allow",
        "Condition": {
          "StringEquals": {
            "aws:ResourceTag/IoTWorkspaceId": "SiteWatch"
          }
        }
      },
      {
        "Effect": "Allow",
        "Action": ["s3:GetObject"],
        "Resource": [
          "S3BucketArn",
          "S3BucketArn/*"
        ]
      }
    ]
  }
```

Amazon Managed Service for Prometheus およびオープンソースの Prometheus データソースへの接続

Amazon Managed Grafana で、Prometheus データソースは、セルフマネージド Prometheus サーバーと Amazon Managed Service for Prometheus ワークスペースの両方のデータソースとしての使用をサポートしています。Amazon Managed Service for Prometheus の詳細については、[Amazon Managed Service for Prometheus とは] を参照してください。

Amazon Managed Grafana では、Grafana ワークスペースコンソールの AWS データソース設定オプションを使用して、Amazon Managed Service for Prometheus ワークスペースをデータソースとして追加できます。この機能は、既存の Amazon Managed Service for Prometheus アカウントを検出して Amazon Managed Service for Prometheus をデータソースとして追加する作業を簡素化し、Amazon Managed Service for Prometheus へのアクセスに必要な認証情報の設定を管理します。

> **Note**
>
> Prometheus アラートは、アラートマネージャーデータソースの設定 により、統合 Grafana アラートインターフェイスで表示できます。

トピック

- AWS データソース設定を使用して Amazon Managed Service for Prometheus をデータソースとして追加する
- Prometheus データソースの手動追加
- Prometheus データソースの使用
- Amazon Managed Service for Prometheus からのアラートを視覚化する
- Exemplars の設定

AWS データソース設定を使用して Amazon Managed Service for Prometheus をデータソースとして追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の Amazon Managed Service for Prometheus リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして Amazon Managed Service for Prometheus を追加します。

AWS データソース設定を使用して Amazon Managed Service for Prometheus をデータソースとして追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。
4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。

5. [データソース] タブを選択します。次に、[Amazon Managed Service for Prometheus] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] の順に選択します。
6. [データソース] タブを再度選択し、[Amazon Managed Service for Prometheus] 行で [Grafana で設定] を選択します。
7. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
8. Grafana ワークスペースコンソールの左側のナビゲーションバーで、 AWS アイコンを選択し、AWS サービスである Prometheus を選択します。
9. Amazon Managed Grafana が Amazon Managed Service for Prometheus ワークスペースを見つけるためのリージョンを選択し、追加するアカウントと Amazon Managed Service for Prometheus ワークスペースを選択して [データソースの追加] を選択します。

Prometheus データソースの手動追加

Prometheus データソースを手動で追加する方法

1. Grafana コンソールのサイドメニューで、[管理] メニュー項目 (または Grafana v8 の設定 (ギア) アイコン) で一時停止して [データソース] を選択します。
2. [データソースの追加] を選択します。
3. [Prometheus] データソースを選択します。必要に応じて、検索ボックスに **`Prometheus`** の入力を開始すると検索を補助することができます。

Prometheus データソースの使用

Prometheus の設定

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Url`	Prometheus サーバーの URL。例: `https://prometheus.example.org:9090`

名前	説明
Access	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。
Basic Auth	Prometheus データソースへの基本認証を有効にします。
User	基本認証のユーザー名。
Passwor	基本認証のパスワード。
Scrape interva	これを Prometheus で設定された一般的なスクレイプと評価の間隔に設定します。デフォルトは 15 秒です。
Disable metrics lookup	このオプションをオンにすると、クエリフィールドのオートコンプリートでメトリクスセレクタとメトリクス/ラベルのサポートが無効になります。これにより、より大きな Prometheus インスタンスでパフォーマンスの問題が発生した場合に役立ちます。
Custom Query Paramet s	Prometheus クエリ URL にカスタムパラメータを追加します。例えば、timeout、partial_response 、dedup または max_source_resolution などです。複数のパラメータを「&」で連結する必要があります。

Prometheus クエリエディタ

次のセクションでは、ダッシュボードと Explore の Prometheus クエリエディタに関する情報とオプションについて説明します。

ダッシュボードのクエリエディタ

タイトルを選択し、[編集] (またはパネルで一時停止中に e キーを押す) を選択して、編集モードでグラフを開きます。

名前	説明
Query	Prometheus のクエリ式の詳細については、Prometheus のドキュメントを参照してください。
Legend	名前またはパターンを使用して、時系列の名前を制御します。例えば、{{hostname}} はラベル hostname のラベル値に置き換えられます。
Min step	Prometheus 範囲クエリの step パラメータと、$__interval および $__rate_interval 変数の追加の下限。制限は絶対であり、解決設定でも変更されません。
Resolution	1/1 は、各ピクセルが 1 つのデータポイントに対応するように、Prometheus 範囲クエリの$__interval 変数と step パラメータの両方を設定します。パフォーマンスを向上させるには、低解像度を使用します。1/2 は 1 ピクセルごとにデータポイントを取得し、1/10 は 10 ピクセルあたり 1 つのデータポイントを取得します。最小時間間隔と最小ステップの両方が、$__interval と step の最終値を制限することに注意してください。
Metric lookup	この入力フィールドでメトリクス名を検索します。
Format as	Table、Time series、または Heatmap を切り替えます。Table はテーブルパネルでのみ機能します。 Heatmap は、ヒートマップパネルにヒストグラムタイプのメトリクスを表示するのに適しています。累積ヒストグラムを通常のヒストグラムに変換し、バケットの境界でシリーズをソートします。
Instant	「インスタント」クエリを実行して、要求された時系列に対して Prometheus がスクレイピングした最新の値のみを返します。インスタントクエリは、通常の範囲クエリよりもはるかに高速に結果を返します。これらを使用して、ラベルセットを検索します。
Min time interval	この値に解決設定の分母を掛けると、Prometheus 範囲クエリの $__interval 変数と step パラメータの両方に下限が設定されます。デフォルトでは、データソースオプションで設定されたスクレイプ間隔が指定されます。

> **Note**
>
> Amazon Managed Grafana は、動的に計算されたステップに合わせてクエリのリクエスト日を変更します。これにより、メトリクスデータの一貫した表示が保証されますが、グラフの右端にあるデータの小さなギャップが生じる可能性があります。

ダッシュボード内のインスタントクエリ

Prometheus データソースを使用すると、最新の値のみをクエリするインスタントクエリを実行できます。テーブルパネルで結果を視覚化して、時系列で使用可能なすべてのラベルを表示できます。

インスタントクエリの結果は、シリーズごとに 1 つのデータポイントのみで構成されます。これらは、シリーズのオーバーライドを使用してグラフパネルに表示できます。最新の値ポイントとしてグラフに表示するには、シリーズオーバーライドを追加して `Points > true` を選択します。グラフ全体に水平線を表示するには、シリーズオーバーライドを追加し、`Transform > constant` を選択します。シリーズオーバーライドの詳細については、「シリーズオーバーライド」を参照してください。

Explore のクエリエディタ

名前	説明
Quexn	Prometheus のクエリ式の詳細については、Prometheus のドキュメントを参照してください。
St	Prometheus 範囲クエリの Step パラメータ。ここで時間単位を使用できます。例えば、5s、1m、3h、1d、1y を使用できます。指定された単位が s (秒) でない場合のデフォルト単位。
Quty	Range、Instant、Both範囲クエリを実行すると、クエリの結果がグラフとテーブルに表示されます。インスタントクエリは、要求された時系列に対してPrometheus がスクレイピングした最新の値のみを返し、テーブルに表示します。[Both] (両方) を選択すると、インスタントクエリと範囲クエリの両方が実行されます。範囲クエリの結果がグラフに表示され、即時クエリの結果がテーブルに表示されます。

メトリクスブラウザ

メトリクスブラウザを使用すると、メトリクスをすばやく検索し、関連するラベルを選択して基本的なクエリを作成できます。ブラウザを開くと、使用可能なすべてのメトリクスとラベルが表示されます。Prometheus インスタンスでサポートされている場合、各メトリクスには HELP と TYPE がツールヒントとして表示されます。

メトリクスを選択すると、ブラウザは使用可能なラベルを絞り込み、メトリクスに該当するラベルのみを表示します。その後、下部セクションにあるリストに使用可能なラベル値が表示され、1 つ以上のラベルを選択できます。各ラベルに 1 つ以上の値を選択して、クエリ範囲を絞り込みます。

> **Note**
>
> 開始するメトリクス名がわからない場合は、まずいくつかのラベルを選択してリストを絞り込み、関連するラベル値を見つけることもできます。

メトリクスブラウザのすべてのリストには、特定の文字列に一致するメトリクスまたはラベルをすばやくフィルタリングするための検索フィールドが上部にあります。値セクションの検索フィールドは 1 つだけです。フィルタリングはすべてのラベルに適用され、ラベルアプリケーション、ジョブ、ジョブ名など、ラベルが選択された後に、探している値を持つ可能性のある 1 つのみの値を見つけるのに役立ちます。

所望のクエリが得られたら、「クエリの使用」をクリックしてクエリを実行します。[レートクエリとして使用] ボタンは、カウンターメトリクスのクエリの書き込みに役立つ rate(...)[$__interval] をクエリの周囲に追加します。[Validate selector] (セレクタの検証) ボタンは、そのセレクタで使用できる時系列の数を Prometheus で確認します。

制限

メトリクスブラウザには、10,000 ラベル (キー) と 50,000 ラベル値 (メトリクス名を含む) の絶対制限があります。Prometheus インスタンスがより多くの結果を返した場合でも、ブラウザは引き続き機能します。ただし、結果セットはこれらの最大制限を超えてカットされます。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

クエリタイプの変数を使用すると、メトリクス、ラベル、またはラベル値のリストを Prometheus にクエリできます。Prometheus データソースプラグインには、クエリ入力フィールドで使用できる以下の関数が用意されています。

名前	説明
`label_names()`	ラベル名のリストを返します。
`label_values(label)`	すべてのメトリクスの `label` 用のラベル値リストを返します。
`label_values(metric, label)`	指定されたメトリクスの `label` 用のラベル値リストを返します。
`metrics(metric)`	指定された `metric` 正規表現に一致するメトリクスのリストを返します。
`query_result(query)`	`query` の Prometheus クエリ結果のリストを返します。

メトリクス名 、ラベル名、ラベル値については、Prometheus ドキュメントを参照してください。

間隔変数と範囲変数の使用

> **Note**
>
> `$__range`、`$__range_s`、`$__range_ms` は Grafana v5.3 でのみサポートされています。

クエリ変数には、`$__interval`、`$__interval_ms`、`$__range`、`$__range_s`、`$__range_ms` などのグローバル変数を使用できます。詳細については、「グローバル変数」を参照してください。`label_values` 関数はクエリをサポートしていないため、可変クエリをフィルタリングする必要がある場合は、`query_result` 関数で使用すると便利です。

ダッシュボードで時間範囲を変更するときに正しいインスタンスを取得するには、変数の `refresh`トリガーを `On Time Range Change` に設定してください。

次のコード例は、ダッシュボードに表示される時間範囲の平均 QPS に基づいて、最もビジーな5つの陸絵sとインスタンスを変数に設定する方法を示しています。

```
Query: query_result(topk(5, sum(rate(http_requests_total[$__range])) by (instance)))
Regex: /"([^"]+)"/
```

次のコード例は、$__range_s を使用して、ダッシュボードに表示される時間範囲にわたって特定の状態を持つインスタンスで変数を入力する方法を示しています。

```
Query: query_result(max_over_time(<metric>[${__range_s}s]) != <state>)
Regex:
```

$__rate_interval 変数の使用

$__rate_interval 変数は、レート関数で使用することを目的としています。これは 最大 ($__interval + スクレイプ間隔 、4 * スクレイプ間隔) として定義されます。スクレイプ間隔は、設定されている場合は最小ステップ設定 (AKA query_interval、PromQL クエリごとの設定) で、それ以外の場合は Prometheus データソースで設定されたスクレイプ間隔です (ただし、後者は解決設定によって変更されるため、パネル内の最小間隔設定は無視します)。

クエリでの変数の使用

2 つの構文があります。

- $<varname> 例: rate(http_requests_total{job=~「$job」}[5m])
- [[varname]] 例: rate(http_requests_total{job=~「[job]]」}[5m])

なぜ 2 つの方法があるのですか？ 最初の構文は読み書きが容易ですが、単語の途中で変数を使用することは許可されません。[複数値]または[すべての値を含める]オプションが有効になっている場合、Grafana はラベルをプレーンテキストから正規表見互換の文字列に変換します。つまり、 = の代わりに =~ を使用する必要があります。

注釈

注釈を使用して、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニューまたは注釈ビューを使用して注釈クエリを追加します。詳細については、「注釈」を参照してください。

Prometheus は、注釈をクエリする 2 つの方法をサポートしています。

- 通常のメトリクスクエリ
- 保留中のアラートと発射アラートの Prometheus クエリ。詳細については、「ランタイム中のアラートの検査」を参照してください。

ステップオプションは、クエリから返されるイベントの数を制限するのに役立ちます。

Amazon Managed Service for Prometheus からのアラートを視覚化する

既に接続されている Prometheus データソースに対して、アラートマネージャーデータソースを設定することで、Amazon Managed Grafana で Amazon Managed Service for Prometheus または Prometheus アラートを 視覚化することができます。

前提条件

アラートマネージャーで Amazon Managed Service for Prometheus を使用するように設定するには、次の前提条件を満たす必要があります。

- [Amazon Managed Service for Prometheus] インスタンスに取り込まれたメトリクスがあり、少なくとも 1 つのアラートまたは記録ルールが設定されている。ワークスペースの URL が必要です ([エンドポイント] URL は、Amazon Managed Service for Prometheus のワークスペースの詳細から確認できます。ワークスペース URI は、エンドポイント URL から末尾の `api/v1/remote_write` を除いたものです)。
- Prometheus インスタンスがデータソースとして設定された Amazon Managed Grafana ワークスペースが作成されている。
- Amazon Managed Grafana には、Prometheus リソースに対する次のアクセス許可が必要です。これらのアクセス許可は、AWS データソースの Amazon Managed Grafana アクセス許可とポリシー で説明されているサービス管理ポリシーとカスタマー管理ポリシーのいずれかに追加する必要があります。
  - `aps:ListRules`
  - `aps:ListAlertManagerSilences`
  - `aps:ListAlertManagerAlerts`
  - `aps:GetAlertManagerStatus`
  - `aps:ListAlertManagerAlertGroups`
  - `aps:PutAlertManagerSilences`

- `aps:DeleteAlertManagerSilence`

Amazon Managed Service for Prometheus で使用するアラートマネージャーのデータソースの設定方法

1. Grafana コンソールの Grafana メニューの [設定] にある [データソース] ページを選択します。
2. [データソースの追加] を選択し、データソースタイプのリストから [アラートマネージャー] を選択します。
3. 新規データソースに次の情報を入力します:
   - [Implementation] (実装) で [Prometheus] を選択します。
   - HTTP の URL に、`alertmanager` を追加した Prometheus のワークスペース URL を入力します。例えば、`https://aps-workspaces.us-east1.amazonaws.com/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager` と指定します。
   - [認証] で、[SigV4Auth] をオンにします。これにより、リクエストに AWS 認証を使用するように Grafana に指示します。
   - [SigV4Auth の詳細] で、[デフォルトのリージョン] に Prometheus インスタンスのリージョンを指定します。例えば、`us-east-1` を指定します。
4. [保存とテスト] を選択して、データソースのセットアップを完了します。

   データソースが正しく設定されている場合、ヘルスチェックに合格したことを示すメッセージが表示されます。

新しいアラートマネージャーデータソースを Prometheus データソースに接続する方法

1. Grafana コンソールの Grafana メニューの [設定] にある [データソース] ページを選択します。
2. 元の Amazon Managed Service for Prometheus データソースを選択し、[アラート UI を使用してアラートを管理する] トグルスイッチを有効にします。
3. [保存とテスト] を選択して、データソースの設定を完了します。

Exemplars の設定

> **Note**
>
> この機能は、Prometheus バージョン 2.26 以降が必要です。

Amazon Managed Service for Prometheus では、Exemplars はサポートされていません。

Exemplars のデータは、Explore と ダッシュボードの両方でメトリクスと共に表示できます。Exemplars は、特定のイベントからの高粒度のメタデータを従来の時系列データに関連付けます。

Exemplars へのリンクを追加することで、データソース設定で Exemplars を設定できます。URL にマクロを使用することができます。例えば、`https://example.com/${__value.raw}` のような URL の作成が可能です。

Amazon Timestream データソースへの接続

Note

バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース設定オプションを使用して、Amazon Timestream を AWS データソースとして追加できます。この機能では、既存の Timestream アカウントを検出して、Timestream をデータソースとして追加する作業が簡素化され、Timestream へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定し、データソースとして Timestream を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

AWS データソース設定を使用して Timestream をデータソースとして追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の Timestream リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして Timestream を追加します。

AWS データソース設定を使用して Timestream をデータソースとして追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。

2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。
4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
5. [データソース] タブを選択します。次に、[Amazon Timestream] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
6. [データソース] タブを再度選択し、[Amazon Timestream] 行で [Grafana で設定] を選択します。
7. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
8. Grafana ワークスペースコンソールの左側のナビゲーションバーで、アプリとAWS データソースを選択します (Grafana v8 では、左側のメニューから AWS アイコンを選択します)。
9. AWS [サービス] タブを選択して、[Timestream] を選択します。
10. [Timestream] データソースがクエリするデフォルトのリージョンを選択して、アカウントを選択し、[データソースの追加] を選択します。

Timestream データソースの手動追加

Timestream データソースの手動追加の方法

1. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンで一時停止して [データソース] を選択します。
2. [データソースの追加] を選択します。
3. [Amazon Timestream] のデータソースを選択します。必要に応じて、検索ボックスに **`Timestream`** の入力を開始すると検索を補助することができます。

Timestream 設定

名前	説明
名前	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
認証プロバイダー	認証情報を取得するプロバイダーを指定します。
デフォルトのリージョン	クエリエディタでリージョンを設定するために使用します (クエリごとに変更できます)。
認証情報プロファイル名	使用するプロファイルの名前を指定します (`~/.aws/credentials` ファイルを使用する場合)。デフォルトでは空白のままにします。
引き受けるロールの Arn	引き受けるロールの ARN を指定します。
エンドポイント (オプション)	代替サービスエンドポイントを指定する必要がある場合に指定します。

認証

このセクションでは、Amazon Timestream データソースで使用できるさまざまなタイプの認証について説明します。

AWS 認証情報の例

Amazon Managed Grafana では、認証情報ファイルメソッドの認証は使用できません。

Timestream データソースの使用

クエリエディタ

クエリエディタは、以前にリストされたマクロとダッシュボードテンプレート変数に加え、Timestream 構文を受け入れます。

Ctrl+スペース を押して IntelliSense の提案を開きます。

マクロ

構文を簡素化し、日付範囲フィルターなどの動的部分を許可するため、クエリにマクロを含めることができます。

マクロの例	説明
$__database	選択済みのデータベースを指定します。これはデータソース設定のデフォルト、またはクエリエディタの明示的な値を使用します。
$__table	選択済みのデータベースを指定します。これはデータソース設定のデフォルト、またはクエリエディタの明示的な値を使用します。
$__measure	選択した測定を指定します。これはデータソース設定のデフォルト、またはクエリエディタの明示的な値を使用します。
$__timeFilter	ダッシュボード範囲までの時間を制限する式で置き換えられます
$__interval_ms	グラフ内の単一ピクセルがカバーする合計時間を表す数値で置き換えられます。

Amazon Athena データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

> **Note**
>
> このガイドでは、Amazon Athena データソースを使用する前に、ユーザーが Amazon Athena サービスに精通していることを前提としています。

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース AWS 設定オプションを使用して、データソースとして Athena を追加できます。この機能は、既存の Athena アカウントを検出してデータソースとして Athena を追加するのを簡素化し、Athena へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定し、データソースとして Athena を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

Amazon Managed Grafana が Athena にアクセスできるようにするには、いくつかの前提条件があります。Athena データソースの使用に関連する前提条件については、「前提条件」を参照してください。

前提条件

Amazon Managed Grafana for Athena のマネージドポリシーを使用するには、Athena データソースを設定する前に以下の作業が完了している必要があります:

- Athena ワークグループに `GrafanaDataSource: true` をタグ付けします。
- `grafana-athena-query-results-` で始まる名前の S3 バケットを作成します。このポリシーにより、その名前付け規則を使用してクエリ結果を S3 バケットに書き込むための権限が提供されます。

Athena クエリの基盤となるデータソースにアクセスするための Amazon S3 の権限は、このマネージドポリシーには含まれません。そのため、必要に応じて対象の Amazon S3 バケットに手動でアクセス権を追加する必要があります。詳細については、このガイドの「Amazon Managed Grafana の ID ベースのポリシーの例」を参照してください。

AWS データソース設定を使用して Amazon Athena をデータソースとして追加する

前提条件

- AWS CLI が環境にインストールされ、構成されている。
- ご自身のアカウントから Athena にアクセスできる。

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールに移動して、アカウントまたは組織単位全体の Athena リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービスマネージド IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして Athena を追加します。

AWS データソース設定を使用して Athena をデータソースとして追加するには

1. ユーザーロールが管理者または編集者であることを確認します。
2. https://console.aws.amazon.com/grafana/ の Amazon Managed Grafana コンソールから、作業するワークスペースを選択します。
3. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
4. [データソース] タブを選択します。次に、[Amazon Athena] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
5. [データソース] タブを再度選択し、Amazon Athena 行で [Grafana で設定] を選択します。
6. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。Athena データソースにアクセスするには、ユーザー/ロールに Athena アクセスポリシーが適用されている必要があります。詳細については、「AWS マネージドポリシー: AmazonGrafanaAthenaAccess 」を参照してください。
7. Grafana ワークスペースコンソールの左側のナビゲーションバーで、下部 AWS のアイコン (2 つあります) を選択し、データソースメニューから Athena を選択します。
8. Athena データソースがクエリするデフォルトリージョンを選択し、対象となるアカウントを選択し、[データソースの追加] を選択します。
9. [Athena Details] (Athena の詳細) の設定 の手順に従って [Athena Details] (Athena の詳細) を設定します。

[Athena Details] (Athena の詳細) の設定

[Athena Details] (Athena の詳細) 設定の設定

1. [接続の詳細] メニューで、認証プロバイダーを選択します (推奨: Workspace IAM ロール)。

2. Athena アカウントがある対象の Athena データソースを選択します。データソースを選択しない場合、ドロップダウンにはデフォルトのデータソースのみが表示されます。

   新しい Athena アカウントを作成するには、「Athena の開始方法」の手順に従ってください。
3. 上記で選択したデータソースで、対象の Athena データベースを選択します。
4. ワークグループを選択します。デフォルトは「プライマリ」です。
5. ワークグループに出力場所がまだ設定されていない場合、クエリ結果に使用する S3 バケットとフォルダを指定します。例えば、`s3://grafana-athena-plugin-test-data/query-result-output/`
6. [保存してテスト] を選択します。

Athena データソースを手動で追加する

前提条件

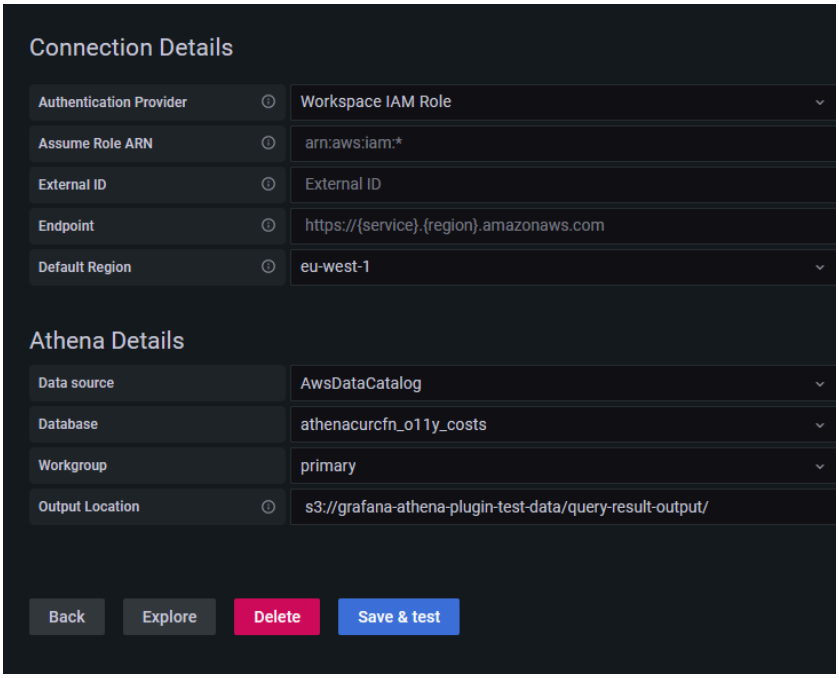
- AWS CLI が環境にインストールされ、構成されている。
- ご自身のアカウントから Amazon Athena にアクセスできる。

Athena データソースを手動で追加する方法：

1. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンで一時停止して [データソース] を選択します。
2. [データソースの追加] を選択します。
3. AWS Athena データソースを選択します。必要に応じて、検索ボックスに **Athena** の入力を開始すると検索を補助することができます。
4. [接続の詳細] メニューで、認証プロバイダーを設定します (推奨: Workspace IAM ロール)。
5. 対象の Athena データソース、データベース、ワークグループを選択します。

   新しい Athena アカウントを作成するには、「Athena の開始方法」の手順に従ってください。
6. ワークグループに出力場所がまだ設定されていない場合、クエリ結果に使用する S3 バケットとフォルダを指定します。例えば、`s3://grafana-athena-plugin-test-data/query-result-output/` と指定します。
7. [保存してテスト] を選択します。

[Athena Details] (Athena の詳細) の設定例を以下に示します。



Athena データソースを使用する

IAM ポリシー

Grafana が Athena のメトリクスを読み取るには、IAM 経由で権限が付与されている必要があります。これらの権限を IAM ロールに付与し、Grafana のビルトイン機能を使用してロールを引き受けることができます。データソースを Grafana に追加する前に、ロールに 必要なポリシーを設定 しておく必要があります。データソースを追加するには、管理者または編集者ロールが必要です。ビルトインの Amazon Grafana Athena アクセスポリシーは、 AWS マネージドポリシー: AmazonGrafanaAthenaAccess セクションで定義されています。

Athena データのクエリ

Athena データソースには、標準の SQL クエリエディタが用意されています。Amazon Managed Grafana には、より複雑な時系列クエリの記述に役立つマクロがいくつか含まれています。

マクロ

マクロ	説明	例	出力例
`$__dateFilter(column)`	`$__dateFilter` は、パネルの日付範囲を基に (column を使用して) データを選択	`$__date(my_date)`	`my_date BETWEEN date '2017-07-18' AND date '2017-07-18'`

マクロ	説明	例	出力例
	する条件付きフィルターを作成します。		
$__parseTime(column,format)	$__parseTime は、指定された形式の varchar 型をtimestamp 型に変換します。	$__parseTime(eventtime, 'yyyy-MM-dd''T''HH:mm:ss''Z')	parse_datetime(time,'yyyy-MM-dd''T''HH:mm:ss''Z')
$__timeFilter(column,format)	$__timeFilter は、パネルの時間範囲を基に (column を使用して) データのフィルタ条件を作成します。第 2 引数は、カラムを varchar 型から指定された形式の timestamp 型に変換するために使用します (オプション)。	$__timeFilter(time, 'yyyy-MM-dd HH:mm:ss')	TIMESTAMP time BETWEEN TIMESTAMP '2017-07-18T11:15:52Z' AND TIMESTAMP '2017-07-18T11:15:52Z'
$__timeFrom()	$__timeFrom は、パネルの範囲の現在の開始時刻を引用符付きで出力します。	$__timeFrom()	TIMESTAMP '2017-07-18 11:15:52'
$__timeTo()	$__timeTo は、パネルの範囲の現在の終了時刻を引用符付きで出力します。	$__timeTo()	TIMESTAMP '2017-07-18 11:15:52'

マクロ	説明	例	出力例
$__timeGroup(column, '1m', format)	$__timeGroup は、タイムスタンプをグループ化して、グラフ上の期間ごとに 1 つのポイントになるようにします。第 3 引数は、カラムを varchar 型から指定された形式の timestamp 型に変換するために使用します (オプション)。	$__timeGroup(time, '5m','yyyy-MM-dd''T''HH:mm:ss.SSSSSS''Z')	FROM_UNIXTIME(FLOOR(TO_UNIXTIME(parse_datetime(time,'yyyy-MM-dd''T''HH:mm:ss.SSSSSS''Z'))/300)*300)
$__table	$__table は、[テーブル] セレクタで選択されたテーブルを返します。	$__table	my_table
$__column	$__column は、[列] セレクタで選択された列を返します (テーブルが必要です)。	$__column	col1

視覚的表現

Athena でのほとんどのクエリは、テーブル形式での視覚的表現が最も適しています。クエリから返されたデータはテーブル形式で表示されます。クエリが可能なデータは、テーブルとして表示することができます。

以下の例は、テーブル形式での視覚的表現に対応した結果を返します。

```
SELECT {column_1}, {column_2} FROM {table};
```

時系列/グラフ表現

時系列とグラフ表現を行うには、以下を行う必要があります。

- `date` または `datetime` 型の列を選択します。`date` 列は昇順である必要があります (`ORDER BY column ASC` を使用)。
- 数値列も選択します。

クエリの検査

Amazon Managed Grafana は、Athena ではサポートされていないマクロもサポートしているため、クエリをそのまま Athena にコピー & ペーストしても動作しない場合があります。Athena で直接実行可能な補完された完全なクエリは、[Query Inspector] ボタンで確認できます。完全なクエリは、[クエリ] タブの下に表示されます。

テンプレートおよび変数

Athena クエリ変数の追加の詳細については、「クエリ変数の追加」を参照してください。Athena データソースを使用可能なクエリのデータソースとして使用します。

Athena テーブルからクエリされた値は、変数として使用できます。値が多すぎると、パフォーマンスの問題が発生する可能性があるため、選択しすぎないようにしてください。

作成した変数は、変数構文 を使用して Athena クエリで使用できます。変数の詳細については、「テンプレートおよび変数」を参照してください。

注釈

注釈 を使用すると、グラフの上にリッチイベント情報をオーバーレイできます。注釈を追加するには、パネルを選択するか、ダッシュボードメニューの注釈ビューを使用して注釈クエリを追加します。

注釈を自動的に追加するクエリの例:

```
SELECT
  time as time,
  environment as tags,
  humidity as text
FROM
  tableName
WHERE
  $__dateFilter(time) and humidity > 95
```

以下の表に、注釈の表示に使用できる列の説明を示します。

名前	説明
Time	日付/時刻フィールドの名前。ネイティブ SQL 日付/時刻データ型またはエポック値を持つ列にすることができます。
Timeend	終了日時フィールドのオプション名。ネイティブ SQL 日付/時刻データ型またはエポック値を持つ列にすることができます。(Grafana v6.6 以降)
Text	イベントの説明フィールド。
Tags	カンマ区切り文字列としてイベントタグに使用するオプションのフィールド名。

非同期クエリデータのサポート

Amazon Managed Grafana の Athena クエリは、タイムアウトを避けるために非同期で処理されます。非同期クエリは、まずクエリを開始するリクエストを送り、次に進行状況を確認し、最後に結果を取得するという形で別々のリクエストを使用します。これにより、実行に時間がかかるクエリでもタイムアウトせずに処理できます。

クエリ結果の再利用

以前のクエリの結果を再利用して、クエリのパフォーマンスを向上させることができます。クエリの再利用は、クエリエディタの「クエリ結果の再利用」セクションで有効にすることができます。この設定は、再利用したい各クエリごとに行う必要があります。

> **Note**
>
> この機能を使用するには、Athena インスタンスのエンジンのバージョンが 3 である必要があります。詳細については、Amazon Athena ユーザーガイド」の「Athena エンジンバージョンの変更」を参照してください。

Amazon Redshift データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

> **Note**
>
> このガイドでは、Amazon Redshift データソースを使用する前に、ユーザーが Amazon Redshift サービスに精通していることを前提としています。

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース設定オプションを使用して、Amazon Redshift を AWS データソースとして追加できます。この機能は、既存の Amazon Redshift アカウントを検出してデータソースとして Amazon Redshift を追加するのを簡素化し、Amazon Redshift へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定し、データソースとして Amazon Redshift を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

Amazon Managed Grafana が Amazon Redshift にアクセスできるようにするには、いくつかの前提条件があります。Amazon Redshift データソースの使用に関連する前提条件については、「前提条件」を参照してください。

前提条件

Amazon Managed Grafana の AWS マネージドポリシーを使用するには、Amazon Redshift データソースを設定する前に以下のタスクを完了します。

- Amazon Redshift クラスターに `GrafanaDataSource: true` をタグ付けします。タグ付けしないと、アクセスできなくなります。
- データベース認証情報は、次の相互に排他的ないずれかの方法で作成します。
  - デフォルトのメカニズム (一時認証情報のオプション) を使用して Redshift データベースに対して認証する場合は、`redshift_data_api_user` という名前のデータベースユーザーを作成する必要があります。

- Secrets Manager の認証情報を使用する場合は、シークレットに `RedshiftQueryOwner: true` をタグ付けする必要があります。詳細については、このガイドの「Amazon Managed Grafana の ID ベースのポリシーの例」を参照してください。

AWS データソース設定を使用して Amazon Redshift をデータソースとして追加する

AWS データソース設定を使用して Amazon Redshift をデータソースとして追加するには

1. ユーザーロールが管理者または編集者であることを確認します。
2. https://console.aws.amazon.com/grafana/ の Amazon Managed Grafana コンソールから、作業するワークスペースを選択します。
3. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
4. [データソース] タブを選択します。次に、[Amazon Redshift] のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
5. [データソース] タブを再度選択し、[Amazon Redshift] 行の [Grafana で設定] を選択します。
6. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
7. Grafana ワークスペースコンソールの左側のナビゲーションバーで、(2 つあるうちの) 下部の AWS アイコンを選択し、[Redshift] を選択します。
8. Amazon Redshift データソースがクエリするデフォルトのリージョンを選択し、目的のアカウントを選択し、[データソースの追加] を選択します。
9. [接続の詳細] 設定 の手順に従って [接続の詳細] を設定します。

Amazon Redshift データソースの手動追加

前提条件

- ご自分のアカウントから [Amazon Redshift] にアクセスできます。

Amazon Redshift データソースの追加方法:

1. AmazonRedshiftAccessPolicy をワークスペースのユーザーロールにアタッチします。
2. ユーザーロールが管理者または編集者であることを確認します。
3. https://console.aws.amazon.com/grafana/ の Amazon Managed Grafana コンソールから、作業するワークスペースを選択します。
4. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンで一時停止して [データソース] を選択します。
5. [データソースの追加] を選択します。
6. [AWS Redshift] データソースを選択します。必要に応じて、検索ボックスに **`Redshift`** の入力を開始すると検索を補助することができます。
7. これにより、[接続の詳細] ページが開きます。「[接続の詳細] 設定」の手順にそってを設定します。

Amazon Redshift の設定

Amazon Redshift データソースをワークスペースに追加してから、次のように Amazon Redshift を設定します。

前提条件

- ご自分のアカウントから [Amazon Redshift] にアクセスできます。

[接続の詳細] 設定

[接続の詳細] 設定の構成

1. [接続の詳細] メニューで、認証プロバイダーを選択します (推奨: Workspace IAM ロール)。
2. クエリする [デフォルトリージョン] を選択します。

[認証] 設定

[認証] 設定の構成

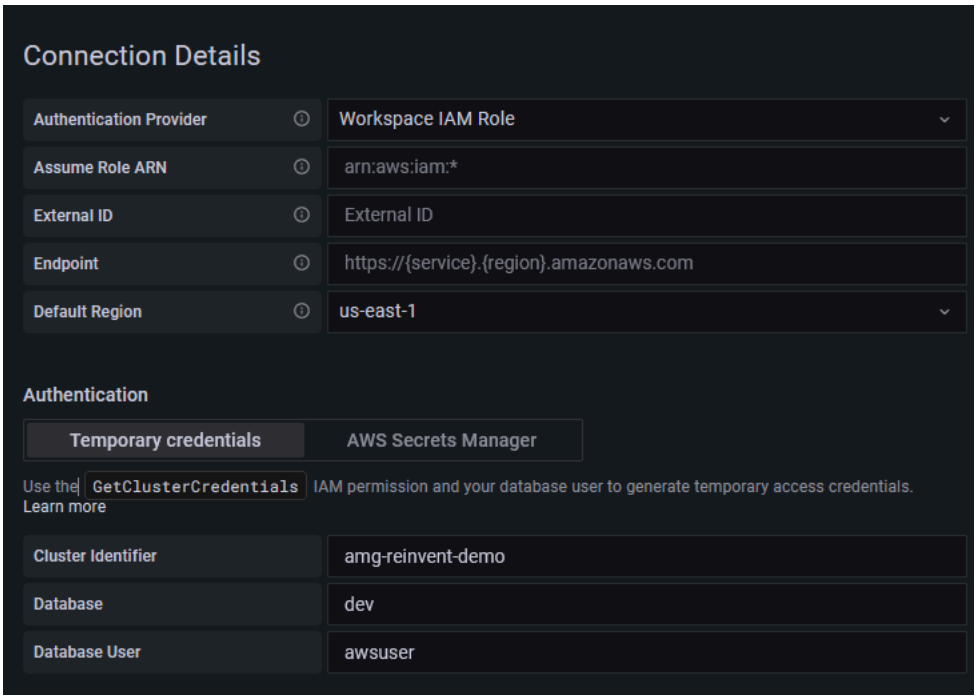
1. [認証] メニューで、アクセス認証情報プロバイダーとして [一時認証情報] または [AWS Secrets Manager] のいずれかのタブを選択します。一時的な認証情報と AWS Secrets

Manager の詳細については、「」を参照してください。 AWS マネージドポリシー: AmazonGrafanaRedshiftAccess

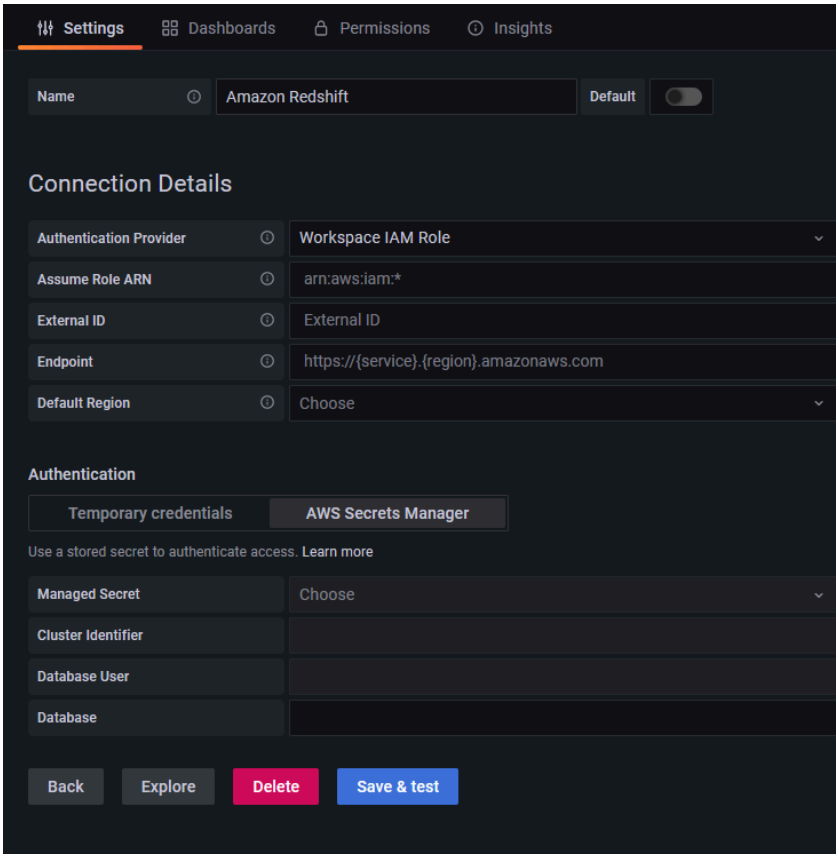
2. デフォルトの [一時的認証情報] を選択した場合は、以下の手順に従います。[AWS Secrets Manager] を選択した場合は、入力フィールドに [AWS Secrets Manager] の認証情報を入力します。
3. Amazon Redshift で作成したクラスターの [クラスター識別子] を選択します。

   Redshift クラスターの詳細については、「Redshift 接続」を参照してください。
4. 目的の Redshift データベースを選択します。
5. 上記のクラスターのために作成したデータベースユーザーを選択します。
6. [保存してテスト] を選択します。

以下は、[一時認証情報] 設定の例です。



以下は [AWS Secrets Manager] メニューの例です。



Amazon Redshift データソースの使用

IAM ポリシー

Grafana が Redshift のメトリクスを読み取るには、IAM を使用して権限が付与されている必要があります。これらの権限を IAM ロールに付与し、Grafana のビルトイン機能を使用してロールを引き受けることができます。組み込みの Amazon Grafana Redshift アクセスポリシーは、AWS マネージドポリシー: AmazonGrafanaRedshiftAccess セクションで定義されています。

Amazon Redshift データのクエリ

Amazon Redshift データソースには標準の SQL クエリエディタが用意されています。Amazon Managed Grafana には、より複雑な時系列クエリの記述に役立つマクロがいくつか含まれています。

マクロ

マクロ	説明	出力例
$__timeEpoch(column)	$__timeEpoch は UNIX タイムスタンプに変換して列の名前を時刻に変更する式に置き換えられます。	UNIX_TIMESTAMP(dateColumn) as "time"
$__timeFilter(column)	$__timeFilter はパネルの時間範囲を基準に (column を使用して) データのフィルタ条件を作成します。	time BETWEEN '2017-07-18T11:15:52Z' AND '2017-07-18T11:15:52Z'
$__timeFrom()	$__timeFrom はパネルの範囲の現在の開始時刻を引用符付きで出力します。	'2017-07-18T11:15:52Z'
$__timeTo()	$__timeTo はパネルの範囲の現在の終了時刻を引用符付きで出力します。	'2017-07-18T11:15:52Z'
$__timeGroup(column, '1m')	$__timeGroup はタイムスタンプをグループ化して、グラフ上の期間ごとに 1 つのポイントになるようにします。	floor(extract(epoch from time)/60)*60 AS "time"
$__schema	$__schema は選択したスキーマを使用します	public
$__table	$__table は特定の $__schema からテーブルを出力します (デフォルトではパブリックスキーマを使用します)	sales
$__column	$__column は現在の $__table から列を出力します	date

マクロ	説明	出力例
$__unixEpochFilter(column)	$__unixEpochFilter は指定された列名と Unix タイムスタンプとして表される時刻を使用して、時間範囲フィルターで置き換えられます。	column >= 1624406400 AND column <= 1624410000
$__unixEpochGroup(column)	$__unixEpochGroup は $__timeGroup と同じですが、Unix タイムスタンプとして保存されている時間用です	floor(time/60)*60 AS "time"

視覚的表現

Redshift でのほとんどのクエリは、テーブル形式での表示が最も適しています。すべてのクエリではテーブルにデータを表示します。クエリができれば、テーブルに入れることができます。

以下の例は、テーブル形式での視覚的表現に対応した結果を返します。

```
SELECT {column_1}, {column_2} FROM {table};
```

時系列とグラフの視覚化

時系列とグラフの視覚化には、以下のようないくつかの要件があります。

- date または datetime タイプの列を選択する必要があります。
- date 列は昇順である必要があります (ORDER BY column ASC を使用)。
- 数値列を選択する必要があります。

より妥当なグラフを作成するには、必ず $__timeFilter および $__timeGroup マクロを使用してください。

時系列クエリの例:

```
SELECT
  avg(execution_time) AS average_execution_time,
  $__timeGroup(start_time, 'hour'),
  query_type
```

```
FROM
  account_usage.query_history
WHERE
  $__timeFilter(start_time)
group by
  query_type,start_time
order by
  start_time,query_type ASC;
```

[フルモード]

Grafana はデフォルトで値を持たないフレームを自動補完します。この値を設定するには、クエリエディタで [フィル値] を変更します。

クエリの検査

Grafana では Redshift がサポートしていないマクロがサポートされるため、Redshift に直接コピーして貼り付けることができる、完全にレンダリングされたクエリはクエリインスペクターに表示されます。補間されたクエリ全体を表示するには、[クエリインスペクター] メニューを選択すると、[クエリ] タブにクエリ全体が表示されます。

テンプレートおよび変数

新しい Redshift クエリ変数を追加する方法の詳細については、「クエリ変数の追加」を参照してください。Redshift データソースを使用可能なクエリのデータソースとして使用します。

Amazon Redshift テーブルからクエリされた値は変数として使用できます。値が多すぎるとパフォーマンスの問題が発生する場合があるため、選択しすぎないようにしてください。

変数を作成したら、 変数構文 を使用して Redshift クエリで使用できます。変数の詳細については、「テンプレートおよび変数」を参照してください。

注釈

注釈 を使用すると、グラフの上にリッチイベント情報をオーバーレイできます。注釈を追加するには、パネルを選択するか、[ダッシュボード] メニューから開く [注釈] ビューを使用して注釈クエリを追加します。

注釈を自動的に追加するクエリの例:

```
SELECT
  time as time,
  environment as tags,
```

```
  humidity as text
FROM
  $__table
WHERE
  $__timeFilter(time) and humidity > 95
```

次の表は、注釈のレンダリングを考慮された列の値を示しています。

名前	説明
Time	日付または時刻のフィールド名。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます。
Timeend	終了日時フィールドのオプション名。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます。
Text	イベントの説明フィールド。
Tags	カンマ区切り文字列としてイベントタグに使用するオプションのフィールド名。

AWS X-Ray データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

をデータソース AWS X-Ray として追加し、ダッシュボードを構築するか、X-Ray で Explore を使用してトレース、分析、インサイトを確認します。

Amazon Managed Grafana では、Grafana ワークスペースコンソールのデータソース設定オプションを使用して、X-Ray を AWS データソースとして追加できます。この機能は、既存の X-Ray アカウントを検出してデータソースとして X-Ray を簡単に追加し、X-Ray へのアクセスに必要な認証情報の設定を管理します。この方法を使用して認証を設定し、データソースとして X-Ray を追加するか、セルフマネージド Grafana サーバーと同じ方法を使用して、データソースと必要な認証情報を手動で設定できます。

トピック

- AWS データソース設定を使用して X-Ray をデータソースとして追加する
- X-Ray データソースの手動追加
- X-Ray 設定
- X-Ray データソースの使用

AWS データソース設定を使用して X-Ray をデータソースとして追加する

AWS データソース設定を使用するには、まず Amazon Managed Grafana コンソールを使用して、アカウントまたは組織単位全体の X-Ray リソースを読み取るために必要な IAM ポリシーをワークスペースに付与するサービス管理 IAM ロールを有効にします。次に、Amazon Managed Grafana ワークスペースコンソールを使用して、データソースとして X-Ray を追加します。

AWS データソース設定を使用して X-Ray をデータソースとして追加するには

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. ワークスペースの名前を選択します。
4. このワークスペースの作成時にサービスマネージド型のアクセス許可を使用しない場合は、カスタマーマネージド型のアクセス許可の使用からサービスマネージド型のアクセス許可の使用に変更して、Grafana ワークスペースコンソールで AWS データソース設定オプションを使用するための適切な IAM ロールとポリシーが有効になっていることを確認します。これを行うには、[IAM ロール] で編集アイコンを選択して [サービス管理]、[変更を保存] を選択します。詳細については、「AWS データソースの Amazon Managed Grafana アクセス許可とポリシー」を参照してください。
5. [データソース] タブを選択します。次に、AWS X-Ray のチェックボックスを選択し、[アクション]、[サービス管理ポリシーを有効にする] を選択します。
6. データソース タブを再度選択し、AWS X-Ray 行で [Configure in Grafana] (Grafana で設定) を選択します。
7. 必要に応じて、IAM Identity Center を使用して Grafana ワークスペースコンソールにサインインします。
8. Grafana ワークスペースコンソールの左側のナビゲーションバーで、 AWS アイコンを選択し、AWS サービス X-Ray を選択します。
9. X-Ray データソースがクエリを実行するデフォルトリージョンを選択し、アカウントを選択し、[データソースの追加] を選択します。

X-Ray データソースの手動追加

X-Ray データソースを手動で追加するには

1. Grafana コンソールのサイドメニューで、[設定] (歯車) アイコンでー時停止して [データソース] を選択します。
2. [データソースの追加] を選択します。
3. X-Ray データソースを選択します。必要に応じて、検索ボックスに **X-Ray** の入力を開始すると検索を補助することができます。

X-Ray 設定

名前	説明
名前	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
デフォルト	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
デフォルトのリージョン	クエリエディタでリージョンを設定するために使用します (クエリごとに変更できます)。
認証プロバイダー	認証情報を取得するプロバイダーを指定します。
認証情報プロファイル名	使用するプロファイルの名前を指定します (`~/.aws/credentials` ファイルを使用する場合)。デフォルトでは空白のままにします。
引き受けるロールの Arn	引き受けるロールの ARN を指定します。
External ID	外部 ID で作成された別のアカウントでロールを引き受ける場合は、ここで外部 ID を指定します。

認証

このセクションでは、X-Ray データソースで使用できるさまざまなタイプの認証について説明します。

IAM ロール

現在、X-Ray へのすべてのアクセスは、公式 AWS SDK を使用して Grafana ワークスペースバックエンドによってサーバー側で行われます。Grafana サーバーが実行されている場合は AWS、IAM ロールを使用できます。認証は自動的に処理されます。

詳細については、「IAM ロール」を参照してください。

IAM ポリシー

Grafana では、X-Ray データおよび EC2 タグ/インスタンス/リージョンを読み取るために、IAM 経由で付与されたアクセス許可が必要です。これらのアクセス許可を IAM ロールに付与し、Grafana の組み込み機能を使用してロールを引き受けることができます。

次のコードは、最小限のポリシーのコードの例です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "xray:BatchGetTraces",
        "xray:GetTraceSummaries",
        "xray:GetTraceGraph",
        "xray:GetGroups",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetInsightSummaries",
        "xray:GetInsight",
        "ec2:DescribeRegions"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS 認証情報の例

Amazon Managed Grafana で認証情報ファイルメソッドを使用することはできません。

X-Ray データソースの使用

クエリエディタ

エディタで最も重要なフィールドは、クエリタイプです。クエリには次の 4 つのタイプがあります。

- トレースリスト (のトレース AWS)
- トレース統計
- トレース分析 (の分析 AWS)
- Insights

トレースリスト

トレースリストタイプを使用すると、トレースを検索できます。このトレースは、テーブルに表示されます。最初の列でトレース ID を選択すると、右側のトレースが開きます。エディタのクエリフィールドに注目してください。クエリを記述したり、式をフィルタリングしたり、トレースビューに表示される単一のトレース ID を挿入したりできます。フィルター式の詳細については、「AWS X-Ray ドキュメント」を参照してください。

> **Note**
>
> トレースリストには、最初の 1,000 個のトレースのみが表示されます。

トレース統計

トレース統計では、エラー、障害、スロットル、成功、合計数に関する情報を示すグラフとテーブルを確認できます。クエリエディタの列フィールドを使用して、指定された列のみを表示できます。

トレース分析

トレース分析では、次のテーブルを視覚化できます。

- 根本原因
  - 応答時間

    - 根本原因サービス (パス内の最後のサービス)
    - パス (複数パス)
  - エラー
    - 根本原因サービス (パス内の最後のサービス)
    - パス
    - エラーメッセージ
  - 障害
    - 根本原因サービス (パス内の最後のサービス)
    - パス
    - エラーメッセージ
- エンドユーザーへの影響
- [URL]
- HTTP ステータスコード

Insights

Insights では、Insights の概要テーブルを確認できます。InsightId を選択すると、 AWS マネジメントコンソールが表示されます。

[アラート]

X-Ray クエリは数値データを返す可能性があるため、アラートがサポートされています。詳細については、「Grafana アラート」を参照してください。

Azure Monitor データソースへの接続

Azure Monitor データソースは Azure クラウド内の複数のサービスをサポートしています。

- Azure Monitor サービスは、Azure リソースをモニタリングするための単一ソースを提供するプラットフォームサービスです。詳細については、「Azure Monitor サービスのクエリ」を参照してください。
- [Application Insights サーバー]は、複数のプラットフォーム上のウェブデベロッパー向けの拡張可能なアプリケーションパフォーマンス管理 (APM) サービスで、ライブウェブアプリケーションのモニタリングに使用できます。これによりパフォーマンスの異常が自動的に検出されます。詳細については、「Application Insights Analytics サービスのクエリ」を参照してください。

- Azure Log Analytics (または Azure Logs) では、Azure Monitor が収集したログデータにアクセスできます。詳細については、「Azure Log Analytics サービスのクエリ」を参照してください。
- [Application Insights Analytics サービス]を使用して、Azure Log Analytics で使用されているのと同じクエリ言語を使用し [Application Insights データ]をクエリします。詳細については、「Application Insights Analytics サービスのクエリ」を参照してください。

データソースの追加

データソースは 4 つの異なるサービスのメトリクスにアクセスできます。使用するサービスへのアクセスを設定できます。Azure Entra ID で設定した場合、複数のサービスに同じ認証情報を使用することもできます。

- Microsoft Entra アプリを登録し、サービスプリンシパルを作成する

1. Grafana のメインメニューからアクセスすると、新しくインストールされたデータソースをデータソースセクション内にすぐに追加できます。次に、右上の [データソースの追加] ボタンを選択します。Azure Monitor のデータソースが、データソースリストの クラウドセクションで選択できるようになります。
2. 名前フィールドでは、Grafana により `Azure Monitor` または `Azure Monitor - 3` などのデータソースの名前が自動的に入力されます。複数のデータソースを設定する場合は、情報を含んだ名前に変更します。
3. Azure Monitor を使用する場合は、Azure ポータルの 4 つの情報が必要です (詳細手順については、前述のリンクを参照してください)。
   - テナント ID (Azure Entra ID、プロパティ、ディレクトリ ID)
   - クライアント ID (Azure Entra ID、アプリ登録、アプリの選択、アプリケーション ID)
   - クライアントシークレット (Azure Entra ID、アプリ登録、アプリの選択、キー)
   - デフォルトのサブスクリプション ID (サブスクリプション、サブスクリプションの選択、概要、サブスクリプション ID)
4. これらの 4 つの項目を Azure Monitor API 詳細 セクション内のフィールドに貼り付けます。
   - サブスクリプション ID はクエリごとに変更できます。データソースを保存してページを更新し、指定のクライアント ID で使用できるサブスクリプションのリストを表示します。
5. Azure Log Analytics サービスも使用する場合は、これら 2 つの設定値を指定、または前のステップのクライアント ID とシークレットを再利用する必要があります。

- クライアント ID (Azure Entra ID、アプリ登録、アプリの選択、アプリケーション ID)
- クライアントシークレット (Azure Entra ID、アプリ登録、アプリの選択、キー、キーの作成、クライアントシークレットの使用)

6. Application Insights を使用する場合は、Azure Portal の 2 つの情報が必要です (詳細手順については、前述のリンクを参照してください)。
   - アプリケーション ID
   - API キー
7. これら 2 つの項目を Application Insights API 詳細 セクション内の適切なフィールドに貼り付けます。
8. [保存してテスト] ボタンを選択して、設定の詳細が正しいことをテストします。

または、ステップ 4 で新しい Azure Entra ID アプリを作成する場合は、Azure CLI を使用します。

```
az ad sp create-for-rbac -n "http://localhost:3000"
```

サービスの選択

パネルのクエリエディタで、ご使用の Azure Monitor データソースを選択した後の最初のステップは、サービスを選択することです。次の 4 つのオプションがあります。

- `Azure Monitor`
- `Application Insights`
- `Azure Log Analytics`
- `Insights Analytics`

選択したオプションに応じて、クエリエディタが異なります。Azure Monitor がデフォルトです。

Azure Monitor サービスのクエリ

Azure Monitor サービスには、実行しているすべての Azure サービスのメトリクスが用意されています。これは Azure 上のアプリケーションのパフォーマンスを把握して、アプリケーションに影響する問題を事前に検出するのに役立ちます。

Azure Monitor の認証情報を使用して複数のサブスクリプションにアクセスできる場合は、まず適切なサブスクリプションを選択します。

サービスから取得できるメトリクス例は次のとおりです。

- `Microsoft.Compute/virtualMachines - Percentage CPU`
- `Microsoft.Network/networkInterfaces - Bytes sent`
- `Microsoft.Storage/storageAccounts - Used Capacity`

クエリエディタを使用すると、複数のディメンションをサポートするメトリクスに対して、複数のディメンションをクエリできます。複数のディメンションをサポートするメトリクスは「Azure Monitor がサポートするメトリクスリスト」に記載されているものですが、メトリクスの [ディメンション] 列には 1 つ以上の値がリストされています。

Azure Monitor のエイリアスを使用した凡例キーの書式設定

Azure Monitor API のデフォルト凡例形式は次のとおりです。

```
metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}
```

これらの書式は長い場合がありますが、この書式設定はエイリアスを使用して変更できます。凡例形式 フィールドでは、次のエイリアスを任意の方法で組み合わせることができます。

Azure Monitor の例:

- `Blob Type: {{ blobtype }}`
- `{{ resourcegroup }} - {{ resourcename }}`

Azure Monitor のエイリアスパターン

- `{{ resourcegroup }}` = リソースグループの値に置き換え
- `{{ namespace }}` = 名前空間の値に置き換え (Microsoft.Compute/virtualMachines など)
- `{{ resourcename }}` = リソース名の値に置き換え
- `{{ metric }}` = メトリクス名に置き換え (Percentage CPU など)
- `{{ dimensionname }}` = 7.1 以降のレガシー (下位互換性用) は、最初のディメンションのキー/ラベル (キー/ラベルでソート) に置き換え (blobtype など)。
- `{{ dimensionvalue }}` = 7.1 以降のレガシー (下位互換性用) は、最初のディメンションの値 (キー/ラベルでソート) に置き換え (BlockBlob など)。

- {{ arbitraryDim }} = 7.1 以降で使用可能は、対応するディメンションの値に置き換え (例えば、{{ blobtype }} は BlockBlob になります)。

Azure Monitor のテンプレート変数の作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

Azure Monitor サービスは複数の値を未サポートであることに注意してください。複数の時系列 (server1 と server2 のメトリクスなど) を視覚化するには、複数のクエリを追加して同じグラフまたは同じテーブルで表示できるようにします。

Azure Monitor データソースプラグインには、変数編集ビューの [クエリ] フィールドで指定できる、以下のクエリが用意されています。これらを使用して、変数のオプションリストを入力できます。

名前	説明
`Subscriptions()`	説明のリストを返します。
`ResourceGroups()`	リソースグループのリストを返します。
`ResourceGroups(12345678-aaaa-bbbb-cccc-123456789aaa)`	指定のサブスクリプションのリソースグループのリストを返します。
`Namespaces(aResourceGroup)`	指定のリソースグループの名前空間のリストを返します。
`Namespaces(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup)`	指定のリソースグループとサブスクリプションの名前空間のリストを返します。
`ResourceNames(aResourceGroup, aNamespace)`	リソース名のリストを返します。
`ResourceNames(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace)`	指定のサブスクリプションのリソース名のリストを返します。

名前	説明
`MetricNamespace(aResourceGroup, aNamespace, aResourceName)`	メトリクス名前空間のリストを返します。
`MetricNamespace(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace, aResourceName)`	指定のサブスクリプションのメトリクス名前空間のリストを返します。
`MetricNames(aResourceGroup, aNamespace, aResourceName)`	メトリクス名のリストを返します。
`MetricNames(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace, aResourceName)`	指定のサブスクリプションのメトリクス名のリストを返します。

例:

- リソースグループのクエリ: `ResourceGroups()`
- メトリクス名変数で渡される: `Namespaces(cosmo)`
- テンプレート変数のチェーン: `ResourceNames($rg, $ns)`
- パラメータの引用なし: `MetricNames(hg, Microsoft.Network/publicIPAddresses, grafanaIP)`

テンプレート作成とテンプレート変数の詳細については、「テンプレート」を参照してください。

サポートされる Azure Monitor メトリクスのリスト

Azure Monitor API が返すすべてのメトリクスに値があるわけではありません。クエリ構築を容易にするため、Grafana データソースにはサポート対象の Azure Monitor メトリクスのリストがあり、値を持たないメトリクスは無視されます。新しいサービスとメトリクスが Azure クラウドに追加されると、このリストは定期的に更新されます。

Azure Monitor アラート

Grafana アラートは Azure Monitor サービスでサポートされています。これは Azure Alerts のサポートではありません。Grafana アラートの詳細については、「Grafana アラート」を参照してください。

Application Insights サービスのクエリ

Application Insights のエイリアスを使用した凡例キーの形式設定

デフォルトの凡例の形式設定は次のとおりです。

```
metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}
```

凡例形式フィールドでは、任意の方法で次のエイリアスを組み合わせることができます。

Application Insights の例 :

- `city: {{ client/city }}`
- `{{ metric }} [Location: {{ client/countryOrRegion }}, {{ client/city }}]`

Application Insights のエイリアスパターン

- `{{ groupbyvalue }}` = Grafana 7.1 以降のレガシー (下位互換性用) は、最初のディメンションのキー/ラベル (キー/ラベルでソート) に置き換え。
- `{{ groupbyname }}` = Grafana 7.1 以降のレガシー (下位互換性用) は、最初のディメンションの値 (キー/ラベルでソート) に置き換え (BlockBlob など)。
- `{{ metric }}` = メトリクス名に置き換え (リクエスト/カウントなど)
- `{{ arbitraryDim }}` = 7.1 以降で使用可能は、対応するディメンションの値に置き換え (例えば、`{{ client/city }}` は Chicago になります)。

Application Insights のフィルター式

フィルターフィールドでは OData フィルター式を使用できます。

例:

- `client/city eq 'Boydton'`
- `client/city ne 'Boydton'`
- `client/city ne 'Boydton' and client/city ne 'Dublin'`
- `client/city eq 'Boydton' or client/city eq 'Dublin'`

Application Insights の 変数を使用したテンプレート作成

変数編集ビューの [クエリ] フィールドで、次のいずれかのクエリを使用します。

テンプレート作成とテンプレート変数の詳細については、「テンプレート」を参照してください。

名前	説明
`AppInsightsMetricNames()`	メトリクス名のリストを返します。
`AppInsightsGroupBys(aMetricName)`	指定のメトリクス名の `group by` 句のリストを返します。

例:

- メトリクス名クエリ: `AppInsightsMetricNames()`
- メトリクス名変数で渡される: `AppInsightsGroupBys(requests/count)`
- テンプレート変数のチェーン: `AppInsightsGroupBys($metricnames)`

Application Insights アラート

Grafana アラートは Application Insights でサポートされます。これは Azure Alerts のサポートではありません。Grafana アラートの詳細については、「Grafana アラート」を参照してください。

Azure Log Analytics サービスのクエリ

クエリは新しい Azure Log Analytics (または KustoDB) クエリ言語で記述されます。Log Analytics クエリは時系列データまたはテーブルデータとして書式設定できます。

認証情報を使用して複数のサブスクリプションにアクセスできる場合は、クエリを入力する前に適切なサブスクリプションを選択します。

時系列クエリ

時系列クエリはグラフパネルおよび SingleStat パネルなどの他のパネル用です。各クエリには少なくとも日時列と数値列が含まれている必要があります。結果は日時列で昇順にソートする必要があります。

次のコード例は、時間別にグループ化された集計カウントを返すクエリを示しています。

```
Perf
| where $__timeFilter(TimeGenerated)
```

```
| summarize count() by bin(TimeGenerated, 1h)
| order by TimeGenerated asc
```

クエリには 1 つ以上の非数値/非日時列を含めることができ、それらの列はディメンションと見なされてレスポンスのラベルになります。例えば、時間、Computer、CounterName でグループ化された集計カウントを返すクエリです。

```
Perf
| where $__timeFilter(TimeGenerated)
| summarize count() by bin(TimeGenerated, 1h), Computer, CounterName
| order by TimeGenerated asc
```

追加の数値列も選択すできます (複数のディメンションの有無は問いません)。例えば、時間、Computer、CounterName 、InstanceName でカウントと平均値を取得する場合:

```
Perf
| where $__timeFilter(TimeGenerated)
| summarize Samples=count(), ["Avg Value"]=avg(CounterValue)
    by bin(TimeGenerated, $__interval), Computer, CounterName, InstanceName
| order by TimeGenerated asc
```

> **Note**
>
> ヒント: 前のクエリでは、列の名前の変更に Kusto 構文と `Samples=count()["Avg Value"]=...` が使用され、これはスペースに使用できる 2 番目の構文です。これにより Grafana が使用するメトリクス名が変更されます。その結果、シリーズの凡例やテーブル列などが指定内容と一致します。この例では、`_count` の代わりに `Samples` が表示されます。

テーブルクエリ

テーブルクエリは主にテーブルパネルで使用され、そこには列と行のリストが表示されます。この例では 6 つの指定列を含む行を返します。

```
AzureActivity
| where $__timeFilter()
```

```
| project TimeGenerated, ResourceGroup, Category, OperationName, ActivityStatus, Caller
| order by TimeGenerated desc
```

Log Analytics の表示名の形式設定

デフォルトの表示名の形式は以下のとおりです。

`metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}`

これは表示名フィールドのオプションを使用してカスタマイズできます。

Azure Log Analytics マクロ

クエリの書き込みを容易にするために、Grafana にはクエリの where 句で使用できるマクロがいくつか用意されています。

- `$__timeFilter()` – `TimeGenerated ≥ datetime(2018-06-05T18:09:58.907Z) and TimeGenerated ≤ datetime(2018-06-05T20:09:58.907Z)` まで拡張されますが、from と to の日時は Grafana タイムピッカーの日時です。
- `$__timeFilter(datetimeColumn)` – `datetimeColumn ≥ datetime(2018-06-05T18:09:58.907Z) and datetimeColumn ≤ datetime(2018-06-05T20:09:58.907Z)` まで拡張されますが、from と to の日時は Grafana タイムピッカーの日時です。
- `$__timeFrom()` – Grafana ピッカーの From 日時を返します。例えば、`datetime(2018-06-05T18:09:58.907Z)` などです。
- `$__timeTo()` – Grafana ピッカーの From 日時を返します。例えば、`datetime(2018-06-05T20:09:58.907Z)` などです。
- `$__escapeMulti($myVar)` – 不正な文字を含む複数値のテンプレート変数で使用されます。`$myVar` が文字列 `'\\grafana-vm\Network(eth0)\Total','\\hello!'` として次の 2 つの値を持つ場合、`@'\\grafana-vm\Network(eth0)\Total', @'\\hello!'` に拡張されます。単一値変数を使用する場合はこのマクロは必要ありません。`@'\$myVar'` の代わりに変数インラインをエスケープします。
- `$__contains(colName, $myVar)` – 複数値のテンプレート変数で使用されます。`$myVar` に値 `'value1','value2'` がある場合、`colName in ('value1','value2')` に拡張されます。

  [すべて] オプションを使用する場合は、[すべてのオプションを含める] チェックボックスをオンにし、[すべての値のカスタム化] フィールドに **all** の値を入力します。`$myVar` に `all` の値がある場合、マクロは代わりに `1 == 1` に拡張されます。テンプレート変数に多数のオプションがある場合、「where..in」句を大きくしないようにすると、クエリのパフォーマンスが向上します。

Azure Log Analytics 組み込み変数

一部の Grafana 変数では Azure Log Analytics クエリでも使用できます。

- `$__interval` - Grafana はクエリで時間別にグループ化するために使用できる最小時間粒度を計算します。これは、`5m` や `1h` などの時間粒度を返し、これは `summarize count() by bin(TimeGenerated, $__interval)` などの bin 関数で使用できます。間隔変数の詳細については、「間隔変数の追加」を参照してください。

Azure Log Analytics の変数を使用したテンプレート作成

値のリストを返す Log Analytics クエリは、変数編集ビューの [クエリ] フィールドで使用できます。Log Analytics にはワークスペースのリストを返す Grafana 関数も 1 つあります。

テンプレートとテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

名前	説明
`workspaces()`	デフォルトサブスクリプションのワークスペースのリストを返します。
`workspaces(12345678-aaaa-bbbb-cccc-123456789aaa)`	指定サブスクリプションのワークスペースのリストを返します (パラメータは引用符または引用符なしでもかまいません)。

次の表は変数クエリの例を示しています。

クエリ	説明
`subscriptions()`	Azure サブスクリプションのリストを返します。
`workspaces()`	デフォルトサブスクリプションのワークスペースのリストを返します。

クエリ	説明
`workspaces("12345678-aaaa-bbbb-cccc-123456789aaa")`	指定サブスクリプションのワークスペースのリストを返します。
`workspaces("$subscription")`	サブスクリプションパラメータのテンプレート変数を使用。
`workspace("myWorkspace").Heartbeat | distinct Computer`	仮想マシンのリストを返します。
`workspace("$workspace").Heartbeat | distinct Computer`	テンプレート変数を持つ仮想マシンのリストを返します。
`workspace("$workspace").Perf | distinct ObjectName`	Perf テーブルからオブジェクトのリストを返します。
`workspace("$workspace").Perf \| where ObjectName == "$object" | distinct CounterName`	Perf テーブルからメトリクス名のリストを返します。

次のコード例は変数を使用した時系列クエリを示しています。

```
Perf
| where ObjectName == "$object" and CounterName == "$metric"
| where TimeGenerated >= $__timeFrom() and TimeGenerated <= $__timeTo()
| where  $__contains(Computer, $computer)
| summarize avg(CounterValue) by bin(TimeGenerated, $__interval), Computer
| order by TimeGenerated asc
```

Grafana パネルから Azure Portal の Log Analytics クエリエディタへのディープリンク

パネルで時系列を選択すると、[Azure Portal で表示] へのリンクを含むコンテキストメニューが表示されます。そのリンクを選択すると Azure ポータルで Azure Log Analytics クエリエディタが開き、その Grafana パネルからクエリを実行します。

現在 Azure Portal にログインしていない場合は、そのリンクによって ログインページが開きます。指定リンクは任意のアカウントで有効ですが、クエリで指定された Azure Log Analytics ワークスペースにアクセスできるアカウントでのみクエリが表示されます。

Azure Log Analytics アラート

Grafana アラートは Application Insights でサポートされます。これは Azure Alerts のサポートではありません。Grafana ワークスペースのアラートの詳細については、「Grafana アラート」を参照してください。

Application Insights Analytics サービスのクエリ

サービスタイプを [Insights Analytics] に変更すると、Log Analytics サービスと似たエディタを使用できます。このサービスでは Kusto 言語も使用しているため、データのクエリ手順は Azure Log Analytics サービスのクエリ と同じです。ただし、代わりに Application Insights Analytics データをクエリする場合を除きます。

Graphite データソースに接続する

Grafana には、メトリクス領域をすばやくナビゲートしたり、関数を追加したり、関数パラメータを変更したりできる高度な Graphite クエリエディタがあります。エディタは、あらゆる種類のグラファイトクエリを処理できます。クエリリファレンスを使用して、複雑なネストされたクエリを処理することもできます。

Graphite 設定

Graphite 設定にアクセスするには、[設定] (歯車) アイコンを一時停止し、[データソース] を選択し、Graphite データソースを選択します。

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。

名前	説明
URL	graphite-web または graphite-api インストールの HTTP プロトコル、IP、ポート。
Access	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。
Auth	
Basic Auth	データソースへの基本認証を有効にします。
User	基本認証のユーザー名。
Password	基本認証のパスワード。
Custom HTTP Headers	[ヘッダーの追加] を選択して、カスタム HTTP ヘッダーを追加します。
Header	カスタムヘッダー名を入力します。
Value	カスタムヘッダー値を入力します。
Graphite details	
Version	Graphite のバージョンを選択します。
Type	Graphite のタイプを選択します。

アクセスモードは、データソースへのリクエストの処理方法を制御します。他に何も記述されていない場合は、サーバーが優先されます。

サーバーアクセスモード (デフォルト)

すべてのリクエストはブラウザから Amazon Managed Grafana に行われ、リクエストはデータソースに転送されるため、クロスオリジンリソース共有 (CORS) 要件を回避できます。このアクセスモードを選択する場合、URL は Amazon Managed Grafana からアクセス可能である必要があります。

ブラウザアクセスモード

Amazon Managed Grafana は、Graphite データソースのブラウザへの直接アクセスをサポートしていません。

Graphite クエリエディタ

Grafana には、クエリの構築に役立つ Graphite 固有のクエリエディタが含まれています。

Graphite に送信されるクエリの未加工テキストを表示するには、[テキスト編集モードの切り替え] (鉛筆) アイコンを選択します。

クエリするメトリクスの選択

メトリクス空間をナビゲートするには、[メトリクスを選択] を選択します。操作開始後は、ポインタまたはキーボードの矢印キーを引き続き使用できます。ワイルドカード文字を選択しても続行できます。

関数

関数を追加するには、[関数] の横にあるプラスアイコンを選択します。関数を検索するか、メニューから選択できます。関数を選択すると、その関数が追加され、最初のパラメータのテキストボックスを次に操作します。パラメータを選択するとテキストボックスになり、パラメータを編集または変更できます。- 関数を削除するには、関数名に続いて x アイコンを選択します。

`aliasByNode` などの一部の関数は、オプションの 2 番目の引数をサポートしています。引数を追加するには、最初の引数を一時停止し、表示される + 記号を選択します。2 番目のオプションパラメータを削除するには、パラメータを選択し、空白のままにします。続いて、エディタにより削除されます。

ラベルのソート

一貫した順序が必要な場合は、`sortByName` を使用します。これは、複数のグラフに同じラベルがあり、それらが別々にソートされ、異なる色を使用している場合に問題を生じる可能性があります。これを修正するには、`sortByName()` を使用します。

ネストされたクエリ

クエリは、クエリがある行の文字で参照できます (Microsoft Excel と同様) 。グラフに 2 番目のクエリを追加する場合は、「#A」と入力して最初のクエリを参照できます。これにより、複合クエリを簡単に作成できます。

ワイルドカード文字を使用して多くのクエリを回避する

場合によっては、同じグラフに複数の時系列をプロットしたいことがあります。例えば、マシンでの CPU の使用状況を確認することができます。最初に、`cpu.percent.user.g`、`cpu.percent.system.g` などの時系列ごとにクエリを追加してグラフを作成できます。これにより、データソースに対してクエリが n 件実行されますが、これは非効率的です。

より効率的にするために、検索でワイルドカード文字を使用して、1 つのクエリですべての時系列を返すことができます。例えば、`cpu.percent.*.g`。

テーブルまたはグラフのメトリクス名の変更

`alias` 関数を使用して、Grafana テーブルまたはグラフのメトリクス名を変更します。例えば、`aliasByNode()` または `aliasSub()` を使用します。

ポイント統合

すべての Graphite メトリクスは、Graphite がグラフ内のピクセル数よりも多くのデータポイントを返さないように統合されます。デフォルトでは、この統合は `avg` 関数を使用して行われます。Graphite consolidateBy 関数を追加することで、Graphite がメトリクスを統合する方法を制御できます。

> **Note**
>
> つまり、凡例の概要値 (最大、最小、合計) がすべて同時に正しくなることはありません。これらは Grafana によってクライアント側で計算されます。また、統合関数によっては、1 つまたは 2 つのみが同時に正しくなります。

時系列の組み合わせ

時系列を組み合わせるには、関数リストで結合を選択します。

データ探索とタグ

Graphite では、すべてがタグです。

データを探索するときは、以前に選択したタグを使用して残りの結果セットをフィルタリングします。データを選択するには、 `seriesByTag` 関数を使用します。この関数は、タグ式 (=、!=、=~、!=~) を使用して時系列をフィルタリングします。

タグを選択すると、Grafana クエリビルダーが自動的にこれを実行します。

> **Note**
>
> ヒント: 正規表現の検索は、高基数タグでは遅くなる可能性があるため、最初に他のタグを使用して範囲を縮小してみてください。特定の名前または名前空間から始めると、結果を減らすことができます。

テンプレート変数

メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

タグ値を使用して変数を作成するには、Grafana 関数 `tags` と `tag_values` を使用します。

クエリ	説明
`tags()`	すべてのタグを返します。
`tags(server=~backend\*)`	フィルター式に一致する連続したタグのみを返します。

クエリ	説明
`tag_values(server)`	指定されたタグのタグ値を返します。
`tag_values(server, server=~backend\*)`	これらの式に一致する一連の指定されたタグで発生するフィルター処理されたタグ値を返します。
`tag_values(server, server=~backend\*, app=~${apps:regex})`	複数のフィルター式と式には、他の変数を含めることができます。

詳細については、「タグのオートコンプリート API に関する Graphite ドキュメント」を参照してください。

クエリ変数

クエリフィールドで指定するクエリは、メトリクス検索タイプのクエリである必要があります。例えば、`prod.servers.*` などのクエリは、ワイルドカード位置に存在する可能性のあるすべての値で変数を埋めます。

定義で他の変数を使用するネストされた変数を作成することもできます。例えば、`apps.$app.servers.*` はクエリ定義で `$app` 変数を使用します。

`__searchFilter` を使用してクエリ変数の結果をフィルタリングする

クエリフィールドで `__searchFilter` を使用すると、ドロップダウン選択ボックスに入力した内容に基づいてクエリ結果がフィルタリングされます。何も入力しない場合、`__searchFilter` のデフォルト値は `*` で、正規表現の一部として使用する場合は `` です。

次の例は、__searchFilter をクエリフィールドの一部として使用してユーザーがドロップダウン選択ボックスにテキストを入力するときに server を検索できるようにします。

クエリ

```
apps.$app.servers.$__searchFilter
```

TagValues

```
tag_values(server, server=~${__searchFilter:regex})
```

変数の使用

変数は、メトリクスノードパスで、または関数のパラメータとして使用できます。

2 つの構文があります。

- $<varname> 例: apps.frontend.$server.requests.count
- ${varname} 例: apps.frontend.${server}.requests.count

なぜ 2 つの方法があるのですか？ 最初の構文は読み書きが容易ですが、単語の途中で変数を使用することは許可されません。my.server${serverNumber}.count などの式で 2 番目の構文を使用します。

タグクエリでの変数の使用

タグクエリの複数値変数は、変数 {var:regex} に対して Grafana 5.0 で導入された高度なフォーマット構文を使用します。タグ以外のクエリでは、複数値変数のデフォルトの glob 形式が使用されます。

次のコード例は、正規表現フォーマットで Equal Tilde 演算子 =~ を使用するタグ式を示しています。

```
server=~${servers:regex}
```

詳細については、「高度な変数フォーマットオプション」を参照してください。

注釈

注釈を使用すると、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニュー/注釈ビューを使用して注釈クエリを追加します。詳細については、「注釈」を参照してください。

Graphite は、注釈をクエリする 2 つの方法をサポートしています。

- 通常のメトリクスクエリ。この方法では、Graphite クエリテキストボックスを使用します。
- Graphite イベントクエリ。この方法では、`Graphite event tags` テキストボックスを使用し、タグまたはワイルドカード文字を指定します (空のままにしておいても機能します) 。

Google Cloud Monitoring データソースへの接続

> **Note**
>
> 以前のバージョンの Grafana では、このデータソースの名前は Google Stackdriver でした。

Google Cloud Monitoring メトリクスのダッシュボードを構築できるようにするため、Google Cloud Monitoring データソースを追加します。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. サイドメニューの [ダッシュボード] リンクの下に、[データソース] リンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [Google Cloud Monitoring] を選択します。
5. サービスアカウントのキーファイルにアップロードまたは貼り付けます。サービスアカウントのキーファイルを作成する手順については、このドキュメントの後半を参照してください。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されない場合、現在のユーザーには `Admin` ロールがありません。

名前	説明
`Name`	データソース名。これはパネルとクエリでデータソースを参照する方法です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Service Account Key`	GCP プロジェクトのサービスアカウントキーファイル。作成方法については、このドキュメントの後半の手順を参照してください。

認証

Google Cloud Monitoring プラグインの認証には 2 つの方法があります。

- Google JWT ファイルのアップロード
- Google メタデータサーバーから認証情報を自動的に取得

後者のオプションは、GCE 仮想マシンで Grafana を実行しているときにのみ使用できます。

Google サービスアカウントのキーファイルの使用

Google Cloud Monitoring API で認証するには、データを表示するプロジェクトの Google Cloud Platform (GCP) サービスのアカウントを作成する必要があります。Grafana データソースは 1 つの GCP プロジェクトと統合されます。複数の GCP プロジェクトのデータを視覚化するには、GCP プロジェクトごとに 1 つのデータソースを作成する必要があります。

API の有効化

最初に次の API を有効にする必要があります。

- Monitoring API
- Cloud Resource Manager API

一覧されているリンクを選択してから [有効化] ボタンを選択します。

プロジェクトの GCP サービスアカウントの作成

1. API とサービスの認証情報ページに移動します。

2. [認証情報の作成] ドロップダウン/ボタンを選択して [サービスアカウントキー] オプションを選択します。

   {{< docs-imagebox img="/img/docs/v71/cloudmonitoring_create_service_account_button.png" class="docs-image–no-shadow" caption="サービスアカウントの作成ボタン" >}}

3. [サービスアカウントキーの作成] ページで、キータイプ `JSON` を選択します。次に、[サービスアカウント] ドロップダウンリストで、[新しいサービスアカウント] オプションを選択します。

   {{< docs-imagebox img="/img/docs/v71/cloudmonitoring_create_service_account_key.png" class="docs-image–no-shadow" caption="サービスアカウントキーの作成" >}}

4. いくつかの新しいフィールドが表示されます。[サービスアカウント名] フィールドにサービスアカウントの名前を入力してから、[ロール] ドロップダウンリストから [モニタリングビューワー] ロールを選択します。

   {{< docs-imagebox img="/img/docs/v71/cloudmonitoring_service_account_choose_role.png" class="docs-image–no-shadow" caption="ロールを選択" >}}

5. [サーバーの作成] ボタンを選択します。JSON キーファイルが作成されてコンピュータにダウンロードされます。このファイルは Google Cloud Monitoring データへのアクセスを許可するためのものですから、安全な場所に保存してください。

6. データソース [設定] ページで Grafana にアップロードします。ファイルをアップロード、またはファイルの内容に貼り付けることができます。

   {{< docs-imagebox img="/img/docs/v71/cloudmonitoring_grafana_upload_key.png" class="docs-image–no-shadow" caption="サービスのキーファイルを Grafana にアップロード" >}}

7. ファイルの内容は暗号化されて Grafana データベースに保存されます。ファイルをアップロードした後は必ず保存してください。

   {{< docs-imagebox img="/img/docs/v71/cloudmonitoring_grafana_key_uploaded.png" class="docs-image–no-shadow" caption="サービスのキーファイルを Grafana にアップロード" >}}

クエリエディタの使用

Google Cloud Monitoring クエリエディタでは、メトリクスとサービスレベル目標 (SLO) の 2 種類のクエリを構築できます。どちらのタイプも時系列データを返します。

メトリクスクエリ

メトリクスクエリエディタでは、メトリクスを選択して、ラベルと時間ごとにグループ化/集計し、フィルターを使用して結果に含める時系列を指定できます。

メトリクスクエリを作成するには、以下の手順に従います。

1. クエリタイプのドロップダウンリストで オプションのメトリクスを選択します。
2. ドロップダウンリストから [プロジェクト] を選択します。
3. サービスのドロップダウンリストから Google Cloud Platform サービスを選択します。
4. メトリクスのドロップダウンリストからメトリクスを選択します。
5. 句によってフィルターやグループを追加または削除するには、フィルターおよびグループ化セクションのプラスとマイナスアイコンを使用します。この手順は省略可能です。

Google Cloud Monitoring メトリクスにはさまざまな種類 (GAUGE、DELTA、CUMULATIVE) があり、これらの種類ではさまざまな集約オプション (リデューサーとアライナー) がサポートされます。Grafana クエリエディタには選択したメトリクスで使用可能な集約方法のリストが表示され、メトリクスを選択するときにデフォルトのリデューサーとアライナーがセットされます。Y 軸の単位もクエリエディタが自動的に選択します。

フィルター

フィルターを追加するには、プラスアイコンを選択して、フィルタリングするフィールドを選択し、フィルター値を入力します。たとえば、`instance_name = grafana-1` と入力します。フィルターを削除するには、フィルター名を選択して `--remove filter--` を選択します。

単純なワイルドカード文字

演算子を = または != に設定すると、フィルター値のフィールドにワイルドカード文字を追加できます。例えば、`us-*` は「us-」で始まるすべての値をキャプチャし、`*central-a` は「central-a」で終わるすべての値をキャプチャします。`*-central-*` は、`central-` の部分文字列を持つすべての値をキャプチャします。単純なワイルドカード文字は正規表現よりも安価です。

正規表現

演算子を =~ または !=~ に設定すると、フィルター値のフィールドに正規表現を追加できます。例えば、`us-central[1-3]-[af]` は「us-central」で始まり、1 ~ 3 の範囲の数値、ダッシュ、「a」または「f」のいずれかが続くすべての値に一致します。正規表現を作成するときは、先頭のスラッシュと末尾のスラッシュは必要ありません。

集計

集計フィールドでは一般的な統計に基づいて時系列を組み合わせることができます。集計の詳細については、「集計オプション」を参照してください。

Aligner フィールドでは、同じグループの後に複数の時系列を時間間隔で配置できます。アライナーの詳細については、「アライメントメトリクスセレクタ」を参照してください。

アライメント期間と時間によるグループ化

集計が選択されている場合、Alignment Period はメトリクスを時間ごとにグループ化します。デフォルトでは GCP Google Cloud Monitoring のデフォルトグループを使用します (これにより、Grafana のグラフと Google Cloud Monitoring UI のグラフを比較できます)。そのオプションは cloud monitoring auto と呼ばれ、デフォルトは次のとおりです。

- 時間範囲が 23 時間未満の場合は 1m
- 時間範囲が 23 時間以上 6 日未満の場合は 5m
- 時間範囲が 6 日以上の場合は 1h

別の自動オプションは grafana auto です。このオプションでは、選択した時間範囲とグラフパネルの幅に応じて、自動的に時間ごとにグループが設定されます。詳細については、「間隔変数の追加」を参照してください。

また、1hや 1d など、グループ化する固定時間の間隔を選択することもできます。

グループ化の条件

リソースラベルまたはメトリクスラベルでグループ化して時系列の数を減らし、グループごとに結果を集計します。例えば、instance_name でグループ化してコンピューティングインスタンスの集計メトリクスを表示します。

メタデータラベル

リソースメタデータラベルには Google Cloud のリソースを一意に識別するための情報が含まれています。メタデータラベルは、時系列リクエストの [グループ化] セグメントの一部である場合にのみ、時系列レスポンスで返されます。メタデータラベルを取得するための API がないため、選択したサービスとメトリクスで使用できるメタデータラベルを使用して、ドロップダウンリストでグループに入力することはできません。ただし、[グループ化] フィールドのドロップダウンリストには、一般的なシステムラベルの定義済みリストが用意されています。

ユーザーラベルは事前に定義することはできませんが、[グループ化] フィールドには手動で入力できます。メタデータラベル、ユーザーラベル、またはシステムラベルが [グループ化] セグメントに含まれている場合は、それに基づいてフィルターを作成し、[エイリアス] フィールドでその値を展開できます。

エイリアスパターン

[エイリアス方法] フィールドを使用すると凡例キーの形式を制御できます。デフォルトではメトリクス名とラベルが表示されます。これは長くて読みにくくなる場合があります。エイリアスフィールドで次のパターンを使用して、お好みの方法で凡例キーをフォーマットできます。

メトリクスタイプのパターン

エイリアスパターン	説明	結果の例
`{{metric.type}}`	完全なメトリクスタイプを返します。	`compute.googleapis.com/instance/cpu/utilization`
`{{metric.name}}`	メトリクス名の部分を返します。	`instance/cpu/utilization`
`{{metric.service}}`	サービスのパートを返します。	`compute`

ラベルパターン

[グループ化] ドロップダウンリストには、メトリクスのリストとメトリクスのリソースラベルが表示されます。エイリアスパターンを使用すると、これらを凡例キーに含めることができます。

エイリアスパターン形式	説明	エイリアスパターンの例	結果の例
`{{metric.label.xxx}}`	メトリクスラベルの値を返します。	`{{metric.label.instance_name}}`	`grafana-1-prod`

エイリアスパターン形式	説明	エイリアスパターンの例	結果の例
{{resource.label.xxx}}	リソースラベルの値を返します。	{{resource.label.zone}}	us-east1-b
{{metadata.system_labels.xxx}}	メタデータシステムラベルの値を返します。	{{metadata.system_labels.name}}	grafana
{{metadata.user_labels.xxx}}	メタデータユーザーラベルの値を返します。	{{metadata.user_labels.tag}}	production

エイリアス方法の例: {{metric.type}} - {{metric.label.instance_name}}

結果の例: compute.googleapis.com/instance/cpu/usage_time - server1-prod

また、モニタリング対象のリソースタイプの名前を解決することもできます。

エイリアスパターン形式	説明	結果の例
{{resource.type}}	モニタリング対象のリソースタイプの名前を返します。	gce_instance

エイリアス方法の例: {{resource.type}} - {{metric.type}}

結果の例: gce_instance - compute.googleapis.com/instance/cpu/usage_time

SLO クエリ

> **Note**
>
> SLO クエリは Grafana v7.0 以降でのみ使用できます

Google Cloud Monitoring データソースの SLO クエリビルダーを使用すると、SLO データを時系列形式で表示できます。サービスモニタリングの基本概念を理解するには、Google Cloud Monitoring の公式ドキュメントを参照してください。

SLO クエリの作成

SLO クエリを作成するには、以下の手順に従います。

1. [クエリタイプ] ドロップダウンリストから [サービスレベル目標 (SLO)] オプションを選択します。
2. ドロップダウンリストから [プロジェクト] を選択します。
3. [サービス] ドロップダウンリストから [SLO サービス] を選択します。
4. [SLO] ドロップダウンリストから [SLO] を選択します。
5. [セレクタ] ドロップダウンリストから [時系列セレクタ] を選択します。

時系列セレクタのフレンドリ名が Grafana に表示されます。次の表は、サービスモニタリングのドキュメントで使用される、わかりやすい名前からシステム名へのマッピングを示しています。

セレクタのドロップダウンリストの値	使用する対応時系列セレクタ
SLI の値	select_slo_health
SLO コンプライアンス	select_slo_compliance
残りの SLO エラーバジェット	select_slo_budget_fraction

SLO クエリのエイリアスパターン

[エイリアス方法] フィールドを使用すると SLO クエリの凡例キーの形式を制御できます。

エイリアスパターン	説明	結果の例
`{{project}}`	GCP プロジェクト名を返します。	`myProject`
`{{service}}`	サービス名を返します。	`myService`
`{{slo}}`	SLO を返します。	`latency-slo`

エイリアスパターン	説明	結果の例
{{selector}}	セレクタを返します。	select_slo_health

SLO クエリのアライメント期間と時間によるグループ化

SLO クエリはメトリクスクエリと同じアライメント期間機能を使用します。詳細については、「メトリクスクエリ」を参照してください。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

クエリタイプの変数を使用すると、さまざまなタイプのデータに対して Google Cloud Monitoring をクエリできます。Google Cloud Monitoring データソースプラグインには次の Query Types が用意されています。

名前	説明
Metric Types	指定のサービスで使用できるメトリクスタイプ名のリストを返します。
Labels Keys	指定のテーブルの metric label および resource label のキーのリストを返します。
Labels Values	指定のメトリクス内のラベルの値のリストを返します。
Resource Types	指定のメトリクスのリソースタイプのリストを返します。
Aggregations	指定のメトリクスの集計のリスト (シリーズリデューサー間) を返します。
Aligners	指定のメトリクスのアライナーのリスト (シリーズアライナーごと) を返します。

名前	説明
`Alignment periods`	Grafana の Google Cloud Monitoring クエリエディタで使用できる、すべてのアライメント期間のリストを返します。
`Selectors`	SLO (サービスレベル目標) クエリで使用できるセレクタのリストを返します。
`SLO Services`	SLO クエリで使用できる Service Monitoring サービスのリストを返します。
`Service Level Objectives (SLO)`	指定の SLO サービスの SLO のリストを返します。

クエリでの変数の使用

2 つの構文があります。

- `$<varname>`例: `metric.label.$metric_label`
- `[[varname]]`例: `metric.label.[[metric_label]]`

なぜ 2 つの方法があるのですか？ 最初の構文は読み書きが容易ですが、単語の途中で変数を使用することは許可されません。[複数値]オプションまたは[すべての値を含める]オプションが有効になっている場合、Grafana はラベルをプレーンテキストから正規表現互換の文字列に変換します。つまり、= の代わりに =~ を使用する必要があります。

注釈

注釈を使用して、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニュー/注釈ビューを使用して注釈クエリを追加します。注釈のレンダリングは高価であるため、返される行数を制限することは重要です。Google Cloud Monitoring の注釈とイベントを表示はまだサポートされていませんが、Google Cloud Monitoring のカスタムメトリクスを使用するとうまく機能します。

注釈の詳細については、「注釈」を参照してください。

注釈のクエリエディタを使用すると、メトリクスとフィルターを選択できます。[タイトル] フィールドと [テキスト] フィールドはテンプレート化をサポートしており、クエリから返されるデータを使用できます。例えば、[タイトル] フィールドには次のテキストを含めることができます。

```
{{metric.type}} has value: {{metric.value}}
```

結果の例: `monitoring.googleapis.com/uptime_check/http_status has this value: 502`

注釈クエリエディタのパターン

エイリアスパターン形式	説明	エイリアスパターンの例	結果の例
`{{metric.value}}`	メトリクス/ポイントの値。	`{{metric.value}}`	555
`{{metric.type}}`	完全なメトリクスタイプを返します。	`{{metric.type}}`	`compute.googleapis.com/instance/cpu/utilization`
`{{metric.name}}`	メトリクス名の部分を返します。	`{{metric.name}}`	`instance/cpu/utilization`
`{{metric.service}}`	サービスのパートを返します。	`{{metric.service}}`	compute
`{{metric.label.xxx}}`	メトリクスラベルの値を返します。	`{{metric.label.instance_name}}`	grafana-1-prod
`{{resource.label.xxx}}`	リソースラベルの値を返します。	`{{resource.label.zone}}`	us-east1-b

Grafana パネルから Google Cloud コンソールのメトリクスエクスプローラーへのディープリンク

> **Note**
>
> この機能はメトリクスクエリでのみ使用できます。

パネルで時系列を選択すると、Google Cloud コンソールの Metrics Explorer で表示へのリンクを含むコンテキストメニューが表示されます。リンクを選択すると、Google Cloud コンソールで Metrics Explorer が開き、そこの Grafana パネルからクエリを実行します。リンクを使用すると、まず Google Account Chooser に移動します。アカウントの選択に成功すると、そのユーザーは Metrics Explorer にリダイレクトされます。指定リンクは任意のアカウントで有効ですが、クエリで指定された GCP プロジェクトにアクセスできるアカウントでのみクエリが表示されます。

InfluxDB ソースデータベースに接続するには

Grafana には、InfluxDB 用の機能豊富なデータソースプラグインが付属しています。プラグインにはカスタムクエリエディタが含まれており、注釈とクエリテンプレートをサポートしています。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. リンクの下にあるサイドメニューで、[ダッシュボード] に[データソース] という名前のリンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [InfluxDB] を選択します。
5. [Query Language] (クエリ言語) リストから InfluxQL または Flux を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

Jaeger データソースへの接続

Jaeger データソースにはオープンソースのエンドツーエンド分散トレースが用意されています。

データソースの追加

Jaeger 設定にアクセスするには、[設定] (歯車) アイコンを選択してから [データソース] を選択し、[Jaeger] を選択します。

名前	説明
Name	データソース名。これはパネル、クエリ、Explore でデータソースを表示する方法です。
Default	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
URL	Jaeger インスタンスの URL。例: `http://localhost:16686` 。
Access	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。
Basic Auth	Jaeger データソースへの基本認証を有効にします。
User	基本認証のユーザー名。
Passwo	基本認証のパスワード。

クエリのトレース

Explore を使用して Jaeger からのトレースをクエリし表示できます。詳細については、「Explore」を参照してください。

Jaeger クエリエディタを使用すると、トレース ID で直接クエリを実行、またはトレースセレクターからトレースを選択できます。トレース ID でクエリするには、ID をテキスト入力に挿入します。

トレースセレクタを使用して、Explore で選択した時間範囲に口グ記録されたすべてのトレースから特定のトレースを選択します。トレースセレクターには 3 つのレベルのネストがあります: 1. これはユーザーが関心を持つサービスです。1. 特定のオペレーションは選択済みのサービスの一部です。1. 選択したオペレーションが発生した特定のトレースであり、ルートオペレーション名とトレース期間で表されます。

ログからトレース ID へのリンク

Loki のログから Jaeger トレースにリンクするためには、内部リンクを使用して派生フィールドを設定します。詳細については、「派生フィールド」を参照してください。

Loki データソースに接続するには

Loki データソースは、Grafana のログ集約システムである Loki へのアクセスを提供します。

データソースの追加

1. Grafana ワークスペースを開き、ログインしていることを確認します。
2. [設定] リンクのサイドメニューに [データソース] リンクがあります。
3. 上部の [データソースの追加] ボタンを選択します。
4. データソースのリストから Loki を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

名前	説明
`Name`	データソース名。これはパネル、クエリ、Explore でデータソースを表示する方法です。
`Defaul`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`URL`	Loki インスタンスの URL。例: `http://localhost:3100` 。これは、Amazon EC2 ホストの URL、Amazon EKS クラスターの前の Application Load Balancer、または Loki インスタンスのその他の URL です。
`Maximu lines`	Loki によって返されるログの行数の上限 (デフォルトは 1000 行)。Explore でログを表示するときにブラウザが遅い場合は減らします。

派生フィールド

派生フィールドの設定を使用して、次のことを行うことができます。

- ログメッセージから解析されたフィールドを追加します。
- フィールドの値を使用するリンクを追加します。

この機能を使用して、ログから直接トレースバックエンドにリンクしたり、ログラインに userId がある場合はユーザープロファイルページにリンクしたりできます。これらのリンクはログの詳細に表示されます。詳細については、「ラベルと検出されたフィールド」を参照してください。

各派生フィールドは以下で構成されます。

- 名前 – ログの詳細にラベルとして表示されます。
- 正規表現 – ログメッセージで実行され、その一部を新しいフィールドの値としてキャプチャする正規表現パターン。1 つのキャプチャグループのみを含めることができます。
- URL/クエリ – リンクが外部である場合は、完全なリンク URL を入力します。リンクが内部リンクの場合、この入力はターゲットデータソースのクエリとして機能します。どちらの場合も、`${__value.raw }` マクロを使用してフィールドから値を補間できます。
- 内部リンク – リンクが内部か外部かを選択します。内部リンクの場合、データソースセレクタを使用すると、ターゲットデータソースを選択できます。データソースのトレースのみがサポートされています。

デバッグセクションを使用して、フィールドが抽出する内容と URL がどのように補間されるかを確認できます。[ログメッセージの例を表示] を選択して、ログメッセージを入力できるテキスト領域を表示します。

ログの詳細に表示されるリンクを含む新しいフィールド。

ログのクエリ

Loki からのログデータのクエリと表示は、Explorer 経由か、視覚化のログパネルで利用できます。Loki データソースを選択し、LogQL クエリを入力してログを表示します。LogQL の詳細については、「LogQL」を参照してください。

ログクエリ

ログクエリは、ログストリームセレクタと検索式 の 2 つの部分で構成されます。パフォーマンス上の理由から、まずログストリームのログラベルを選択する必要があります。

クエリフィールドの横にある Logs Explorer ([ログラベル] ボタン) には、使用可能なログストリームのラベルのリストが表示されます。クエリを記述する別の方法は、クエリフィールドの自動入力を使用することです。まず、左の中括弧 { を入力すると、オートコンプリートメニューにラベルのリストが表示されます。Enter キーを押してクエリを実行します。

結果が返されると、ログパネルにログ行のリストと棒グラフが表示されます。x 軸は時間、y 軸は頻度またはカウントを示します。

ログストリームセレクタ

クエリ式のラベル部分については、それを中括弧 {} で囲み、キー値の構文を使用してラベルを選択します。複数のラベル式はカンマで区切ります。

```
{app="mysql",name="mysql-backup"}
```

以下のラベルマッチングオペレーションは現在サポートされています。

- = 完全に等しい。
- != 等しくない。
- =~ 正規表現一致。
- !~ 正規表現に一致しない。

例:

- `{name=~"mysql.+"}`
- `{name!~"mysql.+"}`

ラベルセレクタを追加するもう 1 つの方法は、テーブルセクションにあります。ラベルの横にあるフィルターを選択して、クエリ式にラベルを追加します。これは複数のクエリでも機能します。複数のクエリの場合、各クエリにラベルセレクタを追加します。

Search 式

ログストリームセレクタを記述した後、検索式を記述することで、結果をさらにフィルタリングできます。検索式は、単なるテキスト式または正規表現にすることができます。

クエリの例:

- `{job="mysql"} |= "error"`

- `{name="kafka"} |~ "tsdb-ops.*io:2003"`
- `{instance=~"kafka-[23]",name="kafka"} != "kafka.server:type=ReplicaManager"`

フィルター演算子は連鎖でき、式を順次フィルタリングします。結果のログラインは、すべてのフィルターに適用されます。

例

```
{job="mysql"} |= "error" != "timeout"
```

現在、次のフィルタータイプがサポートされています。

- |= 行には文字列が含まれている。
- != 行に文字列が含まれていない。
- |~ 行が正規表現と一致する。
- !~ 行が正規表現パターンに一致しない。

> **Note**
>
> Loki のクエリ言語である LogQL の詳細については、「Loki LogQL」を参照してください。

ログのコンテキスト

上記の検索式を使用すると、フィルタリングされた結果を囲むコンテキストを取得できるようになりました。フィルタリングされた行の Show Context リンクを選択すると、関心のあるログメッセージの前後に発生したログメッセージを調査できます。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

注釈

非メトリクス Loki クエリは、注釈のソースとして使用できます。ログコンテンツは注釈テキストとして使用され、ログストリームラベルはタグとして使用されるため、追加のマッピングは必要ありません。

Microsoft SQL Server データソースに接続する

Microsoft SQL Server (MSSQL) データソースを使用して、Microsoft Azure SQL Database を含む Microsoft SQL Server 2005 以降からのデータをクエリおよび視覚化します。

> **Important**
>
> Grafana バージョン 8.0 は、Microsoft SQL Server、Postgres、MySQL のデータフレームの基盤となるデータ構造を変更します。その結果、時系列のクエリ結果が幅広い形式で返されます。詳細については、Grafana データフレームドキュメントの「ワイドフォーマット」を参照してください。
> 視覚エフェクトを以前と同じように機能させるには、手動による移行が必要になる場合があります。1 つの解決策は、Github の 「Postgres/MySQL/MSSQL: Breaking change in v8.0 related to time series queries and ordering of data column」に記載されています。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. リンク下のサイドメニューの [設定] に、[データソース] リンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [Microsoft SQL Server] を選択します。

データソースオプション

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。

名前	説明
Host	MSSQL インスタンスの IP アドレス/ホスト名とオプションのポート。ポートを省略すると、デフォルトのポート 1433 が使用されます。
Database	MSSQL データベースの名前。
User	データベースユーザーのログイン/ユーザー名。
Password	データベースユーザーのパスワード。
Encrypt	このオプションは、安全な SSL TCP/IP 接続をサーバーとネゴシエートするかどうか、またはどの程度までネゴシエートするかを決定します。デフォルトは `false` です(Grafana v5.4 以降)。
Max open	データベースへのオープン接続の最大数。デフォルトは `unlimited` です (Grafana v5.4 以降)。
Max idle	アイドル接続プールの最大接続数。デフォルトは 2 です (Grafana v5.4 以降)。
Max lifetime	接続を再利用できる秒単位の最大時間。デフォルトは `14400`/4 時間です。

最小時間間隔

`$_interval` `$_interval_ms` 変数の下限。データを 1 分ごとに書き込む場合は `1m` など、書き込み頻度に設定することをお勧めします。このオプションは、[Data Source] (データソース) オプションのダッシュボードパネルで上書き/設定することもできます。この値は、`1m` (1 分) や `30s` (30 秒) など、有効な時間識別子が続く数値としてフォーマットする必要があります。以下の時間識別子がサポートされています。

識別子	説明
y	年

識別子	説明
M	月
w	週
d	日
h	時間
m	分
s	秒
ms	ミリ秒

データベースユーザーのアクセス許可

> **Important**
>
> データソースを追加するときに指定するデータベースユーザーには、クエリする指定されたデータベースとテーブルに対してのみ SELECT アクセス許可を付与する必要があります。Grafana は、クエリが安全であることを検証しません。クエリには任意の SQL ステートメントを含めることができます。例えば、 `DELETE FROM user;` や`DROP TABLE user;` などのステートメントが実行されます。これを防ぐために、アクセス許可が制限された特定の MSSQL ユーザーを作成することを強くお勧めします。

次のコード例は、アクセス許可が制限された特定の MSSQL ユーザーの作成を示しています。

```
CREATE USER grafanareader WITH PASSWORD 'password'
GRANT SELECT ON dbo.YourTable3 TO grafanareader
```

ユーザーがパブリックロールから不要なアクセス許可を付与されていないことを確認します。

既知の問題

2008 や 2008R2 などの古いバージョンの Microsoft SQL Server を使用している場合は、暗号化を無効にしないと接続できないことがあります。可能な場合は、最適な互換性を得るために利用可能な最新のサービスパックを適用することをお勧めします。

クエリエディタ

MSSQL クエリエディタは、Graph、Singlestat、または Table パネルの編集モードの [メトリクス] タブにあります。パネルタイトルを選択し、[編集] を選択して編集モードに入ります。エディタでは、視覚化するデータを選択する SQL クエリを定義できます。

1. [Format as] `Time series` (Graph または Singlestat パネルなどで使用) または `Table` (Table パネルなどで使用) を選択します。
2. これは、SQL クエリを記述する実際のエディタです。
3. クエリエディタの下に MSSQL のヘルプセクションを表示します。
4. 実行された SQL クエリを表示します。クエリが正常に実行された後、最初に使用可能になります。
5. 追加のクエリエディターが表示される場所に、追加のクエリを追加します。

マクロ

構文を簡素化し、日付範囲フィルターなどの動的部分を許可するため、クエリにマクロを含めることができます。

マクロの例	説明
`$__time(dateColumn)`	列の名前を時間 に変更する式に置き換えられます。例えば、dateColumn が時間に置き換えられます。
`$__timeEpoch(dateColumn)`	DATETIME 列タイプを Unix タイムスタンプ

マクロの例	説明
	に変換し、名前を時間 に変更する式に置き換えられます。例えば、DATEDIFF(second, "1970-01-01", dateColumn) が時間に置き換えられます。
`$__timeFilter(dateColumn)`	指定された列名を使用して時間範囲フィルターに置き換えられます。例えば、dateColumn BETWEEN "2017-04-21T05:01:17Z" AND "2017-04-21T05:06:17Z"です。
`$__timeFrom()`	現在アクティブな時間選択の開始に置き換えられます。例えば、"2017-04-21T05:01:17Z" です。

マクロの例	説明
`$__timeTo()`	現在アクティブな時間選択の終了に置き換えられます。例えば、"2017-04-21T05:06:17Z" です。
`$__timeGroup(dateColumn,'5m'[, fillvalue])`	GROUP BY 句で使用できる式に置き換えられます。[fillValue] に [NULL] または[浮動値]を指定すると、時間範囲内の空のシリーズにその値が自動的に入力されます。例えば、CAST(ROUND(DATEDIFF(second, "1970-01-01", time_column)/300.0, 0) as bigint)*300 です。

マクロの例	説明
`$__timeGroup(dateColumn,'5m', 0)`	前述と同じですが、fill パラメーターがあるため、その系列の欠落しているポイントは grafana によって追加され、0 が値として使用されます。
`$__timeGroup(dateColumn,'5m', NULL)`	上記と同じですが、NULL は欠落ポイントの値として使用されます。
`$__timeGroup(dateColumn,'5m', previous)`	上記と同じですが、その系列の前の値がフィル値として使用されます。まだ値が見つからない場合は NULL が使用されます (Grafana 5.3 以降でのみ使用可能) 。

クエリエディタには、パネル編集モードでクエリが実行された後に表示される[生成 SQL] リンクがあります。これを選択すると、実行された未加工の補間 SQL 文字列が展開されて表示されます。

テーブルクエリ

クエリオプションが[Format as][テーブル]として設定されている場合、基本的に任意のタイプの SQL クエリを実行できます。テーブルパネルには、クエリが返す列と行の結果が自動的に表示されます。

次のコード例は、データベーステーブルを示しています。

```
CREATE TABLE [event] (
  time_sec bigint,
  description nvarchar(100),
  tags nvarchar(100),
)
```

```
CREATE TABLE [mssql_types] (
  c_bit bit, c_tinyint tinyint, c_smallint smallint, c_int int, c_bigint bigint,
 c_money money, c_smallmoney smallmoney, c_numeric numeric(10,5),
  c_real real, c_decimal decimal(10,2), c_float float,
  c_char char(10), c_varchar varchar(10), c_text text,
  c_nchar nchar(12), c_nvarchar nvarchar(12), c_ntext ntext,
  c_datetime datetime,  c_datetime2 datetime2, c_smalldatetime smalldatetime, c_date
 date, c_time time, c_datetimeoffset datetimeoffset
)

INSERT INTO [mssql_types]
SELECT
  1, 5, 20020, 980300, 1420070400, '$20000.15', '£2.15', 12345.12,
  1.11, 2.22, 3.33,
  'char10', 'varchar10', 'text',
  N'#nchar12#', N'#nvarchar12#', N'#text#',
  GETDATE(), CAST(GETDATE() AS DATETIME2), CAST(GETDATE() AS SMALLDATETIME),
 CAST(GETDATE() AS DATE), CAST(GETDATE() AS TIME), SWITCHOFFSET(CAST(GETDATE() AS
 DATETIMEOFFSET), '-07:00')
```

次のコード例は、クエリを示しています。

```
SELECT * FROM [mssql_types]
```

テーブルパネルの列名は、次のコード例に示すように、通常の AS SQL 列選択構文を使用して制御できます。

```
SELECT
  c_bit as [column1], c_tinyint as [column2]
FROM
  [mssql_types]
```

結果のテーブルパネル:

時系列クエリ

例えば Graph パネルで使用するために [Format as] を [時系列]に設定した場合、クエリは SQL 日時または Unix エポックを表す任意の数値データ型を返す time という名前の列が必要です。値列のメトリクス名として使用される metric という名前の列を返すことができます。time および metric を除くすべての列は、値列として扱われます。metric 列を省略すると、値列の名前がメトリクス名になります。複数の値列を選択でき、それぞれにメトリクスとして名前が付けられます。複数の値列と metric という名前の列を返す場合、この列はシリーズ名のプレフィックスとして使用されます。

時系列クエリの結果セットは、時間別にソートする必要があります。

次のコード例は、データベーステーブルを示しています。

```
CREATE TABLE [event] (
  time_sec bigint,
  description nvarchar(100),
  tags nvarchar(100),
)
```

```
CREATE TABLE metric_values (
  time datetime,
  measurement nvarchar(100),
  valueOne int,
  valueTwo int,
)

INSERT metric_values (time, measurement, valueOne, valueTwo) VALUES('2018-03-15
 12:30:00', 'Metric A', 62, 6)
INSERT metric_values (time, measurement, valueOne, valueTwo) VALUES('2018-03-15
 12:30:00', 'Metric B', 49, 11)
```

```
...
INSERT metric_values (time, measurement, valueOne, valueTwo) VALUES('2018-03-15
 13:55:00', 'Metric A', 14, 25)
INSERT metric_values (time, measurement, valueOne, valueTwo) VALUES('2018-03-15
 13:55:00', 'Metric B', 48, 10)
```

次のコード例は、1 列 value と 1 列 metric を示しています。

```
SELECT
  time,
  valueOne,
  measurement as metric
FROM
  metric_values
WHERE
  $__timeFilter(time)
ORDER BY 1
```

前述のクエリがグラフパネルで使用されると、Metric A と Metric B という名前の 2 つのシリーズが生成され、値 valueOne と valueTwo が time にプロットされます。

次のコード例は、複数の value 列を示しています。

```
SELECT
  time,
  valueOne,
  valueTwo
FROM
  metric_values
WHERE
  $__timeFilter(time)
ORDER BY 1
```

前述のクエリがグラフパネルで使用されると、Metric A と Metric B という名前の 2 つのシリーズが生成され、値 valueOne と valueTwo が time にプロットされます。

次のコード例は、$__timeGroup マクロの使用を示しています。

```
SELECT
```

```
  $__timeGroup(time, '3m') as time,
  measurement as metric,
  avg(valueOne)
FROM
  metric_values
WHERE
  $__timeFilter(time)
GROUP BY
  $__timeGroup(time, '3m'),
  measurement
ORDER BY 1
```

前述のクエリがグラフパネルで使用されると、Metric A と Metric B という名前の 2 つのシリーズが生成され、値 valueOne と valueTwo が time にプロットされます。3 分間のウィンドウに値がない 2 つのシリーズでは、その 2 つの行の間に線がレンダリングされます。右側のグラフがゼロに下がらないことがわかります。

次のコード例では、fill パラメータを 0 に設定した上での $__ timeGroup マクロの使用を示しています。

```
SELECT
  $__timeGroup(time, '3m', 0) as time,
  measurement as metric,
  sum(valueTwo)
FROM
  metric_values
WHERE
  $__timeFilter(time)
GROUP BY
  $__timeGroup(time, '3m'),
  measurement
ORDER BY 1
```

このクエリがグラフパネルで使用される場合、結果は Metric Aと Metric B という名前の 2 つのシリーズで、valueTwo の合計が time にプロットされます。3 分間のウィンドウで値が欠落しているシリーズの値は 0 になり、右側のグラフに表示されます。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして

表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

タイプ Query のテンプレート変数を追加すると、測定名、キー名、ドロップダウン選択ボックスとして表示されるキー値などを返すことができる MSSQL クエリを記述できます。

例えば、テンプレート変数クエリ設定でこのようなクエリを指定した場合、テーブル内の hostname 列のすべての値を含む変数を指定できます。

```
SELECT hostname FROM host
```

クエリは複数の列を返すことができ、Grafana はそれらの列からリストを自動的に作成します。例えば、次のクエリは、 hostname と hostname2 の値を含むリストを返します。

```
SELECT [host].[hostname], [other_host].[hostname2] FROM host JOIN other_host ON [host].
[city] = [other_host].[city]
```

もう 1 つのオプションは、キー/値変数を作成できるクエリです。クエリは、 __text と __value という名前の 2 つの列を返します。__text 列の値は一意である必要があります (一意でない場合は、最初の値が使用されます)。ドロップダウンリストのオプションには、わかりやすい名前をテキストとして、ID を値として指定できるテキストと値があります。hostname をテキスト、id を値とするクエリの例:

```
SELECT hostname __text, id __value FROM host
```

ネストされた変数も作成できます。例えば、region という名前の別の変数がある場合です。次に、ホスト変数に、このようなクエリを含む、現在選択されているリージョンからのホストのみを表示させることができます (region が複数値変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します) 。

```
SELECT hostname FROM host WHERE region IN ($region)
```

クエリでの変数の使用

> **Note**
>
> テンプレート変数の値は、テンプレート変数が multi-value の場合にのみ引用符で囲まれます。

変数が複数値の変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します。

2 つの構文があります。

$<varname> hostname という名前のテンプレート変数の例:

```
SELECT
  atimestamp time,
  aint value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in($hostname)
ORDER BY atimestamp
```

[[varname]] hostname という名前のテンプレート変数の例:

```
SELECT
  atimestamp as time,
  aint as value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in([[hostname]])
ORDER BY atimestamp
```

複数値変数の引用をオフにする

Grafana は、複数値変数の引用符で区切られたカンマ区切り文字列を自動的に作成します。例えば、server01 と server02 を選択した場合、'server01', 'server02' の形式になります。引用を無効にするには、変数に csv フォーマットオプションを使用します。

${servers:csv}

変数形式オプションの詳細については、「テンプレートおよび変数」を参照してください。

注釈

注釈を使用して、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニュー/注釈ビューを使用して注釈クエリを追加します。詳細については、「注釈」を参照してください。

列

名前	説明
time	日付/時刻フィールドの名前。ネイティブ SQL 日付/時刻データ型またはエポック値を持つ列にすることができます。
timeend	終了日時フィールドのオプション名。ネイティブ SQL 日付/時刻データ型またはエポック値を持つ列にすることができます。
text	イベントの説明フィールド。
tags	カンマ区切り文字列としてイベントタグに使用するオプションのフィールド名。

次のコード例は、データベーステーブルを示しています。

```
CREATE TABLE [events] (
  time_sec bigint,
  description nvarchar(100),
  tags nvarchar(100),
)
```

また、時系列クエリ で定義されているデータベーステーブルも使用します。

次のコード例は、エポック値を持つ時間列を使用したクエリを示しています。

```
SELECT
  time_sec as time,
  description as [text],
  tags
FROM
  [events]
WHERE
  $__unixEpochFilter(time_sec)
ORDER BY 1
```

次のコード例は、エポック値を持つ時間列と timeend 列を使用するリージョンクエリを示しています。

```
SELECT
  time_sec as time,
  time_end_sec as timeend,
  description as [text],
  tags
FROM
  [events]
WHERE
  $__unixEpochFilter(time_sec)
ORDER BY 1
```

次のコード例は、ネイティブ SQL 日付/時刻データ型の時刻列を使用したクエリを示しています。

```
SELECT
  time,
  measurement as text,
  convert(varchar, valueOne) + ',' + convert(varchar, valueTwo) as tags
FROM
  metric_values
WHERE
  $__timeFilter(time_column)
ORDER BY 1
```

ストアドプロシージャのサポート

ストアドプロシージャが機能することが確認されています。ただし、予想どおりに機能しないエッジケースがあるかもしれません。ストアドプロシージャは、前述の各セクションで説明したのと同じ形

式で列の名前を使用し、データを返す限り、テーブル、時系列、および注釈クエリでサポートされています。

マクロ関数は、ストアドプロシージャ内では機能しません。

例

次の例では、データベーステーブルは時系列クエリで定義されます。列 valueOne、valueTwo、measurement のすべての組み合わせなど、グラフパネル内の 4 つのシリーズを視覚化するとします。右側のグラフパネルは、達成したいことを視覚化します。これを解決するには、次の 2 つのクエリを使用する必要があります。

次のコード例は、1 つ目のクエリを示しています。

```
SELECT
  $__timeGroup(time, '5m') as time,
  measurement + ' - value one' as metric,
  avg(valueOne) as valueOne
FROM
  metric_values
WHERE
  $__timeFilter(time)
GROUP BY
  $__timeGroup(time, '5m'),
  measurement
ORDER BY 1
```

次のコード例は、2 つ目のクエリを示しています。

```
SELECT
  $__timeGroup(time, '5m') as time,
  measurement + ' - value two' as metric,
  avg(valueTwo) as valueTwo
FROM
  metric_values
GROUP BY
  $__timeGroup(time, '5m'),
  measurement
ORDER BY 1
```

エポック形式の時間を使用したストアドプロシージャ

上記のようなグラフパネルで 4 つのシリーズをレンダリングするために、必要なすべてのデータを返すストアドプロシージャを定義できます。この場合、ストアドプロシージャは、int データ型の 2 つのパラメータ @from と @to を受け入れます。これは、ストアドプロシージャから返されるデータをフィルタリングするために使用される エポック形式の時間範囲 (開始から終了まで) である必要があります。

これは、select および group by 式の $__timeGroup(time, '5m') を模倣するため、多数の長い式が必要です。これらは、必要に応じて MSSQL 関数に抽出できます。

```
CREATE PROCEDURE sp_test_epoch(
  @from int,
  @to   int
)   AS
BEGIN
  SELECT
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, DATEADD(second,
 DATEDIFF(second,GETDATE(),GETUTCDATE()), time))/600 as int)*600 as int) as time,
    measurement + ' - value one' as metric,
    avg(valueOne) as value
  FROM
    metric_values
  WHERE
    time >= DATEADD(s, @from, '1970-01-01') AND time <= DATEADD(s, @to, '1970-01-01')
  GROUP BY
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, DATEADD(second,
 DATEDIFF(second,GETDATE(),GETUTCDATE()), time))/600 as int)*600 as int),
    measurement
  UNION ALL
  SELECT
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, DATEADD(second,
 DATEDIFF(second,GETDATE(),GETUTCDATE()), time))/600 as int)*600 as int) as time,
    measurement + ' - value two' as metric,
    avg(valueTwo) as value
  FROM
    metric_values
  WHERE
    time >= DATEADD(s, @from, '1970-01-01') AND time <= DATEADD(s, @to, '1970-01-01')
  GROUP BY
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, DATEADD(second,
 DATEDIFF(second,GETDATE(),GETUTCDATE()), time))/600 as int)*600 as int),
```

```
    measurement
  ORDER BY 1
END
```

それに続き、グラフパネルに次のクエリを使用できます。

```
DECLARE
  @from int = $__unixEpochFrom(),
  @to int = $__unixEpochTo()

EXEC dbo.sp_test_epoch @from, @to
```

日時形式の時刻を使用したストアドプロシージャ

上記のようなグラフパネルで 4 つのシリーズをレンダリングするために、必要なすべてのデータを返すストアドプロシージャを定義できます。この場合、ストアドプロシージャは、datetime データ型の 2 つのパラメータ @from と @to を受け入れます。これは、ストアドプロシージャから返されるデータをフィルタリングするために使用される時間範囲 (開始から終了まで) である必要があります。

これは、select および group by 式の $__timeGroup(time, '5m') を模倣するため、多数の長い式が必要です。これらは、必要に応じて MSSQL 関数に抽出できます。

```
CREATE PROCEDURE sp_test_datetime(
  @from datetime,
  @to   datetime
)   AS
BEGIN
  SELECT
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, time)/600 as int)*600 as int) as time,
    measurement + ' - value one' as metric,
    avg(valueOne) as value
  FROM
    metric_values
  WHERE
    time >= @from AND time <= @to
  GROUP BY
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, time)/600 as int)*600 as int),
    measurement
```

```
  UNION ALL
  SELECT
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, time)/600 as int)*600 as int) as time,
    measurement + ' - value two' as metric,
    avg(valueTwo) as value
  FROM
    metric_values
  WHERE
    time >= @from AND time <= @to
  GROUP BY
    cast(cast(DATEDIFF(second, {d '1970-01-01'}, time)/600 as int)*600 as int),
    measurement
  ORDER BY 1
END
```

それに続き、グラフパネルに次のクエリを使用できます。

```

DECLARE
  @from datetime = $__timeFrom(),
  @to datetime = $__timeTo()

EXEC dbo.sp_test_datetime @from, @to
```

[アラート]

時系列クエリは、アラート条件で機能します。テーブル形式のクエリは、アラートルール条件ではまだサポートされていません。

MySQL データソースに接続する

MySQL データソースを追加して、MySQL 互換データベースからデータをクエリおよび視覚化できるようにします。

> **Important**
>
> Grafana バージョン 8.0 では、MySQL 、Postgres、Microsoft SQL Server データソースのデータフレームの基盤となるデータ構造が変更されます。その結果、時系列のクエリ結果が幅広い形式で返されます。詳細については、Grafana データフレームドキュメントの「ワイドフォーマット」を参照してください。

視覚エフェクトを以前と同じように機能させるには、手動による移行が必要になる場合があります。1 つの解決策は、Github の「Postgres/MySQL/MSSQL: Breaking change in v8.0 related to time series queries and ordering of data column」に記載されています。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. [ダッシュボード] リンクのサイドメニューには、[データソース] という名前のリンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [MySQL] を選択します。

データソースオプション

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Host`	MySQL インスタンスの IP アドレス/ホスト名とオプションのポート。
`Database`	MySQL データベースの名前。
`User`	データベースユーザーのログイン/ユーザー名。
`Password`	データベースユーザーのパスワード。
`Max open`	データベースへのオープン接続の最大数。デフォルトは `unlimited` です (Grafana v5.4 以降)。

名前	説明
`Max idle`	アイドル接続プールの最大接続数。デフォルトは 2 です (Grafana v5.4 以降)。
`Max lifetime`	接続を再利用できる秒単位の最大時間。デフォルトは `14400`/4 時間です。これは常に MySQL (Grafana v5.4 以降) で設定された [wait_timeout] よりも小さくする必要があります。

最小時間間隔

`$_interval` `$_interval_ms` 変数の下限。データを 1 分ごとに書き込む場合は `1m` など、書き込み頻度に設定することをお勧めします。このオプションは、[Data Source] (データソース) オプションのダッシュボードパネルで上書き/設定することもできます。この値は、`1m` (1 分) や `30s` (30 秒) など、有効な時間識別子が続く数値としてフォーマットする必要があります。以下の時間識別子がサポートされています。

識別子	説明
`y`	年
`M`	月
`w`	週
`d`	日
`h`	時間
`m`	分
`s`	秒
`ms`	ミリ秒

データベースユーザーのアクセス許可

> **Important**
>
> データソースを追加するときに指定するデータベースユーザーには、クエリ対象の指定データベースとテーブルに対する SELECT アクセス許可のみを付与する必要があります。Grafana は、クエリが安全であることを検証しません。クエリには任意の SQL ステートメントを含めることができます。例えば、`USE otherdb;` や`DROP TABLE user;` などのステートメントが実行されます。これを防ぐために、アクセス許可が制限された特定の MySQL ユーザーを作成することを強くお勧めします。

次のコード例は、アクセス許可が制限された特定の MySQL ユーザーの作成を示しています。

```
CREATE USER 'grafanaReader' IDENTIFIED BY 'password';
GRANT SELECT ON mydatabase.mytable TO 'grafanaReader';
```

より多くのデータベースやテーブルへのアクセスを許可するには、必要に応じてデータベースやテーブルの代わりにワイルドカード文字 (*) を使用できます。

クエリエディタ

MySQL クエリエディタは、パネルの [編集] モードの [メトリクス] タブにあります。パネルタイトルを選択して [編集] モードに入り、次に [編集] を選択します。

クエリエディタには、パネル編集モードでクエリが実行された後に表示される[生成 SQL] リンクがあります。これを選択すると、実行された未加工の補間 SQL 文字列が展開されて表示されます。

テーブル、時間列、メトリクス列 (FROM) を選択する

初めて編集モードに入るか新しいクエリを追加したとき、Grafana はタイムスタンプ列と数値列を持つ最初のテーブルをクエリビルダーに事前入力しようとします。

FROM フィールドでは、Grafana は設定されたデータベースにあるテーブルを提案します。データベースユーザーがアクセスできる別のデータベースでテーブルまたはビューを選択するには、`otherDb.metrics` などの完全修飾名 (database.table) を手動で入力できます。

Time 列フィールドは、時間値を保持する列の名前を指します。メトリクス列フィールドの値の選択はオプションです。値を選択すると、メトリクス列フィールドがシリーズ名として使用されます。

メトリクス列の提案には、テキストデータ型 (テキスト、小テキスト、中テキスト、ロングテキスト、varchar、char) の列のみが含まれます。メトリクス列として異なるデータ型を持つ列を使用する場合は、キャスト CAST(numericColumn as CHAR) を使用して列名を入力できます。CONCAT(column1, " ", CAST(numericColumn as CHAR)) などのテキストデータ型に評価される任意の SQL 式をメトリクス列フィールドに入力することもできます。

列と集計関数 (SELECT)

SELECT 行では、使用する列と関数を指定できます。列フィールドでは、column1 * column2 / column3 などの列名の代わりに任意の式を記述できます。

集計関数を使用する場合は、結果セットをグループ化する必要があります。集計関数を追加すると、エディタは自動的に GROUP BY time を追加します。

さらに値列を追加するには、プラスボタンを選択し、メニューから Column を選択します。グラフパネルには、複数の値列が別々のシリーズとしてプロットされます。

データのフィルタリング (WHERE)

フィルターを追加するには、WHERE 条件の右側にあるプラスアイコンを選択します。フィルターを削除するには、対象のフィルター、Remove を続けて選択します。現在選択されている時間範囲のフィルターが、新しいクエリに自動的に追加されます。

グループ化の条件

時間または他の列でグループ化するには、GROUP BY 行の末尾にあるプラスアイコンを選択します。提案ドロップダウンリストには、現在選択されているテーブルのテキスト列のみが表示されますが、手動で任意の列を入力できます。項目を選択し、Remove を選択すると、グループを削除できます。

グループ化を追加する場合、選択したすべての列に集計関数を適用する必要があります。グループ化を追加すると、クエリビルダーは集計関数なしですべての列に集計関数を自動的に追加します。

ギャップ埋め

Grafana では、時間別にグループ化すると、欠損値を入力できます。時間関数は 2 つの引数を受け入れます。最初の引数はグループ化する時間枠で、2 つ目の引数は Grafana が不足している項目を埋める値です。

テキストエディタモード (raw)

ハンバーガーアイコン (三) を選択し、[スイッチエディタモード]を選択するか、クエリの下にある [SQL の編集] を選択して、未加エクエリエディタモードに切り替えることができます。

> **Note**
>
> 未加エクエリエディタを使用する場合は、クエリに少なくとも `ORDER BY time` と、返された時間範囲のフィルターがあることを確認してください。

マクロ

構文を簡素化し、日付範囲フィルターなどの動的部分を許可するため、クエリにマクロを含めることができます。

マクロの例	説明
`$__time(dateColumn)`	UNIX タイムスタンプに変換し、列の名前を `time_sec` に変更する式に置き換えられます。例えば、UNIX_TIMESTAMP (dateColumn) を time_sec として指定します。
`$__timeEpoch(dateColumn)`	UNIX タイムスタンプに変換し、列の名前を `time_sec` に変更する式に置き換えら

マクロの例	説明
	れます。例えば、UNIX_TIMESTAMP (dateColumn) を time_sec として指定します。
$__timeFilter(dateColumn)	指定された列名を使用して時間範囲フィルターに置き換えられます。例えば、dateColumn BETWEEN FROM_UNIXTIME(1494410783) AND FROM_UNIXTIME(1494410983) などです。
$__timeFrom()	現在アクティブな時間選択の開始に置き換えられます。例えば、FROM_UNIXTIME(1494410783) などです。

マクロの例	説明
`$__timeTo()`	現在アクティブな時間選択の終了に置き換えられます。例えば、FROM_UNIXTIME(1494410983) などです。
`$__timeGroup(dateColumn,'5m')`	GROUP BY 句で使用できる式に置き換えられます。例えば、cast(cast(UNIX_TIMESTAMP(dateColumn)/(300) as signed)300 as signed),* などです
`$__timeGroup(dateColumn,'5m', 0)`	前の行と同じですが、fill パラメーターがあるため、その系列の欠落しているポイントは grafana によって追加され、0 が値として使用されます。

マクロの例	説明
`$__timeGroup(dateColumn,'5m', NULL)`	上記と同じですが、NULL は欠落ポイントの値として使用されます。
`$__timeGroup(dateColumn,'5m', previous)`	上記と同じですが、その系列の前の値がフィル値として使用されます。まだ値が見つからない場合は NULL が使用されます (Grafana 5.3 以降でのみ使用可能) 。
`$__timeGroupAlias(dateColumn,'5m')`	$__timeGroup と同じように置き換えられますが、列エイリアスが追加されます (Grafana 5.3 以降でのみ使用可能) 。

マクロの例	説明
`$__unixEpochFilter(dateColumn)`	指定された列名を使用して、Unix タイムスタンプとして表される時間を使用して、時間範囲フィルターに置き換えられます。例えば、`dateColumn > 1494410783 AND dateColumn < 1494497183` 。
`$__unixEpochFrom()`	現在アクティブな時間選択の開始が Unix タイムスタンプに置き換えられます。例えば、`1494410783` 。
`$__unixEpochTo()`	現在アクティブな時間選択の最後が Unix タイムスタンプに置き換えられます。例えば、`1494497183` 。

マクロの例	説明
$__unixEpochNanoFilter(dateColumn)	指定された列名とナノ秒タイムスタンプとして表される時刻を使用して、時間範囲フィルターに置き換えられます。例えば、dateColumn > 1494410783152415214 AND dateColumn < 1494497183142514872 。
$__unixEpochNanoFrom()	現在アクティブな時間選択の開始がナノ秒タイムスタンプに置き換えられます。例えば、1494410783152415214 。

マクロの例	説明
`$__unixEpochNanoTo()`	現在アクティブな時間選択の終了がナノ秒タイムスタンプに置き換えられます。例えば、1494497183142514872 。
`$__unixEpochGroup(dateColumn,"5m", [fillmode])`	`$__timeGroup` と同様ですが、Unix タイムスタンプとして保存される時刻に置き換えられます (Grafana 5.3 以降でのみ使用可能)。
`$__unixEpochGroupAlias(dateColumn,"5m", [fillmode])`	上記と同じですが、列エイリアスも追加されます (Grafana 5.3 以降でのみ使用可能)。

クエリエディタには、パネル編集モードでクエリが実行された後に表示される [生成 SQL] リンクがあります。これを選択すると、実行された未加工の補間 SQL 文字列が展開されて表示されます。

テーブルクエリ

[Format as] クエリオプションが[テーブル]に設定されている場合、基本的に任意のタイプの SQL クエリを実行できます。テーブルパネルには、クエリが返す列と行の結果が自動的に表示されます。

次のコードは、クエリの例を示しています。

```
SELECT
  title as 'Title',
  user.login as 'Created By' ,
  dashboard.created as 'Created On'
 FROM dashboard
INNER JOIN user on user.id = dashboard.created_by
WHERE $__timeFilter(dashboard.created)
```

テーブルパネルの列名は、通常の as SQL 列選択構文を使用して制御できます。

時系列クエリ

例えばグラフパネルで使用するために [Format as] を [時系列]に設定した場合、クエリは SQL 日時または Unix エポックを表す任意の数値データ型を返す time という名前の列を返す必要があります。time および metric を除くすべての列は、値列として扱われます。値列のメトリクス名として使用される metric という名前の列を返すことができます。複数の値列と metric という名前の列を返す場合、この列はシリーズ名のプレフィックスとして使用されます (Grafana 5.3 以降でのみ使用可能) 。

時系列クエリの結果セットは、時間別にソートする必要があります。

次のコード例は、metric 列を示しています。

```
SELECT
  $__timeGroup(time_date_time,'5m'),
  min(value_double),
  'min' as metric
FROM test_data
WHERE $__timeFilter(time_date_time)
GROUP BY time
ORDER BY time
```

次のコード例は、$__timeGroup マクロの fill パラメータを使用して NULL 値をゼロに変換する方法を示しています。

```
SELECT
  $__timeGroup(createdAt,'5m',0),
  sum(value_double) as value,
  measurement
```

```
FROM test_data
WHERE
  $__timeFilter(createdAt)
GROUP BY time, measurement
ORDER BY time
```

次のコード例は、複数の列を示しています。

```
SELECT
  $__timeGroup(time_date_time,'5m'),
  min(value_double) as min_value,
  max(value_double) as max_value
FROM test_data
WHERE $__timeFilter(time_date_time)
GROUP BY time
ORDER BY time
```

時間範囲とパネル幅に基づく時間ごとの動的グループはサポートされていません。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレート」を参照してください。

クエリ変数

タイプ Query のテンプレート変数を追加すると、測定名、キー名、ドロップダウン選択ボックスとして表示されるキー値などを返すことができる MySQL クエリを記述できます。

例えば、テンプレート変数クエリ設定でこのようなクエリを指定した場合、テーブル内の hostname 列のすべての値を含む変数を指定できます。

```
SELECT hostname FROM my_host
```

クエリは複数の列を返すことができ、Grafana はそれらの列からリストを自動的に作成します。例えば、次のクエリは、 hostname と hostname2 の値を含むリストを返します。

```
SELECT my_host.hostname, my_other_host.hostname2 FROM my_host JOIN my_other_host ON
 my_host.city = my_other_host.city
```

クエリで $__timeFilter(column) などの時間範囲依存マクロを使用するには、テンプレート変数の更新モードを [時間範囲変更] に設定する必要があります。

```
SELECT event_name FROM event_log WHERE $__timeFilter(time_column)
```

もう 1 つのオプションは、キー/値変数を作成できるクエリです。クエリは、 __text と __value という名前の 2 つの列を返します。__text 列値は一意である必要があります (一意でない場合は、最初の値が使用されます) 。ドロップダウンリストのオプションにはテキストと値があるため、わかりやすい名前のテキストとして、ID を値として使用できます。

次のコード例は、 hostname をテキストとして、 id を値として持つクエリを示しています。

```
SELECT hostname AS __text, id AS __value FROM my_host
```

ネストされた変数も作成できます。例えば、region という名前の別の変数がある場合です。次に、ホスト変数に、このようなクエリを含む、現在選択されているリージョンからのホストのみを表示させることができます (region が複数値変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します) 。

```
SELECT hostname FROM my_host  WHERE region IN($region)
```

__searchFilter を使用してクエリ変数で結果をフィルタリングする

クエリフィールドで __searchFilter を使用すると、ドロップダウン選択ボックスのユーザータイプに基づいてクエリ結果がフィルタリングされます。ユーザーによって何も入力されていない場合、__searchFilter のデフォルト値は % です。

> **Note**
>
> Grafana はこれを行わないため、__searchFilter 式を引用符で囲むことが重要です。

次の例は、__searchFilter をクエリフィールドの一部として使用してユーザーがドロップダウン選択ボックスに入力するときに hostname を検索できるようにします。

```
SELECT hostname FROM my_host  WHERE hostname LIKE '$__searchFilter'
```

クエリでの変数の使用

Grafana 4.3.0 から 4.6.0 まで、テンプレート変数は常に自動的に引用符で囲まれるため、文字列値の場合は、その変数を引用符で囲まないでください。

Grafana 4.7.0 以降、テンプレート変数の値は、テンプレート変数が multi-value の場合にのみ引用符で囲まれます。

変数が複数値の変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します。

2 つの構文があります。

$<varname> hostname という名前のテンプレート変数の例:

```
SELECT
  UNIX_TIMESTAMP(atimestamp) as time,
  aint as value,
  avarchar as metric
FROM my_table
WHERE $__timeFilter(atimestamp) and hostname in($hostname)
ORDER BY atimestamp ASC
```

[[varname]] hostname という名前のテンプレート変数の例:

```
SELECT
  UNIX_TIMESTAMP(atimestamp) as time,
  aint as value,
  avarchar as metric
FROM my_table
WHERE $__timeFilter(atimestamp) and hostname in([[hostname]])
ORDER BY atimestamp ASC
```

複数値変数の引用をオフにする

Grafana は、複数値変数の引用符で区切られたカンマ区切り文字列を自動的に作成します。例えば、server01 と server02 を選択した場合、'server01', 'server02' の形式になります。引用を無効にするには、変数に csv フォーマットオプションを使用します。

```
${servers:csv}
```

変数形式オプションの詳細については、「高度な変数フォーマットオプション」を参照してください。

注釈

注釈を使用して、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニュー/注釈ビューを使用して注釈クエリを追加します。詳細については、「」を参照してください。

次のコード例は、エポック値を持つ時間列を使用したクエリを示しています。

```
SELECT
  epoch_time as time,
  metric1 as text,
  CONCAT(tag1, ',', tag2) as tags
FROM
  public.test_data
WHERE
  $__unixEpochFilter(epoch_time)
```

次のコード例は、エポック値を持つ時間列と timeend 列を使用するリージョンクエリを示しています。

> **Note**
>
> Grafana v6.6 以降でのみ使用できます。

```
SELECT
  epoch_time as time,
```

```
  epoch_timeend as timeend,
  metric1 as text,
  CONCAT(tag1, ',', tag2) as tags
FROM
  public.test_data
WHERE
  $__unixEpochFilter(epoch_time)
```

次のコード例は、ネイティブ SQL 日付/時刻データ型の時刻列を使用したクエリを示しています。

```
SELECT
  native_date_time as time,
  metric1 as text,
  CONCAT(tag1, ',', tag2) as tags
FROM
  public.test_data
WHERE
  $__timeFilter(native_date_time)
```

名前	説明
time	日付/時刻フィールドの名前。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます。
timeend	終了日時フィールドのオプション名。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます。
text	イベントの説明フィールド。
tags	カンマ区切り文字列としてイベントタグに使用するオプションのフィールド名。

[アラート]

時系列クエリは、アラート条件で機能します。テーブル形式のクエリは、アラートルール条件ではまだサポートされていません。

OpenSearch データソースに接続する

> **Note**
>
> バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインのインストールが必要になる場合があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Amazon Managed Grafana では、オープンソースの OpenSearch (またはレガシー Elasticsearch) をデータソースとして追加できます。OpenSearch に保存されているログまたはメトリクスを視覚化するために、単純または複雑なさまざまな種類の OpenSearch クエリを実行できます。OpenSearch に保存されているログイベントを使用してグラフに注釈を付けることもできます。

OpenSearch をデータソースとして追加する

> **Note**
>
> OpenSearch データソースを追加するには、Grafana IAM アカウントを ALL_ACCESS ロールと SECURITY_MANAGER ロールに追加する必要があります。

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. [ダッシュボード] リンクのサイドメニューには、名前付き [データソース] リンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [OpenSearch] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されない場合は、現在のユーザーに `Admin` ロールが付与されていないことを示します。

名前	説明
Name	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
Defa	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
Url	OpenSearch サーバーの HTTP プロトコル、IP、ポート。
Acce	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。ブラウザ = URL はブラウザからアクセス可能である必要があります。

アクセスモードは、データソースへのリクエストの処理方法を制御します。他に何も記述されていない場合は、サーバーが優先されます。

サーバーアクセスモード (デフォルト)

すべてのリクエストはブラウザから Grafana バックエンドまたはサーバーに行われ、リクエストはデータソースに転送されるため、クロスオリジンリソース共有 (CORS) 要件を回避できます。このアクセスモードを選択する場合、URL は Grafana バックエンドまたはサーバーからアクセス可能である必要があります。

ブラウザ (直接) アクセス

Amazon Managed Grafana は、OpenSearch データソースのブラウザへの直接アクセスをサポートしていません。

インデックス設定

ここでは、`time field` のデフォルトを指定し、OpenSearch インデックスの名前を指定できます。インデックス名またはワイルドカード文字に時間パターンを使用できます。

OpenSearch/Elasticsear バージョン

バージョンドロップダウンメニューで OpenSearch またはレガシー Elasticsearch バージョンを指定します。バージョンごとにクエリの構成方法が異なるため、バージョンは重要です。現在、Grafana は OpenSearch 1.0.x をサポートしています。サポートされている Elasticsearch のバージョンは、`2.0+`、`5.0+`、`5.6+`、`6.0+`、`7.0+` です。値 `5.6+` はバージョン 5.6 以降、6.0 未満を意味します。値 `6.0+` はバージョン 6.0 以降、7.0 未満を意味します。最後に、`7.0+` はバージョン 7.0 以降、8.0 未満を意味します。

最小時間間隔

時間間隔による自動グループの下限。データを 1 分ごとに書き込む場合は 1m など、書き込み頻度に設定することをお勧めします。このオプションは、[Data Source] (データソース) オプションのダッシュボードパネルで上書き/設定することもできます。この値は、1m (1 分) や 30s (30 秒) など、有効な時間識別子が続く数値としてフォーマットする必要があります。以下の時間識別子がサポートされています。

識別子	説明
y	年
M	月
w	週
d	日
h	時間
m	分
s	秒
ms	ミリ秒

ログ

2 つのパラメータである `Message field name` と `Level field name` は、オプションで、Explore のログを視覚化するときにログメッセージとログレベルに使用されるフィールドを決定するデータソース設定ページから設定できます。

例えば、OpenSearch へのログの配送に Filebeat のデフォルト設定を使用する場合、次の設定が機能します。

- メッセージフィールド名: メッセージ
- レベルフィールド名: fields.level

データリンク

データリンクは、Explore のログビューでアクセスできる指定されたフィールドからリンクを作成します。

各データリンク設定は、以下で構成されます。

- フィールド – データリンクで使用されるフィールドの名前。
- URL/クエリ – リンクが外部である場合は、完全なリンク URL を入力します。リンクが内部リンクの場合、この入力はターゲットデータソースのクエリとして機能します。どちらの場合も、`${__value.raw }` マクロを使用してフィールドから値を補間できます。
- 内部リンク – リンクが内部または外部である場合に選択します。リンクが内部の場合、データソースセレクタを使用すると、ターゲットデータソースを選択できます。データソースのトレースのみがサポートされています。

OpenSearch データソースの使用

メトリクスクエリエディタ

OpenSearch クエリエディタを使用すると、複数のメトリクスを選択し、複数の用語またはフィルターでグループ化できます。右側のプラスアイコンとマイナスアイコンを使用して、メトリクスまたはグループを句で追加/削除します。一部のメトリクスと group by 句にはオプションがあります。オプションテキストを選択して行を展開し、オプション別にメトリクスまたはグループを表示および編集します。

シリーズの名前付けとエイリアスパターン

`Alias` 入力フィールドを使用して時系列の名前を制御できます。

パターン	説明
`{{term fieldname}}`	Group By という用語の値に置き換え。
`{{metric}}`	メトリクス名に置き換え (例：平均、最小、最大) 。
`{{field}}`	メトリクスフィールド名に置き換え。

パイプラインメトリクス

例えば、移動平均や派生など、一部のメトリクス集約はパイプライン集約と呼ばれます。OpenSearch パイプラインメトリクスは、別のメトリクスに基づく必要があります。メトリクスの横にある目のアイコンを使用して、メトリクスがグラフに表示されないようにします。これは、パイプラインメトリクスで使用するクエリにのみあるメトリクスに役立ちます。

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

OpenSearch データソースは、[クエリ]変数の[クエリ]フィールドで使用できる 2 種類のクエリをサポートしています。クエリは、カスタム JSON 文字列を使用して記述します。

クエリ	説明
`{"find": "fields", "type": "keyword"}`	インデックスタイプ keyword のフィールド名のリストを返します。
`{"find": "terms", "field": "@hostname", "size": 1000}`	用語集約を使用してフィールドの値のリストを返します。クエリは、クエリの時間範囲として現在のダッシュボードの時間範囲を使用します。
`{"find": "terms", "field": "@hostname", "query": '<lucene query>'}`	用語集約と指定された Lucene クエリフィルターを使用して、フィールドの値のリストを返します。クエリは、クエリの時間範囲として現在のダッシュボードの時間範囲を使用します。

用語クエリにはデフォルトのサイズ制限 500 があります。カスタム制限を設定するには、クエリのサイズプロパティを設定します。クエリでは、その他の変数を使用できます。次のコード例は、$host という名前の変数のクエリ定義を示しています。

```
{"find": "terms", "field": "@hostname", "query": "@source:$source"}
```

前の例では、クエリ定義内で $source という名前の別の変数を使用します。$source 変数の現在の値をドロップダウンリストで変更すると、$host 変数の更新が開始されます。更新後、$host 変数にはホスト名のみが含まれます。この場合、@source ドキュメントプロパティによってフィルタリングされます。

デフォルトでは、これらのクエリは用語の順序で返されます (その後、任意の変数と同様にアルファベット順または数値順にソートできます) 。ドキュメント数でソートされた用語のリスト (上位 N の値リスト) を作成するには、doc_count の orderBy プロパティを追加します。これにより、降順ソートが自動的に選択されます。doc_count (降順 N リスト) で asc を使用するには、order: "asc" を設定しますが、ドキュメントカウントのエラーが増加するため、推奨されません。ドキュメントカウントの順序で用語を保持するには、変数の [ソート] ドロップダウンリストを[無効]に設定します。または、[アルファベット]順を使用して再ソートすることもできます。

```
{"find": "terms", "field": "@hostname", "orderBy": "doc_count"}
```

クエリでの変数の使用

2 つの構文があります。

- $<varname> 例: @hostname:$hostname
- [[varname]] 例: @hostname:[[hostname]]

なぜ 2 つの方法があるのですか？ 最初の構文は読み書きが容易ですが、単語の途中で変数を使用することは許可されません。[複数値]または[すべての値を含める]オプションが有効になっている場合、Grafana はラベルをプレーンテキストから Lucene 互換の条件に変換します。

前の例では、$hostname という名前の変数を使用して @hostname プロパティに基づいてドキュメントをフィルタリングする Lucene クエリがあります。また、フィールド入力ボックス別の用語グループで変数を使用しています。これにより、変数を使用してデータのグループ化方法をすばやく変更できます。

注釈

注釈を使用すると、グラフの上にリッチイベント情報をオーバーレイできます。ダッシュボードメニューまたは注釈ビューを使用して注釈クエリを追加します。Grafana は、注釈イベントの OpenSearch インデックスをクエリできます。詳細については、「注釈」を参照してください。

名前	説明
Que	検索クエリは空白のままにするか、Lucene クエリを指定できます。
Tim	時間フィールドの名前。 日付フィールドである必要があります。
Tim End	終了時刻フィールドのオプション名は日付フィールドである必要があります。設定されている場合、注釈は時間と終了時間の間のリージョンとしてマークされます。
Tex	イベントの説明フィールド。
Tag	イベントタグに使用するオプションのフィールド名 (配列または CSV 文字列) 。

ログのクエリ

OpenSearch からのログデータのクエリと表示は、Explore で確認できます。ログを表示するには、OpenSearch データソースを選択し、オプションで Lucene クエリを入力します。詳細については、「Explore」を参照してください。

ログクエリ

結果が返されると、ログパネルにログ行のリストと棒グラフが表示されます。x 軸は時間、y 軸は頻度またはカウントを示します。

ログメッセージのフィルタリング

必要に応じて、クエリフィールドに Lucene クエリを入力して、ログメッセージをフィルタリングできます。例えば、デフォルトの Filebeat 設定を使用すると、`fields.level:error` を使用してエラーメッセージのみを表示できます。

OpenTSDB データソースに接続する

Amazon Managed Grafana は、OpenTSDB を高度にサポートします。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. [ダッシュボード] リンクのサイドメニューには、[データソース] という名前のリンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [OpenTSDB] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Defau`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Url`	opentsdb サーバーの HTTP プロトコル、ip、ポート (デフォルトポートは通常 4242)。
`Acces`	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。
`Versi`	バージョン = opentsdb バージョン、2.1 以前 または 2.2。
`Resol n`	opentsdb からのメトリクスには、2 秒またはミリ秒の解像度のデータポイントを含めることができます。

クエリエディタ

タイトルを選択して、編集モードでグラフを開きます。データソースのバージョンが 2.1 以前または 2.2 の場合、クエリエディタは異なります。以前のバージョンでは、OpenTSDB のクエリに使用できるのはタグのみです。ただし、後者のバージョンでは、フィルターとタグを使用して opentsdb をクエリできます。フィルポリシーは OpenTSDB 2.2 でも導入されています。

> **Note**
>
> OpenTSDB 2.2 データソースを使用する際は、相互に排他的であるため、フィルターまたはタグのいずれかを使用してください。一緒に使用すると、想定しない結果が得られる可能性があります。

オートコンプリート候補の使用

メトリクス名、タグ名、タグ値の入力を開始すると、オートコンプリート候補が強調表示されます。オートコンプリートは、OpenTSDB で API が有効になっている場合にのみ機能します。

クエリのテンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

クエリ変数

OpenTSDB データソースは、テンプレート変数クエリをサポートしています。つまり、OpenTSDB から値を取得するテンプレート変数を作成できます。例えば、メトリクス名、タグ名、タグ値などを取得できます。

query タイプのテンプレート変数で OpenTSDB を使用する場合、次の構文を使用して検索できます。

クエリ	説明
`metrics(prefix)`	特定のプレフィックスを持つメトリクス名を返します (空になる場合もあります) 。
`tag_names(cpu)`	特定の cpu メトリクスのタグ名 (キー) を返します。
`tag_values(cpu, hostname)`	メトリクス cpu とタグキーホスト名のタグ値を返します。

クエリ	説明
`suggest_tagk(prefix)`	特定のプレフィックスを持つすべてのメトリクスのタグ名 (キー) を返します (空になる場合もあります)。
`suggest_tagv(prefix)`	特定のプレフィックスを持つすべてのメトリクスのタグ値を返します (空になる場合もあります)。

`Preview of values` セクションにテンプレート変数が入力されていない場合は、OpenTSDB サーバー設定で `tsd.core.meta.enable_realtime_ts` を有効にする必要があります。また、OpenTSDB の既存の時系列データのメタデータを入力するには、OpenTSDB サーバーで `tsdb uid metasync` を実行する必要があります。

テンプレートのネスト

1 つのテンプレート変数を使用して、別のテンプレート変数のタグ値をフィルタリングできます。1 番目のパラメータはメトリクス名、2 番目のパラメータはタグ値を見つける必要があるタグキー、その後は他のすべての依存テンプレート変数です。ネストされたテンプレートクエリが正常に機能することを示すため、いくつかの例を以下に挙げます。

クエリ	説明
`tag_values(cpu, hostname, env=$env)`	cpu メトリクスのタグ値、選択した env タグ値、およびタグキーホスト名を返します。
`tag_values(cpu, hostname, env=$env, region=$region)`	cpu メトリクス、選択した env タグ値、選択したリージョンタグ値、およびタグキーホスト名のタグ値を返します。

OpenTSDB メトリクスクエリの詳細については、「OpenTSDB ドキュメント」を参照してください。

PostgreSQL データソースに接続する

PostgreSQL データソースを使用して、Amazon Aurora PostgreSQL データベースからのデータをクエリおよび視覚化できます。

> **Important**
>
> [Grafana バージョン 8] は、Postgres、MySQL 、および Microsoft SQL Server データソースのデータフレームの基盤となるデータ構造を変更します。その結果、時系列のクエリ結果が幅広い形式で返されます。詳細については、Grafana データフレームドキュメントの「ワイドフォーマット」を参照してください。視覚化をバージョン 8 以前と同じように機能させるには、手動で移行する必要がある場合があります。1 つの解決策は、Github の 「Postgres/MySQL/MSSQL: Breaking change in v8.0 related to time series queries and ordering of data column」に記載されています。
> [Grafana バージョン 9] では、PostgreSQL データソースは、以前のバージョンとは異なる方法でデータベースに接続するためのルート証明書を設定します。ワークスペースをバージョン 8 から 9 に更新する場合、接続方法を変更する必要がある場合があります。詳細については「ワークスペースの更新後のトラブルシューティング」を参照してください。

データソースの追加

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. 設定アイコンのサイドメニューに、[データソース] リンクがあります。
3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [PostgreSQL] を選択します。

データソースオプション

名前	説明
`Name`	データソース名。これは、パネルとクエリ内へのデータソースの表示のされ方です。
`Default`	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
`Host`	PostgreSQL インスタンスの IP アドレス/ホスト名とオプションのポート。データベース名を含めないでください。Postgres に接続する

名前	説明
	ための接続文字列が誤っているためにエラーが発生します。
Database	PostgreSQL データベースの名前。
User	データベースユーザーのログイン/ユーザー名。
Password	データベースユーザーのパスワード。
SSL Mode	このオプションは、安全な SSL TCP/IP 接続をサーバーとネゴシエートするかどうか、またはどの優先度でネゴシエートするかを決定します。
Max open	データベースへのオープン接続の最大数。デフォルトは unlimited です (Grafana v5.4 以降)。
Max idle	アイドル接続プールの最大接続数。デフォルトは 2 です (Grafana v5.4 以降)。
Max lifetime	接続を再利用できる秒単位の最大時間、デフォルト 14400/4 時間 (Grafana v5.4 以降)。
Version	このオプションは、クエリビルダーで使用できる関数を決定します (Grafana 5.3 以降でのみ使用可能) 。
TimescaleDB	TimescaleDB は、PostgreSQL 拡張機能として構築された時系列データベースです。有効にすると、Grafana は $__timeGroup マクロで time_bucket を使用し、クエリビルダーに TimescaleDB 固有の集計関数を表示します (Grafana 5.3 以降でのみ使用可能) 。

最小時間間隔

`$_interval` `$_interval_ms` 変数の下限。データを 1 分ごとに書き込む場合は `1m` など、書き込み頻度に設定することをお勧めします。このオプションは、[Data Source] (データソース) オプションのダッシュボードパネルで上書き/設定することもできます。この値は、`1m` (1 分) や `30s` (30 秒) など、有効な時間識別子が続く数値としてフォーマットする必要があります。以下の時間識別子がサポートされています。

識別子	説明
`y`	年
`M`	月
`w`	週
`d`	日
`h`	時間
`m`	分
`s`	秒
`ms`	ミリ秒

データベースユーザーのアクセス許可

> **Important**
> データソースを追加するときに指定するデータベースユーザーには、クエリする指定されたデータベースとテーブルに対してのみ SELECT アクセス許可を付与する必要があります。Grafana は、クエリが安全であることを検証しません。クエリには任意の SQL ステートメントを含めることができます。例えば、 `DELETE FROM user;` や`DROP TABLE user;` などのステートメントが実行されます。これを防ぐために、アクセス許可が制限された特定の PostgreSQL ユーザーを作成することを強くお勧めします。

次のコード例は、アクセス許可が制限された特定の PostgreSQL ユーザーの作成を示しています。

```
CREATE USER grafanareader WITH PASSWORD 'password';
GRANT USAGE ON SCHEMA schema TO grafanareader;
GRANT SELECT ON schema.table TO grafanareader;
```

ユーザーがパブリックロールから不要なアクセス許可を付与されていないことを確認します。

クエリエディタ

PostgreSQL クエリエディタは、Graph または Singlestat パネルの編集モードのメトリクスタブにあります。パネルタイトルを選択して [編集] モードに入り、次に [編集] を選択します。

クエリエディタには、パネル編集モードでクエリが実行された後に表示される[生成 SQL] リンクがあります。これを選択すると、実行された未加工の補間 SQL 文字列が展開されて表示されます。

テーブル、時間列、メトリクス列 (FROM) を選択する

初めて編集モードに入るか新しいクエリを追加したとき、Grafana はタイムスタンプ列と数値列を持つ最初のテーブルをクエリビルダーに事前入力しようとします。

FROM フィールドでは、Grafana はデータベースユーザーの `search_path` にあるテーブルを提案します。`search_path` にないテーブルまたはビューを選択するには、`public.metrics` などの完全修飾名 (schema.table) を手動で入力できます。

Time 列フィールドは、時間値を保持する列の名前を指します。メトリクス列フィールドの値の選択はオプションです。値を選択すると、メトリクス列フィールドがシリーズ名として使用されます。

メトリクス列の提案には、テキストデータ型 (char、varchar、text) の列のみが含まれます。異なるデータ型を持つ列をメトリクス列として使用するには、キャスト `ip::text` を使用して列名を入力できます。`hostname || ' ' || container_name` などのテキストデータ型に評価される任意の SQL 式をメトリクス列フィールドに入力することもできます。

列、ウィンドウ、集計関数 (SELECT)

`SELECT` 行では、使用する列と関数を指定できます。列フィールドでは、`column1 * column2 / column3` などの列名の代わりに任意の式を記述できます。

クエリエディタで使用できる関数は、データソースの設定時に選択した PostgreSQL バージョンによって異なります。集計関数を使用する場合は、結果セットをグループ化する必要があります。集計関数を追加すると、エディタは自動的に `GROUP BY time` を追加します。

エディタは、クエリのこの部分を簡素化および統合しようとします。

プラスボタンを選択し、メニューから列を選択すると、さらに値列を追加できます。グラフパネルには、複数の値列が別々のシリーズとしてプロットされます。

データのフィルタリング (WHERE)

フィルターを追加するには、WHERE 条件の右側にあるプラスアイコンを選択します。フィルターを削除するには、フィルターを選択し、[削除] を選択します。現在選択されている時間範囲のフィルターが、新しいクエリに自動的に追加されます。

グループ化の条件

時間または他の列でグループ化するには、GROUP BY 行の末尾にあるプラスアイコンを選択します。提案ドロップダウンリストには、現在選択されているテーブルのテキスト列のみが表示されますが、手動で任意の列を入力できます。項目を選択し、[削除] を選択して、グループを削除できます。

グループ化を追加する場合、選択したすべての列に集計関数を適用する必要があります。グループ化を追加すると、クエリビルダーは集計関数なしですべての列に集計関数を自動的に追加します。

ギャップ埋め

Amazon Managed Grafana は、時間別にグループ化すると、欠損値を入力できます。時間関数は 2 つの引数を受け入れます。最初の引数はグループ化する時間枠で、2 つ目の引数は Grafana が不足している項目を埋める値です。

テキストエディタモード (RAW)

ハンバーガーアイコン (三) を選択し、[スイッチエディタモード]を選択するか、クエリの下にある [SQL の編集] を選択して、未加エクエリエディタモードに切り替えることができます。

> **Note**
>
> 未加エクエリエディタを使用する場合は、クエリに少なくとも `ORDER BY time` と、返された時間範囲のフィルターがあることを確認してください。

マクロ

クエリ内でマクロは使用すると構文を簡素化し、動的な部分を使用できます。

マクロの例	説明
`$__time(dateColumn)`	UNIX タイムスタンプに変換し、列の名前を `time_sec` に変更する式に置き換えられます。例えば、UNIX_TIMESTAMP(dateColumn) を time_sec として指定します。
`$__timeEpoch(dateColumn)`	UNIX タイムスタンプに変換し、列の名前を `time_sec` に変更する式に置き換えられます。例えば、UNIX_TIMESTAMP(dateColumn) を time_sec として指定します。
`$__timeFilter(dateColumn)`	指定された列名を使用して時間範囲フィルターに置き換えられます。例えば、dateColumn BETWEEN FROM_UNIXTIME(1494

マクロの例	説明
	410783) AND FROM_UNIX TIME(1494 410983) などです。
`$__timeFrom()`	現在アクティブな時間選択の開始に置き換えられます。例えば、FROM_UNIX TIME(1494 410783) などです。
`$__timeTo()`	現在アクティブな時間選択の終了に置き換えられます。例えば、FROM_UNIX TIME(1494 410983) などです。
`$__timeGroup(dateColumn,'5m')`	GROUP BY 句で使用できる式に置き換えられます。例えば、cast(cast (UNIX_TIM ESTAMP(da teColumn)/(300) as signed)300 as signed),* などです

マクロの例	説明
`$__timeGroup(dateColumn,'5m', 0)`	前の行と同じですが、fill パラメーターがあるため、その系列の欠落しているポイントは grafana によって追加され、0 が値として使用されます。
`$__timeGroup(dateColumn,'5m', NULL)`	上記と同じですが、NULL は欠落ポイントの値として使用されます。
`$__timeGroup(dateColumn,'5m', previous)`	上記と同じですが、その系列の前の値がフィル値として使用されます。まだ値が見つからない場合は NULL が使用されます (Grafana 5.3 以降でのみ使用可能) 。
`$__timeGroupAlias(dateColumn,'5m')`	$__timeGroup と同じですが、列エイリアスが追加されます

マクロの例	説明
$__unixEpochFilter(dateColumn)	指定された列名を使用して、Unix タイムスタンプとして表される時間を使用して、時間範囲フィルターに置き換えられます。例えば、*dateColumn > 1494410783 AND dateColumn < 1494497183* などです
$__unixEpochFrom() `	現在アクティブな時間選択の開始が Unix タイムスタンプに置き換えられます。例えば、*1494410783* などです
$__unixEpochTo()	現在アクティブな時間選択の最後が Unix タイムスタンプに置き換えられます。例えば、*1494497183* などです

マクロの例	説明
`$__unixEpochNanoFilter(dateColumn)`	指定された列名とナノ秒タイムスタンプとして表される時刻を使用して、時間範囲フィルターに置き換えられます。例えば、*dateColumn > 1494410783152415214 AND dateColumn < 1494497183142514872* などです
`$__unixEpochNanoFrom()`	現在アクティブな時間選択の開始がナノ秒タイムスタンプに置き換えられます。例えば、*1494410783152415214* などです

マクロの例	説明
`$__unixEpochNanoTo()`	現在アクティブな時間選択の終了がナノ秒タイムスタンプに置き換えられます。例えば、*1494497183142514872* などです
`$__unixEpochGroup(dateColumn,"5m", [fillmode])`	$__ timeGroup と同じですが、時刻は Unix タイムスタンプとして格納されます。

テーブルクエリ

クエリオプションが[]テーブル としてフォーマットに設定されている場合、基本的に任意のタイプの SQL クエリを実行できます。テーブルパネルには、クエリが返す列と行の結果が自動的に表示されます。

テーブルパネルの列名は、通常の `as` SQL 列選択構文を使用して制御できます。

時系列クエリ

[Format as] を `Time series` に設定する場合、例えばグラフパネルで使用するには、クエリは SQL 日時または Unix 工ポックを表す任意の数値データ型を返す `time` という名前の列を返す必要があります。`time` および `metric` を除くすべての列は、値列として扱われます。値列のメトリクス名として使用される `metric` という名前の列を返すことができます。複数の値列と `metric` という名前の列を返す場合、この列はシリーズ名のプレフィックスとして使用されます。

時系列クエリの結果セットは、時間別にソートする必要があります。

次のコード例は、`metric` 列を示しています。

```
SELECT
  $__timeGroup("time_date_time",'5m'),
  min("value_double"),
  'min' as metric
FROM test_data
WHERE $__timeFilter("time_date_time")
GROUP BY time
ORDER BY time
```

次のコード例は、$__timeGroup マクロの fill パラメータを使用して NULL 値をゼロに変換する方法を示しています。

```
SELECT
  $__timeGroup("createdAt",'5m',0),
  sum(value) as value,
  measurement
FROM test_data
WHERE
  $__timeFilter("createdAt")
GROUP BY time, measurement
ORDER BY time
```

次のコード例は、複数の 列を示しています。

```
SELECT
  $__timeGroup("time_date_time",'5m'),
  min("value_double") as "min_value",
  max("value_double") as "max_value"
FROM test_data
WHERE $__timeFilter("time_date_time")
GROUP BY time
ORDER BY time
```

テンプレート作成

メトリクスクエリでサーバー、アプリケーション、センサー名などのものをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして

表示されます。これらのドロップダウンボックスを使用して、ダッシュボードに表示されるデータを変更できます。

テンプレート作成とテンプレート変数の詳細については、「テンプレート」を参照してください。

クエリ変数

タイプ Query のテンプレート変数を追加すると、測定名、キー名、ドロップダウン選択ボックスとして表示されるキー値などを返すことができる PostgreSQL クエリを記述できます。

例えば、テンプレート変数クエリ設定でこのようなクエリを指定した場合、テーブル内の hostname 列のすべての値を含む変数を指定できます。

```
SELECT hostname FROM host
```

クエリは複数の列を返すことができ、Grafana はそれらの列からリストを自動的に作成します。例えば、次のクエリは、 hostname と hostname2 の値を含むリストを返します。

```
SELECT host.hostname, other_host.hostname2 FROM host JOIN other_host ON host.city =
 other_host.city
```

クエリで $__timeFilter(column) などの時間範囲依存マクロを使用するには、テンプレート変数の更新モードを [時間範囲変更] に設定する必要があります。

```
SELECT event_name FROM event_log WHERE $__timeFilter(time_column)
```

もう 1 つのオプションは、キー/値変数を作成できるクエリです。クエリは、 __text と __value という名前の 2 つの列を返します。__text 列値は一意である必要があります (一意でない場合は、最初の値が使用されます) 。ドロップダウンリストのオプションには、わかりやすい名前をテキストとして、ID を値として指定できるテキストと値があります。hostname をテキスト、id を値とするクエリの例:

```
SELECT hostname AS __text, id AS __value FROM host
```

ネストされた変数も作成できます。region という名前の変数を使用すると、ホスト変数に現在選択されているリージョンのホストのみを表示させることができます。次のコード例は、このような

クエリを示しています (region が複数値変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します)。

```
SELECT hostname FROM host  WHERE region IN($region)
```

__searchFilter を使用してクエリ変数で結果をフィルタリングする

クエリフィールドで __searchFilter を使用すると、ドロップダウン選択ボックスのユーザータイプに基づいてクエリ結果がフィルタリングされます。ユーザーによって何も入力されていない場合、__searchFilter のデフォルト値は % です。

> **Note**
>
> Grafana はこれを行わないため、__searchFilter 式を引用符で囲むことが重要です。

次の例は、__searchFilter をクエリフィールドの一部として使用してユーザーがドロップダウン選択ボックスに入力するときに hostname を検索できるようにします。

```
SELECT hostname FROM my_host  WHERE hostname LIKE '$__searchFilter'
```

クエリでの変数の使用

テンプレート変数の値は、テンプレート変数が multi-value の場合にのみ引用符で囲まれます。

変数が複数値の変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します。

2 つの構文があります。

$<varname> hostname という名前のテンプレート変数の例:

```
SELECT
  atimestamp as time,
  aint as value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in($hostname)
```

```
ORDER BY atimestamp ASC
```

[[varname]] hostname という名前のテンプレート変数の例:

```
SELECT
  atimestamp as time,
  aint as value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in([[hostname]])
ORDER BY atimestamp ASC
```

複数値変数の引用をオフにする

Amazon Managed Grafana は、複数値変数に対して引用符で囲まれたカンマ区切り文字列を自動的に作成します。例えば、server01 と server02 を選択した場合、'server01', 'server02' の形式になります。引用を無効にするには、変数に csv フォーマットオプションを使用します。

${servers:csv}

変数形式オプションの詳細については、「テンプレートおよび変数」を参照してください。

注釈

注釈を使用して、グラフの上にリッチイベント情報をオーバーレイします。ダッシュボードメニュー/注釈ビューを使用して注釈クエリを追加します。詳細については、「注釈」を参照してください。

次のコード例は、エポック値を持つ時間列を使用したクエリを示しています。

```
SELECT
  epoch_time as time,
  metric1 as text,
  concat_ws(', ', metric1::text, metric2::text) as tags
FROM
  public.test_data
WHERE
  $__unixEpochFilter(epoch_time)
```

次のコード例は、エポック値を持つ時間列と timeend 列を使用するリージョンクエリを示しています。

> **Note**
>
> これは Grafana v6.6 以降でのみ使用できます。

```
SELECT
  epoch_time as time,
  epoch_time_end as timeend,
  metric1 as text,
  concat_ws(', ', metric1::text, metric2::text) as tags
FROM
  public.test_data
WHERE
  $__unixEpochFilter(epoch_time)
```

次のコード例は、ネイティブ SQL 日付/時刻データ型の時刻列を使用したクエリを示しています。

```
SELECT
  native_date_time as time,
  metric1 as text,
  concat_ws(', ', metric1::text, metric2::text) as tags
FROM
  public.test_data
WHERE
  $__timeFilter(native_date_time)
```

名前	説明
time	日付/時刻フィールドの名前。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます。
timeend	終了日時フィールドのオプション名。ネイティブ SQL 日付/時刻データ型または工ポック値を持つ列にすることができます (Grafana v6.6 以降)。
text	イベントの説明フィールド。

名前	説明
tags	カンマ区切り文字列としてイベントタグに使用するオプションのフィールド名。

[アラート]

時系列クエリは、アラート条件で機能します。テーブル形式のクエリは、アラートルール条件ではまだサポートされていません。

Tempo データソースへの接続

Tempo は、Grafana Labs の大容量で、依存関係が最小のトレースストレージ、OSS トレースソリューションです。

データソースの追加

Tempo 設定にアクセスするには、[設定] (歯車) アイコンを選択してから[データソース] を選択し、[Tempo] を選択します。

名前	説明
Name	パネル、クエリ、Explore でデータソースを参照するために使用する名前。
Default	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
URL	Tempo インスタンスの URL。例: http://tempo 。
Basic Auth	Tempo データソースへの基本認証を有効にします。
User	基本認証のユーザー名。
Passwor	基本認証のパスワード。

ログへのトレース

これは ログへのトレース機能の設定です。現在、ターゲットデータソースは Loki である必要があります。詳細については、「トレース統合」を参照してください。

- データソース – ターゲットデータソース。
- タグ – Loki クエリで使用するタグ。デフォルトは `'cluster', 'hostname', 'namespace', pod'` です。
- スパン開始時刻シフト – スパン開始時刻に基づいた Loki クエリの開始時刻のシフト。過去に拡張するには、負の値を使用する必要があります。ここで使用できる時間単位は、5s、1m、3h などです。デフォルトは 0 です。
- スパン終了時刻シフト – スパン終了時刻に基づいた Loki クエリの終了時刻のシフト。ここで使用できる時間単位は、5s、1m、3h などです。デフォルトは 0 です。

クエリのトレース

Explore を使用して Tempo からトレースをクエリおよび表示できます。データソース設定ページでログの追跡設定をすると、トレースを検索できます。視覚化するトレースを見つける際は、Loki クエリエディタを使用します。検索結果を取得する場合は、このデータソースを指す派生フィールドを設定する必要があります。

特定のトレースをクエリするには、TraceID クエリタイプを選択してから、その ID を Trace ID フィールドに入力します。

ログからトレース ID へのリンク

内部リンクを設定すると、Loki または Elastic のログから Tempo トレースにリンクできます。詳細については、「派生フィールド」を参照してください。

テスト用 TestData データソースの設定

Grafana には、任意のパネルにシミュレーションされた時系列データを作成するための TestData データソースが付属しています。このデータソースを使用すると、独自の即席のランダムな時系列データを構築して、任意のパネルにレンダリングできます。これにより、ダッシュボードの機能を検証して、安全かつ簡単にデータを共有できます。

データソースの設定

TestData のデータソース設定にアクセスする方法

1. [設定] (歯車) アイコンを選択します。
2. [データソース] を選択します。
3. [TestData] を選択します。

データソースには、すべてのデータソースに共通する最も基本的なオプション以外の設定はありません。

名前	説明
Name	パネル、クエリ、Explore のデータソース名。
Default	このデータソースを新しいパネル用に事前選択するかどうか。

モックデータの作成

TestData データソースが追加されました。Grafana インスタンスのユーザーはこれをデータソースとして任意のメトリクスパネルで使用できます。このデータソースには、選択した TestData シナリオに基づいて使用できるモックデータが用意されています。

シナリオの選択

クエリエディタの代わりに TestData データソースを使用すると、シミュレーションされたデータをパネルに生成するシナリオを選択する際に役立ちます。

各シナリオにはエイリアスを割り当てることができ、多くのシナリオには選択時に表示される独自のオプションがあります。

使用可能なシナリオ:

- 注釈
- 条件付きエラー
- CSV コンテンツ
- CSV ファイル
- CSV メトリクス値
- 範囲外のデータポイント
- 指数ヒートマップバケットデータ
- Grafana API
- [Grafana Live]
- 線形ヒートマップバケットデータ

- Apache Arrow データのロード
- ログ
- データポイントなし
- ノードグラフ
- 予測可能 CSV ウェーブ
- 予測可能パルス
- ランダムウォーク
- ランダムウォーク (エラーあり)
- ランダムウォークテーブル
- 未加工フレーム
- シミュレーション
- スロークエリ
- ストリーミングクライアント
- テーブル静的
- USA 生成データ

事前設定のダッシュボードのインポート

TestData にはダッシュボード例も用意されています。

サンプルダッシュボードのインポート方法

1. データソースの設定ページに移動します。
2. [ダッシュボード] タブを選択します。
3. [シンプルなストリーミング例] ダッシュボードの [インポート] を選択します。

インポートされたダッシュボードのカスタマイズ方法:

インポートしたダッシュボードをカスタマイズする際は、別の名前で保存することをお勧めします。そうしないと、Grafana をアップグレードしたときにカスタマイズされたダッシュボードが新しいバージョンで上書きされる可能性があります。

テストデータを使用して問題をレポート

GitHub の GrafanaLabs に時系列データの使用またはレンダリングに関する問題を報告する場合は、このデータソースを使用して問題を再現することを強くお勧めします。そうすると、デベロッパーが問題を再現して解決するのが容易になります。

Zipkin データソースへの接続

Zipkin はオープンソースで、分散トレースシステムです。Amazon Managed Grafana の Explore でトレースに対してクエリを実行できるようにする Zipkin データソースを追加します。

データソースの追加

Zipkin 設定にアクセスするには、設定 (歯車) アイコンを選択してからデータソースを選択し、[Zipkin] を選択します。

名前	説明
Name	データソース名。これはパネル、クエリ、Explore でデータソースを表示する方法です。
Default	デフォルトのデータソースは、新しいパネル用に事前に選択されたデータソースです。
URL	Zipkin インスタンスの URL。例: `http://localhost:9411` 。
Access	サーバー (デフォルト) = URL は Grafana バックエンド/サーバーからアクセス可能である必要があります。
Basic Auth	Zipkin データソースへの基本認証を有効にします。
User	基本認証のユーザー名。
Passwo	基本認証のパスワード。

クエリのトレース

Zipkin からのトレースのクエリ実行と表示は、Explore から可能です。

Zipkin クエリエディタを使用すると、トレース ID で直接クエリを実行、またはトレースセレクタからトレースを選択できます。トレース ID でクエリするには、ID をテキスト入力に挿入します。

トレースセレクタを使用して、Explore で選択した時間範囲に口グ記録されたすべてのトレースから特定のトレースを選択します。トレースセレクターには 3 つのレベルのネストがあります: 1. これはユーザーが関心を持つサービスです。1. 特定のオペレーションは、選択したサービス 1 の一部です。選択したオペレーションが発生した特定のトレースであり、ルートオペレーション名とトレース期間で表されます。

トレース UI のデータマッピング

Zipkin の注釈は、注釈キーに注釈値を示すログとしてトレースビューに表示されます。

ログからトレース ID へのリンク

Loki のログから Zipkin トレースにリンクするためには、内部リンクを使用して派生フィールドを設定します。詳細については、「派生フィールド」を参照してください。

Enterprise データソースに接続する

以下のデータソースは、Amazon Managed Grafana Enterprise プラグインにアップグレードされたワークスペースでサポートされています。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。

Enterprise プラグインは定期的に更新されますが、既存のプラグインが更新されたり、場合によっては新しいデータソースが追加されることもあります。このドキュメントで紹介するデータソースがすべてではない可能性があります。現在 Amazon Managed Grafana Enterprise プラグインでサポートされている Enterprise プラグインのリストについては、Grafana ドキュメントの「Grafana Enterprise プラグイン」を参照してください。

バージョン 9 以降をサポートするワークスペースでは、エンタープライズデータソースはデフォルトではインストールされていません。適切なデータソースプラグインをインストールする必要があります。ここに記載されていないものを含め、すべての Enterprise データソース用のプラグインをインストールできます。また、インストール済みのプラグインのバージョンを上げることもできます。プラグインの管理の詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

トピック

- AppDynamics データソースに接続する
- Databricks データソースに接続する
- Datadog データソースに接続する

- Dynatrace データソースに接続する
- GitLab データソースへの接続
- Honeycomb データソースへの接続
- Jira データソースへの接続
- MongoDB データソースに接続する
- New Relic データソースへの接続
- Oracle Database のデータソースへの接続
- Salesforce データソースへの接続
- SAP HANA データソースへの接続
- ServiceNow データソースへの接続
- Snowflake データベースへの接続
- Splunk データソースへの接続
- Splunk Infrastructure Monitoring データソースへの接続
- Wavefront データソースに接続する (VMware Tanzu Observability by Wavefront)

AppDynamics データソースに接続する

Amazon Managed Grafana の AppDynamics データソースを使用すると、Metrics API を使用して AppDynamics からメトリクスをクエリし、Grafana ダッシュボードで視覚化できます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

データソースの設定に関する注意事項

サーバー (プロキシ) アクセス (CORS やユーザーがパスワードを検索できないようにするため) と Basic 認証を使用します。ユーザー名は「user@account」の形式で設定します (例: my.name@customer1 または my_user@saas_account_name)。

次の手順に従い、パスワードを設定します。

1. https://accounts.appdynamics.com/subscriptions に移動します。
2. サブスクリプションの行の「名前」列のリンクを選択します。
3. ページ上部のタブを選択して、「ライセンスの詳細」に移動します。
4. アクセスキーフィールドに [表示] ボタンがあります。[表示] ボタンを選択してアクセスキーを表示します。
5. Grafana の設定ページにある「Basic 認証の詳細」の「パスワード」フィールドにアクセスキーをコピーします。

次の手順を実行して、Amazon Managed Grafana のユーザーとロールを設定します。

1. AppDynamics で、[設定]、[管理] の順に移動します。
2. [ロール] タブを選択して [+] ボタンを選択し、新しいロール (例: `grafana_readonly`.) を作成します。
3. ロールの作成 セクションの [アカウント] タブで、 アクセス権限 `View Business Flow` を追加します。
4. [アプリケーション] タブで、[表示] チェックボックスをオンにして、Grafana がアプリケーションデータを表示できるようにします。
5. [データベース] タブで、[表示] チェックボックスをオンにして、Grafana がデータベースデータを表示できるようにします。
6. [分析] タブで、[すべてのアプリケーションからデータを表示できる] チェックボックスをオンにして、Grafana がアプリケーション分析データを表示できるようにします。
7. 管理ページのユーザータブで、新しいユーザー (例: `grafana`) を作成します。新しいユーザー (またはユーザーが属するグループ) を、先ほど作成したロール (例: `grafana_readonly`) に割り当てます。

テンプレート作成

現在サポートされているテンプレートクエリは次のとおりです。

1. `Applications` (すべてのアプリケーション)
2. `AppName.BusinessTransactions` (アプリケーション名のすべての BT)
3. `AppName.Tiers` (アプリケーション名のすべての階層)
4. `AppName.Nodes` (アプリケーション名のすべてのノード)

5. AppName.TierName.BusinessTransactions (特定の階層のすべての BT)
6. AppName.TierName.Nodes (特定の階層のすべてのノード)
7. AppName.Path.<Any Metric Path> (任意のメトリクスパスの指定が可能)

凡例キー

凡例キーのデフォルトはかなり長くなることがありますが、このフォーマットはカスタマイズが可能です。

App on legend オプションを選択すると、アプリケーション名を凡例キーの前に追加できます。例: MyApp - Overall Application Performance|Average Response Time (ms)。

クエリが単一統計などのパネル用で凡例キーが見えない場合は、「Show Metadata」(メタデータを表示) オプションを選択することで、クエリの凡例キー (エイリアスとも呼ばれます) を確認することができます。

凡例ドロップダウンリストには、Full Path、Segments、Custom の 3 つのオプションがあります。

凡例オプション – フルパス

凡例キーは、完全なメトリクスパス (例: Overall Application Performance|Average Response Time (ms)) です。

凡例オプション – セグメント

メトリクス名は複数のセグメントで構成されます。どのセグメントを表示するかの選択が可能です。

例えば、メトリクス名が次の場合:

Errors|mywebsite|Error|Errors per Minute

セグメントフィールドに 2,4 と入力すると mywebsite|Errors per minute が返されます。

インデックスは 1 で始まるため、1 と入力すると Errors が返されます。

凡例オプション – カスタム

テキストを次のエイリアスパターンと組み合わせてカスタム凡例を作成し、メトリクスメタデータを混在させることができます。

- {{app}} はアプリケーション名を返します
- {{1}} は、メトリクスパスからセグメントを返します。

  例えば、メトリクス Overall Application Performance|Average Response Time (ms) には 2 つのセグメントがあります。 {{1}} は最初のセグメントを返し、{{2}} は 2 番目のセグメントを返します。

以下は凡例キーのパターン例と生成される凡例キーです：

- custom legend key => custom legend key
- App: {{app}} MetricPart2: {{2}} => App: myApp MetricPart2: Average Response Time (ms)

Databricks データソースに接続する

Databricks データソースを使用すると、Amazon Managed Grafana 内の Databricks データをクエリおよび視覚化できます。クエリのフォーマットやカラーコードを設定できる SQL エディタも付属しています。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Databricks データソースの追加

以下に Grafana コンソールに Databricks データソースを追加する手順を紹介します。

データソースの追加方法

1. 上部ヘッダーの Grafana アイコンを選択して、サイドメニューを開きます。
2. サイドメニューの [ダッシュボード] リンクから [データソース] を選択します。

> **Note**
>
> [データソース] リンクが表示されていない場合は、Grafana の Admin ロールが付与されていません。

3. 上部ヘッダーの [+ データソースの追加] ボタンを選択します。
4. [タイプ] ドロップダウンリストから [Databricks] を選択します。

> **Note**
>
> Databricks が表示されない場合、Grafana Enterprise にアップグレードする必要があります。

5. データに接続して編集するオプションを選択します。

Databricks データソースを使用する際の注意事項

時系列

時系列の視覚的表現は、クエリに `datetime` フィールドを追加するときに選択できます。このフィールドは、シリーズのタイムスタンプとして使用されます。フィールドで特にタイムゾーンが指定されていない場合、Grafana はその時間を UTC として扱います。

複数行の時系列

複数行の時系列の視覚的表現を作成するには、クエリに以下の順序で少なくとも 3 つのフィールドを含める必要があります。

1. エイリアスが `time` の `datetime` フィールド。
2. `GROUP BY` の値。
3. 視覚化する 1 つ以上のメトリクス値。

以下は、複数行の時系列オプションを返すクエリの例です。

```
SELECT log_time AS time, machine_group, avg(disk_free) AS avg_disk_free
FROM mgbench.logs1
GROUP BY machine_group, log_time
```

```
ORDER BY log_time
```

Datadog データソースに接続する

Datadog データソースを使用すると、Amazon Managed Grafana の Datadog モニタリングサービスのメトリクスを視覚化できます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

使用方法

キャッシュ

Datadog API を使用することで、大量のクエリを行う大規模ダッシュボードのレート (Datadog API が許可する 1 時間あたりの API 呼び出しの最大数) を制限することができます。一意のクエリはキャッシュ機能により 60 秒間キャッシュされます。この間隔は、設定ページで長くしたり短くしたりすることができます。

クエリエディタ

操作は簡単。集約とメトリクスを選択します。選択したタグ (複数可) で結果をフィルタリングすることができます。

Datadog データソースは、Datadog クエリエディタがサポートするすべての高度な関数をサポートします。ドロップダウンリストから関数名を選択して配置します。

フィールド別のエイリアスの使用方法:

- 「エイリアス方法」フィールドにエイリアスを入力します。
- スコープ付き変数を使用:
  - `$__metric` = メトリクス名に置き換えます
  - `$__display_name` = メトリクス名に置き換えます

  - $__expression = 完全なメトリクス式に置き換えます
  - $__aggr = メトリクス集計関数に置き換えます (例: avg、max、min、sum)
  - $__scope = メトリクススコープ (リージョン、サイト、環境、ホストなど) に置き換えます

- 正規表現を使用:
  - 正規表現を「Alias RegExp」フィールドに /you regexp here/flags 形式で入力します。
  - 「エイリアス方法」フィールドが空の場合、RegExp の結果は次のように結合されます。例: メトリクス式 = avg:system.load.5{*} :「エイリアス方法」フィールド入力 :「」「Alias RegExp」フィールド入力: avg:(.+)\.(\d) 結果: system.load, 5
  - 「エイリアス方法」フィールドで $<group_number> 変数を使用します。例: メトリクス式 = avg:system.load.5{*} :「エイリアス方法」フィールド入力 : $1: $2 seconds「Alias RegExp」フィールド入力: avg:(.+)\.(\d) 結果: system.load: 5 seconds
  - $0 を使用して式全体を取得します。例: メトリクス式 = datadog.dogstatsd.packet.count{*} :「エイリアス方法」フィールド入力 : Expression: $0「Alias RegExp」フィールド入力: DOGstatsd\.(.*)\.(.*){*}/i 結果: Expression: datadog.dogstatsd.packet.count{*}

  注: 存在しないグループ番号を使用すると、エラーが発生します。

メトリクス演算

メトリクス演算を使用するには、[クエリタイプ] を [演算] に設定します。# 記号を使用して、必要なメトリクスにリンクします。例えば、#A * 2 はクエリ A の結果を 2 倍にします。同様の方法で 2 つのメトリクス間の演算も可能です。#A / #B のように、計算に使用する結果を含むクエリを追加し、3番目のクエリでこれらのメトリックにリンクします。

注釈

注釈はグラフの上に重ねて表示されるイベント (デプロイメントや障害など) です。このデータソースを使用すると、Datadog からイベントを取得し、そのイベントを Amazon Managed Grafana のグラフの上に重ねて表示することができます。注釈のイベントは、ソース、タグ、または優先度でフィルタリングできます。

テンプレート作成

テンプレート変数の値を取得するためのオプションには、メトリクスとタグがあります。利用可能なメトリクスのリストを取得するには、[Query] (クエリ) フィールドに * を指定します。

すべてのタグを返すには、値に `tag` または `scope` を使用します。

指定したタググループのタグを返すには、次のデフォルトカテゴリ値のいずれかを使用します。

- `host`
- `device`
- `env`
- `region`
- `site`
- `status`
- `version`

カスタムタググループの場合は、タググループ名を入力するだけです。例えば、カスタムタググループ名が `subscription_name` の場合、その名前を [Query] (クエリ) フィールドに入力します。

[Regex] (正規表現) フィールドを使用して結果をフィルタリングします。タグを使用する場合、複数値の変数がサポートされます。選択したタグの複数の値は、カンマ区切りのタグリストに変換されます。

アドホックフィルター

Grafana には、[アドホックフィルター]と呼ばれる新しい特殊なタイプのテンプレート変数があります。この変数は、ダッシュボード内のすべての Datadog クエリに適用されます。これにより、クイックフィルターのように使用できます。Datadog のアドホック変数は、タグからすべてのキーと値のペア (例: `region:east, region:west`) を取得し、クエリタグとして使用します。この変数を作成するには、[アドホックフィルター] タイプを選択し、Datadog データソースを選択します。この変数には任意の名前を設定できます。

Dynatrace データソースに接続する

https://www.dynatrace.com/ 向けのデータソース。このデータソースの使用には、Dynatrace アカウントが必要になります。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。

さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

既知の制限事項

テンプレート変数は複数選択できません。単一選択のみがサポートされています。

v2 メトリクス API のみがサポートされています。

機能

主要機能

- テンプレート変数
  - メトリクス名
  - 単一選択のみ (複数選択不可)
  - アドホックフィルター
- 注釈
  - 現在サポートされていません
- エイリアス定義
  - メトリクス名
  - 集計
  - [Display Name] (表示名)
  - ホスト
  - 説明
- [アラート]
  - アラートの完全サポート

Dynatrace 固有の機能

Dynatrace メトリクス v2 API により、ビルトインのメトリクスとカスタムメトリクスの両方をサポートします。詳細については、「Dynatrace ドキュメント: Metrics API v2」と「Metric Ingestion」を参照してください。

メトリクスによっては、API が追加の変換オプションをサポートする場合があります。

Dynatrace の権限

Dynatrace では、「API V2 (metrics.read) 権限を使用したメトリクスの読み取り権限」、「API V2 (entities.read) 権限を使用したエンティティの読み取り権限」が必要になります。

Dynatrace から API キーを取得する

API トークンを設定するには、「Dynatrace API - トークンと認証」を参照してください。

API トークンの `metrics.read` と `entities.read` の権限を設定します。

設定

1. 論理 Grafana サーバー UI で [設定/データソース]、[データソースの追加] の順に選択します。
2. [データソースの追加] ページで、Dynatrace でフィルタリングして Dynatrace プラグインを選択します。
3. Dynatrace データソースの設定では、以下のパラメータを設定する必要があります。

   - `Name` - Dynatrace データソースに適用する名前 (デフォルト: Dynatrace)。
   - `Dynatrace API Type` - 接続先の Dynatrace インスタンスの種類。これは `SaaS` または `Managed Cluster` のいずれかになります。
   - `Dynatrace API Token` - 前のステップで生成した API トークンです。

   以下の 2 つの設定は、Dynatrace SaaS か マネージドかによって異なります

   - `yfc55578.live.dynatrace.com` の SaaS の例では、環境 ID は `yfc55578` になります。
   - `yd8888.managed-sprint.dynalabs.io/e/abc99984-3af2-55tt-72kl-0672983gc45` のマネージドの例では、環境 ID は `abc99984-3af2-55tt-72kl-0672983gc45` で、ドメインは `yd8888.managed-sprint.dynalabs.io` になります
4. すべての設定値の設定が完了したら、[保存とテスト] を選択して設定を検証し、変更を保存します。

データソースのクエリ

Dynatrace のメトリクスと問題のクエリには、クエリエディタを使用します。クエリの種類は `metric` または `problem` です。

メトリクスクエリタイプ

- `Metric` — 表示するメトリクスを選択します。Dynatrace からメトリクスリストを再度取得するには、[更新] ボタンを選択します。
- `Aggregations` — 特定のメトリクスに使用する集計を選択します。集計値を選択して集計タイプを変更するか、 + を選択して別の集計を追加します。
- `Transformations`— クエリエディタで変換を選択することができます。選択した変換にいくつかのパラメータを入力します。現在、マージ変換のみがサポートされています。マージ変換の詳細については、 ユーザーガイドの「マージ変換」を参照してください。
- `Filters` — Dynatrace データソースは、各メトリクスに適したフィルターを動的にクエリします。フィルターを追加するには、Dynatrace クエリエディタの [フィルター] ラベルの横にある [+] を選択し、フィルタリングするフィールド、使用する演算子、フィルタリングする値の順に選択します。Dynatrace データソースでは、フィルターグループを作成し、それらを組み合わせて複雑な論理比較を作成することができます。ほとんどのユースケースでは、フィルターグループは必要ありません。Dynatrace では、タグでフィルターを作成する際に選択した組み合わせに関係なく、常に AND 演算子が使用されます。Dynatrace では、タグを使用したフィルタリングに OR 演算子をサポートしていません。
- `Alias` — Dynatrace データソースを使用していると、2 種類のエイリアスを目にすることがあります。1 つ目は静的エイリアスです。エイリアス名は小文字で始まり、構築するすべてのクエリで使用できます。2 つ目は動的エイリアスで、エイリアス名は大文字で始まり、クエリで使用しているメトリクスに基づいて変更されます。Dynatrace プラグインは、`Metric Names`、`Aggregation`、`Display Name`、`Host`、`Description` の数種類のエイリアスをサポートしています。

名前	値
`$name`	builtin:apps.other.keyUserActions.reportedErrorCount.os
`$aggregation`	自動,値
`$displayName`	報告されたエラー数 (キーユーザーアクション、OS 別) [モバイル、カスタム]

問題クエリタイプ

- Problem Query Type — 問題クエリタイプを選択します。現在、フィード問題クエリタイプのみがサポートされています。フィード問題クエリタイプの詳細については、「マージ変換」を参照してください。
- Status Filter — ステータスで結果問題をフィルタリングします。
- Impact Filter — 影響レベルで結果問題をフィルタリングします。
- Severity Filter - 結果の問題を重要度レベルでフィルタリングします。
- Expand Details — 設定されている場合、レスポンスに関連するイベントを含めます。

テンプレート変数の使用

新しい Dynatrace クエリ変数を追加するには、「新しいテンプレート変数の追加」を参照してください。Dynatrace データソースを、以下の使用可能なクエリのデータソースとして使用します。

- Query type — クエリタイプを選択します。クエリタイプは、一部のデータを一部のキーまたは記述子に関連付けます。

クエリタイプ	説明
Metric names	すべてのメトリクス名のリストを返します
Filter keys	フィルタリングに使用できるすべてのディメンション (ホスト名など) のリストを返します。
Filter values for key	キー名またはキー名テンプレート変数でフィルタリングされたすべての値のリストを返します
Problem status options	すべての問題ステータスのリストを返します
Problem impact options	問題が影響を受けるすべてのエリアのリストを返します。
Problem severity options	すべての問題の重要度タイプのリストを返します

- Regex — (オプション) クエリから返された値を正規表現でフィルタリングします。

> **Note**
>
> `Multi-value` および `Include All option` は現在、Dynatrace データソースではサポートされていません。

作成した変数は、[メトリクス] ドロップダウンメニューに表示されます。

Dynatrace のダッシュボードをインポートする

ダッシュボードをインポートするには、「ダッシュボードのインポート」を参照してください。インポートされたダッシュボードは、[設定] > [データソース] > Dynatraceデータソースを選択 > [ダッシュボード]] タブにあり、事前作成済みのダッシュボードとして使用することができます。

GitLab データソースへの接続

GitLab データソースを使用すると、上位の寄与者、1 日あたりのコミット数、1 日あたりのデプロイ数など、GitLab の詳細な統計を追跡できます。プロジェクトなどのテンプレート変数を使用すると、ダッシュボードのフィルターを設定することもできます。GitLab API のデータは他のソースのデータと組み合わせることができます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

既知の制限事項

GitLab API の未加工データから意味のある集計メトリクスを取得する唯一の方法は変換ですが、アラートクエリでは変換がサポートされていないため、このプラグインではアラートがまだサポートされていません。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。

2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。
3. [データソースの追加] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

4. データソースのリストから [GitLab] を選択します。
5. 次の情報を入力します。
   - [名前] に、この GitLab データソースの名前を入力します。
   - [URL] に、**`https://gitlab.com/api/v4`** などの GitLab インスタンスのルート URL を入力します。
   - [アクセストークン] に、GitLab パーソナルアクセストークンを入力します。

GitLab データソースのクエリ

GitLab クエリエディタから、コミット、問題点、リリースなどの、さまざまなリソースタイプを選択できます。

プロジェクトのフィルタリングおよび表示

1. ドロップダウンリストから、[プロジェクト] を選択します。
2. (オプション) 所有するプロジェクトでフィルタリングします。
3. ドロップダウンを使用して、[あり] または [いいえ] を選択して結果をフィルタリングします。

> **Note**
>
> すべてのプロジェクトの取得 [所有 = いいえ] には時間がかかる場合があります。

コミットのフィルタリングおよび表示

1. ドロップダウンメニューから、[コミット] を選択します。
2. 入力フィールドを使用してプロジェクト ID を追加します。

3. (オプション) ブランチ/タグでフィルタリングするには、入力フィールドを使用してブランチ/タグのリファレンスを追加します。

問題点のフィルタリングおよび表示

1. ドロップダウンメニューから、[問題点] を選択します。
2. 入力フィールドを使用してプロジェクト ID を追加します。
3. (オプション) タイトル/説明でフィルタリングするには、入力フィールドを使用して、[タイトル] と [説明] に基づいて問題を検索します。

リリースの表示

1. ドロップダウンメニューから [デプロイ] を選択します。
2. 入力フィールドを使用してプロジェクト ID を追加します。
3. (オプション) 環境/ステータスでフィルタリングするには、入力フィールドを使用します。[ステータス] は、`created`、`running`、`success`、`failed`、`canceled` のいずれかの値になります。

ラベルの表示

1. ドロップダウンメニューから、[ラベル] を選択します。
2. 入力フィールドを使用してプロジェクト ID を追加します。

テンプレートおよび変数

新しい GitLab クエリ変数を追加するには、「クエリ変数の追加」を参照してください。ご使用の GitLab データソースをデータソースとして使用します。リソースタイプを [リリース]、[プロジェクト]、または [ラベル] から選択します。

プロジェクト、ラベルなどの動的リストを選択して取得するには、クエリタイプ変数を作成します。クエリタイプの変数は GitLab クエリエディタを使用して、プロジェクト、ラベルなどをクエリして返します。次の例では、ご使用のクエリをパラメータ化するプロジェクト変数を作成します。

クエリをパラメータ化するプロジェクト変数の作成

1. **`project`** という名前の [クエリ] タイプの変数を追加します。
2. GitLab データソースを選択して [ダッシュボード上のロード] を更新します。

3. [プロジェクト] リソースタイプを選択し、[所有] に [あり]、[表示フィールド] に [名前]、[値フィールド] に [ID] を選択します。
4. [更新] を選択してダッシュボードに変数を追加します。
5. ダッシュボードに新しいパネルを追加して、**$project** をプロジェクト ID として使用します。

   ドロップダウンから選択すると、そのプロジェクトに属する結果を得られるようになりました。

Grafana からの変換を使用して一般的な質問に答える

これで、基本的な GitLab クエリを実行してコミットや問題点などを検索できるようになりました。変換を使用してデータセットを視覚化、集計、グループ化、結合したり、他のタイプの変換を使用して単純な結果を複雑な質問に対する回答に変換したりできます。以下は、いくつかの一般的な質問と、変換を使用して回答する方法です。

プロジェクトでのコミット/問題点/デプロイは 1 日に何回ですか?

1. クエリを追加します。リソースタイプの [コミット] を選択してプロジェクト ID を追加します。
2. 新しい [グループ化の条件] 変換を追加: [グループ化の条件] では、[create_at_date] を選択してから [(Count)=id] を計算します。
3. [グラフ] の視覚化を選択します。

プロジェクトで問題点を終了するまでの平均時間はどのくらいですか?

1. クエリを追加します。リソースタイプの [問題点] を選択してプロジェクト ID を追加します。
2. 新しい [計算変換からフィールドを追加] 変換を追加: [モード] で [バイナリオペレーション] を選択し、[オペレーション] で [closed_at = created_at] を選択し、[エイリアス] で [resolution_time] を選択します。
3. 新しい [計算変換からフィールドを追加] 変換を追加: [モード] で [バイナリオペレーション] を選択し、[オペレーション] で [resolution_time / 86400000] を選択し、[エイリアス] で [resolution_time] を選択します。

   [すべてのフィールド を置き換え] で [真] を選択します。
4. [統計] 視覚化を選択します。

   - 表示 = 計算
   - 計算 = 平均

- フィールド = [resolution_time]

Honeycomb データソースへの接続

Honeycomb データソースを使用すると、Amazon Managed Grafana 内から Honeycomb メトリクスをクエリおよび視覚化し、Honeycomb トレースにリンクできます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

既知の制限事項

- このデータソースではアドホッククエリをサポートしていません。
- API の制限により、変数エディタは選択列の最初の 1000 個の一意の値のみ返すことができます。
- API の制限により、データソースは過去 7 日間のデータのみクエリできます。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。
2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。
3. [データソースの追加] を選択します。
4. データソースのリストから [Honeycomb] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

[Honeycomb 設定]

名前	説明
Name	データソース名。これはパネル、クエリ、Explore でデータソースを表示する方法です。
Honeyc API key	Honeycomb から取得した API キー。
URL	Honeycomb API の URL。例えば、https://api.honeycomb.io と指定します。
Team	API キーに関連付けられた Honeycomb チーム。

Honeycomb データソースのクエリ

メトリクスをクエリするため、エディタフィールドに値を入力します。

- データセットを選択します。
- デフォルトのクエリは、選択したデータセットの上の COUNT です。
- クエリを絞り込むには、[視覚化] 、[視覚化]、[場所]、[制約事項] 、[グループ化の条件]、 [順]、[制限] などの、残りのフィールドの値を選択します。

テンプレートおよび変数

新しい Honeycomb クエリ変数を追加するには、「クエリ変数の追加」を参照してください。

データセット、列、または列の値を含む変数を作成できます。

- データセットを選択しない場合は、変数にデータセットが含まれます。
- データセットのみを選択した場合は、変数には列名が含まれます。
- データセットと列の両方を選択した場合は、変数に列値が含まれます。エディタの [場所] フィールドを使用して、列の値をさらに制限できます。

Honeycomb UI でのクエリの表示

Honeycomb UI で作成したクエリをダッシュボードパネルから表示するには、グラフ内の任意のポイントを選択し、[Honeycomb で開く] を選択します。

クエリエディタから Honeycomb UI で作成したクエリを表示するには、[Honeycomb で開く] を選択します。

Honeycomb のダッシュボードのインポート

ダッシュボードをインポートするには、「ダッシュボードのインポート」を参照してください。

インポートしたダッシュボードを検索するには、[設定]、[データソース] を選択します。

使用可能な事前作成ダッシュボードを表示するには、Honeycomb データソースを選択して [ダッシュボード] タブを選択します。

Jira データソースへの接続

Jira の問題データを他のソースのアプリケーションパフォーマンスデータと組み合わせ、開発プロセスの全体像を取得します。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

- 問題の作成または解決に基づき注釈を作成し、問題とメトリクスの関係を確認します。
- 平均解決時間や問題スループットなどの、詳細な Jira 統計を追跡します。

Jira データソースを使用するには、Jira プロジェクトへのアクセス許可を持つ Atlassian アカウントが必要です。

既知の制限事項

Jira アドオンのカスタムフィールドタイプはサポートされない場合があります。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。
2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。
3. [データソースの追加] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

4. データソースのリストから [Jira] を選択します。
5. 次の情報を入力します。

   - [名前] に、この Jira データソースの名前を入力します。
   - [URL] に、**`https://bletchleypark.atlassian.net`** などの Atlassian インスタンスのルート URL を入力します。
   - [E メール] に、ユーザー/サービスアカウントの E メールアドレスを入力します。
   - [API トークン] に、ユーザー用に生成された API トークンを入力します。

Jira データソースのクエリ

Jira クエリエディタからフィールドとクエリの問題を選択できます。

Jira データソースは、Jira のバグ、ユーザーストーリー、サポートチケット、またはその他のタスクを表す可能性のある問題についてクエリします。

問題点のフィルタリングおよび表示

1. [フィールド] を選択してドロップダウンを選択し、先行入力を使用し、カスタムフィールドを含む Jira インスタンス内の任意のフィールドから選択します。一部の試すフィールド:

   - [概要] — 問題の名前
   - [エピック名] — 問題が属するエピック
   - [ストーリーポイントの推定] — チームが問題に関して推定したストーリーポイント数

2. 問題をフィルタリングまたはソートします。これを実行するには、有効な JQL 式を入力して、Atlanssian クエリ言語 JQL で [プロジェクト] 、[担当者] 、[スプリント] などのフィールドに基づき問題をフィルタリングまたはソートします。

ここから、テーブルにデータを表示、または Grafana 変換を使用してその問題データを操作し、計算を実行し、データを時系列グラフに変換できます。詳細については、「変換の適用」を参照してください。

時系列クエリ

時系列データを表示するためには、数値フィールドとともに [日付] フィールドを選択してから、グラフの視覚化に切り替えます。例: [スプリント開始日]、[ストーリーポイントの見積り]。

上記の例は、それだけではあまり役に立ちません。数値フィールドは、(多くの場合) 変換から計算されます。[グループ化の条件] に変換を使用すると、[スプリント開始日] 別にグループ化し、[ストーリーポイントの見積り] を要約し、スプリントごとにストーリーポイントを経時的に視覚化できます。変換の詳細については、「変換の適用」をご参照ください。

テンプレートおよび変数

新しい Jira クエリ変数を追加するには、「クエリ変数の追加」を参照してください。ご使用の Jira データソースをデータソースとして使用します。

ダッシュボードで変数を定義して JQL 式で参照できます。例えば、プロジェクトのステータスダッシュボードを作成してプロジェクト間から選択、またはエピックのステータスダッシュボードを作成して異なるエピックを選択、またはタスクのステータスダッシュボードを作成して異なる担当者を選択できます。

プロジェクト、エピック、担当者などの動的リストを選択して取得するには、クエリタイプ変数を作成します。クエリタイプの変数は JQL を使用して問題をクエリし、プロジェクト、エピック、担当者、または問題に関係するものすべてを返します。以下に例を示します。

担当者変数を作成して担当者ごとの問題ステータスを取得

1. **`assignee`** という名前の [クエリ] タイプの変数を追加します。
2. [フィールド: 担当者] を選択します。
3. (オプション) JQL フィルター [project = 'your project'] を追加します。
4. [実行] を選択して担当者のリストを表示します。

5. [更新] を選択してダッシュボードに変数を追加します。
6. ダッシュボードに新しいパネルを追加して JQL を編集し、新しい変数 [assignee = $assignee] を使用してフィルタリングします。

   ドロップダウンから選択すると、そのユーザーに割り当てられた問題のみが表示されるようになります。

複数値の変数を使用すると複数のオプションを選択でき、IN 句の一部として使用できます。例えば、[assignee IN ($assignee)] などです。

Grafana からの変換を使用して一般的な質問に答える

マクロはダッシュボードの時間ウィンドウを参照する変数であり、ダッシュボードウィンドウの範囲内でのみ問題をフィルタリングできます。以下の 2 つのマクロがあります。

- $__timeFrom
- $__timeTo

次の JQL クエリ例では、ダッシュボードの時間ウィンドウ `createdDate >= $__timeFrom AND createdDate <= $__timeTo` 内に作成された問題をフィルタリングします。

データソースの最大限の活用

Grafana の変換やその他の組み込み機能を使用すると、Jira データの表示に便利です。

変換を使用した JQL の拡大

Grafana では多くの変換を選択できますが、以下のような、一部の SQL の機能/能力を JQL に提供する強力な拡張が用意されています。

[グループ化の条件] この変換は、標準 Jira JQL 構文の一部ではない主な機能である、グループ化を提供します。[グループ化の条件] 変換を使用すると、スプリントまたは他の問題フィールドでグループ化して、グループ別に集計し、速度およびストーリーポイントの見積もり対スプリントでの実際の計算などのメトリクスを取得できます。

外部結合 SQL 結合と同様に、共通フィールドで 2 つ以上のクエリを結合できます。これにより、クエリのデータセットを組み合わせ、他の変換を使用して、複数のクエリ/データセットからの値を計算できます。

[計算からフィールドを追加] SQL 式と同様に、この変換を使用すると、他のフィールドの計算に基づいてデータセットに新しいフィールドを追加できます。計算で使用するフィールドは、単一のクエリから、または結合したクエリからのフィールドである可能性があります。また、計算を連結して、計算フィールドから計算を実行できます。

Grafana からの変換を使用して一般的な質問に答える

変換を使用してデータセットを視覚化、集計、グループ化、結合したり、他のタイプの変換を使用して単純な結果を複雑な質問に対する回答に変換したりできます。

スプリントあたりの速度を表示するにはどうすればよいですか?

1. [スプリント名]、[ストーリーポイントの見積り] の各フィールドを選択します。
2. JQL フィルター `project = "Your Project" AND type != epic AND status = done order by created ASC` を追加します。
3. [グループ化の条件] 変換を追加します:
   - スプリント名 | グループ化の条件
   - ストーリーポイントの見積り | 計算 | 合計
4. [棒ゲージ] 視覚化を選択します。

スプリントでの完了済み対推定の内容を表示するにはどうすればよいですか?

1. クエリを追加します。まず、[スプリント名]、[スプリント開始日]、[ストーリーポイントの見積り] の各フィールドを選択します。

   次に、JQL フィルター `project = 'Your Project' AND type != epic` を追加します。
2. 2 つ目のクエリを追加します。まず、[スプリント名]、[スプリント開始日]、[ストーリーポイントの見積り] の各フィールドを選択します。

   次に、JQL フィルター `project = 'Your Project' AND type != epic AND status = done` を追加します。
3. [グループ化の条件] 変換を追加します:
   - スプリント名 | グループ化の条件
   - スプリント開始日 | グループ化の条件
   - ストーリーポイントの見積り | 計算 | 合計

4. [グラフ] の視覚化を選択します。

プロジェクトで問題点を閉じるまでの平均時間はどのくらいですか?

1. クエリを追加します。まず、[作成日]、[ステータスカテゴリ変更済み] の各フィールドを選択します。

   次に、JQL フィルター `project = 'Your Project' AND type != epic AND status = done` を追加します。
2. 変換を追加: [計算からフィールドを追加]
   - モード = 行を減らす
   - 計算 = 差分
3. 変換を追加: [計算からフィールドを追加]
   - モード = バイナリオペレーション
   - オペレーション = 差/86000000
   - エイリアス = 日数
4. 変換の追加: [フィールドの整理]
   - 異なるフィールドを非表示
5. 変換の追加: [値によるデータのフィルタリング]
   - フィルタータイプ = 含める
   - 条件 = いずれかに一致
     - フィールド = 日数 | 一致 = より大きい | 値 = 1
6. 変換の追加: [削減]
   - モード = シリーズから行
   - 計算 = 平均
7. [統計] 視覚化を選択します。

MongoDB データソースに接続する

MongoDB データソースを使用すると、Amazon Managed Grafana の MongoDB からのデータを視覚的に表現することができます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

使用方法

クエリエディタ

クエリエディタは、MongoDB Shell と同じ構文をサポートしていますが、いくつかの制限があります。* 実行できるのは 1 つのコマンド/クエリのみです。* find (検索) と aggregate (集計)の読み取り専用のコマンドのみがサポートされています。* ほとんどのオブジェクトコンストラクターはサポートされていません (ISODate は例外としてサポートされています)

エディタは次の方法で MongoDB Shell 構文を拡張します。

- データベースの選択 — 通常の「db」の代わりにデータベースの名前を指定できます。

  > **Note**
  >
  > 「db」を引き続き使用できます。接続文字列のデフォルトデータベースを参照します。

  ```
  sample_mflix.movies.find()
  ```

- 集計のソート – 通常、ソートは集計パイプライン内のステップで行われますが、MongoDB Atlas の無料プランではソートは実行できません。そのため、無料プランを利用しているユーザー向けに構文を拡張しました。

  > **Note**
  >
  > この構文でソートを実行するのは MongoDB ではなく、ソートは、コレクションから結果がクエリされた後に行われます。

```
sample_mflix.movies.aggregate({}).sort({"time": 1})
```

- エディタが空の状態で [Ctrl + スペース] を押すと、使用可能なすべてのデータベースの選択肢が表示されます。
- データベースの後にドットを入力すると、そのデータベースで使用可能なすべてのコレクションの選択肢が表示されます。
- コレクションの後にドットを入力すると、使用可能なクエリメソッドが表示されます。
- クエリメソッドの後にドットを入力すると、追加の関数 (sort/limit) が表示されます。

クエリの実行

Cmd + S を押すとクエリが実行されます。

時系列

時系列データを視覚化する場合、プラグインにどのフィールドを時間として使用するかを伝える必要があります。これは、単純にそのフィールドを「time」という名前のエイリアスで投影するだけです。このフィールドのデータ型は日付型である必要があります。

日付型以外のデータ型を強制的に日付型にすることができます。これにより、日付以外のフィールドを時系列時間として使用できるようになります。以下に、MongoDB $dateFromParts パイプライン演算子を使用して、int フィールド「year」を日付型に変換した「time」として射影する例を示します。

```
sample_mflix.movies.aggregate([
{"$match": { "year": {"$gt" : 2000} }},
{"$group": { "_id": "$year", "count": { "$sum": 1 }}},
{"$project": { "_id": 0, "count": 1, "time": { "$dateFromParts": {"year": "$_id",
 "month": 2}}}}
]
).sort({"time": 1})
```

診断

診断コマンド

現時点で、「stats」、「serverStatus」、「replSetGetStatus」、「getLog」、「connPoolStats」、「connectionStatus」、「buildInfo」、「dbStats」、「hostInfo」、「lockInfo」の診断コマンドがサポートされています。

例:

```
admin.connectionStatus()  // run the connectionStatus command
admin.connectionStatus({"authInfo.authenticatedUserRoles": 1})  // run and only return
 the "authInfo.authenticatedUserRoles" field
admin.connPoolStats({arg: "pool"})  // run the connPoolStats command and pass 1
 argument
admin.serverStatus({args: {repl: 0, metrics:0}})  // run the serverStatus command and
 pass multiple args
```

マクロ

クエリ内でダッシュボードの時間範囲をすることが参照できます。

- $__timeFrom — ダッシュボードの開始時刻を参照するマクロ
- $__timeTo — ダッシュボードの終了時刻を参照するマクロ

```
        $__timeTo -  ``` sample_mflix.movies.find({released: {$gt:
        "$__timeFrom"}}).sort({year: 1})
```

テンプレート変数

MongoDB では「複合変数」という概念がサポートされており、これを使用することで 1 つの変数を複数の変数として扱い、複雑な複数キーのフィルターを実行することができます。

複合変数を作成するには、_var1_var2 のように変数をアンダースコアで区切る命名規則を使用します (アンダースコアで始める必要があります)。クエリを実行する際、応答は val1-val2 の形式である必要があります。

例:結果を「映画のタイトル」と「年」の両方でフィルタリングしたい場合。

1. クエリ型の変数 _movie_year を作成します。

2. 変数クエリを設定して、以下の例のように「movie-year」プロパティを持つアイテムの配列を返すクエリにします。

```
// Example sample_mflix.movies.aggregate([
          {"$match": {year: {"$gt": 2011}}},
          {"$project": {_id: 0, movie_year: {"$concat":
          ["$title", " - ", {"$toString":"$year"}]}}}
          ])
```

```
 // [{"movie-year": "Ted - 2016"},
          {"movie-year": "The Terminator -
          1985"}]
```

3. これで、クエリ内で「Movie」と「Year」をそれぞれ別のテンプレート変数として、「$_variable」構文を使用して参照できます。

アドホックフィルターの使用

任意の名前の標準の「ad-hoc filter」型変数に加えて、2 番目のヘルパー変数を作成する必要があります。これは「mongodb_adhoc_query」という名前の「定数」型で、クエリエディタと互換性のある値でなければなりません。クエリの結果は、選択可能なフィルターを埋めるために使用されます。この変数はそれ以上の目的がないため、非表示にすることもできます。

```
        sample_mflix.movies.aggregate([
        {"$group": { "_id": "$year"}},
        {"$project": { "year": "$_id","_id":
        0 }} ] )
```

New Relic データソースへの接続

このセクションでは、New Relic APM と Grafana 用 Insights について説明します。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。

さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

機能

- テンプレート変数
  - メトリクス名
  - メトリクス値
- 注釈
- エイリアス定義
  - メトリクス名
  - メトリクス値
- アドホックフィルター
  - 現在サポートされていません
- [アラート]

設定

データソースを追加して、[管理者 API キー]、[個人用 API キー]、および[アカウント ID] の各フィールドに入力します。

使用方法

サービスカテゴリ

- メトリクス: New Relic の REST API を介して New Relic APM をクエリする場合。
- Insights: NRQL 経由で New Relic Insights をクエリする場合。

エイリアス

プレーンテキストを次の変数と組み合わせてカスタム出力を生成できます。

可変	説明	値の例
$__nr_metric	メトリクス名	CPU/ユーザー時間
$__nr_metric_value	メトリクス値	average_value

例:

```
<para>
  Server: $__nr_server Metric: $__nr_metric
</para>
<programlisting>
```

テンプレートおよび変数

1. ダッシュボード用テンプレート変数を作成します。詳細については、「テンプレートおよび変数」を参照してください。
2. 「クエリ」タイプを選択します。
3. 「New Relic」データソースを選択します。
4. 相対 REST API エンドポイント (ファイル拡張子を除く) を使用してクエリを定式化します。

使用可能なアプリケーションのリスト:

```
<para>
  applications
</para>
<programlisting>
```

アプリケーションで使用可能なメトリクスのリスト:

```
<para>
  applications/{application_id}/metrics
</para>
<programlisting>
```

NRQL マクロ

New Relic クエリ言語 (NRQL) のクエリを作成するときの書き込みエクスペリエンスを向上させるため、エディタは事前定義マクロをサポートしています。

- `$__timeFilter` (または `[[timeFilter]]`) は、ダッシュボードの時間範囲に基づいて `SINCE <from> UNTIL <to>` を変更します。

例:

```
<para>
  SELECT average(value) FROM $event_template_variable
  $__timeFilter TIMESERIES
</para>
<programlisting>
```

マクロ変数およびテンプレート変数の使用方法の詳細については、エディタのヘルプセクションを参照してください。

アラートイベント

New Relic のデータソースを選択して追加のフィルターを設定します。フィルターを設定しない場合は、すべてのイベントが返されます。

エンティティ ID でイベントをフィルタリングする場合は、ID の代わりにエンティティ名を選択できる、テンプレート変数を使用します。例えば、特定のアプリケーションのイベントをフィルタリングするには、アプリのリストを取得してエンティティ ID フィルターとして使用する変数 _$app_ を作成します。

デプロイイベント

アプリケーション ID は必須のフィールドです。

Oracle Database のデータソースへの接続

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。

さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

データソースの追加

Grafana の左側パネルで [データソース] を選択します。

以下のようにデータソースの追加を選択します。

`oracle` を入力してデータソースを検索します。

Oracle サーバーの詳細を入力します。

ホスト名 (または IP アドレス) とポート番号、接続するユーザー名およびパスワードを入力します。

tnsnames オプションのトグルを使用すると、tnsnames.ora の設定ファイルにある有効なエントリを、基本認証とともに使用できます。

前の例と似ていますが、認証には Kerberos を使用しています。OS または docker コンテナで kerberos を使用するための設定方法については、kerberos 固有のセットアップガイドを参照してください。

必要に応じて、Oracle サーバーへの接続に使用、およびタイムゾーン対応マクロで使用される、タイムゾーンを変更します。デフォルト設定は UTC です。

データソースを保存してテストすると、「データベース接続 OK」という緑のメッセージが表示されます。

使用方法

マクロ

構文を簡素化し、日付範囲フィルターなどの動的部分を許可するため、クエリにマクロを含めることができます。列名は二重引用符 (") 内に含める必要があります。

マクロの例	説明
$__time(dateColumn) | 列の名前を「time」に変更した式に置き換えられます。	列の名前を `time` に変更し、値を Unix タイムスタンプ (ミリ秒単位) に変換する式に置き換えられます。

マクロの例	説明
例えば、`dateColumn as time` *$__timeEpoch(dateColumn)* は	
$__timeFilter(dateColumn)* \| 指定された列名を使用して時間範囲フィルターに置き換えられます。例えば、`dateColumn BETWEEN TO_DATE('19700101','yyyymmdd') + (1/24/60/60/1000) * 1500376552001 AND TO_DATE('19700101','yyyymmdd') + (1/24/60/60/1000) * 1500376552002` *$__timeFrom() は	DATE データ型に変換された現在アクティブな時間選択の開始に置き換えられます。例えば、`TO_DATE('19700101','yyyymmdd') + (1/24/60/60/1000) * 1500376552001` 。
$__timeTo ()* \| 現在アクティブな時間選択の末尾が「DATE」のデータ型に変換されます。`DATE` data type. *$__timeGroup(dateColumn,"5m") は	GROUP BY 句で使用できる式に置き換えられます。
$__timeGroup(dateColumn,"5m"[, fillvalue]) は	GROUP BY 句で使用できる式に置き換えられます。fillValue に NULL または浮動値を指定すると、時間範囲内の空のシリーズにその値が自動的に入力されます。例えば、*timeGroupcreatedAt,'1m',0.*__timeGroup(dateColumn,"5m", 0)*.
*timeGroup(dateColumn,'5m', NULL) * |Sameasroup(dateColumn,"5m", previous)*	上記と同じですが、しのシリーズの前の値がフィル値として使用されます。まだ値が見つからない場合は NULL が使用されます。 _timeG
$__unixEpochFilter(dateColumn)* \| 指定された列名を使用して、Unix タイムスタンプとして表される時間 (ミリ秒単位) で時間範囲フィルターに置き換えられます。例えば、`dateColumn >= 1500376552001 AND dateColumn <= 1500376552002` *$__unixEpochFrom() は	現在アクティブな時間選択の開始が Unix タイムスタンプに置き換えられます。例えば、`1500376552001` 。

マクロの例	説明
$__unixEpochTo() は	現在アクティブな時間選択の最後が Unix タイムスタンプに置き換えられます。例えば、1500376552002 。

プラグインは中括弧 {} を使用した表記もサポートしています。パラメータ内でクエリが必要な場合はこの表記を使用します。

> **Note**
>
> クエリごとに 1 つの表記タイプを使用します。クエリに中括弧が必要な場合、クエリ内のすべてのマクロで中括弧を使用する必要があります。

```
$__timeGroup{"dateColumn",'5m'}
$__timeGroup{SYS_DATE_UTC("SDATE"),'5m'}
$__timeGroup{FROM_TZ(CAST("SDATE" as timestamp), 'UTC'), '1h'}
```

クエリエディタには、パネル編集モードでクエリが実行された後に表示される [生成 SQL] リンクがあります。リンクを選択すると、リンクが展開されて実行済みの未加工の補間 SQL 文字列が表示されます。

テーブルクエリ

[Format as] クエリオプションが [テーブル] に設定されている場合は、基本的に任意のタイプの SQL クエリを実行できます。テーブルパネルにはクエリが返す列と行の結果が自動的に表示されます。テーブルパネルの列名は、通常の `as` SQL 列選択構文を使用して制御できます。

時系列クエリ

例えば Graph パネルで使用するために [Format as] を [時系列] に設定した場合、クエリは SQL 日時または Unix エポック (秒単位) を表す任意の数値データ型を返す `time` という名前の列を返す必要があります。Grafana は明示的なタイムゾーンがない DATE 列と TIMESTAMP 列を UTC として解釈します。`time` および `metric` を除くすべての列は、値列として扱われます。値列のメトリクス名として使用される `metric` という名前の列を返すことができます。

次のコード例は、`metric` 列を示しています。

```
SELECT
  $__timeGroup("time_date_time", '5m') AS time,
  MIN("value_double"),
  'MIN' as metric
FROM test_data
WHERE $__timeFilter("time_date_time")
GROUP BY $__timeGroup("time_date_time", '5m')
ORDER BY time
```

その他のクエリ – oracle-fake-data-gen を使用します

```
SELECT
  $__timeGroup("createdAt", '5m') AS time,
  MIN("value"),
  'MIN' as metric
FROM "grafana_metric"
WHERE $__timeFilter("createdAt")
GROUP BY $__timeGroup("createdAt", '5m')
ORDER BY time
```

次のコード例は偽データの時系列を示しています。

```
SELECT
  "createdAt",
  "value"
FROM "grafana_metric"
WHERE $__timeFilter("createdAt")
ORDER BY "createdAt" ASC
```

```
SELECT
  "createdAt" as time,
  "value" as value
FROM "grafana_metric"
WHERE $__timeFilter("createdAt")
ORDER BY time ASC
```

次の例は便利なテーブルの結果を示しています。

```
select tc.table_name Table_name
,tc.column_id Column_id
,lower(tc.column_name) Column_name
,lower(tc.data_type) Data_type
,nvl(tc.data_precision,tc.data_length) Length
,lower(tc.data_scale) Data_scale
,tc.nullable nullable
FROM all_tab_columns tc
,all_tables t
WHERE tc.table_name = t.table_name
```

テンプレート作成

メトリクスクエリではサーバー、アプリケーション、センサー名などをハードコーディングする代わりに、変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用すると、ダッシュボード内に表示されるデータを簡単に変更できます。

クエリ変数

タイプ Query のテンプレート変数を追加すると、測定名、キー名、ドロップダウン選択ボックスとして表示されるキー値などを返すことができる Oracle クエリを記述できます。

例えば、テンプレート変数のクエリ設定でこのようなクエリを指定した場合、テーブル内の hostname 列のすべての値を含む変数を指定できます。

```
SELECT "hostname" FROM host
```

クエリは複数の列を返すことができ、Grafana はそれらの列からリストを自動的に作成します。例えば、次のクエリは、 hostname と hostname2 の値を含むリストを返します。

```
SELECT "host.hostname", "other_host.hostname2" FROM host JOIN other_host ON host.city =
 other_host.city
```

クエリで $__timeFilter("time_column") などの時間範囲依存マクロを使用するには、テンプレート変数の更新モードを [時間範囲変更] に設定する必要があります。

```
SELECT "event_name" FROM event_log WHERE $__timeFilter("time_column")
```

もう 1 つのオプションは、キー/値変数を作成できるクエリです。クエリは、 __text と __value という名前の 2 つの列を返します。__text 列の値は一意である必要があります (一意でない場合は、最初の値が使用されます)。ドロップダウンリストのオプションには、わかりやすい名前をテキストとして、ID を値として指定できるテキストと値があります。次のコード例は、hostname をテキストとして、id を値として持つクエリを示しています。

```
SELECT "hostname" AS __text, "id" AS __value FROM host
```

ネストされた変数も作成できます。例えば、region という名前の別の変数がある場合です。次に、ホスト変数に、このようなクエリを含む、現在選択されているリージョンからのホストのみを表示させることができます (region が複数値変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します)。

```
SELECT "hostname" FROM host WHERE region IN('$region')
```

クエリでの変数の使用

テンプレート変数の値は、テンプレート変数が multi-value の場合にのみ引用符で囲まれます。

変数が複数値の変数の場合は、= を使用して複数の値と照合するのではなく、IN 比較演算子を使用します。

2 つの構文があります。

$<varname> hostname という名前のテンプレート変数の例:

```
SELECT
  "atimestamp" as time,
  "aint" as value
FROM table
WHERE $__timeFilter("atimestamp") AND "hostname" IN('$hostname')
ORDER BY "atimestamp" ASC
```

[[varname]] hostname という名前のテンプレート変数の例:

```
SELECT
  "atimestamp" as time,
  "aint" as value
FROM table
WHERE $__timeFilter("atimestamp") AND "hostname" IN('[[hostname]]')
ORDER BY atimestamp ASC
```

Salesforce データソースへの接続

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Salesforce データソースを使用すると、Amazon Managed Grafana 内で Salesforce のデータを視覚化できます。

このデータソースを使用するには、Salesforce アカウントと Salesforce 接続アプリが必要です。

既知の制限事項

- アドホックフィルターは未サポートです。
- 現在、SOQL クエリ、および SOQL 経由でアクセスできるデータのみがサポートされています。SOSL および SAQL のクエリ形式は未サポートです。

必須の設定

次のオプション設定が必要です。

> **Note**
>
> プラグインは現在、OAuth 2.0 のユーザー名パスワードフローを使用しています。接続アプリに必要なコールバック URL は使用しません。したがって、有効な任意の URL に設定することができます。

名前	説明
Enable OAuth settin	OAuth を有効にするにはこれをチェックする必要があります。
Callba URL	このプラグインでは使用されないため、有効な任意の URL を指定できます。
Select OAuth Scopes (minim requir nts)	データ (api) にアクセスして管理します。
Requir Secret for Refres Token Flow	これは有効化または無効化のいずれかを選択できます。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。
2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。
3. [データソースの追加] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに Admin ロールが付与されていないことを意味します。

4. データソースのリストから [Salesforce] を選択します。

5. 次の情報を入力します。

   - [ユーザー名] に、Salesforce の接続とクエリに使用する Salesforce アカウントのユーザー名を入力します。
   - [パスワード] にユーザーのパスワードを入力します。
   - [セキュリティトークン] に、そのユーザーのセキュリティトークンを入力します。
   - [コンシューマーキー] に、Salesforce に接続するコンシューマーキーを入力します。これは Salesforce 接続アプリから取得できます。
   - [コンシューマーシークレット] に、Salesforce に接続するコンシューマーシークレットを入力します。これは Salesforce 接続アプリから取得できます。
   - [サンドボックスを使用] をは、Salesforce サンドボックスを使用する場合に選択します。

Salesforce データソースのクエリ

クエリエディタはクエリビルダーと SOQL エディタのモードをサポートしています。SOQL は Salesforce Object Query Language の略です。

クエリビルダー (SOQL ビルダー)

クエリビルダーは SOQL クエリ構築用の使いやすいインターフェイスです。SOQL クエリの記述に慣れていない場合は、このモードを使用して SOQL を構築し Salesforce オブジェクトをクエリできます。クエリビルダーの [FROM] フィールドは Salesforce のエンティティ (1 つ以上) を参照します。クエリビルダーで他の操作を行う前に、[FROM] フィールドを選択する必要があります。[FROM] フィールドを選択したら、ビルダーモードを選択する必要があります。SOQL ビルダーは現在次のモードをサポートしています。

- `List` — 選択したテーブル/salesforce のフィールドを含む項目を一覧表示します。このモードを使用して、「この四半期に作成された機会のリストを、名前、値、ステージとともに表示」などの結果を取得します。
- `Aggregate` — エンティティ内の項目を集約します。このモードを使用して、「先月作成された機会をカウント」 または「ステージ名によりグループ化した機会の合計値」などの結果を取得します。
- `Trend` — 集計結果を経時的に表示します。このモードを使用して、「CreatedDate による機会数のカウント」 または「機会によりグループ化された値の合計数」などの結果を取得します。

クエリエディタで Entity/FROM と [モード] を選択してから、次のオプションを使用してクエリを構築します。

[フィールド]	適用対象	説明
SELECT	すべて	表示するフィールドのリストを選択します。集計ビューまたはトレンドビューで、値の集計方法も選択します。
WHERE	すべて	(オプション) フィルター条件を指定します。結果は選択した条件に基づいてフィルタリングされます。
ORDER BY	LIST、AGGREGATE	(オプション) 結果に使用するフィールド名とソート順序を選択します。
LIMIT	LIST、AGGREGATE	(オプション) 返される結果数を制限します。デフォルトは 100 です。
グループ化の条件	AGGREGATE	(オプション) 集計値を特定のフィールドで分割する場合にフィールドを選択します。
時間フィールド	トレンド	結果をグループ化するための日付フィールドを指定します。結果は Grafana のタイムピッカー範囲に基づいてフィルタリングされます。

クエリエディタで前述のフィールドを設定すると、生成された SOQL のプレビューもクエリエディタの下に表示されます。クエリビルダーの制限でブロックされる場合は、安全に SOQL エディタに切り替えることができます。SOQL エディタでは生成された SOQL クエリをカスタマイズできます。

SOQL エディタ

未加工 SOQL エディタには、 未加工の SOQL クエリを介して Salesforce オブジェクトをクエリするオプションがあります。SOQL エディタには、テーブルごとに使用可能なエンティティや対応するフィールドなどの、自動入力の提案が用意されています。テーブルごとに使用可能なエンティティを表示するには、SELECT または WHERE の後に Ctrl+スペースを使用します。エンティティ名の後にドットを入力すると、使用可能なフィールドが表示されます。

ショートカット

CTRL + スペースを使用すると、使用可能なコンテキストオプションを示すコード完了が表示されます。

CMD + S ではクエリを実行します。

時系列としてクエリ

日付フィールドを時間に、メトリクスフィールドをメトリクスにエイリアシングして時系列クエリを作り、メトリクスと日付でグループ化します。以下に例を示します。

```
SELECT sum(Amount) amount, CloseDate time, Type metric from Opportunity
group by Type, CloseDate
```

マクロ

ダッシュボードの時間範囲によりフィルタリングする場合は、SOQL クエリでマクロを使用できます。

- $__timeFrom — time データ型に変換された現在アクティブな時間選択の開始に置き換えられます。
- $__timeTo — time データ型に変換された現在アクティブな時間選択の終了に置き換えられます。
- $__quarterStart — 会計四半期の開始 (SalesForce 会計年度設定から派生)。
- $__quarterEnd— 会計四半期の終了 (SalesForce 会計年度設定から派生)。

```
SELECT UserId, LoginTime from LoginHistory where LoginTime > $__timeFrom
```

テンプレートおよび変数

新しい Salesforce クエリ変数を追加するには、「クエリ変数の追加」を参照してください。Salesforce データソースをデータソースとして使用します。ここでは任意の SOQL クエリを使用できます。

ユーザー ID やユーザー名など、名前/値のペアを使用する場合は、SOQL クエリから 2 つのフィールドを返します。最初のフィールドは ID として使用されます。これはクエリエディタ SOQL のキー (ID など) でフィルタリングする場合に実行します。

変数構文を使用して SOQL クエリで 変数を使用します。詳細については、「変数構文」を参照してください。

SAP HANA データソースへの接続

SAP HANA は、データ駆動型のリアルタイムの決定と \ アクションを高速化する、高性能のインメモリデータベースです。SAP によって開発、販売されています。SAP HANA データソースプラグインは、SAP HANA インスタンスを Grafana に接続するために役立ちます。

SAP HANA Grafana Enterprise プラグインを使用すると、ご使用の SAP HANA データを Grafana の他のすべてのデータソースと一緒に視覚化し、コンテキスト内のログデータとメトリクスデータを視覚化できます。このプラグインにはビルトインのクエリエディタが含まれており、注釈がサポートされ、アラートしきい値の設定、アクセスの制御、アクセス許可の設定などを実行できます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

機能

- [クエリエディタ] — プラグインには、時系列またはテーブルデータを視覚化し、基本的な Grafana マクロを自動補完できる、構文強調表示付きの SQL クエリエディタが組み込まれています。
- [データソースのアクセス許可] — Grafana で SAP HANA データを表示またはクエリできるユーザーを制御します。
- [注釈] — SAP HANA イベントまたはデータを Grafana グラフにオーバーレイして、イベントを他のグラフデータに関連付けます。
- [アラート] — SAP HANA にアラートベースのメトリクスストアを設定します。
- [クエリの変数] — Grafana に SAP HANA データに基づくテンプレート変数を作成し、SAP HANA クエリに変数を含めてダッシュボードをインタラクティブにします。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。
2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。

3. [データソースの追加] を選択します。

> **Note**
>
> サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに Admin ロールが付与されていないことを意味します。

4. データソースのリストから [SAP HANA] を選択します。
5. Config エディタで、以下の情報を入力します。

   - [サーバーアドレス] には、SAP HANA インスタンスのアドレスを指定します。例えば、`xxxxxxx-xxxx-xxxx-xxxx-xxxxxxx.hana.trial-us10.hanacloud.ondemand.com` などです。
   - サーバーポート] には、SAP HANA インスタンスのポートを指定します。
   - [ユーザー名] には、SAP HANA インスタンスへの接続に使用するユーザー名を入力します。
   - [パスワード] には、このユーザーのパスワードを入力します。
   - (オプション) TLS 検証をスキップする場合は [TLS 検証のスキップ] を有効にします。
   - (オプション) クライアント証明書とキーを指定する必要がある場合は、[TLS クライアント認証] を有効にします。
   - (オプション) 自己署名 TLS 証明書の認証を有効にする場合は、[CA 証明書を使用] を有効にします。
   - (オプション) [デフォルトスキーマ] には、使用するデフォルトスキーマを入力します。これを省略すると、すべてのクエリでスキーマを指定する必要があります。

アクセスおよび許可

Grafana を SAP HANA に接続するには、専用の認証情報を使用します。ユーザーに必要なアクセス許可のみを提供します。まず、ユーザー名とパスワードを使用して制限付きユーザーを作成します。次のクエリは制限付きユーザーの作成例です。このクエリではパスワード変更の強制も無効になります。

```
CREATE RESTRICTED USER <USER> PASSWORD <PASSWORD> NO FORCE_FIRST_PASSWORD_CHANGE;
```

次に、Grafana などのクライアントを介して、システムを以下に接続することをユーザーに許可します。

```
ALTER USER <USER> ENABLE CLIENT CONNECT;
```

最後に、必要なビュー、テーブル、スキーマへのアクセス許可をユーザーに付与します。

```
ALTER USER <USER> GRANT ROLE PUBLIC;
GRANT SELECT ON SCHEMA <SCHEMA> TO <USER>;
```

ユーザーレベルのアクセス許可

データソース設定ページの [アクセス許可] タブをクリックして、データソースのアクセス許可を有効にすることで、SAP HANA へのアクセスを制限します。アクセス許可ページでは、管理者はアクセス許可を有効にし、クエリのアクセス許可を特定のユーザーとチームに制限できます。

クエリエディタ

SAP HANA Grafana プラグインには、任意の HANA クエリを入力できる SQL クエリエディタが用意されています。クエリが時系列データを返す場合は、グラフパネルでそれらを視覚化するための時系列としてフォーマットできます。クエリエディタには、サポートされている Grafana マクロと SQL クエリの構文強調表示の自動補完が用意されています。

注釈

SAP HANA クエリは Grafana 注釈のソースとして使用できます。注釈クエリは、少なくとも 1 つの時間列と 1 つのテキスト列を返す必要があります。注釈の詳細については、「注釈」を参照してください。

SAP HANA から注釈を作成する方法

1. [ダッシュボード設定] 歯車アイコンを選択します。
2. 左メニューから、[注釈]、[新規] を選択します。
3. [データソース] ドロップダウンメニューから、SAP HANA データソースのインスタンスを選択します。
4. [クエリ] フィールドに、少なくとも 1 つの時間フィールドと 1 つのテキストフィールドを返す SAP HANA クエリを入力します。
5. [フォーマット形式] で、[時系列] を選択します。
6. 注釈ごとに、[From] フィールドを設定します。

テンプレートおよび変数

新しい SAP HANA クエリ変数を追加するには、「クエリ変数の追加」を参照してください。SAP HANA データソースをデータソースとして使用します。

次のクエリ例では、 `users` テーブルの `username` の個別のリストを返します。

```
select distinct("username") from "users"
```

> **Note**
>
> 変数クエリでは必ず 1 つの列のみを選択してください。クエリが 2 つの列を返す場合は、最初の列は表示値として使用され、2 番目の列は変数の実際の値として使用されます。クエリが 3 つ以上の列を返す場合、それらは拒否されます。

テンプレートおよび変数

クエリでは任意の Grafana 変数を使用できます。次の例は、クエリで単一/複数の変数を使用する方法を示しています。

```
-- For example, following query
select * from "users" where "city" = ${city}
-- will be translated into
select * from "users" where "city" = 'london'
--- where you can see ${city} variable translated into actual value in the variable
```

テキストと同様に、変数は数値フィールドでも機能します。次の例で、`${age}` は、数値を受け入れ、テーブル内の数値フィールドと比較するテキストボックスの変数です。

```
select * from "users" where "age" > ${age}
--- wil be translated into
select * from "users" where "age" > '36'
```

変数が複数の値を返す場合は、次のような SAP HANA クエリの `in` 条件で使用できます。SAP HANA で `where in` 条件を有効にするための、変数を囲む角括弧に注意してください。

```
select * from "users" where "city" in (${cities})
```

```
--- will be translated into
select * from "users" where "city" in ('london','perth','delhi')
--- where you can see ${cities} turned into a list of grafana variables selected.
--- You can also write the same query using shorthand notation as shown below
select * from "users" where "city" in ($cities)
```

マクロ

- `$__timeFilter(<time_column>)` — 未加工のクエリで使用する場合、Grafana の時間範囲を指定列に適用します。time/timestamp/long 時間列に適用されます。
- `$__timeFilter(<time_column>, <format>)` - 上記と同じ。ただし、データベースに保存される time_column の形式を指定できます。
- `$__timeFilter(<time_column>, "epoch", <format>)` — 上記と同じですが、時間列がエポックであるときに使用できます。 形式は、「s」、「ms」、「ns」のいずれかです。
- `$__fromTimeFilter(<time_column>)` — 上記と同じですが、時間列がエポックであるときに使用できます。 形式は、「s」、「ms」、「ns」のいずれかです。
- `$__fromTimeFilter(<time_column>, <comparison_predicate>)` — 上記と同じですが、 comparison_predicate を指定できます。
- `$__fromTimeFilter(<time_column>, <format>)` — 上記と同じですが、時間列の形式を指定できます。
- `$__fromTimeFilter(<time_column>, <format>, <comparison_predicate>)` — 上記と同じですが、 comparison_predicate を指定できます。
- `$__toTimeFilter(<time_column>)` — grafana の [to time over a time] フィールドに基づく時間条件を返します。
- `$__toTimeFilter(<time_column>, <comparison_predicate>)` — 上記と同じですが、comparison_predicate を指定できます。
- `$__toTimeFilter(<time_column>, <format>)` — 上記と同じですが、時間列の形式を指定できます。
- `$__toTimeFilter(<time_column>, <comparison_predicate>)` — 上記と同じですが、comparison_predicate を指定できます。
- `$__timeGroup(<time_column>, <interval>)` — 時間列を間隔グループに展開します。time/timestamp/long 時間列に適用されます。

$__timeFilter(<time_column>) マクロ

次の例では、$__timeFilter(<time_column>) マクロについて説明します。

```
- In the following example, the query
select ts, temperature from weather where $__timeFilter(ts)
--- will be translated into
select ts, temperature from weather where ts > '2021-02-24T12:52:48Z' AND ts <
 '2021-03-24T12:52:48Z'
--- where you can see the grafana dashboard's time range is applied to the column ts in
 the query.
```

$__timeFilter(<time_column>, <format>) マクロ

場合によっては、データベース内の時間列がカスタム形式で保存されます。次の例では、$__timeFilter(<time_column>, <format>) マクロについて説明します。このマクロは、grafana のタイムピッカーに基づいてカスタムタイムスタンプをフィルタリングするのに役立ちます。

```
SELECT TO_TIMESTAMP("TS",'YYYYMMDDHH24MISS') AS METRIC_TIME , "VALUE" FROM "SCH"."TBL"
 WHERE $__timeFilter("TS","YYYYMMDDHH24MISS") -- TS is in 20210421162012 format
SELECT TO_TIMESTAMP("TS",'YYYY-MON-DD') AS METRIC_TIME , "VALUE" FROM "SCH"."TBL" WHERE
 $__timeFilter("TS","YYYY-MON-DD") -- TS is in 2021-JAN-15 format
```

そのマクロでは、形式をタイムスタンプ列に一致するいずれかの有効な HANA 形式にすることができます。例えば、YYYYMMDDHH24MISS はデータが 20210421162012 形式で保存される場合に有効な形式です。

$__timeFilter(<time_column>, "epoch" <format>) マクロ

場合によっては、タイムスタンプはデータベースにエポックタイムスタンプとして保存されます。次の例では、$__timeFilter(<time_column>, "epoch" <format>) マクロについて説明します。このマクロは、grafana のタイムピッカーに基づいてエポックタイムスタンプをフィルタリングするのに役立ちます。マクロでは、形式は ms、s、または ns のいずれかです。指定しない場合、s がデフォルトの形式として扱われます。

```
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP") AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch") -- Example : TIMESTAMP field
 stored in epoch_second format 1257894000
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP") AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch","s") -- Example : TIMESTAMP field
 stored in epoch_second format 1257894000
```

```
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP"/1000) AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch","ms") -- Example : TIMESTAMP field
 stored in epoch_ms format 1257894000000
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP"/1000000000) AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch","ns") -- Example : TIMESTAMP field
 stored in epoch_nanoseconds format 1257894000000000000
```

$__timeFilter に 3 番目の引数を使用する代わりに、epoch_s、epoch_ms、または epoch_ns のいずれかを 2 番目の引数として使用できます。

```
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP"/1000) AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch","ms")
-- is same as
SELECT ADD_SECONDS('1970-01-01', "TIMESTAMP"/1000) AS "METRIC_TIME", "VALUE" FROM
 "SCH"."TBL" WHERE $__timeFilter("TIMESTAMP","epoch_ms")
```

$__fromTimeFilter() および $__toTimeFilter() マクロ

`$__fromTimeFilter()` マクロは、Grafana タイムピッカーの from 時間に基づく時間フィールドにわたる条件に拡張されます。

これには 3 つのパラメータを使用できます。最初のパラメータは時間フィールドの名前です。2 番目の引数として、comparison_predicate または時刻列の形式を渡すことができます。その両方を渡す場合、format が 2 番目のパラメータとなり、3 番目のパラメータとして comparison_predicate を使用します。

<format> 形式が指定されない場合、プラグインは時間列が timestamp/date タイプであると想定します。時間列が timestamp/date 以外の形式で保存されている場合は、その形式を 2 番目の引数として渡します。<format> は、epoch_s、epoch_ms、epoch_ns、または YYYY-MM-DD などの他のカスタム形式のいずれかになります。

<comparison_predicate> オプションパラメータ。渡されない場合、プラグインは比較述語として > を使用します。<comparison_predicate> は =、!=、<>、 < 、<=、 > 、>= のいずれかになります。

`$__toTimeFilter()` は、$__fromTimeFilter() と同じように機能します。Grafana の from 時間を使用する代わりに、to 時間を使用します。また、デフォルトの比較述語は < になります。

$__timeGroup(<time_column>, <interval>)

例えば、マクロ $__timeGroup(timecol,1h) はクエリで SERIES_ROUND("timecol", 'INTERVAL 1 HOUR') に拡張されます。

次の例では $__timeGroup(<time_column>, <interval>) macro. について説明します。

```
SELECT $__timeGroup(timestamp,1h),  "user", sum("value") as "value"
FROM "salesdata"
WHERE $__timeFilter("timestamp")
GROUP BY $__timeGroup(timestamp,1h), "user"
ORDER BY $__timeGroup(timestamp,1h) ASC
```

これは次のクエリに変換され、$__timeGroup(timestamp,1h) は SERIES_ROUND("timestamp", 'INTERVAL 1 HOUR') に拡張されます。

```
SELECT SERIES_ROUND("timestamp", 'INTERVAL 1 HOUR') as "timestamp",  "user",
 sum("value") as "value"
FROM "salesdata"
WHERE "timestamp" > '2020-01-01T00:00:00Z' AND "timestamp" < '2020-01-01T23:00:00Z'
GROUP BY SERIES_ROUND("timestamp", 'INTERVAL 1 HOUR'), "user"
ORDER BY "timestamp" ASC
```

> **Note**
>
> グループを $__timeGroup マクロで使用する場合は、選択した [並べ替え] フィールドが [group by] フィールドと同じ名前になっていることを確認してください。異なる場合は、HANA がクエリを認識しない可能性があります。

$__timeGroup() 関数で間隔をハードコーディングしない場合は、間隔として $__interval を指定して Grafana に残すことができます。Grafana はダッシュボードの時間範囲からその間隔を計算します。クエリの例:

```
SELECT $__timeGroup(timestamp, $__interval), sum("value") as "value"
FROM "salesdata"
WHERE $__timeFilter("timestamp")
GROUP BY $__timeGroup(timestamp, $__interval)
ORDER BY $__timeGroup(timestamp, $__interval) ASC
```

このクエリはダッシュボードの時間範囲に基づいて後続クエリに変換されます。

```
SELECT SERIES_ROUND("timestamp", 'INTERVAL 1 MINUTE'), sum("value") as "value"
FROM "salesdata"
```

```
WHERE "timestamp" > '2019-12-31T23:09:14Z' AND "timestamp" < '2020-01-01T23:17:54Z'
GROUP BY SERIES_ROUND("timestamp", 'INTERVAL 1 MINUTE')
ORDER BY SERIES_ROUND("timestamp", 'INTERVAL 1 MINUTE') ASC
```

[アラート]

Grafana で SAP HANA アラートを設定する方法

1. ダッシュボード内にグラフパネルを作成します。
2. 時系列形式で SAP HANA クエリを作成します。
3. [アラート] タブを選択してアラート条件を指定します。
4. [ルールのテスト] を選択してアラートクエリをテストします。
5. アラートの受信者、メッセージ、エラー処理を指定します。
6. ダッシュボードを保存します。

時系列以外のデータのアラート

時系列以外のデータをアラートするには、TO_TIMESTAMP('${__to:date}') マクロを使用して時系列以外のメトリクスを時系列にします。これによりメトリクスが単一ポイントの時系列クエリに変換されます。クエリの形式は以下のようになります。

```
SELECT TO_TIMESTAMP('${__to:date}'),  <METRIC> FROM <TABLE# WHERE <YOUR CONDITIONS>
```

次の例では、テーブルにユーザー名、年齢、都市、ロールという 4 つのフィールドがあります。このテーブルには時間フィールドがありません。開発ロールを持つユーザー数が 3 未満になったときに通知させます。

```
SELECT  TO_TIMESTAMP('${__to:date}'), count(*) as "count" FROM (
   SELECT 'John' AS "username", 32 AS "age", 'Chennai' as "city", 'dev' as "role" FROM
 dummy
   UNION ALL SELECT 'Jacob' AS "username", 32 AS "age", 'London' as "city",
 'accountant' as "role" FROM dummy
   UNION ALL SELECT 'Ali' AS "username", 42 AS "age", 'Delhi' as "city", 'admin' as
 "role" FROM dummy
   UNION ALL SELECT 'Raja' AS "username", 12 AS "age", 'New York' as "city", 'ceo' as
 "role" FROM dummy
   UNION ALL SELECT 'Sara' AS "username", 35 AS "age", 'Cape Town' as "city", 'dev' as
 "role" FROM dummy
```

```
   UNION ALL SELECT 'Ricky' AS "username", 25 AS "age", 'London' as "city",
 'accountant' as "role" FROM dummy
   UNION ALL SELECT 'Angelina' AS "username", 31 AS "age", 'London' as "city", 'cxo' as
 "role" FROM dummy
) WHERE "role" = 'dev'
```

ServiceNow データソースへの接続

これは、ServiceNow インスタンスへの接続に使用する ServiceNow データソースです。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

機能

- クエリ
  - 統計 API クエリ
  - テーブル API クエリ
    - インシデント、変更、その他のテーブル
- アラート
- 注釈 (ベータ版機能)
- テンプレート変数

設定

Grafana の左側パネルでデータソースを選択します。

以下のようにデータソースの追加を選択します。

`servicenow` を入力してデータソースプラグインを検索します。

ServiceNow URL を入力します。

[保存してテスト] を選択します。「ServiceNow 接続 OK」という緑色のメッセージが表示されます。

ダッシュボードの例

事前に作成されたダッシュボードがプラグインと共に含まれており、ダッシュボードタブのデータソース設定ページからインポートできます。

使用方法

クエリエディタでデータを返す 2 つの方法があります。

- TableAPI
- AggregateAPI

現在ユーザーは次のような事前定義されたテーブルのクエリを選択できます。

- 変更
- インシデント

または、`v1.4.0` では、[その他 (カスタムテーブル)] オプションを使用したテーブルとフィールドの API 駆動型リストです。このオプションを使用すると、ServiceNow データソースのセットアップに使用したユーザーが利用可能なテーブルのデータをクエリできます。

[カスタムテーブル] オプションは、事前定義されたテーブルリストと同じ機能をすべてサポートしている必要があります。

TableAPI クエリ

TableAPI はテーブルパネルの表示に適したデータを返します。これにより、順番に選択したフィールドに追加のフィルタリングオプションを表示できるようになります。クエリエディタには、クエリにより返される行数を制限するためのフィールドも用意されています。

前のクエリの結果を示すテーブルパネル例。

[表示]

表示行には、表示するフィールドのセレクターが表示されます。また、複数のフィールドを指定できます。フィールドは指定された順序で返されます。

値の表示

値の表示フラグでは、数値の代わりに人間にわかりやすい値でクエリが返されるか、値が表示されます。

例えば、このフラグがない 1 の重要度では、1 のみが表示されます。フラグが有効な場合、表示される値は 1 - High になります。

ServiceNow API ドキュメントによると、これによりパフォーマンスに悪影響が及ぶ可能性があります。

> **Note**
>
> 表示値を示す [...] では、データベースから直接読み取られず、他のフィールドやレコードの参照が含まれる可能性があるため、パフォーマンスの問題が発生する可能性があります。

フィルター (全般)

フィルター行では、複数のフィールドと値の条件に基づき、表示される行を絞り込むことができます。

すべてのフィルターは AND または OR オペレーションと組み合わされます。

カスタムテーブルを使用しない場合は次のフィールドを使用できます (このリストは今後拡張されます)。

```
Active
Asset
Group
Assigned To
Escalation
Issue Number
Description
Priority
State
Type
Change Risk
Change State
Start Date
End Date
```

```
On Hold
```

カスタムテーブルを選択すると、Service Now API からフィールドに自動入力されます。

日付フィルター

時間フィールド	演算子	値
Opened At	今日または以前 今日以外 当日または以前 当日または以降	timestamp javascript:gs.daysAgo(30)
Activity Due		
Closed At		
Due Date		
Expecte Start		
Reopen Time		
Resolve At		
Work End		
Work Start		
Ignore Time		

その他の日付値の詳細については、以下を参照してください: https://developer.servicenow.com/app.do#!/api_doc?v=newyork&id=r_SGSYS-dateGenerate_S_S

演算子 (全般、文字列ベース)

- 先頭が一致
- Ends With
- Like
- Not Like
- Equals (等しい)
- Not Equals
- Is Empty

演算子 (時間ベース)

- Today
- Not Today
- 前
- 今日または以前
- 後
- At or After

[Values] (値)

値の選択は選択したフィルターのタイプによって異なります。

- ブールフィルターには、真/偽 オプションがあります
- テキストフィルターでは、任意の値を入力できます
- エスカレーション、優先度には固定された数値セットがあります

並べ替え

並び替え行では、複数のフィールドと値の条件に基づき、表示される行を絞り込むことができます。

すべてのフィルターは AND オペレーションと組み合わされます。さらに演算子のサポートが追加されます。

[制限]

行の制限を指定して、返されるデータ量を制限できます。デフォルト値は 25 です。

時間フィールド

Time Field では、クエリされたデータを時系列に変換します。データが時系列として処理されると、選択した「時間フィールド」の値がダッシュボード/パネルの時間範囲にない場合は表示されません。

デフォルトの時間フィールドは「Opened At」ですが、使用可能な時間値を持つ任意のフィールドに変更できます。

「Ignore Time」という特別な値では、「up until now」の結果を使用でき、フィルターを使用して表示するデータを制御できます。

AggregateAPI クエリ (統計)

AggregateAPI は常にメトリクスを返し、avg、min、max、sum の集計が含まれます。また、フィルタリングはクエリを絞り込むためにも使用できます。

[表示]

表示行には、表示するメトリクスのセレクターが表示されます。また、複数のメトリクスを指定できます。

フィルター (全般)

集計[フィルター]には、テーブルオプションのような、フィールドと値の基準に基づいて表示メトリクスを絞り込む機能が用意されています。

すべてのフィルターは AND オペレーションと組み合わされます。さらに演算子のサポートが追加されます。

統計フィルターオプションは TableAPI と同じです。

集計

メトリクスの集計には 4 つのタイプと「カウント」があります。

- 平均
- 最小値

- 最大値
- 合計
- カウント - クエリによって返されるメトリクス「数」を返します。

グループ化の条件

このセレクタを使用すると、メトリクスをより小さい集計に分割できます。「優先度」でグループ化すると、優先度の「タグ」および一意の値が区切られたメトリクスが返されます。

テンプレート作成

クエリで名前をハードコーディングする代わりに、その場所に変数を使用できます。変数は、ダッシュボードの上部にドロップダウン選択ボックスとして表示されます。これらのドロップダウンボックスを使用して、ダッシュボード上に表示されるデータを変更できます。

クエリ変数を追加して、それらをテンプレート値で参照する方法については、クエリ変数セクションの例を参照してください。

クエリ変数

`Query` タイプのテンプレート変数を追加すると、カテゴリ名、キー名、ドロップダウン選択ボックスとして表示されるキー値などの項目を返すクエリを記述できます。

例えば、テンプレート変数クエリ設定でこのようなクエリを指定した場合、`categories` のすべての値を含む変数を指定できます。

クエリ設定を選択すると、フィルターセクションが表示され、タイプとフィールドを選択できます。現在、タイプは Incidents と Changes に限定されています。タイプを選択すると、そのタイプで使用できるフィールドのリストが表示されます。タイプとフィールドを選択すると、そのタイプ/フィールドで使用できるオプションを示す値のプレビューが、下部に表示されます。これらの値はダッシュボードのドロップダウンリストに表示され、テンプレート作成と一緒に使用して、ダッシュボードパネルのデータをフィルタリングできます。

例えば、category という名前の変数を追加し、タイプ = Incidents、フィールド = Category を選択すると、Category のオプションリストが表示されます。次にフィルターをパネルに追加し、[Category Equals ${category}] を選択すると、パネルデータに、ダッシュボードのドロップダウンリストから選択されたそのカテゴリのデータのみが表示されます。

Incidents By Category ダッシュボードをインポートして例を表示します。

クエリでの変数の使用

2 つの構文があります。

$<varname> hostname という名前のテンプレート変数の例:

[[varname]] hostname という名前のテンプレート変数の例:

[アラート]

標準 Grafana アラートがサポートされています。グラフパネルで定義されたクエリは、アラート生成に使用できます。

以下はクエリとアラートの例です。このクエリは、重要度の高い未解決のインシデントのグラフすべてを返します。

このアラートは、重要度の高い未解決のインシデントが 5 件以上ある場合に開始されます。

アラートルールをテストするとアラートルールからの出力が表示され、状態履歴を選択すると OK からアラート保留に変換されたアラートが表示されます。

グラフビューに垂直線が表示され、アラートが保留中の間は上部のハートアイコンがオレンジ色に変わります。

アラートの基準が満たされると、ルールは赤に変わります。

グラフビューに赤い垂直線が表示され、上部のハートアイコンは赤に変わります。

アラートのインシデントの書き込み

ベータ版機能

- ServiceNow データソース用の通知チャネルを設定します。

これにより、設定したユーザーを使用してこのデータソースの ServiceNow インスタンスにインシデントを作成する、Grafana 通知チャネルが設定されます。

このアクションでは、ServiceNow データソースユーザーにインシデント記述のためのアクセス許可を付与する必要があります。

注釈

Grafana 注釈はこのデータソースの v1.4.0 時点でのベータ版機能です。注釈を使用するとイベントをグラフにオーバーレイできます。

注釈クエリは標準クエリエディタと同じオプションをサポートしていますが、小さな違いがいくつかあります。

- 選択できる「表示」列は 1 つのみです。これは今後の改善で修正される可能性があります。
- 時間フィールドは必須です。

よくある質問

ITSM ロールプラグインがない場合はどうしますか？

次のアクションの実行には管理者アクセス許可が必要です

オプション 1: すべてのテーブルへのアクセス許可を Grafana ユーザー管理者に付与します。

オプション 2: ロールを作成し、Grafana がアクセスする必要があるすべてのテーブルに ACL を適用します。

次のアクションの実行には管理者アクセス許可が必要です。

1. ログインした管理者は security_admin にアクセス許可を昇格する必要があります。
    a. 右上ナビゲーションペインで、プロファイルアイコンを選択します。プロファイルアイコンには、ドロップダウンキャレットインジケータがあります。
    b. ドロップダウンリストから [ロールの昇格] を選択します。
    c. 表示されているモーダルから、[security_admin] チェックボックスを選択します。
    d. [OK] を選択してください。
2. お好みの命名規則で新しいロールを作成します。
    a. 左側のナビゲーションで、[システムセキュリティ] => [ユーザーとグループ] => [ロール] のロールセクションに移動します。
    b. 上部で [新規] を選択します。
    c. ロールの名前および関連する説明を入力します。
    d. [Submit] を選択してください。
3. 新しいユーザーを作成、または必要なロールを持つ既存のユーザーを修正します。
    a. ステップ 2 で作成したロール
    b. personalize_dictionary
    c. personalize_choices

d. cmdb_read (これにより、すべての cmdb テーブルへの読み取りアクセス許可が付与されます)

4. 必要なテーブルとフィールドのためにテーブル ACL を作成します。

   - sys_db_object テーブルのために ACL を作成します。

     i. 2 番目の検索ヘッダー列 [名前] に、**`sys_db_object`** と入力して、Enter を押します。

     ii. フィルタリングされた結果に [テーブル] が表示されます。[テーブル] を選択してレコードに移動します。

     iii. タブセクションで、[コントロール] を選択します。

     iv. ページの下部で、選択したタブが [アクセスコントロール] であることを確認します。

     v. [新規] をクリックして新しい ACL を作成します。

     vi. [オペレーション] の選択を読み取りに変更します。

     vii. 画面下部の [ロールが必要] セクションで、[新しい行を挿入] を選択し (ダブルクリック)、作成したロールを検索します。

     viii. 作成済みのロールを選択したら、緑色のチェックマークを選択します。

     ix. 画面の下部で [送信] を選択して ACL を作成し、モーダルが表示されたら [続行] を選択します。

5. 特定の sys_db_object フィールドに ACL を作成します。名前、ラベル、表示名、拡張テーブルの各フィールドで、次のステップを繰り返す必要があります。

   a. sys_db_object のテーブルレコードビューのまま、画面の上部に最も近いタブグループの [列] タブを選択します。

   b. フィールド名を見つけて選択します。

   c. 下部タブセクションの [アクセスコントロール] タブで [新規] を選択します。

   d. オペレーションの読み取りへの変更

   e. 下部の「ロールが必要」テーブルで、[行テキストを挿入] を選択します (ダブルクリック)。

   f. 作成済みのロールを検索して、緑のチェックマークを選択します。

   g. [Submit] を選択してください。

   h. 必須フィールドである、名前、ラベル、表示名、拡張テーブルのすべてで、これらのステップを繰り返したことを確認してください。

6. 変更、インシデント、および Grafana からクエリする他の CMDB ではないテーブルで、4.1 の手順を繰り返します。4.2 のステップは繰り返しないでください。そのステップは sys_db_object にのみ必要です。

Snowflake データベースへの接続

Snowflake Enterprise データソースを使用すると、Snowflake データを Grafana の他のすべてのデータソースと一緒に視覚化し、コンテキスト内のログデータとメトリクスデータを視覚化できます。このデータソースには強力な先行入力のクエリエディタが含まれており、複雑な注釈、アラートしきい値の設定、アクセスとアクセス許可の制御などがサポートされます。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

概要

Snowflake とは

Snowflake は、「データウェアハウスアズアサービス」と一般的に呼ばれるクラウドベースのデータストレージおよび分析サービスを提供します。これは、データウェアハウス、データレイク、データエンジニアリング、データサイエンス、データアプリケーション開発、データ共有のためのソリューションを提供します。過去数年間、Snowflake はクラウドベースのハードウェアとソフトウェアを使用して手頃な価格でデータを保存および分析できるため、大きな人気を得ています。最近では、最大のソフトウェア IPO になりました。現在、多くの企業がプライマリデータベースとして Snowflake を使用して、トランザクションカウント、アクティブなユーザーセッション、さらには時系列やメトリクスデータなどのアプリケーションやビジネスデータを保存しています。

Snowflake および Amazon Managed Grafana の最大限の活用

Snowflake データを移動せずに視覚化する: Grafana の独自のアーキテクチャは、データを移動して冗長ストレージと取り込みに消費するのではなく、データが置かれている場所に直接クエリを実行します。

さまざまなソースからパネルを作成する: 事前に構築されたカスタムダッシュボードを使用すると、さまざまなデータソースのデータを 1 つのウィンドウにまとめることができます。

ユーザーレベルでの変換とコンピューティング: ユーザーはデータを変換し、表示されるデータに対してさまざまな計算を実行できるため、データの準備が少なくて済みます。

パネル内で結合、計算、視覚化: Snowflake やその他のソースからの関連データを表示する混合データソースパネルを作成します。

機能

[クエリエディタ:] クエリエディタは、時系列またはテーブルデータを視覚化し、SQL 構文エラーを処理し、基本的な SQL キーワードを自動補完できる、スマート SQL 自動補完エディタです。

データソースのアクセス許可: Grafana で Snowflake データを表示またはクエリできるユーザーを制御します。

注釈: Snowflake イベントを任意の Grafana グラフにオーバーレイし、イベントを他のグラフデータと関連付けます

アラート: Snowflake にアラートベースのメトリクスストアを設定します

クエリの変数: Snowflake データに基づいて Grafana にテンプレート変数を作成し、Snowflake クエリに変数を含めてダッシュボードをインタラクティブにします。

マルチメトリクスクエリ: 複数のメトリクスを返す単一のクエリを、それぞれ独自の列に書き込みます。

Snowflake プラグインの使用の開始

Grafana の Snowflake プラグインの使用を開始するには、次の 5 つの簡単な手順を実行します。

ステップ 1: Snowflake データソースの設定

データソースを設定するには、[設定]、[データソース]、[データソースの追加]、[Snowflake] を選択します。

認証の詳細を追加すると、データソースのクエリ実行準備が整います。

次の設定フィールドを使用できます。

名前	説明
アカウント	Snowflake のアカウント。
ユーザーネーム	サービスアカウントのユーザー名。

名前	説明
パスワード	サービスアカウントのパスワード。
スキーマ (オプション)	クエリのデフォルトスキーマを設定します。
ウェアハウス (オプション)	クエリのデフォルトのウェアハウスを設定します。
データベース (オプション)	クエリのデフォルトのデータベースを設定します。
ロール (オプション)	クエリのロールを担います。

ステップ 2: Snowflake データのクエリの記述

ダッシュボード内にパネルを作成して、Snowflake データソースを選択しクエリ工ディタの使用を開始します。

- 日付/時刻が含まれている限り、クエリのいずれかの場所に表示される可能性があります。
- 数値列を含める必要があります。これは集計でも int/float 列でもかまいません。
- 時系列データがさまざまなメトリクス用にフォーマットされている場合、必要に応じて文字列列を含めて個別のデータシリーズを作成できます。

Snowflake クエリのレイアウト

```
select
  <time_column>,
  <any_numerical_column>
  <other_column_1>,
  <other_column_2>,
  <...>
from
  <any_table>
where
  $__timeFilter(<time_column>) // predefined where clause for time range
  and $<custom_variable> = 1 // custom variables start with dollar sign
```

間隔による時系列グループ用 SQL クエリ形式

```
select
  $__timeGroup(created_ts, '1h'), // group time by interval of 1h
  <time_column>,
  <any_numerical_column>,
  <metric_column>
from
  <any_table>
where
  $__timeFilter(<time_column>) // predefined where clause for time range
  and $<custom_variable> = 1 // custom variables start with dollar sign
group by <time_column>
```

テーブル用 SQL クエリ形式

```
select
  <time_column>, // optional if result format option is table
  <any_column_1>
  <any_column_2>
  <any_column_3>
from
  <any_table>
where
  $__timeFilter(time_column) // macro for time range, optional if format as option is
 table
  and $<custom_variable> = 1 // custom variables start with dollar sign
```

ステップ 3: テンプレート変数の作成および使用

テンプレート変数の使用

次の例に示すように、クエリにはテンプレート変数を含めることができます。

```
 select
   <column>
 from
   <table>
 WHERE column >= '$variable'
```

次の例は、クエリで複数値の変数を使用する方法を示しています。

```
select
  <column>
from
  <table>
WHERE <column> regexp '${variable:regex}'
```

変数作成のための Snowflake データソースの使用

ダッシュボード設定で、[変数] を選択し、[新規] を選択します。

「Query」変数型を使用して、Snowflake データソースを「データソース」として選択します。

> **Important**
>
> 変数クエリでは必ず 1 つの列のみを選択してください。

例:

```
SELECT DISTINCT query_type from account_usage.query_history;
```

次の変数が提供されます。

```
All DESCRIBE USE UNKNOWN GRANT SELECT CREATE DROP SHOW
```

ステップ 4: アラートのセットアップ

特定の Snowflake メトリクス、または作成したクエリに、アラートを設定できます。

クエリエディタ内のアラートタブボタンを選択して、[アラートを作成] を選択します。

ステップ 5. 注釈の作成

注釈を使用すると、イベントをグラフ上にオーバーレイできます。

注釈を作成するには、ダッシュボード設定で、[注釈]、[新規] を選択して、データソースとして [Snowflake] を選択します。

注釈はイベントであるため、イベントを記述するために、少なくとも 1 つの時間列と 1 つの列が必要です。

次のサンプルコードは、Snowflake へのログイン失敗すべてに注釈を付けるクエリを示しています。

```
SELECT
  EVENT_TIMESTAMP as time,
  EVENT_TYPE,
  CLIENT_IP
FROM ACCOUNT_USAGE.LOGIN_HISTORY
WHERE $__timeFilter(time) AND IS_SUCCESS!='YES'
ORDER BY time ASC;
```

And

- 時間: `TIME`
- タイトル: `EVENT_TYPE`
- テキスト: `CLIENT_IP`

これにより、ダッシュボードパネル上の Snowflake へのログイン失敗すべての注釈がオーバーレイされます。

追加機能

表示名フィールドの使用

このプラグインは、[オプション] パネルの [フィールド] タブにある [表示名] フィールドを使用して、名前、ラベル、または値に基づいて凡例キーを短縮または変更します。他のデータソースでは、カスタム`alias`機能を使用して凡例キーを変更しますが、表示名関数の方がより一貫性のある方法です。

データソースの権限機能

Snowflake へのアクセスを制限するには、データソース設定ページの [アクセス許可] タブを選択してデータソースのアクセス許可を有効にします。アクセス許可ページでは、管理者はアクセス許可を有効にし、クエリのアクセス許可を特定のユーザーとチームに制限できます。

Snowflake の請求と使用状況データについて

Snowflake データソースでは、便利な請求情報と使用状況情報を表示する、請求および使用状況ダッシュボードをインポートできます。

Snowflake データソース設定ページにダッシュボードを追加します。

このダッシュボードは ACCOUNT_USAGE データベースを使用し、クエリには ACCOUNTADMIN ロールが必要です。これを安全に行うには、ACCOUNTADMIN ロールのユーザーを持つ新しい Grafana データソースを作成します。変数でそのデータソースを選択します。

Splunk データソースへの接続

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

設定

データソースの設定

データソースを設定するときは、URL フィールドが `https` を使用して、設定された Splunk ポートを指していることを確認します。デフォルトの Splunk API ポイントは 8000 (デフォルトのウェブ UI ポート) ではなく 8089 です。基本認証を有効にして Splunk のユーザー名とパスワードを指定します。

ブラウザ (直接) アクセスモードおよび CORS

Amazon Managed Grafana は、Splunk データソースのブラウザ直接アクセスをサポートしていません。

詳細オプション

ストリームモード

検索結果が利用可能になったときに取得する場合は、ストリームモードを有効にします。これは実験的機能です。実際に必要になるまで有効にしないでください。

結果のポーリング

検索を実行してから定期的に結果をチェックします。このオプションは内部で、 `exec_mode` を `normal` に設定した `search/jobs` API コールを実行します。この場合、API リクエストはジョブ SID を返し、Grafana はジョブ結果を取得するためにジョブステータスを随時チェックします。このオプションはスロークエリに役立ちます。デフォルトではこのオプションが無効になっており、Grafana は `oneshot` を `exec_mode` に設定します。これにより同じ API コールで検索結果を返すことができます。`search/jobs` の API エンドポイントの詳細については、Splunk のドキュメントを参照してください。

検索ポーリング間隔

このオプションでは、Amazon Managed Grafana が検索結果のために splunk をポーリングする頻度を調整できます。[最小、最大] 間隔からランダムに選択する次のポーリングの時間。大量の検索を実行する場合は、これらの値を増やすことは合理的です。ヒント: 検索ジョブの実行に時間がかかる場合は最小、多くの並列検索 (Grafana ダッシュボードに splunk メトリクスが多い) を実行する場合は最大を増やします。デフォルトは [500、3000] ミリ秒間隔です。

自動キャンセル

指定した場合、この数秒間非アクティブ状態が続くとジョブは自動的にキャンセルされます (0 では自動キャンセルしません)。デフォルトは 30 です。

ステータスバケット

生成する大半のステータスバケットです。0 ではタイムライン情報が生成されません。デフォルトは 300 です。

フィールド検索モード

ビジュアルクエリエディタを使用すると、データソースは選択したソースタイプで使用できるフィールドのリストを取得しようとします。

- クイック - プレビューで最初に利用可能な結果を使用
- フル - ジョブが終了するのを待って完全な結果を取得します。

デフォルトの最も早い時刻

一部の検索ではダッシュボードの時間範囲を使用できません (テンプレート変数クエリなど)。このオプションは、Splunk を遅くする可能性のある、全時間検索を防ぐのに役立ちます。構文は整数で

時間単位は [+|-]<time_integer><time_unit> です。例: -1w。時間単位は s, m, h, d, w, mon, q, y にできます。

変数検索モード

テンプレート変数クエリの検索モードです。使用できる値:

- 高速 - イベント検索でフィールド検出をオフにします。統計検索にはイベントまたはフィールドデータがありません。
- スマート - イベント検索でフィールド検出をオンにします。統計検索にはイベントまたはフィールドデータがありません。
- 詳細 - すべてのイベントおよびフィールドデータ。

使用方法

クエリエディタ

エディタモード

クエリエディタは、未加工とビジュアルの 2 つのモードをサポートしています。これらのモードを切り替えるには、エディタの右側にあるハンバーガーアイコンを選択してエディタモードの切り替えを選択します。

未加工モード

次のコード例に示すように、時系列データに timechart コマンドを使用します。

```
index=os sourcetype=cpu | timechart span=1m avg(pctSystem) as system, avg(pctUser) as
 user, avg(pctIowait) as iowait
index=os sourcetype=ps | timechart span=1m limit=5 useother=false avg(cpu_load_percent)
 by process_name
```

次の例に示すように、クエリはテンプレート変数をサポートしています。

```
sourcetype=cpu | timechart span=1m avg($cpu)
```

Grafana は時系列指向のアプリケーションであり、検索では時系列データ (タイムスタンプと値) または単一値を返すことに注意してください。timechart コマンドと検索例については、公式の「Splunk 検索リファレンス」を参照してください。

Splunk メトリクスおよび **mstats**

Splunk 7.x にはメトリクスを分析するための mstats コマンドが用意されています。mstats でグラフを適切に機能させるには、timeseries コマンドと組み合わせて prestats=t オプションを設定する必要があります。

```
Deprecated syntax:
| mstats prestats=t avg(_value) AS Value WHERE index="collectd"
 metric_name="disk.disk_ops.read" OR metric_name="disk.disk_ops.write" by metric_name
 span=1m
| timechart avg(_value) span=1m by metric_name

Actual:
| mstats prestats=t avg(disk.disk_ops.read) avg(disk.disk_ops.write) WHERE
 index="collectd" by metric_name span=1m
| timechart avg(disk.disk_ops.read) avg(disk.disk_ops.write) span=1m
```

mstats コマンドの詳細については、「Splunk 検索リファレンス」を参照してください。

フォーマット形式

サポートされている結果形式モードには、時系列 (デフォルト) とテーブルの 2 つがあります。集計データを表示するときにテーブルパネルで使用するのに適した、テーブルモード。これは未加工イベント (選択したすべてのフィールドを返す) と stats 検索機能で動作し、テーブルのようなデータを返します。例:

```
index="os" sourcetype="vmstat" | fields host, memUsedMB
index="os" sourcetype="ps" | stats avg(PercentProcessorTime) as "CPU time",
 latest(process_name) as "Process", avg(UsedBytes) as "Memory" by PID
```

その結果は Splunk UI の統計タブに似ています。

stats 機能の使用方法の詳細については、「Splunk 検索リファレンス」を参照してください。

ビジュアルモード

このモードでは段階的に検索を作成できます。このモードでは timechart splunk 検索が作成されることに注意してください。インデックス、ソースタイプ、メトリクスを選択して、必要に応じてフィールドごとの分割を設定します。

メトリクス

メトリクス行の右側にあるプラスボタンを選択すると、検索する複数のメトリクスを追加できます。メトリクスエディタには頻繁に使用される集計のリストが含まれていますが、ここでは他の関数を指定できます。agg セグメント (デフォルトで avg) を選択して必要なものを入力します。ドロップダウンリストから関心のあるフィールドを選択して (または入力)、必要に応じてエイリアスを設定します。

by と where で分割

[Split by] フィールドを設定して[時系列]モードを使用すると、エディタが使用可能になります。プラスを選択して、演算子、集計、値を選択します。例えば、Where avg in top 10 二様に選択します。この Where 句は Split by の一部であることに注意してください。詳細については、timechart のドキュメントを参照してください。

オプション

デフォルトの timechart オプションを変更するには、最後の行の [オプション] を選択します。

これらのオプションの詳細については、timechart のドキュメントを参照してください。

レンダリングされた splunk 検索

左のターゲット文字を選択すると エディタが折りたたまれ、レンダリングされた splunk 検索が表示されます。

注釈

Splunk アラートまたはイベントをグラフに表示する場合は注釈を使用します。注釈は事前定義された Splunk アラートまたは通常の splunk 検索のいずれかです。

Splunk アラート

アラート名を指定、またはフィールドを空白のままにして発生したすべてのアラートを取得します。テンプレート変数がサポートされます。

Splunk 検索

次の例のように、スプランク検索を使用して必要なイベントを取得します。

```
index=os sourcetype=iostat | where total_ops > 400
index=os sourcetype=iostat | where total_ops > $io_threshold
```

テンプレート変数がサポートされます。

フィールド値を注釈テキストとして使用する場合は、テキストとしてのイベントフィールドオプションが適しています。次の例は、ログのエラーメッセージテキストを示しています。

```
Event field as text: _raw
Regex: WirelessRadioManagerd\[\d*\]: (.*)
```

Regex ではメッセージの一部を抽出できます。

テンプレート変数

テンプレート変数機能は `stats` コマンドなどで値のリストを返す Splunk クエリをサポートします。

```
index=os sourcetype="iostat" | stats values(Device)
```

このクエリは `iostat` ソースから `Device` フィールド値のリストを返します。これらのデバイス名は、時系列クエリまたは注釈に使用できます。

Grafana で使用できる変数クエリには 2 つのタイプがあります。1 つ目は単純なクエリで (前述のとおり)、値のリストを返します。2 番目のタイプはキー/値変数を作成できるクエリです。クエリは、`_text` と `_value` という名前の 2 つの列を返します。`_text` 列の値は一意である必要があります (一意でない場合は、最初の値が使用されます)。ドロップダウンリストのオプションにはテキストと値があるため、わかりやすい名前のテキストとして、ID を値として使用できます。

例えば、この検索では列 `Name` (Docker コンテナ名) と `Id` (コンテナ ID) を含むテーブルが返されます。

```
source=docker_inspect | stats count latest(Name) as Name by Id | table Name, Id
```

コンテナ名を変数の可視値として使用し、ID を実際の値として使用するには、次の例のようにクエリを変更する必要があります。

```
source=docker_inspect | stats count latest(Name) as Name by Id | table Name, Id |
 rename Name as "_text", Id as "_value"
```

複数値の変数

クエリでは複数値の変数を使用できます。補間検索は変数使用コンテキストによって異なります。プラグインがサポートしているコンテキストは多数あります。foo と bar の値を選択した変数 $container があるとします。

- search コマンド用基本フィルター

```
source=docker_stats $container
=>
source=docker_stats (foo OR bar)
```

- フィールド値フィルター

```
source=docker_stats container_name=$container
=>
source=docker_stats (container_name=foo OR container_name=bar)
```

- IN 演算子と in( ) 関数を使用したフィールド値フィルター

```
source=docker_stats container_name IN ($container)
=>
source=docker_stats container_name IN (foo, bar)

source=docker_stats | where container_name in($container)
=>
source=docker_stats | where container_name in(foo, bar)
```

複数値の変数と引用符

変数が引用符で囲まれている場合 (二重または単一の両方)、次の例のように、その値も引用符で囲まれます。

```
source=docker_stats container_name="$container"
=>
source=docker_stats (container_name="foo" OR container_name="bar")

source=docker_stats container_name='$container'
=>
source=docker_stats (container_name='foo' OR container_name='bar')
```

Splunk Infrastructure Monitoring データソースへの接続

Splunk Infrastructure Monitoring (旧 SignalFx) のサポートを提供します。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開いて、ログインしていることを確認します。
2. [設定] (歯車アイコン) のサイドメニューで、[データソース] を選択します。
3. [データソースの追加] を選択します。

   > **Note**
   >
   > サイドメニューに [データソース] リンクが表示されていない場合は、現在のユーザーに `Admin` ロールが付与されていないことを意味します。

4. データソースのリストから [Splunk Infrastructure Monitoring] を選択します。
5. 次の情報を入力します。

   - [アクセストークン] に、SignalFx アカウントによって生成されたトークンを入力します。詳細については、「認証トークン」を参照してください。

- Realm (領域)。組織をホストする自己完結型デプロイ。領域名は、SignalFx ユーザーインターフェイスにサインインしたときにプロファイルページで確認できます。

クエリエディタの使用

クエリエディタは SignalFlow プログラム/クエリを受け入れます。

ラベルについて、Signalflow のラベル `publish(label = 'foo')` がメタデータとして結果に適用されます: "label":"foo"

クエリタイプのテンプレート変数には Query (クエリ) フィールドがありません。代わりに、次のいずれかのクエリタイプを選択します。

- ディメンション
- メトリクス
- [タグ]

アドホックフィルターがサポートされており、ディメンションを使ったグローバルフィルターが利用できます。

Grafana の注釈がサポートされています。注釈を作成するときは、SignalFlow アラートまたはイベントクエリを使用します。

ディテクターのアラートを取得する例:

```
alerts(detector_name='Deployment').publish();
```

カスタムイベントをタイプ別に取得する例:

```
events(eventType='simulated').publish();
```

Wavefront データソースに接続する (VMware Tanzu Observability by Wavefront)

Wavefront (VMware Tanzu Observability by Wavefront) データソースを使用すると、Amazon Managed Grafana ユーザーは、Wavefront から直接収集したデータをクエリおよび視覚化し、他のメトリクス、ログ、トレース、またはその他のデータソースとともに簡単に視覚化できます。この柔軟な単一ペインビューにより、システムのヘルス状態の追跡と問題のデバッグが容易になります。

> **Note**
>
> このデータソースは、Grafana Enterprise 専用です。詳細については、「Enterprise プラグインへのアクセス許可の管理」を参照してください。
> さらに、バージョン 9 以降をサポートするワークスペースでは、このデータソースに適切なプラグインをインストール必要がある可能性があります。詳細については、「プラグインを使用したワークスペースの拡張」を参照してください。

Wavefront とは

Wavefront は、VMware によって開発されたクラウドのモニタリングおよび分析ツールです。Wavefront は、CollectD、StatsD、JMX、Ruby のロガー AWS、またはその他のツールから時系列 (メトリクス) データを送信するクラウドホスト型サービスです。Wavefront を使用すると、ユーザーはこれらの系列で数学的操作の実行、異常を表示するためのグラフのレンダリング、KPI の追跡、アラートの作成が可能です。

Wavefront と Grafana でテクノロジースタックを最大化する

表面的には、Grafana と Wavefront は似ていますが、多くの組織は、オブザーバビリティワークフローの重要な部分として Wavefront と Grafana の両方を使用しています。

データソースを移動せずに視覚化する: Grafana の独自のアーキテクチャは、データを移動して冗長ストレージと取り込みに消費するのではなく、データが置かれている場所に直接クエリを実行します。

さまざまなソースからパネルを作成する 事前に構築されたカスタムダッシュボードを使用すると、さまざまなデータソースのデータを 1 つのウィンドウにまとめることができます。

ユーザーレベルでの変換とコンピューティング: ユーザーはデータを変換し、表示されるデータに対してさまざまな計算を実行できるため、データの準備が少なくて済みます。

パネル内で結合、コンピューティング、視覚化: Prometheus や InfluxDB などの Waveferont やその他のソースからの関連データを表示する混合データソースパネルを作成します。

ドキュメント

機能

- 時系列の視覚化

- テーブルの視覚化
- ヒートマップの視覚化
- 単一統計の視覚化
- ガイド付きクエリエディタ
- 生の WQL クエリエディタ
- イベントデータの注釈
- テンプレート変数
- アドホックフィルター
- [アラート]

設定

Wavefront データソースの設定は比較的簡単です。設定を完了するために必要なフィールドは API URL と Token の 2 つだけです。

- API URL は、wavefront 環境へのアクセスに使用する URL です。例えば、`https://myenvironment.wavefront.com` などです。
- Token は、ユーザーアカウントまたはサービスアカウントから生成する必要があります。

  1. ユーザーアカウントベースのトークンを作成するには、Wavefront 環境にログインし、ページの右上隅にある歯車を選択し、ユーザー名 (例: `me@grafana.com`) を選択し、ユーザーページの上部にある [API アクセス] タブを選択し、既存のキーをコピーするか、[生成] を選択します。
  2. サービスアカウントベースのトークンを作成するには、Wavefront 環境にログインし、ページの右上隅にある歯車を選択し、アカウント管理を選択します。左側のナビゲーションで、[アカウント、グループ、ロール] を選択し、上部の [サービスアカウント] タブを選択し、[新しいアカウントの作成] を選択します。サービスアカウントの名前を入力します。これは、任意のものになります。トークンセクションで提供されているトークンをコピーします。
  3. 最後のステップとして、[アクセス許可] で [アカウント、グループ、ロール]チェックボックスが選択されていることを確認してください。

トークンを取得したら、Token 設定フィールドに追加します。これで準備完了です。

確定された設定ページは次のようになります。

使用方法

クエリエディタの使用

Wavefront クエリエディタには、Query Builder と Raw Query の 2 つのモードがあります。モードを切り替えるには、クエリフォームの右上にあるセレクタを使用します。

Query Builder モードでは、次の 4 つの選択肢が表示されます。

1. どのメトリクスに対してクエリを実行しますか?
2. そのメトリクスに対してどのような集計を実行しますか?
3. そのメトリクスクエリから結果をフィルタリングするにはどうすればよいですか?
4. 結果に追加の関数を適用しますか?

メトリクスセレクタは分類された階層です。カテゴリを選択してから、もう一度選択してサブカテゴリにドリルダウンします。必要なメトリクスに達するまで、このプロセスを繰り返します。

メトリクスを選択すると、使用可能なフィルターとフィルター値が自動的に入力されます。

Raw Query モードでは、[Query] とラベル付けされた 1 つのフィールドが表示されます。これにより、任意の WQL クエリを実行できます。

フィルターの使用

Wavefront プラグインは、各メトリクスに適したフィルターに対して動的にクエリを実行します。

フィルターを追加するには、Wavefront クエリエディタの [Filters] ラベルの横にある [+] を選択し、フィルタリングするフィールドを選択し、フィルタリングする値を選択します。

関数の使用

関数は、メトリクスレスポンスデータを集計、操作、および計算実行するための追加の方法を提供します。使用可能な関数を表示するには、Query Builder の関数ラベルでドロップダウンリストを選択します。選択した関数に基づいて、フィールドによるグループの設定やしきい値の適用など、さらなるアクションを実行できます。ユーザーは複数の関数を連鎖させて、高度な計算やデータ操作を実行できます。

クエリテンプレート変数の追加

1. ダッシュボードの新しい Wavefront テンプレート変数を作成するには、ダッシュボードの右上にある設定 (歯車) を選択します。

2. 左側のパネルで、[変数] を選択します。
3. 変数ページの右上で、[新規] を選択します。
4. 作成するテンプレート変数の [名前] と [Label] を入力します。[名前] は、クエリ内でテンプレート変数を参照するために使用する値です。ラベル は、ダッシュボード選択パネルのテンプレート変数に表示するわかりやすい名前です。
5. タイプフィールドのクエリタイプを選択します (デフォルトで選択する必要があります)。
6. [Query Options] の見出しにあるデータソースドロップダウンリストで [Wavefront] を選択します。
7. クエリ フィールドに入力する内容の詳細については、「テンプレート変数クエリ構造」を参照してください。
8. クエリから返された値をフィルタリングする場合は、[Regex] 入力フィールドに正規表現を入力します。
9. [ソート] ドロップダウンリストでソートタイプを選択して、ソート設定を適用します。
10. 設定を確認したら、[追加] を選択してテンプレート変数を追加し、左側のナビゲーションパネルで [ダッシュボードの保存] を選択して変更を保存します。

テンプレート変数クエリ構造

metric lists: metrics: ts(…)

source lists: sources: ts(…)

source tag lists: sourceTags: ts(…)

matching source tag lists: matchingSourceTags: ts(…)

tag name lists: tagNames: ts(…)

tag value lists: tagValues(<tag>): ts(…)

Notes (メモ)

- 各クエリタイプの末尾にある「s」は任意です
- すべての小文字をサポートしています。tagnames または tagNames を使用できますが、TAGNAMES は使用できません。
- : を囲むスペースの使用は任意です

警告

`Multi-value` および `Include All option` は現在、Wavefront プラグインではサポートされていません。

テンプレート変数の使用

新しいテンプレート変数を追加するステップの完了で、ダッシュボードパネル内のテンプレート変数を使用して動的視覚化を作成する準備が整いました。

1. ダッシュボードの右上隅にある [panel+] アイコンを使用して、新しいダッシュボードパネルを追加します。
2. クエリに使用する集計を選択します。
3. [Filters] ラベルの横にある [+] アイコンを選択し、テンプレート変数に一致するキータイプを選択します。例えば、`host=` はホストフィルターに対応しています。
4. フィルターの値入力フィールドに作成したテンプレート変数の名前を入力します。
5. ダッシュボードを保存します。

これで、テンプレート変数のさまざまな値を循環させ、パネルを動的に更新できるようになりました。

アドホックフィルターの使用

アドホックフィルターを使用するには、2 つのテンプレート変数を作成する必要があります。1 つ目は、メトリクスの選択に使用されるヘルパー変数です。そのメトリクス名にアドホックフィルターを入力できるようになります。もう 1 つは実際のアドホックフィルター変数です。

> **Important**
>
> 必要なヘルパー変数には `metriclink` という名前を付ける必要があります。これは、使用するメトリクスのリストを含むカスタム変数でも、テンプレート変数クエリ構造を使用したクエリベースの変数でもかまいません。アドホックフィルターフィールドに 1 つのメトリクスの値のみを入力する場合は、`metriclink` テンプレート変数を非表示にできます。

`metriclink` 変数を作成したら、「クエリテンプレート変数の追加」で説明されているのと同じステップに従って、アドホックフィルターを追加できるようになります。違いは、タイプとしてアドホックフィルターを選択し、クエリの入力を必要としないことです。

注釈の追加

1. ダッシュボードの新しい Wavefront 注釈を作成するには、ダッシュボードの右上にある設定 (歯車) を選択します。
2. 左側のパネルで、[注釈] を選択します。
3. 注釈ページの右上で、[新規] を選択します。
4. 注釈の名前を入力します (ダッシュボードのトグルの名前として使用されます)。
5. Wavefront のデータソースを選択します。
6. デフォルトでは、注釈には返されるアラートイベントが 100 件に制限されています。これを変更するには、[制限]フィールドを必要な値に設定します。
7. [追加] を選択します。

注釈の使用

注釈をオンにすると、特定の期間に関連するアラートイベントと問題が表示されるようになりました。

視覚化の注釈付きセクションの下部で一時停止すると、アラート名と Wavefront のアラートへの直接リンクを示すポップアップウィンドウが表示されます。

表示名フィールドの使用

このデータソースは、オプションパネルのフィールドタブにある [Display Name] (表示名) フィールドを使用して、名前、ラベル、または値に基づいて凡例キーを短縮または変更します。他のデータソースでは、カスタム`alias`機能を使用して凡例キーを変更しますが、Display Name 関数の方がより一貫性のある方法です。

リファレンス

- WQL (Wavefront クエリ言語)

Grafana バージョン 10 での作業

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana ワークスペースを作成するときに、使用する Grafana のバージョンを選択できます。以下のトピックでは、Grafana バージョン 10 を使用する Grafana ワークスペースの使用について説明します。

トピック

- Grafana バージョン 10 のダッシュボード
- Grafana バージョン 10 のパネルと視覚化
- Grafana バージョン 10 で探索する
- Grafana バージョン 10 の相関関係
- Grafana バージョン 10 のアラート

Grafana バージョン 10 のダッシュボード

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードは、1 つ以上の行に編成および配置された 1 つ以上のパネルのセットです。Grafana にはさまざまなパネルが付属しているため、適切なクエリを簡単に作成し、必要に応じて最適なダッシュボードを作成できるように視覚化をカスタマイズできます。各パネルは、設定された任意のデータソースに接続する からのデータを操作できます。

ダッシュボードスナップショットは静的です。クエリと式をスナップショットから再実行することはできません。そのため、クエリまたは式内の変数を更新しても、ダッシュボードデータは変更されません。

トピック

- ダッシュボードの使用
- ダッシュボードの構築
- ダッシュボードの管理
- プレイリストの管理
- ダッシュボードとパネルの共有
- 変数
- ダッシュボード使用状況の評価
- ダッシュボードのトラブルシューティング
- Grafana バージョン 10 でのダッシュボードの検索

ダッシュボードの使用

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、ダッシュボード機能、ショートカットの概要、およびダッシュボード検索の使用方法について説明します。

機能

ダッシュボードを使用して、データの表示をカスタマイズできます。次の図は、Amazon Managed Grafana ワークスペースのダッシュボードインターフェイスを示しています。



機能	説明
1. ホーム	Grafana インスタンスで設定されたホームページにリダイレクトする Grafana ホームアイコンを選択します。
2。タイトル	ダッシュボードタイトルを選択すると、現在のフォルダに含まれるダッシュボードを検索できます。
3. ダッシュボードの共有	このオプションを使用して、リンクまたはスナップショットで現在のダッシュボードを共有します。ダッシュボード定義を共有モーダルからエクスポートすることもできます。
4. 新しいパネルの追加	このオプションを使用して、パネル、ダッシュボード行、またはライブラリパネルを現在のダッシュボードに追加します。
5. [ダッシュボードの保存]	[保存] アイコンを選択して、ダッシュボードに変更を保存します。

機能	説明
6. ダッシュボードインサイト	ユーザー、アクティビティ、クエリ数に関する情報など、ダッシュボードに関する分析を表示する場合に選択します。詳細については、「ダッシュボード使用状況の評価」を参照してください。
7。ダッシュボード設定	このオプションを使用して、ダッシュボード名、フォルダ、またはタグを変更し、変数と注釈クエリを管理します。ダッシュボードのセットアップ方法の詳細については、「ダッシュボード設定の変更」を参照してください。
8。タイムピッカーのドロップダウン	このオプションを使用して、相対時間範囲オプションを選択し、カスタム絶対時間範囲を設定します。 [タイムゾーン] と [会計年度] の設定は、[時間設定の変更] ボタンをクリックして、時間範囲コントロールから変更できます。 時間設定はダッシュボードごとに保存されます。
9。時間範囲のズームアウト	このオプションを使用して、時間範囲をズームアウトします。時間範囲の制御方法の詳細については、「ダッシュボードの時間範囲の設定」を参照してください。
10。[ダッシュボードの更新]	選択すると、クエリをすぐにトリガーし、ダッシュボードデータを更新します。
11。ダッシュボードの時間間隔の更新	ダッシュボードの自動更新時間間隔を選択します。

機能	説明
12。表示モード	このオプションを使用して、テレビやキオスクなどの大きな画面にダッシュボードを表示します。表示モードでは、ナビゲーションメニューなどの無関係な情報が非表示になります。
13。ダッシュボードパネル	ダッシュボードの主な構成要素はパネルです。新しいパネル、ダッシュボード行、またはライブラリパネルを追加するには、[パネルの追加] を選択します。 • ライブラリパネルは、多くのダッシュボード間で共有できます。 • パネルを移動するには、パネルヘッダーを別の場所にドラッグします。 • パネルのサイズを変更するには、パネルの右下隅を選択してドラッグします。
14。グラフ凡例	シリーズの色、y 軸、およびシリーズの可視性を凡例から直接変更します。
15。ダッシュボード行	ダッシュボード行は、パネルをグループ化するダッシュボード内の論理的なディバイダーです。 • 行を折りたたんだり展開したりして、ダッシュボードの一部を非表示にできます。 • 折りたたまれた行内のパネルはクエリを発行しません。 • 繰り返し行を使用して、テンプレート変数に基づいて行を動的に作成します。繰り返し行の詳細については、「ダッシュボードの作成」を参照してください。

キーボードショートカット

Grafana では多数のキーボードショートカットを利用できます。使用可能なすべてのキーボードショートカットを表示するには、キーボードで？または h を押します。

- Ctrl+S は、現在のダッシュボードを保存します。
- f は、ダッシュボードファインダー/検索を開きます。
- d+k は、キオスクモードを切り替えます (メニューを非表示にします)。
- d+e は、すべての行を展開します。
- d+s は、ダッシュボード設定を開きます。
- Ctrl+K は、コマンドパレットを開きます。
- Esc は、全画面表示または編集モードのときにパネルを終了します。また、ダッシュボード設定からダッシュボードに戻ります。

フォーカスされたパネル

特定のパネルをターゲットとするショートカットを使用するには、ポインターでパネルにカーソルを合わせます。

- e は、パネル編集ビューを切り替えます
- v は、パネル全画面表示を切り替えます
- ps は、パネル共有機能を開きます
- pd は、パネルを複製します
- pr は、パネルを削除します
- pl は、パネル凡例を切り替えます

ダッシュボードの時間範囲の設定

Grafana では、ダッシュボード、パネル、アラートなど、視覚化されるデータの時間範囲を管理する方法がいくつか用意されています。

このセクションでは、サポートされている時間単位と相対範囲、一般的な時間コントロール、ダッシュボード全体の時間設定、パネル固有の時間設定について説明します。

時間単位と相対範囲

Grafana は、次の時間単位に対応しています: s (seconds)、m (minutes)、h (hours)、d (days)、w (weeks)、M (months)、Q (quarters)、y (years)。

マイナス演算子を使用すると、現在の日付と時刻または now を基準にして、時間を戻すことができます。単位の全期間 (日、週、月など) を表示するには、末尾に /<time unit> を追加します。会計期間を表示するには、fQ (fiscal quarter) と fy (fiscal year) の時間単位を使用します。

プラス演算子を使用すると、現在の時間を基準に未来を指定することができます。例えば、この機能を使用して、将来の予測データを確認できます。

次の表に、相対範囲の例を示します。

相対範囲の例	From	To
過去 5 分間	now-5m	now
今日のこれまでのところ	now/d	now
今週	now/w	now/w
今週始めから現在まで	now/w	now
今月	now/M	now/M
今月始めから現在まで	now/M	now
前月	now-1M/M	now-1M/M
今年のこれまでのところ	now/Y	now
今年	now/Y	now/Y
前会計年度	now-1y/fy	now-1y/fy

> **Note**
>
> Grafana アラートは、次の構文をサポートしていません。
>
> - now+n: 将来のタイムスタンプ。

- now-1n/n: n の開始から n の終了まで (これは絶対タイムスタンプであるため)。

一般的な時間範囲コントロール

ダッシュボードとパネルの時間コントロールには、共通のユーザーインターフェイスがあります。一般的な時間範囲コントロールを以下に示します。

- 現在の時間範囲 (別名タイムピッカー)は、そのダッシュボードまたはパネルに現在表示されている時間範囲を示します。フィールドにカーソルを合わせると、範囲内の正確なタイムスタンプとそのソース (ローカルブラウザの時刻など) が表示されます。現在の時間範囲をクリックして変更します。現在の時間は、過去 15 分などの相対時間範囲や、 のような `2020-05-14 00:00:00 ~ 2020-05-15 23:59:59` などの絶対時間範囲で変更できます。
- [相対時間範囲]は、[相対時間範囲] リストから選択できます。上部の入力フィールドを使用してリストをフィルタリングできます。時間範囲の例には、[過去 30 分]、[過去 12 時間]、[過去 7 日間]、[過去 2 年間] 、[昨日]、[昨日の前日]、[先週のこの曜日]、[今日のこれまでのところ]、[今週始めから現在まで]、[今月始めから現在まで]などがあります。
- 絶対時間範囲は、[開始] フィールドと [終了] フィールドに正確な時間値または相対時間値を入力して [時間範囲を適用] をクリックするか、[開始] フィールドまたは [終了] フィールドをクリックしたときに表示されるカレンダーから日付または日付範囲をクリックするという 2 つの方法で設定できます。選択を適用するには、[時間範囲の適用] をクリックします。最近使用した絶対時間範囲のリストから選択することもできます。
- 半相対時間範囲は、絶対時間範囲設定で選択できます。例えば、特定の日付以降のアクティビティを表示するには、開始時刻に絶対時間、終了時刻に相対時間 (now など) を選択できます。

  半相対時間範囲を使用すると、時間が進むにつれて、ダッシュボードは自動的かつ段階的にズームアウトし、より多くの履歴とより少ない詳細を表示します。同時に、高いデータ解像度が低下するにつれて、全期間にわたる過去の傾向がより明確になります。

  > **Note**
  >
  > アラートは、半相対時間範囲をサポートしていません。

- [ズームアウト] アイコンを選択してズームアウトします (またはキーボードショートカットとして Cmd+Z または Ctrl+Z を使用) 。これにより、ビューが拡大され、ダッシュボードまたはパネルの視覚化の時間範囲が大きくなります。
- ズームインするには、視覚化でグラフに表示する時間範囲を選択します。

> **Note**
> ズームインはグラフの視覚化にのみ適用されます。

[ダッシュボードの更新]

[ダッシュボードの更新] アイコンをクリックすると、ダッシュボード上のすべてのクエリが即座に実行され、視覚化が更新されます。更新をトリガーすると、Grafana は保留中のリクエストをすべてキャンセルします。

デフォルトでは、Grafana はダッシュボードを自動的に更新しません。クエリは、パネル設定に従って独自のスケジュールで実行されます。[ダッシュボードの更新] アイコンの横にある下矢印をクリックし、更新間隔を選択することで、ダッシュボードを定期的に更新することができます。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、ダッシュボードの時間範囲を制御できます。

- `from` は、エポックミリ秒で指定された時間範囲の下限、または相対時間を定義します。
- `to` は、エポックミリ秒で指定された時間範囲の上限、または相対時間を定義します。
- `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータはミリ秒で指定する必要があります。たとえば、`?time=1500000000000&time.window=10000` は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードの構築

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana ワークスペースを作成してサインインすると、ダッシュボードを作成し、ニーズに合わせて設定を変更できます。ダッシュボードは、視覚化を含むパネルで構成されます。各パネルには、いずれかの データソースに接続する からデータを取得するためのクエリが関連付けられています。

変数を追加して使用することで、よりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、またはその他の名前をハードコーディングする代わりに、変数を使用できます。

トピック

- ダッシュボードの作成
- ダッシュボードのインポート
- ダッシュボードのエクスポート
- ダッシュボード設定の変更
- ダッシュボード URL 変数
- ライブラリパネルの管理
- ダッシュボードのバージョン履歴の管理
- ダッシュボードリンクの管理
- 視覚化に注釈を付ける
- ダッシュボード JSON モデル
- ダッシュボードのベストプラクティス

ダッシュボードの作成

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードの作成

ダッシュボードとパネルを使用すると、Grafana を使用してデータを視覚的に表示できます。各パネルには、視覚化を表示するために少なくとも 1 つのクエリが必要です。開始する前に、前提条件として次の作業を完了します。

- 適切なアクセス許可を持っていることを確認してください。権限の詳細については、「ユーザー、チーム、およびアクセス許可」を参照してください。
- パネルを追加するダッシュボードを特定します。
- ターゲットデータソースのクエリ言語を理解します。
- クエリを記述するデータソースが追加されていることを確認します。詳細については、「データソースに接続する」を参照してください。

ダッシュボードを作成するには

1. Grafana にサインインし、左側のメニューから [ダッシュボード] を選択します。
2. [新規] を選択し、次に [新しいダッシュボード] を選択します。
3. 空のダッシュボードで、[+ 視覚化の追加] を選択します。これにより、新しい視覚化ダイアログボックスが開きます。
4. データソースの選択。既存のデータソース、Grafana の組み込みデータソースのいずれかをテスト用に選択するか、[新しいデータソースを設定] を選択して新しいデータソースを設定できます (管理者権限を持つユーザーのみが新しいデータソースを設定できます)。

   データソースが選択された [編集パネル] ビューが開きます。必要に応じて、パネルエディタの [クエリ] タブを使用して、後でパネルのデータソースを変更できます。
5. データソースのクエリ言語でクエリを記述または作成します。ダッシュボードの更新アイコンを選択すると、データソースでクエリが実行され、結果が表示されます。
6. [視覚化] リストで、視覚化タイプを選択します。Grafana は、視覚化が適用されたクエリ結果のプレビューを表示します。詳細については、「視覚化のオプション」を参照してください。
7. [パネルオプション] で、パネルのタイトルと説明を入力できます。
8. ほとんどの視覚化では、必要な正確な情報を表示する前にある程度の調整が必要です。パネル設定は、次の方法で調整できます。
   - 値マッピングを設定する
   - 視覚化固有のオプション
   - フィールド値を上書きする
   - しきい値を設定する

- 標準オプションを設定する

9. パネルの設定が完了したら、[保存] を選択してダッシュボードを保存します。

   または、[適用] を選択して、パネルエディタを離れずに変更を表示します。

10. 視覚化を記述する (または変更を記述する) メモを追加し、ページの右上にある [保存] をクリックします。

> **Note**
>
> ダッシュボードを以前のバージョンに戻す必要が生じた際、このメモが役に立ちます。

11. [保存] を選択します。

12. 必要に応じて、ダッシュボードヘッダーで [追加] を選択し、ドロップダウンから [視覚化] を選択して、ダッシュボードにパネルを追加できます。

既存のダッシュボードのコピー

既存のダッシュボードをすばやくコピーして、新しいダッシュボードの作成をすぐに開始できます。

既存のダッシュボードをコピーするには

1. 左側のメニューから [ダッシュボード] を選択します。
2. コピーするダッシュボードを選択して開きます。
3. ダッシュボードの右上にある [設定] (歯車アイコン) を選択します。
4. ダッシュボードの右上隅にある [名前を付けて保存] を選択します。
5. (オプション) 名前、フォルダ、説明、およびコピーしたダッシュボードの元のダッシュボードタグをコピーするかどうかを指定します。
6. [保存] を選択します。

繰り返し行の設定

変数の値に基づいてダッシュボードにパネルまたは行を動的に追加するように Grafana を設定できます。変数は、ダッシュボード内のすべての行にわたってクエリを動的に変更します。パネルの繰り返しの詳細については、「繰り返しパネルの設定」を参照してください。

`Multi-value` [または]`Include all values` が選択された変数セットがある場合は、行を繰り返すこともできます。

開始する前に、クエリに複数値変数が含まれていることを確認し、次のステップを実行します。

繰り返し行を設定するには

1. 左のメニューから [ダッシュボード] を選択し、変更するダッシュボードを選択します。
2. ダッシュボードの上部で、[追加] を選択し、ドロップダウンから [Row] (行) を選択します。

   ダッシュボードが空の場合は、ダッシュボードの中央にある [+ 行の追加] ボタンを代替的に選択できます。
3. 行タイトルにカーソルを合わせ、表示される [設定] (歯車) アイコンを選択します。
4. [パネルオプション] ダイアログボックスで、タイトルを追加し、繰り返し行を追加する変数を選択します。

   > **Note**
   >
   > ダッシュボードユーザーにコンテキストを提供するには、行タイトルに変数を追加します。

5. [更新] を選択します。

繰り返し行とダッシュボードの特別なデータソース

行に、同じダッシュボード内の別のパネルの結果セットを使用するデータソースである特殊なダッシュボードデータソースを使用するパネルが含まれている場合、繰り返し行の対応するパネルは、繰り返し行のパネルではなく、元の行のパネルを参照します。

例えば、ダッシュボードでは:

- Row 1 には Panel 1A と Panel 1B が含まれます。
- Panel 1B は、Dashboard データソースを使用して Panel 1A の結果を使用します。
- 繰り返し Row 2 には、Panel 2A と Panel 2B が含まれます。
- Panel 2B は Panel 2A ではなく Panel 1A を参照します。

パネルを移動するには

1. ダッシュボードを開きます。

2. パネルタイトルを選択し、パネルを新しい場所にドラッグします。パネルは、任意の場所のダッシュボードに配置することができます。

パネルのサイズを変更するには

1. ダッシュボードを開きます。
2. パネルのサイズを調整するには、パネルの右下隅をドラッグします。ダッシュボードパネルは、ニーズに合わせてサイズ調整できます。

ダッシュボードのインポート

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

事前設定されたダッシュボードを Amazon Managed Grafana ワークスペースにインポートできます。

ダッシュボードをインポートするには

1. Amazon Managed Grafana ワークスペースにサインインします。
2. 左側のメニューから [ダッシュボード] を選択します。
3. [新規] を選択し、ドロップダウンメニューで [インポート] を選択します。
4. 次に、インポートするダッシュボード JSON 定義を選択します。JSON をインポートする方法には、次の 3 つの選択肢があります。
   - ダッシュボード JSON を含むファイルをアップロードします。
   - JSON テキストをテキストエリアに直接コピーします。
   - フィールドに Grafana Labs ダッシュボードの URL または ID を貼り付けます。grafana.com ダッシュボード URL の詳細については、次のセクションを参照してください。
   - (オプション) 変更するダッシュボードの詳細を変更します。

- 必要に応じてデータソースを選択します。
- [インポート] を選択します。
- ダッシュボードを保存します。

grafana.com でのダッシュボードの検索

grafana.com の [ダッシュボード] ページには、一般的なサーバーアプリケーションのダッシュボードが用意されています。公式ダッシュボードとコミュニティ構築ダッシュボードのライブラリを参照してインポートし、すぐに起動して実行できるようにします。

> **Note**
>
> grafana.com からダッシュボードをインポートするには、Amazon Managed Grafana ワークスペースがインターネットにアクセスできる必要があります。

ダッシュボードのエクスポート

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana UI または HTTP API を使用してダッシュボードをエクスポートできます。

ダッシュボードのエクスポートアクションは、レイアウト、変数、スタイル、データソース、クエリなど、必要なすべてを含む Grafana JSON ファイルを作成し、後でダッシュボードをインポートできるようにします。

ダッシュボードをポータブルにする

他のユーザーが使用するためにダッシュボードをエクスポートする場合は、メトリクスプレフィックス (定数変数を使用) やサーバー名などのテンプレート変数を追加できます。

タイプ Constant のテンプレート変数はダッシュボードで自動的に非表示になり、ダッシュボードのインポート時に必要な入力として追加されます。

ダッシュボードをエクスポートするには

1. エクスポートするダッシュボードを開きます。
2. 共有アイコンを選択します。
3. [エクスポート] をクリックします。
4. [ファイルに保存] を選択します。

> **Note**
>
> Grafana は JSON ファイルをローカルマシンにダウンロードします。

ダッシュボード設定の変更

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボード設定ページでは、次の操作を実行できます。

- 時間設定を含む一般的なダッシュボードプロパティを編集する。
- 注釈クエリを追加する。
- ダッシュボード変数を追加する。
- リンクを追加する。
- ダッシュボードの JSON モデルを表示する

ダッシュボード設定ページにアクセスするには

1. ダッシュボードを編集モードで開きます。
2. ページの上部にある [ダッシュボード設定] (歯車アイコン) をクリックします。

ダッシュボードの時間設定の変更

ダッシュボードのタイムゾーン、ローカルブラウザの時刻を変更するときにダッシュボードの時刻設定を調整し、自動更新の時間間隔を指定します。

ダッシュボードの時間設定を変更するには

1. [ダッシュボード] 設定ページで、[全般] を選択します。
2. [時間オプション] セクションに移動します。
3. 以下の説明に従って時間設定を指定します。
4. • [タイムゾーン] – モニタリングするサービスまたはシステムのローカルタイムゾーンを指定します。これは、複数のタイムゾーンで動作するシステムまたはサービスを監視する場合に役立ちます。
   - [デフォルト] – Grafana は、ユーザープロファイル、チーム、または組織にデフォルトで選択されたタイムゾーンを使用します。ユーザープロファイル、ユーザーが所属するチーム、または組織にタイムゾーンが指定されていない場合、Grafana はローカルブラウザの時間を使用します。
   - ローカルブラウザ時間 – ユーザーブラウザの表示用に設定されたタイムゾーンが使用されます。これは通常、コンピュータに設定されているタイムゾーンと同じです。
   - UTC を含む標準の ISO 8601 タイムゾーン を使用します。
   - [自動更新]– 相対時間に表示されるオプションと自動更新オプションをカスタマイズできます。エントリはカンマ区切りで、任意の有効な時間単位を受け入れます。
   - Now delay (遅延時間の指定) – 遅延時間を入力して now 時間を上書きします。このオプションを使用して、null 値を避けるために、データ集約の既知の遅延に対応します。
   - [タイムピッカーを非表示] – このオプションを選択すると、ダッシュボードでタイムピッカーを非表示にできます。

> **Note**
>
> 時間コントロールの使用には、データに時間列が含まれている必要があります。時間列を含める方法の詳細については、特定の「データソース」のドキュメントを参照してください。

[注釈クエリの追加]

注釈クエリは、イベントをクエリするクエリです。これらのイベントは、カーソルを合わせるとイベント情報を表示できる小さなアイコンとともに、ダッシュボード全体のグラフで垂直線として視覚化できます。

注釈クエリを追加するには

1. [ダッシュボード設定] ページで、[注釈] を選択します。
2. [注釈クエリの追加] を選択します。
3. 名前を入力し、データソースを選択します。
4. フォームの残りの部分に入力して、クエリと注釈を作成します。

クエリエディタの UI は、選択したデータソースに基づいて変わります。クエリの作成方法の詳細については、「データソース」ドキュメントを参照してください。または、プラグインカタログによるプラグインの検索 からインストールするデータソースプラグインについては、「Grafana Labs ウェブサイト のドキュメント」を参照してください。

変数の追加

変数を使用すると、よりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウンリストとして表示されます。これらのドロップダウンを使用すると、ダッシュボード内に表示されるデータを簡単に変更できます。

変数の詳細については、「変数」を参照してください。

変数を追加するには

1. [ダッシュボード設定] ページで、左側のセクションメニューの [変数]、[変数を追加] ボタンを続けてクリックします。
2. [General] (全般) セクションで、変数の名前を追加します。これは、後でクエリで使用する名前です。

3. 変数タイプを選択します。

> **Note**
>
> 選択した変数タイプは、ページに入力するフィールドに影響します。

4. 変数を定義し、[更新] をクリックします。

リンクの追加

ダッシュボードリンクを使用すると、他のダッシュボードやウェブサイトへのリンクをダッシュボードヘッダーの直下に配置できます。リンクを使用すると、他の関連するダッシュボードやコンテンツに簡単に移動できます。

リンクを追加するには

1. [ダッシュボード設定] ページで、左側のセクションメニューから [リンク] を選択し、[リンクの追加] ボタンを選択します。
2. タイトルを入力し、[タイプ] フィールドに [ダッシュボード] または [リンク] を選択します。
3. ダッシュボードリンクを追加するには、オプションのタグを追加し、ダッシュボードリンクのオプションを選択して、[適用] をクリックします。

> **Note**
>
> タグを使用すると、すべてのダッシュボードに特定のタグを持つ動的なドロップダウンが作成されます。

4. ウェブリンクを追加するには、ユーザーがリンクの上にカーソルを置いたときに表示される URL とツールヒントテキストを追加し、リンクの横に表示されるアイコンを選択して、ダッシュボードのリンクオプションを選択します。

ダッシュボードの JSON モデルを表示する

Grafana のダッシュボードは、ダッシュボードのメタデータを保存する JSON オブジェクトによって表されます。ダッシュボードメタデータには、ダッシュボードプロパティ、パネルからのメタデータ、テンプレート変数、パネルクエリなどが含まれます。JSON メタデータはダッシュボードを定義します。

ダッシュボード JSON モデルを表示するには、[ダッシュボード設定] ページで [JSON] をクリックします。

JSON フィールドの詳細については、「JSON フィールド」を参照してください。

ダッシュボード URL 変数

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は、ダッシュボード URL でクエリパラメータとして渡された変数値を適用できます。詳細については、「ダッシュボードリンクの管理」と「テンプレートおよび変数の管理」を参照してください。

変数をクエリパラメータとして渡す

Grafana は、`var-` のプレフィックスが付加されたクエリ文字列パラメータを、特定のダッシュボードの変数として解釈します。

例えば、この URL では、次のようになります。

```
https://${your-domain}/path/to/your/dashboard?var-example=value
```

クエリパラメータ `var-example=value` は、ダッシュボード変数の例を表し、値は `value` です。

変数に複数の値を渡す

複数の値を渡すには、値ごとに変数パラメータを 1 回繰り返します。

```
https://${your-domain}/path/to/your/dashboard?var-example=value1&var-example=value2
```

Grafana は、 `var-example=value1&var-example=value2` を `value1` と `value2` の 2 つの値を持つダッシュボード変数の例として解釈します。

ダッシュボードリンクへの変数の追加

Grafana は、ダッシュボードの設定から変数を生成するときに、ダッシュボードリンクに変数を追加できます。変数を追加する方法と手順の詳細については、「ダッシュボードリンクの管理」を参照してください。

アドホックフィルターを渡す

アドホックフィルターは、指定されたデータソースを使用するすべてのメトリクスクエリにキーまたは値フィルターを適用します。詳細については、「アドホックフィルター」を参照してください。

アドホックフィルターをクエリパラメータとして渡すには、変数構文を使用してアドホックフィルター変数を渡し、キー、演算子を値、値をパイプ区切りリストとして指定します。

例えば、この URL では、次のようになります。

```
https://${your-domain}/path/to/your/dashboard?var-adhoc=example_key|=|
example_value
```

クエリパラメータ `var-adhoc=key|=|value` は、 `example_key` キー、 `=` 演算子、および `example_value` 値を使用して、アドホックダッシュボード変数として設定されたアドホックフィルターを適用します。

> **Note**
>
> アドホックフィルターを使用して URL を共有する場合は、必ず URL をエンコードしてください。上記の例では、パイプ (|) を %7Cに、等価演算子 (=) を %3D に置き換えます。

URL の使用による時間範囲の制御

ダッシュボードの時間範囲を設定するには、`from`、`to`、`time`、および `time.window` クエリパラメータを使用します。これらは変数ではないため、`var-` プレフィックスは必要ありません。

ライブラリパネルの管理

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ライブラリパネルは、任意のダッシュボードで使用できる再利用可能なパネルです。ライブラリパネルを変更すると、その変更はパネルが使用されているすべてのインスタンスに反映されます。ライブラリパネルは、複数のダッシュボード間でパネルの再利用を合理化します。

ライブラリパネルは、保存されたダッシュボードとともにフォルダに保存できます。

ライブラリパネルの作成

ライブラリパネルを作成すると、ソースダッシュボードのパネルもライブラリパネルに変換されます。パネルが変換されたら、元のダッシュボードを保存する必要があります。

ライブラリパネルの作成方法

1. 編集モードでライブラリパネルに変換するパネルを開きます。
2. パネル表示オプションで、下矢印オプションをクリックして、視覚化の変更を開始します。
3. [ライブラリパネル]、[+ ライブラリパネルの作成] を続けて選択します。これにより、作成ダイアログが開きます。
4. [ライブラリパネル名] に、パネルに使用する名前を入力します。
5. [フォルダに保存] で、ライブラリパネルを保存するフォルダを選択します。
6. [ライブラリパネルの作成] を選択して、ライブラリに変更を保存します。
7. ダッシュボードを保存します。

ライブラリパネルの作成後、そのパネルは、パネルが表示される任意のダッシュボードを使用して変更することができます。変更を保存すると、ライブラリパネルのすべてのインスタンスにこれらの変更が反映されます。

また、[(詳細...])、[ライブラリパネルの作成] を続けて選択して、任意のパネルの編集メニューからライブラリパネルを直接作成することもできます。

ダッシュボードへのライブラリパネルの追加

他のダッシュボードユーザーに視覚化を提供する場合は、Grafana ライブラリパネルをダッシュボードに追加します。

ダッシュボードへのライブラリパネルの追加方法

1. 左側のメニューで [ダッシュボード] を選択します。
2. [新規] を選択し、ドロップダウンから [新しいダッシュボード] を選択します。
3. 空のダッシュボードで、[+ ライブラリパネルをインポート] を選択します。ライブラリパネルのリストが表示されます。
4. リストを検索またはフィルタリングして、追加するパネルを見つけます。
5. パネルをクリックしてダッシュボードに追加します。

ライブラリパネルのリンク解除

ライブラリパネルに変更を加える必要があるときにライブラリパネルの他のインスタンスに影響を与えたくない場合は、ライブラリパネルのリンクを解除します。

ライブラリパネルのリンクを解除するには

1. 左側のメニューで [ダッシュボード] を選択します。
2. [ライブラリパネルl] を選択します。
3. さまざまなダッシュボードで使用されているライブラリパネルを選択します。
4. リンクを解除するパネルを選択します。
5. パネルのタイトル (またはパネル上の任意の場所にカーソルを合わせる) を選択すると、パネルの右上隅にアクションメニューが表示されます。
6. [Edit] (編集) を選択します。パネルは編集モードで開きます。
7. ページの右上隅の [リンク解除] を選択します。
8. [はい、リンク解除する] を選択します。

ライブラリパネルのリストの表示

使用可能なライブラリパネルのリストを表示し、ライブラリパネルを検索できます。

ライブラリパネルのリストを表示するには

1. 左側のメニューで [ダッシュボード] を選択します。
2. [ライブラリパネルl] を選択します。以前に定義したライブラリパネルのリストが表示されます。

3. 名前がわかっている場合は、特定のライブラリパネルを検索します。パネルをフォルダまたはタイプでフィルタリングすることもできます。

ライブラリパネルの削除

不要になったライブラリパネルを削除します。

ライブラリパネルの削除方法

1. 左側のメニューで [ダッシュボード] を選択します。
2. [ライブラリパネル] を選択します。
3. 削除するパネルのライブラリパネル名の横にある [削除] アイコンを選択します。

ダッシュボードのバージョン履歴の管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードのバージョンを保存するたびに、以前のバージョンのダッシュボードが失われないように、そのバージョンのコピーが保存されます。これらのバージョンのリストは、ダッシュボード設定に進み、左側のメニューで [Versions] (バージョン) を選択することで確認できます。

> **Note**
> ダッシュボードの最近のバージョンは 20 個まで保存されます。

ダッシュボードのバージョン履歴機能では、以前に保存したダッシュボードバージョンを比較して復元できます。

2 つのダッシュボードバージョンの比較

2 つのダッシュボードバージョンを比較するには、比較するリストから 2 つのバージョンを選択します。[Compare versions] (バージョンの比較) をクリックして、2 つのバージョン間の差分を表示します。これにより、バージョン差分ビューが表示されます。デフォルトでは、変更の概要がテキストで表示されます。

ダッシュボードを表す raw JSON の差分を表示するには、下部の [JSON 差分の表示] ボタンをクリックすることで表示できます。

以前に保存したダッシュボードバージョンへの復元

以前に保存したダッシュボードバージョンに復元する必要がある場合は、ダッシュボードバージョンリストの行の右側にある [Restore] (復元) ボタンを選択するか、差分ビューで [Restore to version <x>] (バージョン <x> に復元) ボタンを選択します。これらのいずれかを選択すると、復元を確認するメッセージが表示されます。

以前のバージョンに復元すると、バージョン番号が異なるだけで、以前のバージョンとまったく同じデータを含む新しいバージョンが作成されます。これは、新しいダッシュボードバージョンの行の [メモ列] に表示されます。これにより、以前のダッシュボードバージョンが変更の影響を受けなくなります。

ダッシュボードリンクの管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

リンクを使用すると、よく使用されるダッシュボード間を移動、または他のユーザーを視覚化に接続できます。リンクでは、他のダッシュボード、パネル、さらには外部ウェブサイトへのショートカットを作成できます。

Grafana では、ダッシュボードリンク、パネルリンク、データリンクをサポートしています。ダッシュボードリンクはダッシュボードの上部に表示されます。パネルリンクにアクセスするためには、パネルの左上隅にあるアイコンを選択します。

使用するリンクの選択

最初にダッシュボード間の現在の移動方法を確認します。頻繁に一連のダッシュボード間でジャンプし、各ダッシュボードで同じコンテキストを見つけるのに苦労している場合、ワークフローの最適化にリンクが役立ちます。

次のステップでは、ご使用のワークフローに適したリンクタイプを特定します。Grafana のすべてのリンクタイプは他のダッシュボードや外部ウェブサイトへのショートカットの作成に使用されますが、それらはさまざまなコンテキストで動作します。

- リンクがダッシュボード内のほとんどのパネルに関連する場合は、ダッシュボードリンクを使用します。
- 特定のパネルにドリルダウンする場合は、パネルリンクを使用します。
- 外部サイトへのリンクには、ダッシュボードリンクまたはパネルリンクのいずれかを使用できます。
- 特定のシリーズ、または 1 つの測定までドリルダウンするには、データリンクを使用します。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、パネルまたはダッシュボードの時間範囲を制御できます。

- `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。
- `to` は、エポックミリ秒で指定された時間範囲の上限を定義します。
- `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータをミリ秒で指定する必要があります。たとえば、`?time=1500000000000&time.window=10000` は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードリンク

ダッシュボードリンクを作成するときに、時間範囲と現在のテンプレート変数を含めることで、別のダッシュボードの同じコンテキストに直接ジャンプできます。これにより、リンクの送信先が適切なデータを表示しているかどうかを心配する必要がなくなります。他のタイプのリンクについては、「データリンク変数」を参照してください。

ダッシュボードリンクは、現在のダッシュボード名で GitHub の問題を送信するなど、外部システムへのショートカットとしても使用できます。

ダッシュボードリンクを追加すると、ダッシュボードの右上隅に表示されます。

ダッシュボードへのリンクの追加

現在のダッシュボードの上部にある他のダッシュボードへのリンクを追加します。

ダッシュボードにリンクを追加するには

1. リンクするダッシュボードを表示しながら、画面上部の歯車をクリックして [ダッシュボード設定] を開きます。
2. [リンク] を選択し、[ダッシュボードリンク] または [新規] を追加します。
3. [タイプ] で、[ダッシュボード] を選択します。
4. 以下からリンクオプションを選択します。

   - [タグあり] – タグを入力して、リンクされたダッシュボードを、入力したタグを持つダッシュボードのみに制限します。それ以外の場合、Grafana には他のすべてのダッシュボードへのリンクが含まれます。
   - [As dropdown] (ドロップダウンとして) - 多くのダッシュボードにリンクしている場合、デフォルトで、Grafana はダッシュボードの上部にそれらをすべて並べて表示します。このオプションを選択し、オプションのタイトルを追加すると、ドロップダウンにリンクが表示されます。
   - [時間範囲] – このオプションを選択すると、ダッシュボードの時間範囲がリンクに含まれます。ユーザーがリンクをクリックすると、リンクされたダッシュボードが開き、指定された時間範囲が既に設定されています。
   - [変数値] – このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクをクリックすると、リンクされたダッシュボード内の一致するテンプレートがリンクの値に設定されます。詳細については、「ダッシュボード URL 変数」を参照してください。
   - [新しいタブで開く] – ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。
5. [追加] をクリックします。

ダッシュボードへの URL リンクの追加

現在のダッシュボードの上部にある URL へのリンクを追加します。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。

ダッシュボードに URL リンクを追加するには

1. リンクするダッシュボードを表示しながら、画面上部の歯車を選択して [ダッシュボード設定] (ダッシュボード設定) を開きます。
2. [リンク] を選択し、[ダッシュボードリンク] または [新規] を追加します。
3. [タイプ] で [リンク] を選択します。
4. 以下からリンクオプションを選択します。

   - URL – リンク先の URL を入力します。ターゲットによっては、フィールド値を含めることもできます。
   - [タイトル] — リンクを表示するタイトルを入力します。
   - [ツールヒント] – リンクを表示するツールヒントを入力します。
   - Icon (アイコン) – リンクとともに表示するアイコンを選択します。
   - [時間範囲] – このオプションを選択すると、ダッシュボードの時間範囲がリンクに含まれます。ユーザーがリンクをクリックすると、リンクされたダッシュボードが開き、指定された時間範囲が設定されます。
     - `from` - は、エポックミリ秒で指定した、時間範囲の下限を定義します。
     - `to` – は、エポックミリ秒で指定された時間範囲の上限を定義します。
     - `time` および `time.window` — `time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータをミリ秒で指定する必要があります。たとえば、`?time=1500000000000&time.window=10000` は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。
   - [変数値] – このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクをクリックすると、リンクされたダッシュボード内の一致するテンプレートがリンクの値に設定されます。

     変数の形式は次のとおりです。

     ```
     https://${you-domain}/path/to/your/dashboard?var-${template-varable1}=value1&var-{template-variable2}=value2
     ```

   - [新しいタブで開く] – ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。
5. [追加] を選択します。

ダッシュボードリンクの更新

既存のダッシュボードリンクを変更または更新するには、次のステップに従います。

ダッシュボードリンクを更新するには

1. [ダッシュボード設定], の[リンク] タブで、編集する既存のリンクを選択します。
2. 設定を変更し、[更新] を選択します。

ダッシュボードリンクの複製

既存のダッシュボードリンクを複製するには、複製する既存のリンクの横にある複製アイコンを選択します。

ダッシュボードリンクの削除

既存のダッシュボードリンクを削除するには、削除するリンクの横にあるゴミ箱アイコンを選択します。

パネルリンク

各パネルには、パネルの左上隅に表示する一連の独自のリンクを含めることができます。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。

使用可能なパネルリンクを表示するには、パネルタイトルの右側にあるアイコンを選択します。

- パネルリンクの追加: 各パネルには、パネルの左上隅に表示される独自のリンクセットを指定できます。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。パネルの左上隅にあるアイコンをクリックすると、使用可能なパネルリンクが表示されます。

  1. リンクを追加するパネルにカーソルを合わせます。
  2. メニューを選択し、[編集] を選択するか、キーボードショートカット e を使用します。
  3. [パネルオプション] セクションを展開し、[パネルリンク] までスクロールします。
  4. [リンクの追加] を選択します。
  5. [タイトル] を入力します。これは、UI に表示されるリンクの判読可能なラベルです。
  6. リンク先の URL を入力します。Ctrl+Space (または Cmd+Space) を押して URL フィールドを選択して、使用可能な変数を表示します。パネルリンクにテンプレート変数を追加するこ

とで、関連する変数がすでに設定されている適切なコンテキストをリンクからユーザーに送信します。

時間変数も使用できます。

- `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。
- `to` は、エポックミリ秒で指定した、時間範囲の上限を定義します。
- `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータはミリ秒で指定する必要があります。たとえば、`?time=1500000000000&time.window=10000` は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

- パネルリンクの更新

1. パネルを選択する (またはパネルの上にカーソルを合わせる) と、パネルの右上にアクションメニューが表示されます。
2. メニューから、[編集] を選択します。

   キーボードショートカット e でも切り替えられます。
3. [パネルオプション] セクションを展開し、[パネルリンク] までスクロールします。
4. 変更するリンクを見つけ、その横の [編集] (鉛筆) アイコンを選択します。
5. 必要に応じて変更します。
6. [保存] を選択して変更を保存し、ウィンドウを閉じます。
7. 右上で [保存] を選択して、ダッシュボードに変更を保存します。

- パネルリンクの削除

1. パネルを選択する (またはパネルの上にカーソルを合わせる) と、パネルの右上にアクションメニューが表示されます。
2. メニューから、[編集] を選択します。

   キーボードショートカット e でも切り替えられます。
3. [パネルオプション] セクションを展開し、[パネルリンク] までスクロールします。
4. 削除するリンクを見つけ、その横の X アイコンを選択します。
5. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

視覚化に注釈を付ける

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

注釈は、リッチイベントで視覚化上のポイントをマークする方法を提供します。これらは、すべてのグラフパネル上に垂直線とアイコンとして視覚化されます。注釈にカーソルを合わせると、イベントの説明とイベントタグを取得できます。テキストフィールドには、より詳細な他のシステムへのリンクを含めることができます。

視覚化には 3 つの方法で注釈を付けることができます。

- 組み込みの注釈クエリを使用して、パネルで直接注釈を付ける。
- Grafana HTTP API の使用。
- ダッシュボード設定での注釈クエリの設定。

最初の 2 つのケースでは、新しい注釈を作成し、最後のケースでは、データソースから既存の注釈をクエリします。組み込みの注釈クエリもこれをサポートしています。

このセクションでは、1 番目と 3 番目のオプションについて説明します。Grafana HTTP API の使用については、「注釈 API」を参照してください。

注釈は、次の視覚化タイプでサポートされています。

- 時系列
- 状態タイムライン
- ローソク足チャート

パネルで注釈を作成する

Grafana には、すべてのダッシュボードに存在する組み込みの注釈クエリを使用して、パネルから直接注釈イベントを追加する機能があります。この方法で作成した注釈は Grafana に保存されます。

パネルに直接注釈を追加するには：

- ダッシュボードは既に保存されている必要があります。
- ビルトインクエリクエリを有効にする必要があります。

注釈を追加するには

1. ダッシュボードで、注釈を追加するパネルを選択します。コンテキストメニューが表示されます。
2. コンテキストメニューで、[注釈の追加] を選択します。
3. （オプション) 注釈の説明とタグを追加します。
4. [保存] を選択します。

または、パネルを選択しながら Ctrl (ないしは Cmd) を押すと、[注釈の追加] のポップアップが表示されます。

リージョン注釈

視覚化でリージョンまたは期間をカバーする注釈を作成することもできます。

リージョン注釈を追加するには

1. ダッシュボードで Ctrl (または Cmd) を押して、パネルの領域を選択します。
2. 注釈の説明とタグを追加します (オプション）。
3. [保存] をクリックします。

注釈を編集するには

1. ダッシュボードで、パネルの注釈インジケータにカーソルを合わせます。
2. 注釈ツールヒントで [編集] (鉛筆) アイコンを選択します。
3. 説明やタグを変更します。
4. [保存] を選択します。

注釈を削除するには

1. ダッシュボードで、パネルの注釈インジケータにカーソルを合わせます。

2. 注釈ツールヒントで [削除] (ゴミ箱) アイコンを選択します。

ダッシュボード設定を使用して注釈を取得する

ダッシュボード設定の [注釈] で、新しいクエリを追加して、組み込みのデータ注釈データソースを含む任意のデータソースを使用して注釈を取得できます。注釈クエリは、ダッシュボード全体のグラフでイベントマーカーとして視覚化できるイベントを返します。

新しい注釈クエリを追加するには

1. ダッシュボードヘッダーの [設定] (歯車) アイコンを選択して、設定メニューを開きます。
2. [注釈] を選択します。
3. [注釈クエリの追加] をクリックします。
4. クエリの名前を入力します。

   この名前は、このクエリから注釈イベントを表示できるようにするトグル (チェックボックス) に付けられます。
5. 注釈のデータソースを選択します。

   また、[高度なデータソースピッカーを開く] を選択して、新しいデータソースの追加 (管理者のみ使用可能) など、その他のオプションを表示することもできます。
6. 注釈クエリをすぐに使用しない場合は、[有効] チェックボックスをオフにします。
7. ダッシュボードに注釈クエリの切り替えを表示しない場合は、[非表示] チェックボックスを選択します。
8. イベントマーカーの色を選択します。
9. [表示対象] ドロップダウンで、次のいずれかのオプションを選択します。

   - [すべてのパネル] – 注釈は、注釈をサポートするすべてのパネルに表示されます。
   - [選択したパネル] – 注釈は、選択したすべてのパネルに表示されます。
   - [選択した対象を除くすべてのパネル] – 注釈は、選択したパネルを除くすべてのパネルに表示されます。
10. クエリを設定します。

    注釈クエリオプションは、データソースごとに異なります。特定のデータソースの注釈の詳細については、「データソースに接続する」を参照してください。

ビルトインクエリクエリ

注釈を追加すると、それらは引き続き表示されます。これは、すべてのダッシュボードに存在する組み込みの注釈クエリが原因です。この注釈クエリは、Grafana に保存されている現在のダッシュボードから発生するすべての注釈イベントを取得し、それらが作成されたパネルに表示されます。これには、アラート状態履歴の注釈が含まれます。

デフォルトでは、組み込みの注釈クエリは Grafana の特殊なデータソースを使用し、手動注釈はこのデータソースでのみサポートされます。組み込みの注釈クエリで別のデータソースを使用できますが、自動注釈を作成できるのは、そのデータソースのクエリエディタのみです。

ダッシュボードに直接注釈を追加するには、このクエリを有効にする必要があります。

ビルトインクエリクエリが有効になっていることを確認するには

1. ダッシュボードヘッダーのダッシュボードの [設定] (歯車) アイコンを選択して、ダッシュボード設定メニューを開きます。
2. [注釈] を選択します。
3. [注釈とアラート (組み込み)] クエリを検索します。

   クエリ名の前に [無効] と表示されている場合は、クエリ名を選択してクエリを開き、設定を更新する必要があります。

注釈の取得と描画を停止するには

1. ダッシュボードヘッダーのダッシュボードの [設定] (歯車) アイコンを選択して、ダッシュボード設定メニューを開きます。
2. [注釈] を選択します。
3. [注釈とアラート (組み込み)] クエリを検索します。
4. [有効] トグルを選択して、オフにします。

[名前を付けて保存] 機能を使用してダッシュボードをコピーすると、新しいダッシュボード ID が取得されるため、ソースダッシュボードで作成された注釈はコピーに表示されなくなります。新しい [注釈クエリ] を追加し、タグでフィルタリングしても、注釈を表示できます。ただし、これは、ソースダッシュボードの注釈にフィルタリングするタグがある場合にのみ機能します。

タグによるクエリのフィルタリング

[~ でフィルタリング] を Tags に設定することで、Grafana データソースを使用して組み込みの注釈クエリから注釈を取得する新しいクエリを作成できます。

例えば、注釈クエリ名 outages を作成し、タグ outage を指定します。このクエリは、 outage タグを含むすべての注釈 (ダッシュボードまたは API 経由) を表示します。注釈クエリで複数のタグが定義されている場合、Grafana はすべてのタグに一致する注釈のみを表示します。動作を変更するには、[任意の一致] を有効にします。Grafana は、指定したタグのいずれかを含む注釈を表示します。

タグクエリでテンプレート変数を使用することもできます。つまり、さまざまなサービスの統計を表示するダッシュボードと、表示するサービスを指定するテンプレート変数がある場合は、注釈クエリで同じテンプレート変数を使用して、それらのサービスの注釈のみを表示できます。

ダッシュボード JSON モデル

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana のダッシュボードは、ダッシュボードのメタデータを保存する JSON オブジェクトによって表されます。ダッシュボードメタデータには、ダッシュボードプロパティ、パネルからのメタデータ、テンプレート変数、パネルクエリが含まれます。

ダッシュボードの JSON を表示する方法

1. ダッシュボードに移動します。
2. 上部のナビゲーションメニューで、[ダッシュボード設定] (歯車) アイコンを選択します。
3. [JSON モデル]を選択します。

JSON フィールド

ユーザーが新しいダッシュボードを作成すると、新しいダッシュボード JSON オブジェクトが次のフィールドで初期化されます。

> **Note**
>
> 次の JSON では、id は null として表示されます。これは、ダッシュボードが保存されるまで割り当てられるデフォルト値です。ダッシュボードを保存すると、整数値が id フィールドに割り当てられます。

```
{
  "id": null,
  "uid": "cLV5GDCkz",
  "title": "New dashboard",
  "tags": [],
  "timezone": "browser",
  "editable": true,
  "graphTooltip": 1,
  "panels": [],
  "time": {
    "from": "now-6h",
    "to": "now"
  },
  "timepicker": {
    "time_options": [],
    "refresh_intervals": []
  },
  "templating": {
    "list": []
  },
  "annotations": {
    "list": []
  },
  "refresh": "5s",
  "schemaVersion": 17,
  "version": 0,
  "links": []
}
```

以下は、ダッシュボード JSON の各フィールドについて説明します。

名前	使用方法
id	ダッシュボードの一意の数値識別子 (db によって生成)
uid	誰でも生成できる一意のダッシュボード識別子。 文字列 (8～40)
title	ダッシュボードの現在のタイトル
tags	ダッシュボードに関連付けられたタグ、文字列の配列
style	dark や light などのダッシュボードのテーマ
timezone	utc や browser などのダッシュボードのタイムゾーン
editable	ダッシュボードを編集可能かどうか
graphTooltip	共有十字線またはツールチップなしの場合は 0 (デフォルト)、共有十字線の場合は 1、共有十字線と共有ツールチップの場合は 2
time	last 6 hours や last 7 days などのダッシュボードの時間範囲
timepicker	timepicker メタデータ。詳細については「timepicker」を参照してください。
templating	メタデータのテンプレート作成。詳細については「テンプレート作成セクション」を参照してください。
annotations	注釈メタデータ。追加方法については「注釈」を参照してください。
refresh	自動更新間隔

名前	使用方法
schemaVersion	JSON スキーマのバージョン (整数) 。Grafana 更新がこのスキーマに変更を加えるたびに増加します。
version	ダッシュボードのバージョン (整数) 。ダッシュボードが更新されるたびに増加します。
panels	パネル配列 (詳細については、次のセクションを参照してください)

パネル

パネルはダッシュボードの構成要素であり、データソースクエリ、グラフのタイプ、エイリアスなどで構成されます。パネル JSON は、JSON オブジェクトの配列で構成され、それぞれが異なるパネルを表します。ほとんどのフィールドはすべてのパネルで共通ですが、一部のフィールドはパネルタイプによって異なります。以下は、テキストパネルのパネル JSON の例です。

```
"panels": [
  {
    "type": "text",
    "title": "Panel Title",
    "gridPos": {
      "x": 0,
      "y": 0,
      "w": 12,
      "h": 9
    },
    "id": 4,
    "mode": "markdown",
    "content": "# title"
  }
```

パネルのサイズと位置

gridPos プロパティは、グリッド座標のパネルサイズと位置を記述します。

- w – 1 ~ 24 (ダッシュボードの幅は 24 列に分割されます)
- h – グリッド高さの単位では、それぞれ 30 ピクセルを表します。

- x – w と同じ単位の x 位置。
- y – h と同じ単位の y 位置。

グリッドには負の重力があり、パネルの上に空のスペースがある場合にパネルを上に移動します。

Timepicker

```
"timepicker": {
    "collapse": false,
    "enable": true,
    "notice": false,
    "now": true,
    "refresh_intervals": [
      "5s",
      "10s",
      "30s",
      "1m",
      "5m",
      "15m",
      "30m",
      "1h",
      "2h",
      "1d"
    ],
    "status": "Stable",
    "type": "timepicker"
  }
```

[テンプレート作成]

templating フィールドには、保存された値を含むテンプレート変数の配列と、他のメタデータが含まれています。

```
"templating": {
    "enable": true,
    "list": [
       {
        "allFormat": "wildcard",
        "current":  {
          "tags": [],
          "text": "prod",
          "value": "prod"
```

```
        },
        "datasource": null,
        "includeAll": true,
        "name": "env",
        "options": [
           {
            "selected": false,
            "text": "All",
            "value": "*"
          },
           {
            "selected": false,
            "text": "stage",
            "value": "stage"
          },
           {
            "selected": false,
            "text": "test",
            "value": "test"
          }
        ],
        "query": "tag_values(cpu.utilization.average,env)",
        "refresh": false,
        "type": "query"
      },
       {
        "allFormat": "wildcard",
        "current":  {
          "text": "apache",
          "value": "apache"
        },
        "datasource": null,
        "includeAll": false,
        "multi": false,
        "multiFormat": "glob",
        "name": "app",
        "options": [
           {
            "selected": true,
            "text": "tomcat",
            "value": "tomcat"
          },
           {
            "selected": false,
```

```
            "text": "cassandra",
            "value": "cassandra"
          }
        ],
        "query": "tag_values(cpu.utilization.average,app)",
        "refresh": false,
        "regex": "",
        "type": "query"
      }
    ]
  }
```

以下の表では、 テーブルのフィールドについて説明しています。

名前	使用方法
enable	テンプレート作成機能が有効かどうか
list	それぞれが 1 つのテンプレート変数を表すオブジェクトの配列
allFormat	wildcard、glob、regex、pipe など、データソースからすべての値を取得するときに使用する形式。
current	ダッシュボードに現在選択されている可変テキスト/値を表示します
datasource	変数のデータソースを表示します
includeAll	すべての値オプションが使用可能かどうか
multi	変数値リストから複数の値を選択できるかどうか
multiFormat	データソースから時系列を取得する際に使用する形式
name	変数の名前

名前	使用方法
options	ダッシュボードで選択できる可変テキスト/値のペアの配列
query	変数の値の取得に使用されるデータソースクエリ
refresh	変数を更新するタイミングを設定します
regex	シリーズ名またはメトリクスノードセグメントの一部を抽出します
type	変数、custom、query、または interval のタイプ

ダッシュボードのベストプラクティス

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このセクションでは、Grafana ダッシュボードを構築および保守する方法について、Grafana 管理者およびユーザー向けのベストプラクティスに関する情報を提供します。

作成できるさまざまな種類のダッシュボードの詳細については、Grafana Labs ウェブサイトのブログ記事をの「Grafana ダッシュボード: 作成できるさまざまなタイプの包括的ガイド」を参照してください。

Note

このセクションでは、モニタリングとダッシュボードのメンテナンスの戦略を作成するのに役立ちます。お使いのシステムを一番理解しているのはユーザーです。このセクションを使

用して理解を深めてください。最終的には、システムに最適な戦略を作成するのはユーザーの責任です。

一般的なオブザーバビリティ戦略

サーバーファームなど、監視対象が多い場合は、監視するのに十分な重要性を判断する戦略が必要です。このページでは、監視対象を選択する一般的な方法をいくつか説明します。

論理戦略を使用すると、均一なダッシュボードを作成し、オブザーバビリティプラットフォームをより簡単に拡張できます。

戦略のガイドライン

- USE メソッドはマシンの満足度を、RED メソッドはユーザーの満足度を示します。
- USE は問題の原因を報告します。
- RED はユーザーエクスペリエンスを報告し、問題の症状を報告する可能性が高くなります。
- 両方を監視することは、システムを理解するために重要です。ベストプラクティスとして、原因ではなく症状に注意してください。通常、アラートは RED ダッシュボードで設定されます。

USE メソッド

USE は、以下を表します。

- [使用率] — ノード CPU 使用率など、リソースがビジー状態である時間の割合。
- [飽和] — リソースが実行する必要がある作業の量。多くの場合、キューの長さやノード負荷です。
- [エラー] – エラーイベントの数。

この方法は、CPU、メモリ、ネットワークデバイスなどのインフラストラクチャのハードウェアリソースに最適です。詳細については、Brendan Gregg のブログ記事「USE メソッド」を参照してください。

RED メソッド

RED は、以下を表します。

- [レート] – 1 秒あたりのリクエスト数

- [エラー] – 失敗したリクエスト数。
- [期間] – これらのリクエストにかかる時間、レイテンシー測定値の分布。

この方法は、サービス、特にマイクロサービス環境に最も適しています。各サービスについて、コードを計測して、各コンポーネントのこれらのメトリクスを公開します。RED ダッシュボードは、アラートや SLA に適しています。適切に設計された RED ダッシュボードは、ユーザーエクスペリエンスのプロキシです。

詳細については、Tom Wilkie のブログ記事「RED メソッド: サービスの計測方法」を参照してください。

4 つのゴールデンシグナル

Google SRE ハンドブックによると、ユーザー向けシステムの 4 つのメトリクスしか測定できない場合は、この 4 つのメトリクスに重点を置いてください。

このメソッドは RED メソッドに似ていますが、飽和が含まれています。

- [レイテンシー] – リクエストの処理にかかる時間。
- [トラフィック] – システムにどれだけの需要があるか。
- [エラー] – 失敗したリクエストのレート。
- [飽和] — システムがどれだけ「飽和」状態であるか

ダッシュボード管理成熟度モデル

ダッシュボード管理の成熟度とは、ダッシュボードエコシステムの設計と効率性を指します。ダッシュボードの設定を定期的に見直して、現在の状況と改善方法を判断することをお勧めします。

大まかに言うと、ダッシュボードの成熟度は低、中、高と定義できます。

このトピックのコンテンツの多くは、KubeCon 2019 トーク「スリープ配信オンコール用のフールプルーフ Kubernetes ダッシュボード」から取得されました。

低 – デフォルト状態

この段階では、一貫性のあるダッシュボード管理戦略はありません。ほぼすべてがここから開始します。

ここに位置しているとどのように認識できるのでしょうか？

- 誰でもダッシュボードを変更できます。
- 多くのダッシュボードがコピーされているが、ダッシュボードの再利用はほとんどまたはまったくありません。
- 永久に残る 1 回限りのダッシュボード。
- バージョンコントロールなし (バージョンコントロールのダッシュボード JSON)。
- ダッシュボードのブラウジングが多く、適切なダッシュボードを検索します。つまり、必要なダッシュボードを探すために多くの時間を無駄にすることになります。
- 適切なダッシュボードに誘導するアラートはありません。

中 – 系統的なダッシュボード

この段階では、系統的なダッシュボードでダッシュボードの使用を管理し始めます。戦略を立てたかもしれませんが、改善できる点がいくつかあります。

ここに位置しているとどのように認識できるのでしょうか？

- テンプレート変数を使用してスプロールを防止します。例えば、ノードごとに個別のダッシュボードは必要ありません。クエリ変数を使用できます。さらに、データソースをテンプレート変数にすることもできます。これにより、異なるクラスター間で同じダッシュボードを再利用したり、バックエンドを監視したりできます。

  アイデアについては、「変数」の例のリストを参照してください。
- オブザーバビリティ戦略に従った系統的なダッシュボード。
- 次のレベルにドリルダウンした階層ダッシュボード。
- ダッシュボードの設計には、サービス階層が反映されます。例えば、サービスごとに 1 行のREDメソッドを使用できます。ダッシュボードをスクロールダウンすると、行の順序にデータフローが反映される可能性があります。
- 大きさが異なる場合にサービスダッシュボードを分割することで比較します。集約されたメトリクスが重要な情報を枯渇させないようにしてください。
- 可能な場合は色を効果的に使用し、軸を正規化する表現豊かなグラフを作成します。
  - 効果的な色の例: 青は良好、赤は問題があることを意味します。しきい値はそれに役立ちます。
  - 軸の正規化の例: CPU 使用率を比較する場合は、マシンのコア数が異なる可能性があるため、生の数値ではなくパーセンテージで測定します。CPU 使用率をコア数で正規化すると、CPU の数を知らなくても、すべてのコアが 100% で使用されていることを信頼できるため、認識負荷が軽減されます。

- 誘導ブラウジングは推測を減らします。
  - テンプレート変数を使用すると、ランダムにまたは目的なくブラウズするのが難しくなります。
  - ほとんどのダッシュボードは、アラートによってリンクする必要があります。
  - ブラウジングはリンクで指示されます。詳細については、「ダッシュボードリンクの管理」を参照してください。
- バージョン管理されたダッシュボード JSON。

高 – 最適化使用

この段階では、一貫性のある思慮深い戦略でダッシュボード管理の使用を最適化できています。メンテナンスが必要ですが、結果には価値があります。

- スプロールを積極的に減らします。
  - 既存のダッシュボードを定期的に確認し、関連性があることを確認します。
  - マスターダッシュボードリストに追加された承認済みダッシュボードのみが対象です。
  - ダッシュボードの使用を追跡します。[使用率のインサイト] を活用できます。
- 設計上の一貫性を確保します。
- スクリプティングライブラリを使用してダッシュボードを生成し、パターンとスタイルの一貫性を確保します。
  - grafonnet (Jsonnet)
  - grafanalib (Python)
- ブラウザでは編集機能を提供しません。ダッシュボードビューワーでは、変数を使用してビューを変更します。
- ダッシュボードの閲覧は例外であり、ルールではありません。
- 本番環境のインスタンスではなく、その目的専用の別の Grafana インスタンスで実験とテストを実行します。テスト環境のダッシュボードが有用であることが証明されたら、そのダッシュボードをメイン Grafana インスタンスに追加します。

ダッシュボードを作成するためのベストプラクティス

このセクションでは、Grafana ダッシュボードを作成するときに従うべきいくつかのベストプラクティスについて説明します。

[開始する前に]

ダッシュボードを作成する前に考慮すべき原則をいくつかご紹介します。

ダッシュボードはストーリーを伝えたり、質問に答えたりするべきである

ダッシュボードでどのようなストーリーを伝えようとしていますか？ 大規模から小規模、または一般的から特定まで、データの論理的な進行を作成してみてください。このダッシュボードの目標は何ですか？ (ヒント: ダッシュボードに目標がない場合は、ダッシュボードが本当に必要かどうかを自分に問いかけてみてください)

グラフはシンプルに保ち、尋ねる質問への回答に集中してください。例えば、質問が「どのサーバーに問題があるか」である場合、すべてのサーバーデータを表示する必要はありません。問題のあるデータのみを表示します。

ダッシュボードは認知負荷を軽減し、付加すべきでない

認知負荷とは、基本的には何かを理解するためにどれだけ多くのことを考える必要があるかということです。ダッシュボードの解釈を容易にします。他のユーザーと未来のあなた (午前 2 時に何が壊れたかを理解しようとしているとき) は、それを有用に思うでしょう。

自分に問いかけてください。

- 各グラフが正確に何を表すかわかりますか？ それは明白ですか、それともそれについて考える必要がありますか？
- それを他の誰かに見せると、その人物がそれを理解するのにどのくらい時間がかかりますか？ わからなくなる場合があるでしょうか？

監視戦略を持つ

新しいダッシュボードを簡単に作成できます。ダッシュボードの作成を最適化し、計画に従うのは難しいですが、その価値はあります。この戦略は、ダッシュボードスキーム全体を管理し、個々のダッシュボード設計に一貫性を持たせるべきです。

詳細については、「一般的なオブザーバビリティ戦略」と「ダッシュボード管理の成熟度レベル」を参照してください。

書き留める

戦略または設計ガイドラインを作成したら、時間の経過に伴う一貫性を維持するために書き留めます。

従うべきベストプラクティス

- 新しいダッシュボードを作成するときは、意味のある名前をつけましょう。
  - 再生または実験目的でダッシュボードを作成する場合は、名前に TESTまたは TMPという単語を入力します。
  - ダッシュボード名に自分の名前またはイニシャルを含めるか、タグとして含めて、ダッシュボードの所有者を知らせることを検討してください。
  - 一時実験ダッシュボードは、完了したら削除するようにします。
- 関連するダッシュボードを多数作成する場合は、ナビゲーションを容易にするためにそれらを相互参照する方法を検討してください。詳細については、このセクション後半の「ダッシュボードを管理するためのベストプラクティス」を参照してください。
- Grafana はデータソースからデータを取得します。データソースに接続する 全般、および特定のデータソースに関する基本的な理解が重要です。
- 不要なダッシュボードの更新を避けて、ネットワークまたはバックエンドの負荷を軽減します。例えば、データが 1 時間ごとに変更される場合、ダッシュボードの更新レートを 30 秒に設定する必要はありません。
- 異なる単位または範囲の時系列を表示するときは、左右の Y 軸を使用します。
- ダッシュボードとパネルにドキュメントを追加します。
  - ダッシュボードにドキュメントを追加するには、テキストパネルの視覚化をダッシュボードに追加します。ダッシュボードの目的、便利なリソースリンク、ユーザーがダッシュボードを操作するために必要な手順などを記録します。
  - パネルにドキュメントを追加するには、パネル設定を編集し、説明を追加します。追加したテキストは、パネルの左上隅にある小さな i 部分にカーソルを合わせると表示されます。
- テンプレートと変数を使用して、ダッシュボードを再使用して一貫性を強化します。
- グラフデータのスタックには注意してください。視覚化は誤解を招き、重要なデータを覆い隠す可能性があります。ほとんどの場合、オフにすることをお勧めします。

ダッシュボードを管理するためのベストプラクティス

このページでは、Grafana ダッシュボードを管理するときに従うべきいくつかのベストプラクティスについて説明します。

[開始する前に]

ダッシュボードの管理を開始する前に考慮すべき原則をいくつかご紹介します。

戦略的オブザーバビリティ

一般的なオブザーバビリティ戦略はいくつかあります。これらを調査し、そのうちの 1 つを採用するか、自分で戦略を考えるかを決定すべきです。いずれにせよ、計画を立て、書き留めて、それに従います。

必要に応じて、変化するニーズに戦略を適応させます。

成熟度レベル

ダッシュボードの成熟度レベルは？ 現在のダッシュボード設定を分析し、ダッシュボード管理成熟度モデルと比較します。現時点の状況を理解することは、希望する場所にたどり着く方法を決定するのに役立ちます。

従うべきベストプラクティス

- ダッシュボードのスプロールは避けてください。つまり、ダッシュボードの制御されない成長は避けてください。ダッシュボードのスプロールは、適切なダッシュボードを見つける時間に悪影響を及ぼします。ダッシュボードの複製と「単一の項目」の変更 (元のタグを保持することが最悪のシナリオです) が最も簡単なスプロールです。
  - ダッシュボードを定期的に確認し、不要なダッシュボードを削除します。
  - 一時的なダッシュボードをテストなどの目的に作成する場合、名前のプレフィックスに TEST: を付けます。完了したらダッシュボードを削除します。
- 大きな変更のないダッシュボードをコピーすることはお勧めしません。
  - ドキュメントの変更、バグ修正、メトリクスの追加など、元のダッシュボードの更新を見逃す可能性があります。
  - 多くの場合、テンプレートパラメータを設定してビューをカスタマイズするためにコピーが行われています。代わりに、マスターダッシュボードへのリンクを維持し、URL パラメータを使用してビューをカスタマイズしてこれを行うべきです。
- ダッシュボードをコピーする必要がある場合は、名前を明確に変更し、ダッシュボードタグをコピーしないでください。タグは、検索中に使用されるダッシュボードの重要なメタデータです。タグをコピーすると、誤一致が発生する可能性があります。
- ダッシュボードまたは相互参照ダッシュボードのダッシュボードを維持します。これには、いくつかの方法があります。
  - ダッシュボードリンク、パネル、またはデータリンクを作成します。リンクは、他のダッシュボードまたは外部システムに移動できます。詳細については、「ダッシュボードリンクの管理」を参照してください。
  - ダッシュボードリストパネルを追加します。その後、タグまたはフォルダ検索を実行して、表示内容をカスタマイズできます。

- テキストパネルを追加し、マークダウンを使用して表示をカスタマイズします。

ダッシュボードの管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ワークスペースの [ダッシュボード] ページ (左側のメニューから [ダッシュボード] を選択して使用可能) で、ダッシュボードをフォルダに整理するなど、ダッシュボード管理タスクを実行できます。

ダッシュボードの作成の詳細については、「ダッシュボードの構築」を参照してください。

ダッシュボードの参照

[ダッシュボード] ページで、フォルダとダッシュボードを参照および管理できます。これには、以下のオプションが含まれます。

- フォルダとダッシュボードを作成します。
- ダッシュボードをフォルダ間で移動します。
- 複数のダッシュボードとフォルダを削除します。
- フォルダに移動します。
- フォルダのアクセス許可を管理します」。詳細については、「ダッシュボードおよびフォルダのアクセス権限」を参照してください。

ダッシュボードフォルダの作成

フォルダはダッシュボードの整理とグループ化に役立ちます。これは、同じGrafanaインスタンスを使用する多くのダッシュボードや複数のチームがある場合に便利です。サブフォルダを使用すると、ダッシュボード組織にネストされた階層を作成できます。

前提条件

Grafana 管理者のアクセス許可があることを確認します。ダッシュボードのアクセス許可の詳細については、「ダッシュボードおよびフォルダのアクセス権限」を参照してください。

ダッシュボードフォルダを作成するには

1. Grafana にサインインします。
2. 左側のメニューで、[ダッシュボード] を選択します。
3. [ダッシュボード] ページで、[新規] を選択し、ドロップダウンで [新しいフォルダ] を選択します。
4. 一意の名前を入力し、[作成] をクリックします。

> **Note**
>
> ダッシュボードを保存するときは、ダッシュボードを保存するフォルダを選択するか、新しいフォルダを作成するかを選択できます。

フォルダの名前を編集するには

1. 左側のメニューで [ダッシュボード] を選択します。
2. 名前を変更するフォルダを選択する
3. ヘッダーで [タイトルの編集] (鉛筆) アイコンを選択し、フォルダの名前を更新します。

   新しいフォルダ名は自動的に保存されます。

フォルダのアクセス許可

フォルダにアクセス許可を割り当てることができます。フォルダ内のダッシュボードは、フォルダに割り当てたアクセス許可を継承します。組織のロール、チーム、ユーザーにアクセス許可を割り当てることができます。

フォルダのアクセス許可を変更するには

1. 左側のメニューから [ダッシュボード] を選択します。
2. リスト内のフォルダを選択します。
3. フォルダの詳細ページで、[フォルダアクション] を選択し、ドロップダウンリストから [アクセス許可の管理] を選択します。

4. 必要に応じてアクセス許可を更新します。

変更は自動的に保存されます。

プレイリストの管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

プレイリストは、順番に表示されるダッシュボードのリストです。プレイリストを使用して状況認識を構築したり、チームや訪問者にメトリクスを提示したりできます。Grafana はダッシュボードを任意の解像度に自動的にスケーリングするため、大きな画面に最適です。[ダッシュボード] サブメニューの Grafana のサイドメニューからプレイリスト機能にアクセスできます。

プレイリストへのアクセス、共有、制御

このセクションの情報を参照して、既存のプレイリストにアクセスします。5 つの使用可能なモードのいずれかを使用して、プレイリストの表示を開始して制御します。

プレイリストにアクセスするには

1. 左側のメニューから [Playlists] (プレイリスト) を選択します。
2. 既存のプレイリストのリストからプレイリストを選択します。

プレイリストの開始

プレイリストの開始は 5 種類のビューモードから選択できます。表示モードによって、ダッシュボードへのメニューとナビゲーションバーの表示方法が異なります。

デフォルトでは、各ダッシュボードは間隔フィールド (プレイリストの作成時または編集時に設定可能) に入力された時間表示されます。プレイリストを開始したら、ページ上部のナビゲーションバーを使用してプレイリストを制御できます。

プレイリストを開始するには

1. プレイリストページにアクセスして、既存のプレイリストのリストを表示します。
2. 開始するプレイリストを検索し、[プレイリストの開始] をクリックします。

   プレイリストの開始ダイアログボックスが開きます。
3. 次の表の情報に基づいて、使用可能な 5 つのプレイリストモードのいずれかを選択します。
4. [<プレイリスト名> を開始] をクリックします。

プレイリストには、プレイリストの作成時または編集時に設定された Interval フィールドで指定された時間の間、各ダッシュボードが表示されます。プレイリスト開始後は、画面上部のナビゲーションバーで制御します。

モード	説明
通常モード	• サイドメニューは表示されたままです。 • ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。
TV モード	• サイドメニューとダッシュボードサブメニュー (変数ドロップダウンとダッシュボードリンクを含む) は非表示となるか、取り除かれます。 • ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。 • ユーザーが 1 分間非アクティブになると、自動的に有効になります。 • d v シーケンスショートカットを使用するか、ダッシュボード URL に ?inactive パラメータを追加して手動で有効にします。 • ポインタ移動またはキーボードアクションで無効にします。
TV モード (自動フィットパネル付き)	• ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。

モード	説明
	• ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。
キオスクモード	• サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。 • プレイリストの開始後、d v シーケンスショートカットを使用して手動で有効にできます。 • 同じショートカットを使用して手動で無効にすることもできます。
キオスクモード (自動フィットパネル付き)	• サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。 • ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。

プレイリストの制御

プレイリストの開始後は、画面上部のナビゲーションバーを使用して、[普通]モードまたは[テレビ]モードでプレイリストを制御できます。キーボードの Esc キーを押して、プレイリストを停止します。

ボタン	アクション
次へ (二重右矢印)	次のダッシュボードに進みます。
戻る (左矢印)	前のダッシュボードに戻ります。
停止 (四角)	プレイリストを終了し、現在のダッシュボードに戻ります。
サイクルビューモード (モニターアイコン)	ダッシュボードの表示をさまざまなビューモードでローテーションします。

ボタン	アクション
[時間範囲]	指定した時間範囲内のデータを表示します。下矢印を使用して、過去 5 分から 5 年前、または表示するカスタム時間範囲を設定します。
更新 (円矢印)	ダッシュボードを再読み込みして、最新のデータを表示します。ドロップダウン矢印を使用して、5 秒から 1 日ごとに自動的にリロードするように設定することもできます。

プレイリストの作成

プレイリストを作成して、ダッシュボード間の順序と時間間隔を設定して、ダッシュボードを順番に表示できます。

プレイリストを作成するには

1. 左側のメニューから [ダッシュボード] を選択します。
2. [Playlists] (プレイリスト) ページでプレイリストを選択します。
3. [新しいプレイリスト] を選択します。
4. [名前] テキストボックスにわかりやすい名前を入力します。
5. [間隔] テキストボックスに時間間隔を入力します。追加したダッシュボードは、順番に表示されます。
6. [ダッシュボード] で、[タイトルで追加] および [タグで追加] ドロップダウンオプションを使用して、既存のダッシュボードをプレイリストに追加します。
7. オプションで次のことが可能です:
   - ダッシュボードの名前、正規表現、またはタグでダッシュボードを検索します。
   - スター付きのステータスやタグで結果をフィルタリングします。
   - 上下矢印アイコンを使用して、追加したダッシュボードの順序を変更します。
   - ダッシュボードの横にある x アイコンをクリックして、プレイリストからダッシュボードを削除します。
8. 変更を保存するには [保存] を選択します。

プレイリストの保存

プレイリストを保存してプレイリストページに追加し、そこでプレイリストを開始できます。

> **Important**
>
> プレイリストを保存する前かつプレイリストを作成または編集するときに、プレイリストに表示するすべてのダッシュボードが追加されていることを確認します。

プレイリストを保存するには

1. 左側のメニューで [ダッシュボード] を選択します。
2. [Playlists] (プレイリスト) を選択して、利用可能なプレイリストを表示します。
3. 所望のプレイリストを選択します。
4. プレイリストを編集します。
5. プレイリストに [名前]、[間隔] が設定されており、少なくとも 1 つの[ダッシュボード]が追加されていることを確認します。
6. 変更を保存するには [保存] を選択します。

プレイリストの編集または削除

プレイリストを編集するには、名前、間隔時間を更新し、ダッシュボードの順序を追加、削除、再配置するか、プレイリストを削除できます。

プレイリストを編集するには

1. プレイリストページで [プレイリストの編集] を選択します。
2. 上記の「プレイリストの作成」のステップに従って、名前と時間間隔を更新し、プレイリストからダッシュボードを追加または削除します。
3. 変更を保存するには [保存] を選択します。

プレイリストを削除するには

1. [Playlists] (プレイリスト) を選択します。
2. 削除するプレイリストの横にある [削除] を選択します。

プレイリストでダッシュボードの順序を変更するには

1. 順序を変更するダッシュボードの横にある上矢印または下矢印をクリックします。
2. 変更を保存するには [保存] を選択します。

ダッシュボードを削除するには

1. [削除] を選択して、プレイリストからダッシュボードを削除します。
2. 変更を保存するには [保存] を選択します。

ビューモードでプレイリストを共有する

お好みのビューモードで URL をコピーし、宛先に URL を貼り付けることで、プレイリストを共有できます。

ビューモードでプレイリストを共有するには

1. 左側のメニューの [ダッシュボード] から、[プレイリスト] を選択します。
2. 共有するプレイリストの横にある [プレイリストの開始] を選択します。
3. ドロップダウンで、目的のビューモードを右クリックします。
4. [リンクアドレスをコピー] を選択すると、URL がクリップボードにコピーされます。
5. 送信先に URL を貼り付けます。

ダッシュボードとパネルの共有

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana を使用すると、組織内の他のユーザーとダッシュボードやパネルを共有したり、特定の状況でウェブ上で公開したりできます。以下を使用して共有できます。

- 直接リンク
- スナップショット
- エクスポートリンク (ダッシュボードのみ)

直接リンクによってレンダリングされた画像を表示するには、許可されたビューワーアクセス許可が必要です。

パネルまたはダッシュボードをスナップショットとして共有すると、スナップショット (スナップショットを作成した時点のパネルまたはダッシュボード) がウェブ上で公開されます。リンクを持つユーザーは誰でもアクセスできます。スナップショットは表示に許可を必要としないため、Grafana は、取得したアカウントに関する情報と機密データをスナップショットから削除します。

ダッシュボードの共有

ダッシュボードは、直接リンクまたはスナップショットとして共有できます。ダッシュボードをエクスポートすることもできます。

> **Note**
>
> ダッシュボードを変更する場合は、共有する前に変更を保存してください。

ダッシュボードを共有するには

1. ワークスペースの左側のメニューから [ダッシュボード] を選択します。
2. 共有するダッシュボードを選択します。
3. 画面上部の共有アイコンを選択します。

   共有ダイアログボックスが開き、[リンク] タブが表示されます。

ダイレクトリンクの共有

[リンク] タブには、現在の時間範囲、テンプレート変数、デフォルトのテーマが表示されます。短縮 URL を共有することもできます。

直接リンクを共有するには

1. [Copy] (コピー) を選択します。このアクションにより、デフォルトまたは短縮 URL がクリップボードにコピーされます。

2. コピーした URL を、リンクを表示する権限を持つ Grafana ユーザーに送信します。

スナップショットの公開

ダッシュボードスナップショットは、インタラクティブダッシュボードをパブリックに共有します。Grafana はクエリ (メトリクス、テンプレート、注釈) やパネルリンクなどの機密データを消去し、ダッシュボードに埋め込まれた可視メトリクスデータとシリーズ名のみが残されます。ダッシュボードスナップショットには、リンクを持つすべてのユーザーがアクセスできます。

スナップショットはローカルインスタンスに発行できます。

スナップショットを公開するには

1. [Snapshot] (スナップショット) タブを選択します。
2. [Local Snapshot] (ローカルスナップショット) を選択します。
3. Grafana がスナップショットのリンクを生成します。スナップショットリンクをコピーし、組織内またはウェブ上で公開します。

ダッシュボードのエクスポート

Grafana ダッシュボードは簡単にエクスポートおよびインポートできます。詳細については、ダッシュボードの構築 の「インポートとエクスポート」を参照してください。

パネルの共有

パネルは、直接リンクまたはスナップショットとして共有できます。任意のパネルで [共有] オプションを使用してライブラリパネルを作成することもできます。

パネルを共有するには

1. 共有するパネルのパネルタイトルを選択します。パネルメニューが開きます。
2. [共有] を選択します。共有ダイアログボックスが開き、[リンク] タブが表示されます。

直接リンクの使用

[リンク] タブには、現在の時間範囲、テンプレート変数、デフォルトのテーマが表示されます。必要に応じて、短縮 URL を有効にして共有できます。

直接リンクの使用

1. [Copy] (コピー) を選択して、デフォルトまたは短縮 URL をクリップボードにコピーします。
2. コピーした URL を、リンクを表示する権限を持つ Grafana ユーザーに送信します。

パネルスナップショットの公開

パネルスナップショットは、インタラクティブパネルのパブリック共有です。Grafana は機密データを取り除き、ダッシュボードに埋め込まれた可視メトリクスデータとシリーズ名のみを残します。パネルスナップショットには、リンクを持つすべてのユーザーがアクセスできます。

スナップショットはローカルインスタンスに発行できます。

パネルスナップショットを公開するには

1. [パネルの共有] ダイアログボックスで、[スナップショット] タブを選択します。
2. [Local Snapshot] (ローカルスナップショット) を選択します。Grafana はスナップショットのリンクを生成します。
3. スナップショットリンクをコピーし、組織内またはウェブ上で公開します。

誤ってスナップショットを作成した場合は、[スナップショットの削除] をクリックして、Grafana インスタンスからスナップショットを削除します。

ライブラリパネルの作成

[パネルの共有] ダイアログボックスからライブラリパネルを作成できます。

ライブラリパネルの作成方法

1. [ライブラリパネルl] を選択します。
2. [ライブラリパネル名] に名前を入力します。
3. [フォルダに保存]、ライブラリパネルを保存するフォルダを選択します。デフォルトでは、ルートフォルダが選択されています。
4. [ライブラリパネルの作成] を選択して変更を保存します。
5. ダッシュボードを保存します。

変数

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

変数は、値のプレースホルダーです。メトリクスクエリとパネルタイトルで変数を使用できます。したがって、ダッシュボードの上部にあるドロップダウンを使用して値を変更すると、パネルのメトリッククエリが変更され、新しい値が反映されます。

変数を使用すると、よりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウンリストとして表示されます。これらのドロップダウンを使用すると、ダッシュボード内に表示されるデータを簡単に変更できます。

これらは、Grafana ビューワーが視覚化をすばやく調整できるようにしたいが、完全な編集権限を付与したくない管理者にとって特に便利です。Grafana ビューワーは変数を使用できます。

変数とテンプレートを使用すると、単一ソースのダッシュボードを使用することもできます。同じデータソースまたはサーバーが複数ある場合は、1 つのダッシュボードを作成し、変数を使用して表示内容を変更できます。これにより、メンテナンスと維持が大幅に簡素化されます。

テンプレート

テンプレートは、変数を含むすべてのクエリです。たとえば、複数のサーバーを監視するダッシュボードを管理する場合は、サーバーごとにダッシュボードを作成するか、1 つのダッシュボードを作成して、次のようなテンプレートクエリを含むパネルを使用できます。

```
wmi_system_threads{instance=~"$server"}
```

変数値は、常に var-<varname>=value という構文を使用して URL に同期されます。

例

変数は、画面上部のドロップダウンリストに表示されます。さまざまな変数を選択して、視覚化がどのように変化するかを確認します。

変数設定を表示するには、[ダッシュボード設定] > [変数] に移動します。リスト内の変数をクリックすると、その設定が表示されます。

変数は、タイトル、説明、テキストパネル、クエリで使用できます。$ で始まるテキストを含むクエリはテンプレートです。すべてのパネルにテンプレートクエリがあるわけではありません。

変数のベストプラクティス

- 変数のドロップダウンリストは、[ダッシュボード設定] の変数リストに一覧されている順序で表示されます。
- 頻繁に変更する変数を一番上に置くと、ダッシュボードの左端に最初に表示されるようになります。
- 変数は、デフォルトでドロップダウンリストの最上位値を事前に選択します。代わりに空の値を選択する場合は、次のように変数の設定を変更します。
  1. [すべてのオプションを含める] チェックボックスを選択します。
  2. [すべての値のカスタム化] フィールドに、値 + を入力します。

トピック

- 変数の追加と管理
- 変数の検査
- 変数構文

変数の追加と管理

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

次の表に、Grafana の変数タイプ一覧を示します。

変数タイプ	説明
クエリ	メトリクス名、サーバー名、センサー ID、データセンターなど、クエリによって生成された値のリストです。
カスタム	カンマ区切りリストを使用して変数オプションを手動で定義します。
テキストボックス	オプションのデフォルト値を使用して、フリーテキスト入力フィールドを表示します。
定数	非表示の定数を定義します。
データソース	ダッシュボード全体のデータソースをすばやく変更します。
間隔	間隔変数は時間範囲を表します。
アドホックフィルター	データソースのすべてのメトリクスクエリに自動的に追加されるキー-値のフィルターを指します (Prometheus、Loki、InfluxDB、Elasticsearch のみ)。
グローバル変数	クエリエディタの式で使用できる組み込み変数です。
連鎖変数	変数クエリには、他の変数を含めることができます。

トピック

- 一般的なオプションの入力
- クエリ変数の追加
- カスタム変数の追加
- テキストボックス変数の追加
- 定数変数の追加
- データソース変数の追加

- 間隔変数の追加
- アドホックフィルターの追加
- 変数選択オプションを設定する
- グローバル変数
- 連鎖変数
- 変数の管理
- 正規表現での変数のフィルタリング

一般的なオプションの入力

作成する変数のタイプには、一般的なオプションを入力する必要があります。

一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新しい変数] を選択します。
3. 変数の [名前] を入力します。
4. [タイプ] 一覧で、[クエリ] を選択します。
5. (オプション) [ラベル] に、変数ドロップダウンの表示名を入力します。

   表示名を入力しない場合、ドロップダウンラベルは変数名です。
6. [非表示] オプションを次から選択します。

   - [選択なし (空白)] – 変数ドロップダウンには、変数の名前またはラベル値が表示されます。
   - [ラベル] – 変数ドロップダウンには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンが表示されません。

クエリ変数の追加

クエリ変数を使用すると、メトリクス名、タグ値、またはキーのリストを返すデータソースクエリを記述できます。例えば、クエリ変数は、サーバー名、センサー ID、またはデータセンターのリストを返す場合があります。変数値は、データソースクエリでオプションを動的に取得するにつれて変化します。

クエリ変数は通常、文字列でのみサポートされます。クエリが数値またはその他のデータ型を返す場合は、変数として使用するために文字列に変換する必要があります。例えば、Azure データソースの場合、この目的のために tostring 関数を使用できます。

クエリ式には、他の変数への参照を含めることができ、実際にはリンクされた変数を作成できます。Grafana はこれを検出し、リンクされた変数のいずれかが変更されると変数を自動的に更新します。

> **Note**
>
> クエリ式はデータソースごとに異なります。詳細については、ご使用のデータソースのドキュメントを参照してください。

クエリ変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [データソース] リストで、クエリのターゲットデータソースを選択します。
3. [更新] リストで、変数がオプションを更新するタイミングを選択します。
   - [ダッシュボード読み込み時] - ダッシュボードが読み込まれるたびにデータソースにクエリを実行します。これにより、ダッシュボードの読み込みが遅くなります。ダッシュボードを初期化する前に変数クエリを完了する必要があるためです。
   - [時間範囲の変更時] - ダッシュボードの時間範囲が変更されたときにデータソースにクエリを実行します。このオプションは、可変オプションクエリに時間範囲フィルターが含まれているか、ダッシュボードの時間範囲に依存している場合にのみ使用します。
4. [クエリ] フィールドにクエリを入力します。
   - クエリフィールドはデータソースによって異なります。一部のデータソースには、カスタムクエリエディタがあります。
   - クエリは、 `__text` および `__value` という名前の値を返す必要があります。例えば、SQL では、`SELECT hostname AS __text, id AS __value from MyTable` などのクエリを使用できます。他の言語のクエリは構文によって異なります。
   - 1 つの入力フィールドクエリエディタにより多くのスペースが必要な場合は、フィールドの右下隅にある線上にカーソルを合わせ、下にドラッグして展開します。

5. (オプション) [正規表現] フィールドに正規表現を入力して、データソースクエリによって返される名前の特定部分をフィルタリングまたはキャプチャします。例については、「正規表現での変数のフィルタリング」を参照してください。
6. [ソート] リストで、ドロップダウンリストに表示される値のソート順序を選択します。デフォルトのオプション [無効] では、データソースクエリによって返されるオプションの順序が使用されます。
7. (オプション) [選択オプション] を入力します。
8. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
9. [追加] を選択してダッシュボードに変数を追加します。

カスタム変数の追加

数値や文字列など、変更されない値にはカスタム変数を使用します。

例えば、サーバー名やリージョン名が変更されない場合は、クエリ変数ではなくカスタム変数として作成できます。これらは変更されないため、他のクエリ変数ではなく、連鎖変数で使用できます。これにより、連鎖変数が更新されたときに Grafana が送信する必要があるクエリの数が減少します。

カスタム変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [カンマで区切られた値] リストで、この変数の値をカンマ区切りリストに入力します。スペースおよびコロンで区切られた数値、文字列、またはキーと値のペアを含めることができます。例えば、`key1 : value1,key2 : value2`。
3. (オプション) [選択オプション] を入力します。
4. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
5. [追加] を選択してダッシュボードに変数を追加します。

テキストボックス変数の追加

テキストボックス変数は、任意のデフォルト値を使用して自由テキスト入力フィールドを表示します。これは、任意の値を入力できるため、最も柔軟な変数です。カーディナリティの高いメトリクスがある場合や、ダッシュボード内の複数のパネルを同時に更新する場合は、このタイプの変数を使用します。

テキストボックス変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. (オプション) [デフォルト値] フィールドで、変数のデフォルト値を選択します。このフィールドに何も入力しない場合、Grafana はユーザーがテキストを入力するための空のテキストボックスを表示します。
3. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

定数変数の追加

定数変数を使用すると、非表示の定数を定義できます。定数変数は、共有するダッシュボードに対するメトリックパスの接頭辞に有用です。ダッシュボードをエクスポートすると、定数変数がインポートオプションに変換されます。

定数変数は複数設定できません。各定数変数は 1 つの値のみを保持し、変数設定を更新しない限り更新できません。

定数変数は、クエリに含める必要がある複素数値があっても、クエリすべてに再入力したくない場合に便利です。例えば、「`i-0b6a61efe2ab843gg`」というサーバーパスがある場合、それを「`$path_gg`」という変数に置き換えることができます。

定数変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [値] フィールドに変数値を入力します。英数字と記号を入力できます。raw 形式 を使用している場合は、ワイルドカードを使用することもできます。
3. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

データソース変数の追加

データソース変数を使用すると、ダッシュボード全体のデータソースをすばやく変更できます。データソースのインスタンスが複数ある場合、おそらく異なる環境にある場合に便利です。

データソース変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [タイプ] リストで、変数のターゲットデータソースを選択します。

   [高度なデータソースピッカーを開く] を選択して、データソースの追加 (管理者のみ可能) など、その他のオプションを表示することもできます。詳細については、「データソースに接続する」を参照してください。
3. (オプション) [インスタンス名フィルター] には、変数値のドロップダウンリストからデータソースインスタンスを選択する正規表現フィルターを入力します。このフィールドは空のままにして、すべてのインスタンスを表示します。
4. (オプション) [選択オプション] を入力します。
5. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
6. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の追加

間隔変数を使用して、1m、1h、1dなどの時間範囲を表します。これらはダッシュボード全体の group by time コマンドと考えることができます。間隔変数は、視覚化でのデータのグループ化方法を変更します。また、[自動] オプションを使用して、時間範囲ごとに複数のデータポイントを返すこともできます。

間隔変数を時間 (InfluxDB の場合)、日付ヒストグラム間隔 (Elasticsearch の場合) でグループ化するパラメータとして、または要約関数パラメータ (Graphite の場合) として使用できます。

間隔変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [値] フィールドに、変数ドロップダウンリストに表示する時間間隔を入力します。次の時間単位に対応しています: `s` (seconds)、`m` (minutes)、`h` (hours)、`d` (days)、`w` (weeks)、`M` (months)、`y` (years)。次のデフォルト値を使用または編集することもできます: `1m,10m,30m,1h,6h,12h,1d,7d,14d,30d`。
3. (オプション) `auto`オプションをリストに追加する場合は、[自動オプション] をオンにします。このオプションを使用すると、現在の時間範囲を分割して現在の `auto` 時間範囲を計算する回数を指定できます。オンにすると、さらに次の 2 つのオプションが表示されます。

- [ステップ数] – [最大データポイント数]クエリオプションと同様に、現在の時間範囲を分類して値を計算する回数を選択します。例えば、現在の表示時間範囲が 30 分の場合、auto 間隔はデータを 1 分単位× 30 でグループ化します。デフォルト値は 30 ステップです。
- [最小間隔] – ステップ数の間隔が時間を分類しない最小しきい値です。引き続き 30 分の例を挙げる場合、最小間隔をに設定すると、Grafana はデータを 2 分単位× 15 でグループ化します。

4. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
5. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の例

Graphite 関数でテンプレート変数 `myinterval` を使用する例:

```
summarize($myinterval, sum, false)
```

アドホックフィルターの追加

アドホックフィルターを使用して、指定されたデータソースを使用するすべてのメトリクスクエリに自動的に追加されるキー/値フィルターを追加できます。他の変数とは異なり、クエリではアドホックフィルターを使用しません。代わりに、アドホックフィルターを使用して既存のクエリのフィルターを書き込みます。

> **Note**
>
> アドホックフィルター変数は、Prometheus、Loki、InfluxDB、および Elasticsearch データソースでのみ機能します。

1. 上記のように、一般的なオプションを入力します。
2. [データソース] リストで、ターゲットデータソースを選択します。

   [高度なデータソースピッカーを開く] を選択して、データソースの追加 (管理者のみ可能) など、その他のオプションを表示することもできます。詳細については、「データソースに接続する」を参照してください。
3. [追加] を選択してダッシュボードに変数を追加します。

アドホックフィルターの作成

アドホックフィルターは、利用可能な最も複雑で変更可能な変数オプションの 1 つです。変数オプションの通常リストの代わりに、この変数によりダッシュボード全体のアドホッククエリを構築できます。この方法で適用するフィルターは、ダッシュボード上のすべてのパネルに適用されます。

変数選択オプションを設定する

選択オプションは、可変オプションの選択を管理するために使用できる機能です。すべての選択オプションは任意で使用でき、デフォルトではオフになっています。

複数値の変数

複数値が選択された変数を補間するのは、変数が使用されている特定のコンテキストで有効な文字列に複数の値をフォーマットする方法が単純ではないため、難しくなります。Grafana は、各データソースプラグインがテンプレート補間エンジンに、複数値に使用する形式を通知できるようにすることで、この問題を解決しようと試みます。

> **Note**
>
> Grafana がすべての値を 1 つの文字列にフォーマットするには、 変数の [すべての値をカスタム化] オプションを空白にする必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (一緒につなげる) します。例えば、`value1,value2,value3`。カスタム `all` 値を使用する場合は、値は代わりに `*` または `all` になります。

Graphite データソースを使用した複数値変数

Graphite は glob 式を使用します。この場合、複数の値を持つ変数は、現在の変数値が host1、host2、host3 であれば、`{host1,host2,host3}` として補間されます。

Prometheus または InfluxDB データソースを使用した複数値変数

InfluxDB と Prometheus は正規表現を使用するため、同じ変数は (`host1|host2|host3`) として補間されます。すべての値も正規表現エスケープされます。そうでない場合、正規表現コントロール文字を持つ値は正規表現を破ります。

Elastic データソースを使用した複数値変数

Elasticsearch は Lucene クエリ構文を使用するため、同じ変数は (`"host1" OR "host2" OR "host3"`) としてフォーマットされます。この場合、すべての値がエスケープされ、値には lucene 制御語と引用符のみが含まれます。

複数値変数のトラブルシューティング

自動エスケープとフォーマットは問題を引き起こす可能性があり、その背後にあるロジックを理解するのは難しい場合があります。特に InfluxDB と Prometheus では、正規表現構文を使用するために、変数を正規表現演算子コンテキストで使用する必要があります。

Grafana でこの自動正規表現エスケープとフォーマットを実行したくない場合は、次のいずれかを実行する必要があります。

- [複数値] をオフにするか、[すべて含めるオプション] オプションを使用します。
- raw 形式 を使用します。

すべてを含めるオプション

Grafana は、変数ドロップダウンリストに `All` オプションを追加します。ユーザーがこのオプションを選択すると、すべての変数オプションが選択されます。

すべての値のカスタム化

このオプションは、[すべてを含めるオプション] が選択されている場合にのみ表示されます。

[すべての値のカスタム化] フィールドに 正規表現、globs、または lucene 構文を入力して、 `All` オプションの値を定義します。

デフォルトでは、`All` 値は結合式内のすべてのオプションを含みます。これは非常に長くなり、パフォーマンスの問題が発生する可能性があります。ワイルドカード正規表現など、すべての値のカスタム化を指定する方がよい場合があります。

[すべての値のカスタム化] オプションにカスタム正規表現、globs、または lucene 構文を含めるには、エスケープされないため、データソースに有効な値は何かを検討する必要があります。

グローバル変数

Grafana には、クエリエディタの式で使用できるグローバル組み込み変数があります。このトピックでは、それらをアルファベット順に一覧表示し、定義します。これらの変数は、クエリ、ダッシュボードリンク、パネルリンク、データリンクに役立ちます。

$__dashboard

この変数は、現在のダッシュボードの名前です。

$__from および $__to

Grafana には、「$__from」と「$__to」の 2 つの組み込み時間範囲変数があります。これらは現在、デフォルトでは常に工ポックミリ秒として補間されていますが、日付の書式設定をユーザーが管理できます。

構文	結果の例	説明
${__from}	1594671549254	Unix 工ポックミリ秒
${__from:date}	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
${__from:date:iso}	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
${__from:date:seconds}	1594671549	Unix 工ポック秒
${__from:date:YYYY-MM}	2020-07	文字を含まないあらゆるカスタム日付形式

上記の構文は、${__to} としても機能します。

$__interval

$__interval 変数を時間 (InfluxDB 、MySQL、Postgres、MSSQL の場合)、日付ヒストグラム間隔 (Elasticsearch の場合) でグループ化するパラメータとして、または[要約]関数パラメータ (Graphite の場合) として使用できます。

Grafana は、クエリで時間別にグループ化するために使用できる間隔を自動的に計算します。グラフに表示されるデータポイントよりも多い場合、クエリはより長い間隔でグループ化することで、より効率的になります。例えば、3 か月分のデータのグラフを見ている場合、分レベルで詳細を表示できない場合があります。時間別または日別にグループ化すると、グラフの表示内容に影響を与えることなく、クエリの効率が向上します。$__interval は、時間範囲とグラフの幅 (ピクセル数) を使用して計算されます。

概算: `(to - from) / resolution`

例えば、時間範囲が 1 時間でグラフが全画面表示の場合、間隔は 2m で計算され、ポイントは 2 分間隔でグループ化されます。時間範囲が 6 か月で、グラフが全画面表示の場合、間隔は 1d (1 日) で、ポイントは日ごとにグループ化されます。

InfluxDB データソースでは、レガシー変数 $interval は同じ変数であり、$__interval を代わりに使用すべきです。

InfluxDB データソースと Elasticsearch データソースには、間隔をハードコード化したり、$__interval 変数の最小制限 (例えば Group by time interval など、> 構文を使用して) を設定したりするために使用される >10m フィールドがあります。

$__interval_ms

この変数はミリ秒単位の $__interval 変数であり、時間間隔形式の文字列ではありません。例えば、$__interval が 20m の場合、$__interval_ms は 1200000 です。

$__org

この変数は現在の組織の ID です。 ${__org.name} は現在の組織の名前を表します。

$__user

${__user.id} は現在のユーザーの ID です。 ${__user.login} は現在のユーザーのログインハンドルです。 ${__user.email}は現在のユーザーの E メールです。

$__range

これは、Prometheus および Loki データソースでのみサポートされています。この変数は、現在のダッシュボードの範囲を表します。to - from で計算されます。ミリ秒は $__range_ms、秒は $__range_s で表示されます。

$__rate_interval

Prometheus データソースでのみサポートされています。$__rate_interval 変数は、レート関数で使用することを目的としています。

$timeFilter または $__timeFilter

$timeFilter 変数は、現在選択されている時間範囲を式として返します。例えば、時間間隔「Last 7 days」の式は time > now() - 7d です。

これは、次のようないくつかの場所で使用されます。

- InfluxDB データソースの WHERE 句。クエリエディタ では、Grafana は InfluxDB クエリに自動的に追加します。テキストエディタモードでは、WHERE $timeFilter を手動で追加できます。

- Log Analytics は、Azure Monitor データソースでクエリを実行します。
- SQL は、MySQL 、Postgres、MSSQL でクエリを実行します。
- `$__timeFilter` 変数は MySQL データソースで使用されます。

$__timezone

`$__timezone` 変数は、現在選択されているタイムゾーン、`utc` または IANA タイムゾーンデータベースのエントリ (例: `America/New_York`) を返します。

現在選択されているタイムゾーンがブラウザ時間である場合、Grafana はブラウザのタイムゾーンを判別しようとします。

連鎖変数

連鎖変数は、リンクされた変数またはネストされた変数とも呼ばれ、変数クエリ内において 1 つ以上の他の変数を持つクエリ変数です。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべてのデータソースで同じです。連鎖変数クエリは、それらを使用できる任意のデータソースで使用できます。

非常に複雑なリンクされたテンプレート化されたダッシュボードは、5 または 10 レベルの深さで作成できます。技術的には、どれだけ深く、複雑にできるかに制限はありませんが、リンクが多いほどクエリの負荷が大きくなります。

ベストプラクティスとヒント

以下のプラクティスにより、ダッシュボードと変数の使用が容易になります。

新しいリンクされた変数の作成

- 連鎖変数は親/子の依存関係を作成します。ラダーまたはツリー構造として想定できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数の基盤となる変数をコピーすることです。変数リストで、変数エントリの右側にある変数の複製アイコンをクリックしてコピーを作成します。その後、親変数のクエリに追加できます。
- この方法で作成した新しい変数は、リストの下部に表示されます。論理的な順序にするには、リスト内の別の位置にドラッグする必要がある場合があります。

変数の順序

ダッシュボード変数リストの変数の順序は、各エントリの右側にある上下の矢印をクリックして変更できます。Grafana は、このリストに従って左から右に変数のドロップダウンを一覧表示し、一番上の変数を左端に表示します。

- 子変数の前に、親変数と依存関係のない変数のリストを作成します。
- 各変数は、依存している変数に従う必要があります。
- UI には、どの変数に依存関係があるのかを示すものがありません。変数を論理的な順序で一覧表示して、他のユーザー (および自分自身) に明確にします。

複雑さに関する考慮事項

変数の依存関係のレイヤーが多いほど、変数を変更した後にダッシュボードを更新するのに時間がかかります。

例えば、4 項目がリンクされた一連の変数 (国、地域、サーバー、メトリック) があり、ルート変数値 (国) を変更する場合、Grafana はダッシュボード内の視覚化を更新する前に、すべての従属変数に対してクエリを実行する必要があります。

変数の管理

変数ページでは、変数を追加し、既存の変数を管理できます。また、変数を検査し、変数が他の変数またはダッシュボードで参照 (または使用されている) されているかどうかを識別することもできます。

[移動] – ドラッグアンドドロップを使用して、変数をリストの上または下に移動できます。

[クローン] — 変数をクローン (コピー) するには、右側の一連のアイコンからクローンアイコンをクリックします。これにより、元の変数の名前に `copy_of_` というプレフィックスが付いた変数のコピーが作成されます。

[削除] – 変数を削除するには、右側の一連のアイコンからごみ箱アイコンをクリックします。

正規表現での変数のフィルタリング

[正規表現クエリ] オプションを使用して、変数クエリによって返されるオプションの一覧をフィルタリングするか、返されるオプションを変更します。

このページでは、正規表現を使用して変数ドロップダウンの値をフィルタリング/変更する方法を説明します。

[正規表現クエリ] オプションを使用して、変数クエリによって返されるオプションの一覧をフィルタリングするか、返されるオプションを変更します。詳細については、[正規表現]に関する Mozilla ガイドを参照してください。

次の例は、次のオプションリストのフィルタリングを示しています。

```
backend_01
backend_02
backend_03
backend_04
```

01 または **02** で終わるオプションのみが返されるようにフィルタリングする

正規表現:

```
/
(
01|02
)
$/
```

結果:

```
backend_01
backend_02
```

正規表現キャプチャグループを使用してテキストの一部を返すオプションのフィルタリングおよび変更

正規表現:

```
/.*
(
01|02
)
/
```

結果:

```
01
```

```
02
```

フィルタリングと変更 - Prometheus の例

このオプションのリストでは、次の操作を行います。

```
up{instance="demo.robustperception.io:9090",job="prometheus"} 1 1521630638000
up{instance="demo.robustperception.io:9093",job="alertmanager"} 1 1521630638000
up{instance="demo.robustperception.io:9100",job="node"} 1 1521630638000
```

次の正規表現:

```
/. *instance="
(
[^"]*
)
.*/
```

これらの結果を返します:

```
demo.robustperception.io:9090
demo.robustperception.io:9093
demo.robustperception.io:9100
```

名前付きテキストおよび値キャプチャグループを使用してフィルタリングおよび変更する

名前付きキャプチャグループを使用すると、変数クエリによって返されるオプションから「テキスト」部分と「値」部分を別々にキャプチャできます。これにより、変数ドロップダウンリストに、選択できる各値に対するわかりやすい名前を含めることができます。

例えば、node_hwmon_chip_names Prometheus メトリクスをクエリする場合、 chip_nameは chip 値よりもはるかにわかりやすくなります。したがって、次の変数クエリの結果は以下のとおりになります。

```
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_0",chip_name="enp216s0f0np0"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_1",chip_name="enp216s0f0np1"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_2",chip_name="enp216s0f0np2"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_3",chip_name="enp216s0f0np3"} 1
```

次の正規表現を渡す:

```
/chip_name="(?<text>[ ^ " ] + ) |chip=" (?<value >[ ^ " ] + )/g
```

次のドロップダウンリストを生成:

```
Display Name          Value
------------          -------------------------
enp216s0f0np0         0000:d7:00_0_0000:d8:00_0
enp216s0f0np1         0000:d7:00_0_0000:d8:00_1
enp216s0f0np2         0000:d7:00_0_0000:d8:00_2
enp216s0f0np3         0000:d7:00_0_0000:d8:00_3
```

`text` および `value` キャプチャグループ名のみがサポートされています。

変数の検査

変数ページでは、変数が他の変数またはダッシュボードで参照 (または使用) されているかどうかを簡単に識別できます。

参照または使用される変数の横には緑色のチェックマークが付き、参照されていない変数の横にはオレンジ色の注意アイコンが付きます。さらに、参照されるすべての変数には、緑色のチェックマークの横にある依存関係アイコンがあります。アイコンを選択すると、依存関係マップが表示されます。依存関係マップは移動できます。マウスホイールまたは同等の機能を使用してズームインまたはズームアウトできます。

変数構文

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルタイトルとメトリクスクエリは、2 つの異なる構文を使用して変数を参照できます。

- `$varname` – この構文は参照しやすいですが、単語の途中で変数を使用することは許可されません。

例: apps.frontend.$server.requests.count

- ${var_name} – 式の中央で変数を使用する場合は、この構文を使用します。
- ${var_name:<format>} – この形式により、Grafana が値を解釈する方法をより細かく制御できます。詳細については、このリストに続く「高度な変数フォーマットオプション」を参照してください。
- [[varname]] – 使用しないでください。この構文は古く、廃止されました。これは、今後のリリースで削除されます。

クエリがデータソースに送信される前に、クエリは[補間されます]。つまり、変数は現在の値に置き換えられます。補間中、変数値は、クエリ言語の構文と使用される場所に準拠するためにエスケープされる可能性があります。例えば、InfluxDB または Prometheus クエリの正規表現で使用される変数は、正規表現がエスケープされます。

高度な変数フォーマットオプション

変数補間のフォーマットはデータソースによって異なりますが、デフォルトの形式を変更する場合があります。

例えば、MySQL データソースのデフォルトは、'server01','server02' などのように、複数の値を引用符でカンマ区切りで結合することです。場合によっては、server01,server02 など、引用符なしでカンマ区切りの文字列を指定することもできます。これを行うには、以下にリストされている高度な変数フォーマットオプションを使用します。

一般的な構文

構文: ${var_name:option}

無効なフォーマットオプションが指定されている場合、glob がデフォルト/フォールバックオプションです。

CSV

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:csv}'
Interpolation result:  'test1,test2'
```

分散 - OpenTSDB

OpenTSDB のカスタム形式の複数の値を持つ変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:distributed}'
Interpolation result:  'test1,servers=test2'
```

二重引用符

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、\" によって各値の " をエスケープし、" で各値を引用符で囲みます。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:doublequote}'
Interpolation result:  '"test1","test2"'
```

Glob - Graphite

複数の値を持つ変数を glob にフォーマットします (Graphite クエリの場合)。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:glob}'
Interpolation result:  '{test1,test2}'
```

JSON

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:json}'
Interpolation result:  '["test1", "test2"]'
```

Lucene - Elasticsearch

Elasticsearch の Lucene 形式で複数の値を持つ変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:lucene}'
```

```
Interpolation result:  '("test1" OR "test2")'
```

Percentencode

URL パラメータで使用する単一値変数と複数値変数をフォーマットします。

```
servers = [ 'foo()bar BAZ',  'test2' ]
String to interpolate:  '${servers:percentencode}'
Interpolation result:  'foo%28%29bar%20BAZ%2Ctest2'
```

パイプ

複数の値を持つ変数をパイプ区切り文字列にフォーマットします。

```
servers = [ 'test1.',  'test2' ]
String to interpolate:  '${servers:pipe}'
Interpolation result:  'test1.|test2'
```

Raw

SQL クエリにおける単一引用符など、データソース固有のフォーマットをオフにします。

```
servers = [ 'test.1',  'test2' ]
String to interpolate:  '${var_name:raw}'
Interpolation result:  'test.1,test2'
```

[正規表現]

複数の値を持つ変数を正規表現文字列にフォーマットします。

```
servers = [ 'test1.',  'test2' ]
String to interpolate:  '${servers:regex}'
Interpolation result:  '(test1\.|test2)'
```

一重引用符

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、\' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = [ 'test1',  'test2' ]
```

```
String to interpolate:  '${servers:singlequote}'
Interpolation result:  "'test1','test2'"
```

Sqlstring

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、'' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = [ "test'1",  "test2" ]
String to interpolate:  '${servers:sqlstring}'
Interpolation result:  "'test''1','test2'"
```

[テキスト]

単一値変数と複数値変数をテキスト表現にフォーマットします。単一値の変数の場合、テキスト表現のみを返します。複数値変数の場合、テキスト表現は + と組み合わせて返されます。

```
servers = [ "test1",  "test2" ]
String to interpolate:  '${servers:text}'
Interpolation result:  "test1 + test2"
```

クエリパラメータ

単一値変数と複数値変数をクエリパラメータ表現にフォーマットします。例: `var-foo=value1&var-foo=value2`

```
servers = [ "test1",  "test2" ]
String to interpolate:  '${servers:queryparam}'
Interpolation result:  "var-servers=test1&var-servers=test2"
```

ダッシュボード使用状況の評価

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

すべてのダッシュボードとデータソースで、使用状況情報にアクセスできます。

ダッシュボードインサイト

ダッシュボードの使用状況情報を表示するには、上部のバーで [ダッシュボードインサイト] を選択します。

ダッシュボードには以下の情報が表示されます。

- [統計] – 過去 30 日間の毎日のクエリとエラーの数。
- [ユーザーとアクティビティ] – 過去 30 日間の日次ビュー数。ダッシュボードと最近のユーザー (制限は 20) に対する最後のアクティビティ。

データソースインサイト

データソースインサイトは、過去 30 日間にデータソースがどのように使用されたかに関する情報を提供します。

- 1 日あたりのクエリ
- 1 日あたりのエラー
- 1 日あたりのクエリロード時間 (平均ミリ秒)

データソースインサイトを検索するには

1. ワークスペースのメインナビゲーションで [接続] を選択します。
2. [データソース] を選択します。
3. データソースを選択します。
4. [インサイト] タブを選択します。

プレゼンスインジケータ

サインインしてダッシュボードを表示すると、プレゼンスインジケーターによって、自分と同じダッシュボードを表示しているユーザーを知ることができます。プレゼンスインジケーターには、ダッシュボードを最近操作したユーザーのアバターが表示されます。デフォルトの時間枠は 10 分です。ユーザー名を表示するには、ユーザーのアバターにカーソルを合わせます。アバターは、ユーザーの E メールに基づいて Gravatar から取得されます。

ダッシュボードにプレゼンスインジケータに収まるユーザーよりもアクティブなユーザーがいる場合は、[+X] アイコンをクリックします。これを行うと、ダッシュボードインサイトが開きます。ダッシュボードインサイトには、最近のユーザーアクティビティに関する詳細が含まれます。

インサイトデータを使用したダッシュボードのソート

検索ビューでは、インサイトデータを使用して、最も使用されているダッシュボード、壊れているダッシュボード、未使用のダッシュボードを見つけることができます。ダッシュボードは、次の項目でソートできます。

- ビュー
- エラー
- ビュー
- 作成日時
- 更新日時

ダッシュボードのトラブルシューティング

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

次の戦略を使用して、一般的なダッシュボードの問題をトラブルシューティングします。

ダッシュボードが遅い

- グラフに数十 (または数百または数千) の時系列をレンダリングしようとしていますか？ これにより、ブラウザが遅延する可能性があります。(Graphite で) `highestMax` のような関数を使用して、返されるシリーズの数を減らしてみてください。
- 場合によっては、シリーズ名が非常に大きくなることがあります。これにより、レスポンスサイズが大きくなります。`alias` を使用して、返されるシリーズ名のサイズを小さくしてみてください。

- 多くの時系列または長い時間範囲をクエリしていますか？ これらの両方の条件により、Grafana またはデータソースが大量のデータを取得し、ダッシュボードの速度が低下する可能性があります。これらの 1 つまたは両方を減らしてみてください。
- ネットワークインフラストラクチャに高負荷が発生している場合があり、特に速度が一貫していない場合は、これが問題である可能性があります。

ダッシュボードの更新レートの問題

デフォルトでは、Grafana は 30 秒ごとにデータソースをクエリします。ただし、ダッシュボードの更新率を低く設定すると、バックエンドに不要なストレスがかかります。多くの場合、データソースが 30 秒ごとに変更されるのに十分なほど頻繁にデータを送信していないため、このクエリを頻繁に行う必要はありません。

次の構成を推奨します。

- 必要に応じて、ダッシュボード、パネル、または変数の自動更新のみを有効にします。ユーザーはブラウザを手動で更新できます。
- 自動更新が必要な場合は、1 分に 1 回、10 分に 1 回、1 時間に 1 回など、更新レートをより長い期間に設定します。
- ダッシュボードの時間範囲を確認します。ダッシュボードの時間範囲が 1 週間など長い場合、自動更新は必要なく、無効にすべきです。

null データの処理またはレンダリングが間違っている、または混乱している

一部のアプリケーションは断続的にデータを公開します。例えば、イベントが発生したときにのみメトリクスを投稿します。デフォルトでは、Grafana グラフはデータポイント間で線を接続しますが、これは誤解を招く可能性があります。

[null 値の接続] オプションが [常に] に設定されているグラフは、値が欠落している線を繋げます。

これを修正する 1 つの方法は、線の代わりにバーを使用し、[値なし] オプション ([標準オプション]の下) を `0` に設定することです。この場合、欠落したデータは、データのないグラフの領域として表示されます。

Grafana バージョン 10 でのダッシュボードの検索

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボード名とパネルタイトルでダッシュボードを検索できます。ダッシュボードを検索すると、ダッシュボードの内容を表示するアクセス許可がない場合でも、システムは Grafana インスタンス内で使用可能なすべてのダッシュボードを返します。

ダッシュボード名を使用してダッシュボードを検索する

検索バーにダッシュボード名の一部を入力します。検索では、入力すると、文字列の部分一致の結果がリアルタイムで返されます。

ダッシュボード検索は次のとおりです。

- リアルタイム
- 大文字と小文字は区別されません。
- 保存済みダッシュボードとファイルベースのダッシュボード全体で機能します。

Tip

キーボードの矢印キーを使用して結果を移動し、Enter を押して選択したダッシュボードを開くことができます。

パネルタイトルを使用してダッシュボードを検索する

ダッシュボードに表示されるパネルのタイトルでダッシュボードを検索できます。パネルのタイトルが検索クエリと一致すると、ダッシュボードが検索結果に表示されます。

ダッシュボードの検索結果をタグでフィルタリングする

ダッシュボードの数が増えるほど、タグをつけることで効果的に整理することができます。ダッシュボード [設定] でタグを追加および管理できます。

複数のタグを選択すると、Grafana は選択したすべてのタグを含むダッシュボードを表示します。

タグでダッシュボード検索結果をフィルタリングするには、次のいずれかの手順を実行します。

- ダッシュボードの検索結果をタグでフィルタリングするには、検索結果の右側の列に表示されるタグを選択します。

  追加のタグを選択して、フィルタリングを続行できます。
- 使用可能なすべてのタグのリストを表示するには、[タグでフィルタリング]ドロップダウンメニューをクリックし、タグを選択します。

  すべてのタグが表示され、タグを選択すると、ダッシュボード検索が即座にフィルタリングされます。

> **Tip**
>
> キーボードのみを使用する場合は、tab キーを押して [タグでフィルタリング] ドロップダウンメニューに移動し、下矢印キーを押してメニューをアクティブ化し、タグを見つけ、Enter を押してタグを選択します。

コマンドパレット

コマンドパレットを使うと、次のことができます。

- ダッシュボードとフォルダを検索して開きます。
- ダッシュボードとアラートルールを作成します。
- Grafana 内のページを見つけます。
- ダークテーマまたはライトテーマに変更します。

コマンドパレットをトップで開き、「ctrl+k」(MacOS では「cmd+k」) と入力します。Grafana ナビゲーションバーで検索入力を選択することもできます。

> **Note**
>
> 前の手順に進むには、コマンドパレットを空にして backspace を押します。

Grafana バージョン 10 のパネルと視覚化

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルは Grafana の基本的な視覚化構築ブロックです。各パネルには、パネルで選択されたデータソース特有のクエリエディタがあります。クエリエディタを使用すると、視覚化するデータを返すクエリを構築できます。

各パネルにはさまざまなスタイルとフォーマットのオプションがあります。パネルをドラッグ、ドロップ、サイズ変更して、ダッシュボード上で配置を変更できます。

パネルを追加する前に、データソースが設定されていることを確認してください。

追加のパネルタイプは、追加のプラグインをワークスペースにインストールすることで使用できます。

特定のデータソース使用の詳細については「データソースに接続する」を参照してください。

トピック

- パネルエディタの概要
- パネル検査ビュー
- データのクエリと変換
- パネルオプションの設定
- 標準オプションを設定する
- 凡例を設定する
- データリンクの設定
- 値マッピングを設定する
- しきい値を設定する
- フィールドのオーバーライドを設定する
- Grafana バージョン 10 で利用可能な視覚化

パネルエディタの概要

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルエディタでは、データソース、クエリ、時間範囲、表示オプションなど、視覚化のすべての要素を更新できます。

パネルを新しいダッシュボードに追加するには、ダッシュボードの中央に [視覚化を追加] を選択します。既存のダッシュボードにパネルを追加するには、ダッシュボードヘッダーの [追加] を選択し、ドロップダウンで [視覚化] を選択します。同じダッシュボードまたは別のダッシュボードから既存のパネルをコピーして貼り付けることもできます。

パネルメニュー

パネルエディタにアクセスするには、任意のパネルの右上隅にカーソルを合わせます。表示されるパネルメニューアイコンを選択し、[編集] を選択します。

パネルメニューでは、次のアクションにもアクセスできます。

- [表示] – パネルを全画面で表示します。
- [編集] – パネルエディタを開いて、パネルと視覚的表現オプションを編集します。
- [共有] – パネルをリンクまたはライブラリパネルとして共有します。
- [Explore] – Explore でパネルを開き、クエリに集中できます。
- [Inspect] (検査) – Inspect ドロワーを開き、パネルデータ、統計、メタデータ、JSON、クエリを確認できます。
  - [Data] (データ) – データタブで Inspect ドロワーを開きます。
  - [クエリ] – クエリタブで Inspect ドロワーを開きます。
  - Panel JSON – JSON タブで Inspect ドロワーを開きます。
- [拡張機能] – インシデント宣言など、インストールされたアプリケーションによって提供される他のアクションにアクセスします。このオプションは、パネルメニューに拡張機能を提供するアプリケーションプラグインがインストールされている場合にのみ表示されます。

- [その他] – 他のパネルアクションにアクセスします。
  - [Duplicate] (複製) – パネルのコピーを作成します。複製されたパネルは、元のパネルとは別にデータに対してクエリを実行します。同じクエリ結果を使用する場合は、2 番目のパネルで `Dashboard` データソースを使用できます。
  - [Copy] (コピー) – パネルをクリップボードにコピーします。
  - ライブラリパネルの作成 – 他のダッシュボードにインポートできるパネルを作成します。
  - [アラートの作成] – [アラート] でアラートルール設定ページを開き、パネルクエリに基づいて Grafana マネージドアラートを作成できます。
  - [凡例を非表示] – パネル凡例を非表示にします。
  - [Get help] (ヘルプを得る) – Grafana Labs テクニカルサポートにスナップショットまたはパネルデータを送信します。
- [削除] – ダッシュボードからパネルを削除します。

パネルエディタ

このセクションでは、Grafana パネルエディタの領域について説明します。

- パネルヘッダー – ヘッダーセクションには、パネルが表示されるダッシュボードと、次のコントロールが一覧表示されます。
  - [破棄] — ダッシュボードを最後に保存してから行ったパネルへの変更が破棄されます。
  - [保存] - パネルに加えた変更を保存します。
  - [適用] – 行った変更が適用され、パネルエディタが閉じてダッシュボードに戻ります。ダッシュボードを保存して、適用された変更を保持する必要があります。
- 視覚化プレビュー – 視覚化プレビューセクションには、次のオプションが含まれています。
  - [テーブルビュー] – 視覚化をテーブルに変換して、データを表示します。テーブルビューはトラブルシューティングに役立ちます。このビューには未加工データのみが含まれます。データに適用した変換や、テーブルの視覚化で使用できる書式設定オプションは含まれません。
  - [塗りつぶし] – 視覚化プレビューで、使用可能なスペースを塗りつぶします。サイドペインの幅またはボトムペインの高さを変更すると、視覚化が変更され、使用可能なスペースが塗りつぶされます。
  - [実寸] — 視覚化プレビューが、ダッシュボードの実際のサイズで表示されます。十分なスペースがない場合、視覚化はアスペクト比が保持されたまま縮小されます。
  - [時間範囲コントロール] – [デフォルト]は、ブラウザのローカルタイムゾーンまたはより高度な設定で選択されたタイムゾーンです。

- データセクション – データセクションには、クエリの入力、データの変換、アラートルールの作成 (該当する場合) を行うタブがあります。
  - [クエリ] タブ – データソースを選択し、ここにクエリを入力します。詳細については、「データのクエリと変換」を参照してください。ダッシュボードを最初に作成すると、データソースを選択するように求められます。このタブでデータソースまたはクエリを更新できます。
  - [変換 タブ] – データ変換を適用します。詳細については、「データのクエリと変換」を参照してください。
  - [アラート] タブ – アラートルールを入力します。詳細については、「Grafana バージョン 10 のアラート」を参照してください。
- パネル表示オプション – 表示オプションセクションには、データ視覚化のほぼすべての側面を設定するタブがあります。詳細は、選択した視覚化タイプによって異なります。

パネル検査ドロワー

検査ドロワーは、パネルの理解とトラブルシューティングに役立ちます。任意のパネルの未加工データの表示、データの CSV (comma-separated values: カンマ区切り) ファイルへのエクスポート、クエリリクエストの表示、JSON 形式のパネルとデータのエクスポートが可能です。

> **Note**
>
> すべてのパネルタイプにタブがすべてあるわけではありません。例えば、ダッシュボードリストパネルには検査する未加工データがないため、[統計]、[データ]、[クエリ] タブは表示されません。

パネルインスペクターは、次のオプションで構成されます。

- パネル検査ドロワーは、右側にドロワーとして表示されます。右上隅の矢印を選択して、ドロワーペインを展開または縮小します。
- [データ タブ] – 変換が適用されたクエリによって返される未加工データを表示します。オーバーライドや値マッピングなどのフィールドオプションは、デフォルトでは適用されません。
- [統計タブ] – クエリにかかる時間および返される量を示します。
- [JSON] タブ – パネル JSON、パネルデータ JSON、およびデータフレーム構造 JSON を、表示およびコピーできます。これは Grafana のプロビジョニングまたは管理に役立ちます。
- [クエリタブ] – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。

- [エラータブ] – クエリによって返されたエラーを表示します。タブは、クエリがエラーを返す場合にのみ表示されます。

パネル検査ビュー

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネル検査ビューは、パネルメニューから開くことができ、パネルの理解とトラブルシューティングに役立ちます。Amazon Managed Grafana パネルの未加工データの検査、データの CSV (comma-separated values: カンマ区切り) ファイルへのエクスポート、クエリリクエストの表示、JSON 形式のパネルとデータのエクスポートが可能です。

Note

すべてのパネルタイプにタブがすべてあるわけではありません。例えば、ダッシュボードリストパネルには検査する未加工データがないため、[統計]、[データ]、[クエリ] タブは表示されません。

パネルインスペクターは、次のオプションで構成されます。

1. パネルインスペクターでは、ペインの上部に [Inspect:] が表示されます。右上隅の矢印を選択して、ペインを展開または縮小します。
2. [データ タブ] – 変換が適用されたクエリによって返される未加工データを表示します。オーバーライドや値マッピングなどのフィールドオプションは、デフォルトでは適用されません。
3. [統計タブ] – クエリにかかる時間および返される量を示します。
4. [JSON] タブ – パネル JSON、パネルデータ JSON、およびデータフレーム構造 JSON を、表示およびコピーできます。これは Grafana のプロビジョニングまたは管理に役立ちます。
5. クエリタブ – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。

6. [エラータブ] – エラーを表示します。クエリがエラーを返す場合にのみ表示されます。

未加工のクエリ結果のダウンロード

Amazon Managed Grafana は、お使いのデータで構成される CSV ファイルを生成します。これには、そのデータに対する任意の変換も含まれます。パネルがフィールドオプションまたはフィールドオプションオーバーライドを適用する前後にデータを表示するように選択できます。

未加工のクエリ結果をダウンロードするには

1. ダウンロードするクエリデータを含むパネルを編集します。
2. クエリエディタで、[Query Inspector] を選択します。
3. [Data] (データ) を選択します。

   パネルに複数のクエリがある、または複数のノードにクエリを実行する場合は、追加のオプションがあります。

   - [結果の選択] – 表示する結果セットデータを選択します。
   - [データを変換する]
   - [(時間で結合] – すべてのクエリの生データを一度に表示し、列ごとに結果セットを 1 つ表示します。列見出しを選択して、データ順序を変更します。
4. システムがフィールドオーバーライドを適用する前にデータを表示するには、[フォーマットされたデータ] トグルを選択します。
5. Excel 用にフォーマットされた CSV ファイルをダウンロードするには、[Excel 用にダウンロード] トグルをを選択します。
6. [CSV ダウンロード] を選択します。

クエリパフォーマンスの検査

[統計] タブには、クエリにかかる時間、送信したクエリの数、返された行数を示す統計が表示されます。この情報は、特に数値が予想外に高い場合や低い場合のクエリのトラブルシューティングに役立ちます。

クエリのパフォーマンスを検査するには

1. クエリを含むパネルを、検査するパフォーマンスで編集します。
2. クエリエディタで、[Query Inspector] を選択します。

3. [Stats] (統計) を選択します。

統計が読み取り専用形式で表示されます。

クエリリクエストとレスポンスの検査

予期しない結果を返すクエリ、または予想される結果を返さないクエリをトラブルシューティングする場合は、クエリリクエストとレスポンスデータを検査できます。

1. エクスポートするクエリを含むパネルを編集します。
2. クエリエディタで、[Query Inspector] を選択します。
3. [更新] を選択します。

   パネルにはレスポンスデータが入力されます。
4. 必要に応じて調整を行い、クエリを再実行します。
5. クエリリクエストとレスポンスデータをダウンロードするには、[Copy to clipboard] (クリップボードにコピー) アイコンをクリックし、結果を別のアプリケーションに貼り付けます。

データのクエリと変換

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Amazon Managed Grafana は、さまざまなタイプのデータソースをサポートしています。データソース[クエリ]は、Grafana が変換および視覚化できるデータを返します。各データソースは独自のクエリ言語を使用し、データソースプラグインはそれぞれクエリエディタと呼ばれるクエリ構築ユーザーインターフェイスを実装します。

トピック

- クエリについて
- クエリタブを操作する

- クエリの追加
- クエリの管理
- クエリオプション
- クエリ式の書き込み
- 別のパネルとクエリ結果を共有する
- [データを変換する]
- クエリのトラブルシューティング
- 計算タイプ

クエリについて

Grafana パネルは、クエリを介してデータソースと通信します。このクエリは、視覚化用のデータを取得します。クエリは、データソースで使用されるクエリ言語で記述された質問です。

パネルのデータソースオプションで、クエリの頻度とデータ収集の制限を設定できます。Grafana は、パネルごとに最大 26 個のクエリをサポートします。

各データソースのクエリ言語の詳細については、「データソース」セクションを参照してください。

クエリエディタ

各データソースのクエリエディタには、独自の機能を活用するクエリの作成に役立つカスタマイズされたユーザーインターフェイスが用意されています。

クエリ言語の違いにより、各データソースクエリエディタの外観と関数は異なります。データソースによっては、クエリエディタが自動補完機能、メトリクス名、変数の提案、ビジュアルクエリ構築インターフェイスを提供する場合があります。

クエリや構文に関する情報など、特定のデータソースの一意のクエリエディタ機能の詳細については、以下のデータソースドキュメントを参照してください。

- Amazon Managed Grafana に含まれるデータソースについては、「組み込みデータソース」を参照してください。
- Grafana Enterprise に含まれるデータソースについては、「Enterprise データソースに接続する」を参照してください。
- プラグインカタログによるプラグインの検索 経由でインストールする他のデータソースプラグインの場合、ドキュメントはプラグインカタログのリスト内でリンクされます。

クエリ構文

データソースは、異なるクエリ言語を使用してデータをリクエストします。特定のデータソースの一意のクエリ言語の詳細については、各ドキュメントを参照してください。

[PostgreSQL の例]:

```
SELECT hostname FROM host WHERE region IN($region)
```

PromQL の例:

```
query_result(max_over_time(<metric>[${__range_s}s]) != <state>)
```

[特別なデータソース]

Grafana には、[Grafana]、[混合]、および [ダッシュボード] の 3 つの特別なデータソースも含まれています。詳細については、「データソースに接続する」を参照してください。

クエリタブを操作する

パネルの [クエリ] タブは、次の要素で構成されます。

- [データソースセレクタ] – クエリが実行されるデータソースを選択します。
- [クエリオプション] — 最大データ取得パラメータとクエリ実行時間間隔を設定します。
- [クエリインスペクターボタン] — クエリインスペクターパネルを開き、クエリを表示および最適化できます。
- [クエリエディタリスト] – 作成したクエリを一覧表示します。
- [式] – 式ビルダーを使用してアラート式を作成します。式の詳細については、「クエリ式の書き込み」を参照してください。

クエリの追加

クエリは、Grafana がダッシュボードパネルで視覚化するデータを返します。パネルを作成すると、Grafana は自動的にデフォルトのデータソースを選択します。

クエリを追加するには

1. クエリを追加するパネルを編集します。
2. [クエリ] タブを選択します。

3. [データソース] ドロップダウンメニューを選択し、データソースを選択します。
4. [クエリオプション] を選択して、必要なデータポイントの最大数を設定します。クエリオプションの詳細については、「クエリオプション」を参照してください。
5. クエリエディタを使用してクエリを書き込みます。
6. [適用] を選択します。

Grafana はデータソースにクエリを実行し、データを視覚化します。

クエリの管理

Grafana は、クエリを折りたたみ可能なクエリ行に整理します。各クエリ行にはクエリエディタが含まれ、文字 (A、B、C など) で識別されます。

クエリを管理するには、クエリのコピー、クエリの非表示、クエリの削除、クエリの順序変更、クエリエディタのヘルプの切り替えを行うことができます。

クエリオプション

データソースセレクタの横にある [クエリオプション] を選択すると、選択したデータソースの設定が表示されます。ここで行った変更は、このパネルで行われたクエリにのみ影響します。

Grafana は、ダークグレーテキストで表示されるデフォルトを設定します。変更はホワイトテキストで表示されます。フィールドをデフォルト設定に戻すには、フィールドからホワイトテキストを削除します。

パネルデータソースのクエリオプションには以下が含まれます。

- [最大データポイント数] – データソースがサポートしている場合、返される各系列の最大データポイント数を設定します。クエリが最大データポイント設定よりも多くのデータポイントを返す場合、データソースは、平均、最大、または別の関数でデータポイントを集約することで、返されるポイント数を減らします。

  ポイントの数を制限して、クエリのパフォーマンスを向上させたり、視覚化された行を平滑にしたりできます。デフォルト値はグラフの幅 (またはピクセル数) です。グラフパネルに表示されるスペースと同じ数のデータポイントしか視覚化できないためです。

  ストリーミングデータでは、Grafana はローリングバッファの最大データポイント値を使用します。ストリーミングはデータの継続的なフローであり、バッファリングはストリームをチャンクに分割します。

- [最小間隔] – 自動計算された間隔の最小制限を設定します。これは通常、最小スクレイプ間隔です。データポイントを 15 秒ごとに保存する場合、それよりも間隔を短くしてもメリットはありません。また、スクレイプ間隔よりも低い最小値に設定して、粗く適切に機能するクエリを取得することもできます。
- [間隔] – データポイントを時間別に集計またはグループ化するときに使用できる時間範囲を設定します。

  Grafana は、テンプレートクエリで変数として使用できる適切な間隔を自動的に計算します。変数は、秒 (`$__interval`) またはミリ秒 (`$__interval_ms`) で測定されます。

  間隔は通常、合計や平均などの集計関数で使用されます。例えば、これは間隔変数 `rate(http_requests_total[$__interval])` を使用する Prometheus クエリです。

  この自動間隔は、グラフの幅に基づいて計算されます。ユーザーがグラフを縮小すると、間隔が大きくなり、より粗視化された集計になります。同様に、ユーザーが拡大すると、間隔が短くなり、より高精細な集計になります。

  詳細については、「グローバル変数」を参照してください。
- [相対時間] – 個々のパネルの相対時間範囲をオーバーライドします。これにより、パネルは、ダッシュボードの右上隅にあるダッシュボードタイムピッカーで選択されている時間とは異なります。これを使用して、同じダッシュボードに異なる期間または日数のメトリクスを表示できます。

> **Note**
>
> ダッシュボードの時間範囲が絶対時間である場合、パネルの時間オーバーライドは効果がありません。

例	相対時間フィールド
過去 5 分間	`now-5m`
今日のこれまでのところ	`now/d`
過去 5 日間	`now-5d/d`
今週始めから現在まで	`now/w`

例	相対時間フィールド
過去 2 年間	now-2y/y

- [タイムシフト] – タイムピッカーに対して開始と終了をシフトすることで、個々のパネルの時間範囲をオーバーライドします。例えば、パネルの時間範囲をダッシュボードのタイムピッカーより 2 時間早くシフトできます。

> **Note**
> ダッシュボードの時間範囲が絶対時間である場合、パネルの時間オーバーライドは効果がありません。

例	タイムシフトフィールド
過去丸 1 週間	1w/w
丸 2 週間前	2w/w
過去丸 1 か月間	1M/M
この丸 1 年	1d/y
過去丸 1 年間	1y/y

- [キャッシュタイムアウト] – (データソースで使用可能な場合のみ表示) 時系列ストアにクエリキャッシュがある場合、デフォルトのキャッシュタイムアウトをオーバーライドします。この値を秒単位で数値として指定します。

クエリ式の書き込み

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

サーバー側の式を使用すると、演算やその他のオペレーションでクエリから返されるデータを操作できます。式は新しいデータを作成し、データソースから返されるデータを操作しません。

式について

サーバー側の式を使用すると、演算やその他のオペレーションでクエリから返されるデータを操作できます。式は新しいデータを作成し、式のデータを許容できる入力にするためにマイナーなデータ再構築を行う以外は、データソースから返されるデータを操作しません。

式の使用

式は、Grafana アラートで最も一般的に使用されます。処理はサーバー側で行われるため、式はブラウザセッションなしで動作できます。ただし、式はバックエンドデータソースや視覚化でも使用できます。

> **Note**
>
> 式は、レガシーダッシュボードアラートでは機能しません。

式は、異なるデータソースからのクエリを組み合わせて有効にするか、データソースで使用できない演算を提供することで、データソースを補強することを目的としています。

> **Note**
>
> 可能であれば、データソース内でデータ処理を行う必要があります。ストレージから Grafana サーバーにデータをコピーして処理することは非効率であるため、式は軽量データ処理を対象としています。

式は、時系列または数値データを返すデータソースクエリで機能します。また、多次元データでも動作します。例えば、複数の系列を返すクエリでは、各系列がラベルまたはタグで識別されます。

個々の式は 1 つ以上のクエリまたはその他の式を入力として受け取り、結果にデータを追加します。個々の式またはクエリは、RefID と呼ばれる名前付き識別子 (例: デフォルト文字 A または B) である変数で表されます。

個々の式または別の式にあるデータソースクエリの出力を参照するには、この識別子を変数として使用します。

式のタイプ

式は 2 種類のデータで動作します。

- 時系列のコレクション。
- 数値のコレクション (各数値は項目)。

各コレクションは、単一のデータソースクエリまたは式から返され、RefID で表されます。各コレクションはセットであり、セット内の各項目は、ラベルまたはキーと値のペアとして保存されるディメンションによって一意に識別されます。

データソースクエリ

サーバー側の式は、バックエンドデータソースのデータソースクエリのみをサポートします。データは一般に、時系列データにラベル付けされているものと見なされます。

データソースクエリは、式とともに使用されると、式エンジンによって実行されます。これを行うと、データはデータフレームごとに 1 つの時系列または 1 つの数値に再構成されます。例えば、テーブルビューの 1 つのフレームで複数の系列を返すデータソースを使用する場合、式で実行したときに異なるように見えることがあります。

現在、データフレームを使用する場合にサポートされる時系列以外の形式 (数値) は、時間、文字列の列、1 つの数値列のないデータフレームを返すテーブルレスポンスのみです。

次の例のテーブルは、式で動作する数値を生成します。文字列の列はラベルになり、数値列は対応する値になります。例えば、`{"Loc": "MIA", "Host": "A"}` の値は 1 です。

Loc	ホスト	Avg_CPU
MIA	A	1
NYC	B	2

オペレーション

式には math、reduce、リサンプリング演算を使用できます。

数値演算

Math は、時系列または数値データに関する自由形式の数式です。Math 演算子では、数値と時系列を入力として受け取り、異なる数値と時系列に変更します。

他のクエリまたは式からのデータは、$A などのドル記号のプレフィックスが付いた RefID で参照されます。変数の名前にスペースがある場合は、${my variable} のような中括弧構文を使用できます。

数値定数は、10 進数 (2.24)、8 進数 (072 のような先頭がゼロ)、または 16 進数 (0x2A のような先頭が 0x) にすることができます。指数と記号もサポートされています (例: -0.8e-2) 。

演算子

算術演算子 (+、二項および単項 -、*、/、%、指数 **)、リレーショナル演算子 (<、>、==、!=、>=、<=)、論理演算子 (&&、||、および単項 !) がサポートされています。

データに対する演算処理は、数値データか時系列データかによって異なります。

$A + $B や $A || $B などの二項演算では、データのタイプに応じて、演算子は次の方法で適用されます。

- $A と $B の両方が数値の場合、2 つの数値の間で演算が実行されます。
- 1 つの変数が数値で、もう 1 つの変数が時系列の場合、時系列の各ポイントの値と数値との間の演算が実行されます。
- $A と $B の両方が時系列データである場合、2 つの系列の各値間の演算は、$A と $B の両方に存在するタイムスタンプごとに実行されます。Resample 演算を使用してタイムスタンプを並べることができます。

まとめ

- Number <Operation> number = 数値
- Number <Operation> series = 系列
- Series <Operation> series = 系列

式は 1 つの変数で表される複数の系列または数値で動作するため、二項演算は 2 つの変数間の統合 (結合) も実行します。これは、個々の系列または数値に関連付けられた識別ラベルに基づいて行われます。

したがって、$A にある {host=web01} のようなラベルを持つ数値と、同じラベルを持つ $B に別の数値がある場合、各変数内のこれら 2 つの項目間で演算が実行され、結果は同じラベルを共有します。この統合の動作のルールは次のとおりです。

- ラベルのない項目は、すべてに結合されます。
- $A と $B の両方に 1 つの項目 (1 つの系列、または 1 つの数値) しか含まれていない場合、それらは結合されます。
- ラベルが完全に一致する場合、それらは結合されます。
- ラベルが他のサブセットである場合、例えば、$A の項目には {host=A,dc=MIA} ラベルが付けられ、$B の項目には {host=A} ラベルが付けられると結合されます。
- 項目ごとに異なるタグキーがある $A などの変数内では、結合動作は不定です。

リレーショナル演算子と論理演算子は、偽の場合は 0、真の場合は 1 を返します。

数値演算関数

ほとんどの関数は独自の式の演算にありますが、数値演算関数には、数学演算子や数学記号に似た関数がいくつかあります。関数が数値または系列のいずれかを取ることができる場合、引数と同じタイプが返されます。系列の場合、系列内の各ポイントの値に対して演算が実行されます。

abs

abs は、引数の絶対値を返します。これは、数値または系列です。例えば、abs(-1)、abs($A) などです。

is_inf

is_inf は数値または系列を取得し、Inf 値 (負または正) の 1 と他の値の 0 を返します。例えば、is_inf($A) などです。

> **Note**
>
> 例えば、負の無限大を具体的にチェックする必要がある場合は、$A == infn() のような比較を行うことができます。

is_nan

is_nan は数値または系列を取得し、NaN 値の 1 と他の値の 0 を返します。例えば、is_nan($A) などです。NaN は NaN と等しくないため、このチェックにはこの関数が必要です。

[is_null]

is_null は数値または系列を取得し、null 値の 1 と他の値の 0 を返します。例えば、is_null($A) などです。

is_number

is_number は数値または系列を取得し、すべての実数値の 1 と他の値 (null、Inf+、Inf-、および NaN) の 0 を返します。例えば、is_number($A) などです。

log

Log は、引数の自然対数を返します。これは、数値または系列です。値が 0 未満の場合、NaN が返されます。例えば、log(-1)、log($A) などです。

[inf、infn、nan、null]

inf、infn、nan、null 関数はすべて、名前の単一の値を返します。これらは主にテスト用です。例えば、null() などです。

round

Round は、丸められた整数値を返します。例えば、round(3.123)、round($A) です。

ceil

Ceil は、数値を最も近い整数値に切り上げます。例えば、ceil(3.123) は 4 を返します。

floor

Floor は、数値を最も近い整数値に切り下げます。例えば、floor(3.123 は 3 を返します。

[削減]

Reduce は、クエリまたは式から返される 1 つ以上の時系列を取得し、各系列を 1 つの数値に変換します。時系列のラベルは、それぞれ出力される削減された数値のラベルとして保持されます。

[フィールド]

- [関数] – 使用する削減関数

- [入力] – リサンプリングする変数 (refID (A など))
- [モード] – 系列に数値以外の値が含まれている場合、削減関数の制御動作を許可します (null、NaN 、+-Inf)

削減関数

Count

Count は、各系列内のポイント数を返します。

Mean

Mean は、各系列のすべての値の合計を、その系列のポイント数で割った値を返します。strict モードでは、系列内の値が null または nan の場合、または系列が空の場合、NaN が返されます。

[最小 と 最大]

最小 と 最大は、系列内の最小値または最大値をそれぞれ返します。strict モードでは、系列内の値が null または nan の場合、または系列が空の場合、NaN が返されます。

Sum

Sum は、系列のすべての値の合計を返します。系列の長さがゼロの場合、合計は 0 になります。strict モードでは、系列内の値に NaN または Null がある場合、NaN が返されます。

Last

Last は、系列内の最後の数値を返します。系列に値がない場合は、NaN を返します。

制限モード

Strict

Strict モードでは、入力系列はそのまま処理されます。系列内の値が数値以外の場合 (null、NaN、または +-Inf)、NaN が返されます。

Drop Non-Numeric

このモードでは、入力系列内のすべての非数値 (null、NaN、または +-Inf) は、制限関数を実行する前に除外されます。

Replace Non-Numeric

このモードでは、数値以外の値はすべて事前定義された値に置き換えられます。

[リサンプリング]

リサンプリングでは、各時系列のタイムスタンプが一貫した時間間隔になるように変更されます。主なユースケースは、同じタイムスタンプを共有しない時系列をリサンプリングして、それらの間で演算を実行できるようにすることです。これは、2 つの系列をそれぞれリサンプリングし、リサンプリングされた変数を参照する Math 演算で行うことができます。

[フィールド]

- [入力] – リサンプリングする時系列データの変数 (refID (`A` など))
- [リサンプリング]先 — リサンプリング先の期間。例: `10s`。単位は、`s` 秒、`m` 分、`h` 時間、`d` 日、`w` 週、`y` 年です。
- [ダウンサンプリング] – ウィンドウサンプルごとに複数のデータポイントがある場合に使用する制限関数。動作の詳細については、「制限関数」を参照してください。
- [アップサンプリング] – データポイントがないウィンドウサンプルを埋めるために使用するメソッド。
  - pad で最後の既知の値に置換
  - backfill で次の既知の値に置換
  - fillna で空のサンプルウィンドウを NaN に置換

式の書き込み

データソースがサポートしている場合、Grafana は [式] ボタンを表示し、クエリエディタリストに既存の式を表示します。

式を記述するには

1. パネルを開きます。
2. クエリの下で、[式] を選択します。
3. [演算] フィールドで、書き込む式のタイプを選択します。
4. 式を書き込みます。
5. [適用] を選択します。

特殊なケース

クエリされたデータソースが系列や数値を返さない場合、式エンジンは NoData を返します。例えば、式によってマージされた 2 つのデータソースクエリがリクエストに含まれている場合、NoData が少なくとも 1 つのデータソースクエリによって返された場合、クエリ全体で返される結果は NoData になります。Grafana アラートが NoData 結果を処理する方法の詳細については、「Grafana マネージドアラートルールを設定する」を参照してください。

複数のクエリで式を使用する場合、式エンジンはすべてのクエリが同一のタイムスタンプを返す必要があります。例えば、演算を使用して、それぞれが SELECT NOW() AS "time" を使用する複数の SQL クエリの結果を組み合わせる場合、式はすべてのクエリが NOW() を同一のタイムスタンプとして評価する場合にのみ機能します。これは必ずしも発生するわけではありません。これを解決するには、NOW() を SELECT 1 AS "time" などの任意の時刻、またはその他の有効な UNIX タイムスタンプに置き換えることができます。

別のパネルとクエリ結果を共有する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、ダッシュボード内にある他の任意のパネルに対して、1 つのパネルからのクエリ結果を使用できます。パネル間でクエリ結果を共有すると、データソースに対して実行されるクエリの数が減少し、ダッシュボードのパフォーマンスが向上します。

ダッシュボードのデータソースを使用すると、結果を共有するクエリを含むダッシュボード内のパネルを選択できます。パネルごとに個別のクエリを送信する代わりに、Grafana は 1 つのクエリを送信し、他のパネルはクエリ結果を使用して視覚化を構築します。

この戦略は、例えば、同じデータを視覚化する複数のパネルがある場合に実行されるクエリの数を大幅に減らすことができます。

クエリ結果を共有するには

1. ダッシュボードを作成します。

2. タイトルを「Source panel」に変更します。このパネルを他のパネルのソースとして使用します。
3. 共有するクエリを (複数) 定義します。

   利用可能なデータソースがない場合は、Grafana データソースを使用します。このデータソースは、テストに使用できるランダムな時系列を返します。
4. 2 番目のパネルを追加し、クエリエディタで [ダッシュボード] データソースを選択します。
5. [パネルリストから結果を使用する] で、最初に作成したパネルを選択します。

ソースパネルで定義されたすべてのクエリが新しいパネルで使用できるようになりました。ソースパネルで行われたクエリは、複数のパネルと共有できます。

任意のクエリをクリックして、そのクエリが定義されたパネルに移動できます。

[データを変換する]

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

変換は、システムが視覚化を適用する前に、クエリによって返されるデータを操作できる強力な方法です。変換を使用すると、次のことができます。

- フィールド名の変更
- 時系列データの結合
- クエリ間で数学演算の実行
- ある変換の出力を別の変換への入力として使用

同じデータセットの複数のビューに依存するユーザーにとって、変換は多数のダッシュボードを作成および維持する効率的な方法を提供します。

また、ある変換の出力を別の変換への入力として使用して、パフォーマンスを向上させることもできます。

> **Note**
>
> システムが変換されたデータをグラフ化できない場合があります。その場合は、視覚化の上にある [テーブルビュー] のトグルをクリックして、データのテーブルビューに切り替えます。これは、変換の最終結果を理解するのに役立ちます。

変換タイプ

Grafana には、データを変換するさまざまな方法があります。変換関数の完全なリストを以下に示します。

変換の順序

複数の変換がある場合、Grafana はリスト順で変換を適用します。各変換では、処理中のパイプラインにある次の変換に渡される新しい結果セットが作成されます。

Grafana が変換を適用する順序は、結果に直接影響します。例えば、縮小変換を使用して 1 つの列のすべての結果を 1 つの値に集約する場合、変換はその 1 つの値にのみ適用できます。

変換関数をデータに追加する

次の手順は、データに変換を追加する方法を示しています。このドキュメントには、変換のタイプごとの手順は含まれていません。

パネルに変換を追加するには

1. 1 つ以上の変換を追加するパネルに移動します。
2. パネルのいずれかの部分にカーソルを合わせると、右上隅にアクションメニューが表示されます。
3. アクションメニューで、[編集] を選択します。
4. [変換] タブを選択します。
5. 変換を選択します。変換オプションを設定する変換行が表示されます。
6. [変換ルールの追加] を選択すると、別の変換ルールを追加できます。この変換は、前の変換が返した結果セットに作用します。

変換をデバッグする

変換の入力と出力の結果セットを表示するには、変換行の右側にあるデバッグ (バグ) アイコンを選択します。これにより、入力データと変換の結果が出力として表示されます。

入出力の結果セットにより、変換のデバッグが可能になります。

変換を無効にする

変換行の右上にある表示 (目) アイコンを選択すると、変換を無効化または非表示にできます。これにより、その特定の変換に適用されたアクションが無効になり、複数の変換を次々に変更するときに問題を特定するのに役立ちます。

変換をフィルタリングする

変換で複数のクエリを使用している場合、これらをフィルタリングし、選択した変換を 1 つのクエリにのみ適用できます。これを行うには、変換行の右上にあるフィルターアイコンを選択します。これにより、パネルで使用されるクエリのリストを含むドロップダウンが開きます。ここから、変換するクエリを選択できます。

(例を含む) 注釈でフィルタリングして、変換を適用することもできます。これを行うと、フィールドのリストが変更され、注釈または例のツールチップに反映されます。

フィルターアイコンは、パネルに複数のクエリまたはデータソース (パネルまたは注釈データ) がある場合に常に表示されますが、クエリの出力をマージするための以前の変換が適用されている場合には機能しない場合があります。これは、1 つの変換が前の変換の出力を取得するためです。

変換を削除する

不要な変換を削除することをお勧めします。変換を削除すると、視覚化からデータが削除されます。

前提条件:

変換に依存するすべてのダッシュボードを特定し、影響を受けるダッシュボードのユーザーに通知します。

変換を削除するには

1. パネルを開いて編集します。
2. [変換] タブを選択します。
3. 削除する変換の横にあるごみ箱アイコンを選択します。

変換関数

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

データに関する次の変換を実行できます。

計算からのフィールドの追加

この変換を使用して、他の 2 つのフィールドから計算した新しいフィールドを追加します。各変換では 1 つの新しいフィールドを追加できます。

- [モード] – 以下のモードを選択します。
  - [行を減らす] – 選択したフィールドの各行に、選択済みの計算を個別に適用します。
  - [二項演算] – 選択した 2 つのフィールドから 1 行の値に基本的な二項演算 (合計や乗算など) を適用します。
  - [単項演算] – 選択したフィールドの 1 行の値に基本的な単項演算を適用します。使用できるオプションは、以下のとおりです。
    - [絶対値 (abs)] - 指定された式の絶対値を返します。これは、ゼロからの距離を正の数として表します。
    - [Natural exponential (exp)] (自然指数) - 特定の式の冪 e を返します。
    - [Natural logarithm (ln)] (自然対数) - 指定された式の自然対数を返します。
    - [Floor (floor)] (床関数) – 特定の式以下の最大整数を返します。
    - [Ceiling (ceil)] (天井関数) - 指定された式以上の最小整数を返します。
  - [累積関数] – 現在の行とそれ以前のすべての行に関数を適用します。
    - [Total] – 現在の行までの累積合計を計算します。
    - [Mean] – 現在の行までの平均を計算します。
  - [ウィンドウ関数] – ウィンドウ関数を適用します。ウィンドウは、[trailing] (末尾) または [centered] (中央) のいずれかになります。末尾のウィンドウでは、現在の行がウィンドウの最後

の行になります。中央のウィンドウでは、ウィンドウは現在の行を中心とします。均等なウィンドウサイズの場合、ウィンドウは現在の行と前の行の中心になります。

- [Mean] - 移動平均または実行平均を計算します。
- – 移動標準偏差を計算します。
- [Variance] – 移動分散を計算します。

- [行インデックス] – 行インデックスを含むフィールドを挿入します。

- [Field name] – 新しいフィールドの計算に使用する、フィールドの名前を選択します。
- [計算] – [掃き出し法] を選択すると、[計算] フィールドが表示されます。フィールドを選択すると、新しいフィールドの作成に使用できる計算の選択肢が一覧で表示されます。使用可能な計算の詳細については、「計算タイプ」を参照してください。
- [演算] – [二項演算] または [単項演算] モードを選択すると、[演算] フィールドが表示されます。これらのフィールドを使用すると、2 つの選択したフィールドから 1 行の値に対して基本的な数学演算を実行できます。二項演算に数値を使用することもできます。
- [As percentile] – [行インデックス] モードを選択すると、パーセンタイルとしてスイッチが表示されます。このスイッチを使用すると、行インデックスを行の合計数に対する割合で変換できます。
- [エイリアス] – (オプション) 新しいフィールドの名前を入力します。このフィールドを空白のままにすると、計算に一致するようにフィールドの名前が付けられます。
- [すべてのフィールドを置き換え] – (オプション) 他のすべてのフィールドを非表示にし、計算したフィールドのみを視覚化に表示する場合は、このオプションを選択します。

> **Note**
>
> [累積関数]と[ウィンドウ関数]は、パブリックプレビューでは最新です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

連結フィールド

[連結フィールド]

この変換を使用して、全フレームの全フィールドを 1 つの結果に結合します。

例えば、温度とアップタイムデータ (クエリ A) と空気品質インデックスとエラー情報 (クエリ b) を取得する個別のクエリがある場合、連結変換を適用すると、すべての関連情報が 1 つのビューに統合されたデータフレームが生成されます。

次の 2 つについて考えてみましょう。

クエリ A

温度	アップタイム
15.4	1230233

クエリ B

AQI	エラー
3.2	5

フィールドを連結すると、データフレームは次のようになります。

温度	アップタイム	AQI	エラー
15.4	1230233	3.2	5

この変換により、さまざまなソースからのデータをマージするプロセスが簡素化され、分析と視覚化の包括的なビューが提供されます。

クエリ結果から設定

[クエリ結果から設定]

この変換を使用して、1 つのクエリを選択し、[最小] 、[最大] 、[単位]、[しきい値]などの標準オプションを抽出し、他のクエリ結果に適用します。これにより、特定のクエリによって返されるデータに基づいて、動的な視覚化設定が可能になります。

オプション

- [Config query] (設定クエリ) – 設定として使用するデータを返すクエリを選択します。
- [適用先] – 設定を適用するフィールドまたは系列を選択します。
- [オプションに適用] – フィールドタイプを指定するか、[適用先] の選択に応じてフィールド名の正規表現を使用します。

フィールドマッピングテーブル

設定オプションの下に、フィールドマッピングテーブルがあります。このテーブルには、設定クエリによって返されるデータに含まれるすべてのフィールドと、[用途] および [選択] オプションが一覧表示されます。設定プロパティへのマッピングフィールドを制御でき、複数の行に対して選択する値を選べます。

次の例は、入力クエリとフィールド設定として使用されるクエリを示しています。

入力クエリ

時間	値
1626178119127	10
1626178119129	30

設定クエリ

時間	値
1626178119127	100
1626178119129	100

出力クエリ (入力と同じですが、値フィールドに 設定が追加されました)

時間	値 (設定: Max=100)
1626178119127	10
1626178119129	30

各フィールドで最大設定オプションが設定されるようになりました。[最小値]、[最大値]、[単位]、[しきい値] などのオプションは、フィールド設定の一部です。設定すると、パネル工ディタのオプションペインで手動で設定されたオプションではなく、視覚化によって使用されます。

[値のマッピング]

クエリ結果を値のマッピングに変換することもできます。このオプションでは、設定クエリ結果のすべての行で 1 つの値のマッピング行を定義します。次の例を参照してください。

設定クエリ結果

値	テキスト	色
L	低	青
M	中	緑色
H	高	赤

フィールドマッピングで、以下を指定します。

フィールド	Use as	Select
値	値マッピング / Value	[すべての値]
テキスト	値マッピング / Text	[すべての値]
色	値マッピング / Color	[すべての値]

Grafana はクエリ結果から値マッピングを構築し、実際のデータクエリ結果に適用します。設定クエリの結果に従って、値がマッピングされ、色付けされているはずです。

フィールドタイプの変換

この変換を使用して、指定されたフィールドのフィールドタイプを変更します。

この変換には、次のオプションがあります。

- [Field] – 使用可能なフィールドから選択します。
- [as] – 変換するフィールドタイプを選択します。
  - [Numeric] – 値の数値の作成を試みます。
  - [String] – 値の文字列を作成します。
  - [時間] – 値を時間として解析しようとします。

    - DateFormat を `yyyy-mm-dd` や `DD MM YYYY hh:mm:ss` などの文字列の入力として指定するオプションを表示します。
- [Boolean] – 値はブール値になります。
- [Enum] – 値は列挙型になります。
    - 列挙を管理するテーブルを表示します。
- [Other] – 値を json として解析しようとします。

例えば、時間フィールドを [時間] として、日付形式を YYYY として選択することで変更できる次のクエリを考えてみましょう。

時間	Mark	値
2017-07-01	above	25
2018-08-02	below	22
2019-09-02	below	29
2020-10-04	above	22

結果

時間	Mark	値
2017-01-01 00:00:00	above	25
2018-01-01 00:00:00	below	22
2019-01-01 00:00:00	below	29
2020-01-01 00:00:00	above	22

この変換により、データタイプを柔軟に適応させ、視覚化の互換性と一貫性を確保できます。

Extract Fields (フィールドの抽出)

この変換を使用して、データソースを選択し、さまざまな形式でそこからコンテンツを抽出します。この変換には次のフィールドがあります。

- [ソース] – データソースのフィールドを選択します。
- [形式] – 次のいずれかを選択します。
  - [JSON] – ソースから JSON コンテンツを解析します。
  - [キーと値のペア] – ソースからの形式 a=b または c:d にあるコンテンツを解析します。
  - [自動] – フィールドを自動的に検出します。
- [すべてのフィールドを置き換え] – (オプション) 他のすべてのフィールドを非表示にし、計算したフィールドのみを視覚化に表示する場合は、このオプションを選択します。
- [時間保持] - (オプション) [すべてのフィールドを置き換え] が真の場合にのみ使用できます。出力の時間フィールドを保持します。

以下のデータセットを考えてみましょう。

データセットの例

Timestamp	json_data
1636678740000000000	{"value": 1}
1636678680000000000	{"value": 5}
1636678620000000000	{"value": 12}

この設定を使用して、時系列パネルで使用されるデータを準備できます。

- Source: json_data
- 形式: JSON
  - Field: value
  - Alias: my_value
- すべてのフィールドを置き換え: 真
- 時間保持: 真

これにより、次の出力が生成されます。

[変換されたデータ]

Timestamp	my_value
1636678740000000000	1
1636678680000000000	5
1636678620000000000	12

この変換では、さまざまな方法でデータを抽出、フォーマットできます。特定のデータニーズに基づいて抽出形式をカスタマイズできます。

リソースからフィールドを検索

この変換を使用して、外部ソースから追加のフィールドを検索してフィールド値を充実させます。

この変換には次のフィールドがあります。

- [Field] (フィールド) – データセットからテキストフィールドを選択します。
- [ルックアップ] - [Countries] (国)、[USA States] (米国の州)、および [Airports] (空港) から選択します。

> **Note**
>
> この変換は空間データのみをサポートします。

例えば、以下のデータがある場合を考えてみましょう。

データセットの例

場所	値
AL	0
AK	10

場所	値
Arizona	5
Arkansas	1
Somewhere	5

この設定では、次のようになります。

- Field: location
- Lookup: USA States

この場合、次の出力が返されます。

[変換されたデータ]

場所	ID	名前	Lng	Lat	値
AL	AL	Alabama	-80.891064	12.448457	0
AK	AK	Arkansas	-100.891064	24.448457	10
Arizona					5
Arkansas					1
Somewhere					5

この変換により、外部ソースから追加情報を取得してデータを拡張し、分析と視覚化のためのより包括的なデータセットを提供できます。

クエリ refId によるデータのフィルタリング

この変換を使用して、複数のクエリがあるパネル内の 1 つ以上のクエリを非表示にします。

Grafana では、クエリ識別文字がダークグレーテキストで表示されます。フィルタリングを切り替えるには、クエリ識別子を選択します。クエリ文字が白い場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示です。

> **Note**
>
> この変換は Graphite では使用できません。このデータソースは、返されたデータとクエリの相関関係をサポートしていないためです。

値によるデータのフィルタリング

この変換を使用して、視覚化内で直接データポイントを選択的にフィルタリングします。この変換では、選択したフィールドに適用された 1 つ以上の条件に基づいて、データを含める、または除外するためのオプションを提供します。

この変換は、データソースが値によってネイティブにフィルタリングしない場合に非常に便利です。共有クエリを使用している場合、これを使用して値を絞り込んで表示することもできます。

すべてのフィールドで使用できる条件は次のとおりです。

- [Regex] – 正規表現を照合します。
- [Is Null] – 値が null の場合に照合。
- [Is Not Null] – 値が null でない場合に照合します。
- Equal – 値が指定値と等しい場合に照合します。
- Different – 値が指定された値と異なる場合に照合します。

数値フィールドで使用可能なその他の条件は次のとおりです。

- Greater – 値が指定値より大きい場合に照合します。
- Lower – 値が指定値よりも小さい場合に照合します。
- Greater or equal – 値が指定値以上の場合に照合します。
- Lower or equal – 値が指定値以下の場合に照合します。
- Range – 指定された最小値と最大値 (最小値と最大値を含む) 間の範囲を照合します。

以下のデータセットを考えてみましょう。

時間	温度	高度
2020-07-07 11:34:23	32	101

時間	温度	高度
2020-07-07 11:34:22	28	125
2020-07-07 11:34:21	26	110
2020-07-07 11:34:20	23	98
2020-07-07 10:32:24	31	95
2020-07-07 10:31:22	20	85
2020-07-07 09:30:57	19	101

温度が 30°C 未満のデータポイントを [含める] にすると、設定は次のようになります。

- フィルタータイプ: [含める]
- Condition:「温度」が [Lower Than] (未満)、「30」と一致する行

30°C 未満の温度のみが含まれる場合、次の結果が得られます。

[変換されたデータ]

時間	温度	高度
2020-07-07 11:34:22	28	125
2020-07-07 11:34:21	26	110
2020-07-07 11:34:20	23	98
2020-07-07 10:31:22	20	85
2020-07-07 09:30:57	19	101

複数の条件をフィルターに追加できます。例えば、高度が 100 を超える場合にのみデータを含めることができます。これを行うには、次の設定にその条件を追加します。

- フィルタータイプ: 条件に [すべて一致] する [含める] 行

- Condition 1:「温度」が [Lower Than] (未満)、「30」と一致する行
- Condition 2:「高度」が [Greater] (超え)、「100」と一致する行

複数の条件がある場合、条件にすべて一致、または追加した条件のいずれかに一致する行にアクション (包含/除外) を適用するかを選択できます。

上記の例では、温度が 30°C 未満かつ高度が 100 を超える行を含めるため、[すべて一致] を選択しました。温度が 30°C 未満または高度が 100 を超える行を含める場合は、[任意の一致] を選択します。これには、元のデータの最初の行が含まれます。元のデータの温度は 32°C (最初の条件と一致しない) ですが、高度は 101 (2 番目の条件と一致する) であるため、含まれます。

無効または不完全な設定条件は無視されます。

この汎用的なデータフィルタリング変換により、特定の条件に基づいてデータポイントを選択的に含めたり除外したりできます。条件をカスタマイズしてデータプレゼンテーションを調整し、独自の分析ニーズを満たすことが可能です。

Filter fields by name (名前によるフィールドのフィルタリング)

この変換を使用して、クエリ結果の一部を削除します。フィールド名をフィルタリングする方法は 3 つあります。

- 正規表現を入力します。
- 含まれるフィールドを手動で選択します。
- ダッシュボード変数を使用します。

正規表現を使用

正規表現を使用してフィルタリングすると、正規表現に一致するフィールド名が含まれます。例えば、正規表現 `'prod.*'` を使用すると、`prod` で始まるフィールドのみが返されます。

正規表現には、`${variableName}` 構文を使用して補間されたダッシュボード変数を含めることができます。

含まれるフィールドを手動で選択

フィールド名を選択または選択解除して、結果から削除します。正規表現も含まれている場合、チェックが入っていない場合でも、式に一致するフィールドが含まれます。

ダッシュボード変数を使用

[変数から] を選択すると、フィールドを含めるために使用されるダッシュボード変数を選択できます。複数の選択肢を持つダッシュボード変数を設定することで、同じフィールドを複数の視覚化にわたって表示できます。

この変換により、効果的な分析と視覚化に必要な特定のフィールドに焦点を当てて、クエリ結果を柔軟に調整できます。

文字列の形式を設定する

この変換を使用して、文字列フィールドの出力をカスタマイズします。この変換には次のフィールドがあります。

- 大文字 – 文字列全体を大文字形式にします。
- 小文字 – 文字列全体を小文字形式にします。
- センテンスケース – 文字列の最初の文字を大文字の形式にします。
- タイトルケース – 文字列内の各単語の最初の文字を大文字の形式にします。
- パスカルケース – 文字列内の各単語の最初の文字を大文字の形式にし、単語間のスペースは含まれません。
- キャメルケース – 最初の単語を除き、文字列内の各単語の最初の文字を大文字の形式にし、単語間のスペースは含まれません。
- スネークケース – 文字列内のすべての文字を小文字の形式にし、単語間のスペースの代わりにアンダースコアを使用します。
- ケバブケース – 文字列内のすべての文字を小文字の形式にし、単語間のスペースの代わりにダッシュを使用します。
- Trim – 文字列から先頭と末尾のすべてのスペースを削除します。
- 文字列の単位 – 指定された開始位置と終了位置を使用して、文字列のサブ文字列を返します。

この変換により、視覚化と分析を向上させるために文字列データのプレゼンテーションを標準化して調整する便利な方法が提供されます。

> **Note**
>
> この変換は現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

時間形式

この変換を使用して、時間フィールドの出力をカスタマイズします。出力は、Moment.js 形式の文字列を使用してフォーマットできます。例えば、時間フィールドの年のみを表示する場合は、形式文字列「YYYY」を使用して暦年 (1999 年や 2012 年など) を表示できます。

[変換前]

Timestamp	イベント
1636678740000000000	システムスタート
1636678680000000000	ユーザーログイン
1636678620000000000	データ更新

「YYYY-MM-DD HH:mm:ss」を適用した後

Timestamp	イベント
2021-11-12 14:25:40	システムスタート
2021-11-12 14:24:40	ユーザーログイン
2021-11-12 14:23:40	データ更新

この変換により、視覚化の時間表現を調整し、時間データを表示する柔軟性と精度を提供できます。

> **Note**
> この変換は、アルファ機能として Grafana v10 と互換性のあるワークスペースで使用できます。

グループ化の条件

この変換では、指定されたフィールド (列) 値でデータをグループ化し、各グループの計算を処理します。計算選択肢のリストの表示を選択します。

オリジナルデータの例を次に示します。

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン
7/7/2020 午前 11 時 34 分 20 秒	サーバー 3	62	OK
7/7/2020 午前 10 時 32 分 20 秒	サーバー 2	90	オーバーロード
7/7/2020 午前 10 時 31 分 22 秒	サーバー 3	55	OK
7/7/2020 午前 9 時 30 分 57 秒	サーバー 3	62	再起動
7/7/2020 午前 9 時 30 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 28 分 6 秒	サーバー 1	80	OK
7/7/2020 午前 9 時 25 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK

この変換では 2 つのステップを実行します。まず、データをグループ化させる 1 つ以上のフィールドを指定します。これにより、これらのフィールドの同じ値すべてが、ソートされたかのようにグループ化されます。例えば、サーバー ID フィールドでグループ化すると、次のようにデータがグループ化されます。

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン
7/7/2020 午前 9 時 28 分 6 秒	サーバー 1	80	OK
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK
7/7/2020 午前 10 時 32 分 20 秒	サーバー 2	90	オーバーロード
7/7/2020 午前 9 時 30 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 25 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 11 時 34 分 20 秒	サーバー 3	62	OK
7/7/2020 午前 10 時 31 分 22 秒	サーバー 3	55	OK
7/7/2020 午前 9 時 30 分 57 秒	サーバー 3	62	再起動

同じ値のサーバー ID を持つすべての行がグループ化されます。

データをグループ化するフィールドを選択すると、他のフィールドにさまざまな計算を追加でき、その計算を行の各グループに適用できます。例えば、これらの各サーバーの平均 CPU 温度を計算できます。そのため、CPU 温度フィールドに適用される[平均]計算を追加して、以下を取得できます。

サーバー ID	CPU 温度 (平均)
サーバー 1	82

サーバー ID	CPU 温度 (平均)
サーバー 2	88.6
サーバー 3	59.6

また、複数の計算を追加できます。例:

- 時間のフィールドでは、[最後]の値を計算して、各サーバーで最後のデータポイントがいつ受信されたかを把握できます。
- サーバーステータスフィールドでは、[最後]の値を計算して、サーバーごとに最後の状態値を確認できます。
- また、温度フィールドでは、最後の値を計算して、サーバーごとに最新の測定温度を知ることもできます。

その後、以下を取得します。

サーバー ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最後)	サーバーステータス (最後)
サーバー 1	82	80	7/7/2020 午前 11 時 34 分 20 秒	シャットダウン
サーバー 2	88.6	90	7/7/2020 午前 10 時 32 分 20 秒	オーバーロード
サーバー 3	59.6	62	7/7/2020 午前 11 時 34 分 20 秒	OK

この変換により、時系列からキー情報を抽出して便利な方法で表示できます。

Grouping to matrix (マトリックスへのグループ化)

この変換を使用して、クエリ出力の [列]、[行]、[セル値] フィールドの入力として使用される 3 つのフィールドを組み合わせて、マトリックスを生成します。以下のようにマトリックスが計算されます。

元のデータ

サーバー ID	CPU 温度	サーバーステータス
サーバー 1	82	OK
サーバー 2	88.6	OK
サーバー 3	59.6	シャットダウン

列名として Server Status の値、行名として Server ID の値、各セルの内容として CPU Temperature の値を使用して、マトリックスを生成できます。既存の列 (Server Status) と行の組み合わせ (Server ID) には、各セルの内容が表示されます。残りのセルでは、[Null]、[真]、[偽]、または [空] のどの値を表示するかを選択できます。

出力

サーバー ID サーバーステータス	OK	シャットダウン
サーバー 1	82	
サーバー 2	88.6	
サーバー 3		59.6

この変換を使用して、クエリ結果からフィールドを指定してマトリックスを作成します。マトリックスの出力には、これらのフィールドの一意の値間の関係が反映されます。これにより、複雑な関係を明確で構造化されたマトリックス形式で提示できます。

Group to nested table (ネストされたテーブルにグループ化)

この変換を使用して、指定されたフィールド (列) 値でデータをグループ化し、各グループの計算を処理します。同じグループ化されたフィールド値を共有するレコードが生成され、ネストされたテーブルに表示されます。

フィールドの統計を計算するには、フィールドの横にあるボックスを選択し、[計算]オプションを選択します。これにより、統計を選択する別の選択ボックスが追加されます。

次の表にサンプルデータを示します。

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン
7/7/2020 午前 11 時 34 分 20 秒	サーバー 3	62	OK
7/7/2020 午前 10 時 32 分 20 秒	サーバー 2	90	オーバーロード
7/7/2020 午前 10 時 31 分 22 秒	サーバー 3	55	OK
7/7/2020 午前 9 時 30 分 57 秒	サーバー 3	62	再起動
7/7/2020 午前 9 時 30 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 28 分 6 秒	サーバー 1	80	OK
7/7/2020 午前 9 時 25 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK

この変換には 2 つのステップがあります。まず、データをグループ化するフィールドを 1 つ以上指定します。これにより、これらのフィールドの同じ値がすべて、ソートされたかのようにグループ化されます。例えば、`Server ID` フィールドでグループ化した場合、Grafana は次のようにデータをグループ化します。

<table>
<tr><th>サーバー ID</th><th>データ</th></tr>
<tr><td>サーバー 1</td><td>
<table>
<tr><th>時間</th><th>CPU 温度</th><th>サーバーステータス</th></tr>
<tr><td>7/7/2020 午前 11 時 34 分 20 秒</td><td>80</td><td>シャットダウン</td></tr>
<tr><td>7/7/2020 午前 9 時 28 分 6 秒</td><td>80</td><td>OK</td></tr>
<tr><td>7/7/2020 午前 9 時 23 分 7 秒</td><td>86</td><td>OK</td></tr>
</table>
</td></tr>
<tr><td>サーバー 2</td><td>
<table>
<tr><th>時間</th><th>CPU 温度</th><th>サーバーステータス</th></tr>
<tr><td>7/7/2020 午前 10 時 32 分 20 秒</td><td>90</td><td>オーバーロード</td></tr>
<tr><td>7/7/2020 午前 9 時 30 分 5 秒</td><td>88</td><td>OK</td></tr>
<tr><td>7/7/2020 午前 9 時 25 分 5 秒</td><td>88</td><td>OK</td></tr>
</table>
</td></tr>
</table>

サーバー ID	データ
サーバー 3	時間 / CPU 温度 / サーバーステータス 7/7/2020 午前 11 時 34 分 20 秒 / 62 / OK 7/7/2020 午前 10 時 31 分 22 秒 / 55 / OK 7/7/2020 午前 9 時 30 分 57 秒 / 62 / 再起動

データをグループ化するフィールドを選択すると、他のフィールドにさまざまな計算を追加でき、その計算を行の各グループに適用できます。例えば、これらの各サーバーの平均 CPU 温度を計算できます。これを行うには、CPU 温度フィールドに適用された平均計算を追加して、次の結果を取得します。

サーバー ID	CPU 温度 (平均)	
サーバー 1	82	時間 / サーバーステータス 7/7/2020 午前 11 時 34 分 20 秒 / シャットダウン 7/7/2020 午前 9:28:06 / OK

<table>
<tr><th>サーバー ID</th><th>CPU 温度 (平均)</th><th></th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>時間</th><th>サーバーステータス</th></tr>
<tr><td>7/7/2020 午前 9:23:07</td><td>OK</td></tr>
</table>
</td></tr>
<tr><td>サーバー 2</td><td>88.6</td><td>
<table>
<tr><th>時間</th><th>サーバーステータス</th></tr>
<tr><td>7/7/2020 午前 10 時 32 分 20 秒</td><td>オーバーロード</td></tr>
<tr><td>7/7/2020 午前 9 時 30 分 5 秒</td><td>OK</td></tr>
<tr><td>7/7/2020 午前 9 時 25 分 5 秒</td><td>OK</td></tr>
</table>
</td></tr>
</table>

サーバー ID	CPU 温度 (平均)	
サーバー 3	59.6	

時間	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	OK
7/7/2020 午前 10 時 31 分 22 秒	OK
7/7/2020 午前 9 時 30 分 57 秒	再起動

Create heatmap (ヒートマップの作成)

この変換を使用して、経時的な傾向を視覚化するためのヒストグラムデータを作成します。ヒートマップ視覚化と同様に、この変換はヒストグラムメトリクスを時間バケットに変換します。

X バケット

この設定は、X 軸をバケットに分割する方法を決定します。

- [サイズ] – 入力フィールドに時間間隔を指定します。例えば、1h の時間範囲により、x 軸に 1 時間幅のセルが作成されます。
- [Count] – 時間関連以外の系列の場合、このオプションを使用してバケット内の要素の数を定義します。

Y バケット

この設定は、Y 軸をバケットに分割する方法を決定します。

- [線形]
- [対数] – 底が 2 のログまたは底が 10 のログから選択します。

- [Symlog] – 正負対象の対数目盛を使用します。負の値を許可して、底が 2 のログまたは底が 10 のログから選択します。

次のデータセットがあるとします。

Timestamp	値
2023-01-01 12:00:00	5
2023-01-01 12:15:00	10
2023-01-01 12:30:00	15
2023-01-01 12:45:00	8

- X バケットを `Size: 15m` に設定し、Y バケットを `Linear` に設定すると、ヒストグラムは値を X 軸で 15 分間隔、Y 軸で線形に編成します。
- X バケットを `Count: 2`、Y バケットを `Logarithmic (base 10)` とすると、ヒストグラムは X 軸の 2 つのバケットに値をグループ化し、Y 軸の対数スケールを使用します。

ヒストグラム

この変換を使用して、入力データに基づいてヒストグラムを生成することで、値の分布を視覚化できます。

- [バケットサイズ] – バケット内の最小項目と最大項目の範囲 (x最小 から x最大)。
- [バケットオフセット] – ゼロベース以外のバケットのオフセット。
- [系列の結合] – 使用可能なすべての系列を使用して統合されたヒストグラムを作成。

元のデータ

系列 1

A	B	C
1	3	5

A	B	C
2	4	6
3	5	7
4	6	8
5	7	9

系列 2

C
5
6
7
8
9

出力

x最小	x最大	A	B	C	C
1	2	1	0	0	0
2	3	1	0	0	0
3	4	1	1	0	0
4	5	1	1	0	0
5	6	1	1	1	1
6	7	0	1	1	1

x最小	x最大	A	B	C	C
7	8	0	1	1	1
8	9	0	0	1	1
9	10	0	0	1	1

生成されたヒストグラムを使用して値の分布を視覚化し、データの広がりと密度に関するインサイトを提供します。

Join by field (フィールドで結合)

この変換を使用して、複数の結果を 1 つのテーブルにマージすることで、異なるクエリからのデータを統合できます。

複数の時系列の結果を共有時間フィールドを持つ 1 つのワイドテーブルに変換する場合に特に便利です。

Inner join (内部結合)

内部結合は、選択したフィールドからすべてのテーブルが同じ値を共有する複数のテーブルからのデータをマージします。このタイプの結合では、すべての結果で値が一致しないデータを除外します。

この変換を使用して、複数のクエリの結果 (渡された結合フィールドまたは最初の列の組み合わせ) を 1 つの結果に結合し、正常な結合を実行できない行を削除します。

次の例では、2 つのクエリがテーブルデータを返します。内部結合変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233

クエリ B

時間	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	サーバー 2	5
7/7/2020 午前 11 時 4 分 20 秒	サーバー 3	10

内部結合変換を適用した後の結果は次のようになります。

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233	サーバー 2	5

Outer join (外部結合)

外部結合には、すべての入力で値が一致しない内部結合と行のすべてのデータが含まれます。内部結合が時間フィールドでクエリ A とクエリ B を結合している間、外部結合には時間フィールドで一致しないすべての行が含まれます。

次の例では、2 つのクエリがテーブルデータを返します。外部結合変換を適用する前に、2 つのテーブルとして視覚化されます。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233

クエリ B

時間	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	サーバー 2	5
7/7/2020 午前 11 時 4 分 20 秒	サーバー 3	10

外部結合変換を適用した後の結果は次のようになります。

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 4 分 20 秒			サーバー 3	10
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233		

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233	サーバー 2	5

Join by labels (ラベルで結合)

この変換を使用して、複数の結果を 1 つのテーブルに結合します。

これは、複数の時系列の結果を共有[ラベル]フィールドを持つ単一のワイドテーブルに変換する場合に特に便利です。

- [Join] (結合) – すべての時系列で使用可能なラベルまたは共通ラベル間で結合するラベルを選択します。
- [値] – 出力結果の名前。

例

入力 1: `series1{what='Temp', cluster='A', job='J1'}`

時間	値
1	10
2	200

入力 2: `series2{what='Temp', cluster='B', job='J1'}`

時間	値
1	10
2	200

入力 3: `series3{what='Speed', cluster='B', job='J1'}`

時間	値
22	22
28	77

構成

```
value: 'what'
```

出力

クラスター	ジョブ	温度	[Speed] (スピード)
A	J1	10	
A	J1	200	
B	J1	10	22
B	J1	200	77

時系列データをこの変換と効果的に組み合わせて整理し、包括的なインサイトを提供します。

フィールドへのラベル付け

この変換を使用して、ラベルまたはタグを含む時系列の結果を、結果の各ラベルのキーと値を含むテーブルに変換します。ラベルを列または行の値として表示して、拡張データの視覚化を行います。

2 つの時系列のクエリ結果を示します。

- 系列 1 – ラベル `Server=Server A`、 `Datacenter=EU`
- 系列 2 – ラベル `Server=Server B`、 `Datacenter=EU`

[列] モードでは、結果は次のように表示されます。

時間	[Server] (サーバー)	データセンター	値
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	EU	1
7/7/2020 午前 11 時 34 分 20 秒	サーバー B	EU	2

[Rows] (行) モードでは、結果には各系列のテーブルがあり、次のような各ラベル値が表示されます。

ラベル	値
[Server] (サーバー)	サーバー A
データセンター	EU

ラベル	値
[Server] (サーバー)	サーバー B
データセンター	EU

値フィールド名

値フィールド名としてサーバーを選択した場合、サーバーラベルの値ごとに 1 つのフィールドを取得します。

時間	データセンター	サーバー A	サーバー B
7/7/2020 午前 11 時 34 分 20 秒	EU	1	2

マージ動作

フィールドトランスフォーマーへのラベルは、内蔵されている 2 つの異なる変換です。1 つ目は、1 つの系列で動作し、ラベルをフィールドに抽出します。2 つ目は、すべての結果を 1 つのテーブルに結合するマージ変換です。マージ変換は、一致するすべてのフィールドで結合を試みます。このマージステップは必須であり、オフにすることはできません。

これを説明するために、重複ラベルのない時系列を返すクエリが 2 つある例を次に示します。

- 系列 1 – ラベル `Server=ServerA`
- 系列 2 – ラベル `Datacenter=EU`

これにより、まず次の 2 つのテーブルが作成されます。

時間	[Server] (サーバー)	値
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	10

時間	データセンター	値
7/7/2020 午前 11 時 34 分 20 秒	EU	20

マージ後

時間	[Server] (サーバー)	値	データセンター
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	10	
7/7/2020 午前 11 時 34 分 20 秒		20	EU

[制限]

この変換を使用して、表示される行数を制限し、データに対してより焦点を絞ったビューを提供します。これは、大規模なデータセットを処理する場合に特に便利です。

以下は、データソースからのレスポンスに対する[制限]変換の影響を示す例です。

時間	メトリクス	値
7/7/2020 午前 11 時 34 分 20 秒	温度	25
7/7/2020 午前 11 時 34 分 20 秒	湿度	22
7/7/2020 午前 10 時 32 分 20 秒	湿度	29
7/7/2020 午前 10 時 31 分 22 秒	温度	22
7/7/2020 午前 9 時 30 分 57 秒	湿度	33
7/7/2020 午前 9 時 30 分 5 秒	温度	19

以下は、値が「3」の制限変換を追加した後の結果です。

時間	メトリクス	値
7/7/2020 午前 11 時 34 分 20 秒	温度	25
7/7/2020 午前 11 時 34 分 20 秒	湿度	22
7/7/2020 午前 10 時 32 分 20 秒	湿度	29

この変換により、データの視覚的な表示を調整して、最も関連性の高いデータに集中できます。

Merge series/tables (系列/テーブルのマージ)

この変換を使用して、複数のクエリの結果を 1 つの結果に結合します。これは、テーブルパネルの視覚化を使用する場合に特に便利です。共有フィールドに同じデータが含まれている場合、変換は値を同じ行にマージします。

以下に、テーブルデータを返す 2 つのクエリに対する [系列/テーブルのマージ] 変換の影響を例示します。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233

クエリ B

時間	ジョブ	エラー
7/7/2020 午前 11 時 34 分 20 秒	ノード	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	5

マージ変換を適用した後の結果は次のとおりです。

時間	ジョブ	エラー	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	15	25260122

時間	ジョブ	エラー	アップタイム
7/7/2020 午前 11 時 24 分 20 秒	postgre	5	123001233

この変換では、クエリ A とクエリ B の値を統一されたテーブルに結合し、より良いインサイトを得るためにプレゼンテーションを強化します。

Organize fields by name (名前でフィールドを整理)

この変換を使用して、パネル内の 1 つのクエリによって返されるフィールドの名前を変更、並べ替え、または非表示にします。この変換は単一クエリを持つパネルでのみ機能します。パネルに複数のクエリがある場合は、[外部結合] 変換を適用するか、追加のクエリを削除する必要があります。

[フィールドの変換]

Grafana では、クエリが返すフィールドのリストが表示されます。次のようにできます :

- フィールドの順序の変更 – リスト内の新しい場所にフィールドをドラッグします。
- フィールドを非表示または表示 – フィールド名の横の目のアイコンを使用して、フィールドの表示を切り替えます。
- 名前変更フィールド – 名前変更ボックスに新しい名前を入力します。

例

この最初のクエリ結果を考慮すると、次のようになります。

時間	メトリクス	値
2020-07-07 11:34:20	温度	25
2020-07-07 11:34:20	湿度	22
2020-07-07 10:32:20	湿度	29

名前変更フィールドオーバーライドを適用して、以下を作成できます。

時間	センサー	読み取り
2020-07-07 11:34:20	温度	25
2020-07-07 11:34:20	湿度	22
2020-07-07 10:32:20	湿度	29

この変換により、クエリ結果の表示をカスタマイズし、Grafana 内のデータに対して明確で本質を突いた表現を確保できます。

値による分割

この変換を使用して、異なる WHERE 句を持つ複数のクエリを必要とせずに、複数の系列をグラフ化するプロセスを合理化します。

> **Note**
>
> この関数は、Grafana バージョン 9 以降と互換性のあるワークスペースで使用できます。

これは、以下の例のように、メトリクス SQL テーブルを使用する場合に特に便利です。

時間	リージョン	値
10/20/2022 12:00:00 PM	米国	1520
10/20/2022 12:00:00 PM	EU	2936
10/20/2022 午前 1:00:00	米国	1327
10/20/2022 午前 1:00:00	EU	912

[値による分割]トランスフォーマーを使用すると、単一のクエリを発行し、選択した 1 つ以上の列 (フィールド) に対して一意の値で結果を分割できます。次の例では Region を使用しています。

```
SELECT Time, Region, Value FROM metrics WHERE Time > '2022-10-20'
```

時間	リージョン	値
10/20/2022 12:00:00 PM	米国	1520
10/20/2022 午前 1:00:00	米国	1327

時間	リージョン	値
10/20/2022 12:00:00 PM	EU	2936
10/20/2022 午前 1:00:00	EU	912

この変換により、プロセスが簡素化され、同じ時系列視覚化内で複数の系列を視覚化する柔軟性が向上します。

Prepare times series (時系列の準備)

この変換を使用して、データソースが目的の視覚化と互換性のない形式で時系列データを返す際の問題に対処します。この変換により、時系列データをワイド形式とロング形式との間で変換できます。

[マルチフレーム時系列]

このオプションを使用して、時系列データフレームをワイド形式からロング形式に変換します。これは、データソースが視覚化と最適な互換性を得るために再形成する必要がある形式で、時系列情報を配信する場合に特に役立ちます。

例

入力

Timestamp	値 1	値 2
2023-01-01 00:00:00	10	20
2023-01-01 01:00:00	15	25

次のように変換できます。

Timestamp	変数	値
2023-01-01 00:00:00	値 1	10
2023-01-01 00:00:00	値 2	20
2023-01-01 01:00:00	値 1	15
2023-01-01 01:00:00	値 2	25

[ワイド時系列]

このオプションを使用して、時系列データフレームをロング形式からワイド形式に変換します。これは、データソースが時系列データをロング形式で配信し、視覚化にワイド形式が必要な場合に特に役立ちます。

例

入力

Timestamp	変数	値
2023-01-01 00:00:00	値 1	10
2023-01-01 00:00:00	値 2	20
2023-01-01 01:00:00	値 1	15
2023-01-01 01:00:00	値 2	25

次のように変換できます。

Timestamp	値 1	値 2
2023-01-01 00:00:00	10	20
2023-01-01 01:00:00	15	25

削減

この変換を使用すると、データフレームの各フィールドに計算が適用され、単一の値が返されます。この変換は、複数の時系列データをよりコンパクトで要約された形式に統合するために特に役立ちます。この変換を適用すると、時間フィールドは削除されます。

以下の入力を検討します。

クエリ A

時間	Temp	アップタイム
2020-07-07 11:34:20	12.3	256122
2020-07-07 11:24:20	15.4	1230233

クエリ B

時間	AQI	エラー
2020-07-07 11:34:20	6.5	15
2020-07-07 11:24:20	3.2	5

縮小のトランスフォーマーには 2 つのモードがあります。

- [シリーズから行] – 各フィールドの行と各計算の列を作成します。
- [縮小フィールド] – 既存のフレーム構造を維持しますが、各フィールドを 1 つの値に折りたたみます。

例えば、系列で[最初]と[最後]の計算を使用して[シリーズから行]変換を行った場合、結果は次のようになります。

フィールド	First	Last
Temp	12.3	15.4

フィールド	First	Last
アップタイム	256122	1230233
AQI	6.5	3.2
エラー	15	5

最後の計算における縮小フィールドは、それぞれ 1 行、2 つのフレームになります。

クエリ A

温度	アップタイム
15.4	1230233

クエリ B

AQI	エラー
3.2	5

[正規表現で名前を変更]

この変換を使用して、正規表現と置換パターンを使用してクエリ結果の一部の名前を変更します。

正規表現を指定できます。正規表現は、バックリファレンスをサポートする置換パターンとともに、一致にのみ適用されます。例えば、ホストあたりの CPU 使用率を視覚化し、ドメイン名を削除するとします。正規表現を `([^\.]+)\..+` に設定し、置換パターンを `$1` に設定することで、`web-01.example.com` は `web-01` になります。

この変換により、視覚化のニーズに合わせてデータをカスタマイズできるため、ダッシュボードはより有益でユーザーフレンドリーになります。

Rows to fields (行からフィールドへ)

この変換を使用して、行を別々のフィールドに変換します。これは、フィールドを個別にスタイル変更および設定できるため便利です。また、動的フィールド設定のソースとして追加のフィールドを使

用したり、フィールドラベルにマッピングしたりすることもできます。その後、追加のラベルを使用して、結果フィールドにより適切な表示名を定義できます。

この変換には、設定クエリによって返されるデータ内のすべてのフィールドを一覧表示するフィールドテーブルが含まれます。このテーブルでは、各設定プロパティにマッピングするフィールドを制御できます ([用途] オプション)。返されるデータに複数の行がある場合、選択する値を選ぶこともできます。

この変換には以下が必要です。

- フィールド名のソースとして使用する 1 つのフィールド。

  デフォルトでは、変換は最初の文字列フィールドをソースとして使用します。このデフォルト設定は、代わりに使用するフィールドの [用途] 列で [フィールド名] を選択してオーバーライドできます。
- 値のソースとして使用する 1 つのフィールド。

  デフォルトでは、変換は最初の数値フィールドをソースとして使用します。ただし、このデフォルト設定は、代わりに使用するフィールドの [用途] 列で [フィールド値] を選択してオーバーライドできます。

以下にデータを視覚化する場合に役立ちます。

- ゲージ
- 統計
- 円グラフ

追加フィールドをラベルにマッピング

フィールドが設定プロパティにマッピングされない場合、Grafana は出力フィールドのラベルのソースとして自動的に使用します。

例

名前	データセンター	値
サーバー A	米国	100
サーバー B	EU	200

出力:

サーバー A (ラベル: データセンター: US)	サーバー B (ラベル: データセンター: EU)
100	200

フィールド表示名で追加のラベルを使用して、より完全なフィールド名を提供できるようになりました。

あるクエリから設定を抽出して別のクエリに適用する場合は、[クエリ結果から設定] 変換を使用する必要があります。

例

入力

名前	値	最大
サーバー A	10	100
サーバー B	20	200
サーバー C	30	300

出力

サーバー A (設定: 最大=100)	サーバー B (設定: 最大=200)	サーバー C (設定: 最大=300)
10	20	30

ソースデータ内の各行が個別のフィールドになります。各フィールドには、最大設定オプションも設定されるようになりました。[最小値]、[最大値]、[単位]、[しきい値] などのオプションはすべてフィールド設定の一部であり、このような設定を行うと、パネルエディタのオプションペインで手動設定されるオプションではなく、視覚化によって使用されます。

この変換により、行を個々のフィールドに変換し、動的なフィールド設定を容易にし、追加のフィールドをラベルにマッピングできます。

シリーズから行

この変換を使用して、複数の時系列データクエリの結果を 1 つの結果にまとめます。これはテーブルパネルの視覚化を使用する場合に役立ちます。

この変換の結果には、時間、メトリクス、値の 3 つの列が含まれます。メトリクス列が追加されると、メトリクスのソースとなるクエリを簡単に確認できます。ソースクエリのラベルを定義して、この値をカスタマイズします。

次の例では、時系列データを返すクエリが 2 つあります。変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	温度
2020-07-07 11:34:20	25
2020-07-07 10:31:22	22
2020-07-07 09:30:05	19

クエリ B

時間	湿度
2020-07-07 11:34:20	24
2020-07-07 10:32:20	29
2020-07-07 09:30:57	33

シリーズから行 変換を適用した後の結果は次のとおりです。

時間	メトリクス	値
2020-07-07 11:34:20	温度	25
2020-07-07 11:34:20	湿度	22

時間	メトリクス	値
2020-07-07 10:32:20	湿度	29
2020-07-07 10:31:22	温度	22
2020-07-07 09:30:57	湿度	33
2020-07-07 09:30:05	温度	19

この変換により、複数の時系列クエリの結果の統合が容易になり、効率的な分析と視覚化のための合理化された統合データセットが表形式で提供されます。

並べ替え

この変換を使用して、指定されたフィールドに基づいてクエリ結果内の各フレームをソートし、データを理解し分析しやすくします。ソートに必要なフィールドを設定することで、テーブルまたは視覚化にデータが表示される順序を制御できます。

[Reverse] (リバース) スイッチを使用して、指定されたフィールド内の値を逆に順序付けします。この機能は、分析ニーズに合わせて昇順と降順をすばやく切り替える場合に特に便利です。

例えば、時系列データがデータソースから取得されるシナリオでは、[並べ替え]変換を適用することで、分析要件に応じてタイムスタンプに基づいてデータフレームを昇順または降順に配置できます。この機能を使用すると、時系列データを簡単にナビゲートして解釈でき、整理された視覚的に一貫性のあるプレゼンテーションから貴重なインサイトを得られます。

Spatial (空間)

この変換を使用して、空間オペレーションをクエリ結果に適用します。

- [Action] – アクションを選択します。
  - [Prepare spatial field] (空間フィールドの準備) – 他のフィールドの結果に基づいてジオメトリフィールドを設定します。
    - [Location mode] – ロケーションモードを選択します (これらのオプションは、値の計算モードと変換モードによって共有されます)。
      - [Auto] – デフォルトのフィールド名に基づいて位置データを自動的に識別します。
      - [Coords] (座標) – 緯度と経度フィールドを指定します。
      - [Geohash] – ジオハッシュフィールドを指定します。

        - [ルックアップ] - Gazetteer のロケーションフィールドを指定します。
    - [計算]– ジオメトリを使用して新しいフィールド (方位/距離/面積) を定義します。
        - [関数] – ジオメトリに適用する数学演算を選択します。
            - [Heading] (方位) – 2 つのポイント間の方位 (方向) を計算します。
            - [面積] – ジオメトリで定義されるポリゴンで囲まれたエリアを計算します。
            - [Distance] (距離) – 2 つのポイント間の距離を計算します。
    - [変換] - ジオメトリに空間オペレーションを適用します。
        - [操作] – ジオメトリに適用する操作を選択します。
            - [線] – 各行の頂点による単線機能を作成します。
            - [ラインビルダー] – 2 つのポイントの間に線を作成します。

この変換により、地理空間データを操作および分析し、ポイント間の線の作成、空間プロパティの計算などの操作が可能になります。

時系列のテーブル変換

この変換を使用して時系列結果をテーブルに変換し、時系列データフレームを[トレンド]フィールドに変換します。[トレンド]フィールドは、[スパークラインセルタイプ]を使用してレンダリングでき、テーブル行ごとにインラインスパークラインを生成します。複数の時系列クエリがある場合、それぞれが個別のテーブルデータフレームになります。これらは結合またはマージ変換を使用して結合し、行ごとに複数のスパークラインを持つ単一のテーブルを生成できます。

生成された[トレンド]フィールド値ごとに、計算関数を選択できます。デフォルトは [最後の null 以外の値]です。この値はスパークラインの隣に表示され、テーブル行のソートに使用されます。

Regression analysis (回帰分析)

この変換を使用して、統計モデルによって予測された値を含む新しいデータフレームを作成します。これは、混沌としたデータの傾向を見つけるのに役立ちます。線形回帰または多項式回帰を使用して、データに数学関数を適合させることによって機能します。その後、データフレームを視覚化で使用してトレンドラインを表示できます。

これには、次のように 2 つの異なるモデルがあります。

- 線形回帰 – 線形関数をデータに適合させます。
- 多項式回帰 – 多項式関数をデータに適合させます。

> **Note**
>
> この変換は現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

クエリのトラブルシューティング

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このページでは、一般的なダッシュボードの問題を解決するための情報を提供します。

関数を再配置すると、異なる結果が得られる

関数の順序は非常に重要です。数学と同様に、関数を配置する順序は結果に影響を与える可能性があります。

クエリリクエストとレスポンスを検査する

最も一般的な問題は、データソースからのクエリとレスポンスに関連しています。Grafana のバグや視覚化の問題のように見えても、ほとんどの場合、データソースクエリまたはデータソースレスポンスの問題です。まず、パネルクエリとレスポンスを検査します。

詳細については、「リクエストとレスポンスデータの検査」を参照してください。

クエリが遅い

クエリが返すデータポイントの数を確認してください。多くのデータポイントを返すクエリは遅くなります。解決策は以下のとおりです。

- [クエリオプション]で、返される[最大データポイント数]を制限します。
- クエリオプションで、最小間隔時間を長くします。
- クエリで、`group by` 関数を使用します。

計算タイプ

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

次の表に、Grafana で実行できる計算の一覧を示します。これらの計算は、[変換] タブと棒ゲージ、ゲージ、および統計の視覚化で確認できます。

計算	説明
All nulls	すべての値が null のとき真
[すべての値]	すべての値を含む配列
すべての一意の値	すべての一意の値を含む配列
All zeros	すべての値が 0 のとき真
Change count	フィールドの値が変更される回数
Count	フィールド内の値の数
Delta	値の累積変化、カウント増分のみ
Difference	フィールドの最初の値と最後の値の差
Difference percent	フィールドの最初と最後の値の変化率
Distinct count	フィールド内の一意の値の数
First	フィールドの最初の値
First* (not null)	First は、フィールドの null 値ではありません (NaN も除く)

計算	説明
Last	フィールドの最後の値
Last* (not null)	Last は、フィールドの null 値ではありません (NaN も除く)
最大	フィールドの最大値
Mean	フィールド内のすべての値の平均値
Variance	フィールド内のすべての値の分散
StdDev	フィールド内のすべての値の標準偏差
最小	フィールドの最小値
最小 (above zero)	フィールドの最小値、正の値
Range	フィールドの最大値と最小値との差
Step	フィールドの値間の最小間隔
Total	フィールド内のすべての値の合計

パネルオプションの設定

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana パネルは、データソースクエリの定義、データの変換と書式設定、視覚化設定の構成によってカスタマイズできるデータの視覚的表現です。

パネルエディタには、クエリビルダーと、データの変換やパネルへの情報の追加に使用できる一連のオプションが含まれています。

このトピックでは、以下の方法について説明します。

- パネルを開いて編集する
- パネルのタイトルと説明を追加する
- パネル JSON モデルを表示する
- 繰り返し行とパネルを設定する

パネルの編集

ダッシュボードにパネルを追加すると、いつでもパネルを開いてクエリを変更または更新したり、データ変換を追加したり、視覚化設定を変更したりできます。

パネルを編集するには

1. 編集するパネルを含むダッシュボードを開きます。
2. パネルのいずれかの部分にカーソルを合わせると、右上隅にアクションメニューが表示されます。
3. メニューを選択し、[編集] を選択します。

   キーボードショートカットを使用してパネルを開くには、パネルにカーソルを合わせ、e を押します。

   パネルが編集モードで開きます。

パネルにタイトルと説明を追加する

パネルにタイトルと説明を追加して、視覚化に関する重要な情報をユーザーと共有します。例えば、説明を使用して、視覚化の目的を文書化します。

1. パネルを編集します。
2. パネル表示オプションペインで、[パネルオプション] セクションを見つけます。
3. [タイトル] を入力します。

   このフィールドに入力されたテキストは、パネルエディタとダッシュボードのツールチップに表示されます。

4. 表示するパネルとデータの説明を記入します。

   このフィールドに入力したテキストは、パネルの左上隅にあるツールチップに表示されます。

   [タイトル] と [説明] フィールドで[定義した変数]を使用できますが、[グローバル変数]は使用できません。

パネル JSON モデルの表示

パネル、パネルデータ、データフレーム JSON モデルを検索してエクスポートします。

パネル JSON モデルを表示するには

1. パネルを含むダッシュボードを開きます。
2. パネルのいずれかの部分にカーソルを合わせると、右上隅にアクションメニューが表示されます。
3. メニューから、[Inspect] (検査) > [Panel JSON] を選択します。
4. [ソースの選択] フィールドで、次のオプションのいずれかを選択します。
   - Panel JSON – パネルを表す JSON オブジェクトを表示します。
   - [Panel data] – パネルに渡されるデータを表す JSON オブジェクトを表示します。
   - [データフレーム構造] – 適用された変換、フィールド設定、オーバーライド設定など、パネルのデータ構造を表示します。
5. JSON を調べるには、[>] を選択して JSON モデルの一部を展開または折りたたみます。

繰り返しパネルの設定

パネルまたは行をダッシュボードに動的に追加するように Grafana を設定できます。動的パネルは、変数の値に基づいてシステムが作成するパネルです。変数は、ダッシュボード内のすべてのパネルでクエリを動的に変更します。繰り返しパネルの詳細については、「ダッシュボードの作成」を参照してください。

> **Note**
>
> パネルを繰り返すには、変数に 1 つ以上の項目を選択する必要があります。パネルを非表示にするためにパネルを 0 回繰り返すことはできません。

前提条件

- クエリに複数値変数が含まれていることを確認します。

繰り返しパネルを設定するには

1. 繰り返すパネルを編集します。
2. 表示オプションペインで、[パネルオプション] > [繰り返しオプション] を選択します。
3. [direction] (方向) を選択します。
   - [水平] を選択してパネルを並べて配置します。Grafana は繰り返しパネルの幅を調整します。行上の他のパネルと繰り返しパネルを混在させることはできません。
   - [垂直] を選択して、列にパネルを配置します。繰り返しパネルの幅は、元の繰り返しパネルの幅と同じです。
4. すべてのパネルに変更を伝達するには、ダッシュボードを再読み込みします。

標準オプションを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana で使用されるデータモデルは、時系列とテーブルのクエリ結果の両方を統合する列指向のテーブル構造です。この構造内の各列は、フィールドと呼ばれます。フィールドは、単一の時系列またはテーブル列を表すことができます。

フィールドオプションを使用すると、視覚的表現でのデータの表示方法を変更できます。適用するオプションとオーバーライドはデータを変更せず、Grafana のデータの表示方法を変更します。オプションを変更すると、すべてのフィールド、つまりすべての系列または列に適用されます。例えば、単位をパーセンテージに変更すると、数値を含むすべてのフィールドがパーセンテージで表示されます。

フィールドの書式設定オプションの完全なリストは、このトピックの後半で説明します。

> **Note**
>
> ほとんどの組み込み Grafana パネルに標準オプションを適用できます。新しいパネルとデータモデルに更新されていない古いパネルとコミュニティパネルの一部には、これらのフィールドオプションのすべてまたは一部が欠落します。

標準オプションを設定するには

1. ダッシュボードを開きます。パネルのいずれかの部分にカーソルを合わせると、ダッシュボードの右上隅にアクションメニューが表示されます。
2. アクションメニューで、[編集] を選択します。
3. パネル表示オプションペインで、[標準オプション] セクションを見つけます。
4. 適用する標準オプションを選択します。
5. 変更をプレビューするには、編集中のフィールドオプションボックスの外部を選択するか、Enter を押します。

標準オプションの定義

このセクションでは、使用可能なすべての標準オプションについて説明します。

ほとんどの組み込み Grafana パネルに標準オプションを適用できます。新しいパネルとデータモデルに更新されていない古いパネルとコミュニティパネルの一部には、これらのフィールドオプションのすべてまたは一部が欠落します。

ほとんどのフィールドオプションは、編集中のフィールドオプションボックスの外部をクリックするか、Enter キーを押すまで、視覚化には影響しません。

> **Note**
>
> Grafana Labs は、すべての視覚化に対するオプションの追加と拡張に絶えず取り組んでいるため、すべてのオプションをすべての視覚化に対して使用できない場合があります。

[単位]

フィールドで使用する単位を選択できます。[単位] フィールドを選択し、目的の単位が見つかるまでドリルダウンします。選択した単位は、時間を除くすべてのフィールドに適用されます。

ユーザー定義の単位

単位ドロップダウンを使用して、ユーザー定義の単位、接頭辞、または接尾辞と、日付時刻形式を指定することもできます。

ユーザー定義の単位を選択するには、単位を入力し、ドロップダウンで最後の [カスタム: xxx] オプションを選択します。

- suffix:<suffix> 値の後に続くユーザー定義単位を指定できます。
- prefix:<prefix> 値の前に置くユーザー定義単位を指定できます。
- time:<format> ユーザー定義の日付時刻形式の場合は、`time:YYYY-MM-DD` のように入力します。形式の構文とオプションについては、「Moment.js ドキュメント」の「形式」を参照してください。
- si:<base scale><unit characters> ユーザー定義の SI 単位を指定できます。例えば、`si: mF` などです。これは、単位とソースデータスケールの両方を指定できるため、少し高度です。したがって、ソースデータがミリ (基本単位の千分の一) で表されている場合、単位の先頭にその SI スケール文字が付けられます。
- count:<unit> ユーザー定義の集計単位を指定できます。
- currency:<unit> ユーザー定義の通貨単位を指定できます。

単位ピッカーにネイティブ絵文字を貼り付けて、ユーザー定義単位として選択することもできます。

[String units] (文字列の単位)

Grafana は、文字列を解析して数値として表示する際に、自動で表示が変わる場合があります。元の文字列値を表示するように Grafana を設定するには、フィールドオーバーライドを作成し、[文字列] 単位で単位プロパティを追加します。

スケールの単位

デフォルトでは、Grafana は値の大きさに基づいて単位を自動的にスケールします。例えば、値が 0.14 kW の場合、Grafana はそれを 140 W として表示します。別の例として、3,000 kW は 3 MW として表示されます。この動作を無効にする場合は、[スケールの単位] スイッチをオフにできます。

最小

パーセンテージしきい値の計算に使用される最小値を設定できます。最小値を自動的に計算するには、空白のままにします。

最大

パーセンテージしきい値の計算に使用される最大値を設定できます。最大値を自動的に計算するには、空白のままにします。

フィールドの最小値/最大値

デフォルトでは、計算された最小値と最大値は、すべての系列とフィールドの最小値と最大値に基づきます。フィールドの最小値/最大値をオンにすると、そのフィールドの最小値または最大値に基づいて、各フィールドの最小値または最大値が個別に計算されます。

小数

Grafana がレンダリングした値に含める小数点以下の桁数を指定します。このフィールドを空白のままにすると、Grafana は値に基づいて小数点以下の桁数を自動的に切り捨てます。例えば、1.1234 は 1.12 として表示され、100.456 は 100 として表示されます。

すべての小数を表示するには、単位を [String] (文字列) に設定します。

Display name (表示名)

すべてのフィールドの表示タイトルを設定できます。フィールドタイトルには変数を使用できます。

複数の統計、フィールド、または系列が表示される場合、このフィールドは各統計のタイトルを制御します。${__field.name} などの式を使用して、タイトルの系列名またはフィールド名のみを使用できます。

Temp という名前のフィールドと、{"Loc"="PBI", "Sensor"="3"} のラベルが付与されます。

式の構文	例	レンダリング先	説明
`${__field.displayName}`	構文と同じ	`Temp {Loc="PBI", Sensor="3"}`	フィールド名とラベルがある場合は {} に表示します。レスポンスにラベルキーが 1 つしかない場合、ラ

式の構文	例	レンダリング先	説明
			ベル部分において、Grafana は括弧なしでラベルの値を表示します。
`${__field.name}`	構文と同じ	`Temp`	フィールドの名前を表示します (ラベルなし)。
`${__field.labels}`	構文と同じ	`Loc="PBI", Sensor="3"`	ラベルを表示します (名前なし)。
`${__field.labels.`*X*`}`	`${__field.labels.Loc}`	`PBI`	指定されたラベルキーの値を表示します。
`${__field.labels._values}`	構文と同じ	`PBI, 3`	ラベルの値をカンマ区切りで表示します (ラベルキーなし)。

特定のフィールドに対する式をレンダリングした後に値が空の文字列である場合、デフォルトの表示方法が使用されます。

カラースキーム

カラーオプションとその視覚化への影響は、作業中の視覚化によって異なります。一部の視覚化には、異なるカラーオプションがあります。

値に基づいて、単一の色を指定するか、連続 (グラデーション) カラースキームを選択できます。連続色は、最小値と最大値に対する値の割合を使用して色を補間します。

次のいずれかのパレットを選択します。

カラーモード	説明
[単一色]	オーバーライドルールに役立つ単一色を指定します

カラーモード	説明
Shades of a color (色のシェード)	オーバーライドルールに役立つ 1 色のシェードを選択します。
[しきい値起点]	一致するしきい値から色を取るように Grafana に通知します
Classic palette (クラシックパレット)	Grafana は、系列のインデックスでパレットの色を検索して色を割り当てます。グラフや円グラフ、その他のカテゴリ別データの視覚化に役立ちます。
Classic palette (by series name) (クラシックパレット (系列名別))	Grafana は、系列の名前に基づいて色を割り当てます。視覚化される系列名が使用可能なデータに依存する場合に便利です。
Green-Yellow-Red (by value) (緑黄赤 (値別))	連続色スキーム
Red-Yellow-Green (by value) (赤黄緑 (値別))	連続色スキーム
Blue-Yellow-Red (by value) (青黄赤 (値別))	連続色スキーム
Yellow-Red (by value) (黄赤 (値別))	連続色スキーム
Blue-Purple (by value) (青紫 (値別))	連続色スキーム
Yellow-Blue (by value) (黄青 (値別))	連続色スキーム
Blues (by value) (青 (値別))	連続色スキーム (パネルの背景を青に)
Reds (by value) (赤 (値別))	連続色スキーム (パネルの背景色を赤に)
Greens (by value) (緑 (値別))	連続色スキーム (パネルの背景色を緑に)
Purple (by value) (紫 (値別))	連続色スキーム (パネルの背景色を紫に)

値なし

フィールド値が空または null の場合に Grafana が表示するものを入力します。デフォルト値はハイフン (-) です。

凡例を設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルには、視覚化に表示されるデータの解釈に使用できる凡例が含まれています。各凡例オプションは、視覚化に示されているデータにコンテキストと明確さを追加します。

凡例は、次の視覚化でサポートされています。

- 棒グラフ
- ローソク足チャート
- ヒストグラム
- 円グラフ
- 状態タイムライン
- ステータス履歴
- 時系列
- [トレンド]

ジオマップとヒートマップにも凡例はありますが、凡例を表示するか表示しないかの選択のみが提供され、他の凡例オプションはサポートされていません。

凡例オプション

次のオプションは、パネル編集ペインの [凡例] セクションにあります。

> **Note**
>
> リストされているすべてのオプションが、凡例を含むすべての視覚化に適用されるわけではありません。

[可視性]

凡例を表示するかどうかを設定します。凡例のオン/オフを切り替えるには、スイッチを使用します。

モード

凡例が表示される形式を設定します。以下から選択します。

- リスト
- [テーブル]

凡例をテーブルとしてフォーマットすると、関連する値や視覚化内の場所など、凡例に関する他の情報も表示されることがあります。

[配置]

視覚化で凡例が表示される場所を設定します。以下から選択します。

- [Bottom] (下)
- [Right] (右)

[幅]

凡例の配置を右に設定すると、幅オプションが使用可能になります。フィールドを空のままにして、Grafana が凡例の幅を自動的に設定できるようにするか、フィールドに値を入力します。

[Values] (値)

凡例に系列データ値または計算を追加することで、視覚化にコンテキストをさらに追加できます。必要な数だけ値を追加できます。変更を適用すると、凡例をスクロールしてすべての値を表示できます。

系列の色の変更

デフォルトでは、Grafana は系列データの色を設定しますが、パネル凡例を使用して変更できます。

系列の色を変更するには

1. 更新するパネルに移動します。
2. 凡例で、系列に関連付けられたカラーバーを選択します。
3. ピッカーまたは RGB 値を使用して、[Colors] タブでプリセット色を選択するか、[カスタム] タブでカスタム色を設定します。
4. ダッシュボードを保存します。

視覚化での系列データの分離

視覚化は視覚的に複雑になることが多く、データ系列が数多く含まれている場合があります。表示したいデータを分離する凡例を使用して、視覚化から凡例データを削除することで、表示を簡素化できます。これを行うと、Grafana は [Override] タブに新しいオーバーライドを自動的に作成します。

視覚化で系列データを分離するには

1. 更新するパネルに移動します。
2. 凡例で、分離する系列のラベルを選択します。

   システムによって、他のすべての系列データ表示から削除されます。
3. 系列データを分離された系列に段階的に追加するには、Ctrl キーまたは Command キーを押して、追加する系列のラベルを選択します。
4. パネルのすべてのビューワーに表示されるように変更を保存するには、ダッシュボードを保存します。

すべてのデータを含むデフォルトの表示に戻すには、任意の系列ラベルを 2 回クリックします。

系列のソート

凡例をテーブルとしてフォーマットして値を追加すると、テーブル内の系列をそれらの値でソートできます。

系列をソートするには

1. 更新するパネルに移動します。

2. 操作するパネルの任意の部分にカーソルを合わせると、パネルの右上隅にメニューが表示されます。
3. メニューで [編集] を選択します。
4. パネル編集ペインの [凡例] セクションまでスクロールします。
5. [Values] (値) で、表示する値または計算を選択します。

   凡例に値が表示されるようになりました。
6. 凡例テーブルの計算名ヘッダーを選択して、テーブル内の値を昇順または降順にソートします。

> **Note**
> この機能は、棒グラフ、ヒストグラム、時系列のパネルでのみサポートされています。

データリンクの設定

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

データリンクを使用すると、リンクへのより詳細なコンテキストを提供できます。カーソルの下に系列名または値を含むリンクを作成できます。例えば、視覚化で 4 つのサーバーが表示されている場合は、そのうちの 1 つまたは 2 つにデータリンクを追加できます。データリンクを使用してパネルをリンクすることもできます。

リンク自体には、視覚化に応じてさまざまな方法でアクセスできます。例えば、時系列の視覚化では、データポイントまたは行を選択します。統計、ゲージ、棒ゲージなどの広域視覚化では、視覚化の任意の場所を選択してコンテキストメニューを開くことができます。

視覚化にデータリンクが 1 つしかない場合、視覚化の任意の場所を選択すると、コンテキストメニューではなくリンクが開きます。

サポートされている視覚化

- 棒グラフ
- 棒ゲージ
- ローソク足チャート
- キャンバス
- ゲージ
- ジオマップ
- ヒートマップ
- ヒストグラム
- 円グラフ
- 統計
- 状態タイムライン
- ステータス履歴
- [テーブル]
- 時系列
- トレンド

データリンク変数

データリンクの変数を使用すると、データフィルターが保持された詳細なダッシュボードにユーザーを送信できます。例えば、 変数を使用して、ラベル、時間範囲、系列、または変数の選択を指定できます。

使用可能な変数のリストを表示するには、データリンク URL フィールドに $ を入力します。

データリンク URL のテンプレート変数も使用できます。「変数」を参照してください。

時間範囲パネル変数

これらの変数を使用すると、データリンク URL に現在の時間範囲を含めることができます。

- `__url_time_range` – 現在のダッシュボードの時間範囲 (`?from=now-6h&to=now` など)
- `$__from` – 詳細については、「グローバル変数」を参照してください。

- $__to – 詳細については、「グローバル変数」を参照してください。

系列変数

系列固有の変数は、__series 名前空間で使用できます。

- __series.name – URL への系統名

フィールド変数

フィールド固有の変数は、__field 名前空間で使用できます。

- __field.name – フィールドの名前
- __field.labels.<LABEL> – URL に対するラベルの値。ラベルにドットが含まれている場合は、__field.labels["<LABEL>"] 構文を使用します。

値変数

値固有の変数は、__value 名前空間で使用できます。

- __value.time – URL への値のタイムスタンプ (Unix ms エポック) (?time=1560268814105 など)
- __value.raw – raw 値
- __value.numeric – 値の数値表現
- __value.text – 値のテキスト表現
- __value.calc – 値が計算の結果である場合の計算名

データリンクで値固有の変数を使用すると、[ツールチップ] モードの設定オプションに応じて異なる結果が表示されることがあります。

データ変数

他のフィールドの値にアクセスするには、以下を使用します。

- __data.fields[i] – (同じ行にある) フィールド i の値。
- __data.fields["NameOfField"] – インデックスの代わりに名前を使用するフィールドの値。
- __data.fields[i].labels.cluster – 別のフィールドのラベルにアクセス。

テンプレート変数

テンプレート変数を使用する別のダッシュボードにリンクするときは、リンクをクリックした人の変数値を選択します。

${var-myvar:queryparam} - ここで、var-myvar は、使用する現在のダッシュボードにあるテンプレート変数と一致するテンプレート変数の名前です。

可変状態	作成された URL の結果
1 つの値を選択済み	var-myvar=value1
複数の値を選択済み	var-myvar=value1&var-myvar=value2
All を選択済み	var-myvar=All

現在のダッシュボードのすべての変数を URL に追加する場合は、${__all_variables} を使用します。

データリンクの追加

パネルにデータリンクを追加できます。

1. データリンクを追加するパネルに移動します。
2. パネルにカーソルを合わせると、右上隅にメニューアイコンが表示されます。
3. メニューから、[編集] を選択してパネル工ディタを開きます。
4. [パネル編集] ペインで、[データリンク] セクションまでスクロールダウンして展開します。
5. [リンクの追加] を選択します。
6. 開いたダイアログボックスに、[タイトル]を入力します。これは、判読可能なリンクラベルで、リンクは UI に表示されます。
7. リンク先の [URL] または変数を入力します。

   データリンク変数を追加するには、URL フィールドを選択し、$ を入力するか、Ctrl+Space または Cmd+Space を押すと、使用可能な変数のリストが表示されます。
8. リンクを新しいタブで開く場合は、[新しいタブで開く] を選択します。
9. [保存] を選択して変更を保存し、ダイアログボックスを閉じます。

10. ダッシュボードに変更を保存します。

値マッピングを設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

フィールドオーバーライドに加えて、値マッピングは、視覚化でのデータの表示方法を変更するために使用できる手法です。

値マッピングは、表示される色や小数点以下の桁数など、パネルエディタの標準オプションで設定された単位の書式設定を介しません。値マッピングがパネルに存在する場合、Grafana はパネルエディタの[値マッピング]セクションにその概要を表示します。

サポートされている視覚化

- 棒グラフ
- 棒ゲージ
- ローソク足チャート
- キャンバス
- ゲージ
- ジオマップ
- ヒストグラム
- 円グラフ
- 統計
- 状態タイムライン
- ステータス履歴
- [テーブル]
- 時系列

- トレンド

値マッピングの種類

Grafana は、次の値マッピングの種類をサポートしています。

- [Value] – 特定の値をテキストと色にマッピングします。例えば、値 `10` のすべてのインスタンスが数値ではなく `Perfection!` として表示されるように値マッピングを設定できます。単一の値をフォーマットする場合は、Value マッピングを使用します。
- [Range] – 数値範囲をテキストと色にマッピングします。例えば、値が特定の範囲内にある場合は、数値ではなく `Low` または `High` を表示するように範囲値マッピングを設定できます。複数の連続値を推定する場合は、Range マッピングを使用します。
- [Regex] – 正規表現をテキストと色にマッピングします。例えば、値が `www.example.com` の場合、Grafana が `www` を表示し、ドメインを切り捨てるように正規表現値マッピングを設定できます。正規表現値のテキストと色をフォーマットする場合は、Regex マッピングを使用します。
- [特殊] – `Null`、`NaN` (数値以外) などの特殊な値、`true` や `false` などのブール値をテキストと色にマッピングします。例えば、`null` 値が `N/A` として表示されるように、特殊な値マッピングを設定できます。まれな値、ブール値、または空の値をフォーマットする場合は、[特殊] マッピングを使用します。

値マッピングの追加

パネルに値マッピングを追加できます。

値マッピングを追加するには

1. 更新するパネルに移動します。
2. パネルの任意の部分にカーソルを合わせると、パネルの右上隅にメニューが表示されます。
3. メニューで [編集] を選択します。
4. [値マッピング] セクションで、[値マッピングを追加] を選択します。
5. [新しいマッピングの追加] を選択し、次のいずれかを選択します。
    - [Value] – 一致する単一の値を入力します。
    - [Range] – 一致する範囲の開始値と終了値を入力します。
    - [Regex] – 一致する正規表現パターンを入力します。
    - [特殊] – 一致する特殊な値を選択します。

6. (オプション) 表示テキストを入力します。
7. (オプション) 色を設定します。
8. [更新] を選択して、値マッピングを保存します。

マッピングを追加したら、[値マッピングの編集] ボタンが [値マッピングの追加] ボタンに置き換わります。マッピングを追加または更新するには、編集ボタンを選択します。

しきい値を設定する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードでは、しきい値はメトリクスに対して設定した値または制限であり、しきい値以上となった場合に視覚的に反映されます。しきい値は、クエリ結果に基づいて視覚化を条件付きでスタイル付けおよび色付けする方法の 1 つです。

しきい値を使用すると、次のことができます。

- 時系列の視覚化において、グリッド線またはグリッド領域を色付けします。
- 統計の視覚化において、背景または値のテキストを色付けします。
- 状態タイムラインで領域と領域の色を定義します。
- 時系列の視覚化で線を色付けします。
- ゲージにおいて、ゲージとしきい値マーカーを色付けします。
- ジオマップのマーカーを色付けします。
- テーブル内のセルのテキストまたは背景を色付けします。

サポートされている視覚化

- 棒グラフ

- 棒ゲージ
- ローソク足チャート
- キャンバス
- ゲージ
- ジオマップ
- ヒストグラム
- 統計
- 状態タイムライン
- ステータス履歴
- [テーブル]
- 時系列
- トレンド

デフォルトのしきい値

しきい値をサポートする視覚化では、Grafana には以下のデフォルトのしきい値設定があります。

- 80 = red (80 = 赤)
- Base = green (ベース = 緑)
- Mode = Absolute (モード = 絶対値)
- しきい値の表示 = オフ (一部の視覚化の場合)

  詳細については、「しきい値を表示する」を参照してください。

しきい値オプション

以下のオプションを設定して、しきい値の外観をさらに定義できます。

しきい値の値

この数値は、しきい値をトリガーする値です。このフィールドでは、しきい値に関連付けられた色を設定することもできます。

[Base] (ベース) 値は、マイナス無限大を表します。デフォルトでは、緑色に設定されています。これは一般的に「良い」色です。

しきい値モード

しきい値モードは 2 つあります。

- [絶対値] しきい値は数値で定義されます。例えば、1 ~ 150 のスケールで 80 とします。
- [パーセンテージ] しきい値は、最小値または最大値に対して定義されます。例えば、80% です。

しきい値を表示する

> **Note**
>
> このオプションは、棒グラフ、ローソク足チャート、時系列、トレンド視覚化でのみサポートされています。

次のオプションを使用して、しきい値が視覚化時に表示されるかどうかと表示方法を設定します。

- [Off] (オフ) – しきい値は表示されません。
- [線] - しきい値は視覚化で水平線として表示されます。
- [破線] - しきい値は、破線の水平線として表示されます。
- [領域の塗りつぶし] - しきい値は水平領域として表示されます。
- [領域の塗りつぶしと線] - しきい値は、線で区切られた水平領域として表示されます。
- [領域の塗りつぶしと破線] - しきい値は、破線で区切られた水平領域として表示されます。

しきい値の追加

視覚化には、必要な数のしきい値を追加できます。Grafana はしきい値を高い順に自動的にソートします。

しきい値を追加するには

1. 更新するパネルに移動します。
2. パネルの任意の部分にカーソルを合わせると、右上隅にメニューが表示されます。
3. メニューから [編集] を選択します。
4. [しきい値] セクションまでスクロールするか、パネル編集ペインの上部にある検索バーに `Thresholds` と入力します。
5. [しきい値の追加] を選択します。

6. 新しいしきい値を入力するか、フィールドの右側にある上下の矢印を使用して、値を少しずつ増減させます。
7. しきい値の左側にある色付き円をクリックしてカラーピッカーを開くと、しきい値の色を更新できます。
8. [しきい値モード] で、[絶対値] または [パーセンテージ] を選択します。
9. [しきい値の表示] で、しきい値の表示方法を設定するか、オフにします。

しきい値を削除するには、しきい値を含むパネルに移動し、削除するしきい値の横にあるごみ箱アイコンを選択します。

フィールドのオーバーライドを設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

オーバーライドを使用すると、特定のフィールドまたは系列に対して視覚化設定をカスタマイズできます。これは、特定のフィールドセットをターゲットとし、それぞれ複数のオプションを定義できるオーバーライドルールを追加することで実現されます。

例えば、[正規表現と一致する名前付きフィールド] を使用してオーバーライドを追加して、テキスト `bytes` を含むすべてのフィールドに単位を設定し、オーバーライドルールに単位オプションを追加できます。

例 1: 温度の形式

結果セットが、`time` と `temperature` の 2 つのフィールドで構成されるデータフレームであると仮定します。

時間	温度
2020-01-02 03:04:00	45.0

時間	温度
2020-01-02 03:05:00	47.0
2020-01-02 03:06:00	48.0

この構造の各フィールド (列) には、値の表示方法を変更するフィールドオプションを適用できます。つまり、例えば、単位を温度 > 摂氏に設定して、次の表を作成できます。

時間	温度
2020-01-02 03:04:00	45.0 °C
2020-01-02 03:05:00	47.0 °C
2020-01-02 03:06:00	48.0 °C

また、小数点は必須ではないため、削除できます。小数を自動からゼロ (0) に変更すると、次の表になります。

時間	温度
2020-01-02 03:04:00	45 °C
2020-01-02 03:05:00	47 °C
2020-01-02 03:06:00	48 °C

例 2: 温度と湿度の形式

結果セットが、`time`、`high temp`、`low temp`、`humidity` の 4 つのフィールドで構成されるデータフレームであるとします。

時間	高温	低温	湿度
2020-01-02 03:04:00	45.0	30.0	67

時間	高温	低温	湿度
2020-01-02 03:05:00	47.0	34.0	68
2020-01-02 03:06:00	48.0	31.0	68

摂氏単位を追加し、小数点以下を削除しましょう。この結果は以下の表になります。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67 °C
2020-01-02 03:05:00	47 °C	34 °C	68 °C
2020-01-02 03:06:00	48 °C	31 °C	68 °C

温度フィールドは良く見えますが、湿度を変更する必要があります。これを修正するには、湿度フィールドにフィールドオプションオーバーライドを適用し、単位を [Misc] (その他) > % (0 ~ 100) に変更します。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67%
2020-01-02 03:05:00	47 °C	34 °C	68%
2020-01-02 03:06:00	48 °C	31 °C	68%

フィールドオーバーライドの追加

フィールドオーバーライドルールは、特定のフィールドまたは系列の視覚化設定をカスタマイズできます。

フィールドオーバーライドの追加方法

1. オーバーライドを追加するパネルを編集します。

2. パネルオプションサイドペインで、ペインの下部にある [フィールドオーバーライドの追加] を選択します。
3. オーバーライドルールが適用されるフィールドを選択します。

   - [名前付きフィールド] – 使用可能なすべてのフィールドの一覧からフィールドを選択します。このセレクタでルールに追加するプロパティは、この単一フィールドにのみ適用されます。
   - [正規表現と一致する名前付きフィールド] – 正規表現でオーバーライドするフィールドを指定します。このセレクタでルールに追加するプロパティは、フィールド名が正規表現と一致するすべてのフィールドに適用されます。これにより、オーバーライドするフィールドが選択されますが、フィールドの名前は変更されません。これを行うには、[正規表現変換による名前の変更] を使用します。
   - [タイプ別フィールド] – 文字列、数値などのフィールドをタイプ別に選択します。このセレクタでルールに追加するプロパティは、選択したタイプに一致するすべてのフィールドに適用されます。
   - [クエリによって返されるフィールド]: A、B、C など、特定のクエリによって返されるすべてのフィールドを選択します。このセレクタでルールに追加するプロパティは、選択したクエリによって返されるすべてのフィールドに適用されます。
4. [オーバーライドプロパティの追加] を選択します。
5. 適用するフィールドオプションを選択します。
6. フィールドに値を追加してオプションを入力します。オプションをデフォルト値に戻すには、フィールドの白い文字を削除します。
7. [オーバーライドプロパティの追加] を選択して、このフィールドにオーバーライドを追加し続けるか、[オーバーライドの追加] を選択し、オーバーライドを追加する別のフィールドを選択できます。
8. 完了したら [保存] を選択して、すべてのパネル編集をダッシュボードに保存します。

フィールドオーバーライドの削除

不要になったフィールドオーバーライドを削除します。オーバーライドを削除すると、値の外観はデフォルトで元の形式になります。この変更は、影響を受けるパネルに依存するダッシュボードとダッシュボードユーザーに影響します。

フィールドオーバーライドを削除するには

1. 削除するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。

3. 削除するオーバーライドを選択し、関連付けられたごみ箱アイコンを選択します。

フィールドオーバーライドの表示

フィールドオーバーライドは、パネル表示オプションで表示できます。

1. 表示するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。

> **Note**
>
> [すべて] タブに表示されるオーバーライド設定は、[オーバーライド] タブに表示される設定と同じです。

フィールドオーバーライドの編集

オーバーライド設定に変更を加える場合は、フィールドオーバーライドを編集します。加えた変更はすぐに反映されます。

フィールドオーバーライドを編集するには

1. 編集するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。
3. 変更するオーバーライドを見つけます。
4. 次のいずれかを実行します。
   - 既存のオーバーライドまたはフィールド選択パラメータの設定を編集します。
   - プロパティの横にある [X] を選択して、既存のオーバーライドプロパティを削除します。
   - [オーバーライドプロパティの追加] を選択して、オーバーライドプロパティを追加します。

Grafana バージョン 10 で利用可能な視覚化

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は、さまざまなユースケースをサポートするさまざまな視覚化を提供します。ドキュメントのこのセクションでは、組み込まれている視覚化、そのオプション、そして一般的な使用方法について説明します。

パネルの使用の開始からその基本まで学べる一般的なパネルは、時系列 パネルです。

> **Note**
>
> どの視覚化を選択するかわからない場合、Grafana はパネルクエリに基づいて視覚化について提案できます。視覚化を選択すると、その視覚化が適用されたプレビューが Grafana に表示されます。

- グラフとチャート
  - 時系列は、デフォルトおよび主要なグラフ視覚化です。
  - 状態タイムラインは、時間の経過に伴う状態変化を表します。
  - ステータス履歴は、時間の経過に伴う定期的な状態を表します。
  - 棒グラフは、カテゴリ別データを表示します。
  - ヒストグラムは、値分布を計算し、棒グラフで表示します。
  - ヒートマップは、データを 2 つのディメンションで視覚化します。通常、現象の大きさを表すのに使用されます。
  - 円グラフは通常、比例性が重要な場合に使用されます。
  - ローソク足チャートは通常、価格/データ移動が焦点となる財務データに使用されます。
  - ゲージは、単一のメトリクスがしきい値からどの程度離れているかを示す従来の丸みを帯びたビジュアルです。
  - [トレンド]は、時間ではない連続した数値 x を持つデータセットに使用されます。
  - XY チャートは、グラフ内の任意の x および y 値を視覚化する方法を提供します。
- 統計と数値
  - [Stat] は、大規模統計とオプションのスパークラインを表示します。

  - [ゲージサイズ] は、水平または垂直の棒ゲージです。
- [Misc] (その他)
  - [テーブル] は、主要かつ唯一のテーブルの視覚化です。
  - [ログ] は、ログの主要な視覚化です。
  - [Node graph] は、有向性グラフまたはネットワークを表します。
  - [Traces] は、トレースに対する主な視覚化です。
  - [Flame graph] は、プロファイリングに対する主な視覚化です。
  - [Geomap] は、地理空間データを視覚化できます。
  - [Datagrid] を使用すると、データを作成および操作でき、他のパネルのデータソースとして機能します。
- ウィジェット
  - ダッシュボードリストはダッシュボードを一覧表示できます。
  - アラートリストはアラートを一覧表示できます。
  - テキストには、マークダウンと html を表示できます。
  - ニュースには RSS フィードが表示されます。

詳細を見る

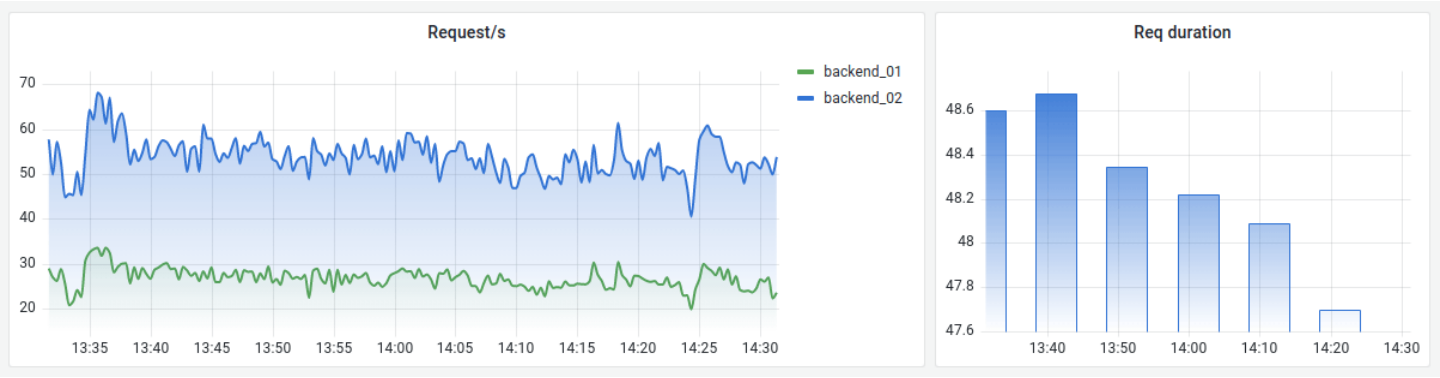
プラグインカタログによるプラグインの検索 からパネルプラグインをインストールすることで、視覚化タイプをさらに追加できます。

例

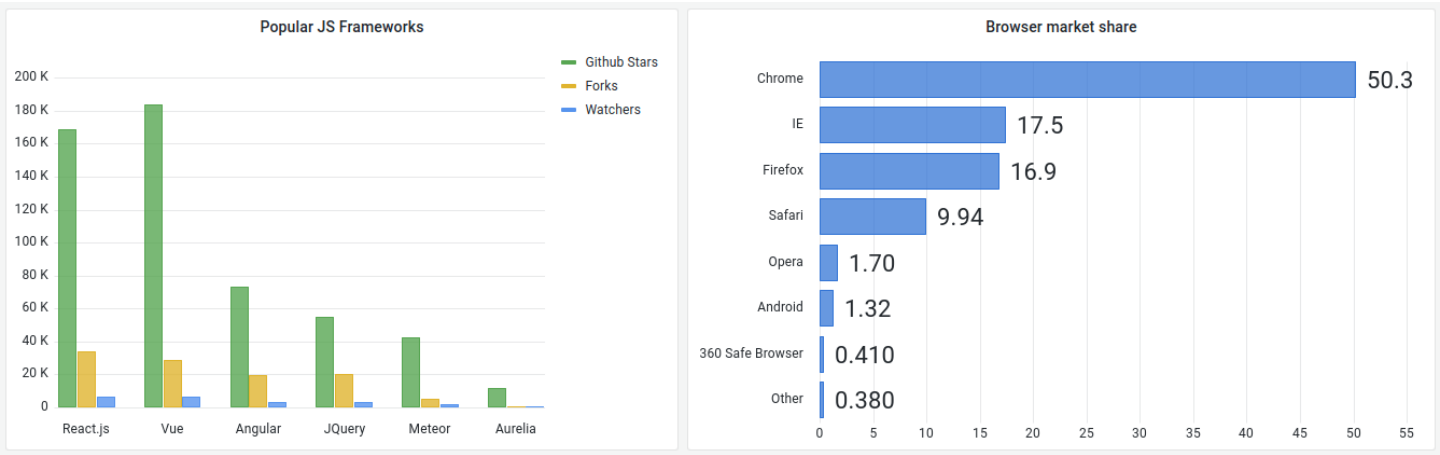
以下のセクションでは、視覚化の例を示します。

グラフ

時間ベースの折れ線、面積、棒グラフの場合は、デフォルトの時系列の視覚化をお勧めします。

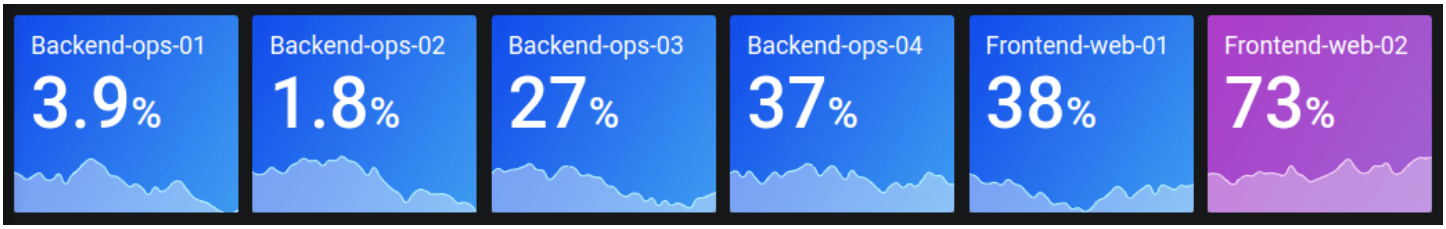


カテゴリ別データには、棒グラフを使用します。



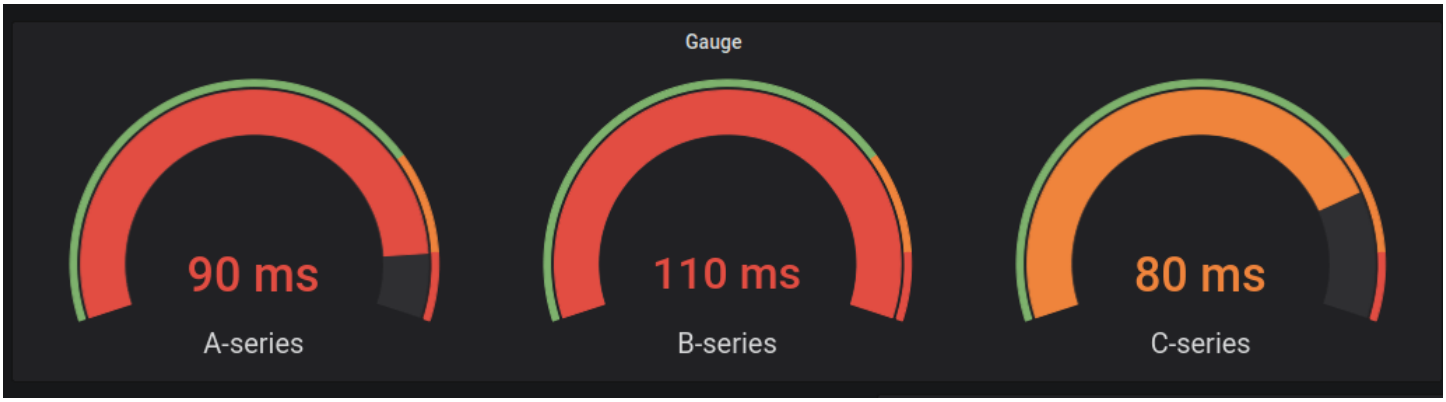
大規模な数値と統計

統計の視覚化は、オプションのグラフスパークラインを含む 1 つの大きな統計値を表示します。しきい値またはカラースケールを使用して、背景または値の色を制御できます。

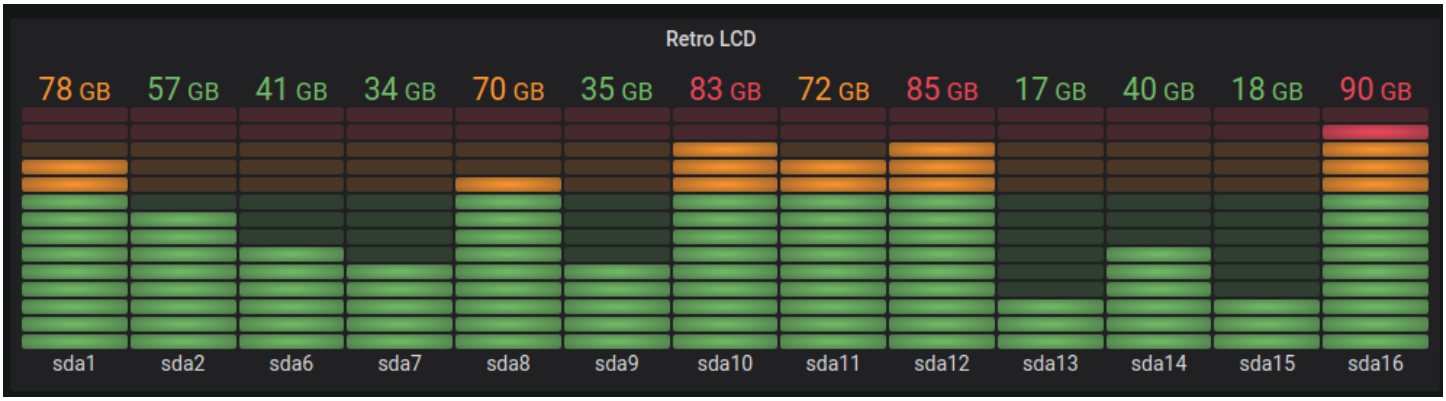


ゲージ

最小値と最大値に関連する値を提示する場合は、2 つのオプションがあります。1 つ目は、標準の放射状ゲージです。

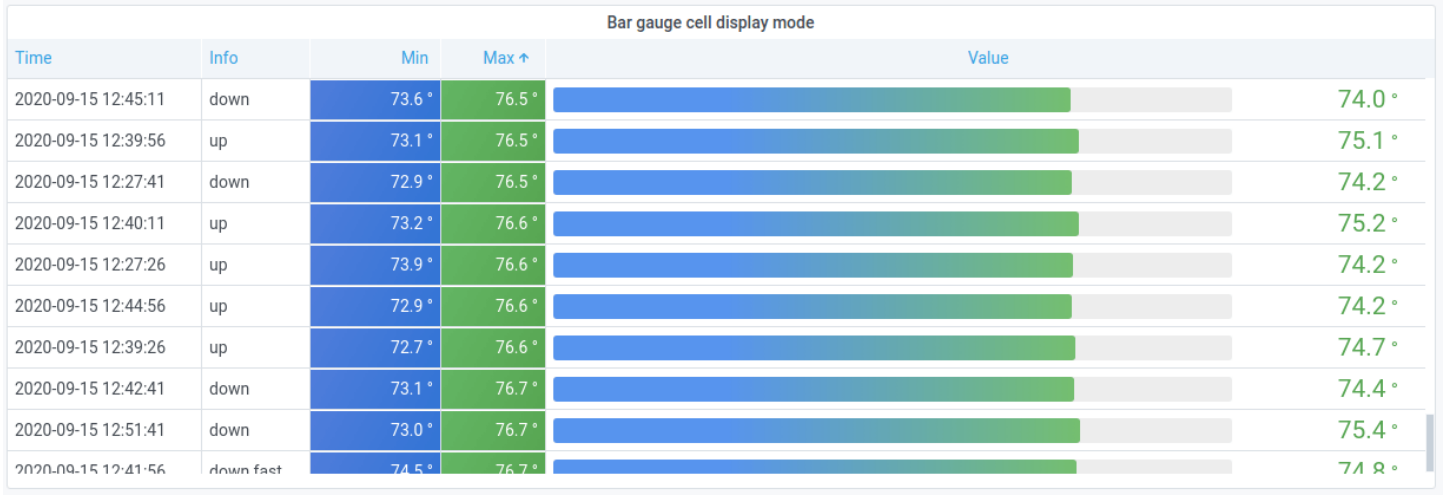


2 つ目として、Grafana には、3 つの異なる表示モードを持つ水平または垂直の棒ゲージもあります。



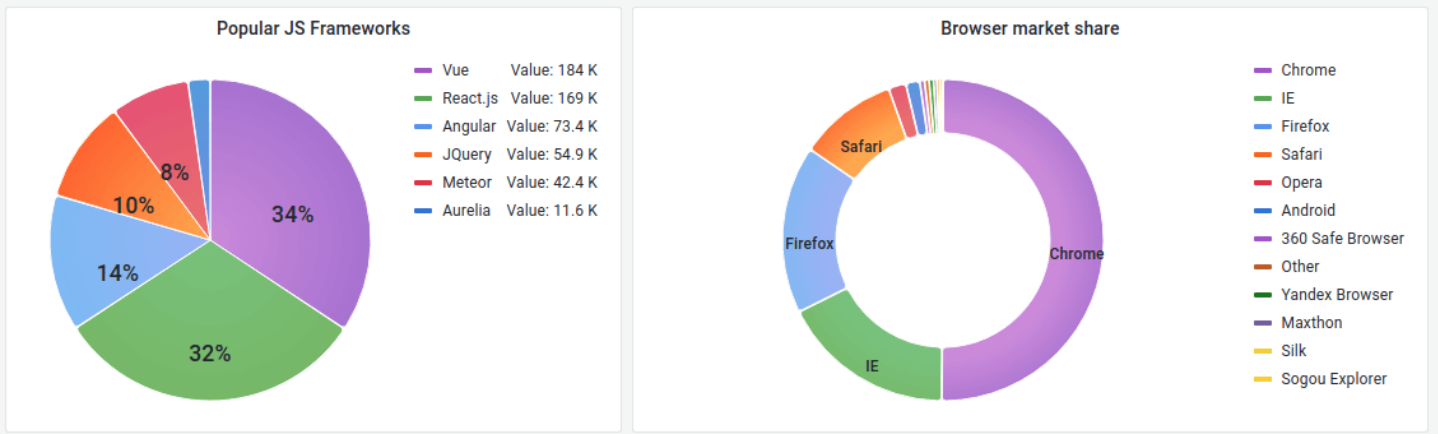
[テーブル]

テーブルレイアウトにデータを表示するには、テーブルの視覚化を使用します。



円グラフ

1 つ以上のクエリから誘導された系列または値は、相互に関連するため、円グラフの視覚化を用いて表示します。



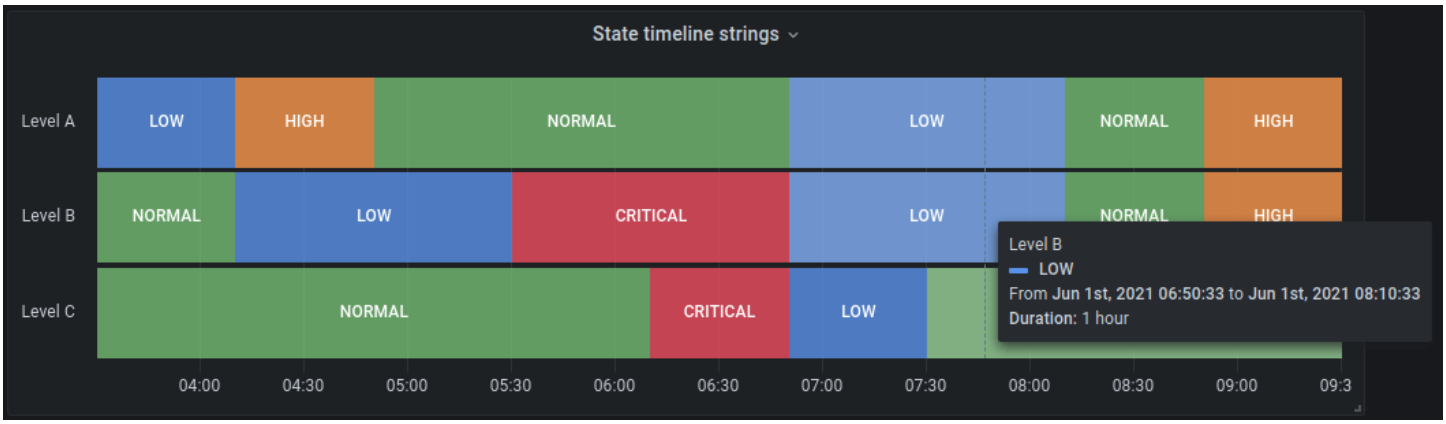
ヒートマップ

時間の経過に伴う値の分布を表示するには、ヒートマップの視覚化を使用します。



状態タイムライン

状態タイムラインには、時間の経過に伴う個別の状態変化が表示されます。時系列で使用すると、しきい値を使用して数値を離散状態の領域に変換します。



アラートリスト

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートリストを使用してアラートを表示します。現在の状態を表示するようにリストを設定できます。アラートの詳細については、「Grafana バージョン 10 のアラート」を参照してください。

この設定で、視覚化する内容を絞り込みます。

オプション

- グループモード – [デフォルトのグループ化] から選択して、アラートルール別にグループ化されたアラートインスタンスを表示します。また、[カスタムグループ化] では、ラベルのカスタムセット別にアラートインスタンスをグループ化します。
- [最大項目数] – 一覧表示するアラートの最大数を設定します。
- [ソート順序] – アラートをどのような順番で表示するかを選択します。
  - アルファベット (昇順) – アルファベット順
  - アルファベット (降順) – 逆アルファベット順
  - [重要性] – 以下を参照。1 が最も高い。
    - `alerting` または `firing`: 1
    - `no_data`: 2
    - `pending`: 3
    - `ok`: 4
    - `paused` または `inactive`: 5
  - [時間 (昇順)] – 最新のアクティブなアラートインスタンスが最初に表示されます。
  - [時間 (降順] – 最も古いアクティブなアラートインスタンスが最初に表示されます。
- [このダッシュボードからのアラート] – アラートリストが含まれているダッシュボードからのみアラートを表示します。

フィルター

これらのオプションを使用すると、表示されるアラートを、選択したクエリ、フォルダ、またはタグに一致するアラートのみに制限できます。

- [アラート名] – アラート名クエリを入力します。
- [アラートインスタンスラベル] – ラベルクエリを使用してアラートインスタンスをフィルタリングします。例えば、`{severity="critical", instance=~"cluster-us-.+"}` と指定します。
- [フォルダ] – フォルダを選択します。選択したフォルダ内のダッシュボードからのアラートのみが表示されます。
- [データソース] – 選択したデータソースからアラートをフィルタリングします。

状態フィルター

このパネルに表示するアラートの状態を選択します。

- アラート中/発生中
- 保留中
- データなし
- 普通
- エラー

注釈リスト

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

注釈リストには、注釈付きデータの表示に使用できる注釈の一覧が表示されます。タグと現在のダッシュボードに基づいてリストをフィルタリングできるさまざまなオプションを使用できます。

注釈クエリ

以下のオプションは、注釈のリストに対するソースクエリを制御します。

[クエリフィルター]

クエリフィルターを使用して、組織内のすべてのダッシュボードまたはこのパネルがある現在のダッシュボードから注釈のリストを作成します。これには次のオプションがあります。

- すべてのダッシュボード - 現在の組織内のすべてのダッシュボードの注釈を一覧表示します。
- このダッシュボード - リストを現在のダッシュボードの注釈に制限します。

[時間範囲]

時間範囲オプションを使用して、リストを現在の時間範囲に制限するかどうかを指定します。これには次のオプションがあります。

- [なし] - 注釈クエリの時間範囲の制限はありません。
- このダッシュボード - リストを、注釈リストパネルが利用可能なダッシュボードの時間範囲に制限します。

[タグ]

タグオプションを使用して、注釈をタグでフィルタリングします。リストを絞り込むために複数のタグを追加できます。

> **Note**
>
> 必要に応じて、タグリストを空のままにして、パネル自体の結果の一部としてリストされているタグを選択して、その場でフィルタリングします。

[制限]

制限オプションを使用して、返される結果の数を制限します。

[Display] (表示)

これらのオプションは、注釈パネル表示に含まれる追加のメタデータを制御します。

[ユーザーの表示]

このオプションを使用して、注釈を作成したユーザーを表示または非表示にします。

[時間の表示]

このオプションを使用して、注釈の作成時刻を表示または非表示にします。

[タグの表示]

このオプションを使用して、注釈に関連付けられたタグを表示または非表示にします。[NB]: タグを使用して、視覚化自体の注釈リストをその場でフィルタリングできます。

リンク動作

[リンクの対象]

このオプションを使用して、注釈付きデータを表示する方法を選択します。これには次のオプションがあります。

- [Panel] (パネル) - このオプションは、対応する注釈を含むパネルのフル画面表示に直接移動します。
- ダッシュボード - このオプションは、全ダッシュボードのコンテキストで注釈に焦点を当てます。

[前に時間指定]

このオプションを使用して、注釈の前に時間範囲を設定します。「1h」= 1 時間、「10m」= 10 分などの期間文字列値を使用します。

[後に時間指定]

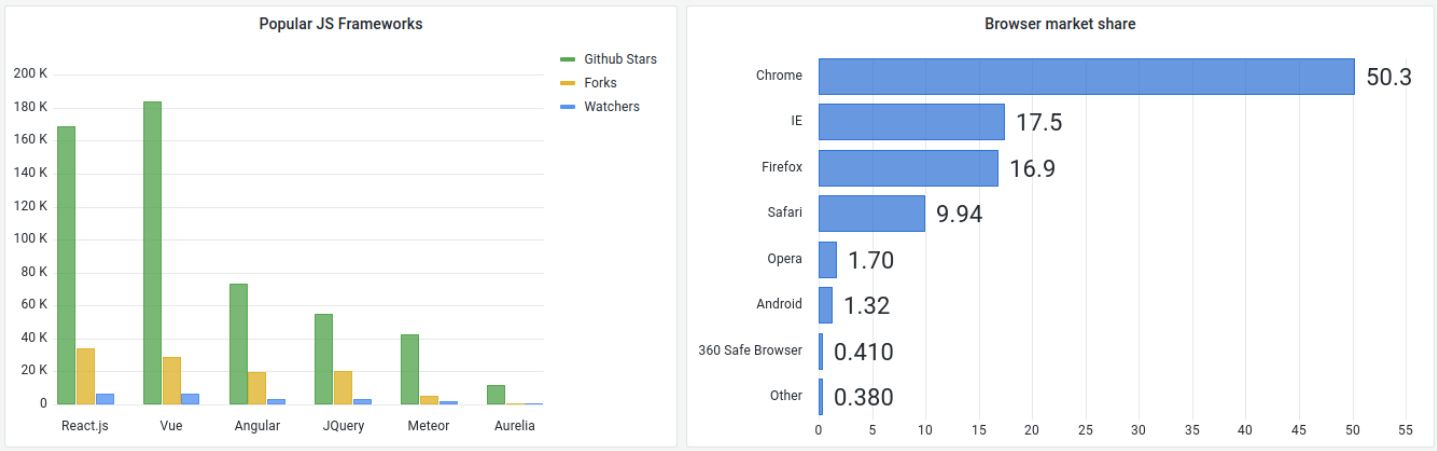
このオプションを使用して、注釈の後に時間範囲を設定します。

棒グラフ

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

棒グラフを使用すると、カテゴリ別のデータをグラフ化できます。



サポートされているデータ形式

サポートされるデータフレームは 1 つだけで、X 軸または Y 軸の項目として使用する文字列フィールドが少なくとも 1 つ、および数値フィールドが 1 つ以上必要です。以下は、データ形式の例です。

ブラウザ	市場シェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合は、パネルにグループ化されたバーが表示されます。

時系列または複数の結果セットの視覚化

複数の時系列またはテーブルがある場合は、まず結合を使用して結合するか、変換を減らす必要があります。例えば、複数の時系列があり、その最終値と最大値を比較する場合は、変換の [削減] を追加し、[計算] で [最大] と [Last] をオプションとして指定します。

棒グラフのオプション

これらのオプションで、視覚化する内容を絞り込みます。

[Orientation] (向き)

- [自動] – Grafana は、パネルのディメンションに基づいてバーの向きを決定します。
- [水平] – X 軸をカテゴリ軸にします。
- [垂直] – Y 軸をカテゴリ軸にします。

[X 軸の目盛ラベルの回転]

グラフが垂直方向の場合、この設定はバーの下のラベルを回転させます。この設定は、棒グラフラベルが長く重複している場合に役立ちます。

[X 軸の目盛ラベルの最大長]

棒グラフラベルの最大長を設定します。最大長を超えるラベルは省略記号で切り捨てられます。

[バーラベルの最小間隔]

バーラベル間の最小間隔を設定します。

[値の表示]

値がバーの上部または左側に表示されるかどうかを制御します。

- [自動] – スペースがある場合は値が表示されます。
- [常に表示] – 常に値を表示します。
- [常に非表示] – 常に値を表示しません。

[積み上げ]

棒グラフの積み上げを制御します。

- [Off] (オフ) – バーは積み重ねられません。
- [普通] – バーは積み重ねられます。
- [パーセント] – バーは積み重ねられ、各バーの高さは積み重ねの合計の高さの割合です。

[グループの幅]

グループの幅を制御します。

- 0 = Minimum width
- 1 = Maximum width

[バーの幅] ()

バーの幅を制御します。

- 0 = Minimum width
- 1 = Maximum width

[Bar radius] (バーの丸角半径)

バーの丸角半径を制御します。

- 0 = Minimum radius
- 0.5 = Maximum radius

[カーソル上の全領域をハイライト]

バーにカーソルを合わせると、バーの周囲全体が強調表示されるかどうかを制御します。

[線の幅]

棒の線幅を制御します。

[塗りつぶしの不透明度]

棒の塗りつぶしの不透明度を制御します。

[グラデーションモード]

グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。

グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーションの塗りつぶしなし。これはデフォルトの設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。

- [色相] – グラデーション色は線の色相に基づいて生成されます。

[ツールチップモード]

視覚化にカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、グラフ上でカーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

> **Note**
>
> オーバーライドを使用して、ツールチップから個々の系列を非表示にできます。

[テキストサイズ]

棒グラフのテキストのサイズを変更する値を入力します。

凡例オプション

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。詳細については、「凡例を設定する」を参照してください。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を配置する場所を選択します。

- [Bottom] – グラフの下。

- [右側] – グラフの右側。

[凡例値]

凡例に表示する系列データ値または標準計算を選択します。複数選択できます。詳細については、「凡例を設定する」を参照してください。

軸オプション

次のフィールド設定を使用して、軸の表示方法を絞り込みます。一部のフィールドオプションは、編集するフィールドオプションボックスの外部をクリックするか、Enter キーを押すまで、視覚的表現には影響しません。

[配置]

Y 軸の配置を設定します。

- [自動] – Grafana は、Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。
- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。

軸を選択的に非表示にするには、特定のフィールドを対象とするフィールドオーバーライドを追加します。

ラベル

Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。

[幅]

軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。これにより、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。

[ソフト最小値とソフト最大値]

Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて Y 軸の範囲を自動的に設定します。

ソフト最小値とソフト最大値の設定により、データがほぼフラットなときに、小さな変動が大きな変動に見えるのを防ぎます。また、標準の最小値と最大値のフィールドオプションから派生したハード最小値または最大値により、定義されたポイントを超えたスパイクを切り取ることで、断続的なスパイクが有用な詳細を平坦化することを防げます。

標準最小/最大値オプションを設定して、Y 軸のハード制限を定義できます。詳細については、「標準オプションを設定する」を参照してください。

[Display multiple y-axes] (複数の Y 軸を表示)

場合によっては、複数の Y 軸を表示したい場合があります。例えば、経時的な温度と湿度の両方を示すデータセットがある場合、これらの 2 つの系列で異なる単位を持つ 2 つの y 軸を表示したいとします。

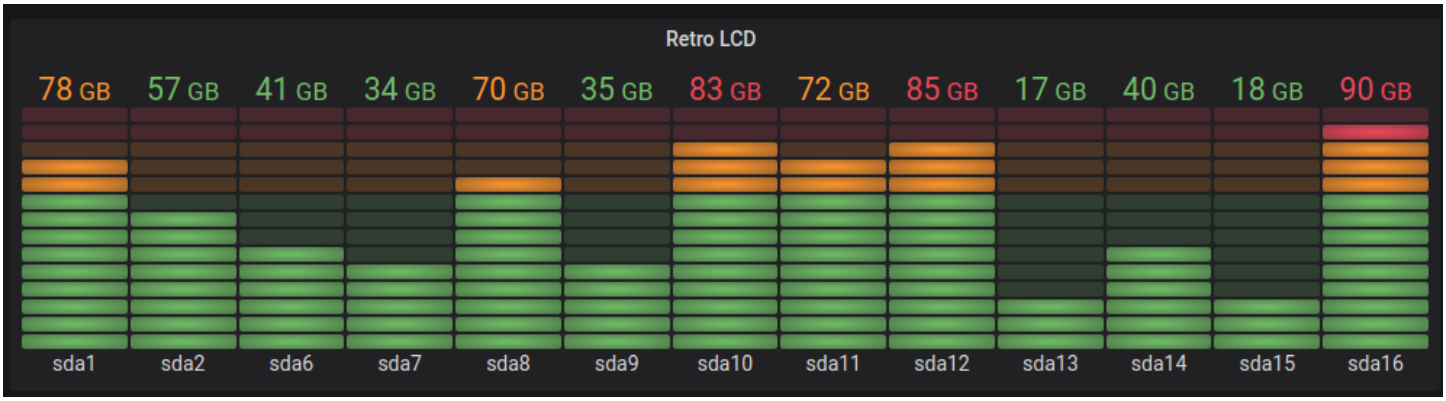
これを行うには、フィールドオーバーライドを追加します。必要な回数だけステップに従うことで、必要な数の y 軸を追加できます。

棒ゲージ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

棒ゲージは、すべてのフィールドを 1 つの値にまとめて表示することで、データを簡素化します。Grafana がまとめるにあたり、計算方法を選択します。

このパネルには、クエリの結果によって、1 つまたは複数の棒ゲージを表示できます (データが複数の系列、行、または列で構成されている場合など)。



値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示]

Grafana がデータを表示する方法を選択します。

[計算]

すべての行を元に計算した値を表示します。

- [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算の一覧については、「計算タイプ」を参照してください。
- [フィールド] – パネルに表示されるフィールドを選択します。

[すべての値]

行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

- [制限] – 表示する行の最大数。デフォルトは 5,000 です。
- [フィールド] – パネルに表示されるフィールドを選択します。

棒ゲージオプション

棒ゲージの表示方法を設定します。

[Orientation] (向き)

積み重ね方向を選択します。

- [自動] – Grafana が最適な方向を選択します。
- [水平] – バーは水平方向に左から右に伸びます。
- [垂直] – バーは垂直に下から上に伸びます。

Display mode (表示モード)

表示モードを選択します。

- グラデーション] – しきい値レベルはグラデーションを定義します。
- [Retro LCD] – ゲージを小さなセルに分け、点灯または消灯させて表示します。
- [基本] – 一致するしきい値に基づいて単一色で表示します。

[Value display] (値の表示)

値の表示モードを選択します。

- [Value color] – 値の色は値によって決まります。
- [テキスト色] – 値の色はデフォルトのテキスト色です。
- [非表示] – 値は非表示です。

[名前配置]

名前配置モードを選択します。

> **Note**
>
> このオプションは、棒ゲージの向きが水平の場合にのみ適用されます。棒ゲージが垂直方向にある場合、名前は常に各バーゲージの下に配置されます。

- [自動] – Grafana が最適な配置を決定します。
- [Top] – 名前は各棒ゲージの上部に配置されます。
- [左] – 名前は各棒ゲージの左側に配置されます。

[塗りつぶされていない領域を表示]

バーの塗りつぶされていない領域を濃い灰色で表示する場合は、このオプションを選択します。Retro LCD 表示モードには適用されません。

[Bar size] (バーサイズ)

バーサイズモードを選択します。

- [自動] – Grafana が最適な棒ゲージサイズを決定します。
- [Manual] (手動) – 棒ゲージのサイズを手動で設定します。

[最小幅]

ゲージが垂直方向のときは、バーの列の最小幅を制限します。

大量のデータがある場合、X 軸スクロールバーを自動的に表示します。

> **Note**
> このオプションは、バーサイズが手動に設定されている場合にのみ適用されます。

[最小高さ]

ゲージが水平方向のときは、バーの行の最小高さを制限します。

大量のデータがある場合、Y 軸スクロールバーを自動的に表示します。

> **Note**
> このオプションは、バーサイズが手動に設定されている場合にのみ適用されます。

[最大高さ]

ゲージが水平方向のときは、バーの行の最大高さを制限します。

大量のデータがある場合、Y 軸スクロールバーを自動的に表示します。

> **Note**
> このオプションは、バーサイズが手動に設定されている場合にのみ適用されます。

ローソク足チャート

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ローソク足チャートの作成により、価格変動に焦点を当てた多数の一貫したディメンションを含むデータを視覚化できます。ローソク足パネルには、OHLC (Open-High-Low-Close: 始値、高値、安値、終値) モードと、時系列データに基づく追加のディメンションのサポートが含まれています。



ローソク足チャートの作成は、時系列 の基礎に基づいて構築され、多くの一般的な設定が含まれています。

モード

モードオプションを使用すると、視覚化に使用されるディメンションを切り替えられます。

- [Candles] (ローソク) – パネルのディメンションを、ローソク足チャートの作成で使用される始値、高値、安値、終値のディメンションに制限します。
- [Volume] (ボリューム) – パネルディメンションをボリュームディメンションに制限します。
- [Both] (両方) - ローソク足パネルのデフォルトの動作です。これには、ローソク足とボリュームの両方の視覚化が含まれます。

Candle style (ローソクのスタイル)

- [Candles] – デフォルトの表示スタイルでは、始値ディメンションと終値ディメンションとの間にローソクスタイルの視覚化が作成されます。
- [OHLC 棒] – 4 つのコアディメンションである始値、高値、安値、終値を表示します。

Color strategy (カラー戦略)

- [始値から] – デフォルトの動作。このモードは、期間内の料金変動が正の場合、Up カラー (下部) を使用します。つまり、クローズ時の値がオープン時の値以上である場合、Up カラーが使用されます。
- [前の終値以降] は、ローソクの色が中間期の価格変動または値の変化に基づいている代替の表示方法です。つまり、オープン時の値がクローズ時の以前の値より大きい場合、Up カラーが使用されます。オープン時の値がクローズ時の以前の値よりも低い場合、Down カラーが使用されます。このオプションでは、中空ローソク足グラフ作成モードもトリガーされます。中空ローソク足は、期間内の変動が正 (値はオープン時よりもクローズ時が高い) であることを示します。塗りつぶされたローソク足は、期間内の変動が負 (値はオープン時よりもクローズ時が低い) であることを示します。詳細については、「違いの説明」を参照してください。

アップカラーとダウンカラー

[Up color] と [Down color] オプションは、料金変動がアップまたはダウンするときにどの色を使用するかを選択します。上記のカラー戦略は、ローソクまたは OHLC バーの色を選択するために、期間内または中間期の料金変動を使用するかどうかを決定します。

始値、高値、安値、終値

ローソク足パネルは、フィールドを適切なディメンションにマッピングしようとします。

- [始値] は、指定された期間の開始値に対応します。
- [高値] は、指定された期間の最大値に対応します。

- [安値] は、指定された期間の最小値に対応します。
- [終値] は、指定された期間の最終 (終了) 値に対応します。
- [ボリューム] は、特定の期間のサンプル数に対応します。(取引数など)

> **Note**
>
> ローソク足の凡例は、これらの値を表示しません。

これらのディメンションを適切にマッピングするには、データからのクエリ結果テーブルに少なくとも次の列が含まれている必要があります。

- `timestamp`
- `open`
- `high`
- `low`
- `close`

何らかの理由でデータをこれらのディメンションにマッピングできない場合 (列名が同じでない場合など)、パネルエディタのローソク足オプションの下にある始値、高値、安値、および終値フィールドを使用して手動でマッピングできます。

追加フィールド

追加フィールド

ローソク足パネルは時系列の視覚化に基づいています。始値、高値、安値、終値、ボリューム以外の追加のデータディメンションを視覚化できます。[含める] オプションと [無視] オプションを使用すると、時系列 で使用できるのと同じスタイルと設定を使用して、単純な移動平均、ボリンジャーバンドなど、含まれている他のデータを視覚化できます。

キャンバス

> **Warning**
>
> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

キャンバスは Grafana の権限とカスタム要素の柔軟性を組み合わせています。キャンバスは、静的レイアウトと動的レイアウトに要素を明示的に配置できる拡張可能なフォームビルドパネルです。これにより、Grafana の UI 内全体で、標準の Grafana パネルでは不可能な方法でカスタムビジュアライゼーションとオーバーレイデータを設計できます。一般的な UI およびウェブ設計ツールを使用したことがある場合、キャンバスパネルの設計は非常に馴染みのあるものになります。

[Elements] (要素)

これらの要素はキャンバスに追加できます。複数の異なる種類の要素を追加すると、他の視覚化では不可能な方法で視覚化をカスタマイズできます。

メトリクス値

メトリクス値要素を使用すると、キャンバスに表示するデータを簡単に選択できます。この要素には、コンテキストメニューの編集オプションを通じて、またはパネルをダブルクリックしてトリガーできる一意の編集モードがあります。編集モードでは、表示するフィールドデータを選択できます。

[テキスト]

テキスト要素を使用すると、キャンバスにテキストを簡単に追加できます。また、要素は、ダブルクリックまたはコンテキストメニューの編集メニューオプションによってトリガーされる編集モードもサポートします。

[Ellipse] (楕円)

楕円要素を使用すると、基本的な楕円をキャンバスに追加できます。楕円要素はテキスト (固定データとフィールドデータの両方) を表示でき、背景色はデータしきい値に基づいて変更できます。

[Rectangle] (長方形)

長方形要素を使用すると、基本的な長方形をキャンバスに追加できます。長方形要素はテキスト (固定データとフィールドデータの両方) を表示でき、背景色はデータしきい値に基づいて変更できます。

[Icon] (アイコン)

アイコン要素を使用すると、サポートされているアイコンをキャンバスに追加できます。アイコンは、しきい値または値マッピングに基づいて色を設定できます。

[Server] (サーバー)

サーバー要素を使用すると、単一のサーバー、サーバーのスタック、データベース、またはターミナルを簡単に表現できます。サーバー要素は、ステータスの色、電球の色、電球の点滅レートをサポートし、すべて固定値またはフィールド値で設定できます。

[Button] (ボタン)

ボタン要素を使用すると、基本的なボタンをキャンバスに追加できます。ボタン要素は、認証されていない基本的な API コールのトリガーをサポートします。API 設定は、ボタン要素エディタにあります。API エディタでテンプレート変数を渡すこともできます。

> **Note**
>
> ボタンを選択すると、インライン編集が無効になっている場合にのみ API コールがトリガーされます。「Canvas の編集」を参照してください。

Connections

キャンバスを構築するときは、要素を連結してより複雑な視覚化を作成できます。ある要素の接続アンカーから別の要素の接続アンカーにドラッグすることで、接続を作成できます。キャンバスの背景への接続を作成することもできます。接続アンカーは、要素にカーソルを合わせ、インライン編集がオンになると表示されます。接続を削除するには、接続を選択し、`Delete` または `Backspace` を押します。

固定値またはフィールド値に基づいて、接続のサイズと色の両方を設定できます。これを行うには、パネル編集モードに入り、接続を選択し、パネルエディタで接続のプロパティを変更します。

Canvas の編集

[インラインエディタ]

ダッシュボードモードのコンテキストでは、キャンバスをインラインで編集できます。

Pan and zoom (パンとズーム)

キャンバスのパンとズームをオンにします。これにより、より複雑な設計を作成およびナビゲートできます。

> **Note**
>
> パンとズームは現在 Grafana Labs によってプレビューされています。サポートには制限があり、一般提供の前に重要な変更が発生する可能性があります。

コンテキストメニュー

コンテキストメニューを使用すると、一般的なタスクを迅速かつ効率的に実行できます。サポートされている機能には、インラインエディタの開閉、要素の複製、要素の削除などがあります。

コンテキストメニューは、パネルまたは特定のキャンバス要素に対する右クリックアクション (または同等のもの) によってトリガーされます。

パネルを右クリックすると、背景イメージを設定し、キャンバスに要素を簡単に追加できます。

要素を右クリックすると、要素を編集、削除、複製したり、要素のレイヤーの位置を変更したりできます。

キャンバスオプション

インライン編集

インライン編集トグルを使用すると、キャンバスパネルをロックまたはロック解除できます。オフにすると、キャンバスパネルがロックされ、要素が所定の位置にフリーズし、意図しない変更を防止できます。

データリンク

キャンバスはデータリンクをサポートしています。以下の手順に従って、メトリクス値要素のデータリンクを作成し、フィールド名を使用するすべての要素に対して表示できます。

要素のデータリンクを作成するには

1. フィールド値に関連付ける要素を設定します。
2. インライン編集トグルをオフにします。
3. [名前付きフィールド] のオーバーライドを作成し、リストから要素フィールド名を選択します。

4. [オーバーライドプロパティの追加] ボタンを選択します。
5. リストから [Datalinks > Datalinks] を選択します。
6. [+ リンクを追加] を選択し、データリンクのタイトルと URL を追加します。
7. 要素にカーソルを合わせると、データリンクのツールチップが表示されます。
8. データリンクを開くことができる要素を選択します。

複数の要素が同じフィールド名を使用し、どの要素がデータリンクを表示するかを制御したい場合は、計算からのフィールドの追加 変換を使用して一意のフィールド名を作成できます。変換で作成したエイリアスは、要素とともに使用できるフィールドとして表示されます。

ダッシュボードの一覧表示

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードリストを使用すると、他のダッシュボードへの動的リンクを表示できます。リストは、スター付きダッシュボード、最近表示されたダッシュボード、検索クエリ、ダッシュボードタグを使用できるよう設定が可能です。

ダッシュボードのロードごとに、このパネルはダッシュボードリストに対してクエリを実行し、常に最新の結果を表示します。

オプション

これらのオプションで、視覚化する内容を絞り込みます。

- [現在の時間範囲を含める] – このオプションを選択すると、現在のダッシュボードの時間範囲がダッシュボードリンクに反映されます。ユーザーがリンクを選択すると、リンクされたダッシュボードが開き、指定された時間範囲が既に設定されています。
- [現在のテンプレート変数値を含める] – このオプションを選択すると、リンクに現在クエリパラメータとして使用されているテンプレート変数を含めることができます。ユーザーがリンクを選

択すると、リンクされたダッシュボード内の一致するテンプレートがリストからの値に設定されます。詳細については、「ダッシュボード URL 変数」を参照してください。

- [Starred] (スター付き) – スター付きダッシュボードをアルファベット順に表示します。
- [最近表示] – 最近表示したダッシュボードをアルファベット順に表示します。
- [検索] – 検索クエリまたはタグでダッシュボードを表示します。クエリまたはタグに少なくとも 1 つの値を入力する必要があります。クエリフィールドとタグフィールドでは、`$my_var` や `${my_var}` など、変数補間がサポートされています。
- [見出しの表示] – 選択したリスト選択 (スター付き、最近表示、検索) が見出しとして表示されます。
- [最大項目数] – セクションごとに一覧表示する項目の最大数を設定します。例えば、これをデフォルト値の 10 のままにして、スター付きダッシュボードと最近表示ダッシュボードを表示した場合、パネルには各セクションに 10 個、合計 20 個のダッシュボードが表示されます。

[検索]

これらのオプションは、検索オプションが選択されている場合にのみ適用されます。

- [クエリ] – 検索するクエリを入力します。クエリでは大文字と小文字が区別されず、部分的な値も受け入れられます。
- [フォルダ] – 表示するダッシュボードフォルダを選択します。
- [タグ] – ここに検索するタグを入力します。既存のタグは入力時に表示されず、大文字と小文字が区別されます。

> **Note**
>
> 複数のタグと文字列が表示されると、ダッシュボードリストにすべての条件に一致するタグと文字列が表示されます。

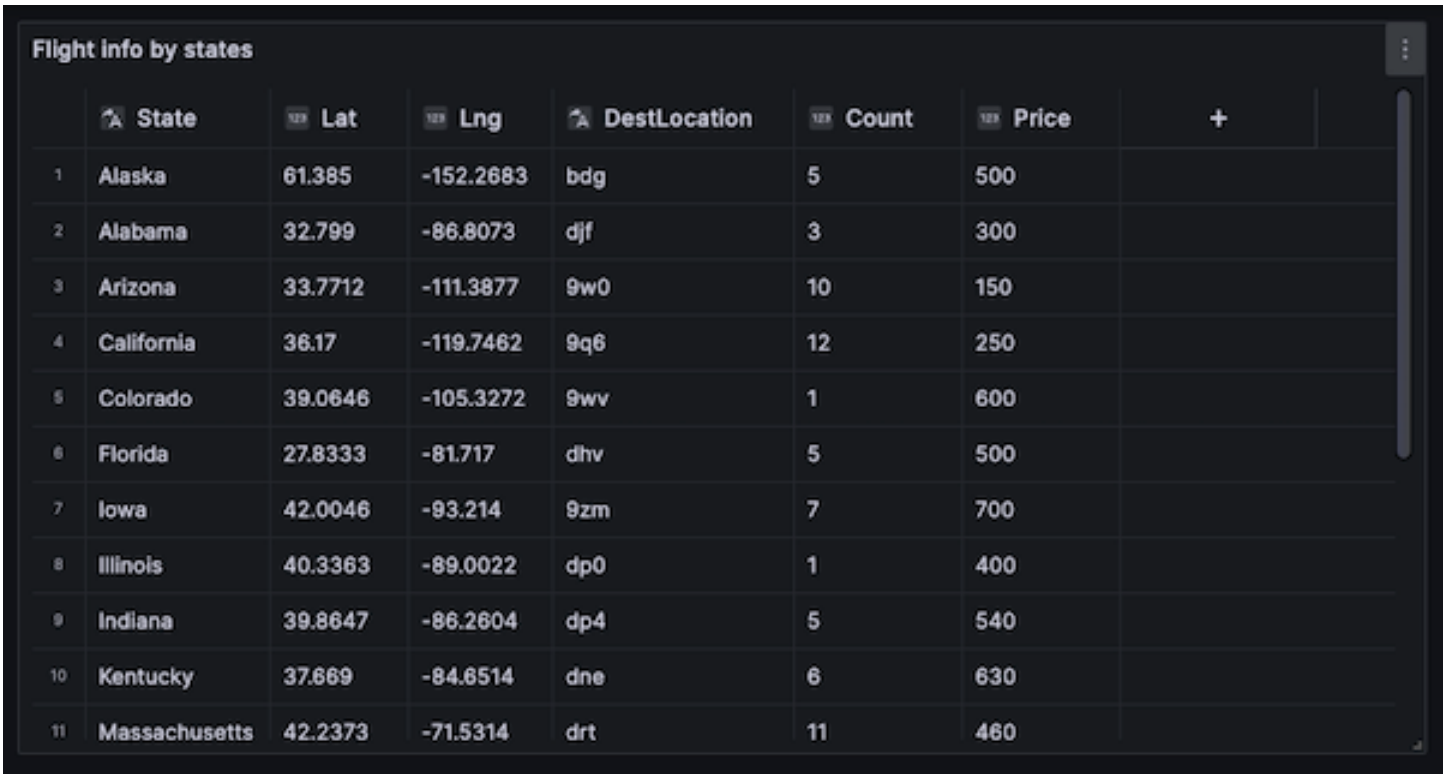
データグリッド

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

> **Note**
>
> データグリッドの視覚化は現在、Grafana Labs によるプレビュー段階です。サポートには制限があり、一般提供の前に重要な変更が発生する可能性があります。

データグリッドでは、Grafana 内でデータの作成、編集、ファインチューニングが可能です。そのため、このパネルはダッシュボード内の他のパネルのデータソースとして機能します。



	State	Lat	Lng	DestLocation	Count	Price
1	Alaska	61.385	-152.2683	bdg	5	500
2	Alabama	32.799	-86.8073	djf	3	300
3	Arizona	33.7712	-111.3877	9w0	10	150
4	California	36.17	-119.7462	9q6	12	250
5	Colorado	39.0646	-105.3272	9wv	1	600
6	Florida	27.8333	-81.717	dhv	5	500
7	Iowa	42.0046	-93.214	9zm	7	700
8	Illinois	40.3363	-89.0022	dp0	1	400
9	Indiana	39.8647	-86.2604	dp4	5	540
10	Kentucky	37.669	-84.6514	dne	6	630
11	Massachusetts	42.2373	-71.5314	drt	11	460

これにより、任意のデータソースからクエリされたデータを操作したり、白紙状態から開始したり、ドラッグアンドドロップされたファイルからデータを取得したりできます。その後、パネルを単純な表形式の視覚化として使用したり、データを変更したり、完全に削除して白紙状態を作成したりできます。

データセットを編集すると、組み込みの [Grafana] データソースを使用するようにデータソースが変更され、古いデータソース設定と関連するクエリが置き換えられます。また、現在のデータセットをダッシュボードモデルにコピーします。

その後、組み込みの ダッシュボード データソースを使用してデータグリッドデータを取得することで、パネルを他のパネルのデータソースとして使用できます。これにより、インタラクティブなダッシュボードエクスペリエンスが提供され、データを変更したり、他のパネルに反映された変更を確認したりできます。

Grafana および ダッシュボード データソースの詳細については、「特別なデータソース」を参照してください。

コンテキストメニュー

より合理化されたエクスペリエンスを提供するために、データグリッドには、セル、列ヘッダー、または行セレクタを右クリックしてアクセスできるコンテキストメニューがあります。データグリッドの状態に応じて、コンテキストメニューには次のようなさまざまなオプションが用意されています。

- すべての行と列を削除またはクリアします。
- 既存のデータをすべて削除します (データグリッドを空白にします)。
- トリガー検索機能により、データセット内のキーワードを検索できます。

行または列を削除するとデータグリッドからデータが削除されますが、行または列をクリアするとセルからデータのみが削除され、行または列はそのまま残ります。

ヘッダーメニュー

ヘッダータイトルの横にあるドロップダウンアイコンを選択して、ヘッダーメニューにアクセスすることもできます。ここから、列を削除またはクリアするだけでなく、名前の変更、フリーズ、列のフィールドタイプの変換もできます。

系列の選択

複数の系列がある場合は、パネルオプションの [系列の選択] ドロップダウンを使用して、任意のデータセットを表示するようにデータグリッドを設定できます。

データグリッドの使用

データグリッドは、データを操作するさまざまな方法を提供します。行と列を編集、移動、クリア、削除したり、組み込みの検索機能を使用して特定のデータを検索したり、フィールドタイプを変換したり、特定の列の水平スクロールをフリーズしたりできます。

データの追加

新しい列または行を作成することで、データグリッドにデータを追加できます。

新しい列を追加するには

1. 既存のパネルで、最後の列の後にテーブルヘッダーの [+] ボタンを選択します。
2. 新しい列の名前を追加します。
3. フィールド外の任意の場所を選択するか、Enter を押して列を保存します。

これで、各セルにデータを追加できます。

新しい行を追加するには、最後の行の後に [+] ボタンを選択します。ボタンは最後の行の後にある各セルに存在し、これを選択すると、選択したセルもアクティブ化しながら、新しい行の作成がトリガーされます。

データの編集

必要に応じて列と行を移動できます。

列を移動するには

1. 移動する必要がある列のヘッダーを長押しします。
2. 列を目的の場所にドラッグします。
3. 列を放して移動を確定します。

行を移動するには、グリッドの左端にある数値列から行セレクタを選択して長押しし、目的の場所にドラッグします。行を放して移動を確定します。

複数のセルの選択

1 つのセルを選択し、他のセル間でドラッグすることで、複数のセルを選択できます。この選択を使用して、選択したセルからデータをコピーしたり、Delete キーを使用して削除したりできます。

複数の行または列の削除またはクリア

複数の行を削除またはクリアするには、以下を実行します。

複数の行または列を削除またはクリアするには

1. 数値列 (グリッドの最初の列左側) にカーソルを合わせると、行チェックボックスが表示されます。

2. 操作する行のチェックボックスを選択します。複数の連続する行を選択するには、Shift キーを押しながら、最初と最後の行をクリックします。連続しない行を選択するには、Ctrl (または Cmd) キーを押しながら目的の行をクリックします。
3. コンテキストメニューにアクセスするには、右クリック (または対応する操作) をします。
4. [行の削除] または [行のクリア] を選択します。

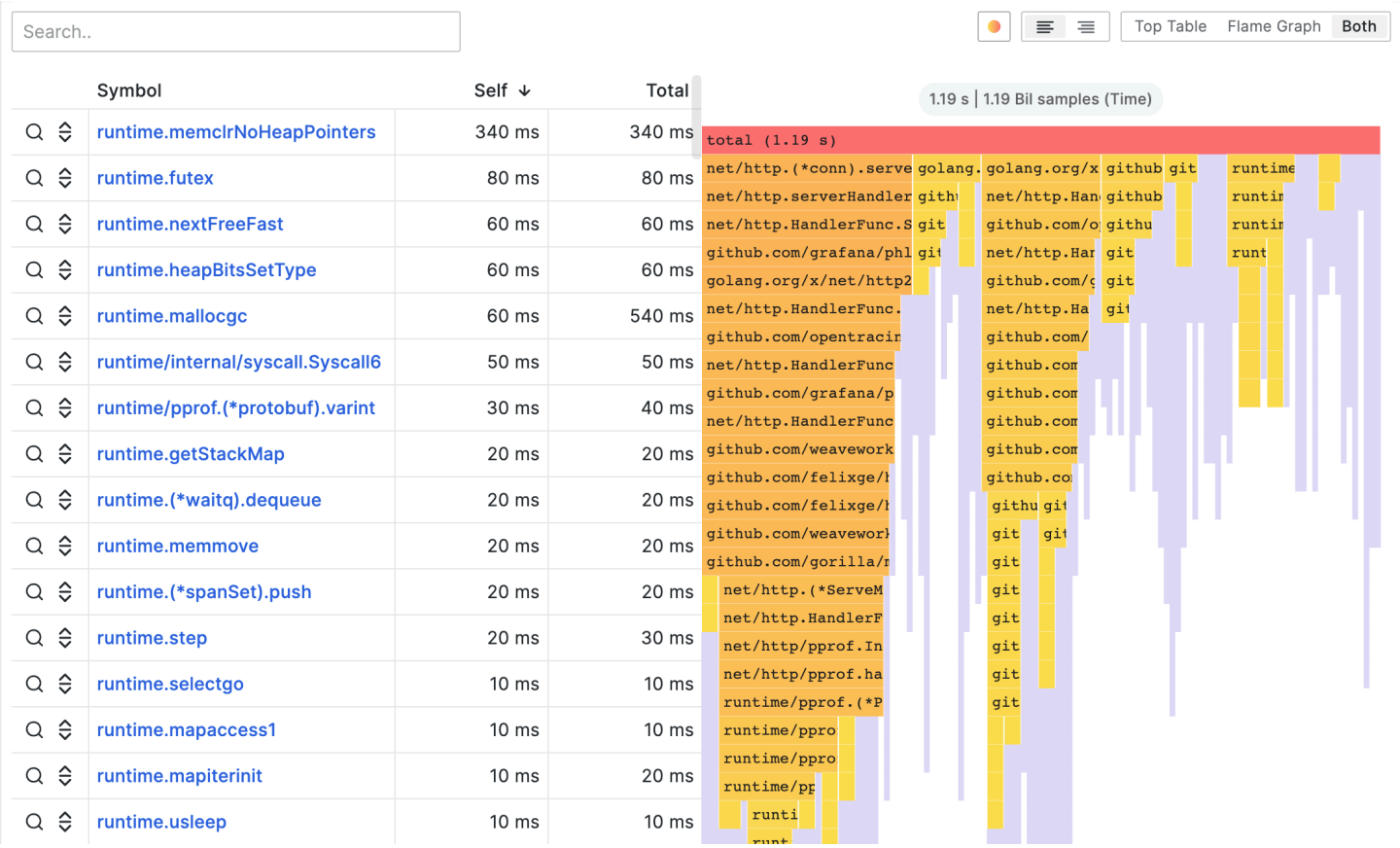
列ヘッダーをクリックすると、同じルールが列に適用されます。

すべての行を削除するには、データグリッドの左上隅にある [すべて選択] チェックボックスを使用します。これにより、すべての行が選択され、コンテキストメニューを使用して行を削除できます。

フレームグラフ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

フレームグラフを使用すると、プロファイリングデータを視覚化できます。この視覚化を使用すると、プロファイルをフレームグラフ、テーブル、またはその両方として表すことができます。



フレームグラフモード

フレームグラフは、プロファイリングデータの階層的な性質を活用します。データを形式に集約することで、メモリを測定するときに、CPU 時間、割り当てられたオブジェクト、スペースなど、どのコードパスが最もシステムリソースを消費しているかを簡単に確認できます。フレームグラフの各ブロックはスタック内の関数呼び出しを表し、ブロックの幅は値を表します。

グレーアウトされたセクションは、比較的小さな値を表す関数のセットであり、パフォーマンス上の理由から 1 つのセクションにまとめられます。

特定の関数にカーソルを合わせると、関数の値、合計値の割合、その関数のサンプル数など、その関数に関する追加データを示すツールチップを表示できます。

ドロップダウンアクション

関数をクリックして、追加のアクションを含む以下のドロップダウンメニューを表示できます。

- [Focus block] (フォーカスブロック) - フォーカスブロックを選択すると、ブロックまたは関数はフレームグラフの幅の 100% に設定され、そのすべての子関数が親関数の幅に対して更新された幅で表示されます。これにより、フレームグラフの小さな部分にドリルダウンしやすくなります。

- [コピー関数名] - [コピー関数名]を選択すると、ブロックが表す関数のフルネームがコピーされます。
- [サンドウィッチビュー] – サンドウィッチビューでは、クリックされた関数のコンテキストを表示できます。関数のすべての呼び出し元が上部に表示され、すべての呼び出し先が下部に表示されます。これは関数の集約コンテキストを示すため、関数がフレームグラフの複数の場所に存在する場合、すべてのコンテキストがサンドウィッチビューに表示され、集約されます。

ステータスバー

ステータスバーには、フレームグラフと現在適用されている変更に関するメタデータが表示されます。例えば、グラフのどの部分がフォーカスされているか、どの関数がサンドイッチビューに表示されるかなどです。ステータスバーのプルダウンにある [X] をクリックして、その変更を削除します。

ツールバー

[検索]

検索フィールドを使用して、特定の名前の関数を検索できます。検索に一致するフレームグラフのすべての関数では色はそのままで、残りの関数はグレーアウトされます。

Color schema picker (カラースキーマピッカー)

色付け関数を値またはパッケージ名で切り替えて、同じパッケージから関数を視覚的に結び付けることができます。

[テキスト配置]

ブロックに収まらない場合、テキストを左または右に揃えて、関数名のより重要な部分を表示できます。

視覚化ピッカー

フレームグラフのみ、テーブルのみ、またはその両方を同時に表示することを選択できます。

トップテーブルモード

トップテーブルは、プロファイルの関数を表形式で示しています。テーブルには、symbols、self、total (記号、自己、合計) の 3 つの列があります。テーブルはデフォルトでセルフタイムでソートされますが、列ヘッダーをクリックして合計時間またはシンボル名で並べ替えることができます。関数がプロファイル内の複数の場所に表示される場合、各行は特定の関数の集計値を表します。

行ごとに左側にアクションボタンもあります。最初のボタンは関数名を検索し、2 番目のボタンは関数のサンドウィッチビューを表示します。

Data API

フレームグラフをレンダリングするには、[ネストされたセットモデル]を使用してデータフレームのデータをフォーマットする必要があります。

ネストされたセットモデルは、フレームグラフの各項目が整数値、メタデータ、およびデータフレーム内の順序としてネストレベルによってのみエンコードされるようにします。つまり、項目の順序は重要であり、正しい必要があります。順序付けは、フレームグラフ内の項目の深さ優先探索であり、子配列のようにデータフレームに可変長値を使用せずにグラフを再作成します。

必須フィールド

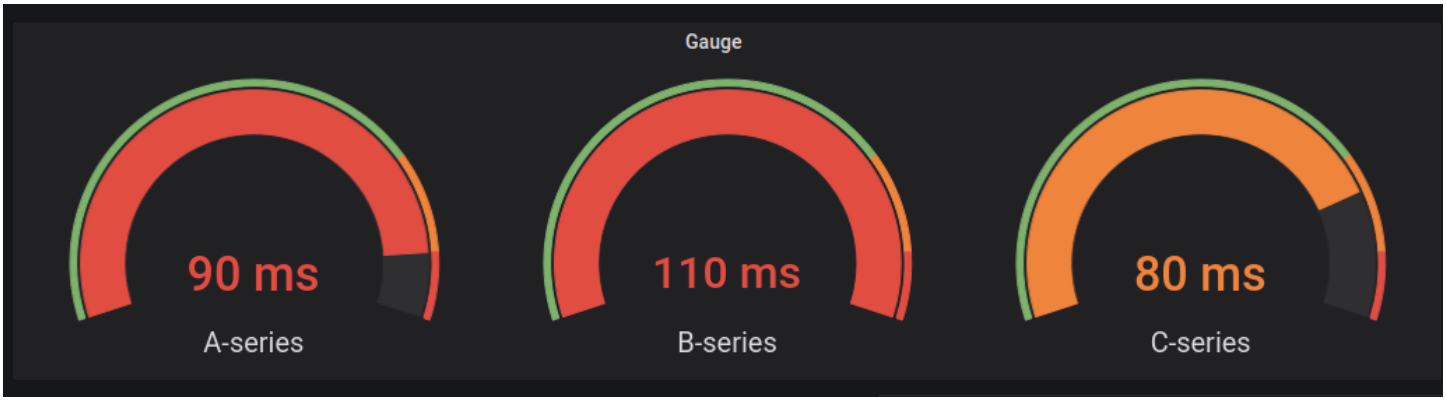
フィールド名	タイプ	説明
level	数値	項目のネストレベル。つまり、この項目とフレームグラフの上位項目の間にある項目の数。
値	数値	項目の絶対値または累積値。これは、グラフ内の項目の幅に変換される。
ラベル	文字列	特定の項目に表示されるラベル。
self	数値	通常、項目の累積値から直近の子の累積値の合計を引いた自己値。

ゲージ

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ゲージは、各系列、列、または行に対してゲージを繰り返すことができる単一値の視覚化です。



値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示]

Grafana がデータを表示する方法を選択します。

[計算]

すべての行を元に計算した値を表示します。

- [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算のリストについては、「計算タイプ」を参照してください。
- [フィールド] – パネルに表示するフィールドを選択します。

[すべての値]

行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

- [制限] – 表示する行の最大数。デフォルトは 5,000 です。
- [フィールド] – パネルに表示するフィールドを選択します。

ゲージ

ゲージの表示方法を設定します。

[Orientation] (向き)

積み重ね方向を選択します。

- [自動] – ゲージは行と列で表示されます。
- [水平] – ゲージは上から下に表示されます。
- [垂直] – ゲージは左から右に表示されます。

[しきい値ラベルの表示]

しきい値を表示するかどうかを制御します。

[しきい値マーカーの表示]

しきい値バンドを内部ゲージ値バンドの外部に表示するかどうかを制御します。

[ゲージサイズ]

ゲージサイズモードを選択します。

- [自動] – Grafana が最適なゲージサイズを決定します。
- [Manual] (手動) – ゲージサイズを手動で設定します。

[最小幅]

垂直方向ゲージの最小幅を設定します。

最小幅を設定すると、大量のデータがある場合、X 軸スクロールバーが自動的に表示されます。

> **Note**
> このオプションは、ゲージサイズが手動に設定されている場合にのみ適用されます。

[最小高さ]

水平方向のゲージの最小高さを設定します。

最小高さを設定すると、大量のデータがある場合、Y 軸スクロールバーが自動的に表示されます。

> **Note**
>
> このオプションは、ゲージサイズが手動に設定されている場合にのみ適用されます。

[Neutral] (ニュートラル)

すべてのゲージを埋める開始値を設定します。

テキストサイズ

ゲージテキストのサイズを調整します。

- [タイトル] – ゲージタイトルサイズの数値を入力します。
- [値] – ゲージ値サイズの数値を入力します。

ジオマップ

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ジオマップを使用すると、地理空間データを用いて世界地図を表示、カスタマイズできます。さまざまなオーバーレイスタイルとマップビュー設定を設定して、データの重要な位置情報に注目しやすくすることができます。

> **Note**
>
> が提供するベースマップレイヤーの上に独自の地理空間データを追加できます AWS。ベースマップレイヤーはすべて https://tiles.maps.search-services.aws.a2z.com から取得する必要があります。



マップビュー

マップビューは、ダッシュボードのロード時にマップの初期ビューを制御します。

[初期ビュー]

初期ビューは、パネルが最初にロードされたときに GeoMap パネルがどのようにレンダリングされるかを設定します。

- [ビュー] は、パネルが最初にロードされたときにマップの中心を設定します。
  - [Fit to data] は、データが変更されたときのマップレイヤーと更新のデータ範囲に基づいてマップビューに適合します。
    - [データ] オプションでは、[すべてのレイヤー] のデータ、単一の [レイヤー]、または選択したレイヤーの [最後の値] に基づいて範囲を選択できます。
    - [レイヤー] において、単一の [Layer] (レイヤー) またはレイヤーの [最後の値] からデータをフィッティングする場合に選択できます。
    - [パディング] は、データ範囲を超えた相対的な割合でパディングを設定します ([最後の値] のみを参照する場合は使用できません)。
    - [Max Zoom] は、データをフィッティングするときに最大ズームレベルを設定します。
  - [座標] は、以下に基づいてマップビューを設定します。
    - Latitude (緯度)
    - Longitude (経度)
  - デフォルトビューには、以下を含むものもあります。
    - (0°, 0°)

  - 北米
  - 南米
  - 欧州
  - アフリカ
  - 西アジア
  - 南アジア
  - 東南アジア
  - 東アジア
  - オーストラリア
  - オセアニア
- [Zoom] は、初期ズームレベルを設定します。

レイヤーのマッピング

ジオマップでは、複数のレイヤーの表示がサポートされています。各レイヤーは、ベースマップ上に地理空間データをどのように表示するかを決定します。

タイプ

ジオマップの視覚的表現では、3 つのマップレイヤータイプを選択できます。

- マーカーレイヤー は、各データポイントにマーカーをレンダリングします。
- ヒートマップレイヤー はデータのヒートマップを視覚化します。
- GeoJSON レイヤー は GeoJSON ファイルから静的データをレンダリングします。
- 夜間/昼間レイヤー (アルファ) は夜間または昼間のリージョンをレンダリングします。
- ルートレイヤー (プレビュー) データポイントをルートとしてレンダリングします。
- 写真レイヤー (プレビュー) は、各データポイントに写真をレンダリングします。
- ネットワークレイヤー (プレビュー) は、データからネットワークグラフを視覚化します。

また、実験 (またはアルファ) レイヤータイプも 2 つあります。

- [Icon at last point (alpha)] は、最後のデータポイントにアイコンをレンダリングします。
- [Dynamic GeoJSON (alpha)] は、クエリ結果に基づいて GeoJSON ファイルをスタイルします。

> **Note**
>
> パブリックプレビューでプレビューまたはアルファとマークされたレイヤーです。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

[レイヤーコントロール]

レイヤーコントロールを使用すると、レイヤーの作成、名前の変更、レイヤーの順序の変更、削除を行うことができます。

- [レイヤーの追加] で、ジオマップ用に設定可能な追加のデータレイヤーを作成できます。レイヤーを追加すると、レイヤータイプを選択するように求められます。レイヤータイプは、パネル設定中にいつでも変更できます。
- レイヤーコントロールを使用すると、パネルのレイヤーの名前変更、削除、順序変更をすることができます。
  - [レイヤー名の編集] (鉛筆アイコン) でレイヤーの名前を変更できます。
  - [Trash Bin] (ごみ箱) でレイヤーを削除できます。
  - [Reorder] (順序変更) (6 つのドット/グラブハンドル) では、レイヤーの順序を変更できます。上位レイヤーのデータは、下位レイヤーのデータの上に表示されます。視覚化では、レイヤーの順序の選択を簡素化するために、ドラッグアンドドロップするとレイヤーの順序が更新されます。

1 つのジオマップパネルに複数のデータレイヤーを追加して、リッチで詳細な視覚化を作成できます。

場所

ジオマップには、地理的データの情報源が必要です。このデータには 4 つのマッピングオプションがあり、データベースクエリを使用して取得します。

- [自動] で口ケーションデータを自動的に検索します。クエリがデータフィールドの次のいずれかの名前に基づいている場合は、このオプションを使用します。
  - geohash: `geohash`
  - latitude: `latitude`, `lat`
  - longitude: `longitude`, `lng`, `lon`
  - [ルックアップ]: `lookup`

- Coords (座標): クエリに座標データが含めることを指定します。データベースクエリから、緯度と経度の数値データフィールドを選択するよう求められます。
- Geohash (ジオハッシュ): クエリがジオハッシュデータが含めることを指定します。データベースクエリから、ジオハッシュの文字列データフィールドを選択するよう求められます。
- [ルックアップ]: クエリに値にマッピングする必要がある口ケーション名データが含めることを指定します。データベースクエリと地名索引からルックアップフィールドを選択するよう求められます。地名索引は、クエリされたデータを地理的ポイントにマッピングするために使用されるディレクトリです。

マーカーレイヤー

マーカーレイヤーを使用すると、データポイントを円、正方形、三角形、星などのさまざまな形のマーカーとして表示することができます。

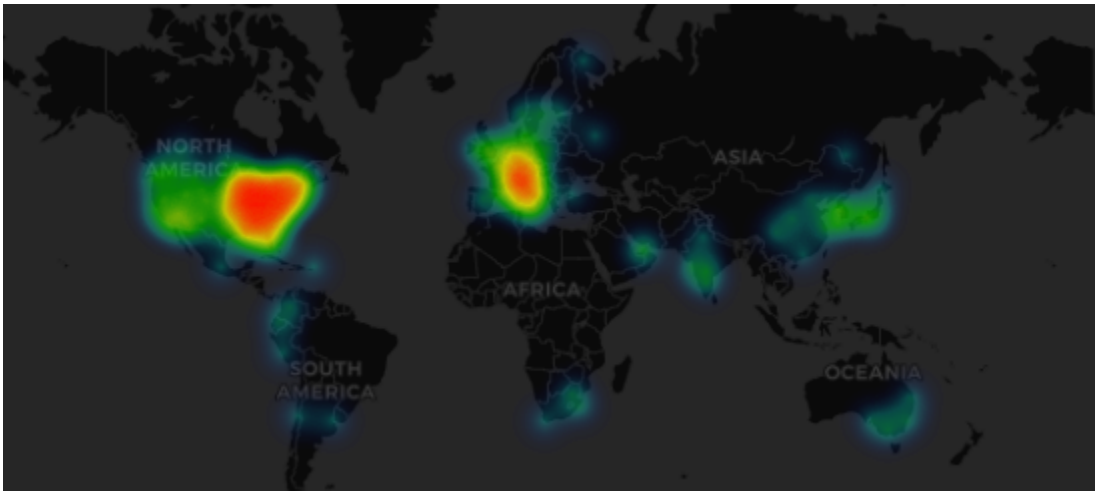
マーカーには多くのカスタマイズオプションがあります。

- [サイズ] ではマーカーのサイズを設定します。デフォルトは `Fixed size` です。これにより、すべてのマーカーサイズはデータに関係なく同じになります。ただし、選択したフィールドに対応するデータに基づいてマーカーのサイズを変更するオプションもあります。マーカーレイヤーがこの範囲内でスケールできるように、`Min` と `Max` のマーカーサイズを設定する必要があります。
- [Symbol] を使用すると、データに追加のビジュアルコンテキストを提供するのに役立つシンボル、アイコン、またはグラフィックを選択できます。シンプルな記号や Unicon ライブラリなど、Grafana に含まれているアセットから選択します。画像アセットを含む URL を指定することもできます。画像は SVG (Scalable Vector Graphics: スケーラブルベクトルグラフィック) である必要があります。
- [Symbol Vertical Align] では、データポイントに対するシンボルの垂直配置を設定します。なお、シンボルの回転角度は、まずデータポイントの周囲に適用され、次に垂直方向のアライメントがシンボルの回転に対して適用されます。
- [Symbol Horizontal Align] では、データポイントに対するシンボルの水平配置を設定します。シンボルの回転角度は、まずデータポイントの周囲に適用され、次に水平方向のアライメントがシンボルの回転に対して適用されます。
- Color ではマーカーの色を設定します。デフォルト `Fixed color` では、すべてのマーカーを特定の色に設定します。標準オプションセクションで選択したフィールドデータポイント値とカラースキーム設定に応じて、条件付き色を使用するオプションもあります。
- [塗りつぶしの透明度]: 各マーカーの透明度を設定します。

- [Rotation angle] では、各マーカーの回転角度を設定します。デフォルトは [Fixed value] (固定値) です。これにより、データに関係なくすべてのマーカーが同じ角度で回転しますが、選択したフィールドに対応するデータに基づいてマーカーの回転を設定するオプションもあります。
- [テキストラベル] では、マーカーごとにテキストラベルを設定します。
- [凡例の表示] を使用すると、レイヤーの凡例を切り替えることができます。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

ヒートマップレイヤー

ヒートマップレイヤーは、さまざまなデータポイントをクラスター化して、密度の異なる場所を視覚化します。



ヒートマップレイヤーを追加するには

[Data レイヤー] のドロップダウンメニューを選択し、`Heatmap` を選択します。

`Markers` と同様に、視覚化するデータポイントとその視覚化方法を決定するオプションが表示されます。

- [重み値] では、ヒートマップクラスターの強度を設定します。[固定値] を選択すると、すべてのデータポイントに対して一定の重みが適用されます。この値は 0~1 の範囲で設定します。マーカーと同様に、ドロップダウンには、データ値に応じて重み値を自動的にスケールする代替オプションがあります。
- [Radius] (半径): ヒートマップクラスターのサイズを設定します。
- [Blur] (ぼかし) では、各クラスターのぼかしの量を設定します。

- [不透明度] では、各クラスターの不透明度を設定します。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

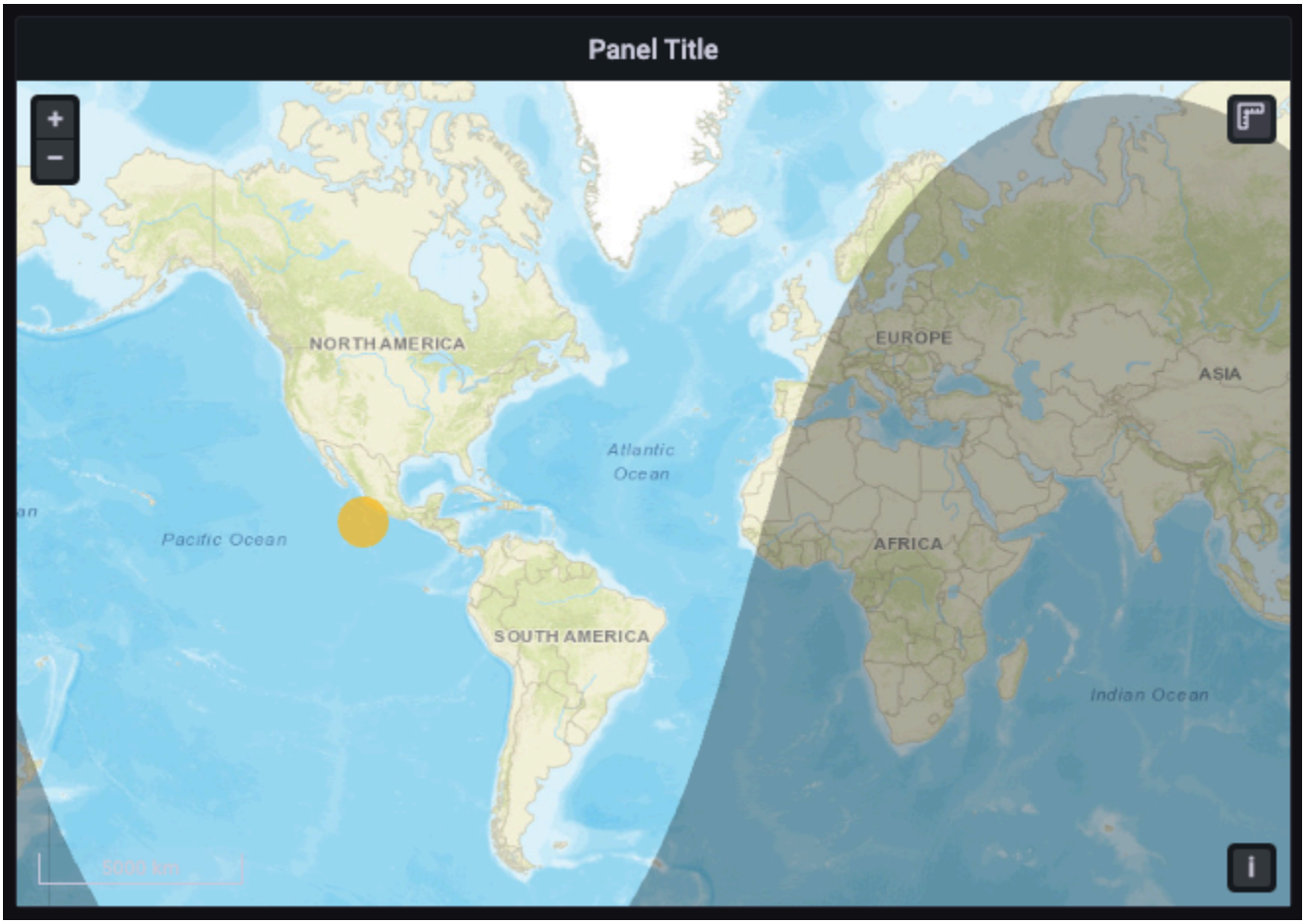
GeoJSON レイヤー

GeoJSON レイヤーを使用すると、ファイルシステムから静的 GeoJSON ファイルを選択してロードできます。

- [GeoJSON URL] では、Grafana に付属する GeoJSON ファイルを選択できます。
- [デフォルトのスタイル] では、上記のルールが一致しない場合に適用するスタイルを制御します。
  - [色] ではデフォルトのスタイルの色を設定します
  - [不透明度] ではデフォルトの不透明度を設定します
- [スタイルルール] では、機能のプロパティに基づいてスタイルを適用します
  - [ルール] を使用すると、GeoJSON ファイルから特徴量、条件、および[値]を選択して、ルールを定義できます。ごみ箱アイコンを使用して、現在のルールを削除できます。
  - [色] では、現在のルールに対するスタイルの色を設定します
  - [不透明度] では、現在のルールの透明性レベルを設定します。
- [スタイルルールの追加] で、追加のスタイルルールを作成できます。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

夜間/昼間レイヤー (アルファ)

夜間/昼間レイヤーには、現在の時間範囲に基づいて夜間および昼間の領域が表示されます。



オプション

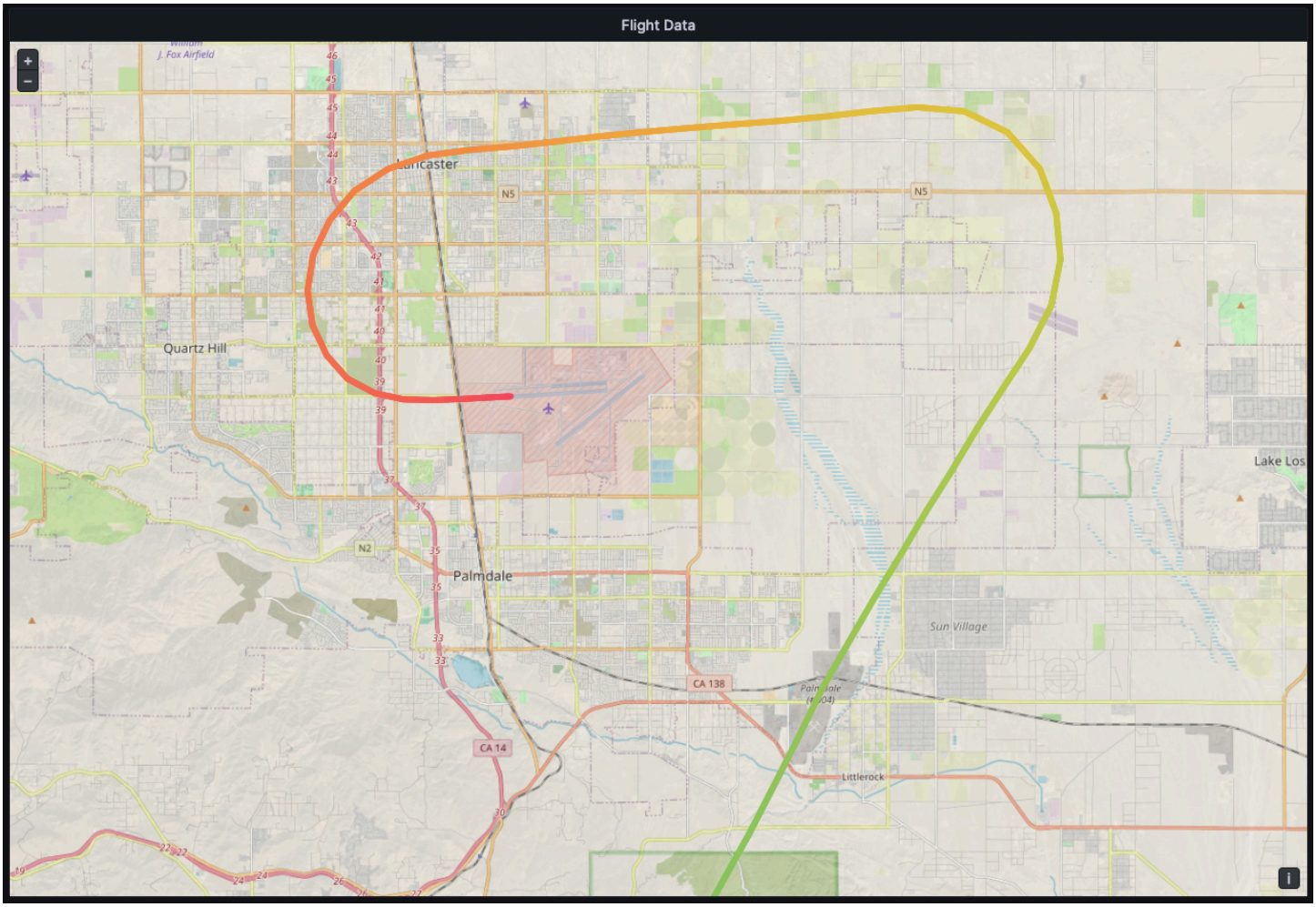
- [表示] では、パネルの時間範囲から時間ソースを切り替えます。
- Night region color では、夜間領域の色を選択します。
- [Display sun] (太陽の表示) で太陽アイコンが切り替えられます。
- [不透明度] は 0 (透明) から 1 (不透明) です。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

> **Note**
>
> 詳細については、「OpenLayers の拡張機能 - DayNight」を参照してください。

ルートレイヤー (プレビュー)

ルートレイヤーは、データポイントをルートとしてレンダリングします。

> **Note**
>
> ルートレイヤーは現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。



オプション

- [サイズ] ではルートの厚さを設定します。デフォルトでは値が固定されています。フィールドデータを選択すると、フィールドデータがスケールすることができる [最小] 範囲と [最大] 範囲を設定できます。

- [Color] ではルートの色を設定します。デフォルトでは固定色に設定されています。色をフィールドデータに関連付けることもできます。
- [塗りつぶしの不透明度] は、ルートの不透明度を設定します。
- [テキストラベル] では、ルートごとにテキストラベルを設定します。
- [Arrow] (矢印) では、データ順にルートに沿って表示する矢印スタイルを設定します。
  - [なし]
  - [前進]
  - [Reverse] (逆順)
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

> **Note**
>
> 詳細については、「OpenLayers の拡張機能 - 動線スタイル」を参照してください。

写真レイヤー (プレビュー)

写真レイヤーは、各データポイントに写真をレンダリングします。

> **Note**
>
> 写真レイヤーは現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

オプション

- [Image Source Field] (画像ソースフィールド) では、画像データを Base64 でエンコードされた画像バイナリ (`...`) として含む文字列フィールドを選択できます。
- [Kind] (種類) では、画像の周りにフレームスタイルを設定します。以下から選択します。
  - [Square] (平方)
  - [Circle] (円形)
  - [Anchored] (アンカー)
  - [Folio] (フォリオ)

- [Crop] (トリミング) では、画像がフィットするようにトリミングされるかどうか切り替えます。
- [Shadow] (シャドウ) では、画像の背後にあるボックスシャドウを切り替えます。
- [Border] (境界線) では、画像の周囲の境界サイズを設定します。
- [Border color (境界線の色)] では、画像の周囲の境界線の色を設定します。
- [Radius] (範囲) では、画像の全体的なサイズをピクセル単位で設定します。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

> **Note**
>
> 詳細については、「OpenLayers の拡張機能 - 画像写真スタイル」を参照してください。

ネットワークレイヤー (プレビュー)

ネットワークレイヤーはネットワークグラフをレンダリングします。このレイヤーは、ノードグラフの視覚化でサポートされているのと同じデータ形式をサポートし、ノードデータには地理空間データが追加されています。地理空間データは、マップ上のノードを見つけてレンダリングするために使用されます。

> **Note**
>
> ネットワークレイヤーは現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。



> **Note**
>
> ネットワークレイヤーは現在パブリックプレビュー中です。Grafana Labs ではサポートが限られており、この機能が一般公開される前に重要な変更が発生する可能性があります。

オプション

- [Arrow] (矢印) では、各エッジに表示する矢印の方向を設定します。これは、ソースからターゲットへの方向を前進とします。以下から選択します。
  - [なし]
  - [前進]
  - [Reverse] (逆順)
  - [Both] (両方)
- [凡例の表示] を使用すると、レイヤーの凡例を切り替えることができます。凡例はノードデータのみをサポートします。
- [Display tooltip] (ツールチップの表示) を使用すると、レイヤーのツールチップを切り替えることができます。

ノードスタイル

- [サイズ] ではノードのサイズを設定します。デフォルトは [Fixed size] (固定サイズ) です。これにより、データに関係なくすべてのノードサイズが同じになりますが、選択したフィールドに対応するデータに基づいてノードのサイズを変更するオプションもあります。[最小] ノードサイズと [最大] ノードサイズは、ノードがこの範囲内でスケールできるように設定する必要があります。
- Color ではノードの色を設定します。デフォルトは [Fixed color] (固定色) で、すべてのノードを特定の色に設定します。標準オプションセクションで選択したフィールドデータポイント値とカラースキーム設定に応じて、条件付き色を使用するオプションもあります。
- [Symbol] を使用すると、データに追加のビジュアルコンテキストを提供するのに役立つシンボル、アイコン、またはグラフィックを選択できます。シンプルな記号や Unicon ライブラリなど、Grafana に含まれているアセットから選択します。画像アセットを含む URL を指定することもできます。画像は SVG (Scalable Vector Graphics: スケーラブルベクトルグラフィック) である必要があります。
- [塗りつぶしの不透明度] は、各ノードの透明度を設定します。
- [Rotation angle] (回転角度) では、各ノードの回転角度を設定します。デフォルトは [Fixed value] (固定値) です。これにより、すべてのノードはデータに関係なく同じ角度で回転しますが、選択したフィールドに対応するデータに基づいてノードの回転を設定するオプションもあります。
- [テキストラベル] では、ノードごとにテキストラベルを設定します。

エッジスタイル

- [サイズ] はエッジの線幅を設定します。デフォルトは [Fixed size] (固定サイズ) です。これにより、データに関係なくすべてのエッジの線幅が同じになりますが、選択したフィールドに対応するデータに基づいてエッジのサイズを変更するオプションもあります。[最小] ノードサイズと [最大] ノードサイズは、ノードがこの範囲内でスケールできるように設定する必要があります。
- Color ではエッジの色を設定します。デフォルトは [Fixed color] (固定色) で、すべてのエッジを特定の色に設定します。標準オプションセクションで選択したフィールドデータポイント値とカラースキーム設定に応じて、条件付き色を使用するオプションもあります。
- [塗りつぶしの不透明度] では、各エッジの透明度を設定します。
- [テキストラベル] では、エッジごとにテキストラベルを設定します。

CARTO レイヤー

CARTO レイヤーは Amazon Managed Grafana ではサポートされていません。

XYZ タイルレイヤー

XYZ タイルレイヤーは Amazon Managed Grafana ではサポートされていません。

Open Street Map レイヤー

Open Street Map レイヤー (デフォルトのベースマップを除く) は、Amazon Managed Grafana ではサポートされていません。

ArcGIS レイヤー

ArcGIS レイヤーは Amazon Managed Grafana ではサポートされていません。

マップコントロール

マップコントロールセクションには、マップ情報とツールオーバーレイに対するさまざまなオプションが含まれています。

Zoom

ズームコントロールのオプションです。

Show zoom control

ズームコントロールを左上隅に表示します。

Mouse wheel zoom

マウスホイールを使用してズームインまたはズームアウトできる設定を、オンまたはオフにします。

Show attribution

マップ上のベースマップレイヤーの属性を表示します。

Show scale

スケール情報を左下隅に表示します。

> **Note**
>
> [m]/[km] 単位で表示します。

Show measure tools

測定ツールを右上隅に表示します。測定値は、このコントロールが開いている場合にのみ表示されます。

- クリックして測定を開始します
- 連続クリックで測定を続行します
- ダブルクリックして測定を終了します

> **Note**
>
> 測定タイプまたは単位を変更すると、以前の測定がマップから削除されます。
> コントロールを閉じてから再度開くと、最新の測定値が表示されます。
> 測定値は、クリックしてドラッグすることで変更できます。

Length

ジオメトリの球面長を取得します。この長さは、座標間の大圏距離の合計です。マルチパートジオメトリの場合、長さは各パートの長さの合計です。ジオメトリは「EPSG:3857」にあると仮定されます。

長さ測定には、次の単位を選択できます。

- メートル (m/km)
- フィート (ft)
- マイル (mi)
- 海里 (nmi)

[面積]

ジオメトリの球の表面積を取得します。この面積は、ポリゴンエッジが球上の大円のセグメントであると仮定して計算されます。ジオメトリは「EPSG:3857」にあると仮定されます。

面積測定には、次の単位を選択できます。

- 平方メートル (m²)

- 平方キロメートル (km²)
- 平方フィート (ft²)
- 平方マイル (mi²)
- エーカー (acre)
- ヘクタール (ha)

Show debug

デバッグ情報をマップの右上隅に表示します。これは、データソースのデバッグまたは検証に役立ちます。

- [Zoom] は、マップの現在のズームレベルを表示します。
- [Center] (中央) には、マップ中央の現在の経度と緯度が表示されます。

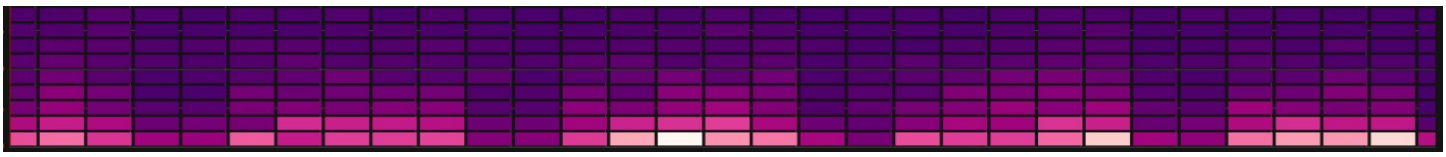
Tooltip (ツールチップ)

- [なし] では、データポイントがクリックされた場合にのみツールチップが表示されます。
- [Details] (詳細) では、データポイントにポインタを合わせると、ツールチップが表示されます。

ヒートマップ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ヒートマップパネルの視覚化により、ヒストグラムを経時的に表示できます。ヒストグラムの詳細については、「ヒストグラムおよびヒートマップの概要」を参照してください。



データから計算する

この設定は、データが既に計算されたヒートマップ (データソース/トランスフォーマーから) であるか、パネルで計算する必要があるヒートマップであるかを決定します。

X バケット

この設定は、X 軸をバケットに分割する方法を決定します。サイズ入力で時間間隔を指定できます。例えば、1h の時間範囲により、セルは X 軸で 1 時間幅になります。

Y バケット

この設定は、Y 軸をバケットに分割する方法を決定します。

Y バケット目盛

次のいずれかの Y 軸値目盛を選択します。

- [線形] – 線形目盛。
- [log (base 2)] – 底が 2 の対数目盛。
- [log (base 10)] – 底が 10 の対数目盛。
- [symlog] – 正負対象の対数目盛。

Y 軸

Y 軸の表示方法を定義します。

[配置]

- [左] – 左側
- [Right] – 右側
- [非表示] – 非表示

[単位]

単位設定

[小数]

この設定により、小数点以下の設定が決まります。

[最小/最大値]

この設定では、軸範囲を設定します。

[Reverse] (逆順)

選択すると、軸は逆の順序で表示されます。

[Display multiple y-axes] (複数の Y 軸を表示)

場合によっては、複数の Y 軸を表示したい場合があります。例えば、経時的な温度と湿度の両方を示すデータセットがある場合、これらの 2 つの系列で異なる単位を持つ 2 つの y 軸を表示したいとします。

これを行うには、フィールドオーバーライドを追加します。必要な回数だけステップに従うことで、必要な数の y 軸を追加できます。

カラー

色のスペクトルは、値の数 (各バケット内) と各バケットに割り当てられた色間とのマッピングを制御します。スペクトルの左端の色は最低数を表し、右端の色は最大数を表します。一部のカラースキームは、ライトテーマを使用するときに自動的に反転されます。

また、カラーモードを [不透明度] に変更することもできます。この場合、色は変わりませんが、不透明度はバケット数に応じて変化します。

- モード
  - [Scheme] (スキーム) – セルの色で表されるバケット値。
    - [Scheme] – モードがスキームの場合、カラースキームを選択します。
  - [不透明度] – セルの不透明度で表されるバケット値。[Opaque] (不透明) セルは最大値を意味します。
    - [Color] – セルのベース色。
    - [Scale] (目盛) – バケット値を不透明度にマッピングするための目盛。
      - [線形] – 線形目盛。バケット値は不透明度に対して直線的にマッピングされます。
      - [sqrt] (平方根) – 対数目盛。セルの不透明度は `value ^ k` として計算され、k は設定された指数値です。指数が 1 より小さい場合、対数スケールが表示されます。指数が 1 より大きい場合、指数スケールが表示されます。1 の場合、目盛は線形と同じになります。
    - [Exponent] – 指数の値 (`0` より大きい場合)。

[値を元にした開始/終了の色]

デフォルトでは、Grafana は最小および最大バケット値に基づいてセルの色を計算します。[最小値] と [最大値] では、これらの値を上書きできます。バケット値は Z 軸、最小値と最大値はそれぞれ Z の最小値と Z の最大値と考えてください。

- [開始値] – セルの色の計算に使用される最小値。バケット値が最小値より小さい場合、「最小値」の色にマッピングされます。系列の最小値はデフォルト値です。
- [終了値] – セルの色の計算に使用される最大値。バケット値が最大値より大きい場合、「最大値」の色にマッピングされます。系列の最大値はデフォルト値です。

セル表示

この設定で、視覚化する内容を絞り込みます。

追加の表示オプション

Tooltip (ツールチップ)

- [Show tooltip] - ヒートマップのツールチップを表示します。
- [Show Histogram] – ツールチップに Y 軸のヒストグラムを表示します。ヒストグラムは、特定のタイムスタンプにおけるバケット値の分布を表します。
- Show color scale – ツールチップにカラースケールを表示します。カラースケールは、バケットの値と色との間のマッピングを表します。

[凡例]

視覚化にヒートマップの凡例を表示するかどうかを選択します。

Exemplars (見本)

見本データの表示に使用する色を設定します。

ヒストグラム

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ヒストグラムの視覚化は値の分布を計算して棒グラフとして表示します。Y 軸と各バーの高さは各括弧に含まれる値の数を表し、X 軸は値の範囲を表します。



サポートされる形式

ヒストグラムの視覚化では、1 つ以上の数値フィールドを持つ時系列とテーブル結果をサポートしています。

表示オプション

これらのオプションで、視覚化する内容を絞り込みます。

[バケットサイズ]

バケットのサイズです。自動バケットサイズ設定 (全範囲の 10% まで) では、この値は空のままにします。

[バケットオフセット]

最初のバケットを 0 で始めない場合です。ゼロ以外のオフセットでは集計ウィンドウがシフトします。例えば、デフォルトの 0 オフセットの 0～5、5～10、10～15 の 5 サイズのバケットは、オフ

セット 2 の 2～7、7～12、12～17 になる場合があります。この場合、オフセット 0、5、10 であれば事実上何も実行されません。通常、このオプションは自動ではなく、明示的に定義されたバケットサイズで使用します。この設定が作用するためには、オフセット量を 0 より大きく、バケットサイズより小さくする必要があります。この範囲外の値には、この範囲内の値と同じ効果があります。

[系列の結合]

これにより、すべての系列とフィールドが結合されたヒストグラムにマージされます。

[線の幅]

棒の線幅を制御します。

[塗りつぶしの不透明度]

棒の塗りつぶしの不透明度を制御します。

[グラデーションモード]

グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーション塗りつぶしなし。これはデフォルト設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。
- [色相] – グラデーション色は線の色相に基づいて生成されます。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。凡例の詳細については、「凡例の設定」を参照してください。

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。

- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。詳細については、「計算タイプ」を参照してください。

[凡例計算]

凡例に表示する計算を選択します。複数選べます。

ログ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ログパネルの視覚化では、Elastic、Influx、Loki などのログをサポートするデータソースのログ行が表示されます。通常、グラフパネルの横のこのパネルを使用して、関連するプロセスのログ出力を表示します。

ログパネルには、[Query] タブに入力されたクエリの結果が表示されます。複数のクエリ結果がマージされ、時間別にソートされます。表示可能な行よりも多くの行がデータソースから返される場合は、パネル内でスクロールできます。

レンダリングされる行数を制限するには、[クエリオプション]で [最大データポイント数] の設定を使用できます。設定しない場合、データソースは通常デフォルトの制限を適用します。

ログレベル

レベルラベルが指定されているログの場合、ラベルの値を使用してログレベルを決定し、それに応じて色を更新します。ログに指定されたレベルラベルがない場合、そのコンテンツが、サポートされている表現のいずれかと一致するかどうかを調べます (詳細については、下記を参照してください)。ログレベルは常に最初の一致によって決まります。Grafana がログレベルを判別できない場合、不明なログレベルとして視覚化されます。詳細については、「ログレベル」を参照してください。

ログの詳細

各ログ行には、より堅牢なインタラクションのために、ラベルと検出されたフィールドを含む拡張可能な領域があります。各フィールドまたはラベルには、表示されているすべてのログに関連する統計を表示する統計アイコンがあります。

データリンク

データリンクを使用すると、ログメッセージの任意の部分を内部リンクまたは外部リンクに変換できます。作成されたリンクは、[ログの詳細] ビュー内の [リンク] セクションにボタンとして表示されます。

表示オプション

以下の設定を使用して、視覚化する内容を絞り込みます。

- [時間] – 時間列を表示または非表示にします。これはデータソースから報告されたログ行と関連するタイムスタンプです。
- [一意のラベル] – 一意のラベル列を表示または非表示にします。この列には、共通ではないラベルのみが表示されます。
- [共通ラベル] - を表示または非表示にします
- [行の折り返し] – 行の折り返しを切り替えます。
- [Prettify JSON] – すべての JSON ログを pretty print (整形表示) するには、これを `true` に設定します。この設定は、JSON 以外の形式のログには影響しません。
- [ログ詳細の有効化] – オプションを切り替えて、各ログ行のログ詳細ビューを表示します。デフォルトの設定は`true` です。
- [Order] (順序) – 結果を降順または昇順で表示します。デフォルトは [降順] で、最新のログが最初に表示されます。[昇順] に設定すると最も古いログ行が最初に表示されます。

ニュース

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ニュースの視覚化には RSS フィードが表示されます。デフォルトでは、Grafana Labs のブログ記事が表示され、ユーザーは別の RSS フィード URL を入力してこれを変更できます。

[表示] セクションに RSS の URL を入力します。この視覚化タイプは他のクエリを受け入れず、ユーザーはこの視覚化を使用して RSS フィードデータを改ざんまたは検索できるとは想定しないでください。

Note

RSS フィードは、プロキシなしで Grafana フロントエンドによって読み込まれます。その結果、適切な CORS ヘッダーで設定された RSS フィードのみが読み込まれます。表示しようとしている RSS フィードが読み込みに失敗する場合は、RSS フィードを再ホストするか、独自のプロキシ作成を検討してください。

ノードグラフ

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ノードグラフは、有向グラフまたはネットワークを視覚化できます。有向フォースレイアウトを使用してノードを効果的に配置し、複雑なインフラストラクチャマップ、階層、または実行図の表示に役立ちます。

データ要件

ノードグラフでは、ノードとエッジを表示するためにデータの特定の形状が必要です。このグラフでは、すべてのデータソースまたはクエリを視覚化できるわけではありません。これをデータソースデベロッパーとして使用する場合は、データ API に関するセクションを参照してください。

ノードグラフは、ノードとエッジで構成されます。

- あるノードは円として表示されます。あるノードは、アプリケーション、サービス、またはアプリケーションの観点から関連するその他のものを表す場合があります。
- あるエッジは、2 つのノードを接続する線として表示されます。接続は、リクエスト、オペレーション、または 2 つのノード間の他の関係である場合があります。

ノードとエッジの両方にメタデータまたは統計を関連付けることができます。データソースは、表示される情報と値を定義するため、異なるデータソースが異なるタイプの値を表示したり、一部の値を表示しなかったりすることができます。

ノード

通常、ノードには、ノード内の 2 つの統計値とノードのすぐ下の 2 つの識別子、通常は名前とタイプが表示されます。ノードでは、ノードの周囲に一連の別の値を色付きの円として表示することもできます。異なる色のセクションは、合計が 1 になる異なる値を表します。例えば、円の赤い部分によって エラーの割合を表すことができます。

ノードを選択すると表示されるコンテキストメニューには、さらに詳細を表示できます。Grafana ワークスペースの他の部分または外部リンクをターゲットにできる追加のリンクを、コンテキストメニューに含めることもできます。

> **Note**
>
> ノードグラフには 1,500 個のノードしか表示できません。この制限を超えると、右上隅に警告が表示され、一部のノードは非表示になります。グラフ内の [非表示ノード] マーカーをクリックすると、グラフの非表示部分を展開できます。

エッジ

エッジでは、エッジにカーソルを合わせると統計を表示することもできます。ノードと同様に、エッジを選択すると、追加の詳細とリンクを含むコンテキストメニューを開けます。

この視覚化をサポートする最初のデータソースは、サービスマップ機能の AWS X-Ray データソースです。詳細については、「AWS X-Ray データソースに接続する」を参照してください。

ノードグラフの操作方法

Pan

ノードグラフ内でパンするには、ノードまたはエッジの外部を選択してポインタをドラッグします。

拡大または縮小

ノードグラフの左上隅にあるボタンを使用してズームするか、マウスホイールまたは `Ctrl` (または `Cmd`) キーで他のスクロール入力を使用できます。

非表示ノードを調べる

特定の時間に表示されるノードの数は、合理的なパフォーマンスを維持するために制限されます。この制限外のノードは、そのエッジに接続されている非表示ノードのおよその数を示す選択可能なマーカーの背後に隠されます。マーカーを選択して、そのノードの周りのグラフを展開できます。

グリッドビュー

グリッドビューに切り替えて、グラフ内の最も興味深いノードの概要を把握できます。グリッドビューは、エッジのないグリッド内のノードを表示し、ノード内に表示される統計、またはノードの色付き境界で表される統計でソートできます。

ノードをソートするには、凡例内の統計を選択します。統計名 (# または #) の横にあるマーカーは、ソートおよびソート方向に現在使用されている統計を示します。

ノードを選択し、[Show in Graph layout (グラフレイアウトで表示)] オプションを選択すると、選択したノードに焦点を当ててグラフレイアウトに切り替え、グラフ全体のコンテキストで表示します。

Data API

この視覚化では、正しく表示するために、データソースから返されるデータの特定の形状が必要です。

ノードグラフには、少なくともグラフのエッジを記述するデータフレームが必要です。デフォルトでは、ノードグラフはこのデータフレームに基づいてノードと統計を計算します。必要に応じて、

ノード固有のメタデータをより多く表示する必要がある場合に備えて、ノードを記述する 2 番目のデータフレームを送信できます。ノードグラフをレンダリングするには、両方のデータフレームに `frame.meta.preferredVisualisationType = 'nodeGraph'` を設定するか、`nodes` と `edges` にそれぞれ名前を付ける必要があります。

構造からのエッジデータ

必須フィールド

フィールド名	タイプ	説明
id	文字列	エッジの一意の識別子。
ソース	文字列	ソースノードの ID。
target	文字列	ターゲットの ID。

オプションのフィールド

フィールド名	タイプ	説明
Mainstat	文字列/数値	エッジにカーソルを合わせたときのオーバーレイに表示される最初の統計。値はそのまま示す文字列でも、数値でもかまいません。数値の場合、そのフィールドに関連付けられている単位も表示されます。
secondarystat	文字列/数値	mainStat と同じですが、そのすぐ下に表示されます。
detail__*	文字列/数値	プレフィックスが `detail__` の任意のフィールドは、エッジをクリックするとコンテキストメニューのヘッダーに表示されます。より判読

フィールド名	タイプ	説明
		可能なラベルに config.displayName を使用します。

構造からのノードデータ

必須フィールド

フィールド名	タイプ	説明
id	文字列	ノードの一意の識別子。この ID は、ソースフィールドとターゲットフィールドのエッジによって参照されます。

オプションのフィールド

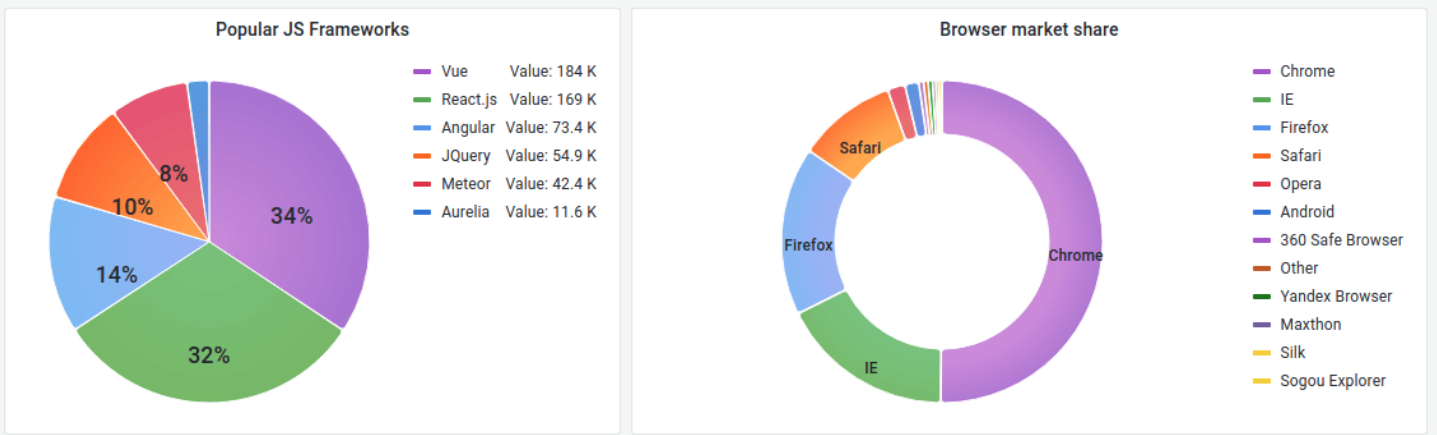
フィールド名	タイプ	説明
title	文字列	ノードのすぐ下に表示されるノードの名前。
subtitle	文字列	タイトルの下に表示される追加の名前、タイプ、またはその他の識別子。
mainstat	文字列/数値	ノード自体の中に表示される最初の統計。これは、値をそのまま表示する文字列でも数値でもかまいません。数値の場合、そのフィールドに関連付けられている単位も表示されます。

フィールド名	タイプ	説明
secondarystat	文字列/数値	mainStat と同じですが、ノード内でその下に表示されます。
arc__*	数値	プレフィックスが `arc__` の任意のフィールドは、ノードの周囲に色円を作成するために使用されます。これらのフィールドのすべての値は、合計で 1 になる必要があります。`config.color.fixedColor` を使用して色を指定できます。
detail__*	文字列/数値	ノードをクリックすると、プレフィックスが `detail__` の任意のフィールドがコンテキストメニューのヘッダーに表示されます。より人間が判読可能なラベルに `config.displayName` を使用します。

フィールド名	タイプ	説明
color	文字列/数値	`arc__` フィールドを使用して色セクションを指定する代わりに、単一の色を指定するために使用できます。文字列 (許容可能な HTML カラー文字列である必要があります) でも、数値でもかまいません。この場合、動作は `field.config.color.mode` 設定によって異なります。例えば、これはフィールド値によって制御されるグラデーション色の作成に使用できます。
icon	文字列	デフォルトの統計ではなく、ノード内に表示されるアイコンの名前。Grafana 組み込みアイコンのみが許可されます (使用可能なアイコンについては、こちらを参照してください)。
nodeRadius	数値	ピクセル単位の半径値。ノードサイズを管理するために使用されます。
highlighted	ブール値	ノードを強調表示するかどうかを設定します。例えば、複数のノードとエッジを強調表示してグラフ内の特定のパスを表すために使用します。デフォルトは `false` です。

円グラフ

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。



円グラフには、1 つ以上のクエリから誘導された系列、または系列内の値が、円グラフのスライスで相互に関連して表示されます。スライスの円弧の長さ、面積、中心角度は、すべての値の合計に関連するため、すべてスライス値に比例します。この種類のグラフは、一連の小規模な値を見た目がきれいな形ですばやく比較する場合に最適です。

値オプション

次のオプションを使用して、視覚化に使う値を絞り込みます。

[表示]

表示する情報量を選択します。

- [計算] – 各値を系列ごとに単一の値に減らします。
- [すべての値] – 1 つの系列におけるすべての値を表示します。

計算

[計算]を選択すると、各系列を削減する計算を選択できます。使用可能な計算の詳細については、「計算タイプ」を参照してください。

[制限]

1 つの系列におけるすべての値を表示する場合、表示される値の数には制限があります。

[フィールド]

視覚化に表示するフィールドを少なくとも 1 つ選択します。各フィールド名はリストで確認できます。または、次のいずれかのオプションを選択できます。

- [数値フィールド] – 数値を持つすべてのフィールド。
- [すべてのフィールド] – 変換によって削除されないすべてのフィールド。
- [時間] – 時間値を持つすべてのフィールド。

円グラフオプション

これらのオプションで、視覚化の見え方を調整します。

円グラフタイプ

円グラフの表示スタイルを選択します。次のいずれかになります。

- [Pie] (円) – 標準円グラフ
- [Donut] (ドーナツ) – 中央に穴がある円グラフ

ラベル

円グラフに表示するラベルを選択します。複数選べます。

- [名前] – 系列名またはフィールド名。
- [パーセント] – 全体の割合。
- [Value] – 未加工の数値。

デフォルトでは、ラベルはグラフ部分の上に白で表示されます。グラフの色を濃く選択して、より見やすくすることができます。長い名前や数字は文字切れする可能性があります。

[ツールチップモード]

視覚化にカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、グラフ上でカーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

凡例オプション

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。凡例の詳細については、「凡例の設定」を参照してください。

凡例の可視性

[Visibility] (可視性) トグルを使用して、凡例を表示または非表示にします。

[凡例モード]

凡例の表示モードを設定します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。

[凡例の配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する値を選択します。複数選べます。

- [パーセント] – 全体の割合。

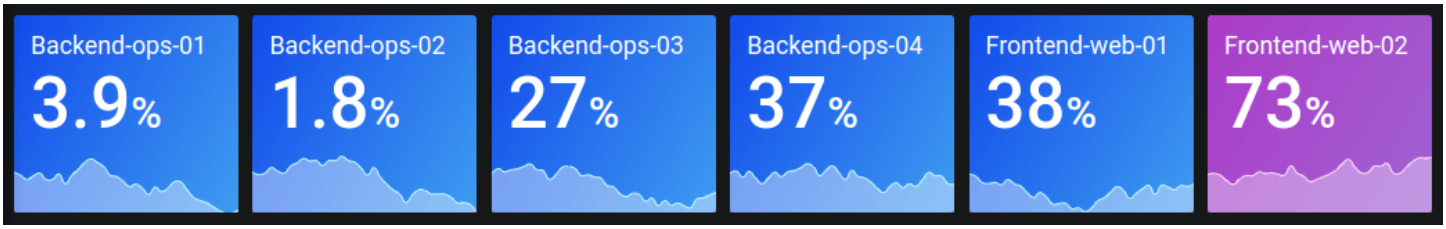
- [Value] – 未加工の数値。

凡例の詳細については、「凡例の設定」を参照してください。

統計

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

統計は、オプションのグラフスパークラインを含む 1 つの大きな統計値を表示します。しきい値またはオーバーライドを使用して、背景または値の色を制御できます。



デフォルトでは、統計パネルには次のいずれかが表示されます。

- 単一の系列またはフィールドの値のみ。
- 複数の系列またはフィールドの値と名前の両方。

テキストモードを使用して、テキストが表示されるかどうかを制御できます。

自動レイアウト調整

パネルは、ダッシュボードで使用可能な幅と高さに応じてレイアウトを自動的に調整します。パネルが小さすぎると、グラフ (スパークライン) が自動的に非表示になります。

値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示]

Grafana がデータを表示する方法を選択します。

- [計算]

  すべての行を元に計算した値を表示します。

  - [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算のリストについては、「標準計算」を参照してください。
  - [Fields] (フィールド) – 視覚化に表示するフィールドを選択します。
- [すべての値]

  行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

  - [制限] – 表示する行の最大数。デフォルトは 5,000 です。
  - [Fields] (フィールド) – 視覚化に表示するフィールドを選択します。

統計スタイル

視覚化のスタイルを変更します。

[Orientation] (向き)

積み重ね方向を選択します。

- [自動] – Grafana が最適な方向を選択します。
- [水平] – バーは水平方向に左から右に伸びます。
- [垂直] – バーは垂直に上から下に伸びます。

テキストモード

テキストモードオプションを使用して、視覚化がレンダリングするテキストを制御できます。値が重要ではなく、名前と色のみが重要である場合は、[テキストモード] を [名前] に変更します。値は色を決定するために引き続き使用され、ツールチップに表示されます。

- [自動] – データに複数の系列またはフィールドが含まれている場合は、名前と値の両方を表示します。
- [値] – 値のみを表示し、名前は表示しません。代わりに、カーソルのツールチップに名前が表示されます。

- [値と名前] – 常に値と名前を表示します。
- [名前] – 値の代わりに名前を表示します。値はカーソルのツールチップに表示されます。
- [なし] – 何も表示しません (空)。名前と値は、カーソルのツールチップに表示されます。

ワイドレイアウト

ワイドレイアウトを有効にするかどうかを設定します。ワイドレイアウトはデフォルトで有効になっています。

- [On] - ワイドレイアウトはオンになっています。
- [Off] – ワイドレイアウトはオフになっています。

> **Note**
>
> このオプションは、テキストモードが値と名前に設定されている場合にのみ適用されます。ワイドレイアウトをオンにすると、パネルが十分に広い場合、値と名前が右側の値と並行して表示されます。ワイドレイアウトをオフにすると、値は常に名前の下にレンダリングされます。

カラーモード

カラーモードを選択します。

- [なし] – 値には色は適用されません。
- [値] – 値とグラフ領域に色を適用します。
- [Background Gradient] (背景グラデーション) – わずかな背景グラデーションで、値、グラフ領域、背景に色を適用します。
- [Background Solid] (無地背景) – 無地の背景色で、値、グラフ領域、背景に色を適用します。

グラフモード

グラフとスパークラインモードを選択します。

- [なし] – グラフを非表示にし、値のみを表示します。
- [エリア] – 値の下に面グラフを表示します。これには、クエリが時間列を返す必要があります。

テキストの整列

アライメントモードを選択します。

- [自動] – 値が 1 つだけ表示されている場合 (繰り返しなし)、値は中央に配置されます。複数の系列または行が表示されている場合、値は左揃えになります。
- [Center] (中央) – 統計値は中央にあります。

変化率の表示

変化率を表示するかどうかを設定します。デフォルトでは表示されていません。

> **Note**
>
> このオプションは、値オプションの表示設定がすべての値に設定されている場合は適用されません。

テキストサイズ

ゲージテキストのサイズを調整します。

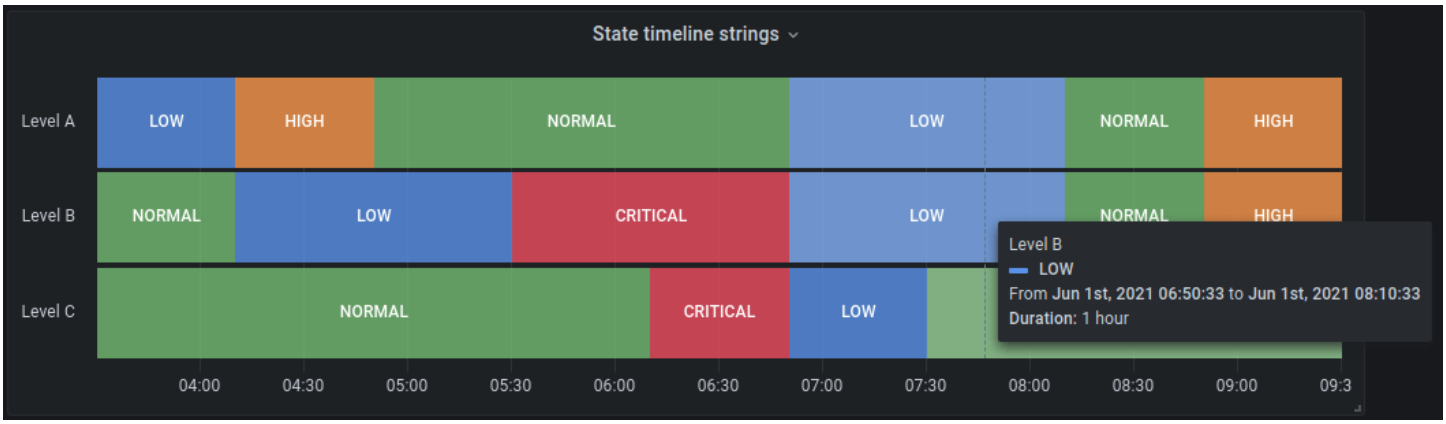
- [タイトル] – ゲージタイトルサイズの数値を入力します。
- [値] – ゲージ値サイズの数値を入力します。

状態タイムライン

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

状態タイムラインには、時間の経過に伴う個別の状態の変化が表示されます。各フィールドまたは系列は、一意の水平バンドとしてレンダリングされます。状態領域は、値の有無にかかわらずレンダリ

ングできます。このパネルは、文字列またはブール値の状態とうまく連携しますが、時系列でも使用できます。時系列で使用すると、しきい値を使用して数値を離散状態の領域に変換します。



状態タイムラインオプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[Merge equal consecutive values] (等しい連続値をマージする)

Grafana は、互いに隣接する同じ値をマージするかどうかを制御します。

[値の表示]

値が状態領域内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[値の整列]

状態領域内の値のアライメントを制御します。

[Row height] (行の高さ)

行間のスペースを制御します。1 = スペースなし = 0.5 = 50% のスペース。

[線の幅]

状態領域の線幅を制御します。

[塗りつぶしの不透明度]

状態領域の不透明度を制御します。

[null 値の接続]

null 値 (データ内のギャップ) をグラフに表示する方法を選択します。null 値は、連続線を形成するために接続することも、超えるとデータ内のギャップが接続されなくなるしきい値を設定することもできます。

- [なし] – データにギャップがある時系列データポイントは接続されません。
- [常に表示] – データにギャップがある時系列データポイントは常に接続されます。
- [しきい値] – データ内のギャップが接続されなくなるしきい値を指定します。これは、データ内の接続されたギャップが既知のサイズまたは既知の範囲内であり、この範囲外のギャップが接続されなくなった場合に便利です。

値の切断

超えるとデータ内の値を切断するしきい値を設定するかどうかを選択します。

- [なし] - データ内の時系列データポイントは切断されません。
- [しきい値] – データ内の値が切断されるしきい値を指定します。これは、データ内の目的の値が既知のサイズまたは既知の範囲内であり、この範囲外の値が接続されないようにする場合に便利です。

値のマッピング

ブール値または文字列値に色を割り当てるには、値マッピングを設定する を使用します。

しきい値を含む時系列データ

視覚化は時系列データでも使用できます。この場合、しきい値を使用して時系列を個別の色付き状態領域に変換します。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、[カラースキーム] を [しきい値起点] に設定します。

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。凡例の詳細については、「凡例の設定」を参照してください。

- [リスト] – 凡例を一覧として表示します。これはデフォルトモードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

ステータス履歴

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ステータス履歴は、時間の経過とともに周期的な状態を表示します。各フィールドまたは系列は、横の列としてレンダリングされます。ボックスがレンダリングされ、各値を中央に配置します。

サポートされているデータ

ステータス履歴は、文字列、ブーリアン値、数値フィールドまたは時系列で機能します。時間フィールドは必須です。値マッピングを使用して、文字列の色付け、またはテキスト値の数値範囲への割り当てができます。

表示オプション

これらのオプションで、視覚化する内容を絞り込みます。

[値の表示]

値が値ボックス内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[列幅]

ボックスの幅を制御します。1 = 最大スペース、0 = 最小スペース。

[線の幅]

状態領域の線幅を制御します。

[塗りつぶしの不透明度]

状態領域の塗りつぶしの不透明度を制御します。

値のマッピング

ブール値または文字列値に色を割り当てるには、値マッピングを設定する を使用します。

しきい値を含む時系列データ

パネルは時系列データでも使用できます。この場合、しきい値を使用してボックスを色付けします。グラデーションのカラースキームを使用して値を色付けすることもできます。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、[カラースキーム] を [しきい値起点] に設定します。

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。凡例の詳細については、「凡例の設定」を参照してください。

- [リスト] – 凡例を一覧として表示します。これはデフォルトモードです。
- [テーブル] – 凡例をテーブルとして表示します。

- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

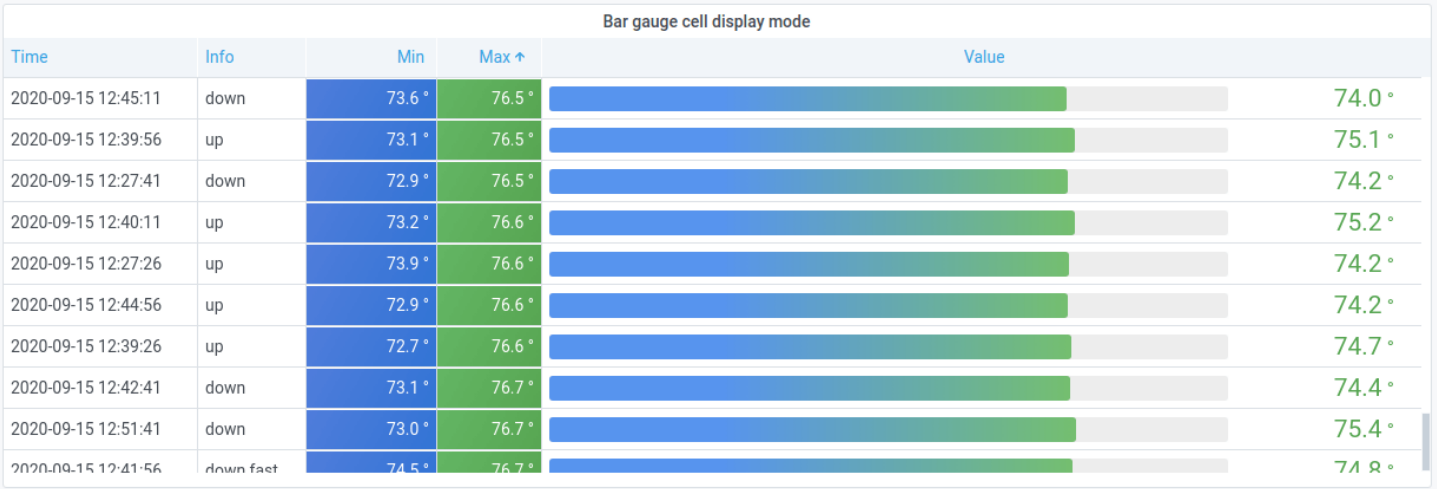
[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

[テーブル]

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

テーブルは非常に柔軟で、時系列やテーブルの複数のモード、注釈、未加工 JSON データをサポートしています。また、この視覚化では、日付の書式設定、値の書式設定、色付けオプションも可能です。



> **Note**
>
> 注釈とアラートはテーブルではサポートされていません。

列の並べ替え

列タイトルを選択して、ソート順序をデフォルトから降順、昇順に変更できます。列を選択するたびに、ソート順序がサイクルの次のオプションに変わります。追加で列を選択するときは、`shift` キーを長押しして複数の列をソートできます。

テーブルオプション

ヘッダーの表示

データソースからインポートされた列名を表示または非表示にします。

列幅

デフォルトでは、Grafana はテーブルサイズと最小列幅に基づいて列幅を自動的に計算します。このフィールドオプションは、設定をオーバーライドし、すべての列の幅をピクセル単位で定義できます。

例えば、`100` と入力すると、すべての列の幅が 100 ピクセルに設定されます (フィールドを終了すると変更が行われます)。

[最小列幅]

デフォルトでは、テーブル列の最小幅は 150 ピクセルです。このフィールドオプションでは、デフォルトを上書きでき、テーブルパネルの新しい最小列幅をピクセル単位で定義します。

例えば、最小値を 75 に設定すると、すべての列は幅 75 ピクセル以上でスケールされます。

スマートフォンやタブレットなどのスモールスクリーンデバイスの場合、デフォルトの `150` ピクセル値を `50` に減らして、テーブルベースのパネルをダッシュボードで正しくレンダリングできます。

[列の配置]

Grafana がセルの内容を調整する方法を選択します。

- Auto (デフォルト)
- Left (左)

- Center (中央)
- Right (右)

セルタイプ

デフォルトでは、Grafana は表示設定を自動的に選択します。以下のオプションのいずれかを選択して、すべてのフィールドのデフォルトを設定することで、設定を上書きできます。一部のセルタイプでは、追加の設定を使用できます。

> **Note**
>
> フィールドタブでこれらを設定すると、時間フィールドを含むすべてのフィールドにタイプが適用されます。オーバーライドタブで設定して、変更を 1 つ以上のフィールドに適用できます。

[文字色]

しきい値が設定されている場合、フィールドテキストは適切なしきい値の色で表示されます。

Color background (gradient or solid) (背景色 (グラデーションまたは無地))

しきい値が設定されている場合、フィールドの背景は適切なしきい値の色で表示されます。

ゲージ

セルは、複数の異なるプレゼンテーションタイプでグラフィカルゲージとして表示できます。

- [Basic] (基本) – 基本モードでは、ゲージの色を定義するしきい値レベルを含むシンプルなゲージが表示されます。
- グラデーション] – しきい値レベルはグラデーションを定義します。
- LCD – ゲージを小さなセルに分け、点灯または消灯させて表示します。

さらに、ゲージの横に表示されるラベルに対して、値で色付けするように設定したり、テーマテキストの色と一致させたり、非表示にしたりできます。

- Value color (値の色)
- [テキスト色]

- [非表示]

[JSON ビュー]

コードとしてフォーマットされた値を表示します。値がオブジェクトの場合、カーソルを合わせると JSON オブジェクトのブラウジングができる JSON ビューが表示されます。

スパークライン

スパークラインとしてレンダリングされた値を表示します。テーブルのデータ変換には時系列が必要です。

セル値の検査

テーブルセルから値の検査を有効にします。raw 値はモーダルウィンドウに表示されます。

> **Note**
>
> セル値の検査は、セル表示モードが [Auto]、[文字色]、[Color background]、または [ビュー] に設定されている場合にのみ使用できます。

[Column filter] (列フィルター)

列データの表示方法を一時的に変更できます。例えば、最大から最小までの値を順序付けたり、特定の値を非表示にしたりできます。詳細については、「テーブル列のフィルタリング」を参照してください。

Pagination (ページ分割)

ページ分割を有効または無効にするには、このオプションを使用します。これは、クエリに影響を与えないフロントエンドオプションです。有効にすると、ページサイズはテーブルの高さに自動的に調整されます。

テーブル列のフィルタリング

列フィルターをオンにすると、テーブルオプションをフィルタリングできます。

列のフィルタリングを有効にするには

1. Grafana で、フィルタリングする列を含むテーブルを使用してダッシュボードに移動します。

2. フィルタリングするテーブルパネルで、パネル工ディタを開きます。
3. [フィールド] タブを選択します。
4. [テーブル] オプションで、[列フィルター] オプションをオンにします。

各列タイトルの横にフィルター (ファネル) アイコンが表示されます。

列の値のフィルタリング

列の値をフィルタリングするには、列タイトルの横にあるフィルター (ファネル) アイコンを選択します。Grafana にその列のフィルターオプションが表示されます。

表示する値の横にあるチェックボックスを選択します。上部の検索フィールドにテキストを入力してそれらの値をディスプレイに表示すると、スクロールして探さずに選択できます。

複数の演算子から選択して、列の値を表示します。

- [Contains] (含む) – 正規表現パターン (デフォルトでは演算子) に一致します。
- [式] - ブール式を評価します。文字 `$` は、式にある列の値を表します (例: `$ >= 10 ≈& $ <= 12`)。
- [比較演算子] – 一般的な比較演算子 =、!=、<、<=、>、>= を使用できます。

[OK] ボタンと [Cancel] ボタンの上にあるチェックボックスを選択して、表示されているすべての値をフィルターに追加または削除します。

列フィルターのクリア

フィルターが適用された列では、タイトルの横に青い漏斗が表示されます。

フィルターを削除するには、青いファネルアイコンを選択し、[Clear filter] (フィルターのクリア) を選択します。

テーブルフッター

テーブルフッターを使用して、フィールドの計算を表示できます。

テーブルフッターを有効にしたら、[計算] を選択し、計算するフィールドを選択します。

フィールドを選択しない場合、システムはすべての数値フィールドに計算を適用します。

行のカウント

選択したフィールドの値の数ではなく、データセット内の行数を表示する場合は、[カウント] 計算を選択し、[カウント行] を有効にします。

テキスト

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

テキスト視覚化を使用すると、ダッシュボードにテキストまたは HTML を直接含めることができます。これは、コンテキスト情報と説明の追加や、複雑な HTML を埋め込むために使用できます。

モード

モードは、埋め込みコンテンツの表示方法を決定します。これには次のオプションがあります。

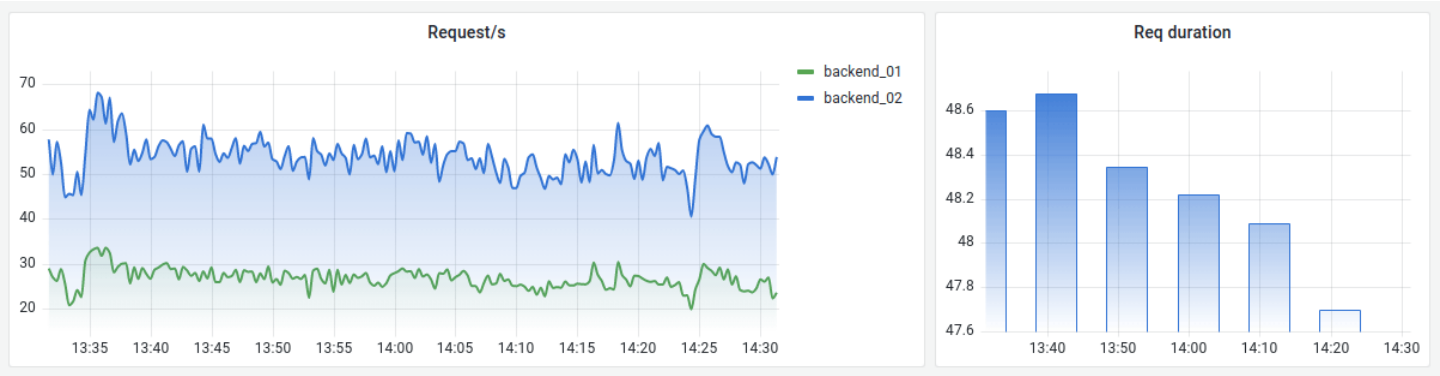
- [Markdown] – このオプションはコンテンツをマークダウンとしてフォーマットします。
- [HTML] – この設定では、コンテンツをサニタイズされた HTML としてレンダリングします。
- [Code] – この設定では、読み取り専用コードエディタ内でコンテンツをレンダリングします。適切な言語を選択して、埋め込みテキストに構文強調表示を適用します。

[変数]

コンテンツの変数が展開されて表示されます。

時系列

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。



時系列の視覚化は、時系列データをグラフとして視覚化するデフォルトかつ主要な方法です。系列を、線、ポイント、またはバーとしてレンダリングできます。ほぼすべての時系列データを表示するのに十分な汎用性を備えています。

> **Note**
>
> グラフパネルの視覚化は時系列の視覚化に移行できます。移行するには、[パネル]タブで [時系列の視覚化] を選択します。Grafana は該当するすべての設定を転送します。

トピック

- ツールチップオプション
- 凡例オプション
- グラフスタイル
- 軸オプション
- 色オプション

ツールチップオプション

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ツールチップオプションは、グラフのデータポイントにカーソルを合わせたときに表示される情報オーバーレイを制御します。

[ツールチップモード]

- [単独] – ツールチップには、カーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

凡例オプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

凡例オプションは、グラフの下または右側に表示される系列名と統計を制御します。

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。凡例の詳細については、「凡例を設定する」を参照してください。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。

- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

グラフスタイル

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このオプションを使用して、時系列データを表示する方法を定義します。オーバーライドを使用して、同じグラフに複数のスタイルを組み合わせることができます。

- [折れ線] – 時系列をグラフに折れ線で表示します。
- [棒] – 時系列をグラフに一連の棒で表示し、データポイントごとに 1 つずつ表示します。
- [ポイント] – 時系列をグラフに点で表示し、データポイントごとに 1 つずつ表示します。

[棒の配置]

グラフ上に描画されるデータポイントに対して棒の位置を設定します。棒には幅があるため、棒をデータポイントの前、後、または中央に配置することができます。このオプションの選択肢は次のとおりです。

- 
  [前] – 棒はポイントの前に描画されます。ポイントは棒の後端の隅に配置されます。
- 
  Center – 棒はポイントの周りに描画されます。ポイントは棒の中央に配置されます。これがデフォルトです。
- 
  After – 棒はポイントの後に描画されます。ポイントは棒の先頭隅に配置されます。

[線の幅]

線幅は、系列線の太さまたは棒のアウトラインを制御するスライダーです。

[塗りつぶしの不透明度]

塗りつぶし色の不透明度を設定します。塗りつぶしは、折れ線グラフの線の下にある領域を表示したり、棒グラフの棒の色として表示したりするために使用します。

[グラデーションモード]

グラデーションモードでは、系列の色に基づいてグラデーションの塗りつぶしを指定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。詳細については、「カラースキーム」を参照してください。

グラデーションモードのオプションは次のとおりです。

- [なし] – グラデーションの塗りつぶしなし。これはデフォルトの設定です。
- [不透明度] – Y 軸の値が増加するにつれて、塗りつぶしの不透明度が増加する不透明度グラデーション。
- [色相] – 系列の色相に基づく繊細なグラデーション。
- [Scheme] (スキーム) – カラースキームによって定義されるカラーグラデーション。この設定は、塗りつぶしエリアと線で使用できます。詳細については、「色オプション」を参照してください。

グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

[ポイント表示]

以下のオプションを使用して、線または棒にポイントを追加するように視覚化を設定できます。

- [自動] – Grafana はデータの密度に基づいてポイントを表示するかどうかを決定します。密度が低い場合、ポイントが表示されます。
- [常に表示] – データ密度に関係なく、ポイントが表示されます。
- [常に非表示] – ポイントを表示しません。

[ポイントサイズ]

ポイントのサイズを、直径 1 ~ 40 ピクセルに設定します。

線形補間

Grafana が系列の線形を補間する方法を選択します。



オプションは以下のとおりです。

- [線形] – ポイントは直線で結合されます。
- [Smooth] (滑らか) – ポイントは、ポイント間の移行が滑らかな曲線で結合されます。
- [前のステップ] – 線はポイント間のステップとして表示されます。ポイントはステップの最後にレンダリングされます。
- [Step after] – 線はポイント間のステップとして表示されます。ポイントはステップの最初にレンダリングされます。

線のスタイル

線のスタイルを設定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。

線のスタイルの選択肢は次のとおりです。

- [Solid] – 実線を表示します。これはデフォルトの設定です。
- [Dash] – 破線を表示します。このオプションを選択すると、線ダッシュの長さとギャップ (長さ, ギャップ) を選択するためのリストが表示されます。ダッシュ間隔はデフォルトで (10, 10) に設定されています。
- [Dots] - 点線を表示します。このオプションを選択すると、点の間隔のギャップ長を選択するためのリストが表示されます。点の間隔はデフォルトで 10 に設定されています。

null 値の接続

null 値 (データ内のギャップ) をグラフに表示する方法を選択します。null 値は、連続線を形成するために接続することも、超えるとデータ内のギャップが接続されなくなるしきい値を設定することもできます。

null 値を接続する方法の選択肢は次のとおりです。

- [なし] – データにギャップがある時系列データポイントは接続されません。

- [常に表示] – データにギャップがある時系列データポイントは常に接続されます。
- [しきい値] – データ内のギャップが接続されなくなるしきい値を指定します。これは、データ内の接続されたギャップが既知のサイズまたは既知の範囲内であり、その範囲外のギャップは接続されないようにする場合に便利です。

値の切断

指定されたしきい値を超える時間があるデータの値との間に、ギャップを追加するかどうかを選択します。

切断値の選択肢は次のとおりです。

- [なし] – 時系列データポイントは切断されません。
- [しきい値] データ内の値が切断されるしきい値を指定します。これは、データ内の目的の値が既知のサイズまたは既知の範囲内であり、この範囲外の値が接続されないようにする場合に便利です。

[Stack series] (系列の積み上げ)

積み上げにより、Grafana は系列の上に別の系列を表示できます。視覚化で積み上げを使用すると、誤解を招くグラフを簡単に作成できるため、注意が必要です。積み上げが最適な手段ではない理由の詳細については、「積み上げに関する問題」を参照してください。

積み上げの選択肢は次のとおりです。

- [Off] (オフ) – 系列の積み上げをオフにします。
- [普通] – 系列の上に別の系列を積み上げます。
- [100%] – パーセンテージで積み上げるため、すべての系列の合計が 100% になります。

グループ内の系列の積み上げ

積み上げ動作をオーバーライドして、グループ内で系列を積み上げることができます。オーバーライドの作成の詳細については、「フィールドのオーバーライドを設定する」を参照してください。

グループ内の系列を積み上げるには

1. パネルを編集し、[オーバーライド] を選択します。
2. [Stack series (系列の積み上げ)] オプションのフィールドオーバーライドを作成します。

3. 積み上げモードで[普通] を選択します。
4. 系列を表示する積み上げグループに名前を付けます。

   積み上げグループ名オプションは、オーバーライドを作成する場合にのみ使用できます。

[下を塗りつぶす]

[下を塗りつぶす]オプションは、2 つの系列間の領域を塗りつぶします。このオプションは、系列またはフィールドのオーバーライドとしてのみ使用できます。

このオプションを使用すると、系列の折れ線から 0 までではなく、2 つの系列間の領域を塗りつぶすことができます。たとえば、Max と Min という 2 つのシリーズがある場合、Max シリーズを選択し、それを以下のフィルに上書きして Min シリーズにすることができます。これにより、2 つの折れ線間の領域のみが塗りつぶされます。

軸オプション

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

軸カテゴリのオプションにより、x 軸と y 軸のレンダリング方法が変わります。一部のオプションは、編集するフィールドオプションボックスの外部をクリックするまで有効になりません。Enter キーを押す方法もあります。

[Time zone] (タイムゾーン)

X 軸に沿って表示するタイムゾーンを設定します。

[配置]

Y 軸の配置を選択します。オプションは以下のとおりです。

- [自動] – Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。

- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。

軸を選択的に非表示にするには、特定のフィールドを対象とするフィールドオーバーライドを追加します。

ラベル

Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。

[幅]

軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。この設定により、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。

[グリッド線の表示]

軸グリッド線の可視性を設定します。

- [自動] – データの密度に基づいてグリッド線を自動的に表示します。
- [On] - 常にグリッド線を表示します。
- [Off] – 常にグリッド線を表示しません。

[色]

軸の色を設定します。

- [テキスト] – テーマテキストの色に基づいて色を設定します。
- [系列] – 系列の色に基づいて色を設定します。

[境界線の表示]

軸の境界の可視性を設定します。

スケール

Y 軸値のスケール方法を設定します。

- [線形] – 目盛を等分します。
- [対数] – 対数目盛を使用します。このオプションを選択すると、二進対数 (底 2) または常用対数 (底 10) スケールを選択するためのリストが表示されます。
- [Symlog] – 正負対象の対数目盛を使用します。このオプションを選択すると、二進対数 (底 2) または常用対数 (底 10) スケールを選択するためのリストが表示されます。線形しきい値オプションを使用すると、スケールが線形から対数に変化するしきい値を設定できます。

[Centered zero] (中央ゼロ)

Y 軸においてゼロが中央になるように設定します。

[ソフト最小値とソフト最大値]

Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて y 軸の範囲を自動的に設定します。

ソフト最小値とソフト最大値の設定により、データがほぼフラットな場合に、データの小さな変化が拡大されるのを防げます。対照的に、ハード最小値とハード最大値は、特定のポイントを超える断続的なスパイクを切り取ることで、データ内の有用な詳細が不明瞭になるのを防げます。

Y 軸のハード制限を定義するには、標準の最小/最大オプションを設定します。詳細については、「標準オプションの設定」を参照してください。

変換

このオプションを使用して、ツールチップ、コンテキストメニュー、または凡例に表示される値に影響を与えずに系列の値を変換します。変換オプションは 2 つあります。

- [負の Y 変換] – 結果を Y 軸の負の値に反転します。
- [Constant] (定数) – 最初の値を定数線として表示します。

> **Note**
>
> 変換オプションはオーバーライドとしてのみ使用できます。

[Display multiple y-axes] (複数の Y 軸を表示)

複数の Y 軸を表示したい場合もあります。例えば、経時的な温度と湿度の両方を示すデータセットがある場合、これらの 2 つの系列で異なる単位を持つ 2 つの y 軸を表示できます。

複数の Y 軸を表示するには、フィールドオーバーライドを追加します。必要な回数だけステップに従うことで、必要な数の y 軸を追加できます。

色オプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

デフォルトでは、グラフは標準のカラースキームオプションを使用して系列の色を割り当てます。凡例にある系列の色アイコンを選択することで、凡例を使用してカラーピッカーを開くこともできます。この方法で色を設定すると、特定の系列に特定の色を設定するオーバーライドルールが自動的に作成されます。

以下は、系列の色のデフォルトを変更するために使用できる追加オプションです。

Classic palette (クラシックパレット)

最も一般的なセットアップは、グラフにクラシックパレットを使用することです。このスキームは、その順序に基づいて、各フィールドまたは系列に色を自動的に割り当てます。クエリでフィールドの順序が変更されると、色も変わります。オーバーライドルールを使用して、特定のフィールドの色を手動で設定できます。

[単一色]

このモードを使用して色を指定します。凡例の各系列の横にある色付き線アイコンを選択して、カラーピッカーを開くこともできます。これにより、カラースキームを単一色と選択した色に設定する新しいオーバーライドが自動的に作成されます。

値別カラースキーム

[しきい値起点: 値別] や [緑黄赤: 値別] などの値別カラースキームを選択すると、[系列に使用する色] オプションが表示されます。このオプションは、系列に色を割り当てるためにどの値 (最終値、最小値、最大値) を使用するかを管理します。

スキームグラデーションモード

グラフスタイルの下にあるグラデーションモードオプションには、Scheme という名前のモードがあります。スキームを有効にすると、折れ線または棒グラフは、選択したカラースキームから定義されたグラデーション色を受け取ります。

[しきい値起点]

カラースキームがしきい値起点 (値別) に設定され、グラデーションモードがスキームに設定されている場合、折れ線または棒グラフの色は定義されたしきい値を超えると変化します。スキームで選択された正確な色のみが表示されます。

グラデーションカラースキーム

グラデーションモードをスキームに設定せずにグラデーションカラースキームを使用すると、系列内の値がしきい値セット間を移動するにつれて、選択した色間でグラデーションを形成します。

トレース

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

トレースの視覚化により、インフラストラクチャ内のサービスを横断するリクエストに従うことができます。トレースの視覚化は、トレースデータを簡単に解釈できる図でトレースデータを表示します。

トレースとその使用方法の詳細については、次のドキュメントを参照してください。

- Explore でのトレース

- Tempo データソース
- Grafana Labs Tempo ドキュメントの「Tempo での使用開始」。

トレースの視覚化を含むパネルの追加

Grafana スタックでトレースデータが利用可能になると、Grafana ダッシュボードにトレースパネルを追加できます。

ダッシュボード変数 `traceID` を使用すると、特定のトレース ID に対する特定のトレースを表示するクエリを作成できます。ダッシュボード変数の詳細については、「 変数ドキュメント」を参照してください。

前提条件

開始するには、以下が必要です。

- Amazon Managed Grafana ワークスペース。
- ワークスペースに接続された Tempo データソース。

ダッシュボード内のトレースデータを表示および分析するには、ダッシュボードにトレースの視覚化を追加し、パネルエディタを使用してクエリを定義する必要があります。クエリは、視覚化に表示されるデータを決定します。パネルエディタの詳細については、「パネルエディタのドキュメント」を参照してください。

この手順では、ダッシュボード変数とテンプレートを使用して、視覚化できるトレース ID を入力できます。`traceId` という変数を使用して、テンプレートクエリとして追加します。

トレース視覚化クエリを追加するには

1. ワークスペースで、新しいダッシュボードを作成するか、トレースの視覚化を追加する既存のダッシュボードに移動します。
2. 新しいダッシュボードから [視覚化を追加] を選択するか、既存のダッシュボードで[パネルの追加] を選択します。
3. 適切なトレースデータソースを選択します。
4. パネルエディタの右上で [視覚化] タブを選択し、トレースを選択します。
5. パネルオプションで、トレースパネルのタイトルを入力します。パネルエディタの詳細については、「パネルオプションの設定」を参照してください。
6. クエリエディタで、TraceQL クエリタイプタブを選択します。

7. TraceQLクエリフィールドに `${traceId}` を入力して、ダッシュボード変数を作成します。この変数はテンプレートクエリとして使用されます。
8. パネルエディタで [適用] を選択して、ダッシュボードにパネルを追加します。
9. 必要に応じて、ダッシュボードの設定に移動し、`traceId` という名前の新しい変数を変数タイプカスタムで追加し、ラベルを付与します。[適用] を選択して、ダッシュボードに変数を追加します。
10. トレースパネルに使用されるデータソースに有効なトレース ID を使用し、ダッシュボード変数で ID を編集して、パネルが機能することを確認します。

テーブルの視覚化による TraceQL の追加

ダッシュボードにトレースの視覚化を追加することはできますが、ダッシュボード変数としてトレース ID を手動で追加しなければならないのは面倒です。代わりに、TraceQL クエリを使用して特定のタイプのトレースを検索し、一致する結果から適切なトレースを選択できる方が便利です。

前提条件

この手順は、前の手順を完了したことを前提としています。

テーブル視覚化で TraceQL を追加するには

1. トレース視覚化を追加したのと同じダッシュボードで、[パネルの追加] を選択して新しい視覚化パネルを追加します。
2. 前のセクションで使用したのと同じトレースデータソースを選択します。
3. パネルエディタの右上で [視覚化] タブを選択し、テーブルを選択します。
4. クエリエディタで、[TraceQL] タブを選択します。
5. パネルオプションで、トレースパネルのタイトルを入力します。
6. ダッシュボードで視覚化するトレースを検索するには、適切な TraceQL クエリを追加します。例えば、*`my-server`* は、サーバーからのシンプルで静的なクエリです。

   ```
   { .service.name = "my-server" && .http.status_code=500 }
   ```

   TraceQL クエリは、他のダッシュボード変数が存在する場合、それらを利用するテンプレートクエリとして記述できます。これにより、これらの変数に基づいて動的クエリを作成できます。

クエリから結果が返されると、結果はパネルのテーブルにレンダリングされます。

トレース視覚化の結果には、トレースをレンダリングする Explore ページへのリンクが含まれます。選択時に traceId ダッシュボード変数を埋めるテーブル内のトレースに他のリンクを追加して、トレースが同じダッシュボードで視覚化されるようにできます。

パネルに一連のデータリンクを作成するには、次の手順を使用します。

変数を使用してトレースに他のリンクを追加するには

1. 右側のメニューにある [データリンク] で、[リンクの追加] を選択します。
2. データリンクのタイトルを追加します。
3. 完全なダッシュボードがレンダリングされているときにブラウザのアドレスバーを見て、ダッシュボードへのパスを見つけます。これは同じ Grafana スタック内のダッシュボードへのリンクであるため、ダッシュボードのパスのみが必要です。

   例えば、パスが以下の場合:

   ```
   https://g-example.grafana-workspace.us-east-1.amazonaws.com/d/1234abcd5/my-dashboard?orgId=1
   ```

   ダッシュボードへのパスは次のとおりです。

   ```
   /d/1234abcd5/my-dashboard?orgId=1
   ```

4. URL フィールドで、両方のパネルを含むダッシュボードを自己参照します。この自己参照では、テーブル内の選択したトレースの値を使用してダッシュボード変数を入力します。前のステップで見つけたダッシュボードのパスを使用して、TraceQL テーブルから選択した結果を使用する traceId の値を入力します。トレース ID は、返された結果の traceID データフィールドを使用して公開されるため、ダッシュボード変数の値として使用します。例:

   ```
   /d/1234abcd5/my-dashboard?orgId=1&var-traceId=${__data.fields["traceID"]}
   ```

5. [保存] を選択すると、データリンクを保存できます。
6. パネル工ディタから [適用] を選択すると、ダッシュボードにパネルを適用できます。
7. ダッシュボードを保存します。

これで、テーブルの視覚化に一致するトレースのリストが表示されます。TraceID フィールドまたは SpanID フィールドを選択すると、トレースを視覚化する Explore ページを開くか、データリンクをたどるオプションが表示されます。他のフィールド (Start time、Name、 Durationなど) を選択

すると、自動的にデータリンクが続き、`traceId` ダッシュボード変数に入力されることで、トレースパネルに関連するトレースが表示されます。

トレンド

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

トレンドの視覚化は、時間ではない連続した数値 X を持つデータセットに使用する必要があります。例としては、関数グラフ、rpm/トルク曲線、供給/需要関係、レースコースに沿った上昇または心拍数プロット (x は開始からの距離または期間) などがあります。

トレンドの視覚化は、以下の例外を除いて、時系列の視覚化で利用可能なすべてのビジュアルスタイルとオプションをサポートします。

- 注釈や時間リージョンなし
- 共有カーソル (または十字線) なし
- マルチタイムゾーン x 軸なし
- ドラッグ選択によりダッシュボードの時間範囲の変更不可

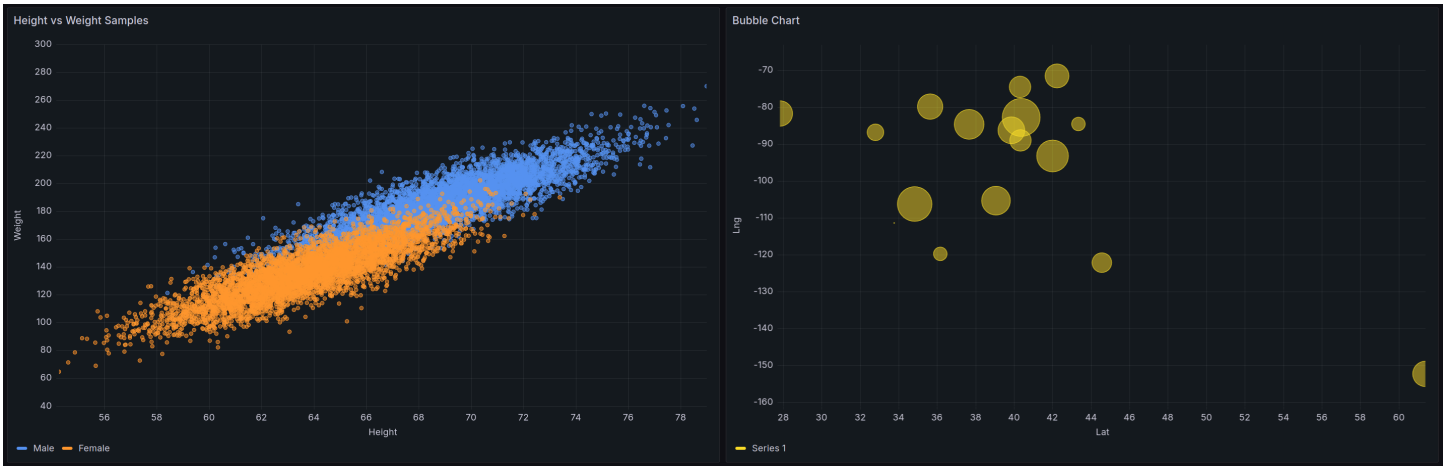
X フィールドの選択

このオプションを使用して、増加する数値を含むフィールドを選択します。

例えば、回転速度を x 軸にプロットし、出力とトルクを y 軸にプロットすることで、エンジンの回転速度に対する出力とトルクを表すことができます。

XY チャート

XY チャートは、グラフ内の任意の x 値と y 値を視覚化する方法を提供し、2 つの変数間の関係を簡単に表示できます。XY チャートは通常、散布図の作成に使用されます。また、それらを使用してバブルチャートの作成も可能です。ここでは、フィールド値によって各バブルのサイズが決まります。



サポートされているデータ形式

XY チャートには、少なくとも 2 つの数値フィールドを持つ任意の表形式データタイプを使用できます。このタイプの視覚化には時間データは必要ありません。

パネルオプション

パネルエディタペインの [パネルオプション] セクションで、パネルのタイトルや説明などの基本的なオプションを設定します。このセクションで繰り返しパネルの設定もできます。詳細については、「パネルオプションの設定」を参照してください。

XY チャートオプション

系列マッピング

チャートで系列データをマッピングする方法を設定します。

- [自動] – 使用可能なすべてのデータフレーム (またはデータセット) から系列を自動的に生成します。フィルタリングして 1 つのフレームのみを選択できます。
- [手動] – 使用可能なデータフレームから選択して、系列を明示的に定義します。

系列マッピングの選択に応じて、フレーム、X フィールド、および Y フィールドオプションが異なります。自動および手動の系列マッピングセクションでは、これらのさまざまなオプションについて説明します。

自動系列マッピングオプション

系列マッピングモードとして自動を選択すると、以下のオプションが事前設定されますが、自分で定義することもできます。

- [Frame] – デフォルトでは、XY チャートにすべてのデータフレームが表示されます。フィルタリングして 1 つのフレームのみを選択できます。
- [X-field] – X が表すフィールドを選択します。デフォルトでは、各データフレームの最初の数値フィールドです。
- [Y-field] – X フィールドが設定されると、デフォルトでは、データフレーム内の残りのすべての数値フィールドが Y フィールドとして指定されます。このオプションを使用して、Y に使用するフィールドを明示的に選択できます。

  一連のチャートは Y フィールドから生成されます。XY チャートの系列を変更するには、Y フィールドをオーバーライドします。サイズフィールドまたは色フィールドで使用するフィールドは、系列を生成しません。

  オーバーライドを使用して Y フィールドを個別に除外することもできます。これを行うには、削除する各 Y フィールドに対して、次のプロパティを持つオーバーライドを追加します。
  - オーバーライドタイプ: [名前付きフィールド]
  - オーバーライドプロパティ: [系列 > エリアで非表示]
  - エリア: Viz

手動系列マッピングオプション

系列モードとして手動を選択すると、系列を追加、編集、削除できます。系列を管理するには、[Series] (系列) フィールドを選択します。系列の名前を変更するには、系列名を選択します。

手動モードでは、次のオプションを設定する必要があります。

- [Frame] – データフレームまたはデータセットを選択します。フレームは必要な数だけ追加できます。
- [X-field] – X が表すフィールドを選択します。
- [Y-field] – Y が表すフィールドを選択します。

[サイズフィールド]

このオプションを使用して、チャート内のポイントのサイズを制御するフィールドの値を設定します。この値は、データフレーム内のすべての値の最小値と最大値に関連しています。

このオプションを選択すると、最小および最大ポイントサイズオプションを設定できます。

[色フィールド]

このオプションを使用して、チャート内のポイントの色を制御するフィールドの値を設定します。[標準オプション]で色値オプションを使用するには、このフィールドを設定する必要があります。

通常、このオプションは、チャートに 1 つの系列しか表示されていない場合に使用されます。

[表示]

視覚化で値がどのように表されるかを設定します。

- [ポイント] – 値をポイントとして表示します。このオプションを選択すると、ポイントサイズオプションも表示されます。
- [線] – 値の間に線を追加します。このオプションを選択すると、線のスタイルと線幅のオプションも表示されます。
- [Both] (両方) - ポイントと線の両方を表示します。

[ポイントサイズ]

チャート内のすべてのポイントのサイズを、直径 1 ~ 100 ピクセルに設定します。デフォルトのサイズは 5 ピクセルです。オーバーライドを設定して、系列 (Y フィールド) ごとにピクセルサイズを設定できます。

最小/最大ポイントサイズ

これらのオプションを使用して、サイズフィールドオプションを設定するときに、最小または最大ポイントサイズを制御できます。これらのオプションを特定の系列に対してオーバーライドできます。

[線のスタイル]

線のスタイルを設定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。

- [Solid] – 実線を表示します。これはデフォルトの設定です。
- [Dash] – 破線を表示します。このオプションを選択すると、ドロップダウンリストが表示され、ダッシュ線の長さとギャップの設定を選べます。デフォルトでは、長さとギャップは `10, 10` に設定されます。
- [Dots] - 点線を表示します。このオプションを選択すると、点の間隔を選択できるドロップダウンリストが表示されます。デフォルトでは、点の間隔は `0, 10` に設定されています (最初の数値はドット長を表し、常にゼロです)。

[線の幅]

線の幅をピクセル単位で設定します。

ツールチップオプション

ツールチップオプションは、グラフのデータポイントにカーソルを合わせたときに表示される情報オーバーレイを制御します。

[ツールチップモード]

- [単独] – ツールチップには、カーソルを合わせている系列が 1 つだけ表示されます。
- [非表示] – グラフを操作するときにツールチップを表示しません。

オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

[最大高さ]

ツールチップボックスの最大高さを設定します。デフォルトは 600 ピクセルです。

凡例オプション

凡例オプションは、グラフの下または右側に表示される系列名と統計を制御します。凡例の詳細については、「凡例を設定する」を参照してください。

[可視性]

スイッチを切り替えて、凡例をオンまたはオフにします。

[モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。

[配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[値]

凡例に表示する標準計算を選択します。複数選択できます。

[幅]

凡例を視覚化の右側に配置するときの幅を制御します。このオプションは、凡例の配置を右に設定した場合にのみ表示されます。

軸オプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

軸カテゴリのオプションにより、x 軸と y 軸のレンダリング方法が変わります。一部のオプションは、編集するフィールドオプションボックスの外部をクリックするまで有効になりません。Enter キーを押す方法もあります。

y 軸の配置

Y 軸の配置を選択します。オプションは以下のとおりです。

- [自動] – Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。
- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。

軸を選択的に非表示にするには、特定のフィールドを対象とするフィールドオーバーライドを追加します。

ラベル

Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。

[幅]

軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。この設定により、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。

[グリッド線の表示]

軸グリッド線の可視性を設定します。

- [自動] – データの密度に基づいてグリッド線を自動的に表示します。
- [On] - 常にグリッド線を表示します。
- [Off] – 常にグリッド線を表示しません。

[色]

軸の色を設定します。

- [テキスト] – テーマテキストの色に基づいて色を設定します。
- [系列] – 系列の色に基づいて色を設定します。

[境界線の表示]

軸の境界の可視性を設定します。

[Scale] (スケール)

Y 軸値のスケール方法を設定します。

- [線形] – 目盛を等分します。
- [対数] – 対数目盛を使用します。このオプションを選択すると、二進対数 (底 2) または常用対数 (底 10) スケールを選択するためのリストが表示されます。
- [Symlog] – 正負対象の対数目盛を使用します。このオプションを選択すると、二進対数 (底 2) または常用対数 (底 10) スケールを選択するためのリストが表示されます。線形しきい値オプションを使用すると、スケールが線形から対数に変化するしきい値を設定できます。

[Centered zero] (中央ゼロ)

Y 軸においてゼロが中央になるように設定します。

[ソフト最小値とソフト最大値]

Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて y 軸の範囲を自動的に設定します。

ソフト最小値とソフト最大値の設定により、データがほぼフラットな場合に、データの小さな変化が拡大されるのを防げます。対照的に、ハード最小値とハード最大値は、特定のポイントを超える断続的なスパイクを切り取ることで、データ内の有用な詳細が不明瞭になるのを防げます。

Y 軸のハード制限を定義するには、標準の最小/最大オプションを設定します。詳細については、「標準オプションの設定」を参照してください。

[Transform] (変換)

このオプションを使用して、ツールチップ、コンテキストメニュー、または凡例に表示される値に影響を与えずに系列の値を変換します。変換オプションは 2 つあります。

- [負の Y 変換] – 結果を Y 軸の負の値に反転します。
- [Constant] (定数) – 最初の値を定数線として表示します。

> **Note**
>
> 変換オプションはオーバーライドとしてのみ使用できます。

[Display multiple y-axes] (複数の Y 軸を表示)

複数の Y 軸を表示したい場合もあります。例えば、経時的な温度と湿度の両方を示すデータセットがある場合、これらの 2 つの系列で異なる単位を持つ 2 つの y 軸を表示できます。

複数の Y 軸を表示するには、フィールドオーバーライドを追加します。必要な回数だけステップに従うことで、必要な数の y 軸を追加できます。

標準オプション

パネルエディタの [標準オプション] を使用すると、視覚化でのフィールドデータの表示方法を変更できます。標準オプションを設定すると、すべてのフィールドまたは系列に変更が適用されます。

フィールドの表示をより詳細に制御するには、「フィールドのオーバーライドを設定する」を参照してください。

以下の標準オプションをカスタマイズできます。

- [フィールドの最小値/最大値] – [フィールドの最小値/最大値] を有効にすると、フィールドの最小値または最大値に基づいて、Grafana が各フィールドの最小値または最大値を個別に計算できるようになります。
- カラースキーム – 視覚化全体に単一または複数の色を設定します。

詳細については、「標準オプションを設定する」を参照してください。

データリンク

データリンクを使用すると、ソースパネルのコンテキストを維持しながら、他のパネル、ダッシュボード、外部リソースにリンクできます。カーソルの下に系列名または値を含むリンクを作成できます。

データリンクごとに、次のオプションを設定します。

- タイトル
- [URL]
- [新しいタブで開く]

詳細については、「データリンクの設定」を参照してください。

フィールドのオーバーライド

オーバーライドを使用すると、特定のフィールドまたは系列に対して視覚化設定をカスタマイズできます。オーバーライドルールを追加すると、特定のフィールドセットをターゲットにし、それらのフィールドの表示方法に複数のオプションを定義できます。

以下のオーバーライドオプションから 1 つ選択してください。

- [名前付きフィールド] – 使用可能なすべてのフィールドの一覧からフィールドを選択します。
- [正規表現と一致する名前付きフィールド] – 正規表現でオーバーライドするフィールドを指定します。
- [タイプ別フィールド] – 文字列、数値などのフィールドをタイプ別 (文字列、数値、時間) に選択します。

- [クエリによって返されるフィールド – 特定のクエリによって返されるすべてのフィールドを選択します。
- [値を持つフィールド)] - 最小値、最大値、カウント、合計 など、定義された reducer 条件によって返されるすべてのフィールドを選択します。

詳細については、「フィールドのオーバーライドを設定する」を参照してください。

Grafana バージョン 10 で探索する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana のダッシュボード UI は、視覚化用のダッシュボードを構築する機能を提供します。Explore は、クエリに集中できるように、ダッシュボードとパネルのオプションを取り除きます。これにより、クエリが動作するまで反復処理を行い、クエリからダッシュボードを構築できます。

Note

ダッシュボードを作成せず単にデータを探索したい場合は、Explore を使用するとはるかに簡単になります。データソースがグラフとテーブルのデータをサポートしている場合、Explore は結果をグラフとテーブルの両方で表示します。これにより、データの傾向と詳細を同時に確認できます。

このページでは、データの探索を開始します。以下のトピックでは、Explore の特定の機能と使用方法について詳しく説明します。

- Explore でのクエリ管理
- Explore のログ
- Explore でのトレース

- Explore の相関エディタ
- Explore の Inspector

探索を開始する

> **Note**
>
> Explore にアクセスするには、エディタまたは管理者ロールが必要です。

Explore にアクセスするには

1. Grafana ワークスペースで、左側のメニューバーから Explore メニュー項目を選択します。

   空の Explore タブが開きます。

   または、パネル内の既存のクエリから開始するには、パネルメニューから Explore オプションを選択します。これにより、パネルからのクエリを含む Explore タブが開き、ダッシュボードの外部でクエリを微調整または反復できます。
2. 左上のドロップダウンからデータソースを選択します。

   また、[Open advanced data source picker(高度なデータソースピッカーを開く)] を選択して、データソースの追加 (管理者のみ可能) など、その他のオプションを表示することもできます。
3. 選択したデータソースから提供されたクエリエディタを使用してクエリを書き込みます。

   クエリの詳細については、「データのクエリと変換」を参照してください。
4. 右上隅のボタンを使用してクエリを実行します。

分割して比較する

分割ビューを使用すると、簡単に視覚化を並べて比較したり、関連データを 1 ページでまとめて確認したりできます。

上部で分割ビューを開く

1. Explore ビューで、[分割] ボタンを選択して現在のクエリを複製し、ページを 2 つの並列クエリに分割します。

> **Note**
>
> 新しいクエリには別のデータソースを選択できます。例えば、2 つの異なるサーバーに対して同じクエリを比較したり、ステージング環境を本番環境と比較したりできます。

分割ビューでは、いずれかのタイムピッカーにある時刻同期ボタンを選択することで、両方のパネルのタイムピッカーを連携できます (一方を変更すると、他方も変更されます)。タイムピッカーを連携すると、分割ビュークエリの開始時刻と終了時刻が同期されます。これにより、両方の分割パネルで同じ時間間隔を確認できます。

2. 新しく作成したクエリを閉じるには、[分割を閉じる] ボタンを選択します。

コンテンツ概要

コンテンツ概要は、Explore で作成したクエリと視覚化を追跡するサイドナビゲーションバーです。これにより、クエリと視覚化との間をすばやく移動できます。

コンテンツ概要は、分割ビューでも機能します。分割ビューでは、コンテンツ概要がペインごとに生成されます。

コンテンツ概要を開くには

1. Explore 画面の左上隅にある [概要] ボタンを選択します。
2. コンテンツ概要でパネルアイコンを選択して、そのパネルに移動します。

Explore の URL を共有する

Explore を使用すると、クエリを変更した際にブラウザのアドレスバーの URL が更新されます。この URL を共有またはブックマークできます。

> **Note**
>
> Explore は比較的長い URL を生成する場合があります。URL がお使いのツールに対して長すぎる場合は、短縮リンクを生成して共有することもできます。

外部ツールから Explore の URLを生成する

Explore URL には定義された構造があるため、外部ツールから URL を構築し、Grafana で開くことができます。URL 構造は次のとおりです。

```
http://<workspace_url>/explore?
panes=<panes>&schemaVersion=<schema_version>&orgId=<org_id>
```

各パラメータの意味は次のとおりです。

- org_id は組織 ID です。
- schema_version はスキーマバージョンです (最新バージョン 1 に設定する必要があります)。
- panes はペインの URL エンコードされた JSON オブジェクトで、各キーはペイン ID、各値は次のスキーマに一致するオブジェクトです。

```
{
  datasource: string; // the pane's root datasource UID, or `-- Mixed --` for mixed
 datasources
  queries: {
    refId: string; // an alphanumeric identifier for this query, must be unique
 within the pane, i.e. "A", "B", "C", etc.
    datasource: {
      uid: string; // the query's datasource UID ie: "AD7864H6422"
      type: string; // the query's datasource type-id, i.e: "loki"
    }
    // ... any other datasource-specific query parameters
  }[]; // array of queries for this pane
  range: {
    from: string; // the start time, in milliseconds since epoch
    to: string; // the end time, in milliseconds since epoch
  }
}
```

> **Note**
>
> from および to フィールドは、ダッシュボードの時間範囲の設定 トピックで説明されている相対範囲も受け入れます。

短縮リンクを共有する

短縮リンクの共有機能を使用すると、クエリパラメータで長い URL を使用する代わりに、/goto/:uid 形式のより短くてシンプルな URL を作成できます。クエリ結果への短縮リンクを作成するには、Explore ツールバーで [共有] オプションを選択します。使用されない短縮リンクは、7 日後に自動的に削除されます。リンクが 1 回以上使用されている場合は削除されません。

短縮リンクを絶対時間で共有する

短縮リンクには、相対時間 (2 時間前から現在の時刻など) または絶対時間 (午前 8 時から午前 10 時など) の 2 つのオプションがあります。デフォルトでは短縮リンクを共有すると、選択した相対時間範囲または絶対時間範囲がコピーされます。短縮リンクの共有ボタン横にあるドロップダウンボタンを選択し、[Time-Sync URL Links] (時刻同期 URL リンク) でオプションのいずれかを選択すると、絶対時間で短縮リンクを作成できます。つまり、リンクを受け取ったすべてのユーザーには、別の時間にリンクを開いても、現在ご覧のものと同じデータが表示されます。このオプションは、Explore ビューで選択した時間範囲には影響しません。

Explore でのクエリ管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Explore で作成したクエリを管理できます。これには、実行したクエリの履歴や、スターを付けたクエリが含まれます。

クエリ履歴

クエリ履歴は、Explore で使用したクエリのリストです。履歴は Grafana データベースに保存され、他のユーザーと共有されません。履歴内のクエリの保持期間は 2 週間です。2 週間以上経過したクエリは自動的に削除されます。履歴を開いて操作するには、Explore の [クエリ履歴] ボタンを選択します。

> **Note**
>
> スター付き (お気に入り) クエリは 2 週間の保持期間の対象ではなく、削除されません。

クエリ履歴の表示

クエリ履歴を使用すると、クエリの履歴を表示できます。個々のクエリごとに、次のことができます。

- クエリを実行します。
- コメントを作成および/または編集します。
- クエリをクリップボードにコピーする。
- クエリを含む短縮リンクをクリップボードにコピーする。
- クエリをスター付き (お気に入り) にする。

お気に入りのクエリを管理する

[クエリ履歴] タブにあるスター付きのすべてのクエリは、[スター付き] タブに表示されます。これにより、お気に入りのクエリにすばやくアクセスし、ゼロから入力することなくこれらのクエリを再利用できます。

[クエリ履歴の並べ替え]

デフォルトでは、クエリ履歴には最新のクエリが表示されます。履歴は日付またはデータソース名で、昇順または降順に並べ替えることができます。

クエリ履歴を並べ替えるには

1. [クエリの並べ替え順序] フィールドを選択します。
2. 以下のオプションのいずれかを選択します。
    - [新しい順]
    - [古い順]

クエリ履歴のフィルタリング

クエリ履歴とスター付きタブで、クエリ履歴を特定のデータソースに対してフィルタリングできます。

データソースに対して履歴をフィルタリングするには

1. [特定のデータソースのクエリをフィルタリング] フィールドを選択します。
2. 履歴をフィルタリングするデータソースを選択します。複数のデータソースを選択することもできます。

> **Note**
>
> 混合データソースを使用して実行されたクエリは、[混合] をフィルタリングする場合にのみ表示され、個々のデータソースでフィルタリングする場合は表示されません。

[クエリ履歴] タブでは、スライダーを使用して日付でクエリをフィルタリングすることもできます。

- 垂直スライダーを使用して、クエリを日付でフィルタリングします。
- 下部のハンドルをドラッグして開始日を設定します。
- 上部のハンドルをドラッグして終了日を設定します。

クエリ履歴の検索

クエリとコメントをまたいで履歴を検索できます。クエリ履歴タブとスター付きタブでクエリを検索できます。

クエリ履歴で検索するには

1. [クエリ検索] フィールドを選択します。
2. 検索する用語を検索フィールドに入力します。

クエリ履歴設定

[設定] タブでクエリ履歴をカスタマイズできます。次の表にオプションの説明を示します。

設定	デフォルト値
デフォルトのアクティブタブを変更する	[クエリ履歴] タブ

> **Note**
>
> クエリ履歴設定は全体に及び、分割モードで両方のパネルに適用されます。

Explore のログ

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Explore を使用すると、次のようなさまざまなデータソースのログを調査できます。

- OpenSearch
- Amazon CloudWatch
- InfluxDB
- Loki

インフラストラクチャのモニタリングとインシデント対応中に、メトリクスとログをより深く掘り下げて原因を見つけることができます。Explore では、ログを並行して表示することで、メトリクス、トレース、プロファイルなどの他のテレメトリシグナルとログを関連付けることもできます。

ログクエリの結果は、個々のログ行として表示され、選択した期間のログボリュームを示すグラフとして表示されます。

ログボリューム

フルレンジのログボリュームをサポートするデータソースを操作すると、Explore には、入力されたすべてのログクエリのログ分布を示すグラフが自動的に表示されます。この機能は、現在 OpenSearch および Loki データソースでサポートされています。

> **Note**
>
> Loki では、このフルレンジのログボリュームをメトリクスクエリによってレンダリングします。このクエリは、クエリされた時間範囲に応じてコストがかかる可能性があります。このクエリは、小規模な Loki インストールで処理するのが特に難しい場合があります。これを軽減するには、Loki の前に nginx などのプロキシを使用して、これらのクエリにおけるユーザー定義のタイムアウト (10 秒など) を設定できます。ログボリュームヒストグラムクエリは、値 `Source=logvolhist` の HTTP ヘッダー `X-Query-Tags` を持つクエリを検索することで識別できます。これらのヘッダーは、Grafana によってすべてのログボリュームヒストグラムクエリに追加されます。

データソースがフルレンジのログボリュームの読み込みに対応していない場合、ログモデルはログ行をカウントし、自動的に計算された時間間隔に基づいてバケットに整理することで時系列を計算します。最初のログ行のタイムスタンプは、結果のログボリュームの開始を固定するために使用されます。時系列の終了点は、タイムピッカーの To 範囲に固定されます。これにより、データソースがフルレンジのサポートを提供していない場合でも、ログデータを効率的に分析および視覚化できます。

ログ

以下のセクションでは、Explore で個々のログを視覚化して操作する詳細な方法を説明します。

ログナビゲーション

ログ行の右側にあるログナビゲーションを使用して、追加のログを簡単にリクエストできます。これを行うには、ナビゲーションの下部にある [古いログ] ボタンをクリックします。これは、行の制限に達した際に、さらにログを表示する場合に特に便利です。ナビゲーションから実行される各リクエストは、ナビゲーションに個別のページとして表示されます。各ページには、受信ログ行のタイムスタンプ `from` と `to` が表示されます。各ページをクリックすると、以前の結果を確認できます。Explore がログナビゲーションから実行された過去 5 つのリクエストをキャッシュすることで、ページをクリックしたときに同じクエリを再実行しないようにでき、時間とリソースを節約できます。

視覚的表現オプション

ログの表示方法をカスタマイズしたり、表示する列を選択することができます。

オプション	説明
[時間]	時間列を表示または非表示にします。これはデータソースから報告されたログ行と関連するタイムスタンプです。
[一意のラベル]	非共通ラベルのみを含む一意のラベル列を表示または非表示にします。すべての一般的なラベルが上に表示されます。
[行の折り返し]	ディスプレイで行の折り返しを使用する場合は、これを `true` に設定します。`false` に設定すると、水平スクロールになります。
Prettify JSON (JSONの整形)	すべての JSON ログを pretty print (整形表示) するには、これを `true` に設定します。この設定は、JSON 以外の形式のログには影響しません。
Deduplication (重複排除)	ログデータは非常に反復的であり、Explore は重複するログ行を非表示にすることで役立ちます。使用できる重複排除アルゴリズムはいくつかあります。[完全] 一致は、日付フィールドを除き、行全体で行われます。[数値] の一致は、期間、IP アドレスなどの数値を除外した後に行で行われます。Signature (署名) は、残りの空白と句読点のすべての文字、数字、一致を削除するため、最も積極的な重複排除です。
Display results order (結果の順序の表示)	受信したログの順序は、デフォルトの降順 (新しい順) から昇順 (古い順) に変更できます。

ログ行のダウンロード

ログ結果を `txt` または `json` 形式でダウンロードするには、[Download] ボタンを使用します。この機能を使用すると、ログデータを保存して詳細な分析を行ったり、便利でアクセスしやすい形式で他のユーザーと共有したりできます。

ログ結果のメタ情報

受信したログ行の上には、次のような重要なメタ情報があります。

- [受信したログの数] – 現在のクエリまたは時間範囲に対して受信したログの合計数を示します。
- [エラー] – ログ結果に考えられるエラーを表示します。
- [共通ラベル] - 共通ラベルを表示します。
- Total bytes processed (処理された合計バイト数) – 処理されたログデータの累積サイズをバイト単位で表します。

> **Note**
>
> 特定のメタ情報の可用性はデータソースに依存する場合があり、その結果、特定のデータソースに対する詳細の一部しか表示されない場合があります。

改行のエスケープ

Explore は、改行 (`\n`、`\r`) やタブ (`\t`) など、ログ行で誤ってエスケープされたシーケンスを自動的に検出します。このようなシーケンスを検出すると、Explore は [改行のエスケープ] オプションを提供します。

Explore が検出した誤ってエスケープされたシーケンスを自動的に修正するには

1. [改行のエスケープ] を選択してシーケンスを置き換えます。
2. 置き換えを手動で確認し、その正確性を確認します。

Explore はこれらのシーケンスを置き換えます。そうすると、オプションは [改行のエスケープ] から [エスケープの削除] に変更されます。受信した入力に基づいて解析が正確ではない可能性があるため、変更を評価します。[エスケープの削除] を選択すると、置き換えを元に戻すことができます。

ログレベル

`level` ラベルが指定されているログについては、このラベルの値を使用してログレベルを決定し、それに応じて各ログ行の色を更新します。ログに指定されたレベルラベルがない場合、そのコンテンツが、サポートされている表現のいずれかと一致するかどうかを調べます (詳細については、次の表を参照してください)。ログレベルは常に最初の一致によって決まります。Grafana がログレベルフィールドを推測できない場合、不明なログレベルで視覚化されます。

> **Note**
>
> Loki データソースを使用し、`level` がログラインの一部である場合は、パーサー (JSON、logfmt、regex など) を使用して、レベル情報をレベル値の決定に使用されるレベルラベルに抽出できます。これにより、ヒストグラムはさまざまなログレベルを個別のバーとして表示できます。

サポートされているログレベルと、ログレベルの略記と表現のマッピング

ログレベル	色	サポートされている式
critical	紫色	emerg、fatal、alert、crit、critical
エラー	赤	err、eror、error
警告	黄色	warn、warning
情報	緑色	info、information、informational、notice
デバッグ	青	dbug、debug
トレース	水色	トレース
不明	灰色	*

検索された単語を強調表示する

クエリに検索する特定の単語または式が含まれている場合、Explore は可視性を高めるためにログ行でこれらを強調表示します。この強調表示機能により、ログ内の関連コンテンツを簡単に特定して注目できます。

> **Note**
>
> 検索語を強調表示する機能は、データソースによって異なります。一部のデータソースでは、検索語の強調表示が使用できない場合があります。

ログの詳細を表示する

Explore では、各ログ行には [ログの詳細] という拡張可能なセクションがあり、ログ行を選択して開くことができます。ログの詳細ビューには、ログ行にあるフィールドとリンクの形式で追加情報と探索オプションが用意されており、より堅牢なインタラクションと分析が可能になります。

[フィールド]

ログの詳細ビューでは、表示されたフィールドを次の 2 つの方法でフィルタリングできます。ポジティブフィルター (特定のフィールドに焦点を当てる) とネガティブフィルター (特定のフィールドを除外する) です。これらのフィルターは、ログ行を生成した対応するクエリを更新し、それに応じて等式と不等式を追加します。データソースがサポートしている場合、Loki および OpenSearch の場合と同様に、ログの詳細によって、フィールドが現在のクエリ表示およびアクティブ状態 (ポジティブフィルターの場合のみ) に既に存在しているかどうかが確認されます。その際、クエリをオフにしたり、フィルター式をポジティブからネガティブに変更したりできます。

目のアイコンをクリックすると、完全なログ行ではなく、ログリストで視覚化するフィールドのサブセットを選択できます。各フィールドには、表示されているすべてのログに関連する統計を表示する統計アイコンがあります。

リンク

Grafana は、データリンクまたは相関機能を提供し、ログメッセージの任意の部分を内部または外部のリンクに変換できます。これらのリンクを使用して、関連するデータや外部リソースに移動し、シームレスで便利な方法で詳細情報を確認できます。

ログのコンテキスト

ログコンテキストは、特定の検索クエリに一致するログエントリを囲む追加のコンテキスト行を表示します。これはログエントリのコンテキストを理解するのに役立ちます。また、grep コマンドの -C パラメータに似ています。

テキスト行が長いため、各ログエントリ全体のコンテキストの読み取りと分析が困難になる場合があります。ここでは、[行の折り返し] の切り替えが役に立つ場合があります。この切り替えを有効にすると、Grafana は長いテキスト行を自動的に折り返して、ビューワーの可視幅に収まるようにします。これにより、ログエントリの読み取りと理解が容易になります。

[分割ビューで開く] ボタンを使用すると、Explore ビューの分割画面でログエントリのコンテキストクエリを実行できます。このボタンを選択すると、ログエントリとともにコンテキストクエリが表示される新しい Explore ペインが開き、周囲のコンテキストの分析と理解が容易になります。

ログコンテキストクエリは、 Ctrl (または Cmd) キーを押しながらボタンを選択してコンテキストモーダルを開くことで、新しいブラウザタブで開くこともできます。新しいタブで開くと、以前に選択したフィルターも適用されます。

ログ行のコピー

選択したログ行の内容を簡単にクリップボードにコピーするには、[Cログ行のコピー] ボタンを選択します。

リンクをログ行にコピーする

Grafana でログ行をリンクすると、特定のログエントリにすばやく移動して正確な分析を行うことができます。ログ行に対して [ショートリンクをコピー] ボタンを選択すると、絶対時間範囲内の正確なログエントリに直接アクセスできる短縮 URL を生成してコピーできます。リンクを開くと、Grafana は対応するログ行まで自動的にスクロールし、青色の背景でログ行が強調表示されるため、関連情報を簡単に特定して注目できます。

> **Note**
>
> これは、id フィールドを提供する Loki およびその他のデータソースでのみサポートされています。

ライブテーリング

サポートされているデータソースからのリアルタイムログを表示するには、Explore のライブテーリング機能を活用できます。

ライブテーリングを使用してログをリアルタイムで表示するには

1. Explore ツールバーの [Live] ボタンを選択して、Live Tail ビューに切り替えます。
2. Live Tail ビューでは、新しいログが画面の下部に表示され、背景が淡いコントラストになるため、最新情報を簡単に追跡できます。
3. ライブテーリングを一時停止し、中断することなく以前のログを調べる場合は、[一時停止] ボタンを選択するか、ログビューをスクロールするだけで確認できます。
4. ビューをクリアし、すべてのログを画面から削除するには、[ログのクリア] ボタンを選択します。このアクションによりログビューがリセットされ、白紙の状態となってログ分析を続行できます。
5. ライブテーリングを再開し、リアルタイムログの表示を続けるには、[再開] ボタンを選択します。
6. ライブテーリングを終了して標準の Explore ビューに戻る場合は、[停止] ボタンを選択します。

ライブテーリング機能を使用すると、最新のログが入ってきたときに注視できるため、リアルタイムイベントのモニタリングや問題の迅速な検出が容易になります。

[ログサンプル]

選択したデータソースがログサンプルを実装し、ログクエリとメトリクスクエリの両方をサポートしている場合、メトリクスクエリの場合は、視覚化されたメトリクスに寄与したログ行のサンプルを自動的に表示できます。この機能は、現在 Loki データソースでサポートされています。

メトリクスからログに切り替える

`DataSourceWithQueryExportSupport` を実装するメトリクスデータソース (Prometheus など) から `DataSourceWithQueryImportSupport` をサポートするログデータソース (Loki など) に移行する場合、ログに存在するクエリのラベルは保持され、それらを使用してログストリームに対してクエリを実行します。

例えば、Loki データソースに切り替えた後の次の Prometheus クエリ `grafana_alerting_active_alerts{job='grafana'}` は、`{job='grafana'}` に変更されま

す。これにより、パターン一致検索/テキスト検索できる選択した時間範囲内の口グのチャンクが返されます。

Explore でのトレース

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Explore を使用して、トレース中のデータソースからトレースを視覚化できます。

以下のデータソースがサポートされています。

- Tempo (サポートされている取り込み形式: OpenTelemetry 、Jaeger、Zipkin)
- Jaeger
- AWS X-Ray
- Zipkin

上記のデータソースのクエリを設定する方法については、特定のデータソースのドキュメントを参照してください。

クエリエディタ

データソースのクエリエディタを使用して、トレース中のデータをクエリおよび検索できます。

各データソースには、独自のクエリエディタを使用できます。Tempo データソースのクエリエディタは、Jaeger データソースのクエリエディタとは若干異なります。

各データソースのクエリについては、各ドキュメントを参照してください。

- Tempo
- Jaeger
- AWS X-Ray
- Zipkin

トレースビューの説明

このセクションでは、Trace View ダッシュボードの要素について説明します。

ヘッダー

トレースビューのヘッダーには、次の要素があります。

- ヘッダータイトル – ルートスパンの名前とトレース ID を表示します。
- 検索 – 検索されたテキストを含むスパンを強調表示します。
- メタデータ – トレースに関するさまざまなメタデータを表示します。

ミニマップ

要約ビューまたはトレースタイムラインを表示します。ポインタをミニマップにドラッグして、より短い時間範囲にズームインできます。ズームはメインタイムラインも更新するため、短いスパンを簡単に確認できます。ミニマップにカーソルを合わせ、ズームすると、ズームをリセットする [選択のリセット] ボタンが表示されます。

スパンフィルター

スパンフィルターを使用すると、トレースタイムラインビューワーでスパンをフィルタリングできます。追加するフィルターが多いほど、フィルターされたスパンがより具体的になります。

次の 1 つ以上のフィルターを追加できます。

- リソースサービス名
- スパン名
- 期間
- タグ (タグ、プロセスタグ、ログフィールドを含む)

一致したスパンのみを表示するには、[Show matches only] (一致のみを表示) トグルを選択します。

タイムライン

トレース内のスパンのリストを表示します。各スパン行は、次の要素で構成されます。

- [Expand children] (子を展開) ボタン – 選択したスパンのすべての子スパンを展開または折りたたみます。
- サービス名 – スパンをログに記録したサービスの名前。

- オペレーション名 – このスパンが表すオペレーションの名前。
- スパン期間バー – トレース内のオペレーション期間の視覚的表現。

スパンの詳細

スパン行を選択すると、以下を含むスパンの詳細が表示されます。

- オペレーション名
- スパンメタデータ
- タグ – このスパンに関連付けられている任意のタグ。
- プロセスメタデータ – このスパンをログに記録したプロセスに関するメタデータ。
- ログ – このスパンおよび関連するキー値によってログに記録されたログのリスト。Zipkin ログの場合、セクションには Zipkin 注釈が表示されます。

ログへのトレース

トレースビューのスパンから、そのスパンに関連するログに直接移動できます。これは、Tempo、Jaeger、および Zipkin データソースで使用できます。各データソースの設定方法については、関連するドキュメントを参照してください。

ドキュメントアイコンを選択して、設定されたデータソースを使用して Explore で分割ビューを開き、スパンに関連するログに対してクエリを実行します。

メトリクスへのトレース

> **Note**
>
> 現在、この機能はベータモードです。

トレースビューのスパンから、そのスパンに関連するメトリクスに直接移動できます。この機能は、Tempo、Jaeger、および Zipkin データソースで使用できます。設定の詳細については、関連するドキュメントを参照してください。

プロファイルへのトレース

[Trace to profiles] (プロファイルへのトレース) を使用すると、トレースとプロファイルをリンクする機能を追加することで、さまざまなシグナルを関連付けられる Grafana の機能を利用できます。

ノードグラフ

オプションで、表示されたトレースのノードグラフを展開できます。データソースによっては、トレースのスパンをグラフ内のノードとして表示したり、現在のトレースに基づくサービスグラフなどのコンテキストを追加したりできます。

サービスグラフビュー

Service Graph ビューは、スパンメトリクス (レート、エラーレート、期間 (RED) のデータを追跡) とサービスグラフを視覚化します。要件が設定されると、この事前設定されたビューがすぐに利用できます。

詳細については、Tempo データソースのページを参照してください。サービスグラフビューページは、「Grafana Labs Tempo ドキュメント」でも確認できます。

Data API

この視覚化では、正しく表示するために、データソースから返されるデータの特定の形状が必要です。

データソースはデータフレームを返し、`frame.meta.preferredVisualisationType = 'trace'` を設定する必要があります。

データフレーム構造

必須フィールド

フィールド名	タイプ	説明
traceID	文字列	トレース全体の識別子。データフレームには 1 つのトレースのみが必要です。
spanID	文字列	現在のスパンの識別子。SpanID はトレースごとに一意である必要があります。
parentSpanID	文字列	トレースビューで子親関係を作成するための親スパンの SpanID。親のないルートスパ

フィールド名	タイプ	説明
		ンでは undefined になる場合があります。
serviceName	文字列	このスパンが属するサービスの名前。
serviceTags	TraceKeyValuePair[]	サービスに関連するタグのリスト。
startTime	数値	スパンの開始時刻をミリ秒エポック時間で表します。
duration	数値	スパンの期間 (ミリ秒)。

オプションのフィールド

フィールド名	タイプ	説明
ログ	TraceLog[]	現在のスパンに関連付けられたログのリスト。
タグ	TraceKeyValuePair[]	現在のスパンに関連付けられたタグのリスト。
warnings	string[]	現在のスパンに関連付けられた警告のリスト。
stackTraces	string[]	現在のスパンに関連付けられたスタックトレースのリスト。
errorIconColor	文字列	スパンに error: true がタグ付けされている場合のエラーアイコンの色。

タイプの詳細については、GitHub の TraceSpanRow、TraceKeyValuePair、および TraceLog を参照してください。

Explore の相関エディタ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

相関関係により、ユーザーは 2 つのデータソース間のリンクを構築できます。相関関係の概要などの詳細については、「Grafana バージョン 10 の相関関係」を参照してください。

相関関係の作成

相関関係は、Explore ページから作成できます。

相関を作成するには

1. Amazon Managed Grafana ワークスペースで、Explore ページに移動します。
2. 新しい相関のソースにするデータソースを選択します。
3. サポートされている視覚化でデータを生成するクエリを実行します。

   > ⓘ Note
   > サポートされている視覚化は、ログとテーブルです。

4. 上部のツールバーで [+ 追加] を選択し、[相関の追加] を選択します ([コマンドパレット] から [相関エディタ] を選択することもできます)。

   Explore が相関エディタモードになりました。このモードは青い境界線と上部のバーで示されます。上部のバーで [Exit] (終了) を選択すると、相関エディタを終了できます。
5. 新しいクエリの構築に使用できるデータのリンクを使用して、視覚化に次の新しい相関関係を作成できるようになりました。

- [ログ] – 各ログ行におけるログ詳細内のフィールド値の横にリンクが表示されます。
- [テーブル] – テーブルセルはすべてリンクです。

6. リンクを選択して、新しい相関関係を追加します。リンクは、相関の結果フィールドとして使用されるフィールドに関連付けられます。詳細については、「相関関係の設定」を参照してください。
7. 開いた分割ビューで、右側のペインを使用して相関のターゲットクエリソースを設定します。詳細については、「ターゲットクエリ」を参照してください。
8. ペイン上部のリストにある変数を有する変数構文を使用して、ターゲットクエリを構築します。リストには、選択したデータ行のサンプル値が含まれます。
9. ラベルと説明を入力します (任意)。ラベルは、視覚化内のリンク名として使用され、変数を含めることができます。
10. 変換を指定します (任意。詳細については、以下を参照してください)。
11. 上部のツールバーで [保存] を選択して相関を保存し、相関エディタモードを終了します。相関関係の作成に使用されるリンクは、各行のデータリンクに置き換えられます。リンクを選択すると、定義したクエリが別のペインで実行され、変数は選択した行の値で動的に置き換えられます。

変換

変換を使用すると、他のデータを含むフィールドにある値を抽出できます。例えば、変換を使用すると、ログ行の一部を抽出して相関関係に使用できます。相関関係の変換の詳細については、「相関関係の変換」を参照してください。

エディタモードで生成されたリンクの 1 つを選択したのち、[変換] ドロップダウンメニューで [変換の追加] を選択すると、変換を追加できます。

相関関係で変換を使用するには

1. 変換を適用するフィールドを選択します。変換に使用するフィールドの一部を選択します。例えば、ログ行です。選択すると、このフィールドの値が変換の構築に使用されます。
2. 変換のタイプを選択します。オプションと関連する設定については、「相関関係の変換」を参照してください。
3. 選択に基づいて、1 つ以上の変数の入力や、表示されるオプションでさらに仕様を指定する必要がある場合があります。
4. [相関に変換を追加] を選択して、指定した変数を使用可能な変数のリストに追加します。

> **Note**
>
> このダイアログボックスにある正規表現の場合、他のドキュメントで参照されている `mapValue` は、ここでは `Variable Name` と呼び出されます。Grafana は、フィールド値の式に一致する任意のテキストを強調表示します。正規表現キャプチャグループを使用して、抽出する一致部分を選択します。有効な正規表現を指定すると、 変数とその変数の値が `Variable Name` フィールドの下に表示されます。

相関の例

次の例は、Explore の相関エディタを使用して相関を作成する方法を示しています。これらの例に従う場合は、必ずテストデータソースを設定してください。

テキストとグラフの相関関係の作成

この例では、Explore の相関エディタを使用して相関を作成する方法を示しています。

相関関係を使用すると、1 つのクエリの結果を使用して、任意のデータソースで新しいクエリを実行できます。この例では、表形式データをレンダリングするクエリを実行します。データは、グラフ結果を生成する別のクエリを実行するために使用されます。

この例に従うには、必ずテストデータソースを設定してください。

テキストとグラフの相関関係を作成するには

1. Grafana で、Explore に移動します。
2. ページの左上にあるドロップダウンメニューからテストデータソースを選択します。
3. 右側のドロップダウンメニューで [+ 追加] を選択し、[相関の追加] を選択します。
4. Explore が相関エディタモードになり、青い境界線で示されます。
5. シナリオドロップダウンメニューから CSV ファイルシナリオを選択します。
6. ファイル population_by_state.csv を選択します。各セルは、新しい相関関係の作成を開始するためにクリックできるリンクです。
7. `State` 列内の任意のセルをクリックして、そのエントリにデータリンクが付属する新しい相関関係を作成します。例えば、`California` を選択します。
8. 分割ビューで、左側のペインで選択したのと同じデータソースを選択します。クエリエディタの上にあるヘルパーには、ターゲットクエリを使用できる利用可能な変数がすべて含まれています。変数には、選択した行のすべてのデータフィールド (テーブル列) が含まれます。

9. [シナリオ] メニューで、[CSV メトリクス値] を選択します。クエリエディタの `String Input` フィールドには、各年の母集団値を持つ変数 `${1980},${2000},${2020}` が表示されます。これにより、変数値を使用してグラフが生成されます。
10. クエリエディタの [Alias] (エイリアス) フィールドに、`${State}` と入力します。

    クエリを実行して、変数のサンプル値を使用してグラフが生成されることを確認します。
11. [保存] を選択して相関を保存し、相関エディタを終了します。

    相関が保存されると、Explore は左側のペインでクエリを再実行します。状態名をクリックすると、右側のクエリが CSV に挿入された行の値で再実行され、グラフが変更されます。クエリは、状態名をクリックするたびに更新された値で再実行されます。

同じステップを任意のデータソースに適用できます。相関関係を使用すると、選択したデータに基づいて動的クエリを実行するためのリンクを視覚化で作成できます。この例では、クエリによって返されるデータを使用して、同じデータソースを使用して異なる視覚化を生成する新しいクエリを構築しました。ただし、任意のデータソース間に相関関係を作成して、カスタム探索フローを作成することもできます。

ログとテーブルの相関関係の作成

この例では、相関関係を作成して、変換を使用してログ行と別のフィールドから値を抽出する方法を示します。

この例に従うには、必ずテストデータソースを設定してください。

ログとテーブルの相関関係を作成するには

1. Grafana で、Explore に移動します。
2. ページの左上にあるドロップダウンメニューからテストデータソースを選択します。
3. 右側のドロップダウンメニューで [+ 追加] を選択し、[相関の追加] を選択します。
4. Explore が相関エディタモードになり、青い境界線で示されます。
5. [Scenario] メニューで、[ログ] を選択します。
6. ログ行を展開して、相関リンクを確認します。`Correlate with hostname` を選択します。
7. Explore が分割ビューで開きます。左側のペインで選択したのと同じデータソースを選択します。クエリエディタの上にあるヘルパーには、ターゲットクエリを使用できる利用可能な変数がすべて含まれています。
8. 変換セクションを展開し、[変換の追加] をクリックします。

9. フィールドドロップダウンメニューでメッセージを選択します。ログ行がサンプルデータとして表示されます。
10. タイプで [Logfmt] を選択します。これにより、変数のリストが入力されます。
11. [相関に変換を追加] を選択します。
12. 再度 [変換の追加] を選択し、[フィールド]で[ホスト名]を選択します。
13. タイプで[正規表現] を選択します。
14. [式] に `-([0-9]\*)` と入力します。これにより、ダッシュの右側にある任意の数値が選択されます。
15. [変数名)] に `hostNumber` と入力します。これにより、変数のリストが入力されます。
16. [相関に変換を追加] を選択して、他の変数に追加します。
17. データソースエディタで [Scenario] (シナリオ) ドロップダウンメニューを開き、[CSV Content] (CSV コンテンツ) を選択します。
18. 下にあるテキストボックスに以下を入力し、相関関係を保存します。

```
time,msg,hostNumber,status
${time},${msg},${hostNumber},${status}
```

これにより、分割ビューが閉じ、左側のクエリが再実行されます。任意のログ行を展開して、相関ボタンを確認します。相関ボタンを選択すると、分割ビューが開き、`time` (フィールド)、`msg` (ログ行から logfmt で抽出)、`host number` (`hostname` から正規表現で抽出)、`status` (ログ行から logfmt で抽出) が表示されます。

Explore の Inspector

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

インスペクターは、クエリの理解とトラブルシューティングに役立ちます。未加工データを検査し、そのデータをカンマ区切り値 (CSV) ファイルにエクスポートし、ログ結果を TXT 形式でエクスポートし、クエリリクエストを表示できます。

Inspector UI

インスペクターには次のタブがあります。

- [統計タブ] – クエリにかかる時間および返される量を示します。
- [クエリタブ] – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。
- [JSON] タブ – データ JSON とデータフレーム構造 JSON を表示およびコピーできます。
- [データタブ] – クエリによって返される未加工データを表示します。
- [エラータブ] – エラーを表示します。クエリがエラーを返す場合にのみ表示されます。

Inspector タスク

Explore Inspector では、さまざまなタスクを実行できます。

Inspector を開く

検査するクエリを実行したら、[Inspector] ボタンを選択します。

画面の下部にインスペクターペインが開きます。

未加工クエリ結果の検査

未加工のクエリ結果、つまりクエリによって返されるデータをテーブルに表示できます。

[Inspector] タブで、[Data] タブをクリックします。

複数のクエリまたは複数のノードのクエリには、追加のオプションがあります。

- [データフレームを表示する]: 表示する結果セットデータを選択します。
- [時間で結合された系列]: すべてのクエリの未加工データを一度に表示し、列ごとに結果セットを 1 つ表示します。列見出しをクリックしてデータをソートできます。

未加工のクエリ結果を CSV としてダウンロード

Grafana は、ブラウザのデフォルトのダウンロード場所に CSV ファイルを生成します。お好みのビューワーで開くことができます。

1. [Inspector] タブで、上記の手順に従って未加工のクエリ結果を取得します。
2. エクスポートする未加工データが表示されるまで、クエリ設定を絞り込みます。
3. [CSV のダウンロード] を選択します。

Excel 用にフォーマットされた CSV ファイルをダウンロードするには、[データオプション] を展開して [Excel 用にダウンロード] トグルをオンにしてから、[CSV のダウンロード] オプションを選択します。

ログ結果を TXT としてダウンロード

[Inspector] タブで [ログのダウンロード] を選択すると、現在表示しているログの TXT ファイルを生成できます。

トレース結果のダウンロード

データソースタイプに基づいて、Grafana は、サポートされている形式 (Jaeger、Zipkin、または OTLP 形式) のいずれかでトレース結果の JSON ファイルを生成します。

1. [Inspector] を開きます。
2. ログクエリの結果を検査します。エクスポートする未加工ログが表示されるまで、結果を絞り込みます。
3. [ログのダウンロード を選択します。

クエリパフォーマンスの検査

[統計] タブには、クエリにかかる時間、送信したクエリの数、返された行数を示す統計が表示されます。この情報は、特に数値が予想外に高い場合や低い場合のクエリのトラブルシューティングに役立ちます。

統計が読み取り専用形式で表示されます。

[JSON モデルの表示]

データだけでなく、データフレーム JSON モデルも探索してエクスポートできます。

JSON モデルを表示するには

1. インスペクターパネルで、[JSON] タブをクリックします。
2. [ソースの選択] ドロップダウンから、以下のいずれかのオプションを選択します。

- [データ] – Explore に返されたデータを表す JSON オブジェクトを表示します。
- [データフレーム構造] – 未加工の結果セットを表示します。

3. JSON の一部を展開または折りたたんで、個別のセクションを表示できます。[クリップボードにコピー] オプションを選択して、JSON 本文をコピーし、別のアプリケーションに貼り付けることもできます。

データソースへの未加工リクエストとレスポンスの表示

Explore と Inspector タブを操作すると、クエリで生成する未加工のリクエストとレスポンスデータを表示できます。Inspector で [クエリ] タブを選択し、[更新] を選択して未加工データを表示します。

Grafana はクエリをサーバーに送信し、結果を表示します。クエリの特定の部分をドリルダウン、そのすべてを展開または折りたたみ、またはデータをクリップボードにコピーして他のアプリケーションで使用できます。

Grafana バージョン 10 の相関関係

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

相関を設定することで、Explore ビジュアライゼーションのインタラクティブリンクを作成して、提示されたデータに関連するクエリを実行できます。

相関は、あるデータソース内のデータを使用して別のデータソース内のデータをクエリする方法を定義します。例:

- ログデータソースで返されるアプリケーション名を使用して、メトリクスデータソース内のそのアプリケーションに関連するメトリクスをクエリできます。
- SQL データソースから返されるユーザー名を使用して、ログデータソース内の特定のユーザーに関連するログをクエリできます。

Explore では、ユーザー定義の相関関係を使用して、視覚化内のリンクを表示します。リンクをクリックすると、関連するクエリが実行され、Explore Split View で結果を確認できます。

相関関係に基づくリンクの表示を現在サポートしている視覚化について説明します。

- ログ
- [テーブル]

相関関係は、Grafana で [管理] > [プラグインとデータ] > [相関] ページを使用するか、[Explore] で直接設定できます。

トピック

- 相関関係の設定
- 新しい相関関係を作成する

相関関係の設定

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

各相関関係は、次のオプションを使用して設定できます。

ラベル

視覚化に表示されるリンクラベル。

説明

オプションの説明。

ソースデータソース

リンクが表示される結果のソース。

結果フィールド

視覚化でリンクが表示される場所を定義します。

ターゲットクエリ

ターゲットクエリは、リンクがクリックされたときに実行されます。

[変換]

ターゲットクエリに渡される、含まれるソースデータへのオプションの操作。

相関関係の作成の詳細については、「相関関係の作成」を参照してください。

ソースデータソースと結果フィールド

リンクは、相関関係のソースデータソースからの結果に関して、Explore の視覚化に表示されます。リンクは、相関設定で提供された結果 (結果フィールド) のフィールドの 1 つに割り当てられます。各視覚化では、異なる方法でリンクを含むフィールドが表示されます。

ターゲットクエリ

ターゲットクエリは、視覚化でリンクがクリックされたときに実行されます。選択したターゲットデータソースのクエリエディタを使用して、ターゲットクエリを指定できます。ソースデータの結果は、変数を使用してターゲットクエリ内でアクセスできます。

相関変数

ターゲットクエリ内の変数を使用して、クエリに関連するソースデータにアクセスできます。相関関係は Grafana 変数構文を使用します。リンクが選択されると、変数にソース結果の値が入力されます。使用できる変数には 2 種類あります。

- フィールド変数 (フィールド値とラベルへのアクセスのため) 。
- 相関変数 (フィールド値と変換へのアクセスのため) 。

例: ソース結果に `employee` というフィールドが含まれている場合、フィールドの値には以下を使用してアクセスできます。

- フィールド変数 `${__data.fields.employee}`。
- 上記のフィールド値を `${employee}` にマッピングする相関変数。

フィールド値を短い変数名にマッピングするだけでなく、既存のフィールドに変換を適用することで、より多くの相関変数を作成できます。

相関関係は、すべての変数が選択したデータ行に値を持つ場合にのみデータリンクを作成します。[グローバル変数]はこのルールの例外であり、返されたデータから入力する必要はありません。これらの変数は、データソースによって自動的に補間されます。

相関関係の変換

変換を使用すると、フィールド値からより多くの変数を抽出できます。変換の出力は、他の変数としてアクセスできる一連の新しい変数です。

変換には logfmt と正規表現の 2 種類があります。

各変換では、選択したフィールド値が入力として使用されます。変換の出力は、変換のタイプとオプションに基づく一連の新しい変数です。

Logfmt 変換

logfmt 変換は、[logfmt キーと値のペア] でフォーマットされたテキストを含むフィールド値を分解します。各ペアは変数になり、キーは変数の名前になります。

logfmt 変換では、結果フィールドとは異なるフィールドに変換を適用する場合にのみ、入力フィールド名を指定する必要があります。`field = "host=srv001 endpoint=/test app=foo"` の出力変数の例:

名前	値
ホスト	srv001
エンドポイント	/test
アプリケーション	foo

正規表現変換

正規表現変換は、指定された正規表現に基づいてフィールド値を分解します。

正規表現変換オプション：

field

入力フィールド名

expression

正規表現 名前付きのキャプチャグループは、グループ名に一致する変数にマッピングされます。名前のない一致グループが使用されている場合、最初の一致から変数が作成されます。この値は、入力フィールドに一致する変数を上書きするか、mapValue が指定されている場合は新しい変数が作成されます (次の表の例を参照) 。

mapValue

一致する名前付きのグループのない単純な正規表現グループで使用されます。デフォルトでは、最初の一致は、入力として使用されるフィールドの名前で変数をオーバーライドします。デフォルトの動作を変更するには、mapValue プロパティを指定します。指定された名前は、新しい変数の作成に使用されます。これは、対象のクエリで、正確な値と変換で抽出された値の一部の両方が必要な場合に役立ちます。

例: 選択したフィールド名が employeeで、フィールド値が John Doe であると仮定します。

式および mapValue オプションに基づくさまざまな出力変数 :

expression	mapValue	出力変数	コメント
/\w+ (\w+)/	-	employee=Doe	mapValue が指定されていません。最初の一致は、既存のフィールド名変数 (employee) にマッピングされます。
/(\w+) (\w+)/	名前	name=John	最初のマッチングは、name という新しい変数にマッピングされます。
/(?\w+) (?\w+)/	-	firstName=John , lastName=Doe	名前付きグループを使用すると、出力変数の名前になり、m

expression	mapValue	出力変数	コメント
			apValue は無視されます。
`/(?\w+) (?\w+)/`	名前	`firstName=John , lastName=Doe`	上記と同じ

新しい相関関係を作成する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

相関関係は、Explore の相関関係エディタで作成することも、Amazon Managed Grafana ワークスペースの Grafana [管理]ページを使用して作成することもできます。

前提条件

新しい相関関係を追加するアクセス許可が必要です。データソースへの書き込み権限を持つユーザーのみが、新しい相関関係を定義できます。

Explore の相関関係エディタでの相関関係の作成

相関関係は、Explore 相関関係エディタで作成できます。詳細については、「相関関係の作成」を参照してください。

管理ページでの相関関係の作成

Grafana コンソールの[管理]ページを使用して相関関係を作成できます。

管理ページで相関関係を作成するには

1. Grafana の[管理]セクションに移動します。

2. [プラグインとデータ] で、[相関関係] を開きます。
3. 右上隅の [追加] ボタンを選択します。
4. 相関のラベルを指定します。
5. (オプション) [説明]を入力します。
6. 次のページに移動します。
7. ターゲットデータソースを指定します。
8. 変数を使用してターゲットクエリを指定します。
9. 次のページに移動します。
10. ソースデータソースを指定します。
11. 結果フィールドを指定します。
12. ソースデータソースのフィールドではない変数が必要な場合は、変換を追加します。
13. [追加] を選択して新しい変換を追加します。
14. 変換のタイプを選択します。
15. 選択したタイプに応じて変換を設定します。
16. 相関関係を保存します。

相関関係は同じ方法で編集できますが、編集時に選択したデータソースを変更することはできません。

Grafana バージョン 10 のアラート

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana バージョン 10 において、Amazon Managed Grafana には最新のアラートシステム [Grafana アラート]が含まれており、アラート情報を一元的に管理、検索できるビューが用意されていま

す。Grafana アラートは Grafana バージョン 8 のオプション機能として導入され、GrafanaLabs はバージョン 11 でレガシーアラートの削除を発表しました。

> **Note**
>
> このドキュメントでは、Grafana アラートについて説明します。レガシーアラートの詳細については、「従来のダッシュボードアラート」を参照してください。

Grafana アラートを使用すると、システムの問題が発生した直後にその問題を把握できます。

受信メトリクスデータまたはログエントリをモニタリングし、アラートシステムをセットアップして特定のイベントや状況を監視し、問題が見つかったときに通知を送信します。

これにより、手動モニタリングが不要になり、重大なインシデントにつながる可能性のあるシステム停止や変更に対する防御策を講じることができます。

Grafana アラートを使用すると、データがどこに保存されていても、複数のデータソースからクエリと式を作成できます。これにより、データを柔軟に結合し、メトリクスとログを新しく独自の方法でアラートを送信できます。それに続いて、単一の統合されたビューからアラートを作成、管理、およびアクションを実行し、チームが問題をすばやく特定して解決する能力を向上させることができます。

Mimir と Loki のアラートルールを使用すると、データ周囲で大規模なアラート式を実行できます。これらはすべて、お使いの Grafana UI によって管理されます。

> **Note**
>
> レガシー Grafana アラートを使用した以前のバージョンの Grafana から移行する場合は、レガシーアラートと新しい Grafana アラートの違いを確認すると役立つ場合があります。

主な特徴と利点

すべてのアラートを単一のページに

Grafana が管理するアラートと、Prometheus 互換データソースに存在するアラートの両方が 1 つの Grafana アラートページに統合されます。

多次元アラート

アラートルールは、「多次元アラート」と呼ばれるアラートルールごとに複数の個別のアラートインスタンスを作成できるため、1 つのアラートルールだけでシステム全体の可視性を得るための機能と柔軟性が得られます。多次元ルールを作成するには、クエリにラベルを追加して、モニタリング対象のコンポーネントを指定し、単一のアラートルールに対して複数のアラートインスタンスを生成します。例えば、クラスター内の各サーバーをモニタリングする場合、各 CPU で多次元アラートがアラートされ、サーバー全体で標準アラートがアラートされます。

ルートアラート

定義したラベルに基づいて、各アラートインスタンスを特定のコンタクトポイントにルーティングします。通知ポリシーは、アラートがどこで、いつ、どのようにコンタクトポイントにルーティングされるかを定義するルールのセットです。

サイレンスアラート

サイレンスは、通知の作成を停止し、指定された期間だけ継続します。サイレンスを使用すると、1 つ以上のアラートルールからの永続通知の受信を停止できます。また、特定の基準に基づいてアラートを部分的に一時停止することもできます。サイレンスには、整理と可視性を向上させるための独自の専用セクションがあるため、メインアラートビューを乱すことなく一時停止したアラートルールをスキャンできます。

ミュートタイミング

ミュートタイミングとは、ポリシーの新しい通知が生成されたり送信されないようにする繰り返しの時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間や繰り返し発生するのを防ぐために使用されます。

サイレンスと似ていますが、ミュートタイミングはアラートルールの評価や、ユーザーインターフェースへのアラート表示を停止せず、通知の作成のみを防ぎます。

アラートシステムを設計する

複雑な IT システムを監視し、すべてが正しく動作しているかどうかを理解することは難しい作業です。したがって、効果的なアラート管理システムを設定して、問題が発生したときにビジネスの成果に影響を与える前に通知することが不可欠です。

機能するアラート管理設定の設計と構成には時間がかかります。

ここでは、ビジネスに効果的なアラート管理の設定を作成する方法に関するヒントをいくつか紹介します。

監視およびアラートを行うビジネスの主要なメトリクスはどれですか？

- 受信者が無視するほど些細で頻繁ではない、知っておくべき重要なイベントを見つけます。
- アラートは、即時の対応や介入を必要とする大きなイベントに対してのみ作成すべきです。
- 量よりも質を考慮してください。

どのタイプのアラートを使用しますか？

- Grafana が管理するアラート、Grafana Mimir または Loki が管理するアラート、またはその両方を選択します。

アラートと通知をどのように整理しますか？

- アラートを受信するように設定するユーザーを選択します。対応中のユーザーまたは特定の Slack チャネルに送信することを検討してください。
- アラート API またはコードとしてのアラート (Terraform) として使用して、可能な限り自動化します。

アラート疲れを減らすにはどうすればよいですか？

- サイレンス、ミュートタイミング、アラートルール評価の一時停止を使用して、ノイズの多い不要なアラートを回避します。
- アラートルールを継続的に調整して、有効性を確認します。アラートルールを削除して、重複や無効なアラートを防ぎます。
- 優先度と重要度レベルを慎重に検討してください。
- しきい値と評価ルールを継続的に確認します。

Grafana アラートの制約事項

- 他のシステムからルールを集約する場合、Grafana アラートシステムは、利用可能なすべての Amazon Managed Service for Prometheus、Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからはルールを取得できない場合があります。

トピック

- 概要
- アラートの設定
- アラートを設定する
- アラートの管理

概要

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

始めたばかりのユーザーでも、Grafana アラートの経験豊富なユーザーでも、アラートの作成、管理、対応に役立つ基本と利用可能な機能の詳細を確認することで、問題を迅速に解決するチームの能力を向上させることができます。

原則

Prometheus ベースのアラートシステムでは、アラートを作成するアラートジェネレーターと、アラートを受信するアラートレシーバーがあります。たとえば、Prometheusはアラートジェネレーターであり、アラートルールの評価を担当します。アラートマネージャーはアラートレシーバーであり、発射アラートと解決済みアラートに関する通知のグループ化、禁止、サイレンシング、送信を担当します。

Grafana アラートは、アラートシステムを設計する Prometheus モデルに基づいて設計されています。アラートルールのスケジュールと評価を担当する内部アラートジェネレーターと、通知のグループ化、禁止、サイレンシング、送信を担当する内部アラートレシーバーがあります。Grafana は Prometheus をアラートジェネレーターとして使用しません。これは、Grafana アラートが Prometheus に加えて他の多くのデータソースと連携する必要があるためです。ただし、アラートマネージャーをアラートレシーバーとして使用します。

アラートはアラートレシーバーに送信され、ルーティング、グループ化、禁止、サイレンシング、通知されます。Grafana アラートでは、デフォルトのアラートレシーバーは Grafana 内に埋め込まれ

たアラートマネージャーであり、Grafana アラートマネージャーと呼ばれます。ただし、他のアラートマネージャーを使用することもできます。これらは外部アラートマネージャーと呼ばれます。

基礎

以下に、Grafana アラートのさまざまな部分の概要を示します。

アラートルール

アラートルールは、アラートがいつ発生するかを決定する一連の基準です。これは、1つ以上のクエリと式、満たす必要がある条件、アラートルールを評価する頻度を決定する間隔、およびアラートを発射するために条件を満たす必要がある期間で構成されます。

アラートルールはその間隔で評価され、各アラートルールは一度に0、1、または任意の数のアラートを発射できます。アラートルールの状態は、通常、保留中、または発射のいずれかの `severe` アラートによって決まります。例えば、アラートルールのアラートの少なくとも 1 つが発射されている場合、アラートルールも発射されます。アラートルールの状態は、最新の評価のステータスによって決まります。これらは、OK、Error、NoData のいずれかです。

アラートルールの非常に重要な特徴は、カスタム注釈とラベルをサポートすることです。これにより、概要や説明などの追加のメタデータを使用してアラートを計測し、特定の通知ポリシーにアラートをルーティングするためのラベルを追加できます。

アラート

アラートは、ラベルと呼ばれるキーと値のペアのセットによって一意に識別されます。各キーはラベル名、各値はラベル値です。例えば、あるアラートにはラベル `foo=bar` があり、別のアラートにはラベル `foo=baz` がある場合があります。アラートには `foo=bar,bar=baz` などの多くのラベルを含めることができますが、`foo=bar,foo=baz` などの同じラベルを 2 回持つことはできません。2 つのアラートに同じラベルを付けることもできず、2 つのアラートに `foo=bar,bar=baz` と `foo=bar,bar=baz` などの同じラベルがある場合、いずれかのアラートが破棄されます。アラートは、アラートルールの条件が満たされなくなるか、アラートルールが削除されると解決されます。

Grafana マネージドアラートでは、アラートは、正常、保留中、アラート中、データなし、エラー状態のいずれかになります。Mimir や Loki などのデータソースマネージドアラートでは、アラートは正常、保留中、アラートにすることができますが、NoData やエラーにすることはできません。

コンタクトポイント

コンタクトポイントは、通知の送信先を決定します。例えば、E メールアドレス、Slack、Grafana OnCall や Pagerduty などのインシデント管理システム (IRM)、またはウェブフックに通知を送信するコンタクトポイントがあるとします。

コンタクトポイントから送信される通知は、通知テンプレートを使用してカスタマイズできます。通知テンプレートを使用して、通知の件名、メッセージ、構造を変更できます。通知テンプレートは、個々の統合やコンタクトポイントに固有のものではありません。

通知ポリシー

通知ポリシーはアラートをグループ化し、コンタクトポイントにルーティングします。通知が送信されるタイミングと、通知を繰り返す頻度を決定します。

アラートは、ラベルマッチャーを使用して通知ポリシーとマッチングされます。これらは、アラートのラベルが完全に一致するかどうか、完全に一致しないかどうか、予期されるテキストが含まれているかどうか、または含まれていないかどうかを示す、人間が判読できる式です。例えば、マッチャー `foo=bar` はアラートをラベル `foo=bar` に一致させ、マッチャー `foo=~[a-zA-Z]+` は foo という名前のラベルを正規表現 `[a-zA-Z]+` に一致する値に一致させます。

デフォルトでは、アラートは 1 つの通知ポリシーにのみ一致できます。ただし、 `continue` 機能を使用すると、任意の数の通知ポリシーを同時に一致させるようにアラートを作成できます。通知ポリシーの詳細については、「通知ポリシー」を参照してください。

サイレンスとミュートタイミング

サイレンスとミュートタイミングを使用すると、特定のアラートまたは通知ポリシー全体の通知を一時停止できます。アラートの修正作業中など、サイレンスを使用してアドホックベースで通知を一時停止し、定期的なメンテナンスウィンドウ中など、ミュートタイミングを使用して定期的に通知を一時停止することができます。

トピック

- データソースと Grafana アラート
- 数値データのアラート
- ラベルと注釈
- アラートルールについて
- アラートマネージャー
- コンタクトポイント
- 通知
- 高可用性のアラート

データソースと Grafana アラート

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana アラートと互換性のあるデータソースは多数あります。各データソースはプラグインでサポートされています。Grafana アラートでは、評価エンジンがバックエンドで実行されるため、データソースを使用してルールを評価するために、データソースプラグインをバックエンドプラグインにする必要があります。プラグインは、Grafana アラートと互換性があることも指定する必要があります。

データソースは時間の経過とともに追加および更新されます。以下のデータソースは、Grafana アラートと互換性があるものをご紹介しています。

- Amazon CloudWatch データソースへの接続
- Azure Monitor データソースへの接続
- Amazon OpenSearch Service データソースへの接続
- Google Cloud Monitoring データソースへの接続
- Graphite データソースに接続する
- InfluxDB ソースデータベースに接続するには
- Loki データソースに接続するには
- Microsoft SQL Server データソースに接続する
- MySQL データソースに接続する
- OpenTSDB データソースに接続する
- PostgreSQL データソースに接続する
- Amazon Managed Service for Prometheus およびオープンソースの Prometheus データソースへの接続
- Jaeger データソースへの接続

- Zipkin データソースへの接続
- Tempo データソースへの接続
- テスト用 TestData データソースの設定

Amazon Managed Grafana のデータソースとデータソースプラグインの詳細については、「データソースに接続する」を参照してください。

数値データのアラート

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、Grafana が時系列データではなく数値データのアラートを処理する方法について説明します。

特定のデータソースでは、時系列ではない数値データをサーバーサイド式 (SSE) で直接アラートしたり、渡したりすることができます。これにより、データソース内の処理と効率が向上し、アラートルールを簡素化することができます。時系列データではなく数値データを基にアラートを送信する場合、ラベル付きの各時系列を 1 つの数値に集約する必要はありません。代わりに、ラベル付き番号が Grafana に返されます。

表形式のデータ

この機能は、表形式データをクエリするバックエンドデータソースでサポートされています。

- MySQL 、Postgres、MSSQL、Oracle などの SQL データソース。
- Azure Kusto ベースのサービス: Azure Monitor (Logs)、Azure Monitor (Azure Resource Graph)、および Azure Data Explorer。

Grafana 管理のアラートまたは SSE を含むクエリは、以下の場合、これらのデータソースでは数値として扱われます。

- データソースクエリの「Format AS」オプションが「テーブル」に設定されている。
- クエリから Grafana に返されるテーブルレスポンスには、1 つの数値 (int、double、float など) 列と、オプションで追加の文字列列のみが含まれている。

文字列がある場合、それらの列はラベルになります。列の名前がラベル名になり、各行の値が対応するラベルの値になります。複数の行が返された場合、各行はラベルによって一意に識別される必要があります。

例

「DiskSpace」という名前の MySQL テーブルの場合:

時間	ホスト	ディスク	PercentFree
2021 年 6 月 7 日	web1	/etc	3
2021 年 6 月 7 日	web2	/var	4
2021 年 6 月 7 日	web3	/var	8
...	...	...	...

データを時間でフィルタリングしてクエリを実行できますが、時間系列は Grafana に返す必要はありません。例えば、空き容量が 5% 未満の場合にホスト、ディスクごとにトリガーされるアラートです。

```
SELECT Host , Disk , CASE WHEN PercentFree  < 5.0 THEN PercentFree  ELSE 0 END FROM (
       SELECT
          Host,
          Disk,
          Avg(PercentFree)
       FROM DiskSpace
       Group By
          Host,
          Disk
       Where __timeFilter(Time)
```

このクエリでは、次のようなレスポンステーブルが Grafana に返されます。

ホスト	ディスク	PercentFree
web1	/etc	3
web2	/var	4
web3	/var	0

このクエリがアラートルールの条件として使用されると、ゼロ以外のクエリはアラートになります。その結果、次の 3 つのアラートインスタンスが生成されます。

ラベル	ステータス
{Host=web1,disk=/etc}	[アラート]
{Host=web2,disk=/var}	[アラート]
{Host=web3,disk=/var}	普通

ラベルと注釈

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ラベルと注釈には、アラートに関する情報が含まれています。ラベルと注釈はどちらも同じ構造であり、名前付き値のセットですが、用途は異なります。ラベルまたは同等の注釈の例は、`alertname="test"` である場合があります。

ラベルと注釈の主な違いは、ラベルはアラートを他のすべてのアラートと区別するために使用され、注釈は既存のアラートに追加情報を追加するために使用されるという点です。

例えば、2 つの高 CPU アラートを考えてみましょう。1 つは server1 用、もう 1 つは server2 用です。このような例では、最初のアラートに server="server1" というラベルがあり、2 番目のアラートに server="server2" というラベルがある、server というラベルがある場合があります。ただし、 server1 や 75% がサーバーの名前と CPU 使用率に置き換えられる "The CPU usage for server1 is above 75%." などの各アラートに説明を追加することもできます (これを行う方法については、ラベルと注釈のテンプレート作成 のドキュメントを参照してください)。このような説明は、注釈としてより適しています。

ラベル

ラベルには、アラートを識別する情報が含まれます。ラベルの例は server=server1 です。各アラートには複数のラベルを含めることができ、アラートのラベルの完全なセットはラベルセットと呼ばれます。アラートを識別するのはこのラベルセットです。

例えば、アラートにラベルセット {alertname="High CPU usage",server="server1"} が設定されているのに対し、別のアラートにラベルセット {alertname="High CPU usage",server="server2"} が設定されている場合があります。alertname ラベルは同じですが、server ラベルは異なるため、これらは 2 つの別々のアラートです。

アラートのラベルセットは、データソースのラベル、アラートルールのカスタムラベル、alertname などの予約済みラベルの組み合わせです。

カスタムラベル

カスタムラベルは、アラートルールからの追加ラベルです。注釈と同様に、カスタムラベルには名前が必要です。その値には、アラートが発射したときに評価されるテキストコードとテンプレートコードの組み合わせを含めることができます。カスタムラベルをテンプレート化する方法については、こちらを参照してください。

テンプレートでカスタムラベルを使用する場合、多数の個別のアラートが生成されるのを避けるため、ラベルの値がアラートルールの連続した評価間で変化しないようにすることが重要です。ただし、テンプレートが異なるアラートに対して異なるラベル値を生成しても問題ありません。例えば、値が変更されるたびに新しいアラートセットが作成されてしまうため、クエリの値をカスタムラベルに入れないでください。代わりに注釈を使用します。

また、アラートのラベルセットに同じ名前のラベルが 2 つ以上ないことを確認することも重要です。カスタムラベルの名前がデータソースのラベルと同じ場合は、そのラベルが置き換えられます。ただし、カスタムラベルの名前が予約済みラベルと同じ場合、カスタムラベルはアラートから省略されます。

注釈

注釈は、既存のアラートに追加情報を追加する名前付きペアです。Grafana には、description、summary、runbook_url、dashboardUId、panelId など、推奨される注釈が多数あります。カスタムラベルと同様に、注釈には名前が必要です。その値には、アラートが発射したときに評価されるテキストコードとテンプレートコードの組み合わせを含めることができます。注釈にテンプレートコードが含まれている場合は、アラートが発生したときにテンプレートが 1 回評価されます。アラートが解決されても、再評価されません。注釈をテンプレートする方法については、こちらを参照してください。

トピック

- ラベル一致の仕組み
- Grafana アラートのラベル
- ラベルと注釈のテンプレート作成

ラベル一致の仕組み

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ラベルとラベルマッチャーを使用して、アラートルールを通知ポリシーとサイレンスにリンクできます。これにより、アラートインスタンスを柔軟に管理し、処理するポリシーとサイレンスするアラートを指定できます。

ラベルマッチャーは、ラベル、値、演算子 の 3 つの異なる部分で構成されます。

- [ラベル]フィールドは、一致するラベルの名前です。ラベル名と完全に一致する必要があります。
- [値]フィールドは、指定されたラベル名の対応する値と一致します。一致方法は、演算子の値によって異なります。
- [演算子] フィールドは、ラベル値と一致させる演算子です。利用できる演算子は次のとおりです。

演算子	説明
=	値と完全に等しいラベルを選択します。
!=	値と等しくないラベルを選択します。
=~	値と正規表現が一致するラベルを選択します。
!~	値と正規表現が一致しないラベルを選択します。

複数のラベルマッチャーを使用している場合は、AND 論理演算子を使用して組み合わせられます。つまり、ルールをポリシーにリンクするには、すべてのマッチャーが一致する必要があります。

例

アラートに次のラベルセットを定義する場合：

```
{ foo=bar, baz=qux, id=12 }
```

次に：

- foo=bar として定義されたラベルマッチャーは、このアラートルールと一致します。
- foo!=bar として定義されたラベルマッチャーがこのアラートルールと一致しません。
- id=~[0-9]+ として定義されたラベルマッチャーは、このアラートルールと一致します。
- baz!~[0-9]+ として定義されたラベルマッチャーは、このアラートルールと一致します。
- foo=barとid=~[0-9]+として定義された 2 つのラベルマッチャーは、このアラートルールと一致します。

ラベルを除外する

また、ラベルを除外するためにラベルマッチャーを記述することもできます。

ラベルteamを除外する方法を示す例を次に示します。これらの値のいずれかを選択して、ラベルを除外できます。

- team=""

- `team!~.+`
- `team=~^$`

Grafana アラートのラベル

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、ラベルがアラートの基本コンポーネントである理由について説明します。

- アラートの完全なラベルセットは、Grafana アラート内のアラートを一意に識別するものです。
- アラートマネージャーはラベルを使用して、通知ポリシーのサイレンスとアラートグループのアラートを照合します。
- アラート UI には、そのルールの評価中に生成されたすべてのアラートインスタンスのラベルが表示されます。
- コンタクトポイントには、通知を生成する際にラベルにアクセスして、そのアラートに固有の情報を含めることができます。
- アラートルールにラベルを追加できます。ラベルは手動で設定でき、テンプレート関数を使用し、他のラベルを参照できます。アラートルールに追加されたラベルは、ラベル間で競合が発生した場合に優先されます (Grafana の予約済みラベルの場合を除きます。詳細については、以下を参照してください)。

外部アラートマネージャーの互換性

Grafana の組み込みアラートマネージャーは、Unicode ラベルキーと値の両方をサポートしています。外部 Prometheus アラートマネージャーを使用している場合、ラベルキーは[データモデル]と互換性がある必要があります。つまり、ラベルキーには、ASCII 文字、数字およびアンダースコアのみが含まれ、正規表現 `[a-zA-Z_][a-zA-Z0-9_]*` と一致する必要があります。無効な文字は、Grafanaアラートエンジンによって削除または置換されてから、次の規則に従って外部アラートマネージャーに送信されます。

- `Whitespace` は削除されます。
- `ASCII characters` は、_に置き換えられます。
- `All other characters` は、小文字の 16 進数表現に置き換えられます。これが最初の文字の場合、_ というプレフィックスがつきます。

> **Note**
>
> 複数のラベルキーが同じ値にサニタイズされている場合、重複にはサフィックスとして追加された元のラベルの短いハッシュが付加されます。

Grafana 予約済みラベル

> **Note**
>
> `grafana_` のプレフィックスがついたラベルは、Grafana によって特別な用途のために予約されています。`grafana_` で始まる手動で設定されたラベルが追加されると、競合時に上書きされます。

Grafana 予約済みラベルは、手動で設定されたラベルと同じ方法で使用できます。利用可能な予約済みラベルの現在のリストは次のとおりです。

ラベル	説明
grafana_folder	アラートを含むフォルダのタイトル。

ラベルと注釈のテンプレート作成

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

テンプレートを使用して、クエリや式からのデータをラベルや注釈に含めることができます。たとえば、クエリの値に基づいてアラートの重大度ラベルを設定したり、クエリのインスタンスラベルをサマリー注釈で使用して、CPU使用率が高いサーバーを特定したりできます。

すべてのテンプレートは、テキスト/テンプレート で記述する必要があります。ラベルと注釈のどちらをテンプレート化するかにかかわらず、テンプレート化するラベルまたは注釈内に各テンプレートをインラインで記述する必要があります。つまり、ラベルと注釈間でテンプレートを共有することはできません。代わりに、テンプレートを使用する場所を問わず、テンプレートをコピーする必要があります。

各テンプレートは、アラートルールが評価されるたびに評価され、アラートごとに個別に評価されます。例えば、アラートルールにテンプレート作成されたサマリー注釈があり、アラートルールに 10 個の起動アラートがある場合、テンプレートはアラートごとに 10 回実行されます。テンプレートで高価な計算をできるだけ行わないようにする必要があります。

例

次の例では、テキスト/テンプレートに関する完全なチュートリアルを記述するのではなく、テンプレートに関連する最も一般的なユースケースを示しています。これらの例を逐語的に使用したり、ユースケースに合わせて必要に応じて変更したりできます。テキスト/テンプレートの記述方法の詳細については、テキスト/テンプレートのドキュメントを参照してください。

すべてのラベルをカンマ区切りで印刷する

すべてのラベルをカンマ区切りで印刷するには、$labels 変数を出力します:

```
{{ $labels }}
```

例えば、ラベル alertname=High CPU usage、grafana_folder=CPU alerts、および instance=server1 を含むアラートがあると、以下が印刷されます:

```
alertname=High CPU usage, grafana_folder=CPU alerts, instance=server1
```

> **Note**
>
> クラシック条件を使用している場合、 $labels にはクエリからのラベルは含まれません。詳細については、$labels 変数を参照してください。

すべてのラベルを 1 行に 1 つずつ印刷する

すべてのラベルを 1 行に 1 つずつ印刷するには、 range を使用して各キーと値のペアを反復し、個別に印刷します。ここでは、$k の名前と現在のラベルの値 $v を参照します。

```
{{ range $k, $v := $labels -}}
{{ $k }}={{ $v }}
{{ end }}
```

例えば、ラベル alertname=High CPU usage、grafana_folder=CPU alerts、および instance=server1 を含むアラートがあると、以下が印刷されます:

```
alertname=High CPU usage
grafana_folder=CPU alerts
instance=server1
```

> **Note**
>
> クラシック条件を使用している場合、 $labels にはクエリからのラベルは含まれません。詳細については、$labels 変数を参照してください。

個々のラベルの印刷

個々のラベルを印刷するには、$labels 変数で index 関数を使用します。

```
The host {{ index $labels "instance" }} has exceeded 80% CPU usage for the last 5
 minutes
```

例えば、ラベル instance=server1 のアラートがあると、以下が印刷されます:

```
The host server1 has exceeded 80% CPU usage for the last 5 minutes
```

> **Note**
>
> クラシック条件を使用している場合、 $labels にはクエリからのラベルは含まれません。詳細については、$labels 変数を参照してください。

クエリの値を印刷する

インスタントクエリの値を印刷するには、index 関数と $values 変数を使用して Ref ID を印刷します。

```
{{ index $values "A" }}
```

例えば、値 81.2345 を返すインスタントクエリがあると、次の内容が印刷されます:

```
81.2345
```

範囲クエリの値を印刷するには、まず縮小式を使用して時系列からインスタントベクトルに値を変換する必要があります。その後、代わりに Ref ID を使用して、縮小式の結果を印刷できます。例えば、縮小式が A の平均を取得し、Ref ID B がある場合、次のように記述します:

```
{{ index $values "B" }}
```

クエリのヒューマナイズされた値を印刷する

インスタントクエリのヒューマナイズされた値を印刷するには、 humanize 関数を使用します:

```
{{ humanize (index $values "A").Value }}
```

例えば、値 81.2345 を返すインスタントクエリがあると、次の内容が印刷されます:

```
81.234
```

範囲クエリのヒューマナイズされた値を印刷するには、まず時系列から、縮小式を使用してインスタントベクトルに減らす必要があります。その後、代わりに Ref ID を使用して、縮小式の結果を印刷できます。例えば、縮小式が A の平均を取得し、Ref ID B がある場合、次のように記述します:

```
{{ humanize (index $values "B").Value }}
```

クエリの値をパーセンテージで印刷する

インスタントクエリの値をパーセンテージで印刷するには、 humanizePercentage 関数を使用します:

```
{{ humanizePercentage (index $values "A").Value }}
```

この関数は、値が 0 から 1 までの 10 進数であることを想定しています。代わりに、値が 0 から 100 までの 10 進数の場合、クエリまたは数式を使用して 100 で割ることができます。クエリが範囲クエリである場合は、まず、時系列から縮小式を使用してインスタントベクトルにクエリを減らす必要があります。

クエリの値から重要度を設定する

クエリの値から重要度ラベルを設定するには、if ステートメントと greater than 比較関数を使用します。テキスト/テンプレートは型強制をサポートしていないため、$values と比較するときは、必ず小数点 (80.0、50.0、0.0 など) を使用してください。サポートされているすべての比較関数のリストは、こちらで確認できます。

```
{{ if (gt $values.A.Value 80.0) -}}
high
{{ else if (gt $values.A.Value 50.0) -}}
medium
{{ else -}}
low
{{- end }}
```

クラシック条件からすべてのラベルを印刷する

従来の条件を使用している場合、 $labels を使用してクエリからラベルを印刷することはできません。代わりに $values を使用する必要があります。これは、クラシック条件がこれらのラベルを破棄して一次元動作を適用するためです (アラートルールごとに最大 1 つのアラート) 。クラシック条件でこれらのラベルが破棄されない場合、多くの時系列を返すクエリでは、アラートルールが評価されるたびにラベルが変更されるため、アラートの発射と解決の間でアラートが頻繁に発生します。

代わりに、 $values 変数には、発射するすべての条件のすべての時系列の減算値が含まれます。例えば、2 つの時系列を返すクエリ A と、2 つの条件を持つクラシック条件 B を持つアラートルールがある場合、 $values には、B0、B1、B2、B3 が含まれます。クラシック条件 B に 1 つの条件しかない場合、 $values には B0 と B1 のみが含まれます。

すべての発射時系列のすべてのラベルを印刷するには、次のテンプレートを使用します (正規表現の B を、異なる場合はクラシック条件の Ref ID に置き換えてください)。

```
{{ range $k, $v := $values -}}
{{ if (match "B[0-9]+" $k) -}}
{{ $k }}: {{ $v.Labels }}{{ end }}
{{ end }}
```

例えば、1 つの条件を超える 2 つの時系列のクラシック条件が出力されます。

```
B0: instance=server1
B1: instance=server2
```

クラシック条件に2つ以上の条件があり、時系列が同時に複数の条件を超える場合、超過した条件ごとにラベルが重複します。

```
B0: instance=server1
B1: instance=server2
B2: instance=server1
B3: instance=server2
```

一意のラベルを印刷する必要がある場合は、代わりにアラートルールを一次元から多次元に変更することを検討する必要があります。これを行うには、クラシック条件を縮小式と数式に置き換えます。

クラシック条件のすべての値を印刷する

クラシック条件のすべての値を印刷するには、前の例を取って $v.Labels を $v.Value に置き換えます。

```
{{ range $k, $v := $values -}}
{{ if (match "B[0-9]+" $k) -}}
{{ $k }}: {{ $v.Value }}{{ end }}
{{ end }}
```

例えば、1 つの条件を超える 2 つの時系列のクラシック条件が出力されます。

```
B0: 81.2345
B1: 84.5678
```

クラシック条件に 2 つ以上の条件があり、時系列が同時に複数の条件を超える場合、$values にはすべての条件の値が含まれます。

```
B0: 81.2345
B1: 92.3456
B2: 84.5678
B3: 95.6789
```

[変数]

ラベルと注釈をテンプレート化する場合、次の変数を使用できます:

ラベル変数

$labels 変数には、クエリのすべてのラベルが含まれます。たとえば、すべてのサーバーの CPU 使用率を返すクエリがあり、いずれかのサーバーの CPU 使用率が過去 5 分間に 80% を超えたときに起動するアラートルールがあるとします。どのサーバーで CPU 使用率が高いかを知らせるサマリー注釈をアラートに追加します。$labels 変数を使用すると、次のような判読可能な文を印刷するテンプレートを記述できます:

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes
```

> **Note**
>
> クラシック条件を使用している場合、 $labels にはクエリからのラベルは含まれません。クラシック条件は、一次元動作 (アラートルールごとに最大 1 つのアラート) を適用するために、これらのラベルを破棄します。テンプレートでクエリのラベルを使用する場合は、前の「クラシック条件からすべてのラベルを印刷する」の例に従います。

値変数

$value 変数は、アラートルールのすべてのインスタントクエリのラベルと値、しきい値、縮小式と数式、およびクラシック条件を含む文字列です。範囲クエリの結果は、10 から 10,000 行またはメトリクスのどれかを返す可能性があるため、含まれていません。そうした場合、特に大きなクエリでは、1 つのアラートで 10 MB のメモリが使用され、Grafana でメモリが非常に迅速に不足する可能性があります。

サマリーで $value 変数を印刷するには、次のような内容を作成します。

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes:
 {{ $value }}
```

次のように表示されます。

```
CPU usage for instance1 has exceeded 80% for the last 5 minutes: [ var='A'
 labels={instance=instance1} value=81.234 ]
```

ここで var='A' は、Ref ID A を使用したインスタントクエリを参照し、ラベル labels={instance=instance1} を参照し、過去 5 分間の平均 CPU 使用率 value=81.234 を参照します。

完全な文字列ではなく一部の文字列のみを印刷する場合は、 $values 変数を使用します。$value と同じ情報が含まれていますが、構造化されたテーブルの形態をとっており、必要なテキストのみに一致する正規表現を記述するよりもはるかに使いやすくなっています。

値変数

$values 変数は、すべてのインスタントクエリと式のラベルと浮動小数点値を含むテーブルで、Ref ID でインデックス化されます。

Ref ID A でインスタントクエリの値を印刷するには:

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes:
 {{ index $values "A" }}
```

例えば、ラベル instance=server1 付きのアラートと値 81.2345 のインスタントクエリがあるとすると、次の内容が出力されます。

```
CPU usage for instance1 has exceeded 80% for the last 5 minutes: 81.2345
```

Ref ID A のクエリがインスタントクエリではなく範囲クエリである場合は、縮小式を Ref ID B に追加し、(index $values "A") を (index $values "B") に置き換えます。

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes:
 {{ index $values "B" }}
```

関数

ラベルと注釈をテンプレート化するときは、次の関数を使用できます。

args

args 関数は、オブジェクトのリストをキー arg0、arg1 などのマップに変換します。これは、複数の引数をテンプレートに渡せるようにすることを目的としています。

```
{{define "x"}}{{.arg0}} {{.arg1}}{{end}}{{template "x" (args 1 "2")}}
```

```
1 2
```

externalURL

`externalURL` 関数は Grafana サーバーの外部 URL を返します。

```
{{ externalURL }}
```

```
https://example.com/grafana
```

graphLink

`graphLink` 関数は、指定された式とデータソースの Grafana バージョン 10 で探索する のグラフィックビューへのパスを返します。

```
{{ graphLink "{\"expr\": \"up\", \"datasource\": \"gdev-prometheus\"}" }}
```

```
/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-
prometheus","expr":"up","instant":false,"range":true}]
```

humanize

`humanize` 関数は 10 進数をヒューマナイズします。

```
{{ humanize 1000.0 }}
```

```
1k
```

humanize1024

`humanize1024` は `humanize` と似ていますが、1000 ではなく 1024 をベースとして使用します。

```
{{ humanize1024 1024.0 }}
```

```
1ki
```

humanizeDuration

humanizeDuration 関数は、期間を秒単位でヒューマナイズします。

```
{{ humanizeDuration 60.0 }}
```

```
1m 0s
```

humanizePercentage

humanizePercentage 関数は、比率値をパーセンテージにヒューマナイズします。

```
{{ humanizePercentage 0.2 }}
```

```
20%
```

humanizeTimestamp

humanizeTimestamp 関数は Unix タイムスタンプをヒューマナイズします。

```
{{ humanizeTimestamp 1577836800.0 }}
```

```
2020-01-01 00:00:00 +0000 UTC
```

match

match 関数は、テキストを正規表現パターンと一致させます。

```
{{ match "a.*" "abc" }}
```

```
true
```

pathPrefix

pathPrefix 関数は Grafana サーバーのパスを返します。

```
{{ pathPrefix }}
```

```
/grafana
```

tableLink

tableLink 関数は、指定された式とデータソースの Grafana バージョン 10 で探索する の表形式ビューへのパスを返します。

```
{{ tableLink "{\"expr\": \"up\", \"datasource\": \"gdev-prometheus\"}" }}
```

```
/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-
prometheus","expr":"up","instant":true,"range":false}]
```

title

title 関数は、各単語の最初の文字を大文字にします。

```
{{ title "hello, world!" }}
```

```
Hello, World!
```

toLower

toLower 関数はすべてのテキストを小文字で返します。

```
{{ toLower "Hello, world!" }}
```

```
hello, world!
```

toUpper

toUpper 関数は、すべてのテキストを大文字で返します。

```
{{ toUpper "Hello, world!" }}
```

```
HELLO, WORLD!
```

reReplaceAll

reReplaceAll 関数は、正規表現に一致するテキストを置き換えます。

```
{{ reReplaceAll "localhost:(.*)" "example.com:$1" "localhost:8080" }}
```

```
example.com:8080
```

アラートルールについて

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールは、アラートインスタンスを起動するかどうかを決定する一連の評価基準です。このルールは、1 つ以上のクエリと式、条件、評価の頻度、発射を開始するために条件を満たす必要がある期間で構成されます。

評価するデータセットを「クエリと式」で選択し、「条件」にデータがアラートを発行するための基準 (しきい値) を設定します。

[間隔]は、アラートルールが評価される頻度を指定します。「持続時間」を設定した場合、その条件下にある状態の継続時間を示します。また、アラートルールでは、データがない場合のアラート動作も設定できます。

トピック

- アラートルールタイプ
- 記録ルール
- クエリと条件
- アラートインスタンス
- 名前空間、フォルダ、グループ

- アラートルールの評価
- アラートルールの状態と正常性
- 通知テンプレートの作成

アラートルールタイプ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は複数のアラートルールタイプをサポートしています。各アラートルールタイプ、その仕組みおよびユースケースに最適なルールタイプの決定を説明します。

Grafana マネージドルール

Grafana マネージドルールは、最も柔軟なアラートルールの種類です。これにより、既存の任意のデータソースのデータに対して動作するアラートを作成できます。

複数のデータソースをサポートするだけでなく、[式]を追加してデータを変換し、アラート条件を表現できます。

Grafana マネージドアラート:

- アラートルールは、1 つ以上のデータソースに基づいて Grafana 内に作成されます。
- アラートルールは、Grafana 内からアラートルール評価エンジンによって評価されます。
- アラートは、内部 Grafana アラートマネージャーを使用して配信されます。

> Note
>
> また、外部アラートマネージャーを使用して配信するようにアラートを設定したり、内部アラートマネージャーと外部アラートマネージャーの両方を使用したりできます。詳細については、「外部アラートマネージャーの追加」を参照してください。

データソースマネージドルール

データソースマネージドアラートルールを作成するには、互換性のある Prometheus または Loki データソースが必要です。データソースをテストし、Ruler API がサポートされているかどうかを観察することで、データソースが Grafana 経由でルール作成をサポートしているかどうかを確認できます。

データソースマネージドアラートでは、次のようになります。

- アラートルールが作成され、データソース自体に保存されます。
- アラートルールは、Prometheus データに基づいてのみ作成できます。
- アラートルールの評価と配信は、高可用性と耐障害性のために複数のノードに分散されます。

アラートルールタイプを選択する

使用するアラートルールタイプを選択するときは、Grafana マネージドアラートルールとデータソースマネージドアラートルールにおける以下の比較を考慮してください。

機能	Grafana マネージドアラートルール	Loki/Mimir マネージドアラートルール
サポートされているデータソースのデータに基づいてアラートルールを作成する	あり	不可: Prometheus データに基づくアラートルールのみを作成できます。データソースで Ruler API が有効になっている必要があります。
データソースを組み合わせる	あり	いいえ
記録ルールへの対応	なし	あり
式を追加してデータを変換し、アラート条件を設定する	あり	不可
アラート通知でイメージを使用する	あり	いいえ
スケーリング	リソースを大量に消費し、データベースに依存し、一時	アラートルールをデータソース自体に保存し、「無限」

機能	Grafana マネージドアラートルール	Loki/Mimir マネージドアラートルール
	的なエラーが発生する可能性があります。垂直方向にのみスケーリングされます。	スケーリングを許可します。データの場所からアラート通知を生成して送信します。
アラートルールの評価と配信	アラートルールの評価と配信は Grafana 内、外部アラートマネージャー、またはその両方を使用して行われます。	アラートルールの評価とアラート配信は分散されます。つまり、単一障害点はありません。

記録ルール

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

記録ルールは、互換性のある Prometheus または Loki データソースでのみ使用できます。

記録ルールでは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列セットとして保存できます。これは、集計データに対してアラートを実行する場合や、計算コストの高い式を繰り返しクエリするダッシュボードがある場合に便利です。

この新しい時系列のクエリは、特にダッシュボードでは、ダッシュボードが更新されるたびに同じ式をクエリするため、より高速になります。

Prometheus での記録ルールの詳細については、こちらを参照してください。

クエリと条件

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、クエリは、MySQL や PostgreSQL などのデータベース、Prometheus、InfluxDB、Graphite などの時系列データベース、OpenSearch 、Amazon CloudWatch 、Azure Monitor、Google Cloud Monitoring などのサービスなど、サポートされているデータソースからのデータの取得と変換に重要な役割を果たします。

サポートされているデータソースの詳細については、「データソースと Grafana アラート」を参照してください。

クエリを実行するプロセスには、データソースの定義、取得するデータの特定、関連するフィルターまたは変換の適用が含まれます。選択したデータソースに固有のクエリ言語または構文は、これらのクエリのコンストラクトに使用されます。

アラートでは、測定するデータと、アラートルールを起動する前に満たす必要がある条件を取得するクエリを定義します。

アラートルールは、測定するデータを選択する 1 つ以上のクエリと式で構成されます。

クエリと式の詳細については、「データのクエリと変換」を参照してください。

データソースクエリ

Grafana のクエリは、使用するデータソースとクエリ言語に応じて、さまざまな方法で適用できます。各データソースのクエリエディタには、独自の機能を活用するクエリの作成に役立つカスタマイズされたユーザーインターフェイスが用意されています。

クエリ言語の違いにより、各データソースクエリエディタの外観と関数は異なります。データソースによっては、クエリエディタが自動補完機能、メトリクス名、変数の提案、ビジュアルクエリ構築インターフェイスを提供する場合があります。

一般的なタイプのクエリコンポーネントには、次のようなものがあります。

メトリクスまたはデータフィールド – CPU 使用量、ネットワークトラフィック、センサー読み取りなど、取得する特定のメトリクスまたはデータフィールドを指定します。

時間範囲 – 過去 1 時間、特定の日、カスタム時間範囲など、データを取得する時間範囲を指定します。

フィルター – フィルターを適用して、特定のタグ、ホスト、またはアプリケーションでデータをフィルタリングするなど、特定の基準に基づいてデータを絞り込みます。

集計 - データを集計して、特定の期間の平均、合計、カウントなどのメトリクスを計算します。

グループ分け – データを特定のディメンションまたはタグでグループ化して、集計ビューまたは内訳を作成します。

> **Note**
>
> Grafana は、テンプレート変数を使用したアラートクエリはサポートしていません。詳細については、Grafana Labs フォーラム内のこちらを参照してください。

クエリ式

Grafana では、式を使用して、クエリされたデータソースデータに対して計算、変換、または集計を実行します。これにより、数学演算、関数、または論理式に基づいてカスタムメトリクスを作成したり、既存のメトリクスを変更したりできます。

クエリ式を活用することで、ユーザーは 2 つの値間の変化率の計算、対数関数や三角関数などの関数の適用、特定の時間範囲やディメンションにわたるデータの集約、さまざまなシナリオを処理するための条件ロジックの実装などのタスクを実行できます。

アラートでは、Grafana 管理のアラートルールにのみ式を使用できます。式ごとに、Math、Reduce、リサンプリング式を選択できます。これらは、シリーズごとに個別のアラートを生成するため、多次元ルールと呼ばれます。

クラシック条件を使用して、条件が満たされたときに単一のアラートをトリガーするアラートルールを作成することもできます。そのため、複数のシリーズでアラート条件が満たされた場合でも、Grafana が送信するアラートは 1 件のみです。

> **Note**
>
> クラシック条件は主に互換性上の理由から存在するため、可能であれば避けるようにしてください。

[削減]

選択した時間範囲の時系列値を 1 つの値に集約します。

数値演算

時系列および数値データに対して自由形式の数学関数/オペレーションを実行します。時系列データの事前処理や、数値データのアラート条件の定義に使用できます。

[リサンプリング]

時間範囲を新しいタイムスタンプのセットに再調整します。これは、タイムスタンプが一致しない異なるデータソースからの時系列データを比較する場合に役立ちます。

Threshold

時系列データがしきい値条件と一致するかどうかを確認します。

しきい値式を使用すると、2 つの単一の値を比較できます。条件が偽の場合 `0` と、条件が真の場合に 1 が返されます。次のしきい値関数が使用可能です。

- Is above (x > y)
- Is below (x < y)
- Is within range (x > y1 AND x < y2)
- Is outside range (x < y1 AND x > y2)

クラシック条件

時系列データがアラート条件と一致するかどうかを確認します。

> **Note**
>
> クラシック条件クエリ式は、条件を満たす時系列の数に関係なく、常に 1 つのアラートインスタンスのみを生成します。クラシック条件は主に互換性上の理由から存在するため、可能であれば避けるようにしてください。

集計

Grafana アラートでは、クエリをさらに絞り込むために、次の集計関数が用意されています。

これらの関数は、削減 式とクラシック条件式でのみ使用できます。

関数	式	その内容
avg	削減/クラシック	値の平均を表示します
min	削減/クラシック	最小値を表示します
max	削減/クラシック	最大値を表示します
sum	削減/クラシック	すべての値の合計を表示します
count	削減/クラシック	結果の値の数をカウントします
last	削減/クラシック	最後の値を表示します
median	削減/クラシック	中央値を表示します。
diff	クラシック	最も新しい値と最も古い値の差を表示します。
diff_abs	クラシック	diff の絶対値を表示します。
percent_diff	クラシック	最も新しい値と最も古い値の差の割合値を表示します。
percent_diff_abs	クラシック	percent_diff の絶対値を表示します
count_non_null	クラシック	結果セット内の値のうち、null でない値の数を表示します。

アラート条件

アラート条件は、アラートが発射するかどうかを決定するクエリまたは式です。この条件は、アラートが生成する値に応じて決定されます。アラートのトリガーを決定する条件は 1 つだけです。

クエリや式を定義したら、そのいずれかをアラートルール条件として選択します。

クエリされたデータが定義された条件を満たすと、Grafana は関連付けられたアラートをトリガーします。このアラートは、E メール、Slack、PagerDuty などのさまざまなチャネルを介して通知を送信するように設定できます。通知は、満たされている条件について通知するため、適切なアクションを実行したり、根本的な問題を調査したりできます。

デフォルトでは、最後に追加された式がアラート条件として使用されます。

復旧しきい値

アラートのフラッピングのノイズを減らすには、アラートしきい値とは異なる復旧しきい値を設定します。

アラートのフラッピングは、メトリクスがアラートのしきい値条件を超えた場合に発生し、頻繁な状態の変化につながる可能性があり、生成される通知が多すぎます。

Grafana 管理のアラートルールは、特定の期間にわたって評価されます。各評価中、クエリの結果はアラートルールで設定されたしきい値に対してチェックされます。メトリクスの値がしきい値を超えると、アラートルールが生成され、通知が送信されます。値がしきい値を下回り、このメトリクスに対してアクティブなアラートがある場合、アラートが解決され、別の通知が送信されます。

ノイズの多いメトリクスのアラートルールを作成するのは難しい場合があります。つまり、メトリクスの値が継続的にしきい値を上回ったり下回ったりする場合です。これはフラッピングと呼ばれ、一連の発射 - 解決 - 発射通知とノイズの多いアラート状態履歴になります。

例えば、しきい値が 1000 ミリ秒のレイテンシーのアラートがあり、その数が 1000 の周囲 (例えば、980 ->1010 -> 990 -> 1020 など) で変動すると、それぞれが通知をトリガーします。

この問題を解決するには、 (カスタム) 復旧しきい値を設定できます。これは基本的に、1 つのしきい値ではなく 2 つのしきい値を持つことを意味します。アラートは、最初のしきい値を超えたときにトリガーされ、2 番目のしきい値を超えたときにのみ解決されます。

例えば、しきい値を 1000 ミリ秒、復旧しきい値を 900 ミリ秒に設定することができます。これにより、アラートルールは 900 ミリ秒未満になり、フラッピングが軽減された場合にのみ発射を停止します。

アラートインスタンス

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana マネージドアラートは、多次元アラートをサポートします。各アラートルールは、複数のアラートインスタンスを作成できます。これは、1 つの式で複数のシリーズを観察する場合に強力です。

次の PromQL 式を検討してください。

```
sum by(cpu) (
  rate(node_cpu_seconds_total{mode!="idle"}[1m])
)
```

この式を使用するルールは、最初の評価後に監視している CPU の数と同じ数のアラートインスタンスを作成し、単一のルールで各 CPU のステータスをレポートできるようにします。

名前空間、フォルダ、グループ

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートは、Grafana マネージドルールのフォルダと、Mimir、Loki、または Prometheus のルールとグループ名の名前空間を使用して整理できます。

名前空間とフォルダ

Grafana マネージドルールを作成する場合、 フォルダを使用してアクセス制御を実行し、特定のフォルダ内のすべてのルールへのアクセスを許可または拒否できます。

名前空間には 1 つ以上のグループが含まれます。グループ内のルールは、一定の間隔で順番に実行されます。デフォルトの間隔は 1 分です。Grafana Mimi または Loki ルールの名前空間とグループの名前を変更したり、グループ評価間隔を編集したりできます。

グループ

グループ内のルールは、一定の間隔で順番に実行されます。つまり、ルールは同時に評価されず、出現順に評価されません。デフォルトの間隔は 1 分です。Grafana Mimir または Loki ルールの名前空間または Loki ルールの名前空間とグループの名前を変更したり、グループ評価間隔を編集したりできます。

> **Tip**
>
> ルールを異なる間隔で同時に評価したい場合は、異なるグループに保存することを検討してください。

> **Note**
>
> Grafana マネージドアラートルールは、順番ではなく同時に評価されます。

アラートルールの評価

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールの評価を使用して、アラートルールを評価する頻度と状態の変更速度を決定します。

これを行うには、アラートルールが適切な評価グループにあることを確認し、ユースケースに最適な保留期間を設定する必要があります。

評価グループ

すべてのアラートルールは評価グループの一部です。各評価グループには、アラートルールをチェックする頻度を決定する評価間隔が含まれます。

同じグループ内のデータソースマネージドアラートルールは順番に評価されますが、異なるグループのアラートルールは同時に評価できます。この機能は、アラートルールの前に記録ルールを評価したい場合に特に便利です。

Grafana マネージドアラートルールは、アラートルールグループに関係なく同時に評価されます。デフォルトの評価間隔は 10 秒に設定されています。つまり、Grafana が管理するアラートルールは、10 秒ごとに順番に (例えば 10:00:00、10:00:10、10:00:20 などに) 評価されます。必要に応じて、独自の評価間隔を設定することもできます。

> **Note**
>
> 通知ポリシー内の評価グループとアラートのグループ化は異なるものです。通知ポリシーでグループ化すると、同じラベルを共有する複数のアラートを同じ時刻メッセージで送信できます。

保留期間

保留期間を設定することで、一時的な問題に対する不要なアラートを回避できます。

保留期間では、条件に違反している可能性があるためにアラートルールを発射するまでの期間を選択します。

例

アラートルールの評価間隔が 30 秒ごとに設定され、保留期間が 90 秒であるとします。

評価は次のように行われます。

[00:30] 最初の評価 - 条件が満たされていない。

[01:00] 2 回目の評価 - 条件に違反している、保留カウンターが開始。アラートの保留が開始

[01:30] 3 回目の評価 - 条件に違反している。保留カウンター = 30 秒。保留中の状態。

[02:00] 4 回目の評価 - 条件に違反している。保留カウンター = 60 秒保留中の状態。

[02:30] 5 回目の評価 - 条件に違反している。保留カウンター = 90 秒。アラートの発射開始

アクションを実行する前に一定時間違反状態である必要がある条件がアラートルールにある場合、その状態は次のように変化します。

- 条件が最初に違反すると、ルールは「保留中」状態になります。
- ルールは、条件が保留期間という必要な期間にわたって中断されるまで、「保留」状態のままになります。
- 必要な時間が過ぎると、ルールは「発射」状態になります。

- 保留期間中に条件の違反が解消された場合、ルールは通常の状態に戻ります。

> **Note**
> 保留状態をスキップする場合は、保留期間を 0 に設定します。これにより、保留期間を効果的にスキップし、条件が満たされるとすぐにアラートルールの発射が開始されます。

アラートルールが発射されると、アラートインスタンスが生成され、アラートマネージャーに送信されます。

アラートルールの状態と正常性

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールの状態と正常性は、アラートに関するいくつかの主要なステータスインジケータを理解するのに役立ちます。

アラートルールの状態 、アラートインスタンスの状態、アラートルールの正常性の 3 つの主要なコンポーネントがあります。それぞれに関連はありますが、各コンポーネントは微妙に異なる情報を伝達します。

アラートルールの状態

アラートルールは、次に示す状態のいずれかになります。

状態	説明
普通	評価エンジンによって返される時系列のいずれも `Pending` または `Firing` 状態ではありません。

状態	説明
保留中	評価エンジンから返される少なくとも 1 つの時系列は Pending です。
発射	評価エンジンから返される少なくとも 1 つの時系列は Firing です。

> **Note**
>
> アラートは最初に pending に移行し、次に firing に移行します。そのため、アラートが発生する前に少なくとも 2 つの評価サイクルが必要になります。

アラートインスタンスの状態

アラートインスタンスは、次に示す状態のいずれかになります。

状態	説明
普通	発射も保留中もなく、すべてが正しく機能しているアラートの状態。
保留中	設定されたしきい値期間未満でアクティブであったアラートの状態。
[アラート]	設定されたしきい値期間よりも長くアクティブであったアラートの状態。
NoData	設定された時間枠のデータを受信しない状態。
エラー	アラートルールの評価を試みたときに発生したエラー。

最後の状態を保持する

アラートルールは、 NoData または Error 状態が発生したときに最後の状態を保持するように設定できます。これにより、アラートの発射、解決と再発射の両方が防止されます。通常の評価と同様に、アラートルールは保留期間が経過すると Pending から Firing に移行します。

アラートルールの正常性

アラートルールは、次のいずれかの正常性ステータスとなります。

状態	説明
Ok	アラートルールを評価するときにエラーがありません。
エラー	アラートルールの評価中にエラーが発生しました。
NoData	ルール評価中に返される少なくとも 1 つの時系列にデータがありません。

NoData および **Error** [の特別なアラート]

アラートルールの評価で状態 NoData または Error が生成されると、Grafana アラートは、次の追加ラベルを含むアラートインスタンスを生成します。

ラベル	説明
AlertName	状態に応じて、DatasourceNoData または DatasourceError のいずれか。
datasource_uid	状態の原因となったデータソースの UID。

これらのアラートは、サイレンスを追加したり、コンタクトポイントにルーティングしたりすることで、通常のアラートと同じ方法で処理できます。

通知テンプレートの作成

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントを介して送信される通知は、通知テンプレートを使用して構築されます。Grafana のデフォルトテンプレートは Go テンプレートシステムに基づいており、一部のフィールドはテキストとして評価され、他のフィールドは HTML (エスケープに影響する可能性があります) として評価されます。

デフォルトのテンプレート [default_template.go] は、カスタムテンプレートの便利なリファレンスとして機能します。

コンタクトポイントに使用するほとんどのフィールドはテンプレート化できるため、再利用可能なカスタムテンプレートを作成し、複数のコンタクトポイントで使用すると良いでしょう。テンプレートを使用したカスタム通知の詳細については、「通知のカスタマイズ」を参照してください。

テンプレートのネスト

テンプレートは他のテンプレートに埋め込むことができます。

例えば、 `define` キーワードを使用してテンプレートフラグメントを定義できます。

```
{{ define "mytemplate" }}
  {{ len .Alerts.Firing }} firing. {{ len .Alerts.Resolved }} resolved.
{{ end }}
```

その後、 `template` キーワードを使用して、このフラグメント内にカスタムテンプレートを埋め込むことができます。例:

```
Alert summary:
{{ template "mytemplate" . }}
```

以下の任意のビルトインのテンプレートオプションを使用して、カスタムテンプレートを埋め込むことができます。

名前	メモ
`default.title`	全体的なステータス情報を表示します。

名前	メモ
`default.message`	発生中および解決済みのアラートの概要をフォーマット付きで提供します。
`teams.default.message`	`default.messsage` と同様に、Microsoft Teams 用にフォーマットされています。

通知テンプレートの HTML

アラート通知テンプレートの HTML はエスケープされます。結果の通知では、HTML のレンダリングはサポートされていません。

一部の通知機能は、結果の通知のルックアンドフィールを変更する代替方法をサポートしています。例えば、Grafana は `<grafana-install-dir>/public/emails/ng_alert_notification.html` に E メールをアラートするためのベーステンプレートをインストールします。このファイルを編集して、すべてのアラートメールの外観を変更できます。

アラートマネージャー

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートマネージャーを使用すると、アラートを迅速かつ効率的に管理して対応できます。E メールや Slack など選択したチャネルを介して通知を送信することで、アラートを受信し、ミュート、抑制、グループ化、ルーティングを処理します。

Grafana では、Grafana アラートマネージャーまたは外部アラートマネージャーを使用できます。複数のアラートマネージャーを実行することもできます。決定は、セットアップとアラートの生成場所によって異なります。

Grafana アラートマネージャー

Grafana アラートマネージャーは、事前設定済みで、デフォルトで選択できる内部アラートマネージャーです。

Grafana アラートマネージャーは Grafana からアラートを受信できますが、Mimir や Loki など、Grafana の外部からアラートを受信することはできません。

> **Note**
>
> 禁止ルールは Grafana アラートマネージャーではサポートされていません。

外部アラートマネージャー

Grafana、Loki、Mimir、Prometheus のすべてのアラートを受信するために 1 つのアラートマネージャーを使用したい場合は、外部のアラートマネージャーを使用するように Grafana を設定できます。この外部アラートマネージャーは、Grafana 内から設定および管理できます。

Grafana アラートマネージャーの代わりに、独自の外部アラートマネージャーを設定し、そこでアラートを送信する場合の例を 2 つ示します。

1. Prometheus などの他のアラートジェネレーターがあるため、独自のクラウドインフラストラクチャに既にオンプレミスのアラートマネージャーを設定しており、引き続き使用したい場合。
2. Prometheus オンプレミスとホストされた Grafana の両方を使用して、クラウドインフラストラクチャで実行されるのと同じアラートマネージャーにアラートを送信したい場合。

アラートマネージャーは、[コンタクトポイントのアラート] および [通知ポリシー] ページのドロップダウンメニューに表示されます。

データソースをプロビジョニングする場合は、`jsonData` フィールドの `handleGrafanaManagedAlerts` フラグを `true` に設定して、Grafana が管理するアラートをこのアラートマネージャーに送信します。

コンタクトポイント

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントには、通知を送信するための設定が含まれています。コンタクトポイントは統合のリストであり、それぞれが特定の E メールアドレス、サービス、または URL に通知を送信します。コンタクトポイントには、同じ種類の複数の統合、または異なる種類の統合の組み合わせを含めることができます。例えば、コンタクトポイントには、Pagerduty 統合、Amazon SNS と Slack 統合、または Pagerduty 統合、Slack 統合、および 2 つの Amazon SNS 統合を含めることができます。統合なしでコンタクトポイントを設定することもできますが、その場合、通知は送信されません。

コンタクトポイントは、通知ポリシーに追加されるまで通知を送信できません。通知ポリシーは 1 つのコンタクトポイントにのみアラートを送信できますが、複数の通知ポリシーに同時にコンタクトポイントを追加できます。アラートが通知ポリシーと一致すると、アラートはその通知ポリシーのコンタクトポイントに送信され、その通知ポリシーはその設定の各統合に通知を送信します。

Grafana アラートマネージャーと外部アラートマネージャーのコンタクトポイントを設定できます。

通知テンプレートを使用して、コンタクトポイントタイプの通知メッセージをカスタマイズすることもできます。

サポートされているコンタクトポイントタイプ

次の表に、Grafana でサポートされているコンタクトポイントのタイプを示します。

名前	タイプ
Amazon SNS	`sns`
OpsGenie	`opsgenie`
Pager Duty	`pagerduty`
Slack	`slack`
VictorOps	`victorops`

コンタクトポイントの詳細については、「コンタクトポイントの設定」と「通知のカスタマイズ」を参照してください。

通知

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラート通知を送信する方法、タイミング、および場所を選択することは、アラートシステムを設定する上で重要です。これらの決定は、問題を迅速に解決し、重要なものを見逃さない能力に直接影響します。

最初のステップとして、アラート通知の送信先を定義するコンタクトポイントを定義します。コンタクトポイントは、通知の配信に使用される 1 つ以上の統合のセットです。通知テンプレートをコンタクトポイントに追加して、通知の再利用と一貫したメッセージングを行います。

次に、アラートがコンタクトポイントにルーティングされる場所、タイミング、方法に関する一連のルールである通知ポリシーを作成します。通知ポリシーでは、作成したコンタクトポイントのいずれかを選択して、アラート通知を送信する場所を定義します。

アラートマネージャー

Grafana はアラートマネージャーを使用して、アラートの発射と解決の通知を送信します。Grafana には、ユーザーインターフェイスで Grafana と呼ばれる独自のアラートマネージャーがありますが、Prometheus アラートマネージャーなどの他のアラートマネージャーからの通知の送信もサポートしています。Grafana アラートマネージャーは、通知ポリシーと問い合わせポイントを使用して、通知の送信方法と送信先、通知の送信頻度、アラートをすべて同じ通知で送信するか、ラベルのセットに基づいてグループ化された通知で送信するか、個別の通知として送信するかを設定します。

通知ポリシー

通知ポリシーは、通知の送信日時と送信場所を制御します。通知ポリシーでは、すべてのアラートを同じ通知でまとめて送信するか、ラベルのセットに基づいてグループ化された通知でアラートを送信するか、個別の通知としてアラートを送信するかを選択できます。各通知ポリシーを設定して、通知の送信頻度を制御するだけでなく、1 日の特定時刻および特定の曜日に通知を禁止するための 1 つ以上のミュートタイミングを設定することもできます。

通知ポリシーは、ツリーのルートにデフォルトポリシーと呼ばれる通知ポリシーがあるツリー構造で構成されています。デフォルトポリシーは 1 つだけ設定でき、削除できません。

特定のルーティングポリシーはルートポリシーの子であり、一致するラベルのセットに基づいてすべてのアラートまたはアラートのサブセットを照合するために使用できます。通知ポリシーは、一致するラベルがアラート内のラベルと一致するときにアラートと一致します。

ネストされたポリシーには独自のネストされたポリシーを含めることができ、アラートの追加マッチングが可能になります。ネストされたポリシーの例としては、Ops チームにインフラストラクチャアラートを送信する場合があります。子ポリシーは、Pagerduty に高優先度アラートを送信し、Slack に低優先度アラートを送信することができます。

ラベルに関係なく、すべてのアラートはデフォルトのポリシーと一致します。ただし、デフォルトのポリシーがアラートを受信すると、ネストされた各ポリシーが調査され、アラートに一致する最初のネストされたポリシーにアラートが送信されます。ネストされたポリシーにさらにネストされたポリシーがある場合、ネストされたポリシーの 1 つとアラートとのマッチングを試行できます。ネストされたポリシーがアラートと一致しない場合、ポリシー自体が一致するポリシーになります。ネストされたポリシーがない場合、またはネストされたポリシーがアラートと一致しない場合、デフォルトのポリシーは一致するポリシーです。

通知ポリシーの詳細については、「通知ポリシー」を参照してください。

通知テンプレート

テンプレートを使用して通知をカスタマイズできます。例えば、テンプレートを使用して、Slack に送信される通知のタイトルとメッセージを変更することができます。

テンプレートは個々の統合やコンタクトポイントに限定されず、同じコンタクトポイント内の多数の統合や、異なるコンタクトポイント間の統合でも使用できます。例えば、Grafana ユーザーは、`custom_subject_or_title` というテンプレートを作成し、2 つの個別のテンプレートを作成することなく、Pager Duty のテンプレート件名と Slack メッセージのタイトルの両方に使用できます。

すべての通知テンプレートはアラートページの [コンタクトポイント] タブにあり、[Go のテンプレート言語]で記述されます。

通知のカスタマイズの詳細については、「通知のカスタマイズ」を参照してください。

サイレンス

サイレンスを使用して、1 つ以上の発射ルールからの通知をミュートできます。サイレンスは、アラートの発射や解決を停止したり、ユーザーインターフェイスで発射アラートを非表示にしたりしません。サイレンスは、分、時間、日、月、または年で設定し、その期間だけ続きます。

サイレンスの使用の詳細については、「アラート通知のサイレンシング」を参照してください。

通知ポリシー

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知ポリシーにより、さまざまな受信者にアラートをルーティングする柔軟な方法が提供されます。ラベルマッチャーを使用すると、個々のアラートルールをすべて更新することなく、アラート通知配信を変更できます。

このセクションでは、通知ポリシーの設定を最大限に活用できるように、通知ポリシーの仕組みと構造について詳しく説明します。

ポリシーツリー

通知ポリシーはリストではなく、ツリー構造に従って構造化されます。つまり、各ポリシーに子ポリシーなどを含めることができます。通知ポリシーツリーのルートは、デフォルトの通知ポリシーと呼ばれます。

各ポリシーは、処理に関心がないラベルを指定する一連のラベルマッチャー (0 以上) で構成されます。

ラベルの一致の詳細については、「ラベル一致の仕組み」を参照してください。

> **Note**
>
> 通知ポリシーにラベルマッチャーを設定していない場合、通知ポリシーはすべてのアラートインスタンスと一致します。これにより、通知ポリシーで [兄弟の照合の続行] を有効にしない限り、子ポリシーが評価されない場合があります。

ルーティング

どの通知ポリシーがどのアラートインスタンスを処理するかを決定するには、既定の通知ポリシーから開始して、既存の通知ポリシーのセットを確認する必要があります。

デフォルトポリシー以外のポリシーが設定されていない場合は、デフォルトポリシーがアラートインスタンスを処理します。

デフォルトポリシー以外のポリシーが定義されている場合は、それらの通知ポリシーが表示される順序で評価されます。

通知ポリシーにアラートインスタンスのラベルと一致するラベルマッチャーがある場合、通知ポリシーは子ポリシーに降りてきます。子ポリシーがある場合は、ラベルのセットをさらに絞り込むラベルマッチャーを持つ可能性のある子ポリシーを探し続け、子ポリシーが見つからなくなるまで続行されます。

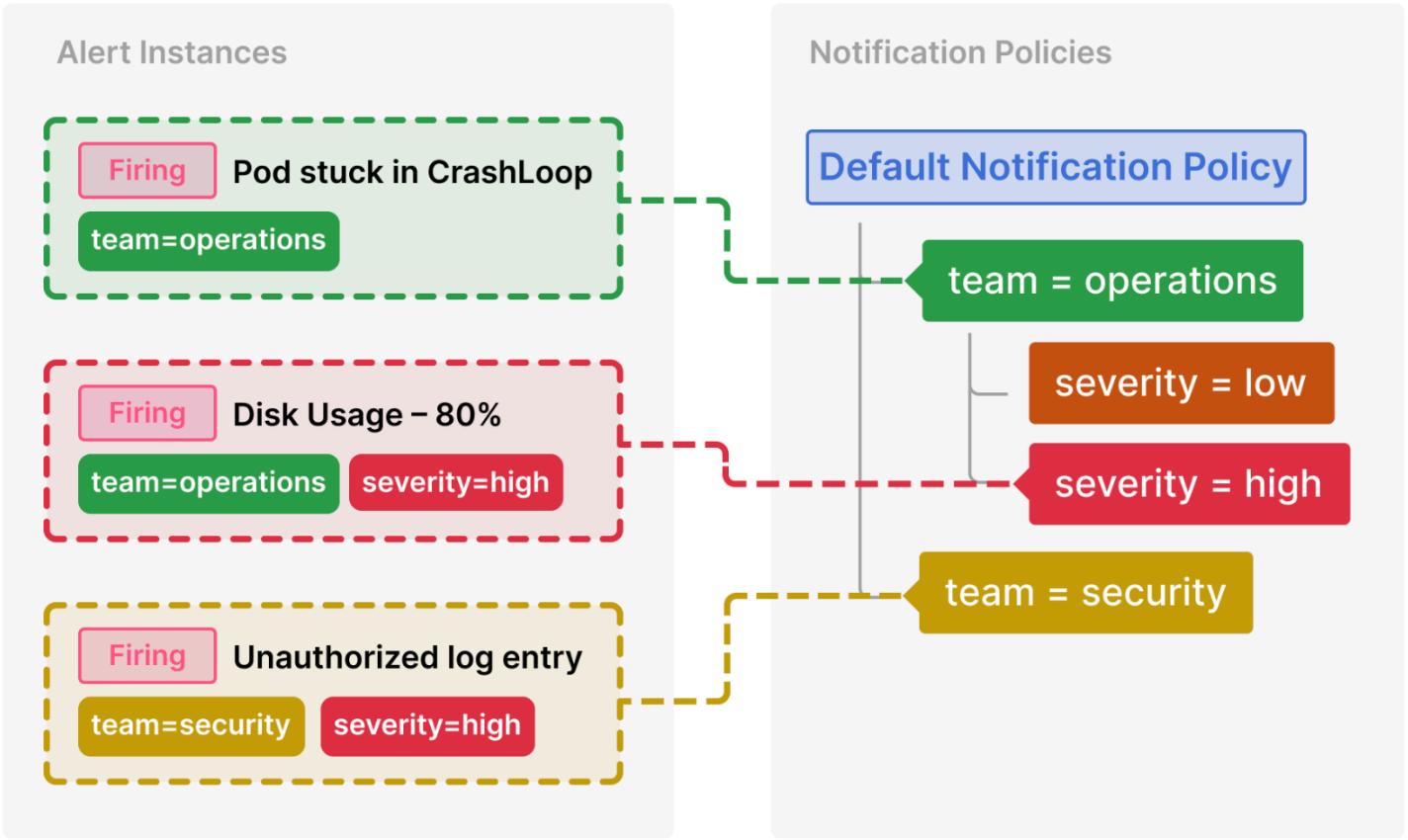
通知ポリシーで子ポリシーが定義されていない場合、またはアラートインスタンスのラベルに一致するラベルマッチャーが子ポリシーにない場合、親通知ポリシーが使用されます。

一致するポリシーが見つかるとすぐに、システムは他の一致するポリシーの検索を中止します。一致する可能性のある他のポリシーを引き続き検索する場合は、その特定のポリシーで [兄弟の照合の続行]を有効にします。

最後に、どの通知ポリシーも選択されていない場合は、デフォルトの通知ポリシーが使用されます。

ルーティングの例

以下は、比較的単純な通知ポリシーツリーといくつかのアラートインスタンスの例です。



これらのポリシーの選択方法の内訳を次に示します。

[CrashLoop でスタックされた Pod]には `severity` ラベルがないため、子ポリシーは一致しません。`team=operations` ラベルが付いているため、最初のポリシーは一致します。

一致が既に見つかり、[兄弟の照合の続行]がそのポリシーに設定されていないため、`team=security` ポリシーは評価されません。

ディスク使用量 – 80% には `team` と `severity` の両方のラベルがあり、オペレーションチームの子ポリシーと一致します。

許可されていないログエントリには `team` ラベルがありますが、値が同じではないことを理由として最初のポリシー (`team=operations`) と一致しないため、検索が続行され、`team=security` ポリシーと一致します。子ポリシーがないため、追加の `severity=high` ラベルは無視されます。

継承

子ポリシーは、アラートインスタンスのルーティングに役立つ概念であるだけでなく、親ポリシーからプロパティを継承します。これは、デフォルトの通知ポリシーの子ポリシーであるポリシーにも適用されます。

次のプロパティは子ポリシーによって継承されます。

- コンタクトポイント
- オプションのグループ化
- タイミングオプション
- ミュートタイミング

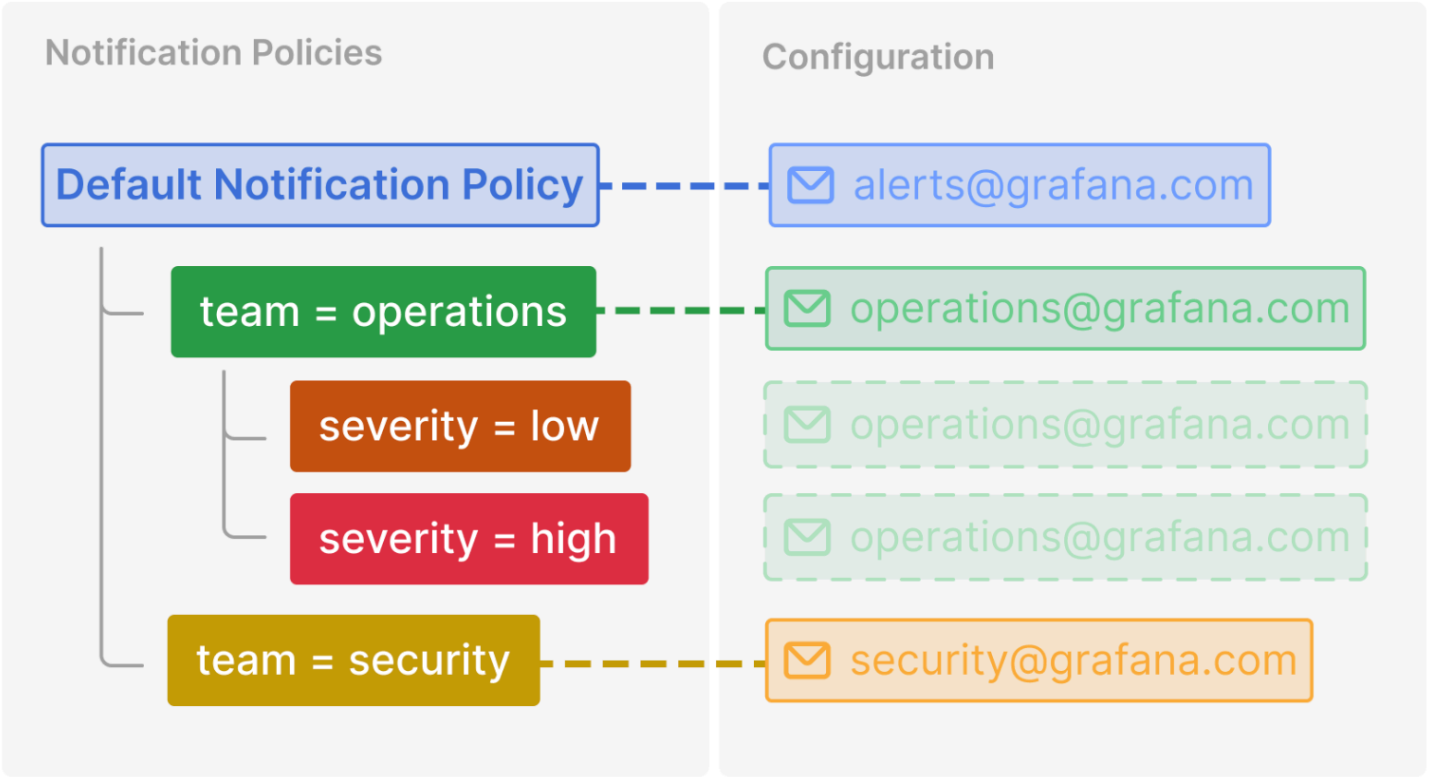
継承されたプロパティを上書きする場合は、これらのプロパティを個々のポリシーで上書きします。

親ポリシーからコンタクトポイントを継承するには、空白のままにします。継承されたグループ化オプションを上書きするには、[グループ化の上書き] を有効にします。継承されたタイミングオプションを上書きするには、[一般的なタイミングの上書き] を有効にします。

継承の例

次の例は、前の例の通知ポリシーツリーで `team=operations` の子ポリシーがコンタクトポイントを継承する方法を示しています。

これにより、子ポリシーごとに同じコンタクトポイントを複数回指定する必要を回避できます。



追加設定のオプション

グループ化

グループ化は Grafana アラートの重要な機能です。これにより、関連するアラートをバッチ処理して通知の数を減らすことができます。これは、エンジニアオンコールなどのファーストレスポンダーに通知が配信される場合に特に重要であります。ファーストレスポンダーが短期間に多くの通知を受信すると圧倒され、場合によってはインシデントに対応する能力に悪影響を及ぼす可能性があります。例えば、多くのシステムがダウンしている大規模な障害を考えてみましょう。この場合、グループ化は、1 回の通話と 100 回の通話の受信の違いになります。

通知ポリシーの Group by オプションを使用して、アラートをグループ分けする方法を選択します。デフォルトでは、Grafana グループの通知ポリシーは、`alertname` および `grafana_folder` ラベルを使用してアラートルールによって一緒にアラートを発します (アラート名は複数のフォルダで一意ではないため)。アラートルール以外の方法でアラートをグループ分けする場合は、グループ化をラベルの他の組み合わせに変更します。

グループ化を無効にする

すべてのアラートを個別の通知として受信する場合は、 `...` という特別なラベルでグループ化することで受信できます。これは、アラートがファーストレスポンダーではなく自動システムに配信される場合に便利です。

すべてのアラートを含む単一のグループ

すべてのアラートを 1 つの通知でまとめて受信する場合は、グループを空のままにすると受信できます。

タイミングオプション

タイミングオプションは、アラートグループごとに通知が送信される頻度を決定します。知っておくべきタイマーは、グループ待機、グループ間隔、繰り返し間隔の 3 つです。

グループ待機

グループ待機は、新しいアラートグループの最初の通知を送信するまでに Grafana が待機する時間です。グループの待機時間が長いほど、他のアラートが到着する時間が長くなります。グループの待機時間が短いほど、最初の通知は早く送信されますが、不完全な通知を送信するリスクがあります。ユースケースに最も適したグループ待機を常に選択するようにするべきです。

デフォルト: 30 秒

グループ間隔

アラートの新しいグループに対して最初の通知が送信されると、Grafana はグループ間隔タイマーを開始します。これは、変更に関する通知をグループに送信する前に Grafana が待機する時間です。例えば、既存のアラートが解決されている間に、別の発射アラートがグループに追加される場合があります。グループの待機が原因でアラートが最初の通知に含めるのに遅すぎる場合、グループの間隔が過ぎた後の後続の通知に含まれます。グループ間隔が経過すると、Grafana はグループ間隔タイマーをリセットします。これは、グループが削除された後にグループ内にアラートがなくなるまで繰り返されます。

デフォルト: 5 分

繰り返し間隔

繰り返し間隔は、最後の通知以降にグループが変更されていない場合に通知を繰り返す頻度を決定します。これらは、一部のアラートがまだ発せられていることのリマインダーと考えることができます。繰り返し間隔はグループ間隔と密接に関連しています。つまり、繰り返し間隔はグループ間隔以上であるだけでなく、グループ間隔の倍数である必要があります。繰り返し間隔がグループ間隔の倍数でない場合、1 に強制されます。例えば、グループ間隔が 5 分で、繰り返し間隔が 9 分の場合、繰り返し間隔は最も近い 5 の倍数である 10 分に切り上げられます。

デフォルト: 4 時間

高可用性のアラート

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Amazon Managed Grafana は、作成するワークスペースごとに複数のアベイラビリティーゾーンで複数のインスタンスを実行するなど、高可用性用に設定されています。

Grafana アラートは、通知の配信からアラートルールの評価を分離する Prometheus モデルを使用しています。このモデルでは、アラートルールの評価はアラートジェネレーターで行われ、通知の配信

はアラートレシーバーで行われます。Grafana アラートでは、アラートジェネレーターはスケジューラで、レシーバーはアラートマネージャーです。

高可用性設定では、すべてのアラートルールがすべてのインスタンスで評価されます。アラートルールの評価は、重複していると考えることができます。これにより、Grafana アラートは、少なくとも 1 つの Grafana インスタンスが動作している限り、アラートルールが評価され、アラートの通知が送信されることを保証します。この重複は状態履歴に表示され、高可用性を使用しているかどうかを判別するのに役立ちます。

アラートの設定

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートの作成と管理に必要な機能と統合を設定します。

前提条件

アラートを設定する前に、以下を実行する必要があります。

- データソースを設定します。
- 選択したデータソースが Grafana アラートと互換性があり、サポートされていることを確認します。

アラートを設定するには

1. アラートルールを設定します。
   - Grafana 管理またはデータソース管理のアラートルールと記録ルールを作成します。
2. コンタクトポイントを設定します。
   - デフォルトのコンタクトポイントを確認し、システムのコンタクトを更新します。

- オプションで、新しいコンタクトポイントと統合を追加します。

3. 通知ポリシーを設定する

- デフォルトの通知ポリシーを確認し、システムの通知ポリシーを更新します。
- 必要に応じて、ネストされたポリシーを追加します。
- また、ラベルとラベルマッチャーを追加してアラートルーティングを制御することもできます。

以下のトピックでは、外部アラートマネージャーの設定や Grafana マネージドアラートを Grafana の外部にルーティングする方法など、追加の設定オプションについて詳しく説明します。

トピック

- 従来のダッシュボードアラートを Grafana アラートに移行する
- 外部アラートマネージャーの追加
- Grafana アラートリソースのプロビジョニング

従来のダッシュボードアラートを Grafana アラートに移行する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana アラートを使用ししないワークスペースは、従来のダッシュボードアラート を使用します。新しい Grafana アラートに切り替えるには、 機能にオプトインする必要があります。クラシックダッシュボードアラートと Grafana アラートの違いの詳細については、「Grafana アラートとレガシーダッシュボードアラート」を参照してください。GrafanaLabs は、クラシックダッシュボードアラートがバージョン 11 で削除されることを発表しました。

クラシックダッシュボードアラートを使用している場合、Amazon Managed Grafana には Grafana アラートのプレビューが表示され、アップグレードを完了する前にアップグレードされたアラートを確認および変更できます。

Grafana アラートのプレビュー

移行前に Grafana アラートでアラートをプレビューできます。プレビューでは、移行を変更するアラートを変更できます。

Grafana アラートの移行をプレビューするには

1. Grafana ワークスペースにサインインします。
2. 左側のメニューから [アラート (レガシー)] を選択して、現在のアラートを表示します。
3. 左側のメニューから、[アラートのアップグレード] を選択して、Grafana アラートでアラートを表示します。

   このビューから、移行後のアラートの状態を確認できます。

> **Note**
>
> このビューから、移行に影響する変更を加えることもできます。変更を元に戻すには、[Upgrade] (アップグレード) ページの右上にある [アップグレードをリセット] を選択します。

アラートをアップグレードする準備ができたら、次のセクションを参照してください。

Grafana アラートシステムへの移行

、、 AWS CLIまたは Amazon Managed Grafana API を使用して AWS Management Console、Grafana アラートを使用するように Amazon Managed Grafana インスタンスを設定できます。Grafana アラートのオン/オフの切り替えなど、Amazon Managed Grafana の設定方法の詳細については、「Amazon Managed Grafana ワークスペースを設定する」を参照してください。

Grafana アラートを有効にすると、既存の従来のダッシュボードアラートは Grafana アラートと互換性のある形式で移行されます。移行後のアラートと新しく作成したアラートは、Grafana インスタンスのアラートページに表示されます。Grafana アラートを使用した場合、Grafana が管理するアラートルールは、一致するときに 1 つのアラートではなく、複数の通知を送信します。

従来のダッシュボードアラートと Grafana アラートへの読み取りおよび書き込みアクセス権は、それらが保存されているフォルダのアクセス権限で制御されます。移行中、従来のダッシュボードアラートアクセス許可は、次のように新しいルールアクセス許可と照合されます。

- 元のアラートが属するダッシュボードにアクセス権限が設定されている場合、移行時に元のダッシュボードの権限 (フォルダから継承された権限を含む) 設定で `Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1}` という形式の名前のフォルダが作成されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードがフォルダ内にある場合、ルールはこのフォルダにリンクされ、そのアクセス権限が継承されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードが General (全般) フォルダにある場合、ルールは General (全般) アラートフォルダにリンクされ、ルールにはデフォルトのアクセス権限が継承されます。

> **Note**
>
> Grafana のアラート設定には `NoData` に対する `Keep Last State` オプションがないため、このオプションは従来のルールを移行する際に `NoData` になります。`Error` 処理のためのオプション `Keep Last State` は、新しいオプション `Error` に移行されます。`Keep Last State` の動作に合わせるため、移行時には各アラートルールに対して 1 年間のサイレンスが Amazon Managed Grafana によって自動的に作成されます。

通知チャネルは、適切なルートと受信者が設定されたアラートマネージャー設定に移行されます。デフォルトの通知チャネルは、コンタクトポイントとしてデフォルトのルートに追加されますが、ダッシュボードアラートに関連付けられていない通知チャネルは `autogen-unlinked-channel-recv` ルートに送られます。

制限

- Grafana アラートシステムは、利用可能なすべての Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからアラートルールを取得できない場合があります。
- Grafana アラートと従来のダッシュボードアラートを交互に移行すると、どちらか一方でのみサポートされる機能のデータが失われる可能性があります。

> **Note**
>
> 従来のダッシュボードアラートに戻した場合、Grafana アラートが有効になっている間に新たに作成されたアラートルールやアラート設定に加えられたすべての変更は失われます。

Grafana アラートとレガシーダッシュボードアラート

Grafana 8 で導入された Grafana アラートでは、従来のダッシュボードアラートと比較していくつかの機能強化が行われています。

多次元アラート

単一のアラートルールでシステム全体の可視性を提供するアラートを作成できるようになりました。複数のアラートインスタンスを 1 つのアラートルールから作成できます。例えば、1 つのルールで 1 台のホスト上に存在する複数のマウントポイントのディスク使用量をモニタリングするルールを作成できます。評価エンジンは 1 つのクエリから複数の時系列を返し、各時系列データはラベルセットで識別されます。

ダッシュボード外のアラートの作成

レガシーダッシュボードアラートとは異なり、Grafanaのアラート機能では複数のデータソースからのデータを独自に組み合わせたクエリや式を作成できます。また、ID を使用してダッシュボードとパネルをアラートルールにリンクし、観察中のシステムのトラブルシューティングをすばやく行うことができます。

統合アラートはパネルクエリに直接関連付けられなくなるため、通知メールに画像やクエリ値は含まれませんが、カスタマイズされた通知テンプレートを使用して、クエリ値を表示できます。

Loki および Grafana Mimir アラートルールの作成

Grafana アラートでは、Grafana 管理のアラートと同じ UI と API を使用して Loki および Grafana Mimir のアラートルールを管理することができます。

Prometheus 互換データソースからのアラートの表示と検索

Prometheus 互換データソースのアラートが Grafana アラートセクションに表示されるようになりました。これにより、複数のデータソースでラベルを検索して、関連するアラートをすばやく見つけることができます。

特別な (NoData、Error アラート状態) アラート

Grafana アラートでは、アラート状態の新しい概念が導入されています。アラートルールの評価によって状態 NoData またはエラーが発生すると、Grafana アラートは、次のラベルを含む特別なアラートを生成します。

- `alertname` (状態に応じて DatasourceNoData または DatasourceError の値を伴う)。

- `rulename` (特別なアラートが属するアラートルールの名前)
- `datasource_uid` は、 状態の原因となったデータソースの UID を持ちます。
- 元のアラートルールのすべてのラベルと注釈

これらのアラートは、サイレンスを追加したり、コンタクトポイントにルーティングしたりすることで、通常のアラートと同じ方法で処理できます。

> **Note**
>
> ルールが複数のデータソースを使用し、そのうち 1 つ以上がデータを返さない場合、アラート状態の原因となったデータソースごとに特別なアラートが作成されます。

外部アラートマネージャーの追加

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana を設定して、外部のアラートマネージャーを単一のアラートマネージャーとして使用してすべてのアラートを受信するようにできます。この外部アラートマネージャーは、Grafana 内から設定および管理できます。

> **Note**
>
> Amazon Managed Service for Prometheus を外部アラートマネージャーとして使用することはできません。

アラートマネージャーを追加したら、Grafana アラート UI を使用して、サイレンス、コンタクトポイント、通知ポリシーを管理できます。これらのページのドロップダウンオプションを使用すると、アラートマネージャーを切り替えることができます。

外部アラートマネージャーは、メインの Grafana ナビゲーションメニューから Grafana 設定を使用してデータソースとして設定されます。これにより、Grafana 内から外部アラートマネージャーの連絡先と通知ポリシーを管理できるようになり、以前は URL で外部アラートマネージャーを構成するときに表示されていた HTTP 基本認証資格情報も暗号化されます。

> **Note**
>
> Grafana 9.2 以降、アラートページの [管理] タブからの外部アラートマネージャーの URL 設定は廃止されました。これは、今後のリリースで削除されます。

外部アラートマネージャーを追加するには

1. 左のメインメニューから、[接続] を選択します。
2. `Alertmanager` を検索します。
3. [新しいデータソースの作成] ボタンを選択します。
4. 必要に応じて、ページのフィールドに入力します。

   データソースをプロビジョニングする場合は、`jsonData` フィールドの `handleGrafanaManagedAlerts` フラグを `true` に設定して、Grafana が管理するアラートをこのアラートマネージャーに送信します。

   > **Note**
   >
   > アラートマネージャーの Prometheus、Grafana Mimir、および Cortex 実装がサポートされています。Prometheus の場合、Grafana アラート UI のコンタクトポイントと通知ポリシーは読み取り専用です。

5. [保存してテスト] を選択します。

Grafana アラートリソースのプロビジョニング

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

多くの場合、アラートインフラストラクチャは複雑で、パイプラインの多くの部分が異なる場所に存在することがよくあります。これを複数のチームや組織にまたがってスケールすることは、特に難しい作業です。Grafana アラートプロビジョニングを使用すると、組織に最適な方法でアラートデータを作成、管理、維持できるため、このプロセスが容易になります。

選択できるオプションは 2 つあります。

1. アラートプロビジョニング HTTP API を使用してアラートリソースをプロビジョニングします。

> **Note**
>
> 通常、Grafana UI から API でプロビジョニングされたアラートルールを編集することはできません。
> 編集を有効にするには、 API でアラートルールを作成または編集するときに、x-disable-provenance ヘッダーを次のリクエストに追加します。
>
> ```
> POST /api/v1/provisioning/alert-rules
> PUT /api/v1/provisioning/alert-rules/{UID}
> ```

2. Terraform を使用してアラートリソースをプロビジョニングします。

> **Note**
>
> 現在、Grafana アラートのプロビジョニングでは、アラートルール、コンタクトポイント、ミュートタイミング、テンプレートがサポートされています。ファイルプロビジョニングまたは Terraform を使用してプロビジョニングされたアラートリソースは、それらを作成したソースでのみ編集でき、Grafana やその他のソース内から編集することはできません。例えば、ディスクからのファイルを使用してアラートリソースをプロビジョニングする場合、Terraform または Grafana 内からデータを編集することはできません。

トピック

- Terraform を使用したアラートリソースの作成と管理

- Grafana でのプロビジョニングされたアラートリソースの表示

Terraform を使用したアラートリソースの作成と管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Terraform の Grafana プロバイダーを使用してアラートリソースを管理し、Grafana システムにプロビジョニングします。Terraform プロバイダーによる Grafana アラートのサポートにより、Grafana アラートスタック全体をコードとして簡単に作成、管理、維持できます。

Terraform を使用してアラートリソースを管理する方法の詳細については、Terraform ドキュメントの [Grafana プロバイダー] ドキュメントを参照してください。

Terraform を使用してアラートリソースを作成および管理するには、次のタスクを実行します。

1. プロビジョニング用の API キーを作成します。
2. Terraform プロバイダーを設定します。
3. Terraform でアラートリソースを定義します。
4. `terraform apply` を実行して、アラートリソースをプロビジョニングします。

前提条件

- grafana/grafana [Terraform プロバイダー] 1.27.0 以降があることを確認します。
- また、Grafana 9.1 以降を使用していることを確認します。Grafana バージョン 9 で Amazon Managed Grafana インスタンスを作成した場合、これは当てはまります。

プロビジョニング用の API キーの作成

通常の Grafana API キーを作成して、Grafana で Terraform を認証できます。API キーを使用する既存のツールのほとんどは、新しい Grafana アラートサポートと自動的に連携します。Terraform で使

用するキーの作成の詳細については、[Amazon Managed Grafana オートメーションでの Terraform の使用]を参照してください。

プロビジョニング用の API キーを作成するには

1. CI パイプラインの新しいサービスアカウントを作成します。
2. ロール「アラートルールプロビジョニング API にアクセス」を割り当てます。
3. 新しいサービスアカウントトークンを作成します。
4. Terraform で使用するトークンに名前を付けて保存します。

または、基本認証を使用することもできます。サポートされているすべての認証形式を表示するには、Terraform ドキュメントの [Grafana 認証] を参照してください。

Terraform プロバイダーの設定

Grafana アラートのサポートは、[Grafana Terraform プロバイダー] の一部として含まれています。

以下は、Terraform プロバイダーの設定に使用できる例です。

```
terraform {
    required_providers {
        grafana = {
            source = "grafana/grafana"
            version = ">= 1.28.2"
        }
    }
}

provider "grafana" {
    url = <YOUR_GRAFANA_URL>
    auth = <YOUR_GRAFANA_API_KEY>
}
```

コンタクトポイントとテンプレートのプロビジョニング

コンタクトポイントは、アラートスタックを外部に接続します。これは、外部システムに接続する方法と通知の配信先を Grafana に伝えます。選択できる統合は 15 種類以上あります。この例では、Slack コンタクトポイントを使用します。

コンタクトポイントとテンプレートをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。*<slack-webhook-url>* を Slack Webhook URL (またはその他のコンタクトポイントの詳細) に置き換えます。

   この例では、アラート通知を Slack に送信するコンタクトポイントを作成します。

   ```
   resource "grafana_contact_point" "my_slack_contact_point" {
       name = "Send to My Slack Channel"

       slack {
           url = <slack-webhook-url>
           text = <<EOT
   {{ len .Alerts.Firing }} alerts are firing!

   Alert summaries:
   {{ range .Alerts.Firing }}
   {{ template "Alert Instance Template" . }}
   {{ end }}
   EOT
       }
   }
   ```

2. 通知のテキストをテキストフィールドに入力します。

   `text` フィールドは Go 形式のテンプレートをサポートします。これにより、Terraform で Grafana アラート通知テンプレートを直接管理できます。

3. `terraform apply` コマンドを実行します。
4. Grafana UI に移動し、コンタクトポイントの詳細を確認します。

   UI から Terraform 経由でプロビジョニングされたリソースを編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、コンタクトポイントが正しく機能することを確認します。

> **Note**
>
> 多くのコンタクトポイントで同じテンプレートを再利用できます。上記の例では、`{{ template "Alert Instance Template" . }}` ステートメントを使用して共有テンプレートが埋め込まれています。
> その後、このフラグメントは Terraform で個別に管理できます。

```
resource "grafana_message_template" "my_alert_template" {
    name = "Alert Instance Template"

    template = <<EOT
{{ define "Alert Instance Template" }}
Firing: {{ .Labels.alertname }}
Silence: {{ .SilenceURL }}
{{ end }}
EOT
}
```

通知ポリシーとルーティングのプロビジョニング

通知ポリシーは、アラートインスタンスをどこにルーティングするかではなく、どのようにルーティングするかを Grafana に指示します。ラベルとマッチャーのシステムを使用して、発動アラートを以前に定義したコンタクトポイントに接続します。

通知ポリシーとルーティングをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。

   この例では、アラートは alertname によってグループ化されます。つまり、同じ名前のアラートから送信される通知は、同じ Slack メッセージにグループ化されます。

   特定の通知を別の方法でルーティングする場合は、サブポリシーを追加できます。サブポリシーを使用すると、ラベルマッチングに基づいて異なるアラートにルーティングを適用できます。この例では、ラベル a=b のすべてのアラートにミュートタイミングを適用します。

```
resource "grafana_notification_policy" "my_policy" {
    group_by = ["alertname"]
    contact_point = grafana_contact_point.my_slack_contact_point.name

    group_wait = "45s"
    group_interval = "6m"
    repeat_interval = "3h"

    policy {
        matcher {
            label = "a"
            match = "="
```

```
            value = "b"
        }
        group_by = ["..."]
        contact_point = grafana_contact_point.a_different_contact_point.name
        mute_timings = [grafana_mute_timing.my_mute_timing.name]

        policy {
            matcher {
                label = "sublabel"
                match = "="
                value = "subvalue"
            }
            contact_point = grafana_contact_point.a_third_contact_point.name
            group_by = ["..."]
        }
    }
}
```

2. mute_timings フィールドで、ミュートタイミングを通知ポリシーにリンクします。
3. `terraform apply` コマンドを実行します。
4. Grafana UI に移動し、通知ポリシーの詳細を確認します。

> **Note**
>
> Terraform からプロビジョニングされたリソースを UI から編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、通知ポイントが正しく機能していることを確認します。

ミュートタイミングのプロビジョニング

ミュートタイミングを使用すると、定義された期間のアラート通知をミュートできます。

ミュートタイミングをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。

   この例では、アラート通知は週末にミュートされます。

```
resource "grafana_mute_timing" "my_mute_timing" {
    name = "My Mute Timing"
```

```
    intervals {
        times {
          start = "04:56"
          end = "14:17"
        }
        weekdays = ["saturday", "sunday", "tuesday:thursday"]
        months = ["january:march", "12"]
        years = ["2025:2027"]
    }
}
```

2. `terraform apply` コマンドを実行します。
3. Grafana UI に移動し、ミュートタイミングの詳細を確認します。
4. `mute_timings` フィールドを使用して、通知ポリシーで新しく作成したミュートタイミングを参照します。これにより、通知の一部またはすべてにミュートタイミングが適用されます。

> **Note**
>
> Terraform からプロビジョニングされたリソースを UI から編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、ミュートタイミングが正しく機能していることを確認します。

アラートルールのプロビジョニング

アラートルールを使用すると、あらゆる Grafana データソースに対してアラートを実行できます。これは、既に設定済みのデータソースを使用することも、アラートルールとともに [Terraform でデータソースを定義]することもできます。

アラートルールをプロビジョニングするには

1. クエリするデータソースと、ルールを保存するフォルダを作成します。

   この例では、テスト用 TestData データソースの設定 データソースが使用されます。

   アラートは Grafana の任意のバックエンドデータソースに対して定義できます。

```
resource "grafana_data_source" "testdata_datasource" {
    name = "TestData"
    type = "testdata"
```

```
}

resource "grafana_folder" "rule_folder" {
    title = "My Rule Folder"
}
```

2. アラートルールを定義します。

   アラートルールの詳細については、[Grafana マネージドアラートの作成方法] を参照してください。

3. 1 つ以上のルールを含むルールグループを作成します。

   この例では、grafana_rule_group リソースグループが使用されます。

```
resource "grafana_rule_group" "my_rule_group" {
    name = "My Alert Rules"
    folder_uid = grafana_folder.rule_folder.uid
    interval_seconds = 60
    org_id = 1

    rule {
        name = "My Random Walk Alert"
        condition = "C"
        for = "0s"

        // Query the datasource.
        data {
            ref_id = "A"
            relative_time_range {
                from = 600
                to = 0
            }
            datasource_uid = grafana_data_source.testdata_datasource.uid
            // `model` is a JSON blob that sends datasource-specific data.
            // It's different for every datasource. The alert's query is defined
 here.
            model = jsonencode({
                intervalMs = 1000
                maxDataPoints = 43200
                refId = "A"
            })
        }
```

```
        // The query was configured to obtain data from the last 60 seconds. Let's
 alert on the average value of that series using a Reduce stage.
        data {
            datasource_uid = "__expr__"
            // You can also create a rule in the UI, then GET that rule to obtain
 the JSON.
            // This can be helpful when using more complex reduce expressions.
            model = <<EOT
{"conditions":[{"evaluator":{"params":[0,0],"type":"gt"},"operator":
{"type":"and"},"query":{"params":["A"]},"reducer":{"params":
[],"type":"last"},"type":"avg"}],"datasource":
{"name":"Expression","type":"__expr__","uid":"__expr__"},"expression":"A","hide":false,"int
EOT
            ref_id = "B"
            relative_time_range {
                from = 0
                to = 0
            }
        }

        // Now, let's use a math expression as our threshold.
        // We want to alert when the value of stage "B" above exceeds 70.
        data {
            datasource_uid = "__expr__"
            ref_id = "C"
            relative_time_range {
                from = 0
                to = 0
            }
            model = jsonencode({
                expression = "$B > 70"
                type = "math"
                refId = "C"
            })
        }
    }
}
```

4. Grafana UI に移動し、アラートルールを確認します。

   アラートルールが発射しているかどうかを確認できます。また、アラートルールの各クエリステージを視覚化することもできます。

   アラートが発生すると、Grafana は定義したポリシーを通じて通知をルーティングします。

例えば、コンタクトポイントとして Slack を選択した場合、Grafana の埋め込みアラートマネージャーは Slack にメッセージを自動的に投稿します。

Grafana でのプロビジョニングされたアラートリソースの表示

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートリソースが Grafana で作成されたことを確認できます。

Grafana でプロビジョニングされたリソースを表示するには

1. Grafana インスタンスを開きます。
2. アラートに移動します。
3. アラートルールなど、アラートリソースフォルダをクリックします。

   プロビジョニング済みのリソースには [プロビジョニング済み] というラベルが付けられているため、手動で作成されていないことがわかります。

> **Note**
>
> Grafana からプロビジョニングされたリソースを編集することはできません。リソースプロパティを変更するには、プロビジョニングファイルを変更して Grafana を再起動するか、ホットリロードを実行します。これにより、ファイルが再度プロビジョニングされた場合やホットリロードが実行された場合に上書きされるリソースへの変更を防止します。

アラートを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートの作成と管理に必要な機能と統合を設定します。

トピック

- Grafana マネージドアラートルールを設定する
- データソースマネージドアラートルールを設定する
- 記録ルールを設定する
- コンタクトポイントの設定
- 通知ポリシーを設定する

Grafana マネージドアラートルールを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana マネージドルールは、最も柔軟なアラートルールの種類です。これにより、サポートされている任意のデータソースのデータに対して動作するアラートを作成できます。複数のデータソースをサポートするだけでなく、式を追加してデータを変換し、アラート条件を設定することもできます。アラート通知での画像の使用もサポートされています。これは、単一のルール定義で複数のデータソースからのアラートを許可する唯一のルールタイプです。

1 つのアラートルール (多次元アラートとも呼ばれます) の結果として、複数のアラートインスタンスを作成できます。

Grafana マネージドアラートルールは、ルールを保存するフォルダの編集権限を持つユーザーのみが編集または削除できます。

UI で作成されたアラートリソースを削除すると、そのリソースを取得できなくなります。設定のバックアップを作成し、削除されたアラートリソースを復元できるようにするには、Terraform またはアラート API を使用してアラートリソースを作成します。

以下の手順では、Grafana マネージドアラートルールを作成するプロセスについて説明します。

Grafanaマネージドアラートルールを作成するには、ワークスペース内のアラート作成フローを使用し、以下の手順に従います。

アラートルール名を設定する

1. [アラート] -> [アラートルール] -> [+ 新しいアラートルール] を選択します。
2. アラートルールを識別する名前を入力します。

   この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。

次に、測定するデータと、アラートルールを起動する前に満たす必要がある条件を取得するクエリを定義します。

クエリと条件の定義方法

1. データソースの選択。
2. [オプション] ドロップダウンから、[時間範囲] を指定します。

   > **Note**
   >
   > Grafana アラートは、`now-24hr: now` などの固定相対時間範囲のみをサポートします。
   > 絶対時間範囲 `2021-12-02 00:00:00 to 2021-12-05 23:59:592` または半相対時間範囲 `now/d to: now` はサポートされていません。

3. クエリを追加します。

複数のクエリを追加するには、[クエリの追加] を選択します。

すべてのアラートルールは、デフォルトで Grafana が管理します。データソース管理のアラートルールに切り替える場合は、[データソースマネージドアラートルールに切り替える] をクリックします。

4. 1 つ以上の式を追加します。

   1. 式ごとに [クラシック条件]を選択して単一のアラートルールを作成するか、[Math]、[削減]、[リサンプリング] オプションから選択して、各シリーズに個別のアラートを作成します。

      > **Note**
      >
      > Prometheus を使用する場合、インスタントベクトルと組み込み関数を使用できます。そのため、式を追加する必要はありません。

   2. [プレビュー] を選択して、式が成功したことを確認します。

5. [オプション] 復旧しきい値を追加するには、[カスタム復旧しきい値] のトグルをオンにして、アラートルールが発射を停止するタイミングの値を入力します。

   クエリに追加できる復旧しきい値は 1 つだけで、アラート条件である必要があります。

6. アラート条件を設定するクエリまたは式で [Set as alert condition (アラート条件として設定)] を選択します。

アラートルールの評価を使用して、アラートルールを評価する頻度と状態の変更速度を決定します。

これを行うには、アラートルールが適切な評価グループにあることを確認し、ユースケースに最適な保留期間を設定する必要があります。

アラート評価の動作の設定方法

1. フォルダを選択するか、[+ 新しいフォルダ] を選択します。
2. 評価グループを選択するか、[+ 新しい評価グループ] を選択します。

   新しい評価グループを作成するには、グループの間隔を指定します。

   同じグループ内のすべてのルールは、同じ時間間隔で同時に評価されます。
3. 保留期間を入力します。

保留期間は、アラートルールが発射するまで条件に違反している可能性がある期間です。

条件が満たされると、アラートは保留状態になります。指定された期間、条件がアクティブなままの場合、アラートは発射状態に移行します。アクティブでない場合は[普通]状態に戻ります。

4. 必要に応じて、アラート通知の一時停止を有効にします。

> **Note**
>
> アラートルールの評価を一時停止して、アラートの調整中にノイズの多いアラートが発生しないようにします。一時停止すると、アラートルールの評価が停止し、アラートインスタンスは作成されません。これは、通知の配信を停止するミュートのタイミングとは異なりますが、アラートルールの評価とアラートインスタンスの作成は可能です。
> アラートの調整中にアラートルールの評価を一時停止して、ノイズの多いアラートを防ぐことができます。一時停止すると、アラートルールの評価が停止し、アラートインスタンスは作成されません。これは、通知の配信を停止するミュートのタイミングとは異なりますが、アラートルールの評価とアラートインスタンスの作成は可能です。

5. [Configure no data and error handling] (データなしとエラー処理の設定) で、データがない場合のアラート動作を設定します。

このセクションの後半のガイドラインに従います。

アラートルールにラベルを追加して、発射アラートインスタンスを処理する通知ポリシーを設定します。

ラベルに関係なく、すべてのアラートルールとインスタンスはデフォルトの通知ポリシーと一致します。ネストされたポリシーがない場合、またはネストされたポリシーがアラートルールまたはアラートインスタンスのラベルと一致しない場合、デフォルトの通知ポリシーは一致するポリシーです。

通知を設定するには

1. 通知のルーティング方法を変更するには、ラベルを追加します。

ドロップダウンから既存のキーと値のペアを選択してカスタムラベルを追加するか、新しいキーまたは値を入力して新しいラベルを追加します。

2. アラートインスタンスのルーティング設定をプレビューします。

追加されたラベルに基づいて、アラートインスタンスは表示された通知ポリシーにルーティングされます。

各通知ポリシーを展開して、詳細を表示します。

3. [詳細を表示] を選択して、アラートルーティングの詳細とプレビューを表示します。

注釈を追加して、アラート通知メッセージにアラートに関するコンテキストをさらに提供できます。

注釈は、アラート通知メッセージのアラートに関する詳細情報を提供するメタデータを追加します。例えば、アラートが発生した値や発生したサーバーを示すサマリー注釈を追加します。

注釈を追加するには

1. [オプション] 概要を追加します。

   何が起こったのか、その理由の簡単な概要。

2. (オプション) 説明を追加します。

   アラートルールの動作の説明。

3. [オプション] ランブック URL を追加します。

   アラートのランブックを保持するウェブページ

4. [オプション] カスタム注釈を追加します
5. [オプション] ダッシュボードとパネルリンクを追加します。

   アラートをダッシュボードのパネルにリンクします。

6. [ルールを保存] を選択します。

1 次元ルールと多次元ルール

Grafana マネージドアラートの場合、従来の条件を使用してルールを作成することも、多次元ルールを作成することもできます。

- クラシック条件のルール

  従来の条件式を使用して、条件が満たされたときに単一のアラートを発射するルールを作成します。複数のシリーズを返すクエリの場合、Grafana は各シリーズのアラート状態を追跡しません。

そのため、複数のシリーズでアラート条件が満たされた場合でも、Grafana が送信するアラートは 1 件のみです。

- 多次元ルール

  シリーズごとに個別のアラートを生成するには、多次元ルールを作成します。Math、Reduce、または Resample 式を使用して、多次元ルールを作成します。例:

  - 各クエリの Reduce 式を追加して、選択した時間範囲の値を 1 つの値に集計します (数値データを使用するルールには必要ありません)。
  - ルールの条件を含む Math 式を追加します。クエリまたは縮小式が、ルールが発射されない場合に 0 を、発射すべき場合は正の数を返すのならば不要です。例: B クエリ/式が 70 を超える時に発射する場合は $B > 70。B の値が C の値に 100 を掛けた時に発射する場合は $B < $C * 100。比較対象のクエリの結果に複数のシリーズがある場合、異なるクエリのシリーズが同じラベルを持っているか、一方が他方のサブセットである場合に一致します。

> **Note**
>
> Grafana は、テンプレート変数を使用したアラートクエリはサポートしていません。詳細については、https://community.grafana.com/t/template-variables-are-not-supported-in-alert-queries-while-setting-up-alert/2514 を参照してください。

データなしまたはエラー処理の設定

アラートルールの評価がデータなしまたはエラーを返すアラートルール評価時のアラート行動を設定します。

> **Note**
>
> 評価がデータなしまたはエラーを返したときに発射するように設定されているアラートルールは、評価期間全体が終了したときにのみ発動します。つまり、アラートルール条件に違反した場合にすぐに発射するのではなく、アラートルールは For フィールドとして設定された時間が終了するまで待機してから発車することで、アラートノイズを減らし、一時的なデータ可用性の問題を回避します。

アラートルールの評価でデータなしが返される場合は、次のようにアラートルールの状態を設定できます。

データなし	説明
データなし	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を使用した新しいアラート DatasourceNoData を作成します。
[アラート]	アラートルールの状態を Alerting に設定します。アラートルールは、For フィールドで設定された時間が終わるまで待機してから発射します。
OK	アラートルールの状態を Normal に設定します。

評価でエラーが返された場合は、アラートルールの状態を次のように設定できます。

エラー	説明
エラー	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を使用したアラートインスタンス DatasourceError を作成します。
[アラート]	アラートルールの状態を Alerting に設定します。アラートルールは、For フィールドで設定された時間が終わるまで待機してから発射します。
OK	アラートルールの状態を Normal に設定します。

古いアラートインスタンスを解決する

アラートインスタンスのディメンションまたはシリーズが 2 つの評価間隔でクエリ結果から完全に消えた場合、アラートインスタンスは「古い」と見なされます。

Alerting/NoData/Error 状態にある古いアラートインスタンスは、自動的に Resolved とマークされ、grafana_state_reason 注釈が理由 MissingSeries でアラートインスタンスに追加されます。

パネルからアラートを作成する

任意のパネルタイプからアラートを作成します。つまり、パネル内のクエリを再利用し、それに基づいてアラートを作成できます。

1. ダッシュボードセクションの[ダッシュボード] に移動します。
2. パネルの右上隅で、3 つのドット (楕円) を選択します。
3. ドロップダウンメニューから、[More...] (その他...) を選択し、[新しいアラートルール] を選択します。

これにより、アラートルールフォームが開き、現在のパネルのクエリーに基づいてアラートを設定および作成できます。

データソースマネージドアラートルールを設定する

このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Ruler API が有効になっている外部 Grafana Mimir または Loki インスタンスのアラートルールを作成します。これらは、データソースマネージドアラートルールと呼ばれます。

Note

外部 Grafana Mimir または Loki インスタンスのアラートルールは、エディタまたは管理者ロールを持つユーザーが編集または削除できます。
UI で作成されたアラートリソースを削除すると、そのリソースを取得できなくなります。設定のバックアップを作成し、削除されたアラートリソースを復元できるようにするには、Terraform またはアラート API を使用してアラートリソースを作成します。

前提条件

- Prometheus または Loki データソースへの書き込みアクセス権限があることを確認します。それ以外の場合、Grafana Mimir 管理のアラートルールを作成または更新することはできません。
- Grafana Mimir および Loki データソースの場合、それぞれのサービスを設定して Ruler API を有効にします。
  - Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集するには、他のルールストレージタイプのいずれかを設定します。
  - Grafana Mimir - `/prometheus` プレフィックスを使用します。Prometheus データソースは Grafana Mimir と Prometheus の両方に対応しており、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを想定しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[Manage alerts via alerting UI] (アラート UI 経由でアラートを管理する) チェックボックスをオフにします。

次の手順では、データソースマネージドアラートルールを作成するプロセスについて説明します。

データソース管理のアラートルールを作成するには、ワークスペース内のアラート作成フローを使用し、以下の手順に従います。

アラートルール名の設定方法

1. [アラート] -> [アラートルール] -> [+ 新しいアラートルール] を選択します。
2. アラートルールを識別する名前を入力します。

   この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。

測定するデータと、アラートルールを起動する前に満たす必要がある条件を取得するクエリを定義します。

クエリと条件の定義方法

1. すべてのアラートルールは、デフォルトで Grafana が管理します。データソースマネージドアラートルールに切り替えるには、[データソースマネージドアラートルールに切り替える] を選択します。
2. ドロップダウンリストからデータソースを選択する。

   [高度なデータソースピッカーを開く] を選択して、データソースの追加 (管理者のみ可能) など、その他のオプションを表示することもできます。
3. PromQL または LogQL クエリを入力します。
4. [アラートのプレビュー] を選択します。

アラートルールの評価を使用して、アラートルールを評価する頻度と状態の変更速度を決定します。

アラート評価の動作の設定方法

1. 名前空間を選択するか、[+ 新しい名前空間] を選択します。
2. 評価グループを選択するか、[+ 新しい評価グループ] を選択します。

   新しい評価グループを作成するには、グループの間隔を指定します。

   同じグループ内のすべてのルールは、同じ時間間隔で順番に評価されます。
3. 保留期間を入力します。

   保留期間は、アラートルールが発射するまで条件に違反している可能性がある期間です。

   条件が満たされると、アラートは `Pending` 状態に入ります。指定された期間、条件がアクティブなままの場合、アラートは `Firing` 状態に移行します。アクティブでない場合は `Normal` 状態に戻ります。

アラートルールにラベルを追加して、発射アラートインスタンスを処理する通知ポリシーを設定します。

ラベルに関係なく、すべてのアラートルールとインスタンスはデフォルトの通知ポリシーと一致します。ネストされたポリシーがない場合、またはネストされたポリシーがアラートルールまたはアラートインスタンスのラベルと一致しない場合、デフォルトの通知ポリシーは一致するポリシーです。

通知の設定

- 通知のルーティング方法を変更するには、ラベルを追加します。

  ドロップダウンから既存のキーと値のペアを選択してカスタムラベルを追加するか、新しいキーまたは値を入力して新しいラベルを追加します。

注釈を追加して、アラート通知にアラートに関するコンテキストをさらに提供できます。

注釈は、アラート通知のアラートに関する詳細情報を提供するメタデータを追加します。例えば、アラートが発生した値や発生したサーバーを示す Summary 注釈を追加します。

注釈を追加するには

1. [オプション] 概要を追加します。

   何が起こったのか、その理由の簡単な概要。
2. (オプション) 説明を追加します。

   アラートルールの動作の説明。
3. [オプション] ランブック URL を追加します。

   アラートのランブックを保持するウェブページ
4. [オプション] カスタム注釈を追加します
5. [オプション] ダッシュボードとパネルリンクを追加します。

   アラートをダッシュボードのパネルにリンクします。
6. [ルールを保存] を選択します。

記録ルールを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

外部 Grafana Mimir または Loki インスタンスの記録ルールを作成および管理することができます。記録ルールは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列セットとして保存します。この新しい時系列のクエリは、特にダッシュボードでは、ダッシュボードが更新されるたびに同じ式をクエリするため、より高速になります。

> **Note**
>
> 記録ルールはインスタンスルールとして実行され、10 秒ごとに実行されます。

前提条件

- Prometheus または Loki データソースへの書き込みアクセス権限があることを確認します。データソースでアラートルールを作成または更新します。
- Grafana Mimir および Loki データソースの場合、それぞれのサービスを設定して Ruler API を有効にします。
  - Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集する場合、他のストレージタイプを設定してください。
  - Grafana Mimir – `/prometheus` プレフィックスを使用します。Prometheus データソースは Grafana Mimir と Prometheus の両方に対応しており、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを想定しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[アラート UI 経由でアラートを管理する] チェックボックスをオフにします。

記録ルールを作成するには

1. Grafana コンソールの Grafana メニューで、[アラート]、[アラートルール] を選択します。
2. [新しい記録ルール] を選択します。
3. ルール名を設定します。

   記録ルール名は Prometheus メトリクス名で、空白を含まない必要があります。

4. クエリの定義

   - Loki または Prometheus データソースを選択します。
   - クエリを入力します。

5. 名前空間とグループを追加します。

   - [名前空間] ドロップダウンから、既存のルール名前空間を選択するか、新しいルール名前空間を追加します。名前空間には 1 つ以上のルールグループを含めることができます。これは、組織的な目的を持たせるためのみに使用します。
   - [グループ] ドロップダウンから、選択した名前空間内の既存のグループを選択するか、新しいものを作成します。新しく作成されたルールは、グループの末尾に追加されます。グループ内のルールは、同じ評価時間で一定の間隔で順番に実行されます。

6. ラベルを追加します。

   - ドロップダウンから既存のキーと値のペアを選択してカスタムラベルを追加するか、新しいキーまたは値を入力して新しいラベルを追加します。

7. [ルールを保存] を選択してルールを保存するか、[ルールを保存して終了] を選択してルールを保存し、アラートページに戻ります。

コンタクトポイントの設定

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントを使用して、アラートルールが発生したときにコンタクトに通知する方法を定義します。

Note

Grafana で管理されるアラートの通知先 (コンタクトポイント) は、作成および編集が可能です。データソース管理アラートのコンタクトポイントは読み取り専用です。

コンタクトポイント (通知先) の使用

次の手順では、通知先 (コンタクトポイント) を追加、編集、削除、テストする方法について説明します。

コンタクトポイントの追加方法

1. 左側のメニューで、[アラート] を選択します。
2. [コンタクトポイント] を選択します。
3. [アラートマネージャーの選択] ドロップダウンから、アラートマネージャーを選択します。デフォルトでは、Grafana アラートマネージャーが選択されています。
4. [コンタクトポイント] タブで、 [+ コンタクトポイントの追加] を選択します。
5. コンタクトポイントの [名前] を入力します。
6. [統合] からタイプを選択し、そのタイプに基づいて必須フィールドを入力します。例えば、Slack を選択した場合は、Slack チャネルと通知先となるユーザーを入力します。
7. 選択したコンタクトポイントで利用可能な場合は、任意のオプション設定を選択して追加の設定を指定します。
8. [通知設定]で[解決済みメッセージの無効化]を選択すると、アラートが解決した際に通知を受け取らないようにすることができます (任意)。
9. 別のコンタクトポイント統合を追加するには、[コンタクトポイント統合を追加] を選択し、必要なコンタクトポイントタイプごとにステップを繰り返します。
10. 変更内容を保存します。

コンタクトポイントの編集方法

1. 左側のメニューで、[アラート] を選択します。
2. [コンタクトポイント]を選択すると、現在設定されているコンタクトポイントのリストが表示されます。
3. 編集する連絡先を選択し、[編集] を選択します。
4. 連絡先を更新し、変更を保存します。

使用されていないコンタクトポイントは、通知ポリシーで削除できます。

連絡先の削除方法

1. 左側のメニューで、[アラート] を選択します。

2. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
3. コンタクトポイント で、削除するコンタクトポイントを選択し、[詳細]、[削除] を選択します。
4. 確認ダイアログボックスで、[はい、削除] を選択します。

> **Note**
>
> コンタクトポイントが通知ポリシーで使用されている場合、コンタクトポイントを削除する前に、通知ポリシーを削除するか、別のコンタクトポイントを使用するように編集する必要があります。

コンタクトポイントを作成したら、テスト通知を送信して、正しく設定されていることを確認します。

テスト通知の送信方法

1. 左側のメニューで、[アラート] を選択します。
2. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
3. コンタクトポイント で、テストするコンタクトポイントを選択し、[編集] を選択します。必要に応じて、新しいコンタクトポイントを作成することもできます。
4. [テスト] を選択して、コンタクトポイントテストダイアログを開きます。
5. 事前定義されたテスト通知を送信するか、[カスタム] を選択してテスト通知に独自の注釈とラベルを追加するかを選択します。
6. [テスト通知を送信]を選択して、指定されたコンタクトポイント宛にアラートをテストします。

コンタクトポイント統合の設定

アラートルールの発射時に通知を受信するための優先通信チャネルを選択するように Grafana でコンタクトポイント統合を設定します。各統合には、独自の設定オプションとセットアッププロセスがあります。ほとんどの場合、これには API キーまたは Webhook URL の提供が含まれます。

設定が完了すると、コンタクトポイントの一部として統合を使用して、アラートの状態が変化するたびに通知を受け取ることができます。このセクションでは、PagerDuty を例として使用して統合を設

定する基本的な手順について説明します。これにより、リアルタイムアラートの受信を開始し、モニタリングデータを常に把握できます。

サポートされている統合のリスト

次の表に、Grafana でサポートされているコンタクトポイントのタイプを示します。

名前	タイプ
Amazon SNS	`sns`
OpsGenie	`opsgenie`
Pager Duty	`pagerduty`
Slack	`slack`
VictorOps	`victorops`

アラート用の PagerDuty の設定

PagerDuty を設定するには、統合キーを指定する必要があります。次の詳細情報を入力します：

設定	説明		
統合キー	PagerDuty の統合キー		
緊急度	動的通知のレベル。デフォルトは `critical` です。		
カスタムの詳細	イベントに関するその他の詳細		

`CustomDetails` フィールドは、任意のキーと値のペアを含むオブジェクトです。ユーザー定義の詳細が、デフォルトで使用されている詳細とマージされます。

`CustomDetails` のデフォルト値は次のとおりです。

```
{
 "firing":       `{{ template "__text_alert_list" .Alerts.Firing }}`,
 "resolved":     `{{ template "__text_alert_list" .Alerts.Resolved }}`,
 "num_firing":   `{{ .Alerts.Firing | len }}`,
 "num_resolved": `{{ .Alerts.Resolved | len }}`,
}
```

キーが重複している場合、ユーザー定義の詳細がデフォルトのキーを上書きします。

通知ポリシーを設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知ポリシーでは、アラートがコンタクトポイント (通知先) にどのように配信されるかを指定します。

ポリシーにはツリー構造があり、1 つ以上のネストされたポリシーを各ポリシーに含めることができます。デフォルトポリシーを除く各ポリシーは、特定のアラートラベルと一致させることもできます。

各アラートは、デフォルトのポリシーによって評価され、それに続き、ネストされた各ポリシーによって評価されます。

ネストされたポリシーの `Continue matching subsequent sibling nodes` オプションを有効にすると、1 つ以上一致があった後も評価が続行されます。子ポリシーのいずれとも一致しない場合、親ポリシーの設定とコンタクトポイント (通知先) の情報によってアラートの動作が制御されます。デフォルトのポリシーは、ネストされたポリシーと一致しないアラートを管理します。

これらの通知ポリシーの詳細については、「通知」を参照してください。

以下に、通知ポリシーを作成および管理するための手順を紹介します。

デフォルトの通知ポリシーの編集方法

1. 左側のメニューで、[アラート] を選択します。
2. [通知ポリシー] を選択します。
3. [アラートマネージャーの選択] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [デフォルトポリシー] セクションで、 [...]、[編集] を続けて選択します。
5. [デフォルトのコンタクトポイント] で、アラートルールが特定のポリシーと一致しない場合に通知を送信するコンタクトポイント (通知先) を更新します。
6. [グループ化] で、アラートをグループ化するラベルを選択します。このポリシーに複数のアラートが一致すると、それらのアラートはこれらのラベルでグループ化されます。グループごとに通知が送信されます。フィールドが空 (デフォルト) の場合、すべての通知は 1 つのグループで送信されます。特別なラベル `...` を使用して、すべてのラベルでアラートをグループ化します (グループ化を効果的に無効にします) 。
7. [タイミングオプション] で、以下のいずれかを選択します。
   - [グループ待機] – 同じグループのアラートをバッファリングしてから、最初の通知を送信するまでの待機時間。デフォルト値は 30 秒です。
   - [グループ間隔] – グループ内の 2 つの通知の最小時間間隔。デフォルト値は 5 分です。
   - [繰り返し間隔] – グループに新しいアラートが追加されなかった場合に、通知を再送信するまでの最小時間間隔。デフォルトは 4 時間です。
8. [保存] を選択して変更を保存します。

新しい通知ポリシーを作成するには、ツリー構造を考慮する必要があります。ツリーのトランクに作成された新しいポリシー (デフォルトポリシー) は、ツリーブランチです。各ブランチには、独自のネストされたポリシーを含めることができます。そのため、デフォルトのポリシーまたは既にネストされたポリシーのいずれかで、常に新しいネストされたポリシーを追加します。

新しいネストされたポリシーを追加するには

1. 左側のメニューで、[アラート] を選択します。
2. [通知ポリシー] を選択します。
3. [アラートマネージャーの選択] ドロップダウンから、編集するアラートマネージャーを選択します。

4. 最上位の特定のポリシーを追加するには、[特定のルーティング] セクション (デフォルトポリシー、または新しいネストされたポリシーを追加する既存の別のポリシーのいずれか) に移動し、[+ 新しいネストされたポリシー] を選択します。
5. 一致するラベルセクションで、アラートラベルを照合するための 1 つ以上のルールを追加します。
6. [コンタクトポイント] ドロップダウンで、アラートがこの特定のポリシーのみに一致し、ネストされたポリシーのいずれにも一致しない場合に通知を送信するコンタクトポイントを選択します。
7. オプションで、アラートが現在のポリシーと一致した後でも、[後続の兄弟ノードを照合し続ける] を有効にして、兄弟ポリシーの照合を続行します。このオプションを有効にすると、1 つのアラートに対して複数の通知を取得できます。
8. オプションで、[グループ化の上書き] を有効にして、デフォルトポリシーと同じグループ化を指定します。オプションが有効になっていない場合、デフォルトのポリシーグループが使用されます。
9. オプションで、[一般的なタイミングの上書き] を有効にして、グループ通知ポリシーで設定されているタイミングオプションを上書きします。
10. [Save policy(ポリシーの保存] を選択して変更を保存します。

ネストされたポリシーを編集するには

1. 左側のメニューで、[アラート] を選択します。
2. [通知ポリシー] を選択します。
3. 編集するポリシーを選択したら、[...]、[編集] を続けて選択します。
4. 変更を加えます (ネストされたポリシーを追加する場合と同様)。
5. 変更内容を保存します。

ポリシーの検索

ポリシーのツリー内では、ラベルマッチャーまたはコンタクトポイントで検索できます。

- コンタクトポイントで検索するには、[コンタクトポイントで検索] フィールドにコンタクトポイントの部分的な名前または完全な名前を入力します。そのコンタクトポイントを使用するポリシーは、ユーザーインターフェイスで強調表示されます。

- ラベルで検索するには、[ラベルマッチャーによる検索] 入力フィールドに有効なラベルマッチャーを入力します。複数のマッチャーをカンマで区切って入力できます。有効なマッチャー入力の例は `severity=high, region=~EMEA|NA` です。

> **Note**
>
> ラベルで検索する場合、一致するすべてのポリシーは完全一致になります。部分一致と正規表現形式の一致はサポートされていません。

アラートの管理

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルール、コンタクトポイント、通知ポリシーを設定したら、Grafana アラートを使用してアラートを実際に管理できます。

トピック

- 通知のカスタマイズ
- コンタクトポイントの管理
- アラート通知のサイレンシング
- アラートルールの表示とフィルタリング
- ミュートタイミング
- アラートルールの状態と正常性を表示する
- アラートグループ別に表示およびフィルタリングする
- 通知エラーの表示

通知のカスタマイズ

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知テンプレートを使用して、通知をカスタマイズします。

通知テンプレートを使用して、通知内のメッセージのタイトル、メッセージ、形式を変更できます。

通知テンプレートは、Amazon SNS や Slack などの特定のコンタクトポイント統合には関連付けられていません。ただし、異なるコンタクトポイント統合用に個別の通知テンプレートを作成することもできます。

通知テンプレートを使用すると、次のことができます。

- 概要、説明、ラベルと注釈、値、リンクなど、通知内の情報を追加、削除、または並べ替える
- テキストを太字と斜体で書式設定し、改行を追加または削除する

通知テンプレートを次の目的には使用できません。

- Slack や Microsoft Teams などのインスタントメッセージングサービスで通知のデザインを変更する

トピック

- Go のテンプレート言語の使用
- 通知テンプレートの作成
- 通知テンプレートの使用
- テンプレートリファレンス

Go のテンプレート言語の使用

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Go のテンプレート言語である text/template で通知テンプレートを書き込みます。

このセクションでは、Go のテンプレート言語の概要と、text/template でのテンプレートの作成について説明します。

dot

text/template には dot と呼ばれる特別なカーソルがあり、. として記述されます。このカーソルは、テンプレート内の使用場所に応じて値が変化する変数と考えることができます。例えば、通知テンプレートの開始時に . は、`Alerts`、`Status`、`GroupLabels`、`CommonLabels`、`CommonAnnotations` および `ExternalURL` を含む多数のフィールドを含む `ExtendedData` オブジェクトを参照します。ただし、dot は、リスト上の `range` で使用される場合、`with` 内で使用される場合、または他のテンプレートで使用する機能テンプレートを記述する場合、他の何かを参照する場合があります。これの例は 通知テンプレートの作成 で、すべてのデータと関数は テンプレートリファレンス で確認できます。

開始と終了のタグ

テキスト/テンプレートでは、テンプレートが変数を印刷するか、if ステートメントなどのコントロール構造を実行するかに関係なく、テンプレートは {{で始まり、}} で終わります。これは、変数の出力に {{ と }} を使用し、制御構造に {% と %} を使用する Jinja などの他のテンプレート言語とは異なります。

印刷

何かの値を印刷するには、{{ と }} を使用します。dot の値、dot のフィールド、関数の結果、[変数]の値を印刷できます。例えば、dot が `ExtendedData` を参照する `Alerts` フィールドを印刷するには、次のように記述します。

```
{{ .Alerts }}
```

アラートを反復処理する

アラートに関するすべての情報ではなく、各アラートのラベルのみを印刷するには、range を使用して ExtendedData でアラートを反復処理できます。

```
{{ range .Alerts }}
{{ .Labels }}
{{ end }}
```

範囲内の dot は、ExtendedData ではなく Alert を参照します。{{ .Labels }} を使用して、各アラートのラベルを印刷できます。これは、アラートのリスト内の現在のアラートを参照するように {{ range .Alerts }} が dot を変えるために機能します。範囲が終了すると、dot は範囲の開始前に存在していた値にリセットされます。この例では ExtendedData です。

```
{{ range .Alerts }}
{{ .Labels }}
{{ end }}
{{/* does not work, .Labels does not exist here */}}
{{ .Labels }}
{{/* works, cursor was reset */}}
{{ .Status }}
```

注釈とラベルを反復処理する

各アラートのラベルを The name of the label is $name, and the value is $value の形式で印刷するテンプレートを書きましょう。ここで、$name と $value には各ラベルの名前と値が含まれます。

前の例と同様に、.Alerts の範囲を使用して のアラートを繰り返し、dot がアラートのリスト内の現在のアラートを参照するようにし、ソートされたラベルで 2 番目の範囲を使用して、dot が現在のラベルを参照するように 2 回目に更新されます。2 番目の範囲内では、 .Name と .Value を使用して各ラベルの名前と値を出力します。

```
{{ range .Alerts }}
{{ range .Labels.SortedPairs }}
The name of the label is {{ .Name }}, and the value is {{ .Value }}
{{ end }}
{{ range .Annotations.SortedPairs }}
```

```
The name of the annotation is {{ .Name }}, and the value is {{ .Value }}
{{ end }}
{{ end }}
```

インデックス関数

特定の注釈またはラベルを出力するには、 index 関数を使用します。

```
{{ range .Alerts }}
The name of the alert is {{ index .Labels "alertname" }}
{{ end }}
```

If ステートメント

テンプレートでは if ステートメントを使用できます。例えば、.Alerts にアラートがない場合に There are no alerts を出力するには、次のように記述します。

```
{{ if .Alerts }}
There are alerts
{{ else }}
There are no alerts
{{ end }}
```

With

With は if ステートメントと似ていますが、if ステートメントとは異なり、with は dot を更新して with の値を参照します。

```
{{ with .Alerts }}
There are {{ len . }} alert(s)
{{ else }}
There are no alerts
{{ end }}
```

[変数]

text/template の変数は、テンプレート内に作成する必要があります。例えば、$variable という変数を現在の dot の値で作成するには、次のように記述します。

```
{{ $variable := . }}
```

範囲内の $variable または with を使用でき、これらは変数が定義された時点の dot の値を参照します。dot の現在の値は参照されません。

例えば、2 番目の範囲で {{ .Labels }} を使用するテンプレートを記述することはできません。ここでは、現在のアラートではなく、現在のラベルが dot で参照されるためです。

```
{{ range .Alerts }}
{{ range .Labels.SortedPairs }}
{{ .Name }} = {{ .Value }}
{{/* does not work because in the second range . is a label not an alert */}}
There are {{ len .Labels }}
{{ end }}
{{ end }}
```

これを修正するには、最初の範囲と 2 番目の範囲の前に $alert という変数を定義します。

```
{{ range .Alerts }}
{{ $alert := . }}
{{ range .Labels.SortedPairs }}
{{ .Name }} = {{ .Value }}
{{/* works because $alert refers to the value of dot inside the first range */}}
There are {{ len $alert.Labels }}
{{ end }}
{{ end }}
```

インデックスを含む範囲

範囲の開始時にインデックス変数と値変数を定義することで、範囲内の各アラートのインデックスを取得できます。

```
{{ $num_alerts := len .Alerts }}
{{ range $index, $alert := .Alerts }}
This is alert {{ $index }} out of {{ $num_alerts }}
{{ end }}
```

テンプレートの定義

define とテンプレートの名前を二重引用符で囲むことで、他のテンプレート内で使用できるテンプレートを定義できます。__subject、__text_values_list、__text_alert_list、default.title、default.messa などのデフォルトテンプレートを含め、他のテンプレートと同じ名前のテンプレートを設定しないで

ください。デフォルトテンプレートと同じ名前のテンプレート、または別の通知テンプレート内のテンプレートと同じ名前で作成されている場合、Grafana はいずれかのテンプレートを使用することになり、混乱が生じます。Grafana は、同じ名前のテンプレートが 2 つ以上ある場合、エラーメッセージを返しません。

```
{{ define "print_labels" }}
{{ end }}
```

テンプレートの実行

`template`、二重引用符で囲まれたテンプレートの名前、およびテンプレートに渡されるカーソルを使用して、テンプレート内で定義されたテンプレートを実行できます。

```
{{ template "print_labels" . }}
```

テンプレートにデータを渡す

テンプレート内では、dot はテンプレートに渡される値を参照します。

例えば、テンプレートに発射アラートのリストが渡された場合、dot は発射アラートのリストを参照します。

```
{{ template "print_alerts" .Alerts }}
```

テンプレートにアラートのソート済みラベルが渡された場合、dot はソート済みラベルのリストを参照します。

```
{{ template "print_labels" .SortedLabels }}
```

これは、再利用可能なテンプレートを記述する場合に役立ちます。例えば、すべてのアラートを出力するには、次のように記述します。

```
{{ template "print_alerts" .Alerts }}
```

次に、発射アラートのみを出力するには、次のように記述します。

```
{{ template "print_alerts" .Alerts.Firing }}
```

これは、.Alerts と .Alerts.Firing の両方がアラートのリストであるために機能します。

```
{{ define "print_alerts" }}
{{ range . }}
{{ template "print_labels" .SortedLabels }}
{{ end }}
{{ end }}
```

コメント

{{/* および */}} を使用してコメントを追加できます。

```
{{/* This is a comment */}}
```

コメントが改行を追加しないようにするには、以下を使用します。

```
{{- /* This is a comment with no leading or trailing line breaks */ -}}
```

インデント

タブとスペースの両方のインデントと改行を使用して、テンプレートをより読みやすくすることができます。

```
{{ range .Alerts }}
  {{ range .Labels.SortedPairs }}
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

ただし、テンプレートのインデントはテキストにも適用されます。次に、削除する方法を説明します。

スペースと改行を削除する

text/template では、 {{- と -}} を使用して、先頭と末尾のスペースと改行を削除します。

例えば、インデントと改行を使用してテンプレートをより読みやすくする場合です。

```
{{ range .Alerts }}
  {{ range .Labels.SortedPairs }}
```

```
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

インデントと改行はテキストにも表示されます。

```
    alertname = "Test"

    grafana_folder = "Test alerts"
```

各範囲の先頭の }} を -}} に変更することで、テキストからインデントと改行を削除できます。

```
{{ range .Alerts -}}
  {{ range .Labels.SortedPairs -}}
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

これにより、テンプレートのインデントと改行がテキストから取り除かれます。

```
alertname = "Test"
grafana_folder = "Test alerts"
```

通知テンプレートの作成

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントに送信するため、再利用可能な通知テンプレートを作成できます。

通知テンプレートには 1 つ以上のテンプレートを追加できます。

通知テンプレート名は、一意である必要があります。同じ通知テンプレートまたは異なる通知テンプレートに、同じ名前のテンプレートを 2 つ使用することはできませ

ん。__subject、__text_values_list、__text_alert_list、default.title、default.messa
などのデフォルトテンプレートと同じ名前のテンプレートを定義しないでください。

[コンタクトポイント] タブには、通知テンプレートのリストが表示されます。

通知テンプレートの作成

通知テンプレートを作成するには

1. [アラート]、[コンタクトポイント] を続けて選択します。
2. [通知テンプレート] タブを選択し、[+ 通知テンプレートの追加] を選択します。
3. email.subject などの通知テンプレートの名前を選択します。
4. テンプレートの内容をコンテンツフィールドに書き込みます。

   例:

   ```
   {{ if .Alerts.Firing -}}
      {{ len .Alerts.Firing }} firing alerts
      {{ end }}
      {{ if .Alerts.Resolved -}}
      {{ len .Alerts.Resolved }} resolved alerts
      {{ end }}
   ```

5. 変更内容を保存します。

   {{ define "email.subject" }} (email.subject はテンプレートの名前) と {{ end }} はコンテンツの先頭と末尾に自動的に追加されます。

複数のテンプレートを含む通知テンプレートを作成するには

1. [アラート]、[コンタクトポイント] を続けて選択します。
2. [通知テンプレート] タブを選択し、[+ 通知テンプレートの追加] を選択します。
3. 通知テンプレート全体の名前を入力します。例えば、email と指定します。
4. 各テンプレートの最初と最後に {{ define "*name-of-template*" }} と {{ end }} を含む各テンプレートをコンテンツフィールドに書き込みます。email.subject や email.message など、通知テンプレート内の各テンプレートにわかりやすい名前を使用できます。この場合、上記で入力した通知テンプレートの名前は再度使用しないでください。

   後のセクションでは、作成するテンプレートの詳細な例を示します。

5. 保存をクリックします。

通知テンプレートをプレビューする

コンタクトポイントで使用する前に通知テンプレートの外観をプレビューすると、作成するテンプレートの結果を理解しやすくなり、テンプレートを保存する前にエラーを修正することができます。

> **Note**
>
> 通知プレビューは Grafana アラートマネージャーでのみ使用できます。

通知テンプレートをプレビューするには

1. [アラート]、[コンタクトポイント] を続けて選択します。
2. [通知テンプレート] タブを選択し、[+ 通知テンプレートの追加] を選択するか、既存のテンプレートを編集します。
3. テンプレートコンテンツを追加または更新します。

   デフォルトのデータが提供され、アラートデータとアラートインスタンスを追加または編集できます。アラートデータは、ペイロードデータウィンドウ自体に直接追加するか、[アラートインスタンスの選択] または[カスタムアラートの追加] をクリックします。
4. [オプション] 既存のアラートインスタンスからアラートデータを追加するには:

   1. [アラートインスタンスの選択] を選択します。
   2. アラートインスタンスにカーソルを合わせると、各アラートインスタンスに関する詳細情報が表示されます。
   3. [Confirm] (確認) を選択して、アラートインスタンスをペイロードに追加します。
5. [オプション] アラートデータエディタを使用してアラートデータを追加するには、[カスタムデータの追加] を選択します。

   1. 注釈/カスタムラベルを追加するか、ダッシュボードまたはパネルを設定します。
   2. 通知に発射アラートを追加するか、解決済みアラートを追加するかに応じて、発射または解決を切り替えます。
   3. [アラートデータの追加] を選択します。
   4. [プレビューの更新] を選択して、テンプレートの内容と対応するペイロードデータを確認します。

テンプレートにエラーがある場合はプレビューに表示され、保存する前に修正できます。

6. 変更内容を保存します。

メッセージの件名のテンプレートの作成

この例では、発射アラートと解決済みアラートの数を含む E メールの件名のテンプレートを作成します。

```
1 firing alerts, 0 resolved alerts
```

E メールの件名のテンプレートを作成するには

1. email.subject というテンプレートを次の内容で作成します。

   ```
   {{ define "email.subject" }}
   {{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
   {{ end }}
   ```

2. テンプレートは、キーワード template を使用して件名フィールドに配置して、コンタクトポイント統合を作成するときに使用します。

   ```
   {{ template "email.subject" . }}
   ```

E メールのメッセージ用のテンプレート作成

この例では、すべての発射アラートと解決済みアラートの概要を含む E メールのメッセージ用のテンプレートを作成します。

```
There are 2 firing alerts, and 1 resolved alerts

Firing alerts:

- alertname=Test 1 grafana_folder=GrafanaCloud has value(s) B=1
- alertname=Test 2 grafana_folder=GrafanaCloud has value(s) B=2

Resolved alerts:
```

```
- alertname=Test 3 grafana_folder=GrafanaCloud has value(s) B=0
```

E メールのメッセージ用のテンプレートを作成するには

1. email という名前の通知テンプレートを、コンテンツに email.message_alert と email.message という 2 つのテンプレートで作成します。

   email.message_alert テンプレートは、email.message テンプレートに E メールの構造が含まれている間、各発射アラートと解決済みアラートのラベルと値を印刷するために使用されます。

   ```
   {{- define "email.message_alert" -}}
   {{- range .Labels.SortedPairs }}{{ .Name }}={{ .Value }} {{ end }} has value(s)
   {{- range $k, $v := .Values }} {{ $k }}={{ $v }}{{ end }}
   {{- end -}}

   {{ define "email.message" }}
   There are {{ len .Alerts.Firing }} firing alerts, and {{ len .Alerts.Resolved }}
    resolved alerts

   {{ if .Alerts.Firing -}}
   Firing alerts:
   {{- range .Alerts.Firing }}
   - {{ template "email.message_alert" . }}
   {{- end }}
   {{- end }}

   {{ if .Alerts.Resolved -}}
   Resolved alerts:
   {{- range .Alerts.Resolved }}
   - {{ template "email.message_alert" . }}
   {{- end }}
   {{- end }}

   {{ end }}
   ```

2. テンプレートは、キーワード template を使用してテキスト本文フィールドに配置して、コンタクトポイント統合を作成するときに使用します。

   ```
   {{ template "email.message" . }}
   ```

Slack メッセージのタイトルのテンプレート作成

この例では、発射アラートと解決済みアラートの数を含む Slack メッセージのタイトルのテンプレートを作成します。

```
1 firing alerts, 0 resolved alerts
```

Slack メッセージのタイトルのテンプレートを作成するには

1. `slack.title` というテンプレートを次の内容で作成します。

   ```
   {{ define "slack.title" }}
   {{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
   {{ end }}
   ```

2. コンタクトポイント統合のタイトルフィールドからテンプレートを実行します。

   ```
   {{ template "slack.title" . }}
   ```

Slack メッセージのコンテンツのテンプレート作成

すべての発射アラートと解決済みアラートの説明 (ラベル、注釈、ダッシュボード URL など) を含む Slack メッセージのコンテンツのテンプレートを作成します。

> **Note**
>
> このテンプレートは Grafana マネージドアラート専用です。データソースマネージドアラートに テンプレートを使用するには、DashboardURL と SilenceURL への参照を削除します。Prometheus 通知の設定に関する詳細については、「通知に関する Prometheus ドキュメント」を参照してください。

```
1 firing alerts:

[firing] Test1
Labels:
- alertname: Test1
- grafana_folder: GrafanaCloud
Annotations:
- description: This is a test alert
```

```
Go to dashboard: https://example.com/d/dlhdLqF4z?orgId=1

1 resolved alerts:

[firing] Test2
Labels:
- alertname: Test2
- grafana_folder: GrafanaCloud
Annotations:
- description: This is another test alert
Go to dashboard: https://example.com/d/dlhdLqF4z?orgId=1
```

Slack メッセージのコンテンツのテンプレートを作成するには

1. コンテンツに slack.print_alert とslack.message の 2 つのテンプレートを使用して、slack という名前のテンプレートを作成します。

   slack.print_alert テンプレートはラベル、注釈、および DashboardURL の印刷に使用され、slack.message テンプレートには通知の構造が含まれます。

   ```
   {{ define "slack.print_alert" -}}
   [{{.Status}}] {{ .Labels.alertname }}
   Labels:
   {{ range .Labels.SortedPairs -}}
   - {{ .Name }}: {{ .Value }}
   {{ end -}}
   {{ if .Annotations -}}
   Annotations:
   {{ range .Annotations.SortedPairs -}}
   - {{ .Name }}: {{ .Value }}
   {{ end -}}
   {{ end -}}
   {{ if .DashboardURL -}}
     Go to dashboard: {{ .DashboardURL }}
   {{- end }}
   {{- end }}

   {{ define "slack.message" -}}
   {{ if .Alerts.Firing -}}
   {{ len .Alerts.Firing }} firing alerts:
   {{ range .Alerts.Firing }}
   {{ template "slack.print_alert" . }}
   {{ end -}}
   ```

```
{{ end }}
{{ if .Alerts.Resolved -}}
{{ len .Alerts.Resolved }} resolved alerts:
{{ range .Alerts.Resolved }}
{{ template "slack.print_alert" .}}
{{ end -}}
{{ end }}
{{- end }}
```

2. コンタクトポイント統合のテキスト本文フィールドからテンプレートを実行します。

```
{{ template "slack.message" . }}
```

共有テンプレートを使用して E メールと Slack の両方にテンプレートを作成する

E メールや Slack など、コンタクトポイントごとに個別の通知テンプレートを作成する代わりに、同じテンプレートを共有できます。

例えば、この件名の E メールを送信し、このタイトル `1 firing alerts, 0 resolved alerts` の Slack メッセージを送信する場合は、共有テンプレートを作成できます。

共有テンプレートを作成するには

1. `common.subject_title` というテンプレートを次の内容で作成します。

```
{{ define "common.subject_title" }}
{{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
{{ end }}
```

2. E メールの場合は、E メールコンタクトポイント統合の件名フィールドからテンプレートを実行します。

```
{{ template "common.subject_title" . }}
```

3. Slack の場合は、Slack コンタクトポイント統合のタイトルフィールドからテンプレートを実行します。

```
{{ template "common.subject_title" . }}
```

通知テンプレートの使用

コンタクトポイントのテンプレートを使用して通知をカスタマイズします。

コンタクトポイントの作成時にテンプレートを使用するには

1. [アラート] メニューから、[コンタクトポイント] タブを選択して、既存のコンタクトポイントのリストを表示します。
2. [New] (新規作成) を選択します。または、[編集] アイコンを選択して、既存のコンタクトポイントを編集することもできます。
3. [Message] (メッセージ) や [Subject] (件名) などのフィールドに、使用するテンプレートを入力します。テンプレートを入力するには、`{{ template "template_name" . }}` フォームを使用し、 *template_name* を使用するテンプレートの名前に置き換えます。
4. [コンタクトポイントの保存] を選択します。

テンプレートリファレンス

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このセクションでは、テンプレートを作成するためのリファレンス情報を提供します。

アラート (型)

アラート型には、次のデータがあります。

名前	[Kind] (種類)	説明	例
ステータス	文字列	`firing` または `resolved`	`{{ .Status }}`

名前	[Kind] (種類)	説明	例
ラベル	KeyValue	アラートに付与された一連のラベル。	`{{ .Labels }}`
注釈	KeyValue	アラートに付与された一連の注釈。	`{{ .Annotations }}`
値	KeyValue	クラシック条件を含むすべての式の値	`{{ .Values }}`
StartsAt	time.Time	アラートが発行された時刻。	`{{ .StartsAt }}`
EndsAt	time.Time	アラートの終了時刻がわかっている場合にのみ設定されます。それ以外の場合、最後にアラートを受信してから設定可能なタイムアウト期間が適用されます。	`{{ .EndsAt }}`
GeneratorURL	文字列	Grafana または外部アラートマネージャーへのリンク。	`{{ .GeneratorURL }}`
SilenceURL	文字列	アラートをサイレンスするリンク (このアラートのラベルがあらかじめ入力されています)。Grafana 管理のアラートのみ。	`{{ .SilenceURL}}`

名前	[Kind] (種類)	説明	例
DashboardURL	文字列	Grafana ダッシュボードへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。	`{{ .DashboardURL }}`
PanelURL	文字列	Grafana ダッシュボードパネルへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。	`{{ .PanelURL }}`
Fingerprint	文字列	アラートの識別に使用するフィンガープリント。	`{{ .Fingerprint }}`
ValueString	文字列	アラート内の各削減された式のラベルと値を含む文字列。	`{{ .ValueString }}`

ExtendedData

ExtendedData オブジェクトには、以下のプロパティが含まれています。

名前	[Kind] (種類)	説明	例
レシーバー	`string`	通知を送信するコンタクトポイントの名前。	`{{ .Receiver }}`
ステータス	`string`	ステータスは、少なくとも 1 つのアラートが発せられている場合は `firing` で、	`{{ .Status }}`

名前	[Kind] (種類)	説明	例
		それ以外の場合は resolved です。	
アラート	[]Alert	この通知のすべての発生アラートと解決済みアラートのリスト。	There are {{ len .Alerts }} alerts
発生アラート	[]Alert	この通知のすべての発生アラートのリスト。	There are {{ len .Alerts.Firing }} firing alerts
解決済みアラート	[]Alert	この通知で解決されたすべてのアラートのリスト。	There are {{ len .Alerts.Resolved }} resolved alerts
GroupLabels	KeyValue	これらのアラートをこの通知にグループ化するラベル。	{{ .GroupLabels }}
CommonLabels	KeyValue	この通知のすべてのアラートに共通するラベル。	{{ .CommonLabels }}
CommonAnnotations	KeyValue	この通知のすべてのアラートに共通する注釈。	{{ .CommonAnnotations }}
ExternalURL	string	この通知を送信した Grafana ワークスペースまたはアラートマネージャーへのリンク。	{{ .ExternalURL }}

KeyValue 型

KeyValue 型は、キー (ラベル) 値 (注釈) の文字列を組み合わせたものです。

KeyValue として保存されたデータに直接アクセスするメソッドだけでなく、データをソート、削除、変換するメソッドも備えています。

名前	引数	戻り値	メモ	例
SortedPairs		キーと値の文字列のソート済みのリスト		{{ .Annotations.SortedPairs }}
削除	[]string	KeyValue	指定したキーを除いたキー/値のマップのコピーを返します。	{{ .Annotations.Remove "summary" }}
名前		[]string	名前のリスト	{{ .Names }}
[Values] (値)		[]string	値のリスト	{{ .Values }}

[時間]

時間は Go time パッケージからのものです。時間はさまざまな形式で印刷できます。例えば、アラートが発生した時刻を Monday, 1st January 2022 at 10:00AM 形式で出力するには、次のテンプレートを作成します。

```
{{ .StartsAt.Format "Monday, 2 January 2006 at 3:04PM" }}
```

Go の時間形式に関するリファレンスは、こちらを参照してください。

コンタクトポイントの管理

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

[コンタクトポイント] リストビューには、既存のすべてのコンタクトポイントと通知テンプレートが一覧表示されます。

[コンタクトポイント] タブでは、次のことができます。

- コンタクトポイントと統合の名前とタイプを検索する。
- 既存のすべての連絡先と統合を表示する。
- 各コンタクトポイントが使用されている通知ポリシーの数を表示し、リンクされた通知ポリシーに直接移動する。
- 通知配信のステータスを表示する。
- 個々のコンタクトポイントまたはすべてのコンタクトポイントを JSON、YAML、または Terraform 形式でエクスポートする。
- 通知ポリシーで使用されていないコンタクトポイントを削除する。

[通知テンプレート] タブでは、次のことができます。

- 既存の通知テンプレートを表示、編集、コピー、または削除する。

アラート通知のサイレンシング

⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

サイレンスを使用してアラート通知を抑制できます。サイレンスは通知の作成を停止するだけで、アラートルールの評価やユーザーインターフェースへのアラートインスタンスの表示は引き続き行われます。アラートをサイレンスにする場合、停止する時間枠を指定します。

> **Note**
>
> アラート通知を一定の時間間隔、定期的なメンテナンス期間中など、で停止するには、サイレンスではなく ミュートタイミング を使用します。

サイレンスの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。
2. [サイレンス] を選択します。
3. アラートマネージャードロップダウンから [アラートマネージャー] を選択します。
4. [サイレンスの作成] を選択します。
5. [サイレンスの開始と終了] でサイレンスを有効にする開始日と終了日を選択します。
6. 終了時刻を設定する代わりに、[期間] でサイレンスが適用される期間を指定します。この方法を使用すると、[サイレンスの開始と終了] フィールドの終了時刻が自動的に更新されます。
7. [ラベル] と [値] フィールドに、1 つ以上の一致するラベルを入力します。照合機能により、どのルールにサイレンスを適用するかが決定されます。一致するアラート (発動状態) はすべて、[Affected alerts instances(影響を受けるアラートインスタンス)] フィールドに表示されます。
8. 必要に応じて、サイレンスに関するコメントを追加します。
9. [送信] を選択します。

サイレンスを編集するには

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。
2. 既存のサイレンスのリストを表示するには、[サイレンス] を選択します。
3. 編集するサイレンスを見つけ、[編集] (ペンアイコン) を選択します。
4. 変更後、[送信] を選択して変更を保存します。

既存のサイレンスを編集するには、[編集] アイコン (ペン) を選択します。

サイレンスフォームへの URL リンクを作成するには

サイレンスフォームにリンクするときは、`matcher` および `comment` クエリパラメータを使用して、デフォルトの一致するラベルとコメントを指定します。`matcher` パラメータは、次の `[label]`

[operator][value] の形式である必要があります。この時、operator パラメータには以下のいずれかを指定できます: = (正規表現ではなく等号)、!= (正規表現ではなく不等号)、=~ (等号、正規表現)、!~ (不等号、正規表現)。URL には、 キー matcher を含む多くのクエリパラメータを含めることができます。例えば、一致するラベル severity=critical と cluster!~europe-.* ならびにコメント Silence critical EU alerts を使用してサイレンスフォームにリンクするには、URL https://mygrafana/alerting/silence/new?matcher=severity%3Dcritical&matcher=cluster!~europe-*&comment=Silence%20critical%20EU%20alert を作成します。

外部アラートマネージャーの新しいサイレンスページにリンクするには、alertmanager クエリパラメータを追加します。

サイレンスを削除するには

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。
2. 既存のサイレンスのリストを表示するには、[サイレンス] を選択します。
3. 終了するサイレンスを選択し、[サイレンス解除] を選択します。この操作によりアラートの停止状態は終了します。

> **Note**
>
> サイレンス解除を行うと、終了時刻が現在の時刻に設定されたものとして、アラートの停止状態が終了します。(自動または手動を問わず) 終了したサイレンスは5 日間保持され、一覧表示されます。リストからサイレンスを手動で削除することはできません。

アラートルールの表示とフィルタリング

> このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

[アラート] ページには、アラートルールが一覧表示されます。デフォルトでは、ルールはデータソースの種類ごとにグループ化されます。Grafana セクションには、Grafana によって管理されるルールが一覧表示されます。Prometheus 互換データソースのアラートルールもここに表示されます。Prometheus 互換データソースのアラートルールは表示できますが、編集することはできません。

Mimir/Cortex/Loki ルールセクションには、Mimir、Cortex、または Loki データソースのすべてのルールが一覧表示されます。クラウドアラートルールもこのセクションに表示されます。

大量のアラートを管理する場合、拡張アラートルール検索機能を使用して、フォルダ、評価グループ、ルールをフィルタリングできます。さらに、ラベル、状態、タイプ、正常性などのプロパティでアラートルールをフィルタリングできます。

アラートルールの表示

Grafana アラートを使用すると、すべてのアラートを 1 ページに表示できます。

アラートの詳細を表示するには

1. Grafana コンソールの Grafana メニューで、[アラート]、[アラートルール] を選択します。デフォルトでは、リストビューが表示されます。
2. [表示形式] では、任意のオプションを選択して、グループ化されたビュー、リストビュー、および状態ビューを切り替えることができます。
3. ルール行を展開して、ルールラベル、注釈、データソース、ルールクエリ、およびルールの結果のアラートインスタンスのリストを表示します。

このページから、アラートルールのコピーを作成して、既存のルールを再利用することもできます。

アラートルールのエクスポート

ルールは Grafana ワークスペースで YAML または JSON にエクスポートできます。

- YAML、JSON、または Terraform にエクスポートする各アラートルールグループの横にある [ルールグループのエクスポート] アイコンを選択します。
- [ルールのエクスポート] を選択して、すべての Grafana マネージドアラートルールを YAML、JSON、または Terraform にエクスポートします。
- グループ内の個々のアラートルールの横にある [その他]、[エクスポートを変更 を選択して、プロビジョニングされたアラートルールを編集し、変更されたバージョンをエクスポートします。

プロビジョニングされたアラートのクエリ定義を表示する

プロビジョニングされたアラートの読み取り専用クエリ定義を表示します。ルール定義の「as-code」リポジトリにアクセスすることなく、アラートルールのクエリが正しいかどうかをすばやく確認できます。

グループ化されたビュー

グループ化されたビューには、フォルダ別にグループ化された Grafana アラートルールと、namespace と group によりグループ化された Loki または Prometheus アラートルールが表示されます。これは、ルールの管理を目的としたデフォルトのルールリストビューです。各グループを展開すると、そのグループ内のルールのリストが表示されます。さらにルールを展開すると、その詳細を見ることができます。また、ルールから生成されたアクションボタンとアラートを展開して、詳細を表示することもできます。

状態ビュー

状態ビューには、アラートルールが状態ごとグループ化されて表示されます。このビューを使用すると、各ルールがどの状態にあるかを一目で確認できます。各ルールは展開して詳細を表示することがでます。アクションボタンとこのルールによって生成されたアラート、および各アラートをさらに展開して詳細を表示することができます。

アラートルールのフィルタリング

[アラート] ページに表示されるアラートルールは、いくつかの方法でフィルタリングできます。

アラートルールをフィルタリングするには

1. [データソースの選択] からデータソースを選択します。選択したデータソースをクエリするアラートルールが表示されます。
2. [ラベルで検索] で、ラベルセレクタを使用して検索条件を入力します。例えば、environment=production;region=~US|EU,severity!=warning と指定します。
3. [Filter alerts by state (状態別にアラートをフィルタリング)] から、表示するアラート状態を選択します。その状態に一致するアラートルールが表示されます。他の状態に一致するルールは非表示になります。

ミュートタイミング

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ミュートタイミングとは、ポリシーの新しい通知が生成されたり送信されないようにする繰り返しの時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間や繰り返し発生するのを防ぐために使用されます。

サイレンスと似ていますが、ミュートタイミングはアラートルールの評価や、ユーザーインターフェースへのアラート表示を停止せず、通知の作成のみを防ぎます。

外部のアラートマネージャーデータソースの Grafana 管理のミュートタイミングとミュートタイミングを設定できます。

ミュートタイミングとサイレンス

次の表に、「ミュートタイミング」と「サイレンス」の違いを示します。

ミュートタイミング	サイレンス
繰り返しの時間間隔を定義します。	開始時刻と終了時刻を固定します。
作成され、通知ポリシーに追加されます。	ラベルを使用してアラートと照合し、サイレンスするかどうかが判断されます。

ミュートタイミングの追加

ミュートタイミングは Grafana ワークスペースで作成できます。

ミュートタイミングを追加するには

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。

2. [通知ポリシー] を選択し、[ミュートタイミング] タブを選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [+ ミュートタイミングの追加] ボタンを選択します。
5. フォームに入力して、ミュートタイミングに合わせて一致する[時間間隔]を作成します。
6. ミュートタイミングを保存します。

通知ポリシーへのミュートタイミングの追加

ミュートタイミングが設定されたら、一定間隔でミュートする通知ポリシーに追加して使用します。

通知ポリシーへのミュートタイミングの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。
2. [通知ポリシー] を選択し、[通知ポリシー] タブを選択します。
3. ミュートタイミングを追加する通知ポリシーを選択し、[...]、[編集] を選択します。
4. [ミュートタイミング] ドロップダウンから、ポリシーに追加するミュートタイミングを選択します。
5. 変更内容を保存します。

時間間隔

時間間隔は、アラートが抑制される特定の期間です。この期間は通常、特定の時間範囲と曜日、月、または年で構成されます。

サポートされている時間間隔のオプションは次のとおりです。

- [時間範囲] – 開始時間を含み、終了時間を含まない時間 (場所が選択されていない場合は UTC で、それ以外の場合は現地時間)。
- Location (場所) – タイミングの場所を設定します。時間範囲は、その場所の現地時間で表示されます。
- [曜日] – 曜日または一連の曜日。例えば、`monday:thursday`。
- [月の日付] – 1 か月内の日付。値は 1～31 の範囲で指定します。負の値は月の日を逆順で指定することを意味します。たとえば、-1 は月の最終日を表します。
- [月] - 暦月の完全な名前の数値で表された、その年の月。例えば、`1, may:august`。
- [年] - 間隔の年または複数年。例えば、`2023:2024`。

これらの要素はそれぞれリストにすることができ、一致するには要素内の少なくとも 1 つの項目を満たす必要があります。フィールドは、: を使用して、範囲もサポートします。例えば、monday:thursday。

フィールドを空白のままにすると、任意の時刻をフィールドと照合します。のインスタントが完全な時間間隔と一致するには、すべてのフィールドが一致している必要があります。ミュートタイミングには複数の時間間隔を含めることができます。

正確な期間を指定する場合は、その期間に必要なすべてのオプションを指定します。例えば、3 月、6 月、9 月、12 月の最初の月曜日の 12:00 から 24:00 UTC の時間間隔を作成する場合、時間間隔の指定は次のようになります。

- 時間範囲
  - 開始時間: 12:00
  - 終了時間: 24:00
- 曜日: monday
- 月: 3, 6, 9, 12
- 日: 1:7

アラートルールの状態と正常性を表示する

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールの状態と正常性は、アラートに関するいくつかの主要なステータスインジケータを提供します。

次の 3 つのコンポーネントがあります。

- アラートルールの状態
- アラートインスタンスの状態

- アラートルールの正常性

それぞれに関連はありますが、各コンポーネントは微妙に異なる情報を伝達します。

アラートルールの状態と正常性を表示するには

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。
2. [アラートルール] を選択して、既存のアラートのリストを表示します。
3. アラートルールを選択して、その状態と正常性を表示します。

アラートルールの状態

アラートルールは、次のいずれかの状態になります。

状態	説明		
[普通]	評価エンジンによって返される時系列のいずれも pending または firing 状態ではありません。		
[保留中]	評価エンジンから返される少なくとも 1 つの時系列は pending です。		
[発射]	評価エンジンから返される少なくとも 1 つの時系列は firing です。		

> **Note**
>
> アラートは最初に pending に移行し、次に firing に移行します。そのため、アラートが発生する前に少なくとも 2 つの評価サイクルが必要です。

アラートインスタンスの状態

アラートインスタンスは、次のいずれかの状態になります。

状態	説明		
[普通]	pending でも firing でもないアラートの状態。すべてが期待どおりに機能しています。		
[保留中]	設定されたしきい値期間未満でアクティブであったアラートの状態。		
[アラート]	設定されたしきい値期間よりも長くアクティブであったアラートの状態。		
[データなし]	設定された時間枠のデータを受信しない状態。		
[アラート]	アラートルールの評価試行中にエラーが発生しました。		

最後の状態を保持する

アラートルールは、NoData または Error 状態が発生したときに最後の状態を保持するように設定できます。これにより、アラートの発射、解決と再発射の両方が防止されます。通常の評価と同様に、アラートルールは保留期間が経過すると pending から firing に移行します。

アラートルールの正常性

アラートルールは、次のいずれかのヘルスステータスを持つことができます。

状態	説明		
[OK]	アラートルールを評価するときにエラーはありません。		
[エラー]	アラートルールの評価中にエラーが発生しました。		
[NoData]	ルール評価中に返される少なくとも 1 つの時系列にデータがありません。		
{status}、KeepLast	ルールは別のステータスを受け取りますが、アラートルールの最後の状態を保持するように設定されています。		

NoData とエラーの特別なアラート

アラートルールの評価で状態 NoData または Error が生成されると、Grafana アラートは、次の追加ラベルを含むアラートインスタンスを生成します。

ラベル	説明		
alertname	状態に応じて、DatasourceNoData または DatasourceError のいずれか。		

ラベル	説明		
datasource_uid	状態の原因となったデータソースのUID。		

> **Note**
>
> 追加のラベルを生成するには、Grafana マネージドアラートルールを設定する のトピックで説明されているように、アラートルールで no data またはエラー処理を `NoData` または `Error` に設定する必要があります。

これらのアラートは、サイレンスを追加したり、コンタクトポイントにルーティングしたりするなど、通常のアラートと同じ方法で処理できます。

アラートグループ別に表示およびフィルタリングする

> **⚠** このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートグループは、アラートマネージャーインスタンスからのグループ化されたアラートを表示します。デフォルトでは、アラートルールは通知ポリシーのデフォルトポリシーのラベルキーによってグループ化されます。共通のアラートルールを 1 つのアラートグループにグループ化することで、重複したアラートルールが発生することを防ぎます。

アラートグループを表示し、特定の条件に一致するアラートルールをフィルター処理することもできます。

アラートグループを表示するには

1. Grafana コンソールの Grafana メニューで、[アラート] を選択します。

2. [グループ] を選択して、既存のグループを表示します。
3. [アラートマネージャー] ドロップダウンから、データソースとして外部アラートマネージャーを選択します。
4. [カスタムグループ化] ドロップダウンでラベルの組み合わせを選択して、デフォルト以外のグループを表示します。これは、通知ポリシーのグループをデバッグおよび検証するのに役立ちます。

アラートにルートポリシーのグループ化またはカスタムグループ化のいずれかで指定されたラベルが含まれていない場合、アラートは `No grouping` のヘッダーを持つ [すべてのグループを取得] に追加されます。

アラートは、ラベルまたはアラートの状態によってフィルタリングできます。

ラベルでフィルタリングするには

- [検索] で既存のラベルを入力して、ラベルに一致するアラートを表示します。

  例えば、`environment=production,region=~US|EU,severity!=warning` と指定します。

状態でフィルタリングするには

- [状態] では、Active (アクティブ)、Suppressed (抑制)、または Unprocessed (未処理) の状態を選択すると、選択した状態に一致するアラートが表示されます。他のすべてのアラートは非表示になります。

通知エラーの表示

> ⚠ このドキュメントのトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知エラーを表示し、送信に失敗したか受信されなかった理由を確認できます。

> **Note**
>
> この機能は Grafana アラートマネージャーでのみサポートされています。

通知エラーを表示するには

1. 左側のメニューから、[アラート]、[コンタクトポイント] を続けて選択します。

   いずれかのコンタクトポイントが失敗した場合、ワークスペースの右隅にあるメッセージに、エラーがあることと、その数が表示されます。
2. コンタクトポイントを選択すると、そのコンタクトポイントのエラーの詳細が表示されます。

   エラーアイコンにカーソルを合わせると、エラーの詳細が表示されます。

   コンタクトポイントに複数の統合がある場合、各統合のすべてのエラーが表示されます。
3. [正常性] 列で、通知のステータスを確認します。

   これは、[OK]、[試行なし]、または [エラー]のいずれかです。

Grafana バージョン 9 での作業

> **⚠** このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana ワークスペースを作成するときに、使用する Grafana のバージョンを選択できます。以下のトピックでは、Grafana のバージョン 9 を使用する Grafana ワークスペースの使用について説明します。

トピック

- Grafana バージョン 9 のダッシュボード
- Grafana バージョン 9 のパネルと視覚化
- Grafana バージョン 9 で探索する
- Grafana バージョン 9 のアラート

Grafana バージョン 9 のダッシュボード

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードは、1 つ以上の行に編成および配置された 1 つ以上のパネルのセットです。Grafana にはさまざまなパネルが付属しているため、適切なクエリを簡単に作成し、必要に応じて最適なダッシュボードを作成できるように視覚化をカスタマイズできます。各パネルは、設定された任意のデータソースに接続する からのデータを操作できます。

ダッシュボードスナップショットは静的です。クエリと式をスナップショットから再実行することはできません。そのため、クエリまたは式内の変数を更新しても、ダッシュボードデータは変更されません。

トピック

- ダッシュボードの使用
- ダッシュボードの構築
- ダッシュボードの管理
- ダッシュボードとパネルの共有
- プレイリストの管理
- ダッシュボード変数の追加と管理
- ダッシュボード使用状況の評価
- Grafana バージョン 9 でのダッシュボードの検索

ダッシュボードの使用

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、ダッシュボード機能、ショートカットの概要、およびダッシュボード検索の使用方法について説明します。

機能

ダッシュボードを使用して、次の方法でデータの表示をカスタマイズできます。

機能	説明
1. ホーム	Grafana インスタンスで設定されたホームページにリダイレクトする Grafana ホームアイコンをクリックします。
2。タイトル	ダッシュボードタイトルをクリックすると、現在のフォルダに含まれるダッシュボードを検索できます。
3. ダッシュボードの共有	このオプションを使用して、リンクまたはスナップショットで現在のダッシュボードを共有します。ダッシュボード定義を共有モーダルからエクスポートすることもできます。
4. 新しいパネルの追加	このオプションを使用して、パネル、ダッシュボード行、またはライブラリパネルを現在のダッシュボードに追加します。
5. ダッシュボード設定	このオプションを使用して、ダッシュボード名、フォルダ、タグを変更し、変数と注釈クエ

機能	説明
	リを管理します。ダッシュボードのセットアップ方法の詳細については、「ダッシュボード設定の変更」を参照してください。
6. タイムピッカーのドロップダウン	このオプションをクリックして、相対時間範囲オプションを選択し、カスタム絶対時間範囲を設定します。 • [タイムゾーン] と [会計年度] の設定は、[時間設定の変更] ボタンをクリックして、時間範囲コントロールから変更できます。 • 時間設定はダッシュボードごとに保存されます。
7。時間範囲のズームアウト	クリックして時間範囲をズームアウトします。時間範囲の制御方法の詳細については、「ダッシュボードの時間範囲の設定」を参照してください。
8。[ダッシュボードの更新]	クリックすると、クエリをすぐにトリガーし、ダッシュボードデータを更新します。
9。ダッシュボードの時間間隔の更新	クリックして、ダッシュボードの自動更新時間間隔を選択します。
10。表示モード	このオプションを使用して、テレビやキオスクなどの大きな画面にダッシュボードを表示します。表示モードでは、ナビゲーションメニューなどの無関係な情報が非表示になります。

機能	説明
11。ダッシュボードパネル	ダッシュボードの主な構成要素はパネルです。新しいパネル、ダッシュボード行、またはライブラリパネルを追加するには、[パネルの追加] を選択します。 • ライブラリパネルは、多くのダッシュボード間で共有できます。 • パネルを移動するには、パネルヘッダーを別の場所にドラッグします。 • パネルのサイズを変更するには、パネルの右下隅をクリックしてドラッグします。
12。グラフ凡例	シリーズの色、y 軸、およびシリーズの可視性を凡例から直接変更します。
13。ダッシュボードの検索	[検索] をクリックして、名前またはパネルタイトルでダッシュボードを検索します。
14。ダッシュボード行	ダッシュボード行は、パネルをグループ化するダッシュボード内の論理的なディバイダーです。 • 行を折りたたんだり展開したりして、ダッシュボードの一部を非表示にできます。 • 折りたたまれた行内のパネルはクエリを発行しません。 • 繰り返し行を使用して、テンプレート変数に基づいて行を動的に作成します。

キーボードショートカット

Grafana では多数のキーボードショートカットを利用できます。使用可能なすべてのキーボードショートカットを表示するには、キーボードで ? または h を押します。

- `Ctrl+S` は、現在のダッシュボードを保存します。

- f は、ダッシュボードファインダー/検索を開きます。
- d+k は、キオスクモードを切り替えます (メニューを非表示にします)。
- d+e は、すべての行を展開します。
- d+s は、ダッシュボード設定を開きます。
- Ctrl+K は、コマンドパレットを開きます。
- Esc は、全画面表示または編集モードのときにパネルを終了します。また、ダッシュボード設定からダッシュボードに戻ります。

フォーカスされたパネル

特定のパネルをターゲットとするショートカットを使用するには、ポインターでパネルにカーソルを合わせます。

- e は、パネル編集ビューを切り替えます
- v は、パネル全画面表示を切り替えます
- ps は、パネル共有機能を開きます
- pd は、パネルを複製します
- pr は、パネルを削除します
- pl は、パネル凡例を切り替えます

ダッシュボードの時間範囲の設定

Grafana では、ダッシュボード、パネル、アラートなど、視覚化されるデータの時間範囲を管理する方法がいくつか用意されています。

このセクションでは、サポートされている時間単位と相対範囲、一般的な時間コントロール、ダッシュボード全体の時間設定、パネル固有の時間設定について説明します。

時間単位と相対範囲

Grafana は、次の時間単位に対応しています: s (seconds)、m (minutes)、h (hours)、d (days)、w (weeks)、M (months)、Q (quarters)、y (years)。

マイナス演算子を使用すると、現在の時間を基準に過去を指定することができます。単位の全期間 (日、週、月など) を表示するには、末尾に /<time unit> を追加します。会計期間を表示するには、fQ (fiscal quarter) と fy (fiscal year) の時間単位を使用します。

プラス演算子を使用すると、現在の時間を基準に未来を指定することができます。例えば、この機能を使用して、将来の予測データを確認できます。

次の表に、相対範囲の例を示します。

相対範囲の例	From	To
過去 5 分間	now-5m	now
今日のこれまでのところ	now/d	now
今週	now/w	now/w
今週始めから現在まで	now/w	now
今月	now/M	now/M
今月始めから現在まで	now/M	now
前月	now-1M/M	now-1M/M
今年のこれまでのところ	now/Y	now
今年	now/Y	now/Y
前会計年度	now-1y/fy	now-1y/fy

> **Note**
>
> Grafana アラートは、将来のタイムスタンプについて now+n をサポートしておらず、n の開始から n の終了までのタイプスタンプについて now-1n/n をサポートしていません。

一般的な時間範囲コントロール

ダッシュボードとパネルの時間コントロールには、共通のユーザーインターフェイスがあります。一般的な時間範囲コントロールを以下に示します。

- 現在の時間範囲 (別名タイムピッカー)は、そのダッシュボードまたはパネルに現在表示されている時間範囲を示します。フィールドにカーソルを合わせると、範囲内の正確なタイムス

タンプとそのソース (ローカルブラウザなど) が表示されます。現在の時間範囲をクリックして変更します。現在の時間は、過去 15 分などの相対時間範囲や、 のような `2020-05-14 00:00:00 ~ 2020-05-15 23:59:59` などの絶対時間範囲で変更できます。

- [相対時間範囲]は、[相対時間範囲] リストから選択できます。上部の入力フィールドを使用してリストをフィルタリングできます。時間範囲の例には、[過去 30 分]、[過去 12 時間]、[過去 7 日間]、[過去 2 年間] 、[昨日]、[昨日の前日]、[先週のこの曜日]、[今日のこれまでのところ]、[今週始めから現在まで]、[今月始めから現在まで]などがあります。
- 絶対時間範囲は、[開始] フィールドと [終了] フィールドに正確な時間値または相対時間値を入力して [時間範囲を適用] をクリックするか、[開始] フィールドまたは [終了] フィールドをクリックしたときに表示されるカレンダーから日付または日付範囲をクリックするという 2 つの方法で設定できます。選択を適用するには、[時間範囲の適用] をクリックします。

その他の時間範囲機能

1. ズームアウトするには、Cmd+Z または Ctrl+Z をクリックします。 アイコンをクリックして、ダッシュボードまたはパネルの視覚化でより広い時間範囲を表示します。
2. ズームイン機能を使用するには、表示したい視覚化の時間範囲をクリックしてドラッグします。

> **Note**
>
> ズームインはグラフの視覚化にのみ適用されます。

[ダッシュボードの更新]

[ダッシュボードの更新] アイコンをクリックすると、ダッシュボード上のすべてのクエリが即座に実行され、視覚化が更新されます。更新をトリガーすると、Grafana は保留中のリクエストをすべてキャンセルします。

デフォルトでは、Grafana はダッシュボードを自動的に更新しません。クエリは、パネル設定に従って独自のスケジュールで実行されます。[ダッシュボードの更新] アイコンの横にある下矢印をクリックし、更新間隔を選択することで、ダッシュボードを定期的に更新することができます。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、ダッシュボードの時間範囲を制御できます。

- from は、エポックミリ秒で指定された時間範囲の下限、または相対時間を定義します。
- to は、エポックミリ秒で指定された時間範囲の上限、または相対時間を定義します。
- time および time.window は、time-time.window/2 から time+time.window/2 までの時間範囲を定義します。両方のパラメータはミリ秒で指定する必要があります。たとえば、?time=1500000000000&time.window=10000 は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードの構築

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana ワークスペースを作成してサインインすると、ダッシュボードを作成し、ニーズに合わせて設定を変更できます。

トピック

- ダッシュボードの作成
- パネルの追加または編集
- ダッシュボード設定の変更
- ダッシュボード URL 変数
- ダッシュボードへのライブラリパネルの追加
- ダッシュボードのバージョン履歴の管理
- ダッシュボードリンクの管理
- ダッシュボード JSON モデル

ダッシュボードの作成

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードの作成

ダッシュボードとパネルを使用すると、Grafana を使用してデータを視覚的に表示できます。各パネルには、視覚化を表示するために少なくとも 1 つのクエリが必要です。開始する前に、前提条件として次の作業を完了します。

- 適切なアクセス許可を持っていることを確認してください。権限の詳細については、「ユーザー、チーム、およびアクセス許可」を参照してください。
- パネルを追加するダッシュボードを特定します。
- ターゲットデータソースのクエリ言語を理解します。
- クエリを記述するデータソースが追加されていることを確認します。

ダッシュボードを作成するには:

1. Grafana にサインインし、[ダッシュボード]にカーソルを合わせ、 [+ 新しいダッシュボード] をクリックします。
2. [新しいパネルの追加] をクリックします。
3. [クエリ] タブの最初の行で、ドロップダウンリストをクリックしてデータソースを選択します。
4. データソースのクエリ言語でクエリを記述または作成します。
5. [視覚化] リストで、視覚化タイプを選択します。Grafana は、視覚化が適用されたクエリ結果のプレビューを表示します。詳細については、「視覚化のオプション」を参照してください。
6. 次の方法でパネル設定を調整します。
   - 値マッピングを設定する
   - 視覚化固有のオプション

- フィールド値を上書きする
- しきい値を設定する
- 標準オプションを設定する

> **Note**
> ほとんどの視覚化では、情報を適切に表示する前にある程度の調整が必要です。

7. 視覚化を記述する (または変更を記述する) メモを追加し、ページの右上にある [保存] をクリックします。

> **Note**
> ダッシュボードを以前のバージョンに戻す必要が生じた際、このメモが役に立ちます。

繰り返し行の設定

変数の値に基づいてダッシュボードにパネルまたは行を動的に追加するように Grafana を設定できます。変数は、ダッシュボード内のすべての行にわたってクエリを動的に変更します。パネルの繰り返しの詳細については、「繰り返しパネルの設定」を参照してください。

`Multi-value` [または]`Include all values` が選択された変数セットがある場合は、行を繰り返すこともできます。

開始する前に、クエリに複数値変数が含まれていることを確認し、次のステップを実行します。

1. ダッシュボードのホームページで、[パネルの追加] をクリックします。
2. [パネルの追加] ダイアログボックスで、[新しい行の追加] をクリックします。
3. 行タイトルにカーソルを合わせ、歯車アイコンをクリックします。
4. [行オプション] ダイアログボックスで、タイトルを追加し、繰り返し行を追加する変数を選択します。

> **Note**
> ダッシュボードユーザーにコンテキストを提供するには、行タイトルに変数を追加します。

パネルを移動するには

1. ダッシュボードを開きます。

1. パネルタイトルをクリックし、パネルを新しい場所にドラッグします。パネルは、任意の場所のダッシュボードに配置することができます。

パネルのサイズを変更するには

1. ダッシュボードを開きます。
2. パネルのサイズを調整するには、パネルの右下隅をクリックしてドラッグします。ダッシュボードパネルは、ニーズに合わせてサイズ調整できます。

パネルの追加または編集

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードを作成したら、パネルをいつでも追加、編集、または削除できます。

- ダッシュボードの表示: ダッシュボードを表示するには、[ホーム]メニューから [ダッシュボード] を選択し、表示するダッシュボードを選択します。ダッシュボードを含むフォルダを展開する必要がある場合があります。
- パネルの追加: ダッシュボードにパネルを追加するには、ページの上部にあるメニューバーで [パネルの追加] アイコンを選択します。
- パネルの編集: ダッシュボード上の既存のパネルを編集するには、パネルにカーソルを合わせたときに表示されるメニューアイコンを選択し、[編集] を選択します。
- パネルの削除: ダッシュボード上の既存のパネルを削除するには、パネルにカーソルを合わせたときに表示されるメニューアイコンを選択し、[削除] を選択します。

ダッシュボード設定の変更

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボード設定ページでは、次の操作を実行できます。

- 時間設定を含む一般的なダッシュボードプロパティを編集する。
- 注釈クエリを追加する。
- ダッシュボード変数を追加する。
- リンクを追加する。
- ダッシュボードの JSON モデルを表示する

ダッシュボード設定ページにアクセスするには

1. ダッシュボードを編集モードで開きます。
2. ページの上部にある [ダッシュボード設定] (歯車アイコン) をクリックします。

ダッシュボードの時間設定の変更

ダッシュボードのタイムゾーン、ローカルブラウザの時刻を変更するときにダッシュボードの時刻設定を調整し、自動更新の時間間隔を指定します。

ダッシュボードの時間設定を変更するには

1. [ダッシュボード] 設定ページで、[全般] を選択します。
2. [時間オプション] セクションに移動します。
3. 以下の説明に従って時間設定を指定します。
4. タイムゾーンは、モニタリングするサービスまたはシステムのローカルタイムゾーンを指定します。これは、複数のタイムゾーンで動作するシステムまたはサービスを監視する場合に役立ちます。

  - Grafana は、ユーザープロファイル、チーム、または組織にデフォルトで選択されたタイムゾーンを使用します。ユーザープロファイル、ユーザーが所属するチーム、または組織にタイムゾーンが指定されていない場合、Grafana はローカルブラウザの時間を使用します。
  - ユーザーブラウザの表示用に設定されたタイムゾーン、ローカルブラウザ時間が使用されます。これは通常、コンピュータに設定されているタイムゾーンと同じです。
  - UTC を含む標準の ISO 8601 タイムゾーン を使用します。

- [自動更新] は、相対時間に表示されるオプションをカスタマイズし、自動更新オプションエントリはカンマ区切りで、有効な時間単位を受け入れます。
- [遅延時間の指定] は、遅延時間を入力して now 時間を上書きします。このオプションを使用して、null 値を避けるために、データ集約の既知の遅延に対応します。
- [タイムピッカーを非表示] は、Grafana タイムピッカーの表示を非表示にします。

> **Note**
>
> 時間コントロールの使用には、データに時間列が含まれている必要があります。時間列を含める方法の詳細については、特定の「データソース」のドキュメントを参照してください。

[注釈クエリの追加]

注釈クエリは、イベントをクエリするクエリです。これらのイベントは、カーソルを合わせるとイベント情報を表示できる小さなアイコンとともに、ダッシュボード全体のグラフで垂直線として視覚化できます。

n 注釈クエリを追加するには

1. [ダッシュボード設定] ページで、[注釈] を選択します。
2. [注釈クエリの追加] を選択します。
3. 名前を入力し、データソースを選択します。
4. フォームの残りの部分に入力して、クエリと注釈を作成します。

クエリエディタの UI は、選択したデータソースに基づいて変わります。クエリの作成方法の詳細については、「データソース」ドキュメントを参照してください。

変数の追加

変数を使用すると、よりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウンリストとして表示されます。これらのドロップダウンを使用すると、ダッシュボード内に表示されるデータを簡単に変更できます。

変数の詳細については、「変数」を参照してください。

1. [ダッシュボード設定] ページで、左側のセクションメニューの [変数]、[変数を追加] ボタンを続けてクリックします。
2. [全般] セクションで、変数の名前を追加します。これは、後でクエリで使用する名前です。
3. 変数タイプを選択します。

> **Note**
>
> 選択した変数タイプは、ページに入力するフィールドに影響します。

4. 変数を定義し、[更新] をクリックします。

リンクの追加

ダッシュボードリンクを使用すると、他のダッシュボードやウェブサイトへのリンクをダッシュボードヘッダーの直下に配置できます。リンクを使用すると、他の関連するダッシュボードやコンテンツに簡単に移動できます。

1. [ダッシュボード設定] ページで、左側のセクションメニューから [リンク] をクリックし、次に [リンクを追加] ボタンをクリックします。
2. タイトルを入力し、[タイプ] フィールドに [ダッシュボード] または [リンク] を選択します。
3. ダッシュボードリンクを追加するには、オプションのタグを追加し、ダッシュボードリンクのオプションを選択して、[適用] をクリックします。

> **Note**
>
> タグは、すべてが特定のタグを持つダッシュボードの動的なドロップダウンを作成するのに役立ちます。

4. リンクを追加するには、ユーザーがリンクの上にカーソルを置いたときに表示される URL とツールヒントテキストを追加し、リンクの横に表示されるアイコンを選択して、ダッシュボードのリンクオプションを選択します。

ダッシュボードの JSON モデルを表示する

Grafana のダッシュボードは、ダッシュボードのメタデータを保存する JSON オブジェクトによって表されます。ダッシュボードメタデータには、ダッシュボードプロパティ、パネルからのメタデータ、テンプレート変数、パネルクエリなどが含まれます。

ダッシュボード JSON モデルを表示するには、[ダッシュボード設定] ページで [JSON] をクリックします。

JSON フィールドの詳細については、「JSON フィールド」を参照してください。

ダッシュボード URL 変数

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は、ダッシュボード URL でクエリパラメータとして渡された変数値を適用できます。詳細については、「ダッシュボードリンクの管理」と「テンプレートおよび変数の管理」を参照してください。

変数をクエリパラメータとして渡す

Grafana は、`var-` のプレフィックスが付加されたクエリ文字列パラメータを、特定のダッシュボードの変数として解釈します。

例えば、この URL では、次のようになります。

```
https://${your-domain}/path/to/your/dashboard?var-example=value
```

クエリパラメータ `var-example=value` は、ダッシュボード変数の例を表し、値は `value` です。

変数に複数の値を渡す

複数の値を渡すには、値ごとに変数パラメータを 1 回繰り返します。

```
https://${your-domain}/path/to/your/dashboard?var-example=value1&var-example=value2
```

Grafana は、 `var-example=value1&var-example=value2` を `value1` と `value2` の 2 つの値を持つダッシュボード変数の例として解釈します。

ダッシュボードリンクへの変数の追加

Grafana は、ダッシュボードの設定から変数を生成するときに、ダッシュボードリンクに変数を追加できます。変数を追加する方法と手順の詳細については、「ダッシュボードリンクの管理」を参照してください。

アドホックフィルターを渡す

アドホックフィルターは、指定されたデータソースを使用するすべてのメトリクスクエリにキーまたは値フィルターを適用します。詳細については、「アドホックフィルター」を参照してください。

アドホックフィルターをクエリパラメータとして渡すには、変数構文を使用してアドホックフィルター変数を渡し、キー、演算子を値、値をパイプ区切りリストとして指定します。

例えば、この URL では、次のようになります。

`https://${your-domain}/path/to/your/dashboard?var-adhoc=example_key|=|example_value`

クエリパラメータ `var-adhoc=key|=|value` は、 `example_key` キー、 `=` 演算子、および `example_value` 値を使用して、アドホックダッシュボード変数として設定されたアドホックフィルターを適用します。

> **Note**
>
> アドホックフィルターを使用して URL を共有する場合は、必ず URL をエンコードしてください。上記の例では、パイプ (`|`) を `%7C`に、等価演算子 (`=`) を `%3D` に置き換えます。

URL の使用による時間範囲の制御

ダッシュボードの時間範囲を設定するには、`from`、`to`、`time`、および `time.window` クエリパラメータを使用します。これらは変数ではないため、`var-` プレフィックスは必要ありません。

ダッシュボードへのライブラリパネルの追加

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ライブラリパネルは、任意のダッシュボードで使用できる再利用可能なパネルです。ライブラリパネルを変更すると、その変更はパネルが使用されているすべてのインスタンスに反映されます。ライブラリパネルは、複数のダッシュボード間でパネルの再利用を合理化します。

ライブラリパネルは、保存されたダッシュボードとともにフォルダに保存できます。

ライブラリパネルの作成

ライブラリパネルを作成すると、ソースダッシュボードのパネルもライブラリパネルに変換されます。パネルが変換されたら、元のダッシュボードを保存する必要があります。

1. 編集モードでパネルを開きます。
2. パネル表示オプションで、下矢印オプションをクリックして、視覚化に変更を加えます。
3. [作成] ダイアログボックスを開くには、[ライブラリパネル] オプションをクリックし、[ライブラリパネルの作成] をクリックします。
4. [ライブラリパネル名] に名前を入力します。
5. [フォルダに保存] で、ライブラリパネルを保存するフォルダを選択します。
6. [ライブラリパネルの作成] をクリックして変更を保存します。
7. ダッシュボードを保存するには、 [保存] をクリックします。

ライブラリパネルの作成後、そのパネルは、パネルが表示される任意のダッシュボードを使用して変更することができます。変更を保存すると、ライブラリパネルのすべてのインスタンスにこれらの変更が反映されます。

ダッシュボードへのライブラリパネルの追加

他のダッシュボードユーザーに視覚化を提供する場合は、Grafana ライブラリパネルをダッシュボードに追加します。

1. 左のメニューの [ダッシュボード] オプションにカーソルを合わせ、ドロップダウンオプションから [新しいダッシュボード] を選択します。[パネルの追加] ダイアログボックスが開きます。
2. [パネルライブラリ] オプションから [パネルの追加] をクリックします。ライブラリパネルのリストが表示されます。
3. リストを検索またはフィルタリングして、追加するパネルを見つけます。
4. パネルをクリックしてダッシュボードに追加します。

ライブラリパネルのリンク解除

ライブラリパネルに変更を加える必要があるときにライブラリパネルの他のインスタンスに影響を与えたくない場合は、ライブラリパネルのリンクを解除します。

1. 左側のメニューの [ダッシュボード] にカーソルを合わせ、[ライブラリパネル] をクリックします。
2. さまざまなダッシュボードで使用されているライブラリパネルを選択します。
3. リンクを解除するパネルを選択します。
4. パネルのタイトルをクリックし、[編集] をクリックします。パネルは編集モードで開きます。
5. ページの右上隅にある [リンク解除] オプションをクリックします。

ライブラリパネルのリストの表示

ライブラリパネルに変更を加える必要があるときにライブラリパネルの他のインスタンスに影響を与えたくない場合は、ライブラリパネルのリンクを解除します。

1. 左側のメニューの [ダッシュボード] オプションにカーソルを合わせ、[ライブラリパネル] をクリックします。以前に定義したライブラリパネルのリストが表示されます。
2. 名前がわかっている場合は、特定のライブラリパネルを検索します。パネルをフォルダまたはタイプでフィルタリングすることもできます。

ライブラリパネルの削除

不要になったライブラリパネルを削除します。

1. 左側のメニューの [ダッシュボード] にカーソルを合わせ、[ライブラリパネル] を選択します。

2. 削除するパネルを選択します。
3. ライブラリ名の横にある [削除] アイコンをクリックします。

ダッシュボードのバージョン履歴の管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードのバージョンを保存するたびに、以前のバージョンのダッシュボードが失われないように、そのバージョンのコピーが保存されます。これらのバージョンのリストは、ダッシュボード設定に進み、左側のメニューで [バージョン] を選択することで確認できます。

ダッシュボードのバージョン履歴機能では、以前に保存したダッシュボードバージョンを比較して復元できます。

2 つのダッシュボードバージョンの比較

2 つのダッシュボードバージョンを比較するには、比較するリストから 2 つのバージョンを選択します。[バージョンの比較] をクリックして、2 つのバージョン間の差分を表示します。

ボタンをクリックすると、差分ビューが表示されます。デフォルトでは、変更の概要がテキストで表示されます。

また、下部の [JSON 差分の表示] ボタンをクリックすることで、ダッシュボードを表す raw JSON の差分を表示できます。

差分対象のバージョンに復元する場合は、右上の [バージョン <x> に復元] ボタンをクリックして復元できます。

以前に保存したダッシュボードバージョンへの復元

以前に保存したダッシュボードバージョンに復元する必要がある場合は、ダッシュボードバージョンリストの行の右側にある[復元] ボタンを選択するか、差分ビューで [バージョン <x> に復元] ボタンを選択します。ボタンをクリックすると、復元の確認を求める次のポップアップが表示されます。

以前のバージョンに復元すると、バージョン番号が異なるだけで、以前のバージョンとまったく同じデータを含む新しいバージョンが作成されます。これは、新しいダッシュボードバージョンの行の [メモ列] に表示されます。これは、以前のダッシュボードバージョンが変更の影響を受けないようにするためにのみ行われます。

ダッシュボードリンクの管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

リンクを使用すると、よく使用されるダッシュボード間を移動、または他のユーザーを視覚化に接続できます。リンクでは、他のダッシュボード、パネル、さらには外部ウェブサイトへのショートカットを作成できます。

Grafana では、ダッシュボードリンク、パネルリンク、データリンクをサポートしています。ダッシュボードリンクはダッシュボードの上部に表示されます。パネルリンクにアクセスするためには、パネルの左上隅にあるアイコンを選択します。

使用するリンクの選択

最初にダッシュボード間の現在の移動方法を確認します。頻繁に一連のダッシュボード間でジャンプし、各ダッシュボードで同じコンテキストを見つけるのに苦労している場合、ワークフローの最適化にリンクが役立ちます。

次のステップでは、ご使用のワークフローに適したリンクタイプを特定します。Grafana のすべてのリンクタイプは他のダッシュボードや外部ウェブサイトへのショートカットの作成に使用されますが、それらはさまざまなコンテキストで動作します。

- リンクがダッシュボード内のほとんどのパネルに関連する場合は、ダッシュボードリンクを使用します。
- 特定のパネルにドリルダウンする場合は、パネルリンクを使用します。
- 外部サイトへのリンクには、ダッシュボードリンクまたはパネルリンクのいずれかを使用できます。

- 特定のシリーズ、または 1 つの測定までドリルダウンするには、データリンクを使用します。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、パネルまたはダッシュボードの時間範囲を制御できます。

- `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。
- `to` は、エポックミリ秒で指定された時間範囲の上限を定義します。
- `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータをミリ秒で指定する必要があります。たとえば、`?time=1500000000000&time.window=10000` は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードリンク

ダッシュボードリンクを作成するときに、時間範囲と現在のテンプレート変数を含めることで、別のダッシュボードの同じコンテキストに直接ジャンプできます。これにより、リンクの送信先が適切なデータを表示しているかどうかを心配する必要がなくなります。他のタイプのリンクについては、「データリンク変数」を参照してください。

ダッシュボードリンクは、現在のダッシュボード名で GitHub の問題を送信するなど、外部システムへのショートカットとしても使用できます。

ダッシュボードリンクを追加すると、ダッシュボードの右上隅に表示されます。

ダッシュボードへのリンクの追加

現在のダッシュボードの上部にある他のダッシュボードへのリンクを追加します。

1. リンクするダッシュボードを表示しながら、画面上部の歯車をクリックして [ダッシュボード設定] を開きます。
2. [リンク] をクリックし、[ダッシュボードリンクを追加] または [新規] をクリックします。
3. [タイプ] で、[ダッシュボード] を選択します。
4. 以下からリンクオプションを選択します。
   - [タグあり]: タグを入力して、リンクされたダッシュボードを、入力したタグを持つダッシュボードのみに制限します。それ以外の場合、Grafana には他のすべてのダッシュボードへのリンクが含まれます。

- [ドロップダウン]: 多くのダッシュボードにリンクしている場合は、このオプションを選択し、ドロップダウンにオプションのタイトルを追加することをお勧めします。それ以外の場合、Grafana はダッシュボードのリンクをダッシュボードの上部に並べて表示します。
- [時間範囲]: リンクにダッシュボードの時間範囲を含めるには、このオプションを選択します。ユーザーがリンクをクリックすると、リンクされたダッシュボードが開き、指定された時間範囲が既に設定されています。
- [変数値]: このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクをクリックすると、リンクされたダッシュボード内の一致するテンプレートがリンクの値に設定されます。詳細については、「ダッシュボード URL 変数」を参照してください。
- [新しいタブで開く]: ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。

5. [追加] をクリックします。

ダッシュボードへの URL リンクの追加

現在のダッシュボードの上部にある URL へのリンクを追加します。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。

1. リンクするダッシュボードを表示しながら、画面上部の歯車をクリックして [ダッシュボード設定] を開きます。
2. [リンク] をクリックし、[ダッシュボードリンクを追加] または [新規] をクリックします。
3. [タイプ] で [リンク] を選択します。
4. 以下からリンクオプションを選択します。
   - URL: リンク先の URL を入力します。ターゲットによっては、フィールド値を含めることもできます。詳細については、この Github の例を参照してください。
   - タイトル: リンクを表示するタイトルを入力します。
   - ツールヒント: リンクを表示するツールヒントを入力します。
   - アイコン: リンクとともに表示するアイコンを選択します。
   - [時間範囲]: リンクにダッシュボードの時間範囲を含めるには、このオプションを選択します。ユーザーがリンクをクリックすると、リンクされたダッシュボードが開き、指定された時間範囲が設定されます。
     - `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。

- to は、エポックミリ秒で指定された時間範囲の上限を定義します。
- time および time.window は、time-time.window/2 から time+time.window/2 までの時間範囲を定義します。両方のパラメータをミリ秒で指定する必要があります。たとえば、?time=1500000000000&time.window=10000 は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

- [変数値]: このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクをクリックすると、リンクされたダッシュボード内の一致するテンプレートがリンクの値に設定されます。

  変数の形式は次のとおりです。

  ```
  https://${you-domain}/path/to/your/dashboard?var-${template-varable1}=value1&var-{template-variable2}=value2
  ```

- [新しいタブで開く]: ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。

5. [追加] をクリックします。

ダッシュボードリンクの更新

既存のダッシュボードリンクを変更または更新するには、次のステップに従います。

1. [ダッシュボード設定], の [リンク] タブで、編集する既存のリンクをクリックします。
2. 設定を変更し、[更新] をクリックします。

ダッシュボードリンクの複製

既存のダッシュボードリンクを複製するには、複製する既存のリンクの横にある複製アイコンをクリックします。

ダッシュボードリンクの削除

既存のダッシュボードリンクを削除するには、削除する複製アイコンの横にあるゴミ箱アイコンをクリックします。

パネルリンク

各パネルには、パネルの左上隅に表示する一連の独自のリンクを含めることができます。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。

使用可能なパネルリンクを表示するには、パネルの左上隅にあるアイコンをクリックします。

- パネルリンクの追加: 各パネルには、パネルの左上隅に表示される独自のリンクセットを指定できます。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータを重点的に確認できるようにすることもできます。パネルの左上隅にあるアイコンをクリックすると、使用可能なパネルリンクが表示されます。
  1. リンクを追加するパネルにカーソルを合わせ、e を押すか、または、パネルタイトルの横にあるドロップダウン矢印をクリックし、[編集] をクリックします。
  2. [パネル] タブで、[リンク] セクションまでスクロールダウンします。
  3. [リンク]を展開し、[リンクの追加] をクリックします。
  4. [タイトル] を入力します。[タイトル]は、UI に表示されるリンクの判読可能なラベルです。
  5. リンク先の URL を入力します。ダッシュボードで定義済みのテンプレート変数のいずれかを追加することもできます。Ctrl+Space または Cmd+Space を押して URL フィールドをクリックし、使用可能な変数を表示します。パネルリンクにテンプレート変数を追加することで、関連する変数がすでに設定されている適切なコンテキストをリンクからユーザーに送信します。

     時間変数も使用できます。

     - from は、エポックミリ秒で指定した、時間範囲の下限を定義します。
     - to は、エポックミリ秒で指定した、時間範囲の上限を定義します。
     - time および time.window は、time-time.window/2 から time+time.window/2 までの時間範囲を定義します。両方のパラメータはミリ秒で指定する必要があります。たとえば、?time=1500000000000&time.window=10000 は、1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。
- パネルリンクの更新
  1. [パネル] タブで、変更するリンクを見つけます。
  2. [編集] (鉛筆) アイコンをクリックして、[リンクの編集] ウィンドウを開きます。
  3. 必要に応じて変更します。
  4. [保存] をクリックして変更を保存し、ウィンドウを閉じます。
  5. 右上の [保存] をクリックして、ダッシュボードに変更を保存します。
- パネルリンクの削除

1. [パネル] タブで、変更するリンクを見つけます。
2. 削除するリンクの横にある [X] アイコンをクリックします。
3. 右上の [保存] をクリックして、ダッシュボードに変更を保存します。

ダッシュボード JSON モデル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana のダッシュボードは、ダッシュボードのメタデータを保存する JSON オブジェクトによって表されます。ダッシュボードメタデータには、ダッシュボードプロパティ、パネルからのメタデータ、テンプレート変数、パネルクエリが含まれます。

ダッシュボードの JSON を表示する方法。

1. ダッシュボードに移動します。
2. 上部のナビゲーションメニューで、[ダッシュボード設定] (歯車) アイコンをクリックします。
3. [JSON モデル] をクリックします。

JSON フィールド

ユーザーが新しいダッシュボードを作成すると、新しいダッシュボード JSON オブジェクトが次のフィールドで初期化されます。

> **Note**
>
> 次の JSON では、id は null として表示されます。これは、ダッシュボードが保存されるまで割り当てられるデフォルト値です。ダッシュボードを保存すると、整数値が `id` フィールドに割り当てられます。

```
{
  "id": null,
  "uid": "cLV5GDCkz",
  "title": "New dashboard",
  "tags": [],
  "style": "dark",
  "timezone": "browser",
  "editable": true,
  "graphTooltip": 1,
  "panels": [],
  "time": {
    "from": "now-6h",
    "to": "now"
  },
  "timepicker": {
    "time_options": [],
    "refresh_intervals": []
  },
  "templating": {
    "list": []
  },
  "annotations": {
    "list": []
  },
  "refresh": "5s",
  "schemaVersion": 17,
  "version": 0,
  "links": []
}
```

以下は、ダッシュボード JSON の各フィールドについて説明します。

名前	使用方法
id	ダッシュボードの一意の数値識別子 (db によって生成)
uid	誰でも生成できる一意のダッシュボード識別子。 文字列 (8 ~ 40)
title	ダッシュボードの現在のタイトル

名前	使用方法
[タグ]	ダッシュボードに関連付けられたタグ、文字列の配列
Style	ダークやライトなどのダッシュボードのテーマ
timezone	utc やブラウザなどのダッシュボードのタイムゾーン
editable	ダッシュボードが編集可能かどうか
graphTooltip	共有十字線またはツールチップなしの場合は 0 (デフォルト) 、共有十字線の場合は 1、共有十字線と共有ツールチップの場合は 2
time	過去 6 時間や過去 7 日など、ダッシュボードの時間範囲
timepicker	timepicker メタデータ。詳細については「timepicker セクション」を参照してください。
[テンプレート作成]	メタデータのテンプレート作成。詳細については[テンプレート作成セクションを参照してください。
annotations	注釈メタデータ。追加方法については「注釈」を参照してください。
[更新]	自動更新間隔
schemaVersion	JSON スキーマのバージョン (整数)。Grafana 更新がこのスキーマに変更を加えるたびに増加します。
version	ダッシュボードのバージョン (整数) 。ダッシュボードが更新されるたびに増加します。
panels	パネル配列 (詳細については、以下を参照)

パネル

パネルはダッシュボードの構成要素であり、データソースクエリ、グラフのタイプ、エイリアスなどで構成されます。パネル JSON は、JSON オブジェクトの配列で構成され、それぞれが異なるパネルを表します。ほとんどのフィールドはすべてのパネルで共通ですが、一部のフィールドはパネルタイプによって異なります。以下は、テキストパネルのパネル JSON の例です。

```
"panels": [
  {
    "type": "text",
    "title": "Panel Title",
    "gridPos": {
      "x": 0,
      "y": 0,
      "w": 12,
      "h": 9
    },
    "id": 4,
    "mode": "markdown",
    "content": "# title"
  }
```

パネルのサイズと位置

gridPos プロパティは、グリッド座標のパネルサイズと位置を記述します。

- w: 1 ~ 24 (ダッシュボードの幅は 24 列に分割されます)
- h: グリッドの高さの単位では、それぞれ 30 ピクセルを表します。
- x: w と同じ単位の x 位置。
- y: h と同じ単位の y 位置。

グリッドには負の重力があり、パネルの上に空のスペースがある場合にパネルを上に移動します。

Timepicker

```
"timepicker": {
    "collapse": false,
    "enable": true,
    "notice": false,
    "now": true,
    "refresh_intervals": [
```

```
      "5s",
      "10s",
      "30s",
      "1m",
      "5m",
      "15m",
      "30m",
      "1h",
      "2h",
      "1d"
    ],
    "status": "Stable",
    "type": "timepicker"
  }
```

[テンプレート作成]

templating フィールドには、保存された値を含むテンプレート変数の配列と、他のメタデータが含まれています。

```
"templating": {
    "enable": true,
    "list": [
       {
        "allFormat": "wildcard",
        "current":  {
          "tags": [],
          "text": "prod",
          "value": "prod"
        },
        "datasource": null,
        "includeAll": true,
        "name": "env",
        "options": [
           {
            "selected": false,
            "text": "All",
            "value": "*"
          },
           {
            "selected": false,
            "text": "stage",
            "value": "stage"
```

```
          },
           {
            "selected": false,
            "text": "test",
            "value": "test"
          }
        ],
        "query": "tag_values(cpu.utilization.average,env)",
        "refresh": false,
        "type": "query"
      },
       {
        "allFormat": "wildcard",
        "current":  {
          "text": "apache",
          "value": "apache"
        },
        "datasource": null,
        "includeAll": false,
        "multi": false,
        "multiFormat": "glob",
        "name": "app",
        "options": [
           {
            "selected": true,
            "text": "tomcat",
            "value": "tomcat"
          },
           {
            "selected": false,
            "text": "cassandra",
            "value": "cassandra"
          }
        ],
        "query": "tag_values(cpu.utilization.average,app)",
        "refresh": false,
        "regex": "",
        "type": "query"
      }
    ]
  }
```

ダッシュボードの管理

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードは、データを 1 つ以上の行に視覚的に表示する 1 つ以上のパネルのセットです。

ダッシュボードの作成の詳細については、[パネルの追加と整理]を参照してください。

ダッシュボードフォルダの作成

フォルダはダッシュボードの整理とグループ化に役立ちます。これは、同じGrafanaインスタンスを使用する多くのダッシュボードや複数のチームがある場合に便利です。

前提条件

Grafana 管理者のアクセス許可があることを確認します。ダッシュボードのアクセス許可の詳細については、「ダッシュボードのアクセス許可」を参照してください。

ダッシュボードフォルダを作成するには

1. Grafana にサインインし、サイドメニューで [ダッシュボード] > [新しいフォルダ] をクリックします。
2. 一意の名前を入力し、[作成] をクリックします。

Note
ダッシュボードを保存するときは、ダッシュボードを保存するフォルダを選択するか、新しいフォルダを作成するかを選択できます。

ダッシュボードとフォルダの管理

ダッシュボードとフォルダの管理ページで、次のことができます。

- フォルダを作成する
- ダッシュボードを作成する
- ダッシュボードをフォルダに移動する
- 複数のダッシュボードを削除する
- フォルダとダッシュボードのアクセス許可を割り当てることができるフォルダページに移動する

ダッシュボードフォルダページ

[ダッシュボードフォルダ] ページで、以下の操作を行うことができます。

- フォルダ内のダッシュボードを移動または削除する。
- フォルダの名前を変更する ([設定] タブで使用可能)。
- フォルダ (フォルダ内のダッシュボードによって継承される) にアクセス許可を割り当てる。

[ダッシュボードフォルダ] ページに移動するには、ダッシュボード検索結果リストのフォルダまたは [ダッシュボードとフォルダの管理] ページにカーソルを合わせると、歯車が表示されます。

ダッシュボードのアクセス許可

フォルダにアクセス許可を割り当てることができます。割り当てたアクセス許可は、フォルダ内のダッシュボードによって継承されます。アクセスコントロールリスト (ACL) は、組織ロール 、チーム、ユーザーに権限を割り当てるために使用します。

詳細については、「アクセス許可」を参照してください。

ダッシュボードのエクスポートとインポート

Grafana UI または HTTP API を使用してダッシュボードをエクスポートおよびインポートできます。

ダッシュボードのエクスポート

ダッシュボードのエクスポートアクションは、レイアウト、変数、スタイル、データソース、クエリなど、必要なすべてを含む Grafana JSON ファイルを作成し、後でダッシュボードをインポートできるようにします。

> **Note**
>
> Grafana は JSON ファイルをローカルマシンにダウンロードします。

1. エクスポートするダッシュボードを開きます。
2. 共有アイコンを選択します。
3. [エクスポート] をクリックします。
4. [ファイルに保存] を選択します。

ダッシュボードをポータブルにする

他のユーザーが使用するためにダッシュボードを エクスポートする場合は、メトリクスプレフィックス (定数変数を使用) やサーバー名などのテンプレート変数を追加できます。

タイプ `Constant` のテンプレート変数はダッシュボードで自動的に非表示になり、ダッシュボードのインポート時に必要な入力として追加されます。

ダッシュボードのインポート

1. サイドメニューで [ダッシュボード] を選択します。
2. [新規] を選択し、ドロップダウンメニューから [インポート] を選択します。
3. 次のいずれかのステップを実行します。
   - ダッシュボード JSON ファイルをアップロードします。
   - Grafana.com ダッシュボード URL を貼り付けます。
   - ダッシュボードの JSON テキストをテキストエリアに直接貼り付けます。

   インポートプロセスでは、ダッシュボードの名前を変更し、ダッシュボードで使用するデータソースを選択し、メトリクスプレフィックス (ダッシュボードが使用している場合) を指定できます。

ダッシュボードのトラブルシューティング

このセクションでは、一般的なダッシュボードの問題を解決するのに役立つ情報を提供します。

ダッシュボードが遅い

ダッシュボードが遅い場合は、次の点を考慮してください。

- グラフに数十 (または数百または数千) の時系列をレンダリングしようとしていますか？ これにより、ブラウザが遅延する可能性があります。highestMax (Graphite 内) などの関数を使用して、返されるシリーズを減らします。

- 場合によっては、シリーズ名が非常に大きくなることがあります。これにより、レスポンスサイズが大きくなります。エイリアスを使用して、返されるシリーズ名のサイズを小さくしてみてください。
- 多くの時系列または長い時間範囲をクエリしていますか？ これらの両方の条件により、Grafana またはデータソースが大量のデータを取得し、ダッシュボードの速度が低下する可能性があります。
- ネットワークインフラストラクチャに高負荷が発生している場合があり、特に速度が一貫していない場合は、これが問題である可能性があります。

ダッシュボードの更新レートの問題

デフォルトでは、Grafana は 30 秒ごとにデータソースをクエリします。ダッシュボードの更新率を低く設定すると、バックエンドに不要なストレスがかかります。多くの場合、変更が表示されるようにデータがシステムに送信されることはないため、頻繁にクエリを実行する必要はありません。

この問題が発生した場合は、次の解決策をお勧めします。

- ダッシュボード、パネル、変数の自動更新は、必要でない限り有効にしないでください。ユーザーはブラウザを手動で更新するか、妥当な期間 (10 分または 1 時間ごとなど) の更新レートを設定することができます。
- 必要に応じて、更新レートを 1 分に 1 回に設定します。ユーザーはいつでもダッシュボードを手動で更新できます。
- ダッシュボードの期間が長い場合 (1 週間など) 、自動更新は必要ない場合があります。

null データの処理またはレンダリングが間違っている、または混乱している

一部のアプリケーションは断続的にデータを発行します。例えば、イベントが発生した場合にのみメトリクスを投稿します。デフォルトでは、Grafana グラフはデータポイント間の線を接続します。

ダッシュボードとパネルの共有

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana を使用すると、組織内の他のユーザーとダッシュボードやパネルを共有したり、特定の状況でウェブ上で公開したりできます。以下を使用して共有できます。

- 直接リンク
- スナップショット
- エクスポートリンク (ダッシュボードのみ)

直接リンクによってレンダリングされた画像を表示するには、許可されたビューワーアクセス許可が必要です。

Grafana インスタンスに対して匿名アクセス許可が有効になっていない限り、埋め込みリンクを表示するには同じアクセス許可も必要です。

パネルまたはダッシュボードをスナップショットとして共有すると、スナップショット (スナップショットを作成した時点のパネルまたはダッシュボード) がウェブ上で公開されます。リンクを持つユーザーは誰でもアクセスできます。スナップショットは表示に許可を必要としないため、Grafana は、取得したアカウントに関する情報と機密データをスナップショットから削除します。

ダッシュボードの共有

ダッシュボードは、直接リンクまたはスナップショットとして共有できます。ダッシュボードをエクスポートすることもできます。

Note

ダッシュボードを変更する場合は、共有する前に変更を保存してください。

1. Grafana インスタンスのホームページに移動します。
2. 上部ナビゲーションの共有アイコンをクリックします。

   共有ダイアログボックスが開き、[リンク] タブが表示されます。

ダイレクトリンクの共有

[リンク] タブには、現在の時間範囲、テンプレート変数、デフォルトのテーマが表示されます。短縮 URL を共有することもできます。

1. [コピー] をクリックします。このアクションにより、デフォルトまたは短縮 URL がクリップボードにコピーされます。
2. コピーした URL を、リンクを表示する権限を持つ Grafana ユーザーに送信します。

スナップショットの公開

ダッシュボードスナップショットは、インタラクティブダッシュボードをパブリックに共有します。Grafana はクエリ (メトリクス、テンプレート、注釈) やパネルリンクなどの機密データを消去し、ダッシュボードに埋め込まれた可視メトリクスデータとシリーズ名のみが残されます。ダッシュボードスナップショットには、リンクを持つすべてのユーザーがアクセスできます。

スナップショットはローカルインスタンスに発行できます。

1. [ローカルスナップショット] をクリックします。
2. Grafana がスナップショットのリンクを生成します。スナップショットリンクをコピーし、組織内またはウェブ上で公開します。

ダッシュボードのエクスポート

Grafana ダッシュボードは簡単にエクスポートおよびインポートできます。詳細については、「インポートとエクスポート」を参照してください。

パネルの共有

パネルは、直接リンクまたはスナップショットとして共有できます。任意のパネルで [共有] オプションを使用してライブラリパネルを作成することもできます。

1. パネルタイトルをクリックして、パネルメニューを開きます。
2. [共有] をクリックします。共有ダイアログボックスが開き、[リンク] タブが表示されます。

直接リンクの使用

[リンク] タブには、現在の時間範囲、テンプレート変数、デフォルトのテーマが表示されます。必要に応じて、短縮 URL を有効にして共有できます。

1. [コピー] をクリックして、デフォルトまたは短縮 URL をクリップボードにコピーします。
2. コピーした URL を、リンクを表示する権限を持つ Grafana ユーザーに送信します。
3. オプションで [レンダリングイメージを直接リンク] をクリックして、パネルのイメージを共有することもできます。

サーバー側のレンダリングされたイメージの文字列パラメータのクエリ

- 幅: ピクセル単位での幅。デフォルトは 800 です。
- 高さ: ピクセル単位での高さ。デフォルトは 400 です。
- tz: HH と MM が UTC の後に時間と分でオフセットされる `UTC%2BHH%3AMM` 形式のタイムゾーン。
- タイムアウト: 秒数。パネルのクエリがデフォルトの 30 秒を超える場合、タイムアウトを延長できます。
- スケール: デバイススケール係数を設定する数値。デフォルトは 1 です。値を大きくすると、より詳細な画像 (DPI が高くなる) が生成されます。Grafana v7.0 以降でサポートされています。

スナップショットの公開

パネルスナップショットは、インタラクティブパネルをパブリックで共有します。Grafana は機密データを取り除き、ダッシュボードに埋め込まれた可視メトリクスデータとシリーズ名のみを残します。パネルスナップショットには、リンクを持つすべてのユーザーがアクセスできます。

スナップショットはローカルインスタンスに発行できます。

1. [パネルの共有] ダイアログボックスで、[スナップショット] タブを選択します。
2. [ローカルスナップショット] をクリックします。Grafana はスナップショットのリンクを生成します。
3. スナップショットリンクをコピーし、組織内またはウェブ上で公開します。

誤ってスナップショットを作成した場合は、[スナップショットの削除] をクリックして、Grafana インスタンスからスナップショットを削除します。

ライブラリパネルの作成

[パネルの共有] ダイアログボックスからライブラリパネルを作成できます。

1. [ライブラリパネル] をクリックします。
2. [ライブラリパネル名] に名前を入力します。
3. [フォルダに保存]、ライブラリパネルを保存するフォルダを選択します。デフォルトでは、[一般] フォルダが選択されています。
4. [ライブラリパネルの作成] を選択して変更を保存します。
5. [保存]をクリックしてダッシュボードを保存します。

プレイリストの管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

プレイリストは、順番に表示されるダッシュボードのリストです。プレイリストを使用して状況認識を構築したり、チームや訪問者にメトリクスを提示したりできます。Grafana はダッシュボードを任意の解像度に自動的にスケーリングするため、大きな画面に最適です。[ダッシュボード] サブメニューの Grafana のサイドメニューからプレイリスト機能にアクセスできます。

プレイリストへのアクセス、共有、制御

このセクションの情報を参照して、既存のプレイリストにアクセスします。5 つの使用可能なモードのいずれかを使用して、プレイリストの表示を開始して制御します。

プレイリストにアクセスするには

1. Grafana のサイドメニューにカーソルを合わせます。
2. [プレイリスト] をクリックします。

   既存のプレイリストのリストが表示されます。

プレイリストの開始

プレイリストの開始は 5 種類のビューモードから選択できます。表示モードによって、ダッシュボードへのメニューとナビゲーションバーの表示方法が異なります。

デフォルトでは、各ダッシュボードは間隔フィールド (プレイリストの作成時または編集時に設定可能) に入力された時間表示されます。プレイリストを開始したら、ページ上部のナビゲーションバーを使用してプレイリストを制御できます。

プレイリストには、プレイリストの作成時または編集時に設定された `Interval` フィールドで指定された時間の間、各ダッシュボードが表示されます。プレイリスト開始後は、画面上部のナビゲーションバーで制御します。

1. プレイリストページにアクセスして、既存のプレイリストのリストを表示します。
2. 開始するプレイリストを検索し、[プレイリストの開始] をクリックします。

   プレイリストの開始ダイアログボックスが開きます。
3. 次の表の情報に基づいて、使用可能な 5 つのプレイリストモードのいずれかを選択します。
4. [開始] をクリックします。

モード	説明
通常モード	• サイドメニューは表示されたままです。 • ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。
TV モード	• サイドメニューとダッシュボードサブメニュー (変数ドロップダウンとダッシュボードリンクを含む) は非表示となるか、取り除かれます。 • ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。 • ユーザーが 1 分間非アクティブになると、自動的に有効になります。 • d ｖシーケンスショートカットを使用するか、ダッシュボード URL に `?inactive` パラメータを追加して手動で有効にします。

モード	説明
	• ポインタ移動またはキーボードアクションで無効にします。
TV モード (自動フィットパネル付き)	• ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。 • ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。
キオスクモード	• サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。 • プレイリストの開始後、d v シーケンスショートカットを使用して手動で有効にできます。 • 同じショートカットを使用して手動で無効にすることもできます。
キオスクモード (自動フィットパネル付き)	• サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。 • ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。

プレイリストの制御

プレイリストの開始後は、画面上部のナビゲーションバーを使用して、[普通]モードまたは[テレビ]モードでプレイリストを制御できます。キーボードの Esc キーを押して、プレイリストを停止します。

ボタン	アクション
次へ (二重右矢印)	次のダッシュボードに進みます。
戻る (左矢印)	前のダッシュボードに戻ります。

ボタン	アクション
停止 (四角)	プレイリストを終了し、現在のダッシュボードに戻ります。
サイクルビューモード (モニターアイコン)	ダッシュボードの表示をさまざまなビューモードでローテーションします。
[時間範囲]	指定した時間範囲内のデータを表示します。下矢印を使用して、過去 5 分から 5 年前、または表示するカスタム時間範囲を設定します。
更新 (円矢印)	ダッシュボードを再読み込みして、最新のデータを表示します。ドロップダウン矢印を使用して、5 秒から 1 日ごとに自動的にリロードするように設定することもできます。

プレイリストの作成

プレイリストを作成して、ダッシュボード間の順序と時間間隔を設定して、ダッシュボードを順番に表示できます。

1. プレイリストページの [新規プレイリスト] をクリックします。
2. [名前] テキストボックスにわかりやすい名前を入力します。
3. [間隔] テキストボックスに時間間隔を入力します。

> **Note**
>
> 追加したダッシュボードは、順番に表示されます。

4. [ダッシュボード] で、[タイトルで追加] および [タグで追加] ドロップダウンオプションを使用して、既存のダッシュボードをプレイリストに追加します。
5. オプションで次のことが可能です:
   - ダッシュボードの名前、正規表現、またはタグでダッシュボードを検索します。
   - スター付きのステータスやタグで結果をフィルタリングします。
   - 上下矢印アイコンを使用して、追加したダッシュボードの順序を変更します。

- ダッシュボードの横にある X アイコンをクリックして、プレイリストからダッシュボードを削除します。

6. [保存] をクリックして変更を保存します。

プレイリストの保存

プレイリストを保存してプレイリストページに追加し、そこでプレイリストを開始できます。

> **Important**
>
> プレイリストを保存する前かつプレイリストを作成または編集するときに、プレイリストに表示するすべてのダッシュボードが追加されていることを確認します。

1. プレイリスト機能にアクセスするには、Grafana のサイドメニューにカーソルを合わせます。
2. [プレイリスト] をクリックして、利用可能なプレイリストを表示します。
3. 選択したプレイリストをクリックします。
4. プレイリストを編集します。
5. プレイリストに [名前]、[間隔] が設定されており、少なくとも 1 つの[ダッシュボード]が追加されていることを確認します。
6. [保存] をクリックして変更を保存します。

プレイリストの編集または削除

プレイリストを編集するには、名前、間隔時間を更新し、ダッシュボードの順序を追加、削除、再配置します。

プレイリストの編集

1. プレイリストページの [プレイリストの編集] をクリックします。
2. 上記の「プレイリストの作成」のステップに従って、名前と時間間隔を更新し、プレイリストからダッシュボードを追加または削除します。
3. [保存] をクリックして変更を保存します。

プレイリストの削除

1. [プレイリスト] をクリックします。
2. 削除するプレイリストの横にある [削除] をクリックします。

ダッシュボードの順序の並べ替え

1. 順序を変更するダッシュボードの横にある上矢印または下矢印をクリックします。
2. [保存] をクリックして変更を保存します。

ダッシュボードの削除

1. [削除] をクリックして、プレイリストからダッシュボードを削除します。
2. [保存] をクリックして変更を保存します。

ビューモードでプレイリストを共有する

お好みのビューモードで URL をコピーし、宛先に URL を貼り付けることで、プレイリストを共有できます。

1. [ダッシュボード] サブメニューで、[プレイリスト] をクリックします。
2. 共有するプレイリストの横にある [プレイリストの開始] をクリックします。
3. ドロップダウンで、目的のビューモードを右クリックします。
4. [リンクアドレスをコピー] をクリックすると、URL がクリップボードにコピーされます。
5. 送信先に URL を貼り付けます。

ダッシュボード変数の追加と管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

変数は、値のプレースホルダーです。メトリクスクエリとパネルタイトルで変数を使用できます。したがって、ダッシュボードの上部にあるドロップダウンを使用して値を変更すると、パネルのメトリッククエリが変更され、新しい値が反映されます。

変数を使用すると、よりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに変数を使用できます。変数は、ダッシュボードの上部にドロップダウンリストとして表示されます。これらのドロップダウンを使用すると、ダッシュボード内に表示されるデータを簡単に変更できます。

これらは、Grafana ビューワーが視覚化をすばやく調整できるようにしたいが、完全な編集権限を付与したくない管理者にとって特に便利です。Grafana ビューワーは変数を使用できます。

変数とテンプレートを使用すると、単一ソースのダッシュボードを使用することもできます。同じデータソースまたはサーバーが複数ある場合は、1 つのダッシュボードを作成し、変数を使用して表示内容を変更できます。これにより、メンテナンスと維持が大幅に簡素化されます。

テンプレート

テンプレートは、変数を含むすべてのクエリです。たとえば、複数のサーバーを監視するダッシュボードを管理する場合は、サーバーごとにダッシュボードを作成するか、1 つのダッシュボードを作成して、次のようなテンプレートクエリを含むパネルを使用できます。

```
wmi_system_threads{instance=~"$server"}
```

変数値は、常に var-<varname>=value という構文を使用して URL に同期されます。

例

変数は、画面上部のドロップダウンリストに表示されます。さまざまな変数を選択して、視覚化がどのように変化するかを確認します。

変数設定を表示するには、[ダッシュボード設定] > [変数] に移動します。リスト内の変数をクリックすると、その設定が表示されます。

変数は、タイトル、説明、テキストパネル、クエリで使用できます。$ で始まるテキストを含むクエリはテンプレートです。すべてのパネルにテンプレートクエリがあるわけではありません。

変数のベストプラクティス

- 変数のドロップダウンリストは、[ダッシュボード設定] の変数リストに一覧されている順序で表示されます。

- 頻繁に変更する変数を一番上に置くと、ダッシュボードの左端に最初に表示されるようになります。

変数の追加と管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

次の表に、Grafana に組み込まれる変数タイプ一覧を示します。

変数タイプ	説明
クエリ	メトリクス名、サーバー名、センサー ID、データセンターなど、クエリによって生成された値のリストです。クエリ変数を追加します。
カスタム	カンマ区切りリストを使用して変数オプションを手動で定義します。カスタム変数を追加します。
テキストボックス	オプションのデフォルト値を使用して、フリーテキスト入力フィールドを表示します。テキストボックス変数を追加します。
定数	非表示の定数を定義します。定数変数を追加します。
データソース	ダッシュボード全体のデータソースをすばやく変更します。データソース変数を追加します。
間隔	間隔変数は時間範囲を表します。間隔変数を追加します。

変数タイプ	説明
アドホックフィルター	データソースのすべてのメトリクスクエリに自動的に追加されるキー-値のフィルターを指します (Prometheus、Loki、InfluxDB、Elasticsearch のみ)。アドホックフィルターを追加します。
グローバル変数	クエリエディタの式で使用できる組み込み変数です。「グローバル変数」を参照してください。
連鎖変数	変数クエリには、他の変数を含めることができます。「連鎖変数」を参照してください。

一般的なオプションの入力

作成する変数のタイプには、一般的なオプションを入力する必要があります。

一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の [名前] を入力します。
4. [タイプ] 一覧で、[クエリ] を選択します。
5. (オプション) [ラベル] に、変数ドロップダウンの表示名を入力します。

   表示名を入力しない場合、ドロップダウンラベルは変数名です。
6. [非表示] オプションを次から選択します。

   - [選択なし (空白)]: 変数ドロップダウンには、変数の名前またはラベル値が表示されます。
   - [ラベル]: 変数ドロップダウンには、選択した変数値と下矢印のみが表示されます。
   - [変数]: ダッシュボードには変数ドロップダウンは表示されません。

クエリ変数の追加

クエリ変数を使用すると、メトリクス名、タグ値、またはキーのリストを返すデータソースクエリを記述できます。例えば、クエリ変数は、サーバー名、センサー ID、またはデータセンターのリストを返す場合があります。変数値は、データソースクエリでオプションを動的に取得するにつれて変化します。

クエリ変数は通常、文字列でのみサポートされます。クエリが数値またはその他のデータ型を返す場合は、変数として使用するために文字列に変換する必要があります。例えば、Azure データソースの場合、この目的のために tostring 関数を使用できます。

クエリ式には、他の変数への参照を含めることができ、実際にはリンクされた変数を作成できます。Grafana はこれを検出し、リンクされた変数のいずれかが変更されると変数を自動的に更新します。

> **Note**
>
> クエリ式はデータソースごとに異なります。詳細については、ご使用のデータソースのドキュメントを参照してください。

クエリ変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [データソース] リストで、クエリのターゲットデータソースを選択します。
3. [更新] リストで、変数がオプションを更新するタイミングを選択します。
   - [ダッシュボード読み込み時]: ダッシュボードが読み込まれるたびにデータソースにクエリを実行します。これにより、ダッシュボードの読み込みが遅くなります。ダッシュボードを初期化する前に変数クエリを完了する必要があるためです。
   - [時間範囲の変更時]: ダッシュボードの時間範囲が変更されたときにデータソースにクエリを実行します。このオプションは、可変オプションクエリに時間範囲フィルターが含まれているか、ダッシュボードの時間範囲に依存している場合にのみ使用します。
4. [クエリ] フィールドにクエリを入力します。
   - クエリフィールドはデータソースによって異なります。一部のデータソースには、カスタムクエリエディタがあります。

- 1 つの入力フィールドクエリエディタにより多くのスペースが必要な場合は、フィールドの右下隅にある線上にカーソルを合わせ、下にドラッグして展開します。

5. (オプション) [正規表現] フィールドに正規表現を入力して、データソースクエリによって返される名前の特定部分をフィルタリングまたはキャプチャします。例については、「正規表現での変数のフィルタリング」を参照してください。
6. [ソート] リストで、ドロップダウンリストに表示される値のソート順序を選択します。デフォルトのオプション [無効] では、データソースクエリによって返されるオプションの順序が使用されます。
7. (オプション) [選択オプション] を入力します。
8. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
9. [追加] を選択してダッシュボードに変数を追加します。

カスタム変数の追加

数値や文字列など、変更されない値にはカスタム変数を使用します。

例えば、サーバー名やリージョン名が変更されない場合は、クエリ変数ではなくカスタム変数として作成できます。これらは変更されないため、他のクエリ変数ではなく、連鎖変数で使用できます。これにより、連鎖変数が更新されたときに Grafana が送信する必要があるクエリの数が減少します。

カスタム変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [

   カンマで区切られた値] リストで、この変数の値をカンマ区切りリストに入力します。スペースおよびコロンで区切られた数値、文字列、またはキーと値のペアを含めることができます。例えば、`key1 : value1,key2 : value2`。
3. (オプション) [選択オプション] を入力します。
4. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
5. [追加] を選択してダッシュボードに変数を追加します。

テキストボックス変数の追加

テキストボックス変数は、任意のデフォルト値を使用して自由テキスト入力フィールドを表示します。これは、任意の値を入力できるため、最も柔軟な変数です。カーディナリティの高いメトリクスがある場合や、ダッシュボード内の複数のパネルを同時に更新する場合は、このタイプの変数を使用します。

テキストボックス変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. (オプション) [デフォルト値] フィールドで、変数のデフォルト値を選択します。このフィールドに何も入力しない場合、Grafana はユーザーがテキストを入力するための空のテキストボックスを表示します。
3. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

定数変数の追加

定数変数を使用すると、非表示の定数を定義できます。定数変数は、共有するダッシュボードに対するメトリックパスの接頭辞に有用です。ダッシュボードをエクスポートすると、定数変数がインポートオプションに変換されます。

定数変数は複数設定できません。各定数変数は 1 つの値のみを保持し、変数設定を更新しない限り更新できません。

定数変数は、クエリに含める必要がある複素数値があっても、クエリすべてに再入力したくない場合に便利です。例えば、「`i-0b6a61efe2ab843gg`」というサーバーパスがある場合、それを「`$path_gg`」という変数に置き換えることができます。

定数変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [値] フィールドに変数値を入力します。英数字と記号を入力できます。raw 形式 を使用している場合は、ワイルドカードを使用することもできます。
3. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

データソース変数の追加

データソース変数を使用すると、ダッシュボード全体のデータソースをすばやく変更できます。データソースのインスタンスが複数ある場合、おそらく異なる環境にある場合に便利です。

データソース変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [タイプ] リストで、変数のターゲットデータソースを選択します。
3. (オプション) [インスタンス名フィルター] には、変数値のドロップダウンリストからデータソースインスタンスを選択する正規表現フィルターを入力します。このフィールドは空のままにして、すべてのインスタンスを表示します。
4. (オプション) [選択オプション] を入力します。
5. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
6. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の追加

間隔変数を使用して、1m、1h、1dなどの時間範囲を表します。これらはダッシュボード全体の group by time コマンドと考えることができます。間隔変数は、視覚化でのデータのグループ化方法を変更します。また、[自動] オプションを使用して、時間範囲ごとに複数のデータポイントを返すこともできます。

間隔変数を時間 (InfluxDB の場合)、日付ヒストグラム間隔 (Elasticsearch の場合) でグループ化するパラメータとして、または要約関数パラメータ (Graphite の場合) として使用できます。

間隔変数を追加するには

1. 上記のように、一般的なオプションを入力します。
2. [値] フィールドに、変数ドロップダウンリストに表示する時間間隔を入力します。次の時間単位に対応しています: `s` (seconds)、`m` (minutes)、`h` (hours)、`d` (days)、`w` (weeks)、`M` (months)、`y` (years)。次のデフォルト値を使用または編集することもできます: `1m,10m,30m,1h,6h,12h,1d,7d,14d,30d`。
3. (オプション) `auto`オプションをリストに追加する場合は、[自動オプション] をオンにします。このオプションを使用すると、現在の時間範囲を分割して現在の `auto` 時間範囲を計算する回数を指定できます。オンにすると、さらに次の 2 つのオプションが表示されます。

- [ステップ数] – [最大データポイント数]クエリオプションと同様に、現在の時間範囲を分類して値を計算する回数を選択します。例えば、現在の表示時間範囲が 30 分の場合、auto 間隔はデータを 1 分単位× 30 でグループ化します。デフォルト値は 30 ステップです。
- [最小間隔] – ステップ数の間隔が時間を分類しない最小しきい値です。引き続き 30 分の例を挙げる場合、最小間隔をに設定すると、Grafana はデータを 2 分単位× 15 でグループ化します。

4. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
5. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の例

Graphite 関数でテンプレート変数 `myinterval` を使用する例:

```
summarize($myinterval, sum, false)
```

アドホックフィルターの追加

アドホックフィルターを使用して、指定されたデータソースを使用するすべてのメトリクスクエリに自動的に追加されるキー/値フィルターを追加できます。他の変数とは異なり、クエリではアドホックフィルターを使用しません。代わりに、アドホックフィルターを使用して既存のクエリのフィルターを書き込みます。

> **Note**
>
> アドホックフィルター変数は、Prometheus、Loki、InfluxDB、および Elasticsearch データソースでのみ機能します。

1. 上記のように、一般的なオプションを入力します。
2. [データソース] リストで、ターゲットデータソースを選択します。
3. [追加] を選択してダッシュボードに変数を追加します。

アドホックフィルターの作成

アドホックフィルターは、利用可能な最も複雑で変更可能な変数オプションの 1 つです。変数オプションの通常リストの代わりに、この変数によりダッシュボード全体のアドホッククエリを構築できます。この方法で適用するフィルターは、ダッシュボード上のすべてのパネルに適用されます。

変数選択オプションを設定する

選択オプションは、可変オプションの選択を管理するために使用できる機能です。すべての選択オプションは任意で使用でき、デフォルトではオフになっています。

複数値の変数

複数値が選択された変数を補間するのは、変数が使用されている特定のコンテキストで有効な文字列に複数の値をフォーマットする方法が単純ではないため、難しくなります。Grafana は、各データソースプラグインがテンプレート補間エンジンに、複数値に使用する形式を通知できるようにすることで、この問題を解決しようと試みます。

> **Note**
>
> Grafana がすべての値を 1 つの文字列にフォーマットするには、 変数の [すべての値をカスタム化] オプションを空白にする必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (一緒につなげる) します。例えば、`value1,value2,value3`。カスタム `all` 値を使用する場合は、値は代わりに `*` または `all` になります。

Graphite データソースを使用した複数値変数

Graphite は glob 式を使用します。この場合、複数の値を持つ変数は、現在の変数値が host1、host2、host3 であった場合は、`{host1,host2,host3}` として補間されます。

Prometheus または InfluxDB データソースを使用した複数値変数

InfluxDB と Prometheus は正規表現を使用するため、同じ変数は `(host1|host2|host3)` として補間されます。すべての値も正規表現エスケープされます。そうでない場合、正規表現コントロール文字を持つ値は正規表現を破ります。

Elastic データソースを使用した複数値変数

Elasticsearch は Lucene クエリ構文を使用するため、同じ変数は `("host1" OR "host2" OR "host3")` としてフォーマットされます。この場合、すべての値がエスケープされ、値には lucene 制御語と引用符のみが含まれます。

複数値変数のトラブルシューティング

自動エスケープとフォーマットは問題を引き起こす可能性があり、その背後にあるロジックを理解するのは難しい場合があります。特に InfluxDB と Prometheus では、正規表現構文を使用するために、変数を正規表現演算子コンテキストで使用する必要があります。

Grafana でこの自動正規表現エスケープとフォーマットを実行したくない場合は、次のいずれかを実行する必要があります。

- [複数値] をオフにするか、[すべて含めるオプション] オプションを使用します。
- raw 変数形式を使用します。

すべてを含めるオプション

Grafana は、変数ドロップダウンリストに `All` オプションを追加します。ユーザーがこのオプションを選択すると、すべての変数オプションが選択されます。

すべての値のカスタム化

このオプションは、[すべてを含めるオプション] が選択されている場合にのみ表示されます。

[すべての値のカスタム化] フィールドに 正規表現、globs、または lucene 構文を入力して、 `All` オプションの値を定義します。

デフォルトでは、`All` 値は結合式内のすべてのオプションを含みます。これは非常に長くなり、パフォーマンスの問題が発生する可能性があります。ワイルドカード正規表現など、すべての値のカスタム化を指定する方がよい場合があります。

[すべての値のカスタム化] オプションにカスタム正規表現、globs、または lucene 構文を含めるには、エスケープされないため、データソースに有効な値は何かを検討する必要があります。

グローバル変数

Grafana には、クエリエディタの式で使用できるグローバル組み込み変数があります。このトピックでは、それらをアルファベット順に一覧表示し、定義します。これらの変数は、クエリ、ダッシュボードリンク、パネルリンク、データリンクに役立ちます。

$__dashboard

この変数は、現在のダッシュボードの名前です。

$__from および $__to

Grafana には、「$__from」と「$__to」の 2 つの組み込み時間範囲変数があります。これらは現在、デフォルトでは常に工ポックミリ秒として補間されていますが、日付の書式設定をユーザーが管理できます。

構文	結果の例	説明
${__from}	1594671549254	Unix 工ポックミリ秒
${__from:date}	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
${__from:date:iso}	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
${__from:date:seconds}	1594671549	Unix 工ポック秒
${__from:date:YYYY-MM}	2020-07	文字を含まないあらゆるカスタム日付形式

上記の構文は、${__to} としても機能します。

$__interval

$__interval 変数を時間 (InfluxDB 、MySQL、Postgres、MSSQL の場合)、日付ヒストグラム間隔 (Elasticsearch の場合) でグループ化するパラメータとして、または[要約]関数パラメータ (Graphite の場合) として使用できます。

Grafana は、クエリで時間別にグループ化するために使用できる間隔を自動的に計算します。グラフに表示されるデータポイントよりも多い場合、クエリはより長い間隔でグループ化することで、より効率的になります。例えば、3 か月分のデータのグラフを見ている場合、分レベルで詳細を表示できない場合があります。時間別または日別にグループ化すると、グラフの表示内容に影響を与えることなく、クエリの効率が向上します。$__interval は、時間範囲とグラフの幅 (ピクセル数) を使用して計算されます。

概算: `(to - from) / resolution`

例えば、時間範囲が 1 時間でグラフが全画面表示の場合、間隔は 2m で計算され、ポイントは 2 分間隔でグループ化されます。時間範囲が 6 か月で、グラフが全画面表示の場合、間隔は 1d (1 日) で、ポイントは日ごとにグループ化されます。

InfluxDB データソースでは、レガシー変数 $interval は同じ変数であり、$__interval を代わりに使用すべきです。

InfluxDB データソースと Elasticsearch データソースには、間隔をハードコード化したり、$__interval 変数の最小制限 (Group by time interval など、> 構文を使用して) を設定したりするために使用される >10m フィールドがあります。

$__interval_ms

この変数はミリ秒単位の $__interval 変数であり、時間間隔形式の文字列ではありません。例えば、$__interval が 20m の場合、$__interval_ms は 1200000 です。

$__org

この変数は現在の組織の ID です。 ${__org.name} は現在の組織の名前を表します。

$__user

${__user.id} は現在のユーザーの ID です。 ${__user.login} は現在のユーザーのログインハンドルです。 ${__user.email}は現在のユーザーの E メールです。

$__range

現在、Prometheus および Loki データソースでのみサポートされています。この変数は、現在のダッシュボードの範囲を表します。to - from で計算されます。ミリ秒は $__range_ms、秒は $__range_s で表示されます。

$__rate_interval

現在、Prometheus データソースでのみサポートされています。$__rate_interval 変数は、レート関数で使用することを目的としています。

$timeFilter または $__timeFilter

$timeFilter 変数は、現在選択されている時間範囲を式として返します。例えば、時間間隔「Last 7 days」の式は time > now() - 7d です。

これは、次のようないくつかの場所で使用されます。

- InfluxDB データソースの WHERE 句。クエリエディタ では、Grafana は InfluxDB クエリに自動的に追加します。テキストエディタモードでは、WHERE $timeFilter を手動で追加できます。
- Log Analytics は、Azure Monitor データソースでクエリを実行します。

- SQL は、MySQL 、Postgres、MSSQL でクエリを実行します。
- `$__timeFilter` 変数は MySQL データソースで使用されます。

連鎖変数

連鎖変数は、リンクされた変数またはネストされた変数とも呼ばれ、変数クエリ内において 1 つ以上の他の変数を持つクエリ変数です。このセクションでは、連鎖変数がどのように機能するかを説明し、連鎖変数を使用するダッシュボード例へのリンクを共有します。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべてのデータソースで同じです。連鎖変数クエリは、それらを使用できる任意のデータソースで使用できます。

非常に複雑なリンクされたテンプレート化されたダッシュボードは、5 または 10 レベルの深さで作成できます。技術的には、どれだけ深く、複雑にできるかに制限はありませんが、リンクが多いほどクエリの負荷が大きくなります。

ベストプラクティスとヒント

以下のプラクティスにより、ダッシュボードと変数の使用が容易になります。

新しいリンクされた変数の作成

- 連鎖変数は親/子の依存関係を作成します。ラダーまたはツリー構造として想定できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数の基盤となる変数をコピーすることです。変数リストで、変数エントリの右側にある変数の複製アイコンをクリックしてコピーを作成します。その後、親変数のクエリに追加できます。
- この方法で作成した新しい変数は、リストの下部に表示されます。論理的な順序にするには、リスト内の別の位置にドラッグする必要がある場合があります。

変数の順序

ダッシュボード変数リストの変数の順序は、各エントリの右側にある上下の矢印をクリックして変更できます。Grafana は、このリストに従って左から右に変数のドロップダウンを一覧表示し、一番上の変数を左端に表示します。

- 子変数の前に、親変数と依存関係のない変数のリストを作成します。
- 各変数は、依存している変数に従う必要があります。
- UI には、どの変数に依存関係があるのかを示すものがありません。変数を論理的な順序で一覧表示して、他のユーザー (および自分自身) に明確にします。

複雑さに関する考慮事項

変数の依存関係のレイヤーが多いほど、変数を変更した後にダッシュボードを更新するのに時間がかかります。

例えば、4 項目がリンクされた一連の変数 (国、地域、サーバー、メトリック) があり、ルート変数値 (国) を変更する場合、Grafana はダッシュボード内の視覚化を更新する前に、すべての従属変数に対してクエリを実行する必要があります。

変数の管理

変数ページでは、変数を追加し、既存の変数を管理できます。また、変数を検査し、変数が他の変数またはダッシュボードで参照 (または使用されている) されているかどうかを識別することもできます。

[移動] ドラッグアンドドロップを使用して、変数をリストの上または下に移動できます。

[クローン] 変数をクローン (コピー) するには、右側の一連のアイコンからクローンアイコンをクリックします。これにより、元の変数の名前に copy_of_ というプレフィックスが付いた変数のコピーが作成されます。

削除： 変数を削除するには、右側の一連のアイコンからごみ箱アイコンをクリックします。

正規表現での変数のフィルタリング

[正規表現クエリ] オプションを使用して、変数クエリによって返されるオプションの一覧をフィルタリングするか、返されるオプションを変更します。

このページでは、正規表現を使用して変数ドロップダウンの値をフィルタリング/変更する方法を説明します。

[正規表現クエリ] オプションを使用して、変数クエリによって返されるオプションの一覧をフィルタリングするか、返されるオプションを変更します。詳細については、[正規表現]に関する Mozilla ガイドを参照してください。

次の例は、次のオプションリストのフィルタリングを示しています。

```
backend_01
backend_02
backend_03
backend_04
```

01 または **02** で終わるオプションのみが返されるようにフィルタリングする

正規表現:

```
/
(
01|02
)
$/
```

結果:

```
backend_01
backend_02
```

正規表現キャプチャグループを使用してテキストの一部を返すオプションのフィルタリングおよび変更

正規表現:

```
/.*
(
01|02
)
/
```

結果:

```
01
02
```

フィルタリングと変更 - Prometheus の例

オプションのリスト:

```
up{instance="demo.robustperception.io:9090",job="prometheus"} 1 1521630638000
up{instance="demo.robustperception.io:9093",job="alertmanager"} 1 1521630638000
up{instance="demo.robustperception.io:9100",job="node"} 1 1521630638000
```

正規表現:

```
/. *instance="
```

```
(
[^"]*
)
.*/
```

結果:

```
demo.robustperception.io:9090
demo.robustperception.io:9093
demo.robustperception.io:9100
```

名前付きテキストおよび値キャプチャグループを使用してフィルタリングおよび変更する

名前付きキャプチャグループを使用すると、変数クエリによって返されるオプションから「テキスト」部分と「値」部分を別々にキャプチャできます。これにより、変数ドロップダウンリストに、選択できる各値に対するわかりやすい名前を含めることができます。

例えば、node_hwmon_chip_names Prometheus メトリクスをクエリする場合、 chip_nameは chip 値よりもはるかにわかりやすくなります。したがって、次の変数クエリの結果は以下のとおりになります。

```
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_0",chip_name="enp216s0f0np0"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_1",chip_name="enp216s0f0np1"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_2",chip_name="enp216s0f0np2"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_3",chip_name="enp216s0f0np3"} 1
```

次の正規表現を渡す:

```
/chip_name="(?<text>[ ^ " ] + ) |chip=" (?<value >[ ^ " ] + )/g
```

次のドロップダウンリストを生成:

```
Display Name          Value
------------          -------------------------
enp216s0f0np0         0000:d7:00_0_0000:d8:00_0
enp216s0f0np1         0000:d7:00_0_0000:d8:00_1
enp216s0f0np2         0000:d7:00_0_0000:d8:00_2
enp216s0f0np3         0000:d7:00_0_0000:d8:00_3
```

text および value キャプチャグループ名のみがサポートされています。

変数ページでは、変数が他の変数またはダッシュボードで参照 (または使用) されているかどうかを簡単に識別できます。

参照または使用される変数の横には緑色のチェックマークが付き、参照されていない変数の横にはオレンジ色の注意アイコンが付きます。さらに、参照されるすべての変数には、緑色のチェックマークの横にある依存関係アイコンがあります。アイコンを選択すると、依存関係マップが表示されます。依存関係マップは移動できます。マウスホイールまたは同等の機能を使用してズームインまたはズームアウトできます。

変数構文

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルタイトルとメトリクスクエリは、2 つの異なる構文を使用して変数を参照できます。

- `$varname` – この構文は参照しやすいですが、単語の途中で変数を使用することは許可されません。

  例: `apps.frontend.$server.requests.count`
- `${var_name}` – 式の中央で変数を使用する場合は、この構文を使用します。
- `${var_name:<format>}` – この形式により、Grafana が値を解釈する方法をより細かく制御できます。詳細については、「高度な変数フォーマットオプション」を参照してください。
- `[[varname]]` – 使用しないでください。この構文は古く、廃止されました。これは、今後のリリースで削除されます。

クエリがデータソースに送信される前に、クエリは[補間されます]。つまり、変数は現在の値に置き換えられます。補間中、変数値は、クエリ言語の構文と使用される場所に準拠するためにエスケープされる可能性があります。例えば、InfluxDB または Prometheus クエリの正規表現で使用される変数は、正規表現がエスケープされます。

高度な変数フォーマットオプション

変数補間のフォーマットはデータソースによって異なりますが、デフォルトの形式を変更する場合があります。

例えば、MySQL データソースのデフォルトは、'server01','server02' などのように、複数の値を引用符でカンマ区切りで結合することです。場合によっては、server01,server02 など、引用符なしでカンマ区切りの文字列を指定することもできます。これを行うには、以下にリストされている高度な変数フォーマットオプションを使用します。

一般的な構文

構文: ${var_name:option}

無効なフォーマットオプションが指定されている場合、glob がデフォルト/フォールバックオプションです。

CSV

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:csv}'
Interpolation result:  'test1,test2'
```

分散 - OpenTSDB

OpenTSDB のカスタム形式の複数の値を持つ変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:distributed}'
Interpolation result:  'test1,servers=test2'
```

二重引用符

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、\" によって各値の " を工スケープし、" で各値を引用符で囲みます。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:doublequote}'
Interpolation result:  '"test1","test2"'
```

Glob - Graphite

複数の値を持つ変数を glob にフォーマットします (Graphite クエリの場合)。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:glob}'
Interpolation result:  '{test1,test2}'
```

JSON

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:json}'
Interpolation result:  '["test1", "test2"]'
```

Lucene - Elasticsearch

Elasticsearch の Lucene 形式で複数の値を持つ変数をフォーマットします。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:lucene}'
Interpolation result:  '("test1" OR "test2")'
```

Percentencode

URL パラメータで使用する単一値変数と複数値変数をフォーマットします。

```
servers = [ 'foo()bar BAZ',  'test2' ]
String to interpolate:  '${servers:percentencode}'
Interpolation result:  'foo%28%29bar%20BAZ%2Ctest2'
```

パイプ

複数の値を持つ変数をパイプ区切り文字列にフォーマットします。

```
servers = [ 'test1.',  'test2' ]
String to interpolate:  '${servers:pipe}'
Interpolation result:  'test1.|test2'
```

Raw

SQL クエリにおける単一引用符など、データソース固有のフォーマットをオフにします。

```
servers = [ 'test.1',  'test2' ]
String to interpolate:  '${var_name:raw}'
Interpolation result:  'test.1,test2'
```

[正規表現]

複数の値を持つ変数を正規表現文字列にフォーマットします。

```
servers = [ 'test1.',  'test2' ]
String to interpolate:  '${servers:regex}'
Interpolation result:  '(test1\.|test2)'
```

一重引用符

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、\' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = [ 'test1',  'test2' ]
String to interpolate:  '${servers:singlequote}'
Interpolation result:  "'test1','test2'"
```

Sqlstring

単一値変数と複数値の変数をカンマ区切り文字列にフォーマットし、'' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = [ "test'1",  "test2" ]
String to interpolate:  '${servers:sqlstring}'
Interpolation result:  "'test''1','test2'"
```

[テキスト]

単一値変数と複数値変数をテキスト表現にフォーマットします。単一値の変数の場合、テキスト表現のみを返します。複数値変数の場合、テキスト表現は + と組み合わせて返されます。

```
servers = [ "test1",  "test2" ]
String to interpolate:  '${servers:text}'
Interpolation result:  "test1 + test2"
```

クエリパラメータ

単一値変数と複数値変数をクエリパラメータ表現にフォーマットします。例: `var-foo=value1&var-foo=value2`

```
servers = [ "test1",  "test2" ]
String to interpolate:  '${servers:queryparam}'
Interpolation result:  "var-servers=test1&var-servers=test2"
```

ダッシュボード使用状況の評価

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

使用状況のインサイトにより、Grafana インスタンスの使用方法をより深く理解できます。

使用状況のインサイト機能は、多数の集計データを収集し、データベースに保存します。

- ダッシュボードビュー (集計およびユーザーごと)
- データソースエラー
- データソースクエリ

集計データを使用すると、ダッシュボードとデータソースのインサイト、プレゼンスインジケータ、インサイトデータを使用したダッシュボードのソート、ダッシュボード内の使用状況インサイトデータの視覚化など、様々な機能にアクセスできます。

この機能を使用すると、Loki にエクスポートできる詳細なログも生成できます。

ダッシュボードインサイトとデータソースインサイト

すべてのダッシュボードとデータソースで、使用状況情報にアクセスできます。

ダッシュボードインサイト

ダッシュボードの使用状況情報を表示するには、上部のバーで [ダッシュボードインサイト] をクリックします。

ダッシュボードには以下の情報が表示されます。

- 統計: 過去 30 日間の毎日のクエリとエラーの数。
- ユーザーとアクティビティ: 過去 30 日間の日次ビュー数。ダッシュボードと最近のユーザー (制限は 20) に対する最後のアクティビティ。

データソースインサイト

データソースインサイトは、過去 30 日間にデータソースがどのように使用されたかに関する情報を提供します。

- 1 日あたりのクエリ
- 1 日あたりのエラー
- 1 日あたりのクエリロード時間 (平均ミリ秒)

データソースインサイトを検索するには:

1. データソースリストビューに移動します。
2. [データソース] をクリックします。
3. [インサイト] タブをクリックします。

プレゼンスインジケータ

サインインしてダッシュボードを表示すると、プレゼンスインジケーターによって、自分と同じダッシュボードを表示しているユーザーを知ることができます。プレゼンスインジケーターには、ダッシュボードを最近操作したユーザーのアバターが表示されます。デフォルトの時間枠は 10 分です。ユーザー名を表示するには、ユーザーのアバターにカーソルを合わせます。アバターは、ユーザーの E メールに基づいて Gravatar から取得されます。

ダッシュボードにプレゼンスインジケータに収まるユーザーよりもアクティブなユーザーがいる場合は、[+X] アイコンをクリックします。これを行うと、ダッシュボードインサイトが開きます。ダッシュボードインサイトには、最近のユーザーアクティビティに関する詳細が含まれます。

インサイトデータを使用したダッシュボードのソート

検索ビューでは、インサイトデータを使用して、最も使用されているダッシュボード、壊れているダッシュボード、未使用のダッシュボードを見つけることができます。

- エラー総数
- 30 日間のエラー
- ビューの合計
- 30 日間のビュー

Grafana バージョン 9 でのダッシュボードの検索

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボード名とパネルタイトルでダッシュボードを検索できます。ダッシュボードを検索すると、ダッシュボードの内容を表示するアクセス許可がない場合でも、システムは Grafana インスタンス内で使用可能なすべてのダッシュボードを返します。

ダッシュボード名を使用してダッシュボードを検索する

検索バーにダッシュボード名の一部を入力します。検索では、入力すると、文字列の部分一致の結果がリアルタイムで返されます。

ダッシュボード検索は次のとおりです。

- リアルタイム
- 大文字と小文字は区別されません。
- 保存済みダッシュボードとファイルベースのダッシュボード全体で機能します。

> Tip
>
> キーボードの矢印キーを使用して結果を移動し、`Enter` を押して選択したダッシュボードを開くことができます。

パネルタイトルを使用してダッシュボードを検索する

ダッシュボードに表示されるパネルのタイトルでダッシュボードを検索できます。パネルのタイトルが検索クエリと一致すると、ダッシュボードが検索結果に表示されます。

ダッシュボードの検索結果をタグでフィルタリングする

ダッシュボードの数が増えるほど、タグをつけることで効果的に整理することができます。ダッシュボード [設定] でタグを追加および管理できます。

複数のタグを選択すると、Grafana は選択したすべてのタグを含むダッシュボードを表示します。

タグでダッシュボード検索結果をフィルタリングするには、次のいずれかの手順を実行します。

- ダッシュボードの検索結果をタグでフィルタリングするには、検索結果の右側の列に表示されるタグを選択します。

  追加のタグを選択して、フィルタリングを続行できます。
- 使用可能なすべてのタグのリストを表示するには、[タグでフィルタリング]ドロップダウンメニューをクリックし、タグを選択します。

  すべてのタグが表示され、タグを選択すると、ダッシュボード検索が即座にフィルタリングされます。

> **Tip**
>
> キーボードのみを使用する場合は、tab キーを押して [タグでフィルタリング] ドロップダウンメニューに移動し、下矢印キーを押してメニューをアクティブ化し、タグを見つけ、Enter を押してタグを選択します。

Grafana バージョン 9 のパネルと視覚化

> **⚠** このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルは Grafana の基本的な視覚化構築ブロックです。各パネルには、パネルで選択されたデータソース特有のクエリエディタがあります。クエリエディタを使用すると、視覚化するデータを返すクエリを構築できます。

各パネルにはさまざまなスタイルとフォーマットのオプションがあります。パネルをドラッグ、ドロップ、サイズ変更して、ダッシュボード上で配置を変更できます。

パネルを追加する前に、データソースが設定されていることを確認してください。

追加のパネルタイプは、追加のプラグインをワークスペースにインストールすることで使用できます。

- 特定のデータソース使用の詳細については「データソースに接続する」を参照してください。

トピック

- パネルエディタの概要
- パネルオプションの設定
- 標準オプションを設定する
- データのクエリと変換
- しきい値を設定する
- データリンクの設定
- フィールドのオーバーライドを設定する
- 値マッピングを設定する
- 凡例を設定する
- 計算タイプ
- 視覚化に注釈を付ける
- パネル検査ビュー
- Grafana バージョン 9 で利用可能な視覚化

パネルエディタの概要

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このセクションでは、Grafana パネルエディタの領域について説明します。

- パネルヘッダー: ヘッダーセクションには、パネルが表示されるダッシュボードと、次のコントロールが一覧表示されます。
  - [ダッシュボード設定 (ギア) アイコン]: ダッシュボード設定へのアクセスをクリックします。
  - [破棄]: ダッシュボードを最後に保存してから行ったパネルへの変更が破棄されます。
  - [保存]: パネルに加えた変更を保存します。
  - [適用]: 行った変更が適用され、パネルエディタが閉じてダッシュボードに戻ります。ダッシュボードを保存して、適用された変更を保持する必要があります。
- 視覚化プレビュー: 視覚化プレビューセクションには、次のオプションが含まれています。
  - [テーブルビュー]: 視覚化をテーブルに変換して、データを表示します。テーブルビューはトラブルシューティングに役立ちます。このビューには未加工データのみが含まれます。データに適用した変換や、テーブルの視覚化で使用できる書式設定オプションは含まれません。
  - [塗りつぶし]: 視覚化プレビューで、使用可能なスペースを塗りつぶします。サイドペインの幅またはボトムペインの高さを変更すると、視覚化が変更され、使用可能なスペースが塗りつぶされます。
  - [実寸]: 視覚化プレビューが、ダッシュボードの実際のサイズで表示されます。十分なスペースがない場合、視覚化はアスペクト比が保持されたまま縮小されます。
  - [時間範囲コントロール] – [デフォルト]は、ブラウザのローカルタイムゾーンまたはより高度な設定で選択されたタイムゾーンです。
- データセクション: データセクションには、クエリの入力、データの変換、アラートルールの作成 (該当する場合) を行うタブがあります。
  - [クエリ] タブ – データソースを選択し、ここにクエリを入力します。
  - [変換 タブ]: データ変換を適用します。

- [アラート] タブ – アラートルールを入力します。
- パネル表示オプション: 表示オプションセクションには、データ視覚化のほぼすべての側面を設定するタブがあります。

パネル検査ドロワーを開く

検査ドロワーは、パネルの理解とトラブルシューティングに役立ちます。任意のパネルの未加工データの表示、データの CSV (comma-separated values: カンマ区切り) ファイルへのエクスポート、クエリリクエストの表示、JSON 形式のパネルとデータのエクスポートが可能です。

> 注: すべてのパネルタイプにタブがすべてあるわけではありません。例えば、ダッシュボードリストパネルには検査する未加工データがないため、[統計]、[データ]、[クエリ] タブは表示されません。

パネルインスペクターは、次のオプションで構成されます。

- パネル検査ドロワーは、右側にドロワーが開いて表示されます。右上隅の矢印をクリックして、ドロワーペインを展開または縮小します。
- [データ タブ] – 変換が適用されたクエリによって返される未加工データを表示します。オーバーライドや値マッピングなどのフィールドオプションは、デフォルトでは適用されません。
- [統計タブ] – クエリにかかる時間および返される量を示します。
- [JSON] タブ - パネル JSON、パネルデータ JSON、およびデータフレーム構造 JSON を、表示およびコピーできます。これは Grafana のプロビジョニングまたは管理に役立ちます。
- [クエリタブ] – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。
- [エラータブ] – エラーを表示します。クエリがエラーを返す場合にのみ表示されます。

パネルオプションの設定

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana パネルは、データソースクエリを定義し、視覚化に表示されるデータを変換およびフォーマットするために使用するユーザーインターフェイスです。

パネルエディタには、クエリビルダーと、データの変換やパネルへの情報の追加に使用できる一連のオプションが含まれています。

このトピックでは、以下の方法について説明します。

- パネルを開いて編集する
- パネルのタイトルと説明を追加する
- パネル JSON モデルを表示する
- 繰り返し行とパネルを追加する

パネルを編集する

ダッシュボードにパネルを追加すると、いつでもパネルを開いてクエリを変更または更新したり、データ変換を追加したり、視覚化設定を変更したりできます。

1. 編集するパネルを含むダッシュボードを開きます。
2. パネルのいずれかの部分にカーソルを合わせると、右上隅にアクションメニューが表示されます。
3. メニューをクリックし、[編集] を選択します。

   キーボードショートカットを使用してパネルを開くには、パネルにカーソルを合わせ、e を押します。

   パネルが編集モードで開きます。

パネルにタイトルと説明を追加する

パネルにタイトルと説明を追加して、視覚化に関する重要な情報をユーザーと共有します。例えば、説明を使用して、視覚化の目的を文書化します。

1. パネルを編集します。
2. パネル表示オプションペインで、[パネルオプション] セクションを見つけます。
3. [タイトル] を入力します。

このフィールドに入力されたテキストは、パネルエディタのパネル上部とダッシュボードに表示されます。

4. 表示するパネルとデータの説明を記入します。

   このフィールドに入力したテキストは、パネルの左上隅にあるツールチップに表示されます。

   [タイトル] と [説明] フィールドで[定義した変数]を使用できますが、[グローバル変数]は使用できません。

パネル JSON モデルを表示する

パネル、パネルデータ、データフレーム JSON モデルを検索してエクスポートします。

1. パネルを含むダッシュボードを開きます。
2. パネルのいずれかの部分にカーソルを合わせると、右上隅にアクションメニューが表示されます。
3. メニューをクリックして、[Inspect] (検査) > [Panel JSON] を選択します。
4. [ソースの選択] フィールドで、次のオプションのいずれかを選択します。
   - Panel JSON: パネルを表す JSON オブジェクトを表示します。
   - [Panel data]: パネルに渡されるデータを表す JSON オブジェクトを表示します。
   - [データフレーム構造]: 変換、フィールド設定、オーバーライド設定が適用された未加工の結果セットを表示します。
5. JSON を調べるには、[>] をクリックして JSON モデルの一部を展開または折りたたみます。

繰り返しパネルを設定する

パネルまたは行をダッシュボードに動的に追加するように Grafana を設定できます。動的パネルは、変数の値に基づいてシステムが作成するパネルです。変数は、ダッシュボード内のすべてのパネルでクエリを動的に変更します。

> **Note**
>
> パネルを繰り返すには、変数に 1 つ以上の項目を選択する必要があります。パネルを非表示にするためにパネルを 0 回繰り返すことはできません。

開始する前に:

- クエリに複数値変数が含まれていることを確認します。

繰り返しパネルを設定するには:

1. 繰り返すパネルを編集します。
2. 表示オプションペインで、[パネルオプション] > [繰り返しオプション] をクリックします。
3. [direction] (方向) を選択します。
   - [水平] を選択してパネルを並べて配置します。Grafana は繰り返しパネルの幅を調整します。現在、行上の他のパネルと繰り返しパネルを混在して使うことはできません。
   - [垂直] を選択して、列にパネルを配置します。繰り返しパネルの幅は、元の繰り返しパネルの幅と同じです。
4. すべてのパネルに変更を伝達するには、ダッシュボードを再読み込みします。

標準オプションを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana で使用されるデータモデルは、時系列とテーブルのクエリ結果の両方を統合する列指向のテーブル構造です。この構造内の各列は、フィールドと呼ばれます。フィールドは、単一の時系列またはテーブル列を表すことができます。

フィールドオプションを使用すると、視覚的表現でのデータの表示方法を変更できます。適用するオプションとオーバーライドはデータを変更せず、Grafana のデータの表示方法を変更します。オプションを変更すると、すべてのフィールド、つまりすべての系列または列に適用されます。例えば、単位をパーセンテージに変更すると、数値を含むすべてのフィールドがパーセンテージで表示されます。

フィールドの書式設定オプションの完全なリストについては、「標準オプション定義」を参照してください。

> **Note**
>
> ほとんどの組み込み Grafana パネルに標準オプションを適用できます。新しいパネルとデータモデルに更新されていない古いパネルとコミュニティパネルの一部には、これらのフィールドオプションのすべてまたは一部が欠落します。

1. ダッシュボードを開き、パネルタイトルをクリックし、[編集] をクリックします。
2. パネル表示オプションペインで、[標準オプション] セクションを見つけます。
3. 適用する標準オプションを選択します。
4. 変更をプレビューするには、編集中のフィールドオプションボックスの外部をクリックするか、Enter を押します。

標準オプションの定義

このセクションでは、使用可能なすべての標準オプションについて説明します。

ほとんどの組み込み Grafana パネルに標準オプションを適用できます。新しいパネルとデータモデルに更新されていない古いパネルとコミュニティパネルの一部には、これらのフィールドオプションのすべてまたは一部が欠落します。

ほとんどのフィールドオプションは、編集中のフィールドオプションボックスの外部をクリックするか、Enter キーを押すまで、視覚化には影響しません。

> **Note**
>
> すべての視覚化に対するオプションの追加と拡張に常に取り組んでいるため、すべてのオプションをすべての視覚化に対して使用できない場合があります。

[単位]

フィールドで使用する単位を選択できます。[単位] フィールドをクリックし、目的の単位が見つかるまでドリルダウンします。選択した単位は、時間を除くすべてのフィールドに適用されます。

ユーザー定義の単位

単位ドロップダウンを使用して、ユーザー定義の単位、接頭辞、または接尾辞と、日付時刻形式を指定することもできます。

ユーザー定義の単位を選択するには、単位を入力し、ドロップダウンで最後の [カスタム: xxx] オプションを選択します。

- suffix:<suffix> 値の後に続くユーザー定義単位を指定できます。
- prefix:<prefix> 値の前に置くユーザー定義単位を指定できます。
- time:<format> ユーザー定義の日付時刻形式の場合は、time:YYYY-MM-DD などのように入力します。形式構文とオプションについては、「形式」を参照してください。
- si:<base scale><unit characters> ユーザー定義の SI 単位を指定できます。例: si: mF。これは、単位とソースデータスケールの両方を指定できるため、少し高度です。したがって、ソースデータがミリ (基本単位の千分の一) で表されている場合、単位の先頭にその SI スケール文字が付けられます。
- count:<unit> ユーザー定義の集計単位を指定できます。
- currency:<unit> ユーザー定義の通貨単位を指定できます。

単位ピッカーにネイティブ絵文字を貼り付けて、ユーザー定義単位として選択することもできます。

[String units] (文字列の単位)

Grafana は、文字列を解析して数値として表示する際に、自動で表示が変わる場合があります。元の文字列値を表示するように Grafana を設定するには、フィールドオーバーライドを作成し、[文字列] 単位で単位プロパティを追加します。

最小

パーセンテージしきい値の計算に使用される最小値を設定できます。すべての系列とフィールドに基づいて自動計算する場合は、空白のままにします。

最大

パーセンテージしきい値の計算に使用される最大値を設定できます。すべての系列とフィールドに基づいて自動計算する場合は、空白のままにします。

小数

Grafana がレンダリングした値に含める小数点以下の桁数を指定します。このフィールドを空白のままにすると、Grafana は値に基づいて小数点以下の桁数を自動的に切り捨てます。例えば、1.1234 は 1.12 として表示され、100.456 は 100 として表示されます。

すべての小数を表示するには、単位を [String] (文字列) に設定します。

Display name (表示名)

すべてのフィールドの表示タイトルを設定できます。変数を使用できます。

複数の統計、フィールド、または系列が表示される場合、このフィールドは各統計のタイトルを制御します。${__field.name} などの式を使用して、タイトルの系列名またはフィールド名のみを使用できます。

「Temp」という名前のフィールドがあり、ラベルが {"Loc"="PBI", "Sensor"="3"} である場合

式の構文	例	レンダリング先	説明
${__field.displayName}	構文と同じ	Temp {Loc="PBI", Sensor="3"}	フィールド名とラベルがある場合は {} で表示します。レスポンスにラベルキーが 1 つしかない場合、ラベル部分において、Grafana は括弧なしでラベルの値を表示します。
${__field.name}	構文と同じ	Temp	フィールドの名前を表示します (ラベルなし)。
${__field.labels}	構文と同じ	Loc="PBI", Sensor="3"	ラベルを表示します (名前なし)。

式の構文	例	レンダリング先	説明
${__fielc .labels.〉 }	${__fie .labels oc}	PBI	指定されたラベルキーの値を表示します。
${__fielc .labels._ _values}	構文と同じ	PBI, 3	ラベルの値をカンマ区切りで表示します (ラベルキーなし)。

特定のフィールドに対する式をレンダリングした後に値が空の文字列である場合、デフォルトの表示方法が使用されます。

カラースキーム

カラーオプションとその視覚化への影響は、作業中の視覚化によって異なります。一部の視覚化には、異なるカラーオプションがあります。

値に基づいて、単一の色を指定するか、連続 (グラデーション) カラースキームを選択できます。連続色は、最小値と最大値に対する値の割合を使用して色を補間します。

次のいずれかのパレットを選択します。

カラーモード	説明
[単一色]	オーバーライドルールに役立つ単一色を指定します
[しきい値起点]	一致するしきい値から色を取るように Grafana に通知します
Classic palette (ク	Grafana は、系列のインデックスでパレットの色を検索して色を割り当てます。グラフや円グラフ、その他のカテゴリ別データの視覚化に役立ちます。

カラーモード	説明
ラシックパレット)	
Green-Yellow-Red (by value) (緑黄赤 (値別))	連続色スキーム
Blue-Yellow-Red (by value) (青黄赤 (値別))	連続色スキーム
Blues (by value) (青 (値別))	連続色スキーム (パネルの背景を青に)
Reds (by value) (赤 (値別))	連続色スキーム (パネルの背景色を青に)
Greens (by value) (緑 (値別))	連続色スキーム (パネルの背景色を青に)
Purple (by value) (紫 (値別))	連続色スキーム (パネルの背景色を青に)

値なし

フィールド値が空または null の場合に Grafana が表示するものを入力します。デフォルト値はハイフン (-) です。

データのクエリと変換

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は、さまざまなタイプのデータソース をサポートしています。データソース[クエリ]は、Grafana が変換および視覚化できるデータを返します。各データソースは独自のクエリ言語を使用し、データソースプラグインはそれぞれクエリエディタと呼ばれるクエリ構築ユーザーインターフェイスを実装します。

クエリについて

Grafana パネルは、クエリを介してデータソースと通信します。このクエリは、視覚化用のデータを取得します。クエリは、データソースで使用されるクエリ言語で記述された質問です。

パネルのデータソースオプションで、クエリの頻度とデータ収集の制限を設定できます。Grafana は、パネルごとに最大 26 個のクエリをサポートします。

各データソースのクエリ言語の詳細については、「データソース」セクションを参照してください。

クエリエディタ

各データソースのクエリエディタには、独自の機能を活用するクエリの作成に役立つカスタマイズされたユーザーインターフェイスが用意されています。

クエリ言語の違いにより、各データソースクエリエディタの外観と関数は異なります。データソースによっては、クエリエディタが自動補完機能、メトリクス名、変数の提案、ビジュアルクエリ構築インターフェイスを提供する場合があります。

特定のデータソースの一意のクエリエディタ機能の詳細については、以下の各ドキュメントを参照してください。

- Grafana に含まれるデータソースについては、「組み込みデータソース」を参照してください。
- Grafana Enterprise の編集に含まれるデータソースについては、「Enterprise データソースに接続する」を参照してください。

クエリ構文

データソースは、異なるクエリ言語を使用してデータをリクエストします。特定のデータソースの一意のクエリ言語の詳細については、各ドキュメントを参照してください。

[PostgreSQL の例]:

```
SELECT hostname FROM host WHERE region IN($region)
```

PromQL の例:

```
query_result(max_over_time(<metric>[${__range_s}s]) != <state>)
```

[特別なデータソース]

Grafana には、[Grafana]、[混合]、および [ダッシュボード] の 3 つの特別なデータソースも含まれています。詳細については、「データソース」を参照してください。

クエリタブを操作する

パネルの [クエリ] タブは、次の要素で構成されます。

- [データソースセレクタ] – クエリが実行されるデータソースを選択します。
- [クエリオプション] — 最大データ取得パラメータとクエリ実行時間間隔を設定します。
- [クエリインスペクターボタン] — クエリインスペクターパネルを開き、クエリを表示および最適化できます。
- [クエリエディタリスト]: – 作成したクエリを一覧表示します。
- [式]: – 式ビルダーを使用してアラート式を作成します。式の詳細については、「クエリ式の書き込み」を参照してください。

クエリの追加

クエリは、Grafana がダッシュボードパネルで視覚化するデータを返します。パネルを作成すると、Grafana は自動的にデフォルトのデータソースを選択します。

クエリを追加するには

1. クエリを追加するパネルを編集します。
2. [クエリ] タブを選択します。

3. [データソース] ドロップダウンメニューを選択し、データソースを選択します。
4. [クエリオプション] を選択して、必要なデータポイントの最大数を設定します。クエリオプションの詳細については、「クエリオプション」を参照してください。
5. クエリエディタを使用してクエリを書き込みます。
6. [適用] を選択します。

Grafana はデータソースにクエリを実行し、データを視覚化します。

クエリの管理

Grafana は、クエリを折りたたみ可能なクエリ行に整理します。各クエリ行にはクエリエディタが含まれ、文字 (A、B、C など) で識別されます。

クエリを管理するには、クエリのコピー、クエリの非表示、クエリの削除、クエリの順序変更、クエリエディタのヘルプの切り替えを行うことができます。

クエリオプション

データソースセレクタの横にある [クエリオプション] を選択すると、選択したデータソースの設定が表示されます。ここで行った変更は、このパネルで行われたクエリにのみ影響します。

Grafana は、ダークグレーテキストで表示されるデフォルトを設定します。変更はホワイトテキストで表示されます。フィールドをデフォルト設定に戻すには、フィールドからホワイトテキストを削除します。

パネルデータソースのクエリオプションには以下が含まれます。

- [最大データポイント数] – データソースがサポートしている場合、返される各系列の最大データポイント数を設定します。クエリが最大データポイント設定よりも多くのデータポイントを返す場合、データソースは、平均、最大、または別の関数でデータポイントを集約することで、返されるポイント数を減らします。

  ポイントの数を制限して、クエリのパフォーマンスを向上させたり、視覚化された行を平滑にしたりできます。デフォルト値はグラフの幅 (またはピクセル数) です。グラフパネルに表示されるスペースと同じ数のデータポイントしか視覚化できないためです。

  ストリーミングデータでは、Grafana はローリングバッファの最大データポイント値を使用します。ストリーミングはデータの継続的なフローであり、バッファリングはストリームをチャンクに分割します。例えば、Loki はライブテーリングモードでデータをストリーミングします。

- [最小間隔] – 自動計算された間隔の最小制限を設定します。これは通常、最小スクレイプ間隔です。データポイントを 15 秒ごとに保存する場合、それよりも間隔を短くしてもメリットはありません。また、スクレイプ間隔よりも低い最小値に設定して、粗く適切に機能するクエリを取得することもできます。
- [間隔] – データポイントを時間別に集計またはグループ化するときに使用できる時間範囲を設定します。

  Grafana は、テンプレートクエリで変数として使用できる適切な間隔を自動的に計算します。変数は、秒 (`$__interval`) またはミリ秒 (`$__interval_ms`) で測定されます。

  間隔は通常、合計や平均などの集計関数で使用されます。例えば、これは間隔変数 `rate(http_requests_total[$__interval])` を使用する Prometheus クエリです。

  この自動間隔は、グラフの幅に基づいて計算されます。ユーザーがグラフを縮小すると、間隔が大きくなり、より粗視化された集計になります。同様に、ユーザーが拡大すると、間隔が短くなり、より高精細な集計になります。

  詳細については、「グローバル変数」を参照してください。
- [相対時間] – 個々のパネルの相対時間範囲をオーバーライドします。これにより、パネルは、ダッシュボードの右上隅にあるダッシュボードタイムピッカーで選択されている時間とは異なります。これを使用して、同じダッシュボードに異なる期間または日数のメトリクスを表示できます。

> **Note**
>
> ダッシュボードの時間範囲が絶対時間である場合、パネルの時間オーバーライドは効果がありません。

例	相対時間フィールド
過去 5 分間	`now-5m`
今日のこれまでのところ	`now/d`
過去 5 日間	`now-5d/d`
今週始めから現在まで	`now/w`

例	相対時間フィールド
過去 2 年間	now-2y/y

- [タイムシフト] – タイムピッカーに対して開始と終了をシフトすることで、個々のパネルの時間範囲をオーバーライドします。例えば、パネルの時間範囲をダッシュボードのタイムピッカーより 2 時間早くシフトできます。

> **Note**
> ダッシュボードの時間範囲が絶対時間である場合、パネルの時間オーバーライドは効果がありません。

例	タイムシフトフィールド
過去丸 1 週間	1w/w
丸 2 週間前	2w/w
過去丸 1 か月間	1M/M
この丸 1 年	1d/y
過去丸 1 年間	1y/y

- [キャッシュタイムアウト] – (データソースで使用可能な場合のみ表示) 時系列ストアにクエリキャッシュがある場合、デフォルトのキャッシュタイムアウトをオーバーライドします。この値を秒単位で数値として指定します。

クエリ式の書き込み

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

サーバー側の式を使用すると、演算やその他のオペレーションでクエリから返されるデータを操作できます。式は新しいデータを作成し、データソースから返されるデータを操作しません。

式について

サーバー側の式を使用すると、演算やその他のオペレーションでクエリから返されるデータを操作できます。式は新しいデータを作成し、式のデータを許容できる入力にするためにマイナーなデータ再構築を行う以外は、データソースから返されるデータを操作しません。

式の使用

式は主に Grafana アラートによって使用されます。処理はサーバー側で行われるため、式はブラウザセッションなしで動作できます。ただし、式はバックエンドデータソースや視覚化でも使用できます。

> **Note**
>
> 式は、レガシーダッシュボードアラートでは機能しません。

式は、異なるデータソースからのクエリを組み合わせて有効にするか、データソースで使用できない演算を提供することで、データソースを補強することを目的としています。

> **Note**
>
> 可能であれば、データソース内でデータ処理を行う必要があります。ストレージから Grafana サーバーにデータをコピーして処理することは非効率であるため、式は軽量データ処理を対象としています。

式は、時系列または数値データを返すデータソースクエリで機能します。また、多次元データでも動作します。例えば、複数の系列を返すクエリでは、各系列がラベルまたはタグで識別されます。

個々の式は 1 つ以上のクエリまたはその他の式を入力として受け取り、結果にデータを追加します。個々の式またはクエリは、RefID と呼ばれる名前付き識別子 (例: デフォルト文字 A または B) である変数で表されます。

個々の式または別の式にあるデータソースクエリの出力を参照するには、この識別子を変数として使用します。

式のタイプ

式は 2 種類のデータで動作します。

- 時系列のコレクション。
- 数値のコレクション (各数値は項目)。

各コレクションは、単一のデータソースクエリまたは式から返され、RefID で表されます。各コレクションはセットであり、セット内の各項目は、ラベルまたはキーと値のペアとして保存されるディメンションによって一意に識別されます。

データソースクエリ

サーバー側の式は、バックエンドデータソースのデータソースクエリのみをサポートします。データは一般に、時系列データにラベル付けされているものと見なされます。将来的には、式が エラーをより適切に処理できるように、クエリの戻り値の型 (数値または時系列) データのアサーションを追加する予定でした。

データソースクエリは、式とともに使用されると、式エンジンによって実行されます。これを行うと、データはデータフレームごとに 1 つの時系列または 1 つの数値に再構成されます。例えば、テーブルビューの 1 つのフレームで複数の系列を返すデータソースを使用する場合、式で実行したときに異なるように見えることがあります。

現在、データフレームを使用する場合にサポートされる時系列以外の形式 (数値) は、時間、文字列の列、1 つの数値列のないデータフレームを返すテーブルレスポンスのみです。

Loc	ホスト	Avg_CPU
MIA	A	1
NYC	B	2

上記の例では、式で動作する数値を生成します。文字列の列はラベルになり、数値列は対応する値になります。例えば、`{"Loc": "MIA", "Host": "A"}` の値は 1 です。

オペレーション

式には math、reduce、リサンプリング演算を使用できます。

数値演算

Math は、時系列または数値データに関する自由形式の数式です。Math 演算子では、数値と時系列を入力として受け取り、異なる数値と時系列に変更します。

他のクエリまたは式からのデータは、$A などのドル記号のプレフィックスが付いた RefID で参照されます。変数の名前にスペースがある場合は、${my variable} のような中括弧構文を使用できます。

数値定数は、10 進数 (2.24)、8 進数 (072 のような先頭がゼロ)、または 16 進数 (0x2A のような先頭が 0x) にすることができます。指数と記号もサポートされています (例: -0.8e-2)。

演算子

算術演算子 (+、二項および単項 -、*、/、%、指数 **)、リレーショナル演算子 (<、>、==、!=、>=、<=)、論理演算子 (&&、||、および単項 !) がサポートされています。

データに対する演算処理は、数値データか時系列データかによって異なります。

$A + $B や $A || $B などの二項演算では、データのタイプに応じて、演算子は次の方法で適用されます。

- $A と $B の両方が数値の場合、2 つの数値の間で演算が実行されます。
- 1 つの変数が数値で、もう 1 つの変数が時系列の場合、時系列の各ポイントの値と数値との間の演算が実行されます。
- $A と $B の両方が時系列データである場合、2 つの系列の各値間の演算は、$A と $B の両方に存在するタイムスタンプごとに実行されます。Resample 演算を使用してタイムスタンプを並べることができます。

まとめ

- 数値 OP 数値 = 数値
- 数値 OP 系列 = 系列
- 系列 OP 系列 = 系列

式は 1 つの変数で表される複数の系列または数値で動作するため、二項演算は 2 つの変数間の統合 (結合) も実行します。これは、個々の系列または数値に関連付けられた識別ラベルに基づいて行われます。

したがって、$A にある {host=web01} のようなラベルを持つ数値と、同じラベルを持つ $B に別の数値がある場合、各変数内のこれら 2 つの項目間で演算が実行され、結果は同じラベルを共有します。この統合の動作のルールは次のとおりです。

- ラベルのない項目は、すべてに結合されます。
- $A と $B の両方に 1 つの項目 (1 つの系列、または 1 つの数値) しか含まれていない場合、それらは結合されます。
- ラベルが正確な数値演算である場合、それらは結合されます。
- ラベルが他のサブセットである場合、例えば、$A の項目には {host=A,dc=MIA} ラベルが付けられ、$B の項目には {host=A} ラベルが付けられると結合されます。
- 項目ごとに異なるタグキーがある $A などの変数内では、結合動作は不定です。

リレーショナル演算子と論理演算子は、偽の場合は 0、真の場合は 1 を返します。

数値演算関数

ほとんどの関数は独自の式の演算にありますが、数値演算関数には、数学演算子や数学記号に似た関数がいくつかあります。関数が数値または系列のいずれかを取ることができる場合、引数と同じタイプが返されます。系列の場合、系列内の各ポイントの値に対して演算が実行されます。

abs

abs は、引数の絶対値を返します。これは、数値または系列です。例えば、abs(-1)、abs($A) などです。

is_inf

is_inf は数値または系列を取得し、Inf 値 (負または正) の 1 と他の値の 0 を返します。例えば、is_inf($A) などです。

> **Note**
>
> 例えば、負の無限大を具体的にチェックする必要がある場合は、$A == infn() のような比較を行うことができます。

is_nan

is_nan は数値または系列を取得し、NaN 値の 1 と他の値の 0 を返します。例えば、is_nan($A) などです。この関数は、NaN が NaN と等しくないために存在します。

[is_null]

is_null は数値または系列を取得し、null 値の 1 と他の値の 0 を返します。例えば、is_null($A) などです。

is_number

is_number は数値または系列を取得し、すべての実数値の 1 と他の値 (null、Inf+、Inf-、および NaN) の 0 を返します。例えば、is_number($A) などです。

log

Log は、引数の自然対数を返します。これは、数値または系列です。値が 0 未満の場合、NaN が返されます。例えば、log(-1)、log($A) などです。

[inf、infn、nan、null]

inf、infn、nan、null 関数はすべて、名前の単一の値を返します。これらは主にテスト用です。例えば、null() などです。

round

Round は、丸められた整数値を返します。例えば、round(3.123)、round($A) です。

ceil

Ceil は、数値を最も近い整数値に切り上げます。例えば、ceil(3.123) は 4 を返します。

floor

Floor は、数値を最も近い整数値に切り下げます。例えば、floor(3.123 は 3 を返します。

[削減]

Reduce は、クエリまたは式から返される 1 つ以上の時系列を取得し、各系列を 1 つの数値に変換します。時系列のラベルは、それぞれ出力される削減された数値のラベルとして保持されます。

[フィールド]

- [関数] – 使用する削減関数
- [入力] – リサンプリングする変数 (refID (A など))
- [モード] – 系列に数値以外の値が含まれている場合、削減関数の制御動作を許可します (null、NaN 、+-Inf)

削減関数

Count

Count は、各系列内のポイント数を返します。

Mean

Mean は、各系列のすべての値の合計を、その系列のポイント数で割った値を返します。strict モードでは、系列内の値が null または nan の場合、または系列が空の場合、NaN が返されます。

[最小 と 最大]

最小 と 最大は、系列内の最小値または最大値をそれぞれ返します。strict モードでは、系列内の値が null または nan の場合、または系列が空の場合、NaN が返されます。

Sum

Sum は、系列のすべての値の合計を返します。系列の長さがゼロの場合、合計は 0 になります。strict モードでは、系列内の値に NaN または Null がある場合、NaN が返されます。

Last

Last は、系列内の最後の数値を返します。系列に値がない場合は、NaN を返します。

制限モード

Strict

Strict モードでは、入力系列はそのまま処理されます。系列内の値が数値以外の場合 (null、NaN、または +-Inf)、NaN が返されます。

Drop Non-Numeric

このモードでは、入力系列内のすべての非数値 (null、NaN、または +-Inf) は、制限関数を実行する前に除外されます。

Replace Non-Numeric

このモードでは、数値以外の値はすべて事前定義された値に置き換えられます。

[リサンプリング]

リサンプリングでは、各時系列のタイムスタンプが一貫した時間間隔になるように変更されます。主なユースケースは、同じタイムスタンプを共有しない時系列をリサンプリングして、それらの間で演算を実行できるようにすることです。これは、2 つの系列をそれぞれリサンプリングし、リサンプリングされた変数を参照する Math 演算で行うことができます。

[フィールド]

- [入力] – リサンプリングする時系列データの変数 (refID (`A` など))
- [リサンプリング先] – リサンプリング先となる期間は、例えば、`10s`. 単位は `s` 秒、`m` 分、`h` 時間、`d` 日、`w` 週、`y` 年です。
- [ダウンサンプリング] – ウィンドウサンプルごとに複数のデータポイントがある場合に使用する制限関数。動作の詳細については、「制限関数」を参照してください。
- [アップサンプリング] – データポイントがないウィンドウサンプルを埋めるために使用するメソッド。
  - pad で最後の既知の値に置換
  - backfill で次の既知の値に置換
  - fillna で空のサンプルウィンドウを NaN に置換

式の書き込み

データソースがサポートしている場合、Grafana は [式] ボタンを表示し、クエリエディタリストに既存の式を表示します。

式を記述するには

1. パネルを開きます。
2. クエリの下で、[式] を選択します。
3. [演算] フィールドで、書き込む式のタイプを選択します。

4. 式を書き込みます。
5. [適用] を選択します。

特殊なケース

クエリされたデータソースが系列や数値を返さない場合、式エンジンは NoData を返します。例えば、式によってマージされた 2 つのデータソースクエリがリクエストに含まれている場合、NoData が少なくとも 1 つのデータソースクエリによって返された場合、クエリ全体で返される結果は NoData になります。Grafana アラートが NoData 結果を処理する方法の詳細については、「データがない場合やエラーが発生した場合の対処」を参照してください。

別のパネルとクエリ結果を共有する

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、ダッシュボード内にある他の任意のパネルに対して、1 つのパネルからのクエリ結果を使用できます。パネル間でクエリ結果を共有すると、データソースに対して実行されるクエリの数が減少し、ダッシュボードのパフォーマンスが向上します。

ダッシュボードのデータソースを使用すると、結果を共有するクエリを含むダッシュボード内のパネルを選択できます。パネルごとに個別のクエリを送信する代わりに、Grafana は 1 つのクエリを送信し、他のパネルはクエリ結果を使用して視覚化を構築します。

この戦略は、例えば、同じデータを視覚化する複数のパネルがある場合に実行されるクエリの数を大幅に減らすことができます。

クエリ結果を共有するには

1. ダッシュボードを作成します。
2. タイトルを「Source panel」に変更します。このパネルを他のパネルのソースとして使用します。

3. 共有するクエリを (複数) 定義します。

   利用可能なデータソースがない場合は、[TestData] データソースを使用します。このデータソースは、テストに使用できるランダムな時系列を返します。

4. 2 番目のパネルを追加し、クエリエディタで [ダッシュボード] データソースを選択します。

5. [パネルリストから結果を使用する] で、最初に作成したパネルを選択します。

ソースパネルで定義されたすべてのクエリが新しいパネルで使用できるようになりました。ソースパネルで行われたクエリは、複数のパネルと共有できます。

任意のクエリをクリックして、そのクエリが定義されたパネルに移動できます。

[データを変換する]

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

変換は、システムが視覚化を適用する前に、クエリによって返されるデータを操作できる強力な方法です。変換を使用すると、次のことができます。

- フィールド名の変更
- 時系列データの結合
- クエリ間で数学演算の実行
- ある変換の出力を別の変換への入力として使用

同じデータセットの複数のビューに依存するユーザーにとって、変換は多数のダッシュボードを作成および維持する効率的な方法を提供します。

また、ある変換の出力を別の変換への入力として使用して、パフォーマンスを向上させることもできます。

> **Note**
>
> システムが変換されたデータをグラフ化できない場合があります。その場合は、視覚化の上にある [テーブルビュー] のトグルをクリックして、データのテーブルビューに切り替えます。これは、変換の最終結果を理解するのに役立ちます。

変換タイプ

Grafana には、データを変換するさまざまな方法があります。変換関数の完全なリストを以下に示します。

変換の順序

複数の変換がある場合、Grafana はリスト順で変換を適用します。各変換では、処理中のパイプラインにある次の変換に渡される新しい結果セットが作成されます。

Grafana が変換を適用する順序は、結果に直接影響します。例えば、縮小変換を使用して 1 つの列のすべての結果を 1 つの値に集約する場合、変換はその 1 つの値にのみ適用できます。

変換関数をデータに追加する

次の手順は、データに変換を追加する方法を示しています。

パネルに変換を追加するには

1. 1 つ以上の変換を追加するパネルに移動します。
2. パネルタイトルを選択し、[編集] をクリックします。
3. [変換] タブを選択します。
4. 変換を選択します。変換オプションを設定する変換行が表示されます。
5. [変換ルールの追加] を選択すると、別の変換ルールを追加できます。この変換は、前の変換が返した結果セットに作用します。

変換をデバッグする

変換の入力および出力結果のセットを表示するには、変換行の右側にあるバグアイコンを選択します。

入出力の結果セットにより、変換のデバッグが可能になります。

変換を削除する

不要な変換を削除することをお勧めします。変換を削除すると、視覚化からデータが削除されます。

前提条件:

変換に依存するすべてのダッシュボードを特定し、影響を受けるダッシュボードのユーザーに通知します。

変換を削除するには

1. パネルを開いて編集します。
2. [変換] タブを選択します。
3. 削除する変換の横にあるごみ箱アイコンを選択します。

変換関数

データに関する次の変換を実行できます。

計算からのフィールドの追加

この変換を使用して、他の 2 つのフィールドから計算した新しいフィールドを追加します。各変換では 1 つの新しいフィールドを追加できます。

- [モード] – 以下のモードを選択します。
  - [行を減らす] – 選択したフィールドの各行に、選択済みの計算を個別に適用します。
  - [二項演算] – 選択した 2 つのフィールドから 1 行の値に基本的な二項演算 (合計や乗算など) を適用します。
- [Field name] – 新しいフィールドの計算に使用する、フィールドの名前を選択します。
- [計算] – [掃き出し法] を選択すると、[計算] フィールドが表示されます。フィールドをクリックして、新しいフィールドの作成に使用できる計算選択肢のリストを表示します。使用可能な計算の詳細については、「計算タイプ」を参照してください。
- [演算] – [二項演算] モードを選択すると、[演算] フィールドが表示されます。これらのフィールドを使用すると、2 つの選択したフィールドから 1 行の値に対して基本的な数学演算を実行できます。二項演算に数値を使用することもできます。
- [エイリアス] – (オプション) 新しいフィールドの名前を入力します。このフィールドを空白のままにすると、計算に一致するようにフィールドの名前が付けられます。

- [すべてのフィールドを置き換え] – (オプション) 他のすべてのフィールドを非表示にし、計算したフィールドのみを視覚化に表示する場合は、このオプションを選択します。

[連結フィールド]

この変換は、全フレームの全フィールドを 1 つの結果に結合します。この 2 つのクエリを検討します。

クエリ A

温度	アップタイム
15.4	1230233

クエリ B

AQI	エラー
3.2	5

フィールドを連結すると、データフレームは次のようになります。

温度	アップタイム	AQI	エラー
15.4	1230233	3.2	5

[クエリ結果から設定]

この変換を使用して、1 つのクエリを選択し、そこから [最小] 、[最大] 、[単位]、[しきい値] などの標準オプションを抽出し、他のクエリ結果に適用できます。これにより、動的なクエリ駆動型の視覚化設定が可能になります。

設定クエリ結果の各行に対して一意の設定を抽出する場合は、行からフィールドへの変換を試行します。

オプション

- [Config query] (設定クエリ) – 設定として使用するデータを返すクエリを選択します。
- [適用先] – 設定を適用するフィールドまたは系列を選択します。
- [オプションに適用] – 通常は、[適用先] で選択したオプションに応じて、フィールドタイプまたはフィールド名の正規表現を使用します。

[フィールドタイプの変換]

この変換により、指定されたフィールドのフィールドタイプが変更されます。

- [Field] – 使用可能なフィールドから選択
- [as] – 変換するフィールドタイプを選択
  - [Numeric] – 値の数値の作成を試行
  - [String] – 値の文字列を作成
  - [時間] – 値を時間として解析
    - yyyy-mm-dd や DD MM YYYY hh:mm:ss などの文字列の入力として DateFormat を指定するオプションを表示
  - [Boolean] – 値はブール値となる

例えば、次のクエリでは、時間フィールドを Time として、日付形式を YYYY として選択することで変更できます。

時間	Mark	値
7/1/2017	above	25
8/2/2018	below	22
9/2/2019	below	29
10/4/2020	above	22

結果

時間	Mark	値
1/1/2017	above	25
1/1/2018	below	22
1/1/2019	below	29
1/1/2020	above	22

[名前によるデータのフィルタリング]

この変換を使用してクエリ結果の一部を削除します。

Grafana は [Identifier] (識別子) フィールドを表示し、続いてクエリによって返されるフィールドを表示します。

フィルターは次の 2 つのいずれかの方法で適用できます。

- 正規表現を入力します。
- フィールドを選択してフィールドのフィルタリングを切り替えます。フィルタリングされたフィールドにはダークグレーテキストが表示され、フィルタリングされていないフィールドには白いテキストが表示されます。

[クエリによるデータのフィルタリング]

1 つ以上のクエリを非表示にするには、複数のクエリがあるパネルでこの変換を使用します。

Grafana では、クエリ識別文字がダークグレーテキストで表示されます。フィルタリングを切り替えるには、クエリ識別子をクリックします。クエリ文字が白い場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示です。

> **Note**
>
> この変換は Graphite では使用できません。このデータソースは、返されたデータとクエリの相関関係をサポートしていないためです。

[値によるデータのフィルタリング]

この変換により、Grafana でデータを直接フィルタリングして、クエリ結果から一部のデータポイントを削除できます。1 つ以上の定義済み条件と一致するデータを含めるか除外するかを選択できます。その条件は選択したフィールドに適用されます。

この変換はデータソースが値によりネイティブにフィルタリングしない場合に役立ちます。共有クエリを使用している場合、これを使用して値を絞り込んで表示することもできます。

すべてのフィールドで使用できる条件は次のとおりです。

- [Regex] – 正規表現を照合
- [Is Null] – 値が null の場合に照合
- [Is Not Null] – 値が null でない場合に照合します。
- Equal – 値が指定値と等しい場合に照合
- Different – 値が指定された値と異なる場合に照合

数値フィールドで使用可能な条件は以下のとおりです。

- Greater – 値が指定値より大きい場合に照合
- Lower – 値が指定値よりも小さい場合に照合
- Greater or equal – 値が指定値以上の場合に照合
- Lower or equal – 値が指定値以下の場合に照合
- Range – 指定された最小値と最大値 (最小値と最大値を含む) 間の範囲を照合します。

以下のデータセットを考慮します。

時間	温度	高度
7/7/2020 午前 11 時 34 分 23 秒	32	101
7/7/2020 午前 11 時 34 分 22 秒	28	125
7/7/2020 午前 11 時 34 分 21 秒	26	110

時間	温度	高度
7/7/2020 午前 11 時 34 分 20 秒	23	98
7/7/2020 午前 10 時 32 分 24 秒	31	95
7/7/2020 午前 10 時 31 分 22 秒	20	85
7/7/2020 午前 9 時 30 分 57 秒	19	101

温度が 30°C 未満のデータポイントを [含める] にすると、設定は次のようになります。

- フィルタータイプ – `Include`
- [条件] – `Temperature` が `Lower Than 30` と一致する行

30°C 未満の温度のみが含まれる場合、次の結果が得られます。

時間	温度	高度
7/7/2020 午前 11 時 34 分 22 秒	28	125
7/7/2020 午前 11 時 34 分 21 秒	26	110
7/7/2020 午前 11 時 34 分 20 秒	23	98
7/7/2020 午前 10 時 31 分 22 秒	20	85
7/7/2020 午前 9 時 30 分 57 秒	19	101

複数の条件をフィルターに追加できます。例えば、高度が 100 を超える場合にのみデータを含めることができます。これを行うには、次の設定にその条件を追加します。

- フィルタータイプ – `Match All` 条件を満たす `Include` 行
- Condition 1 – `Temperature` が `30` より `Lower` に一致する行
- Condition 2 – `Altitude` が `Greater` より `100` に一致する行

複数の条件がある場合、条件にすべて一致、または追加した条件のいずれかに一致する行にアクション (包含/除外) を適用するかを選択できます。

上記の例では、温度が 30°C 未満かつ高度が 100 を超える行を含めるため、[すべて一致] を選択しました。温度が 30°C 未満または高度が 100 を超える行を含める場合は、[任意の一致] を選択します。これには、元のデータの最初の行が含まれます。元のデータの温度は 32°C (最初の条件と一致しない) ですが、高度は 101 (2 番目の条件と一致する) であるため、含まれます。

無効または不完全な設定条件は無視されます。

[グループ化の条件]

この変換では、指定されたフィールド (列) 値でデータをグループ化し、各グループの計算を処理します。計算選択肢のリストの表示をクリックします。

オリジナルデータの例を次に示します。

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン
7/7/2020 午前 11 時 34 分 20 秒	サーバー 3	62	OK
7/7/2020 午前 10 時 32 分 20 秒	サーバー 2	90	オーバーロード
7/7/2020 午前 10 時 31 分 22 秒	サーバー 3	55	OK

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 9 時 30 分 57 秒	サーバー 3	62	再起動
7/7/2020 午前 9 時 30 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 28 分 6 秒	サーバー 1	80	OK
7/7/2020 午前 9 時 25 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK

この変換では 2 つのステップを実行します。まず、データをグループ化させる 1 つ以上のフィールドを指定します。これにより、これらのフィールドの同じ値すべてが、ソートされたかのようにグループ化されます。例えば、サーバー ID フィールドでグループ化すると、次のようにデータがグループ化されます。

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン
7/7/2020 午前 9 時 28 分 6 秒	サーバー 1	80	OK
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK
7/7/2020 午前 10 時 32 分 20 秒	サーバー 2	90	オーバーロード
7/7/2020 午前 9 時 30 分 5 秒	サーバー 2	88	OK

時間	サーバー ID	CPU 温度	サーバーステータス
7/7/2020 午前 9 時 25 分 5 秒	サーバー 2	88	OK
7/7/2020 午前 11 時 34 分 20 秒	サーバー 3	62	OK
7/7/2020 午前 10 時 31 分 22 秒	サーバー 3	55	OK
7/7/2020 午前 9 時 30 分 57 秒	サーバー 3	62	再起動

同じ値のサーバー ID を持つすべての行がグループ化されます。

データをグループ化するフィールドを選択すると、他のフィールドにさまざまな計算を追加でき、その計算を行の各グループに適用できます。例えば、これらの各サーバーの平均 CPU 温度を計算できます。そのため、CPU 温度フィールドに適用される[平均]計算を追加して、以下を取得できます。

サーバー ID	CPU 温度 (平均)
サーバー 1	82
サーバー 2	88.6
サーバー 3	59.6

また、複数の計算を追加できます。例:

- 時間のフィールドでは、[最後]の値を計算して、各サーバーで最後のデータポイントがいつ受信されたかを把握できます。
- サーバーステータスフィールドでは、[最後]の値を計算して、サーバーごとに最後の状態値を確認できます。
- また、温度フィールドでは、最後の値を計算して、サーバーごとに最新の測定温度を知ることもできます。

その後、以下を取得します。

サーバー ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最後)	サーバーステータス (最後)
サーバー 1	82	80	7/7/2020 午前 11 時 34 分 20 秒	シャットダウン
サーバー 2	88.6	90	7/7/2020 午前 10 時 32 分 20 秒	オーバーロード
サーバー 3	59.6	62	7/7/2020 午前 11 時 34 分 20 秒	OK

この変換により、時系列からキー情報を抽出して便利な方法で表示できます。

Join by field (フィールドで結合)

この変換を使用して、複数の結果を 1 つのテーブルに結合します。これは、複数の時系列の結果を共有時間フィールドを持つ単一のワイドテーブルに変換する場合に特に便利です。

Inner join (内部結合)

内部結合は、選択したフィールドからすべてのテーブルが同じ値を共有する複数のテーブルからのデータをマージします。このタイプの結合では、すべての結果で値が一致しないデータを除外します。

この変換を使用して、複数のクエリの結果 (渡された結合フィールドまたは最初の列の組み合わせ) を 1 つの結果に結合し、正常な結合を実行できない行を削除します。

次の例では、2 つのクエリがテーブルデータを返します。内部結合変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233

クエリ B

時間	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	サーバー 2	5
7/7/2020 午前 11 時 4 分 20 秒	サーバー 3	10

内部結合変換を適用した後の結果は次のようになります。

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233	サーバー 2	5

Outer join (外部結合)

外部結合には、すべての入力で値が一致しない内部結合と行のすべてのデータが含まれます。内部結合が時間フィールドでクエリ A とクエリ B を結合している間、外部結合には時間フィールドで一致しないすべての行が含まれます。

次の例では、2 つのクエリがテーブルデータを返します。外部結合変換を適用する前に、2 つのテーブルとして視覚化されます。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233

クエリ B

時間	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	サーバー 2	5
7/7/2020 午前 11 時 4 分 20 秒	サーバー 3	10

外部結合変換を適用した後の結果は次のようになります。

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 4 分 20 秒			サーバー 3	10

時間	ジョブ	アップタイム	[Server] (サーバー)	エラー
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233		
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122	サーバー 1	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233	サーバー 2	5

[フィールドへのラベル付け]

この変換により、ラベルまたはタグを含む時系列の結果がテーブルに変更され、各ラベルキーと値がテーブルの結果に含まれます。ラベルは、列または行の値として表示できます。

2 つの時系列のクエリ結果を示します。

- 系列 1 – ラベル サーバー = サーバー A、データセンター = EU
- 系列 2 – ラベル サーバー = サーバー B、データセンター = EU

[列] モードでは、結果は次のように表示されます。

時間	[Server] (サーバー)	データセンター	値
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	EU	1
7/7/2020 午前 11 時 34 分 20 秒	サーバー B	EU	2

[Rows] (行) モードでは、結果には各系列のテーブルがあり、次のような各ラベル値が表示されます。

ラベル	値
[Server] (サーバー)	サーバー A
データセンター	EU

ラベル	値
[Server] (サーバー)	サーバー B
データセンター	EU

値フィールド名

値フィールド名としてサーバーを選択した場合、サーバーラベルの値ごとに 1 つのフィールドを取得します。

時間	データセンター	サーバー A	サーバー B
7/7/2020 午前 11 時 34 分 20 秒	EU	1	2

マージ動作

フィールドトランスフォーマーへのラベルは、内蔵されている 2 つの異なる変換です。1 つ目は、1 つの系列で動作し、ラベルをフィールドに抽出します。2 つ目は、すべての結果を 1 つのテーブルに結合するマージ変換です。マージ変換は、一致するすべてのフィールドで結合を試みます。このマージステップは必須であり、オフにすることはできません。

> **Note**
>
> [マージ]変換は単独で使用でき、以下で詳細に説明します。

これを説明するために、重複ラベルのない時系列を返すクエリが 2 つある例を次に示します。

- 系列 1 – ラベル サーバー = サーバー A
- 系列 2 – ラベル データセンター = EU

これにより、まず次の 2 つのテーブルが作成されます。

時間	[Server] (サーバー)	値
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	10

時間	データセンター	値
7/7/2020 午前 11 時 34 分 20 秒	EU	20

マージ後

時間	[Server] (サーバー)	値	データセンター
7/7/2020 午前 11 時 34 分 20 秒	サーバー A	10	
7/7/2020 午前 11 時 34 分 20 秒		20	EU

Merge (マージ)

この変換を使用して、複数クエリの結果を 1 つの結果にまとめます。これはテーブルパネルの視覚化を使用する場合に役立ちます。マージできる値は同じ行に結合されます。共有フィールドに同じデータが含まれている場合は値をマージできます。

次の例では、テーブルデータを返すクエリが 2 つあります。変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	ジョブ	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233

クエリ B

時間	ジョブ	エラー
7/7/2020 午前 11 時 34 分 20 秒	ノード	15
7/7/2020 午前 11 時 24 分 20 秒	postgre	5

マージ変換を適用した後の結果は次のとおりです。

時間	ジョブ	エラー	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	ノード	15	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	5	123001233

[フィールドの整理]

この変換を使用して、クエリが返すフィールドの名前を変更、並べ替え、非表示にします。

> **Note**
>
> この変換は単一クエリを持つパネルでのみ機能します。パネルに複数のクエリがある場合は、外部結合変換を適用するか、追加のクエリを削除する必要があります。

Grafana では、クエリが返すフィールドのリストが表示されます。次のようにできます：

- フィールドの上にカーソルを置いて、フィールドの順序を変更します。カーソルが手の形になると、フィールドを新しい場所にドラッグできます。
- フィールド名の隣の目アイコンをクリックして、フィールドを非表示または表示します。
- 名前変更ボックスに新しい名前を入力して、フィールドの名前を変更します。

[値による分割]

この変換により、複数の系列をグラフ化する際に、異なる WHERE 句を持つ同じデータソースへの複数のクエリが不要になります。次のデータを含むメトリクス SQL テーブルを検討します。

時間	リージョン	値
10/20/2022 12:00:00 PM	米国	1520
10/20/2022 12:00:00 PM	EU	2936
10/20/2022 午前 1 時 00 分	米国	1327
10/20/2022 午前 1 時 00 分	EU	912

v9.3 より前では、US の赤いトレンドラインと EU の青いトレンドラインを同じ時系列パネルにプロットする場合は、これを 2 つのクエリに分割する必要があります。

```
SELECT Time, Value FROM metrics WHERE Time > '2022-10-20' AND Region='US'
SELECT Time, Value FROM metrics WHERE Time > '2022-10-20' AND Region='EU'
```

また、メトリクステーブルに実際に存在するリージョンを事前に知る必要があります。

[値による分割]トランスフォーマーを使用すると、単一のクエリを発行し、選択した 1 つ以上の列 (fields) に対して一意の値で結果を分割できるようになりました。次の例では Region を使用しています。

```
SELECT Time, Region, Value FROM metrics WHERE Time > '2022-10-20'
```

時間	リージョン	値
10/20/2022 12:00:00 PM	米国	1520
10/20/2022 午前 1 時 00 分	米国	1327

時間	リージョン	値
10/20/2022 12:00:00 PM	EU	2936
10/20/2022 午前 1 時 00 分	EU	912

[削減]

[削減] 変換は、フレーム内の各フィールドに計算を適用し、単一の値を返します。この変換を適用すると、時間フィールドは削除されます。

以下の入力を検討します。

クエリ A

時間	Temp	アップタイム
7/7/2020 午前 11 時 34 分 20 秒	12.3	256122
7/7/2020 午前 11 時 24 分 20 秒	15.4	1230233

クエリ B

時間	AQI	エラー
7/7/2020 午前 11 時 34 分 20 秒	6.5	15

時間	AQI	エラー
7/7/2020 午前 11 時 24 分 20 秒	3.2	5

縮小のトランスフォーマーには 2 つのモードがあります。

- [シリーズから行] – 各フィールドの行と各計算の列を作成します。
- [縮小フィールド] – 既存のフレーム構造を維持しますが、各フィールドを 1 つの値に折りたたみます。

例えば、系列で [最初] と [最後] の計算を使用して [シリーズから行] 変換を行った場合、結果は次のようになります。

フィールド	First	Last
Temp	12.3	15.4
アップタイム	256122	1230233
AQI	6.5	3.2
エラー	15	5

最後の計算における縮小フィールドは、それぞれ 1 行、2 つのフレームになります。

クエリ A

温度	アップタイム
15.4	1230233

クエリ B

AQI	エラー
3.2	5

[正規表現で名前を変更]

この変換を使用して、正規表現と置換パターンを使用してクエリ結果の一部の名前を変更します。

正規表現を指定できます。正規表現は、バックリファレンスをサポートする置換パターンとともに、一致にのみ適用されます。例えば、ホストあたりの CPU 使用率を視覚化し、ドメイン名を削除するとします。正規表現を `([^\.]+)\..+` に設定し、置換パターンを `$1` に設定することで、`web-01.example.com` は `web-01` になります。

Rows to fields (行からフィールドへ)

行からフィールドへの変換は、行を別々のフィールドに変換します。これは、フィールドを個別にスタイル変更および設定できるため便利です。また、動的フィールド設定のソースとして追加のフィールドを使用したり、フィールドラベルにマッピングしたりすることもできます。その後、追加のラベルを使用して、結果フィールドにより適切な表示名を定義できます。

この変換には、設定クエリによって返されるデータ内のすべてのフィールドを一覧表示するフィールドテーブルが含まれます。このテーブルでは、各設定プロパティにマッピングするフィールドを制御できます ([Use as] (用途) オプション)。返されるデータに複数の行がある場合、選択する値を選ぶこともできます。

この変換には以下が必要です。

- フィールド名のソースとして使用する 1 つのフィールド。

  デフォルトでは、変換は最初の文字列フィールドをソースとして使用します。このデフォルト設定は、代わりに使用するフィールドの [用途] 列で [フィールド名] を選択してオーバーライドできます。
- 値のソースとして使用する 1 つのフィールド。

  デフォルトでは、変換は最初の数値フィールドをソースとして使用します。ただし、このデフォルト設定は、代わりに使用するフィールドの [用途] 列で [フィールド値] を選択してオーバーライドできます。

以下にデータを視覚化する場合に役立ちます。

- ゲージ
- 統計
- 円グラフ

追加フィールドをラベルにマッピング

フィールドが設定プロパティにマッピングされない場合、Grafana は出力フィールドのラベルのソースとして自動的に使用します。

例:

名前	データセンター	値
サーバー A	米国	100
サーバー B	EU	200

出力:

サーバー A (ラベル: データセンター: US)	サーバー B (ラベル: データセンター: EU)
10	20

フィールド表示名で追加のラベルを使用して、より完全なフィールド名を提供できるようになりました。

あるクエリから設定を抽出して別のクエリに適用する場合は、[クエリ結果から設定] 変換を使用する必要があります。

例

入力

名前	値	最大
サーバー A	10	100

名前	値	最大
サーバー B	20	200
サーバー C	30	300

出力

サーバー A (設定: 最大=100)	サーバー B (設定: 最大=200)	サーバー C (設定: 最大=300)
10	20	30

ソースデータ内の各行が個別のフィールドになります。各フィールドには、最大設定オプションも設定されるようになりました。最小値、最大値、単位、しきい値などのオプションはすべてフィールド設定の一部であり、このような設定を行うと、パネル工ディタのオプションペインで手動設定されるオプションではなく、視覚化によって使用されます。

[時系列の準備]

時系列の準備変換は、データソースが、使用するパネルでサポートされていない形式で時系列データを返す場合に役立ちます。

この変換は、時系列データをワイド形式からロング形式、またはその逆に変換することで、この問題を解決するのに役立ちます。

[マルチフレーム時系列] オプションを選択して、時系列データフレームをワイド形式からロング形式に変換します。

[ワイド時系列] オプションを選択して、時系列データフレームをロング形式からワイド形式に変換します。

[シリーズから行]

この変換を使用して、複数の時系列データクエリの結果を 1 つの結果にまとめます。これはテーブルパネルの視覚化を使用する場合に役立ちます。

この変換の結果には、時間、メトリクス、値の 3 つの列が含まれます。メトリクス列が追加されると、メトリクスのソースとなるクエリを簡単に確認できます。ソースクエリのラベルを定義して、この値をカスタマイズします。

次の例では、時系列データを返すクエリが 2 つあります。変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	温度
7/7/2020 午前 11 時 34 分 20 秒	25
7/7/2020 午前 10 時 31 分 22 秒	22
7/7/2020 午前 9 時 30 分 5 秒	19

クエリ B

時間	湿度
7/7/2020 午前 11 時 34 分 20 秒	24
7/7/2020 午前 10 時 32 分 20 秒	29
7/7/2020 午前 9 時 30 分 57 秒	33

シリーズから行 変換を適用した後の結果は次のとおりです。

時間	メトリクス	値
7/7/2020 午前 11 時 34 分 20 秒	温度	25
7/7/2020 午前 11 時 34 分 20 秒	湿度	22
7/7/2020 午前 10 時 32 分 20 秒	湿度	29
7/7/2020 午前 10 時 31 分 22 秒	温度	22

時間	メトリクス	値
7/7/2020 午前 9 時 30 分 57 秒	湿度	33
7/7/2020 午前 9 時 30 分 5 秒	温度	19

[並べ替え]

この変換では、設定されたフィールドで各フレームがソートされます。reverse にチェックがついている場合、値は逆の順序で返されます。

[制限]

この変換を使用して、表示される行数を制限します。

次の例では、データソースから次のレスポンスがあります。

時間	メトリクス	値
7/7/2020 午前 11 時 34 分 20 秒	温度	25
7/7/2020 午前 11 時 34 分 20 秒	湿度	22
7/7/2020 午前 10 時 32 分 20 秒	湿度	29
7/7/2020 午前 10 時 31 分 22 秒	温度	22
7/7/2020 午前 9 時 30 分 57 秒	湿度	33
7/7/2020 午前 9 時 30 分 5 秒	温度	19

以下は、値が「3」の制限変換を追加した後の結果です。

時間	メトリクス	値
7/7/2020 午前 11 時 34 分 20 秒	温度	25
7/7/2020 午前 11 時 34 分 20 秒	湿度	22
7/7/2020 午前 10 時 32 分 20 秒	湿度	29

クエリのトラブルシューティング

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このページでは、一般的なダッシュボードの問題を解決するための情報を提供します。

関数を再配置すると、異なる結果が得られる

関数の順序は非常に重要です。数学と同様に、関数を配置する順序は結果に影響を与える可能性があります。

クエリリクエストとレスポンスを検査する

最も一般的な問題は、データソースからのクエリとレスポンスに関連しています。Grafana のバグや視覚化の問題のように見えても、ほとんどの場合、データソースクエリまたはデータソースレスポンスの問題です。まず、パネルクエリとレスポンスを検査します。

詳細については、「リクエストとレスポンスデータの検査」を参照してください。

クエリが遅い

クエリが返すデータポイントの数を確認してください。多くのデータポイントを返すクエリは遅くなります。解決策は以下のとおりです。

- [クエリオプション]で、返される[最大データポイント数]を制限します。
- クエリオプションで、最小間隔時間を長くします。
- クエリで、`group by` 関数を使用します。

しきい値を設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このセクションでは、視覚化でしきい値を使用する方法について説明します。しきい値とそのデフォルト、しきい値を追加または削除する方法、レガシーパネルにしきい値を追加する方法について紹介します。

しきい値について

しきい値は、しきい値に達するか超えたときに、ダッシュボードに視覚的に反映されるメトリクスに対して指定する値です。

しきい値は、クエリ結果に基づいて視覚化を条件付きでスタイル付けおよび色付けするための 1 つの方法をもたらします。しきい値は、すべてではないのですが、ほとんどの視覚化に適用できます。視覚化の詳細については、「視覚化パネル」を参照してください。

しきい値を使用すると、次のことができます。

- 時系列の視覚化でグリッド線またはグリッド領域を色付け
- 時系列の視覚化で線を色付け
- 統計の視覚化で背景または値のテキストを色付け
- ゲージ視覚化のゲージマーカーとしきい値マーカーを色分け
- ジオマップ視覚化でマーカーを色付け

- テーブル視覚化でセルテキストまたは背景を色付け
- 状態タイムラインの視覚化でリージョンとリージョンの色を定義

しきい値には、次の 2 つのタイプがあります。

- [絶対値] しきい値は数値で定義されます。例えば、1 ~ 150 のスケールで 80 とします。
- [パーセンテージ] しきい値は、最小値または最大値に対して定義されます。例えば、80% です。

デフォルトのしきい値

しきい値をサポートする視覚化において、Grafana はデフォルトのしきい値を次のように設定します。

- 80 = red (80 = 赤)
- Base = green (ベース = 緑)
- Mode = Absolute (モード = 絶対値)

[Base] (ベース) 値は、マイナス無限大を表します。一般的に「良い」色です。

しきい値の追加または削除

パネルにはお好みの数のしきい値を追加できます。Grafana はしきい値を高い順に自動的にソートします。

しきい値がビジネスオペレーションに関連しなくなった場合は削除します。しきい値を削除すると、システムはしきい値を含むすべての視覚化からしきい値を削除します。

1. しきい値を追加するには
    1. しきい値を追加するパネルを編集します。
    2. オプションサイドペインで [しきい値] セクションを見つけ、[しきい値を追加] をクリックします。
    3. しきい値の色、数値、モードを選択します。しきい値モードは、このパネルのすべてのしきい値に適用されます。
    4. 時系列パネルで、[しきい値の表示] オプションを選択します。
2. しきい値を削除するには、しきい値を含むパネルに移動し、削除するしきい値の横にあるごみ箱アイコンをクリックします。

レガシーグラフパネルにしきい値を追加する

グラフパネルの視覚化では、しきい値を使用すると、グラフに線やセクションを追加できるため、グラフがしきい値を超えたタイミングを簡単に認識できます。

1. しきい値を追加するグラフパネルに移動します。
2. [Panel]タブで、[しきい値] をクリックします。
3. [しきい値を追加] をクリックします。
4. 以下のフィールドに値を入力します。
   - T1 – しきい値を表示するために両方の値が必要です。
     - lt または gt – しきい値の適用対象示すため、より小さい場合は lt、より大きい場合は gt を選択します。
     - [値] – しきい値を入力します。Grafana では、その値で Y 軸に沿ってしきい値線が描画されます。
   - Color – 色に対応する条件を選択、または独自の色を定義します。
     - [カスタム] – 塗りつぶしの色と線の色を定義します。
     - [重大] - 塗りつぶしの色と線の色は赤色です。
     - [警告] – 塗りつぶしの色と線の色は黄色です。
     - OK – 塗りつぶしの色と線の色は緑色です。
   - [塗りつぶし] - しきい値の塗りつぶしを表示するかどうかを制御します。
   - [線] – しきい値の線を表示するかどうかを制御します。
   - [Y 軸] – 左または右を選択します。
5. [保存] をクリックして、ダッシュボードに変更を保存します。

データリンクの設定

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

データリンク変数またはデータリンクを使用して、パネル間のリンクを作成できます。

データリンク変数

データリンクで変数を使用すると、系列フィールド、ラベル、値を参照できます。データリンクの詳細については、「データリンク」を参照してください。

使用可能な変数のリストを表示するには、データリンク URL フィールドに「$」と入力して、使用できる変数のリストを表示します。

> **Note**
>
> これらの変数はバージョン 6.4 で変更されたため、古いバージョンの Grafana の場合は、バージョンピッカーを使用して古いバージョンの Grafana のドキュメントを選択してください。

データリンク URL のテンプレート変数も使用できます。「ダッシュボード変数の追加と管理」を参照してください。

時間範囲パネル変数

これらの変数を使用すると、データリンク URL に現在の時間範囲を含めることができます。

- __url_time_range - 現在のダッシュボードの時間範囲 (つまり?from=now-6h&to=now)
- $__from および $__to - 詳細については、「グローバル変数」を参照してください。

系列変数

系列固有の変数は、__series 名前空間で使用できます。

- __series.name - URL への系列名

フィールド変数

フィールド固有の変数は、__field 名前空間で使用できます。

- __field.name - フィールドの名前
- __field.labels.<LABEL> - URL へのラベルの値。ラベルにドットが含まれている場合は、__field.labels["<LABEL>"] 構文を使用します。

値変数

値固有の変数は、__value 名前空間で使用できます。

- [__value.time - URL への値のタイムスタンプ (Unix ms エポック) (例: ?time=1560268814105)
- [__value.raw] - raw 値
- [__value.numeric] - 値の数値表現
- [__value.text] - 値のテキスト表現
- [__value.calc - 値が計算の結果である場合の計算名]

テンプレート変数

テンプレート変数を使用する別のダッシュボードにリンクするときは、リンクをクリックした人の変数値を選択します。

${var-myvar:queryparam} - ここで、var-myvar は、使用する現在のダッシュボードのテンプレート変数と一致するテンプレート変数の名前です。

可変状態	作成された URL の結果
1 つの値を選択済み	var-myvar=value1
複数の値を選択済み	var-myvar=value1&var-myvar=value2
すべて選択済み	var-myvar=All

現在のダッシュボードのすべての変数を URL に追加する場合は、${__all_variables} を使用します。

データリンク

データリンクを使用すると、リンクへのより詳細なコンテキストを提供できます。カーソルの下に系列名または値を含むリンクを作成できます。例えば、視覚化で 4 つのサーバーが表示されていた場合は、そのうちの 1 つまたは 2 つにデータリンクを追加できます。

リンク自体には、視覚化に応じてさまざまな方法でアクセスできます。グラフでは、データポイントまたは線をクリックする必要があります。統計、ゲージ、棒ゲージなどのパネルでは、視覚化の任意の場所をクリックしてコンテキストメニューを開くことができます。

データリンクに変数を使用すると、データフィルターが設定された状態の詳細なダッシュボードにユーザーを移動させることができます。例えば、 変数を使用して、時間範囲、系列、または変数の選択を指定できます。詳細については、「データリンク変数」を参照してください。

入力候補機能

データリンクを作成または更新する際に、キーボードの Cmd+Space または Ctrl+Space を押すと入力候補が表示され、URL に変数をより簡単に追加できます。

データリンクを追加する

1. リンクを追加するパネルにカーソルを合わせ、e を押します。または、パネルタイトルの横にあるドロップダウン矢印をクリックし、[編集] をクリックします。
2. [フィールド] タブで、[データリンク] セクションまでスクロールダウンします。
3. データリンクを展開し、[リンクの追加] をクリックします。
4. [タイトル] を入力します。[タイトル]は、UI に表示されるリンクの判読可能なラベルです。
5. リンク先の URL を入力します。

   ダッシュボードで定義済みのテンプレート変数のいずれかを追加することもできます。URL フィールドをクリックして「$」と入力するか、Ctrl+Space または Cmd+Space を押すと、使用可能な変数の一覧が表示されます。パネルリンクにテンプレート変数を追加することで、関連する変数がすでに設定されている適切なコンテキストをリンクからユーザーに送信します。詳細については、「データリンク変数」を参照してください。
6. リンクを新しいタブで開く場合は、[新しいタブで開く] を選択します。
7. [保存] をクリックして変更を保存し、ウィンドウを閉じます。
8. 右上の [保存] をクリックし、この変更をダッシュボードに保存します。

データリンクの更新

1. [フィールド] タブで、変更するリンクを見つけます。
2. 編集 (鉛筆) アイコンをクリックして、リンクの編集ウィンドウを開きます。
3. 必要に応じて変更します。
4. [保存] をクリックして変更を保存し、ウィンドウを閉じます。
5. 右上の [保存] をクリックし、この変更をダッシュボードに保存します。

データリンクの削除

1. [フィールド] タブで、削除するリンクを見つけます。
2. 削除するリンクの横にある [X] アイコンをクリックします。
3. 右上の [保存] をクリックして、ダッシュボードに変更を保存します。

フィールドのオーバーライドを設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

オーバーライドを使用すると、特定のフィールドまたは系列に対して視覚化設定をカスタマイズできます。これは、特定のフィールドセットをターゲットとし、それぞれ複数のオプションを定義できるオーバーライドルールを追加することで実現されます。

例えば、[正規表現と一致する名前付きフィールド]を使用してオーバーライドを追加して、テキスト「bytes」を含むすべてのフィールドに単位を設定し、オーバーライドルールに単位オプションを追加できます。

例 1: 温度の形式

結果セットは、時間と温度の 2 つのフィールドで構成されるデータフレームであるとします。

時間	温度
2020-01-02 03:04:00	45.0
2020-01-02 03:05:00	47.0
2020-01-02 03:06:00	48.0

この構造の各フィールド (列) には、値の表示方法を変更するフィールドオプションを適用できます。つまり、例えば、単位を温度 > 摂氏に設定して、次の表を作成できます。

時間	温度
2020-01-02 03:04:00	45.0 °C
2020-01-02 03:05:00	47.0 °C
2020-01-02 03:06:00	48.0 °C

また、小数点は必須ではないため、削除できます。小数を自動からゼロ (0) に変更すると、次の表になります。

時間	温度
2020-01-02 03:04:00	45 °C
2020-01-02 03:05:00	47 °C
2020-01-02 03:06:00	48 °C

例 2: 温度と湿度の形式

結果セットが、「時間」、「高温」、「低温」、「湿度」の 4 つのフィールドで構成されるデータフレームである場合を例にします。

時間	高温	低温	湿度
2020-01-02 03:04:00	45.0	30.0	67
2020-01-02 03:05:00	47.0	34.0	68
2020-01-02 03:06:00	48.0	31.0	68

摂氏単位を追加し、小数点以下を削除しましょう。この結果は以下の表になります。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67 °C
2020-01-02 03:05:00	47 °C	34 °C	68 °C
2020-01-02 03:06:00	48 °C	31 °C	68 °C

温度フィールドは良く見えますが、湿度を変更する必要があります。これを修正するには、湿度フィールドにフィールドオプションオーバーライドを適用し、単位を [Misc] (その他) > % (0 ~ 100) に変更します。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67%
2020-01-02 03:05:00	47 °C	34 °C	68%
2020-01-02 03:06:00	48 °C	31 °C	68%

フィールドオーバーライドを追加する

フィールドオーバーライドルールは、特定のフィールドまたは系列の視覚化設定をカスタマイズできます。

1. オーバーライドを追加するパネルを編集します。
2. パネルオプションサイドペインで、ペインの下部にある [フィールドオーバーライドの追加] をクリックします。
3. オーバーライドルールが適用されるフィールドを選択します。

   - [名前付きフィールド]: 使用可能なすべてのフィールドの一覧からフィールドを選択します。このセレクタでルールに追加するプロパティは、この単一フィールドにのみ適用されます。
   - [正規表現と一致する名前付きフィールド]: 正規表現でオーバーライドするフィールドを指定します。このセレクタでルールに追加するプロパティは、フィールド名が正規表現と一致するすべてのフィールドに適用されます。

- [タイプ別フィールド]: 文字列、数値などのフィールドをタイプ別に選択します。このセレクタでルールに追加するプロパティは、選択したタイプに一致するすべてのフィールドに適用されます。
- [クエリによって返されるフィールド]: A、B、C など、特定のクエリによって返されるすべてのフィールドを選択します。このセレクタでルールに追加するプロパティは、選択したクエリによって返されるすべてのフィールドに適用されます。

4. [オーバーライドプロパティの追加] をクリックします。
5. 適用するフィールドオプションを選択します。
6. フィールドに値を追加してオプションを入力します。オプションをデフォルト値に戻すには、フィールドの白い文字を削除します。
7. [オーバーライドプロパティの追加] をクリックして、このフィールドにオーバーライドを追加し続けます。または、[オーバーライドの追加] をクリックして、オーバーライドを追加する別のフィールドを選択できます。
8. 完了したら [保存] をクリックして、すべてのパネル編集をダッシュボードに保存します。

フィールドオーバーライドの削除

不要になったフィールドオーバーライドを削除します。オーバーライドを削除すると、値の外観はデフォルトで元の形式になります。この変更は、影響を受けるパネルに依存するダッシュボードとダッシュボードユーザーに影響します。

1. 削除するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。
3. 削除するオーバーライドをクリックし、関連付けられたごみ箱アイコンをクリックします。

フィールドオーバーライドの表示

フィールドオーバーライドは、パネル表示オプションで表示できます。

1. 表示するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。

> [すべて] タブに表示されるオーバーライド設定は、[オーバーライド] タブに表示される設定と同じです。

フィールドオーバーライドの編集

オーバーライド設定に変更を加える場合は、フィールドオーバーライドを編集します。加えた変更はすぐに反映されます。

1. 編集するオーバーライドを含むパネルを編集します。
2. パネルオプションサイドペインで、オーバーライドが表示されるまで下にスクロールします。
3. 変更するオーバーライドを見つけます。
4. 次のいずれかを実行します。
   - 既存のオーバーライドまたはフィールド選択パラメータの設定を編集します。
   - プロパティの横にある [X] をクリックして、既存のオーバーライドプロパティを削除します。
   - [オーバーライドプロパティの追加] をクリックして、オーバーライドプロパティを追加します。

値マッピングを設定する

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

フィールドオーバーライドに加えて、値マッピングは、視覚化に表示されるデータの視覚的処理を変更するために使用できる手法です。

値マッピングを介してマッピングされた値は、単位の書式設定を介します。つまり、数値にマッピングされたテキスト値は、設定された単位を使用してフォーマットされません。

値マッピングがパネルに存在する場合、Grafana はパネル工ディタのサイドペインに概要を表示します。

> **Note**
>
> 新しい値マッピングは、(古い) グラフ、テキスト、ヒートマップなどの一部の視覚化と互換性がありません。

値マッピングの種類

Grafana は、次の値マッピングをサポートしています。

- [値]: テキスト値を色または別の表示テキストにマッピングします。例えば、値 10 のすべてのインスタンスが数値ではなく Perfection! として表示されるように値マッピングを設定できます。
- [Range]: 数値範囲を表示テキストと色にマッピングします。例えば、値が特定の範囲内にある場合、数値ではなく下限または上限を表示するように範囲値マッピングを設定できます。
- [Regex]: 正規表現を代替テキストと色にマッピングします。例えば、値が www.example.com の場合、Grafana が www を表示し、ドメインを切り捨てるように正規表現値マッピングを設定できます。
- [特殊] [Null]、[NaN] (数値以外) などの特殊な値、[真] や [偽] などのブール値を表示テキストと色にマッピングします。例えば、[Null] 値が N/A として表示されるように、特殊な値マッピングを設定できます。

また、左側のドットを使用して、リスト内の値マッピングをドラッグして並べ替えることもできます。

値をマッピングする

単一の値をフォーマットする場合は、値をマッピングします。

1. 値をマッピングするパネルを開きます。
2. パネル表示オプションで [値マッピング] セクションを見つけ、[値マッピングを追加] をクリックします。
3. [新しいマッピングを追加] をクリックし、[値l を選択します。
4. Grafana が一致する値を入力します。
5. (オプション) 表示テキストを入力します。
6. (オプション) 色を設定します。
7. [更新] をクリックして、値マッピングを保存します。

範囲をマッピングする

複数の連続値をフォーマットする場合は、値の範囲をマッピングします。

1. 値の範囲をマッピングするパネルを編集します。
2. パネル表示オプションの [値マッピング] セクションで、[値マッピングを追加] をクリックします。
3. [新しいマッピングを追加] をクリックし、[Range] (範囲) を選択します。
4. Grafana が一致する範囲内に開始値と終了値を入力します。
5. (オプション) 表示テキストを入力します。
6. (オプション) 色を設定します。
7. [更新] をクリックして、値マッピングを保存します。

正規表現をマッピングする

正規表現値のテキストと色をフォーマットする場合は、正規表現をマッピングします。

1. 正規表現をマッピングするパネルを編集します。
2. パネル表示オプションの [値マッピング] セクションで、[値マッピングを追加] をクリックします。
3. [新しいマッピングを追加] をクリックし、[Regex] (正規表現) を選択します。
4. Grafana が一致する正規表現パターンを入力します。
5. (オプション) 表示テキストを入力します。
6. (オプション) 色を設定します。
7. [更新] をクリックして、値マッピングを保存します。

特殊な値をマッピングする

まれな値、ブール値、または空の値をフォーマットする場合は、特殊な値をマッピングします。

1. 特殊な値をマッピングするパネルを編集します。
2. パネル表示オプションで [値マッピング] セクションを見つけ、[値マッピングを追加] をクリックします。
3. [新しいマッピングを追加]をクリックし、[特殊] を選択します。
4. Grafana が一致する特殊な値を選択します。

5. (オプション) 表示テキストを入力します。
6. (オプション) 色を設定します。
7. [更新] をクリックして、値マッピングを保存します。

値マッピングを編集する

値マッピングはいつでも変更できます。

1. 編集する値マッピングを含むパネルを編集します。
2. パネル表示オプションの [値マッピング] セクションで、[値マッピングの編集] をクリックします。
3. 変更を行い、[更新] をクリックします。

凡例を設定する

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネルには、視覚化に表示されるデータの解釈に使用できる凡例が含まれています。各凡例オプションは、視覚化に示されているデータにコンテキストと明確さを追加します。

視覚化での系列データの分離

視覚化は視覚的に複雑になることが多く、データ系列が数多く含まれている場合があります。視覚化から系列データを削除することで、表示を簡素化できます。これにより、表示したいデータが分離されます。Grafana は [Override] タブに新しいオーバーライドを自動的に作成します。

変更を適用すると、視覚化の変更がパネルのすべてのユーザーに表示されます。

視覚化で系列データを分離するには

1. パネルを開きます。

2. 凡例で、分離する系列のラベルを選択します。

   システムによって、他のすべての系列データ表示から削除されます。

3. 系列データを分離された系列に段階的に追加するには、Ctrl キーまたは Command キーを押して、追加する系列のラベルを選択します。
4. すべてのデータを含むデフォルトの表示に戻すには、任意の系列ラベルを 2 回クリックします。
5. パネルのすべてのビューワーに表示されるように変更を保存するには、[適用] を選択します。

このトピックは現在、次の視覚化に適用されます。

- 棒グラフ
- ヒストグラム
- 円グラフ
- 状態タイムライン
- ステータス履歴
- 時系列

凡例への値の追加

視覚化にコンテキストを追加する方法として、系列データ値を凡例に追加できます。必要な数だけ値を追加できます。変更を適用したら、凡例を水平にスクロールするとすべての値を表示できます。

凡例に値を追加するには

1. パネルを編集します。
2. パネル表示オプションペインで、[凡例] セクションを見つけます。
3. [凡例値] フィールドで、凡例に表示する値を選択します。
4. [適用] を選択して変更を保存すると、ダッシュボードに戻ります。

系列の色の変更

デフォルトでは、Grafana は系列データの色を指定しますが、これは変更できます。

系列の色を変更するには

1. パネルを編集します。
2. 凡例で、系列に関連付けられたカラーバーを選択します。
3. カラーパレットから事前設定された色またはユーザー定義の色を選択します。
4. [適用] を選択して変更を保存すると、ダッシュボードに戻ります。

系列の並べ替え

凡例モードをテーブルに変更し、凡例に表示されるように 計算タイプ を選択できます。凡例テーブルの計算名ヘッダーを選択して、テーブル内の値を昇順または降順に並べ替えます。

並べ替えの順序は、棒グラフパネルの棒の位置と、時系列および棒グラフパネルの積み上げ系列の順序に影響します。

> **Note**
>
> この機能は、棒グラフ、ヒストグラム、時系列、XY チャートのパネルにのみ対応しています。

計算タイプ

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

次の表に、Grafana で実行できる計算の一覧を示します。これらの計算は、[変換] タブと棒ゲージ、ゲージ、および統計の視覚化で確認できます。

計算	説明
All nulls	すべての値が null のとき真

計算	説明
All zeros	すべての値が 0 のとき真
Change count	フィールドの値が変更される回数
Count	フィールド内の値の数
Delta	値の累積変化、カウント増分のみ
Difference	フィールドの最初の値と最後の値の差
Difference percent	フィールドの最初と最後の値の変化率
Distinct count	フィールド内の一意の値の数
最初 (null ではない)	最初、null でないフィールド内の値
最大	フィールドの最大値
Mean	フィールド内のすべての値の平均値
Variance	フィールド内のすべての値の分散
StdDev	フィールド内のすべての値の標準偏差
最小	フィールドの最小値
最小 (above zero)	フィールドの最小値、正の値
Range	フィールドの最大値と最小値との差
Step	フィールドの値間の最小間隔
Total	フィールド内のすべての値の合計

視覚化に注釈を付ける

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

注釈は、リッチイベントでグラフ上のポイントをマークする方法を提供します。注釈にカーソルを合わせると、イベントの説明とイベントタグを取得できます。テキストフィールドには、より詳細な他のシステムへのリンクを含めることができます。

ネイティブ注釈

Grafana には、ネイティブ注釈ストアと、グラフパネルから直接、または HTTP API を介して注釈イベントを追加する機能があります。

注釈の追加

1. ダッシュボードで、[時系列] パネルをクリックします。コンテキストメニューが表示されます。
2. コンテキストメニューで、[注釈の追加] をクリックします。
3. 注釈の説明と、オプションでタグを追加します。
4. [保存] をクリックします。

または、注釈を追加するには、[時系列] パネル上で `Ctrl+Click` または `Cmd+Click` し、[注釈の追加] ポップアップが表示されます。

リージョン注釈の追加

1. ダッシュボードで、時系列パネル上で `Ctrl+Click` または `Cmd+Click` します。
2. コンテキストメニューで、[注釈の追加] をクリックします。
3. 注釈の説明と、オプションでタグを追加します。
4. [保存] をクリックします。

注釈の編集

1. ダッシュボードで、[時系列] パネルの注釈インジケータにカーソルを合わせます。
2. 注釈ツールチップの編集 (鉛筆) アイコンをクリックします。

3. 説明や、オプションでタグを変更します。
4. [保存] をクリックします。

注釈の削除

1. ダッシュボードで、[時系列] パネルの注釈インジケータにカーソルを合わせます。
2. 注釈ツールチップのごみ箱アイコンをクリックします。

ビルトインクエリクエリ

注釈を追加すると、それらは引き続き表示されます。これは、すべてのダッシュボードに存在する組み込みの注釈クエリが原因です。この注釈クエリは、現在のダッシュボードから発生するすべての注釈イベントを取得し、それらが作成されたパネルに表示されます。これには、アラート状態履歴の注釈が含まれます。[注釈] 設定 (ダッシュボードの歯車メニュー) を開き、`Annotations & Alerts (Built-in)` という名前のクエリを変更することで、注釈の取得と描画を停止できます。

[名前を付けて保存] 機能を使用してダッシュボードをコピーすると、新しいダッシュボード ID が取得されるため、ソースダッシュボードで作成された注釈はコピーに表示されなくなります。新しい [注釈クエリ] を追加し、タグでフィルタリングしても、注釈を表示できます。これは、ソースダッシュボードの注釈にフィルタリングするタグがある場合にのみ機能します。

タグによるクエリ

`-- Grafana --` データソースを使用して [~ でフィルタリング] を `Tags` に設定することで、ネイティブ注釈ストアから注釈を取得する新しいクエリを作成できます。

Grafana v8.1 以降のバージョンでは、既存のタグの先行入力もサポートされており、少なくとも 1 つのタグを提供します。

例えば、注釈クエリ名 `outages` を作成し、タグ `outage` を指定します。このクエリは、停止タグが付いたすべての注釈 (ダッシュボードまたは API 経由) を表示します。注釈クエリで複数のタグが定義されている場合、Grafana はすべてのタグに一致する注釈のみを表示します。動作を変更するには、`Match any` を有効にすると、Grafana は指定したタグのいずれかを含む注釈を表示します。

Grafana v5.3+ では、タグクエリでテンプレート変数を使用できます。したがって、さまざまなサービスの統計を表示するダッシュボードと、表示するサービスを指定するテンプレート変数がある場合は、注釈クエリで同じテンプレート変数を使用して、それらのサービスの注釈のみを表示できます。

その他のデータソースのクエリ

注釈イベントは、注釈クエリを介して取得されます。ダッシュボードに新しい注釈クエリを追加するには、ダッシュボード設定メニューを開き、[注釈] を選択します。これにより、ダッシュボードの注釈設定ビューが開きます。新しい注釈クエリを作成するには、[新規] ボタンをクリックします。

注釈クエリの名前を指定します。この名前は、このクエリから注釈イベントを表示または非表示にするトグル (チェックボックス) に付けられます。例えば、2 つの注釈クエリ(`Deploys` と `Outages`) があるとします。トグルを使用すると、表示する注釈を決定できます。

注釈クエリの詳細

注釈クエリオプションは、データソースごとに異なります。特定のデータソースの注釈の詳細については、特定の「データソース」トピックを参照してください。

パネル検査ビュー

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

パネル検査ビューは、パネルメニューから開くことができ、パネルの理解とトラブルシューティングに役立ちます。Grafana パネルの未加工データの検査、データの CSV (comma-separated values: カンマ区切り) ファイルへのエクスポート、クエリリクエストの表示、JSON 形式のパネルとデータのエクスポートが可能です。

> **Note**
>
> すべてのパネルタイプにタブがすべてあるわけではありません。例えば、ダッシュボードリストパネルには検査する未加工データがないため、[統計]、[データ]、[クエリ] タブは表示されません。

パネルインスペクターは、次のオプションで構成されます。

1. パネルインスペクターでは、ペインの上部に [Inspect: [NameOfPanelBeingInspected]] が表示されます。右上隅の矢印をクリックして、ペインを展開または縮小できます。

2. [データ タブ] – 変換が適用されたクエリによって返される未加工データを表示します。オーバーライドや値マッピングなどのフィールドオプションは、デフォルトでは適用されません。
3. [統計タブ] – クエリにかかる時間および返される量を示します。
4. [JSON] タブ - パネル JSON、パネルデータ JSON、およびデータフレーム構造 JSON を、表示およびコピーできます。これは Grafana のプロビジョニングまたは管理に役立ちます。
5. [クエリタブ] – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。
6. [エラータブ] – エラーを表示します。クエリがエラーを返す場合にのみ表示されます。

未加工のクエリ結果をダウンロードする

Grafana は、お使いのデータで構成される CSV ファイルを生成します。これには、そのデータに対する任意の変換も含まれます。パネルがフィールドオプションまたはフィールドオプションオーバーライドを適用する前後にデータを表示するように選択できます。

1. ダウンロードするクエリデータを含むパネルを編集します。
2. クエリエディタで、[Query Inspector] (クエリインスペクター) をクリックします。
3. [Data] (データ) をクリックします。

   パネルに複数のクエリがある、または複数のノードにクエリを実行する場合は、追加のオプションがあります。

   - [結果の選択]: 表示する結果セットデータを選択します。
   - [データを変換する]
   - [時間で結合]: すべてのクエリの生データを一度に表示し、列ごとに結果セットを 1 つ表示します。列見出しをクリックして、データの順序を変更します。
4. システムがフィールドのオーバーライドを適用する前にデータを表示するには、[フォーマットされたデータ] トグルをクリックします。
5. Excel 専用にフォーマットされた CSV ファイルをダウンロードするには、[Excel 用にダウンロード] トグル をクリックします。
6. [CSV ダウンロード] をクリックします。

クエリパフォーマンスの検査

[統計] タブには、クエリにかかる時間、送信したクエリの数、返された行数を示す統計が表示されます。この情報は、特に数値が予想外に高い場合や低い場合のクエリのトラブルシューティングに役立ちます。

1. クエリを含むパネルを、検査するパフォーマンスで編集します。
2. クエリエディタで、[Query Inspector] (クエリインスペクター) をクリックします。
3. [Stats] (統計) をクリックします。

統計が読み取り専用形式で表示されます。

クエリリクエストとレスポンスを検査する

予期しない結果を返すクエリ、または期待される結果を返さないクエリのトラブルシューティングを行う場合は、クエリリクエストとレスポンスデータを検査します。

1. エクスポートするクエリを含むパネルを編集します。
2. クエリエディタで、[Query Inspector] (クエリインスペクター) をクリックします。
3. [更新] をクリックします。

   パネルにはレスポンスデータが入力されます。
4. 必要に応じて調整を行い、クエリを再実行します。
5. クエリリクエストとレスポンスデータをダウンロードするには、[Copy to clipboard] (クリップボードにコピー) アイコンをクリックし、結果を別のアプリケーションに貼り付けます。

Grafana バージョン 9 で利用可能な視覚化

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は、さまざまなユースケースをサポートするさまざまな視覚化を提供します。ドキュメントのこのセクションでは、組み込まれているパネル、そのオプション、そして一般的な使用方法について説明します。

パネルの使用の開始からその基本まで学べる一般的なパネルは、時系列パネル パネルです。

トピック

- アラートリストパネル
- 注釈パネル
- 棒グラフパネル
- 棒ゲージ
- ローソク足パネル
- Canvas パネル
- 時計パネル
- ダッシュボードの一覧表示
- ゲージパネル
- ジオマップパネル
- グラフパネル
- ヒートマップパネル
- ヒストグラムパネル
- ログパネル
- ニュースパネル
- ノードグラフパネル
- 円グラフパネル
- Plotly パネル
- サンキーパネル
- 散布図パネル
- 統計パネル
- 状態タイムラインパネル
- ステータス履歴パネル
- テーブルパネル

- テキストパネル
- 時系列パネル
- トレースパネル (ベータ)
- WindRose

アラートリストパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートリストパネルには、ダッシュボードのアラートが表示されます。このリストは、現在の状態を表示、または最近の状態の変更を表示するように設定することができます。アラートの詳細については、「Grafana バージョン 9 のアラート」を参照してください。

この設定で、視覚化する内容を絞り込みます。

オプション

- [グループモード] – [デフォルトのグループ化] を選択して、アラートルール別にグループ化されたアラートインスタンスを表示します。また、[カスタムグループ化] では、ラベルのカスタムセット別にアラートインスタンスをグループ化します。
- [最大項目数] – 一覧表示するアラートの最大数を設定します。
- [ソート順序] – アラートをどのような順番で表示するかを選択します。
  - アルファベット (昇順) – アルファベット順
  - アルファベット (降順) – 逆アルファベット順
  - [重要性] – 以下を参照。1 が最も高い。
    - `alerting` または `firing`: 1
    - `no_data`: 2
    - `pending`: 3

  - ok: 4
  - paused または inactive: 5
- [このダッシュボードからのアラート] – アラートリストが含まれているダッシュボードからのみアラートを表示します。

フィルター

選択したクエリ、フォルダ、またはタグでアラートをフィルタリングする場合、次のオプションを使用できます。

- [アラート名] – アラート名クエリを入力します。
- [アラートインスタンスラベル] – ラベルクエリを使用してアラートインスタンスをフィルタリングします。例えば、`{severity="critical", instance=~"cluster-us-.+"}` と指定します。
- [フォルダ] – フォルダを選択します。選択したフォルダ内のダッシュボードからのアラートのみが表示されます。
- [データソース] – 選択したデータソースからアラートをフィルタリングします。

状態フィルター

このパネルに表示するアラートの状態を選択します。

- アラート中/発生中
- 保留中
- データなし
- 普通
- エラー

注釈パネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

注釈パネルには、注釈付きデータの表示に使用できる注釈の一覧が表示されます。タグと現在のダッシュボードに基づいてリストをフィルタリングできるさまざまなオプションを使用できます。

注釈クエリ

以下のオプションは、注釈のリストに対するソースクエリを制御します。

[クエリフィルター]

クエリフィルターを使用して、組織内のすべてのダッシュボードまたはこのパネルがある現在のダッシュボードから注釈のリストを作成します。これには次のオプションがあります。

- すべてのダッシュボード - 現在の組織内のすべてのダッシュボードの注釈を一覧表示します。
- このダッシュボード - リストを現在のダッシュボードの注釈に制限します。

[時間範囲]

時間範囲オプションを使用して、リストを現在の時間範囲に制限するかどうかを指定します。これには次のオプションがあります。

- [なし] - 注釈クエリの時間範囲の制限はありません。
- このダッシュボード - リストを、注釈リストパネルが利用可能なダッシュボードの時間範囲に制限します。

[タグ]

タグオプションを使用して、注釈をタグでフィルタリングします。リストを絞り込むために複数のタグを追加できます。

> **Note**
>
> 必要に応じて、タグリストを空のままにして、パネル自体の結果の一部としてリストされているタグを選択して、その場でフィルタリングします。

[制限]

制限オプションを使用して、返される結果の数を制限します。

[Display] (表示)

これらのオプションは、注釈パネル表示に含まれる追加のメタデータを制御します。

[ユーザーの表示]

このオプションを使用して、注釈を作成したユーザーを表示または非表示にします。

[時間の表示]

このオプションを使用して、注釈の作成時刻を表示または非表示にします。

[タグの表示]

このオプションを使用して、注釈に関連付けられたタグを表示または非表示にします。[NB]: タグを使用して、パネル自体の注釈リストをその場でフィルタリングできます。

リンク動作

[リンクの対象]

このオプションを使用して、注釈付きデータを表示する方法を選択します。これには次のオプションがあります。

- [Panel] (パネル) - このオプションは、対応する注釈を含むパネルのフル画面表示に直接移動します。
- ダッシュボード - このオプションは、全ダッシュボードのコンテキストで注釈に焦点を当てます。

[前に時間指定]

このオプションを使用して、注釈の前に時間範囲を設定します。「1h」= 1 時間、「10m」= 10 分などの期間文字列値を使用します。

[後に時間指定]

このオプションを使用して、注釈の後に時間範囲を設定します。

棒グラフパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このパネル表示では、カテゴリデータをグラフ化できます。

サポートされているデータ形式

サポートされるデータフレームは 1 つだけで、X 軸または Y 軸の項目として使用する文字列フィールドが少なくとも 1 つ、および数値フィールドが 1 つ以上必要です。以下は、データ形式の例です。

ブラウザ	市場シェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合は、パネルにグループ化されたバーが表示されます。

時系列または複数の結果セットの視覚化

複数の時系列またはテーブルがある場合は、まず結合を使用して結合するか、変換を減らす必要があります。例えば、複数の時系列があり、その最終値と最大値を比較する場合は、変換の [削減] を追加し、[計算] で [最大] と [Last] をオプションとして指定します。

棒グラフのオプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[Orientation] (向き)

- [自動] – Grafana は、パネルのディメンションに基づいてバーの向きを決定します。
- [水平] – X 軸をカテゴリ軸にします。
- [垂直] – Y 軸をカテゴリ軸にします。

[X 軸の目盛ラベルの最大長] は、棒グラフラベルの最大長を設定します。最大長を超えるラベルは省略記号で切り捨てられます。

[バーラベルの最小間隔] は、バーラベル間の最小間隔を設定します。

[値の表示]

値がバーの上部または左側に表示されるかどうかを制御します。

- [自動] – スペースがある場合は値が表示されます。
- [常に表示] – 常に値を表示します。
- [常に非表示] – 常に値を表示しません。

[積み上げ]

棒グラフの積み上げを制御します。

- [Off] (オフ) – バーは積み重ねられません。
- [普通] – バーは積み重ねられます。
- [パーセント] – バーは積み重ねられ、各バーの高さは積み重ねの合計の高さの割合です (バーの積み重ねはすべて同じ高さになり、最大 100% となります)。

[グループの幅] は、グループの幅を制御します。0=最小、1=最大幅です。

[Bar width] は、バーの幅を制御します。0=最小、1=最大幅です。

[Bar radius] は、バーの半径を制御します。0 = 最小、0.5 = 最大半径です。

[カーソル上の全領域をハイライト] は、ポインターでバーにカーソルを合わせた際に、バーの周囲を強調表示するかを制御します。

[線の幅]: 棒の線幅を制御します。

[塗りつぶしの不透明度]: 棒の塗りつぶし不透明度を制御します。

[グラデーションモード]: グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーション塗りつぶしなし。これはデフォルト設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。
- [色相] – グラデーション色は線の色相に基づいて生成されます。

[ツールチップモード] – 視覚化にカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、グラフ上でカーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

> **Note**
>
> オーバーライドを使用して、ツールチップから個々の系列を非表示にできます。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。詳細については、「凡例を設定する」を参照してください。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] - 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例計算] - 凡例に表示する標準計算を選択します。複数選択できます。

[テキストサイズ] – 棒グラフのテキストのサイズを変更する値を入力します。

[軸] – 次のフィールド設定を使用して、軸の表示方法を絞り込みます。一部のフィールドオプションは、編集するフィールドオプションボックスの外部をクリックするか、Enter キーを押すまで、視覚的表現には影響しません。

- [配置] – Y 軸の配置を設定します。
- [自動] – Grafana は、Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。
- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。
- [ラベル] – Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。
- [幅] – 軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

  軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。これにより、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。
- [ソフト最小値とソフト最大値] – Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて Y 軸の範囲を自動的に設定します。

  ソフト最小値とソフト最大値の設定により、データがほぼフラットなときに、小さな変動が大きな変動に見えるのを防ぎます。また、標準の最小値と最大値のフィールドオプションから派生したハード最小値または最大値により、定義されたポイントを超えたスパイクを切り取ることで、断続的なスパイクが有用な詳細を平坦化することを防げます。

  標準最小/最大値オプションを設定して、Y 軸のハード制限を定義できます。

棒ゲージ

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

棒ゲージは、各フィールドのデータを 1 つの値にまとめて表示することで情報を簡素化します。Grafana がまとめるにあたり、計算方法を選択します。

このパネルには、クエリの結果によって、1 つまたは複数の棒ゲージを表示できます (データが複数の系列、行、または列で構成されている場合など)。

値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示] – Grafana がデータを表示する方法を選択します。

[計算] – すべての行を元に計算した値を表示します。

- [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算の一覧については、「計算タイプ」を参照してください。
- [フィールド] – パネルに表示されるフィールドを選択します。

[すべての値] - 行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

- [制限] – 表示する行の最大数。デフォルトは 5,000 です。
- [フィールド] – パネルに表示されるフィールドを選択します。

棒ゲージオプション

棒ゲージの表示方法を設定します。

[Orientation] (向き) – 積み重ね方向を選択します。

- [自動] – Grafana が最適な方向を選択します。
- [水平] – バーは水平方向に左から右に伸びます。
- [垂直] – バーは垂直に下から上に伸びます。

Display mode (表示モード) – 表示モードを選択します。

- グラデーション] – しきい値レベルはグラデーションを定義します。
- [Retro LCD] – ゲージを小さなセルに分け、点灯または消灯させて表示します。
- [基本] – 一致するしきい値に基づいて単一色で表示します。

[塗りつぶされていない領域を表示] – バーの塗りつぶされていない領域を濃い灰色で表示する場合は、このオプションを選択します。Retro LCD 表示モードには適用されません。

[最小幅]

垂直方向におけるバーの列の最小幅を制限します。

大量のデータがある場合、X 軸スクロールバーを自動的に表示します。

[最小高さ]

水平方向におけるバーの行の最小高さを制限します。

大量のデータがある場合、Y 軸スクロールバーを自動的に表示します。

ローソク足パネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ローソク足パネルの作成により、価格変動に焦点を当てた多数の一貫したディメンションを含むデータを視覚化できます。ローソク足パネルには、OHLC (Open-High-Low-Close: 始値、高値、安値、終値) モードと、時系列データに基づく追加のディメンションのサポートが含まれています。

ローソク足パネルの作成は、時系列パネル の基礎に基づいて構築され、多くの一般的な設定が含まれています。

モード

モードオプションを使用すると、視覚化に使用されるディメンションを切り替えられます。

- [Candles] (ローソク) は、パネルのディメンションを、ローソク足チャートの作成で使用される始値、高値、安値、終値のディメンションに制限します。
- [Volume] (ボリューム) は、パネルディメンションをボリュームディメンションに制限します。
- [Both] (両方) は、ローソク足パネルのデフォルトの動作です。これには、ローソク足とボリュームの両方の視覚化が含まれます。

Candle style (ローソクのスタイル)

- [Candles] はデフォルトの表示スタイルで、始値ディメンションと終値ディメンションとの間にローソクスタイルの視覚化が作成されます。
- [OHLC 棒] は、4 つのコアディメンションである始値、高値、安値、終値を表示します。

Color strategy (カラー戦略)

- [始値から] は、デフォルトの動作です。このモードは、期間内の料金変動が正の場合、Up カラー (下部) を使用します。つまり、クローズ時の値がオープン時の値以上である場合、Up カラーが使用されます。
- [前の終値以降] は、ローソクの色が中間期の価格変動または値の変化に基づいている代替の表示方法です。つまり、オープン時の値がクローズ時の以前の値より大きい場合、Up カラーが使用されます。オープン時の値がクローズ時の以前の値よりも低い場合、Down カラーが使用されます。このオプションでは、中空ローソク足グラフ作成モードもトリガーされます。中空ローソク足は、期間内の変動が正 (値はオープン時よりもクローズ時が高い) であることを示します。塗りつぶされたローソク足は、期間内の変動が負 (値はオープン時よりもクローズ時が低い) であることを示します。詳細については、「違いの説明」を参照してください。

アップカラーとダウンカラー

[Up color] と [Down color] オプションは、料金変動がアップまたはダウンするときにどの色を使用するかを選択します。上記のカラー戦略は、ローソクまたは OHLC バーの色を選択するために、期間内または中間期の料金変動を使用するかどうかを決定します。

始値、高値、安値、終値

ローソク足パネルは、フィールドを適切なディメンションにマッピングしようとします。[始値]、[高値]、[安値]、および [終値] オプションを使用すると、パネルが適切なマッピングをできない場合に、データをこれらのディメンションにマッピングできます。

> **Note**
>
> これらの値は凡例から非表示になります。

- [始値] は、指定された期間の開始値に対応します。
- [高値] は、指定された期間の最大値に対応します。
- [安値] は、指定された期間の最小値に対応します。
- [終値] は、指定された期間の最終 (終了) 値に対応します。
- [ボリューム] は、特定の期間のサンプル数に対応します。(取引数など)

追加フィールド

ローソク足パネルは時系列パネルに基づいています。始値、高値、安値、終値、ボリューム以外の追加のデータディメンションを視覚化できます。[含める] オプションと [無視] オプションを使用すると、パネルは 時系列パネル で使用できるのと同じスタイルと設定を使用して、単純な移動平均、ボリンジャーバンドなど、含まれている他のデータを視覚化できます。

Canvas パネル

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Canvas は、Grafana の権限とカスタム要素の柔軟性を組み合わせた新しいパネルです。Canvas 視覚化は、静的レイアウトと動的レイアウトに要素を明示的に配置できる拡張可能なフォームビルドパネルです。これにより、Grafana の UI 内全体で、標準の Grafana パネルでは不可能な方法でカスタムビジュアライゼーションとオーバーレイデータを設計できます。一般的な UI およびウェブ設計ツールを使用したことがある場合、キャンバスパネルの設計は非常に馴染みのあるものになります。

[Elements] (要素)

メトリクス値

メトリクス値要素を使用すると、キャンバスに表示するデータを簡単に選択できます。この要素には、コンテキストメニューの「編集」オプションを通じて、またはパネルをダブルクリックしてトリガーできる一意の「編集」モードがあります。編集モードでは、表示するフィールドデータを選択できます。

[テキスト]

テキスト要素を使用すると、キャンバスにテキストを簡単に追加できます。また、要素は、ダブルクリックまたはコンテキストメニューの編集メニューオプションによってトリガーされる編集モードもサポートします。

[Rectangle] (長方形)

長方形要素を使用すると、基本的な長方形をキャンバスに追加できます。長方形要素は、テキスト (固定データとフィールドデータの両方) の表示をサポートし、データのしきい値に基づいて背景色を変更できます。

[Icon] (アイコン)

アイコン要素を使用すると、サポートされているアイコンをキャンバスに追加できます。アイコンは、しきい値または値マッピングに基づいて色を設定できます。

Canvas の編集

[インラインエディタ]

Canvas は新しい編集エクスペリエンスを導入しました。ダッシュボードモードのコンテキストで、キャンバスパネルをインラインで編集できるようになりました。

コンテキストメニュー

コンテキストメニューでは、一般的なタスクにアクセスできます。サポートされている機能には、インラインエディタの開閉、要素の複製、要素の削除などがあります。

コンテキストメニューは、パネルまたは特定のキャンバス要素に対する右クリックアクションによってトリガーされます。パネルを右クリックすると、背景イメージを設定し、キャンバスに要素を簡単に追加できます。

要素を右クリックすると、要素を編集、削除、複製したり、要素のレイヤーの位置を変更したりできます。

キャンバスオプション

インライン編集

インライン編集トグルを使用すると、キャンバスパネルをロックまたはロック解除できます。オフにすると、キャンバスパネルがロックされ、要素が所定の位置にフリーズし、意図しない変更を防止できます。

時計パネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

時計パネルは、現在の時刻またはカウントダウンを表示します。毎秒更新されます。

- [モード] (モード) – デフォルトは時間です。カウントダウンを選択した場合、カウントダウンの期限を設定してカウントダウンを開始します。
- [12 時間または 24 時間表記] – 時間は 12 時間形式と 24 時間形式のいずれかを選択できます。
- [タイムゾーン] – タイムゾーンは、moment timezone のライブラリから提供されます。デフォルトでは、使用しているコンピュータのタイムゾーンが使用されます。
- [カウントダウン期限] – [モード]を[カウントダウン]に設定した場合、カウントダウンが終わる日時を指定します。
- [カウントダウン終了テキスト] – カウントダウンが終了するタイミングで表示されるテキストを指定します。
- 日付/時刻の書式設定オプション – フォントサイズ、フォントの太さ、日付/時刻の書式設定をカスタマイズします。カウントダウンを表示していて、秒単位のカウントダウンを表示したくない場合は、24 時間表記の場合は `hh:mm`、12 時間時表記の場合は `h:mm A` に時間形式を変更してください。オプションの完全な一覧については、「表示」を参照してください。
- [背景色] – 時計の背景色を選択します。

ダッシュボードの一覧表示

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ダッシュボードリストの視覚化により、他のダッシュボードへの動的リンクを表示できます。リストは、スター付きダッシュボード、最近表示されたダッシュボード、検索クエリ、ダッシュボードタグを使用できるよう設定が可能です。

ダッシュボードのロードごとに、このパネルはダッシュボードリストに対してクエリを実行し、常に最新の結果を表示します。

オプション

これらのオプションで、視覚化する内容を絞り込みます。

- [Starred] (スター付き) – スター付きダッシュボードをアルファベット順に表示します。
- [最近表示] – 最近表示したダッシュボードをアルファベット順に表示します。
- [検索] – 検索クエリまたはタグでダッシュボードを表示します。クエリまたはタグに少なくとも 1 つの値を入力する必要があります。クエリフィールドとタグフィールドでは、`$my_var` や `${my_var}` など、変数補間がサポートされています。
- [見出しの表示] – 選択したリスト選択 (スター付き、最近表示、検索) が見出しとして表示されます。
- [最大項目数] – セクションごとに一覧表示する項目の最大数を設定します。例えば、これをデフォルト値の 10 のままにして、スター付きダッシュボードと最近表示ダッシュボードを表示した場合、パネルには各セクションに 10 個、合計 20 個のダッシュボードが表示されます。

[検索]

これらのオプションは、検索オプションが選択されている場合にのみ適用されます。

- [クエリ] – 検索するクエリを入力します。クエリでは大文字と小文字が区別されず、部分的な値も受け入れられます。
- [フォルダ] – 表示するダッシュボードフォルダを選択します。
- [タグ] – ここに検索するタグを入力します。既存のタグは入力時に表示されず、大文字と小文字が区別されます。

> **Note**
>
> 複数のタグと文字列が表示されると、ダッシュボードリストにすべての条件に一致するタグと文字列が表示されます。

ゲージパネル

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ゲージは、各系列、列、または行に対してゲージを繰り返すことができる単一値の視覚化です。

値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示]

Grafana がデータを表示する方法を選択します。

[計算]

すべての行を元に計算した値を表示します。

- [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算のリストについては、「計算タイプ」を参照してください。

- [フィールド] – パネルに表示するフィールドを選択します。

[すべての値]

行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

- [制限] – 表示する行の最大数。デフォルトは 5,000 です。
- [フィールド] – パネルに表示するフィールドを選択します。

ゲージ

ゲージの表示方法を設定します。

- [しきい値ラベルの表示] – しきい値を表示するかどうかを制御します。
- [しきい値マーカーの表示] – しきい値バンドを内部ゲージ値バンドの外部に表示するかどうかを制御します。

[テキストサイズ]

ゲージテキストのサイズを調整します。

- [タイトル] – ゲージタイトルサイズの数値を入力します。
- [値] – ゲージ値サイズの数値を入力します。

ジオマップパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ジオマップパネルの視覚的表現により、地理空間データを用いて世界地図を表示、カスタマイズできます。さまざまなオーバーレイスタイルとマップビュー設定を設定して、データの重要な位置情報に注目しやすくすることができます。

マップビュー

マップビューは、ダッシュボードのロード時にマップの初期ビューを制御します。

[初期ビュー]

初期ビューは、パネルが最初にロードされたときに GeoMap パネルがどのようにレンダリングされるかを設定します。

- [ビュー] は、パネルが最初にロードされたときにマップの中心を設定します。
  - [Fit to data] は、データが変更されたときのマップレイヤーと更新のデータ範囲に基づいてマップビューに適合します。
    - [データ] オプションでは、[すべてのレイヤー] のデータ、単一の [レイヤー]、または選択したレイヤーの [最後の値] に基づいて範囲を選択できます。
    - [レイヤー] において、単一の [レイヤー] またはレイヤーの [最後の値] からデータをフィッティングする場合に選択できます。
    - [パディング] は、データ範囲を超えた相対的な割合でパディングを設定します ([最後の値] のみを参照する場合は使用できません)。
    - [Max Zoom] は、データをフィッティングするときに最大ズームレベルを設定します。
  - [座標] は、以下に基づいてマップビューを設定します。
    - Latitude (緯度)
    - Longitude (経度)
  - デフォルトビューには、以下を含むものもあります。
    - (0°, 0°)
    - 北米
    - 南米
    - 欧州
    - アフリカ
    - 西アジア
    - 南アジア
    - 東南アジア

  - 東アジア
  - オーストラリア
  - オセアニア
- [Zoom] は、初期ズームレベルを設定します。

レイヤーのマッピング

Geomap 視覚化は、複数のレイヤーの表示をサポートします。各レイヤーは、ベースマップ上に地理空間データをどのように表示するかを決定します。

タイプ

ジオマップの視覚的表現では、3 つのマップレイヤータイプを選択できます。

- マーカーレイヤー は、各データポイントにマーカーをレンダリングします。
- ヒートマップレイヤー はデータのヒートマップを視覚化します。
- GeoJSON レイヤー は GeoJSON ファイルから静的データをレンダリングします。

また、現在アルファに入っているレイヤータイプは 5 つあります。

- 夜間/昼間レイヤー (アルファ) は夜間または昼間のリージョンをレンダリングします。
- [Icon at last point (alpha)] は、最後のデータポイントにアイコンをレンダリングします。
- [Dynamic GeoJSON (alpha)] は、クエリ結果に基づいて GeoJSON ファイルをスタイルします。
- [Route (alpha)] は、データポイントをルートとしてレンダリングします。
- 写真レイヤー (アルファ) は、各データポイントに写真をレンダリングします。

[レイヤーコントロール]

レイヤーコントロールを使用すると、レイヤーの作成、名前の変更、レイヤーの順序の変更、削除を行うことができます。

- [レイヤーを追加] で、Geomap 視覚化用に設定可能な追加のデータレイヤーを作成します。レイヤーを追加すると、レイヤータイプを選択するように求められます。レイヤータイプは、パネル設定中にいつでも変更できます。各レイヤータイプの詳細については、上記の「レイヤータイプ」セクションを参照してください。

- レイヤーコントロールを使用すると、パネルのレイヤーの名前変更、削除、順序変更をすることができます。
  - [レイヤー名の編集] (鉛筆アイコン) でレイヤーの名前を変更できます。
  - [Trash Bin] (ごみ箱) でレイヤーを削除できます。
  - [Reorder] (順序変更) (6 つのドット/グラブハンドル) では、レイヤーの順序を変更できます。上位レイヤーのデータは、下位レイヤーのデータの上に表示されます。パネルはドラッグアンドドロップするたびにレイヤーの順序を更新するため、レイヤーの順序の選択を簡素化します。

1 つのジオマップパネルに複数のデータレイヤーを追加して、リッチで詳細な視覚化を作成できます。

場所

ジオマップパネルには、地理的データの情報源が必要です。このデータには 4 つのマッピングオプションがあり、データベースクエリを使用して取得します。

- [自動] で口ケーションデータを自動的に検索します。クエリがデータフィールドの次のいずれかの名前に基づいている場合は、このオプションを使用します。
  - ジオハッシュ:「geohash」
  - 緯度:「latitude」、「lat」
  - 経度:「longitude」、「lng」、「lon」
  - ルックアップ:「lookup」
- Coords (座標): クエリに座標データが含めることを指定します。データベースクエリから、緯度と経度の数値データフィールドを選択するよう求められます。
- Geohash (ジオハッシュ): クエリがジオハッシュデータが含めることを指定します。データベースクエリから、ジオハッシュの文字列データフィールドを選択するよう求められます。
- [ルックアップ]: クエリに値にマッピングする必要がある口ケーション名データが含めることを指定します。データベースクエリと地名索引からルックアップフィールドを選択するよう求められます。地名索引は、クエリされたデータを地理的ポイントにマッピングするために使用されるディレクトリです。

マーカーレイヤー

マーカーレイヤーを使用すると、データポイントを円、正方形、三角形、星などのさまざまな形のマーカーとして表示することができます。

マーカーには多くのカスタマイズオプションがあります。

- [マーカーの色] で、マーカーの色を設定します。デフォルトの `Single color` では、すべてのポイントが 1 色に保たれます。`Thresholds` セクションで設定されたデータポイント値としきい値に応じて、複数の色を使用できる代替オプションがあります。
- マーカーサイズ: マーカーのサイズを設定します。デフォルトは `Fixed size` で、すべてのマーカーサイズはデータポイントに関係なく同じになります。ただし、対応するデータポイントに円を拡大縮小するオプションもあります。マーカーレイヤーの拡大縮小の範囲を指定するため、マーカーサイズの `Min` と `Max` を設定する必要があります。
- [Marker Shape] を使用すると、データに追加のビジュアルコンテキストを提供するのに役立つ図形、アイコン、またはグラフィックを選択できます。シンプルな図形や Unicon ライブラリなど、Grafana に含まれているアセットから選択します。画像アセットを含む URL を指定することもできます。画像は SVG (Scalable Vector Graphics: スケーラブルベクトルグラフィック) である必要があります。
- [塗りつぶしの透明度]: 各マーカーの透明度を設定します。

ヒートマップレイヤー

ヒートマップレイヤーは、さまざまなデータポイントをクラスター化して、密度の異なる場所を視覚化します。ヒートマップレイヤーを追加するには

[Data レイヤー] のドロップダウンメニューをクリックし、`Heatmap` を選択します。

`Markers` と同様に、視覚化するデータポイントとその視覚化方法を決定するオプションが表示されます。

- [重み値] では、ヒートマップクラスターの強度を設定します。`Fixed value` は、すべてのデータポイントで一定の重み値を維持します。この値は 0 ~ 1 の範囲で設定します。マーカーと同様に、ドロップダウンには、データ値に応じて重み値を自動的にスケールする代替オプションがあります。
- [Radius] (半径): ヒートマップクラスターのサイズを設定します。
- [Blur] (ぼかし) では、各クラスターのぼかしの量を設定します。

GeoJSON レイヤー

GeoJSON レイヤーを使用すると、ファイルシステムから静的 GeoJSON ファイルを選択してロードできます。

- [GeoJSON URL] では、Grafana に付属する GeoJSON ファイルを選択できます。
- [デフォルトのスタイル] では、上記のルールが一致しない場合に適用するスタイルを制御します。
  - [色] ではデフォルトのスタイルの色を設定します
  - [不透明度] ではデフォルトの不透明度を設定します
- [スタイルルール] では、機能のプロパティに基づいてスタイルを適用します
  - [ルール] を使用すると、GeoJSON ファイルから特徴量、条件、および[値]を選択して、ルールを定義できます。ごみ箱アイコンを使用して、現在のルールを削除できます。
  - [色] では、現在のルールに対するスタイルの色を設定します
  - [不透明度] では、現在のルールの透明性レベルを設定します
- [スタイルルールの追加] で、追加のスタイルルールを作成できます。

CARTO レイヤー

CARTO レイヤーは CARTO Raster ベースマップからのものです。

オプション

- Theme (テーマ)

  Light テーマ、Dark テーマ、または Auto テーマのいずれかのテーマを選択します。
- [ラベルの表示] で、マップの上部に国の詳細が表示されます。
- [不透明度] は 0 (透明) から 1 (不透明) です

XYZ タイルレイヤー

XYZ タイルレイヤーは、汎用タイルレイヤーからのマップです。

> **Note**
>
> 汎用タイルレイヤーの詳細については、「Web マップのタイル化」および「Open Street Map タイルサーバーの一覧」を参照してください。

オプション

- URL テンプレート

> **Note**
>
> `https://tile.openstreetmap.org/{z}/{x}/{y}.png` など、{z}/{x}/{y} が付いた有効なタイルサーバー URL を設定します。

- [Attribution] (属性) では、マップコントロールに表示されるレイヤーのリファレンス文字列を設定します。
- [不透明度] は 0 (透明) から 1 (不透明) です

Open Street Map レイヤー

Open Street Map からのマップで、これは共同による無料の地理的世界データベースです。

オプション

- [不透明度] は 0 (透明) から 1 (不透明) です

ArcGIS レイヤー

ArcGIS レイヤーは、ESRI ArcGIS MapServer からのレイヤーです。

オプション

- サーバーインスタンスで、次のマップタイプから選択します。
  - World Street Map (世界の道路地図)
  - World Imagery (衛星画像)
  - World Physical
  - Topographic (トポロジ)
  - USA Topographic (米国トポロジ)
  - World Ocean (海洋図)
  - Custom MapServer (書式設定については XYZ を参照)
    - URL テンプレート
    - Attribution (属性)
- [不透明度] は 0 (透明) から 1 (不透明) です

夜間/昼間レイヤー (アルファ)

夜間/昼間レイヤーには、現在の時間範囲に基づいて夜間および昼間の領域が表示されます。

> **Note**
> 詳細については、「OpenLayers の拡張機能 - DayNight」を参照してください。

オプション

- [表示] では、パネルの時間範囲から時間ソースを切り替えます
- [Night region color] では、夜間領域の色を選択します
- [Display sun] (太陽の表示) で太陽アイコンが切り替えられます
- [不透明度] は 0 (透明) から 1 (不透明) です

写真レイヤー (アルファ)

写真レイヤーは、各データポイントに写真をレンダリングします。

> **Note**
> 詳細については、「OpenLayers の拡張機能 - 画像写真スタイル」を参照してください。

オプション

- 画像ソースフィールド

  次のいずれかの形式で、画像データを含む文字列フィールドを選択します。
  - 画像 URL
  - Base64 でエンコードされたバイナリ画像 (`…`)
- [Kind] (種類)

  画像の周りのフレームスタイルを選択します
  - [Square] (平方)
  - [Circle] (円形)
  - [Anchored] (アンカー)

  - [Folio] (フォリオ)
- [Crop] (トリミング) では、画像がフィットするようにトリミングされるかどうか切り替えます
- [Shadow] (シャドウ) では、画像の背後にあるボックスシャドウを切り替えます
- [Border] (境界線) では、画像の周囲の境界サイズを設定します
- [境界線の色] では、画像の周囲の境界線の色を設定します
- [Radius] (範囲) では、画像の全体的なサイズをピクセル単位で設定します

マップコントロール

マップコントロールインターフェイスには、マップ情報とツールオーバーレイの以下のオプションが含まれています。

Zoom

このセクションでは、各ズームコントロールについて説明します。

Show zoom control

ズームコントロールを左上隅に表示します。

Mouse wheel zoom

マウスホイールを使用してズームインまたはズームアウトできる設定を、オンまたはオフにします。

Show attribution

マップ上のベースマップレイヤーの属性を表示します。

Show scale

スケール情報を左下隅に表示します。

> **Note**
>
> [m]/[km] 単位で表示します。

Show measure tools

測定ツールを右上隅に表示します。測定値は、このコントロールが開いている場合にのみ表示されます。

- クリックして測定を開始します
- 連続クリックで測定を続行します
- ダブルクリックして測定を終了します

> **Note**
>
> 測定タイプまたは単位を変更すると、以前の測定がマップから削除されます。
> コントロールを閉じてから再度開くと、最新の測定値が表示されます。
> 測定値は、クリックしてドラッグすることで変更できます。

Length

ジオメトリの球面長を取得します。この長さは、座標間の大圏距離の合計です。マルチパートジオメトリの場合、長さは各パートの長さの合計です。ジオメトリは「EPSG:3857」にあると仮定されます。

長さ測定には、次の単位を選択できます。

- メートル (m/km)
- フィート (ft)
- マイル (mi)
- 海里 (nmi)

[面積]

ジオメトリの球の表面積を取得します。この面積は、ポリゴンエッジが球上の大円のセグメントであると仮定して計算されます。ジオメトリは「EPSG:3857」にあると仮定されます。

面積測定には、次の単位を選択できます。

- 平方メートル (m²)
- 平方キロメートル (km²)
- 平方フィート (ft²)
- 平方マイル (mi²)

- エーカー (acre)
- ヘクタール (ha)

Show debug

デバッグ情報をマップの右上隅に表示します。これは、データソースのデバッグまたは検証に役立ちます。

- [Zoom] は、マップの現在のズームレベルを表示します。
- [Center] (中央) には、マップ中央の現在の経度と緯度が表示されます。

Tooltip (ツールチップ)

- [なし] では、データポイントがクリックされた場合にのみツールチップが表示されます。
- [Details] (詳細) では、データポイントにポインタを合わせると、ツールチップが表示されます。

グラフパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

グラフパネルは、線、ドットの軌跡、または一連のバーとしてレンダリングできます。このタイプのグラフは、ほぼすべての時系列データを表示するための十分な汎用性があります。

データオプションとフィールドオプション

グラフ視覚化を使用すると、次のオプションを適用できます。

- [データを変換する]
- アラート。これはアラートを設定できる唯一の視覚化タイプです。詳細については、「Grafana バージョン 9 のアラート」を参照してください。

- しきい値を設定する

表示オプション

視覚化を絞り込むためには、次の設定を使用します。

- [棒] – 値を棒グラフとして表示します。
- [線] – 値を折れ線グラフとして表示します。
- [線の幅] – 系列の線幅を指定します。デフォルトは 1 です。
- [階段] – 隣接するポイントを階段として描画します。
- [領域塗りつぶし] – シリーズの色塗りつぶし量を指定します。デフォルトは 1 で、0 はなしです。
- [塗りつぶしグラデーション] – エリア塗りつぶしのグラデーションの程度を指定します。デフォルトは 0 で、グラデーションはありません。10 は急なグラデーションです。
- [ポイント] – 値のポイントを表示します。
- [ポイント半径] – ポイントの大きさを制御します。
- [アラートしきい値] – アラートしきい値とリージョンをパネルに表示します。

スタッキングおよび null 値

- [スタック] – 各シリーズが別のシリーズの上にスタックされます。
- [パーセント] – 各系列が全系列の合計のパーセンテージとして描画されます。このオプションは [スタック] が選択されている場合に使用できます。
- [Null 値] – null 値の表示方法を指定します。これは重要な設定です。以下の注記を参照してください。
  - [接続] – 系列にギャップがある場合、つまり 1 つ以上の null 値がある場合、線はギャップをスキップして次の null 以外の値に接続します。
  - [null] - 系列にギャップがある場合、つまり null 値の場合、グラフの線は分割されてギャップが表示されます。これはデフォルトの設定です。
  - [ゼロとしての null] – シリーズにギャップがある場合、つまり null 値は、グラフパネルにゼロ値として表示されます。

> **Important**
>
> サーバーの CPU 負荷をモニタリングしているときに負荷が 100% に達すると、サーバーはロックアップされ、統計情報を送信するエージェントは負荷統計を収集できなくなりま

す。これによりメトリクスにギャップが生じ、デフォルトの null 設定を使用すると Amazon Managed Grafana にはギャップが表示され、何かが間違っていることを示します。これが[接続]に設定されていると、このシグナルを見逃しやすくなります。

ホバー時のツールヒント

これらの設定を使用して、グラフの視覚化上にカーソルを置いたときに表示されるツールヒントの外観を変更します。

- [モード] - カーソルのツールチップに表示される系列数を決定します。
  - [すべての系列] – カーソルのツールチップには、グラフ内の系列がすべて表示されます。Grafana ワークスペースは、ツールヒント内のシリーズリストで、一時停止しているシリーズを太字で強調表示します。
  - [単独] – ツールチップには、カーソルを合わせている系列が 1 つだけ表示されます。
- [ソート順序] – [すべての系列] モードを選択した場合、カーソルのツールチップにおいて系列の順序を並べ替えます。グラフで一時停止すると、Amazon Managed Grafana は線に関連付けられた値を表示します。一般的に、ユーザーは最大値や最小値に最も関心があります。これらの値をソートすると、目的のデータを簡単に見つけることができます。
  - [なし] – ツールチップ内のシリーズ順序は、クエリのソート順序によって決まります。例えば、シリーズ名によりシリーズをアルファベット順にソートできます。
  - [増加] – ホバー時のツールチップ内のシリーズは値が昇順にソートされ、リストの上部には最小値が表示されます。
  - [減少] – ホバー時のツールチップ内のシリーズは値が降順にソートされ、リストの上部に最大値が表示されます。

シリーズオーバーライド

シリーズオーバーライドを使用すると、グラフパネル内の系列を他の系列とは異なるレンダリングにできます。表示オプションは、シリーズごとに、または正規表現ルールを使用してカスタマイズできます。例えば、1 つのシリーズで線の幅を太くして目立つようにしたり、右の Y 軸に移動したりできます。

シリーズオーバーライドは複数追加できます。

シリーズオーバーライドの追加方法

1. [シリーズオーバーライドの追加] を選択します。

2. [エイリアスまたは正規表現] で、シリーズを入力または選択します。フィールドを選択すると使用可能なシリーズのリストが表示されます。

   例えば、`/Network.*/` は `Network out` および `Network in` という名前の 2 つのシリーズと一致します。

3. [+] を選択してからシリーズに適用するスタイルを選択します。各エントリには複数のスタイルを追加できます。

- [棒] – 系列を棒グラフとして表示します。
- [折れ線] – シリーズを折れ線グラフとして表示します。
- [折れ線塗りつぶし] – 領域塗りつぶしで折れ線グラフを表示します。
- [塗りつぶしグラデーション] – 領域塗りつぶしグラデーション量を指定します。
- [線の幅] – 線幅を設定します。
- [Null ポイントモード] – このオプションを使用して null 値を無視、またはゼロに置き換えます。これはデータ内ののギャップを無視する場合に重要です。
- [下を塗りつぶし] – 2 つのシリーズ間の領域を塗りつぶします。
- [階段線] — シリーズを階段線として表示します。
- [ダッシュ] – ダッシュで線を表示します。
- [シリーズを非表示] – シリーズを非表示にします。
- [ダッシュ長さ] – 線内のダッシュの長さを設定します。
- [ダッシュスペース] – 線内のダッシュ間のスペースの長さを設定します。
- [ポイント] – 個別のポイントとして系列を表示します。
- [ポイント半径] – ポイントのレンダリング半径を設定します。
- [スタック] – シリーズのスタックグループを設定します。
- [色] – シリーズの色を設定します。
- [Y 軸] – 系列の Y 軸を設定します。
- [Z インデックス] – 系列の z インデックス (レンダリング順序) を設定します。このオプションは、棒グラフや面グラフなどの、さまざまなスタイルをオーバーレイする場合に重要です。
- [変換] – 負の値に変換して Y 軸の下にレンダリングします。
- [凡例] – 凡例内にシリーズを表示するかどうかを制御します。
- [ツールヒントで非表示] – グラフのツールヒントにシリーズを表示するかどうかを制御します。

[Axes] (軸)

これらのオプションを使用して、視覚化内の軸表示を制御します。

[Left Y]/[Right Y]

オプションは両方の Y 軸で同じです。

- [表示] – 軸の表示または非表示を選択します。
- [単位] – Y 値の表示単位を選択します。
- [スケール] – Y 値に使用するスケールを、[線形]、または [対数] から選択します。デフォルトは [線形] です。
- [Y 最小] – Y の最小値です。デフォルトは [自動] です。
- [Y 最大] – Y の最大値です。デフォルトは [自動] です。
- [小数] – Y 値に表示される小数点以下の桁数を定義します。デフォルトは [自動] です。
- [ラベル] – Y 軸のラベルを指定します。デフォルトは [“",] です

Y 軸

- [配置] – 左右の Y 軸を値で配置します。デフォルトは [未チェック/偽] です。
- [レベル] – Y=0 から始まる、左右の Y 軸の配置に使用する値を入力します。デフォルトは 0 です。このオプションは、[配置] が選択されている場合に使用できます。

X 軸

- [表示] – 軸の表示または非表示を選択します。
- [モード] – 表示モードではグラフパネルの視覚化が完全に変更されます。3 つのパネルを 1 つにしたようなものです。メインモードは、X 軸の時間による時系列モードです。他の 2 つのモードは、時間とヒストグラムのモードではなく、X 軸上のシリーズ使用する基本的な棒グラフモードです。
  - [時間] (デフォルト) – X 軸は時間を表し、データは時間 (時間、分など) でグループ化されます。
  - [系列] – データは時間ではなく系列でグループ化されます。Y 軸は引き続き値を表します。
    - [値] – これは値のために使用する集計タイプです。デフォルトは [合計] (値を合計) です。
  - [ヒストグラム] – このオプションではグラフをヒストグラムに変換します。ヒストグラムは数値を範囲にグループ化する棒グラフの一種で、バケットやビンとも呼ばれます。背の高い棒は、より多くのデータがその範囲内にあることを示します。

ヒストグラムの詳細については、「ヒストグラムおよびヒートマップの概要」を参照してください。

- [バケット] – 値をグループ化するためのバケットの数を設定します。空にすると、Amazon Managed Grafana は適切な数のバケットを計算しようとします。
- [X 最小] – この下限未満のヒストグラムから値をフィルタリングします。
- [X 最大] – この上限を超える値をフィルタリングします。

凡例

これらの設定を使用して、グラフ内の凡例の表示方法を改善します。

オプション

- [表示] – クリアすると凡例は非表示になります。デフォルトが選択されます (真)。
- [テーブルとして] – 選択すると凡例がテーブルに表示されます。デフォルトがチェックされます (真)。
- [右側] - 選択すると右側に凡例が表示されます。
- [幅] – 凡例の最小幅をピクセル単位で入力します。このオプションは、[右側] を選択した場合に使用できます。

値

凡例名の横に追加の値を表示できます。

- [最小] – メトリクスクエリから返される最小値です。
- [最大] – メトリクスクエリから返される最大値です。
- 平均 – メトリクスクエリから返される平均値です。
- [現在] – メトリクスクエリから返された最後の値です。
- [合計] – メトリクスクエリから返されるすべての値の合計です。
- [小数] – 凡例値およびグラフカーソルのツールチップに表示される小数点以下の桁数です。

Amazon Managed Grafana はクライアント側の凡例値を計算します。これらの凡例値は、メトリクスクエリが使用している集計またはポイント統合のタイプによって異なります。上記の凡例値がすべて同時に正確になることはありません。

例えば、おそらく平均をアグリゲーターとして使用する、リクエスト/秒の割合などをプロットする場合、凡例の合計はリクエストの合計数を表しません。これは Amazon Managed Grafana が受信したすべてのデータポイントの合計にすぎません。

シリーズの非表示

メトリクスクエリからのシリーズの値すべてが特定の値であるときに、シリーズを非表示にします。

- [null のみを使用] – Value=null (デフォルトは未チェック)
- [ゼロのみを使用] – Value=ゼロ (デフォルトは未チェック)

時間リージョン

週末、営業時間、勤務時間外など、グラフ上の特定の時間リージョンを強調表示して見やすくできます。設定されたすべての時間リージョンは UTC 時間です。

ヒートマップパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ヒートマップパネルの視覚化により、ヒストグラムを経時的に表示できます。ヒストグラムの詳細については、「ヒストグラムおよびヒートマップの概要」を参照してください。

データから計算する

この設定は、データが既に計算されたヒートマップ (データソース/トランスフォーマーから) であるか、パネルで計算する必要があるヒートマップであるかを決定します。

X バケット

この設定は、X 軸をバケットに分割する方法を決定します。サイズ入力で時間間隔を指定できます。例えば、1h の時間範囲により、セルは X 軸で 1 時間幅になります。

Y バケット

この設定は、Y 軸をバケットに分割する方法を決定します。

Y バケット目盛

次のいずれかの Y 軸値目盛を選択します。

- [線形] – 線形目盛。
- [log (base 2)] – 底が 2 の対数目盛。
- [log (base 10)] – 底が 10 の対数目盛。

Y 軸

Y 軸の表示方法を定義します。

[配置]

- [左] – 左側
- [Right] – 右側
- [非表示] – 非表示

[単位]

単位設定

[小数]

この設定により、小数点以下の設定が決まります。

[最小/最大値]

この設定では、軸範囲を設定します。

[Reverse] (逆順)

選択すると、軸は逆の順序で表示されます。

カラー

色のスペクトルは、値の数 (各バケット内) と各バケットに割り当てられた色間とのマッピングを制御します。スペクトルの左端の色は最低数を表し、右端の色は最大数を表します。一部のカラースキームは、ライトテーマを使用するときに自動的に反転されます。

また、カラーモードを [不透明度] に変更することもできます。この場合、色は変わりませんが、不透明度はバケット数に応じて変化します。

- モード
  - [Scheme] (スキーム) – セルの色で表されるバケット値。
    - [Scheme] – モードがスキームの場合、カラースキームを選択します。
  - [不透明度] – セルの不透明度で表されるバケット値。[Opaque] (不透明) セルは最大値を意味します。
    - [Color] – セルのベース色。
    - [Scale] (目盛) – バケット値を不透明度にマッピングするための目盛。
      - [線形] – 線形目盛。バケット値は不透明度に対して直線的にマッピングされます。
      - [sqrt] (平方根) – 対数目盛。セルの不透明度は `value ^ k` として計算され、`k` は設定された指数値です。指数が `1` より小さい場合、対数スケールが表示されます。指数が `1` より大きい場合、指数スケールが表示されます。`1` の場合、目盛は線形と同じになります。
    - [Exponent] – 指数の値 (`0` より大きい場合)。

[値を元にした開始/終了の色]

デフォルトでは、Grafana は最小および最大バケット値に基づいてセルの色を計算します。[最小値] と [最大値] では、これらの値を上書きできます。バケット値は Z 軸、最小値と最大値はそれぞれ Z の最小値と Z の最大値と考えてください。

- [開始値] – セルの色の計算に使用される最小値。バケット値が最小値より小さい場合、「最小値」の色にマッピングされます。系列の最小値はデフォルト値です。
- [終了値] – セルの色の計算に使用される最大値。バケット値が最大値より大きい場合、「最大値」の色にマッピングされます。系列の最大値はデフォルト値です。

セル表示

セル表示設定を使用して、ヒートマップ内のセルの視覚化を絞り込みます。

追加の表示オプション

Tooltip (ツールチップ)

- [Show tooltip] - ヒートマップのツールチップを表示します。

- [Show Histogram] – ツールチップに Y 軸のヒストグラムを表示します。ヒストグラムは、特定のタイムスタンプにおけるバケット値の分布を表します。

[凡例]

視覚化にヒートマップの凡例を表示するかどうかを選択します。

Exemplars (見本)

見本データの表示に使用する色を設定します。

ヒストグラムパネル

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ヒストグラムの視覚化は値の分布を計算して棒グラフとして表示します。Y 軸と各バーの高さは各括弧に含まれる値の数を表し、X 軸は値の範囲を表します。

ヒストグラムの視覚化では、1 つ以上の数値フィールドを持つ時系列とテーブル結果をサポートしています。

サポートされる形式

ヒストグラムの視覚化では、1 つ以上の数値フィールドを持つ時系列とテーブル結果をサポートしています。

表示オプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[バケットサイズ]

バケットのサイズです。自動バケットサイズ設定 (全範囲の 10% まで) では、この値は空のままにします。

[バケットオフセット]

最初のバケットを 0 で始めない場合です。ゼロ以外のオフセットでは集計ウィンドウがシフトします。例えば、デフォルトの 0 オフセットの 0～5、5～10、10～15 の 5 サイズのバケットは、オフセット 2 の 2～7、7～12、12～17 になる場合があります。この場合、オフセット 0、5、10 であれば事実上何も実行されません。通常、このオプションは自動ではなく、明示的に定義されたバケットサイズで使用します。この設定が作用するためには、オフセット量を 0 より大きく、バケットサイズより小さくする必要があります。この範囲外の値には、この範囲内の値と同じ効果があります。

[系列の結合]

これにより、すべての系列とフィールドが結合されたヒストグラムにマージされます。

[線の幅]: 棒の線幅を制御します。

[塗りつぶしの不透明度]: 棒の塗りつぶし不透明度を制御します。

[グラデーションモード]: グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーション塗りつぶしなし。これはデフォルト設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。
- [色相] – グラデーション色は線の色相に基づいて生成されます。

[ツールチップモード] グラフにカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、カーソルを合わせている系列のみが表示されます。
- [すべて] – ホバー時のツールチップには視覚化のすべてのシリーズが表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – ツールチップを表示しません。

> **Note**
>
> オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。詳細については、「計算タイプ」を参照してください。

[凡例計算]

凡例に表示する計算を選択します。複数選べます。

ログパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ログパネルの視覚化では、Elastic、Influx、Loki などのログをサポートするデータソースのログ行が表示されます。通常、グラフパネルの横のこのパネルを使用して、関連するプロセスのログ出力を表示します。

ログパネルには、[Query] タブに入力されたクエリの結果が表示されます。複数のクエリ結果がマージされ、時間別にソートされます。表示可能な行よりも多くの行がデータソースから返される場合は、パネル内でスクロールできます。

レンダリングされる行数を制限するには、[クエリオプション]で [最大データポイント数] の設定を使用できます。設定しない場合、データソースは通常デフォルトの制限を適用します。

ログレベル

レベルラベルが指定されているログの場合、ラベルの値を使用してログレベルを決定し、それに応じて色を更新します。ログに指定されたレベルラベルがない場合、そのコンテンツが、サポートされている表現のいずれかと一致するかどうかを調べます (詳細については、下記を参照してください)。ログレベルは常に最初の一致によって決まります。Grafana がログレベルを判別できない場合、不明なログレベルとして視覚化されます。詳細については、「ログの視覚的表現」を参照してください。

ログの詳細

各ログ行には、より堅牢なインタラクションのために、ラベルと検出されたフィールドを含む拡張可能な領域があります。各フィールドまたはラベルには、表示されているすべてのログに関連する統計を表示する統計アイコンがあります。

データリンク

データリンクを使用すると、ログメッセージの任意の部分を内部リンクまたは外部リンクに変換できます。作成されたリンクは、[ログの詳細] ビュー内の [リンク] セクションにボタンとして表示されます。

表示オプション

以下の設定を使用して、視覚化する内容を絞り込みます。

- [時間] – 時間列を表示または非表示にします。これはデータソースから報告されたログ行と関連するタイムスタンプです。
- [一意のラベル] – 一意のラベル列を表示または非表示にします。この列には、共通ではないラベルのみが表示されます。
- [共通ラベル] - を表示または非表示にします

- [行の折り返し] – 行の折り返しを切り替えます。
- [Prettify JSON] – すべての JSON ログを pretty print (整形表示) するには、これを `true` に設定します。この設定は、JSON 以外の形式のログには影響しません。
- [ログ詳細の有効化] – オプションを切り替えて、各ログ行のログ詳細ビューを表示します。デフォルトの設定は`true` です。
- [Order] (順序) – 結果を降順または昇順で表示します。デフォルトは [降順] で、最新のログが最初に表示されます。[昇順] に設定すると最も古いログ行が最初に表示されます。

ニュースパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このパネルには RSS フィードが表示されます。デフォルトでは、Grafana Labs のブログ記事が表示されます。

[表示] セクションに RSS の URL を入力します。このパネルタイプは他のクエリを受け入れません。

Note

RSS フィードは、プロキシなしで Grafana フロントエンドによって読み込まれます。その結果、適切な CORS ヘッダーで設定された RSS フィードのみが読み込まれます。表示しようとしている RSS フィードが読み込みに失敗する場合は、RSS フィードを再ホストするか、独自のプロキシ作成を検討してください。

ノードグラフパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ノードグラフパネルは有向グラフまたはネットワークを視覚化します。有向フォースレイアウトを使用してノードを効果的に配置し、複雑なインフラストラクチャマップ、階層、または実行図の表示に役立ちます。

データ要件

ノードグラフのパネルでは、ノードとエッジを表示するためにデータの特定の形状が必要です。このパネルでは、すべてのデータソースやクエリを視覚化できるわけではありません。

ノードグラフの視覚化は、ノードおよびエッジで構成されます。

- あるノードは円として表示されます。あるノードは、アプリケーション、サービス、またはアプリケーションの観点から関連するその他のものを表す場合があります。
- あるエッジは、2 つのノードを接続する線として表示されます。接続は、リクエスト、オペレーション、または 2 つのノード間の他の関係である場合があります。

ノードと エッジの両方にメタデータまたは統計を関連付けることができます。データソースは、表示される情報と値を定義するため、異なるデータソースが異なるタイプの値を表示したり、一部の値を表示しなかったりすることができます。

ノード

通常、ノードには、ノード内の 2 つの統計値とノードのすぐ下の 2 つの識別子、通常は名前とタイプが表示されます。ノードでは、ノードの周囲に一連の別の値を色付きの円として表示することもできます。異なる色のセクションは、合計が 1 になる異なる値を表します。例えば、円の赤い部分によって エラーの割合を表すことができます。

ノードを選択すると表示されるコンテキストメニューには、さらに詳細を表示できます。Grafana ワークスペースの他の部分または外部リンクをターゲットにできる追加のリンクを、コンテキストメニューに含めることもできます。

> **Note**
>
> ノードグラフには 1,500 個のノードしか表示できません。この制限を超えると、右上隅に警告が表示され、一部のノードは非表示になります。グラフ内の [非表示ノード] マーカーをクリックすると、グラフの非表示部分を展開できます。

エッジ

エッジでは、エッジにカーソルを合わせると統計を表示することもできます。ノードと同様に、エッジを選択すると、追加の詳細とリンクを含むコンテキストメニューを開けます。

この視覚化をサポートする最初のデータソースは、サービスマップ機能の AWS X-Ray データソースです。詳細については、「AWS X-Ray データソースに接続する」を参照してください。

ノードグラフの操作方法

ノードグラフ内でパンするには、ノードまたはエッジの外部を選択してポインタをドラッグします。

ノードグラフの左上隅にあるボタンを使用するとズームできます。

非表示ノード

特定の時間に表示されるノードの数は、合理的なパフォーマンスを維持するために制限されます。この制限外のノードは、そのエッジに接続されている非表示ノードのおよその数を示すクリック可能なマーカーの背後に隠されます。マーカーを選択して、そのノードの周りのグラフを展開できます。

グリッドビュー

グリッドビューに切り替えて、グラフ内の最も興味深いノードの概要を把握できます。グリッドビューは、エッジのないグリッド内のノードを表示し、ノード内に表示される統計、またはノードの色付き境界で表される統計でソートできます。

ノードをソートするには、凡例内の統計を選択します。統計名の横にあるマーカーは、ソートおよびソート方向に現在使用されている統計を示します。

ノードを選択し、[グラフレイアウトで表示] オプションで、選択したノードに焦点を当ててグラフレイアウトに切り替え、グラフ全体のコンテキストで表示します。

Data API

この視覚化では、正しく表示するために、データソースから返されるデータの特定の形状が必要です。

ノードグラフには、少なくともグラフのエッジを記述するデータフレームが必要です。デフォルトでは、ノードグラフはこのデータフレームに基づいてノードと統計を計算します。必要に応じて、ノード固有のメタデータをより多く表示する必要がある場合に備えて、ノードを記述する 2 番目のデータフレームを送信できます。ノードグラフをレンダリングするには、両方のデータフレームに `frame.meta.preferredVisualisationType = 'nodeGraph'` を設定するか、`nodes` と `edges` にそれぞれ名前を付ける必要があります。

構造からのエッジデータ

必須フィールド

フィールド名	タイプ	説明
id	文字列	エッジの一意の識別子。
ソース	文字列	ソースノードの ID。
target	文字列	ターゲットの ID。

オプションのフィールド

フィールド名	タイプ	説明
Mainstat	文字列/数値	エッジにカーソルを合わせたときのオーバーレイに表示される最初の統計。値はそのまま示す文字列でも、数値でもかまいません。数値の場合、そのフィールドに関連付けられている単位も表示されます。
secondarystat	文字列/数値	mainStat と同じですが、そのすぐ下に表示されます。
detail__*	文字列/数値	プレフィックスが `detail__` の任意のフィールドは、エッジをクリックするとコンテ

フィールド名	タイプ	説明
		キストメニューのヘッダーに表示されます。より判読可能なラベルに config.displayName を使用します。

構造からのノードデータ

必須フィールド

フィールド名	タイプ	説明
id	文字列	ノードの一意の識別子。この ID は、ソースフィールドとターゲットフィールドのエッジによって参照されます。

オプションのフィールド

フィールド名	タイプ	説明
title	文字列	ノードのすぐ下に表示されるノードの名前。
subtitle	文字列	タイトルの下に表示される追加の名前、タイプ、またはその他の識別子。
mainstat	文字列/数値	ノード自体の中に表示される最初の統計。これは、値をそのまま表示する文字列でも数値でもかまいません。数値の場合、そのフィールドに関連

フィールド名	タイプ	説明
		付けられている単位も表示されます。
secondarystat	文字列/数値	mainStat と同じですが、ノード内でその下に表示されます。
arc__*	数値	プレフィックスが `arc__` の任意のフィールドは、ノードの周囲に色円を作成するために使用されます。これらのフィールドのすべての値は、合計で 1 になる必要があります。`config.color.fixedColor` を使用して色を指定できます。
detail__*	文字列/数値	ノードをクリックすると、プレフィックスが `detail__` の任意のフィールドがコンテキストメニューのヘッダーに表示されます。より人間が判読可能なラベルに `config.displayName` を使用します。

円グラフパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

円グラフには、1 つ以上のクエリから誘導された系列、または系列内の値が、円グラフのスライスで相互に関連して表示されます。スライスの円弧の長さ、面積、中心角度は、すべての値の合計に関連するため、すべてスライス値に比例します。この種類のグラフは、一連の小規模な値を見た目がきれいな形ですばやく比較する場合に最適です。

値オプション

次のオプションを使用して、視覚化に使う値を絞り込みます。

[表示]

表示する情報量を選択します。

- [計算] – 各値を系列ごとに単一の値に減らします。
- [すべての値] – 1 つの系列におけるすべての値を表示します。

計算

計算を選択すると、各系列を削減する計算を選択できます。使用可能な計算の詳細については、「計算タイプ」を参照してください。

[制限]

1 つの系列におけるすべての値を表示する場合、表示される値の数には制限があります。

[フィールド]

視覚化するフィールドを 1 つ以上選択します。各フィールド名はリストで確認できます。または、次のいずれかのオプションを選択できます。

- [数値フィールド] – 数値を持つすべてのフィールド。
- [すべてのフィールド] – 変換によって削除されないすべてのフィールド。
- [時間] – 時間値を持つすべてのフィールド。

円グラフオプション

これらのオプションで、視覚化の見え方を調整します。

円グラフタイプ

円グラフの表示スタイルを選択します。次のいずれかになります。

- [Pie] (円) – 標準円グラフ
- [Donut] (ドーナツ) – 中央に穴がある円グラフ

ラベル

円グラフに表示するラベルを選択します。複数選べます。

- [名前] – 系列名またはフィールド名。
- [パーセント] – 全体の割合。
- [Value] – 未加工の数値。

デフォルトでは、ラベルはグラフ部分の上に白で表示されます。グラフの色を濃く選択して、より見やすくすることができます。長い名前や数字は文字切れする可能性があります。

[ツールチップモード]

視覚化にカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、グラフ上でカーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

[凡例モード]

これらの設定を使用して、グラフで凡例がどのように表示されるかを定義します。凡例の詳細については、「凡例の設定」を参照してください。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置]

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

凡例に表示する値を選択します。複数選べます。

- [パーセント] – 全体の割合。
- [Value] – 未加工の数値。

凡例の詳細については、「凡例の設定」を参照してください。

Plotly パネル

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Plotly パネルは、オープンソースの javascript グラフ作成ライブラリである Plotly を使用してグラフをレンダリングします。

データ、レイアウト、および設定 フィールドは、「Plotly ドキュメント」で説明されている一般的なパラメータと一致します。これらは JSON 形式である必要があります。

データソースによって提供されるデータは、Plotly グラフに挿入される前に、ユーザー定義スクリプトを介して変換できます。スクリプトには 2 つの引数が含まれます。

- `data` – データソースから返されるデータ。
- `variables` – 現在のダッシュボードに Grafana 変数を含むオブジェクト (ユーザー変数と、グローバル変数である `__from`、`__to`、`__interval`、`__interval_ms`)。

スクリプトは、data、layout、config、frames のプロパティから 1 つ以上を有するオブジェクトを返す必要があります。以下に例を示します。

```
let x  = data.series[0].fields[0].values.buffer
let y  = data.series[0].fields[1].values.buffer
let serie = {
    x : x,
    y : y,
    name : variables.project //where project is the name of a Grafana's variable
}

return {
    data : [serie],
    config : {
    displayModeBar: false
    }
}
```

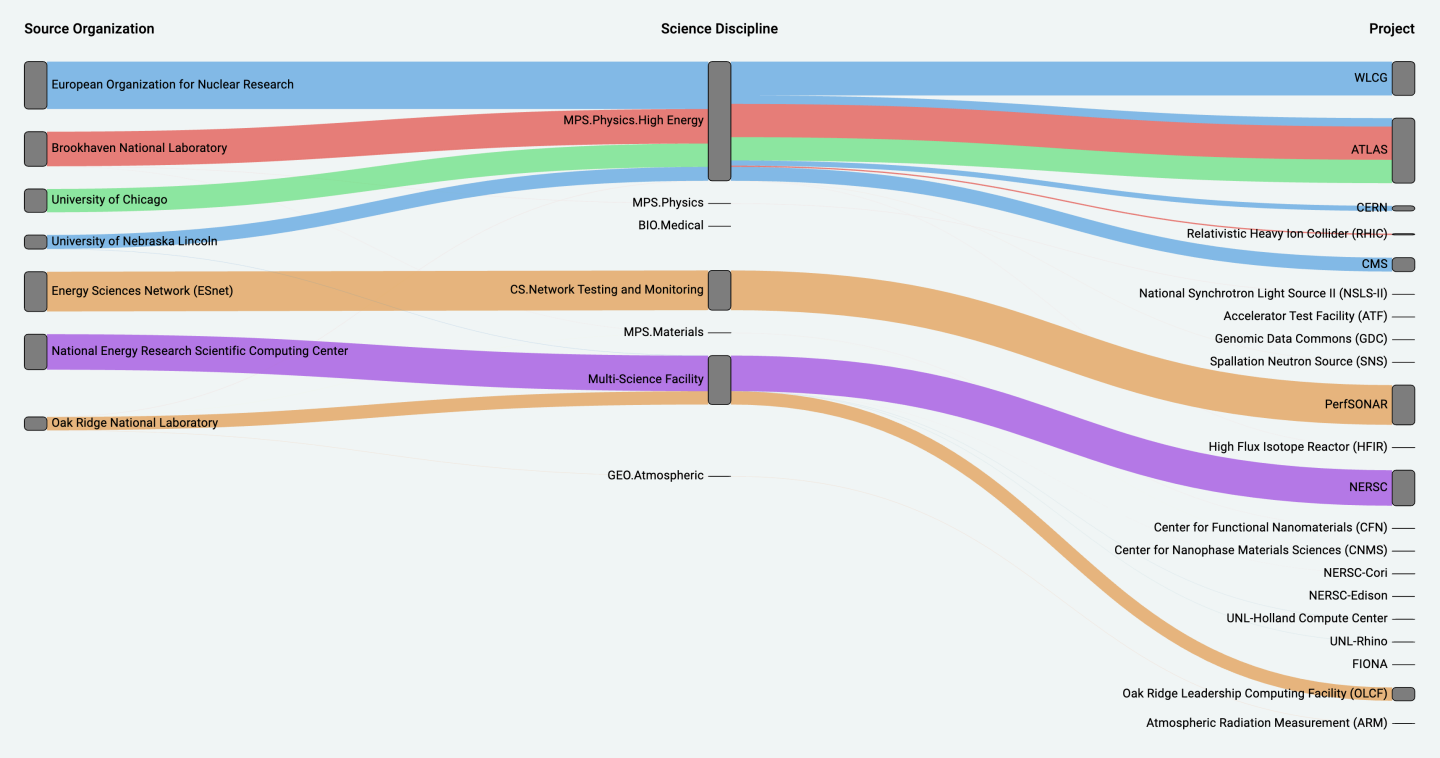
データ、レイアウト、および 設定 フィールドで指定されたスクリプトと JSON によって返されるオブジェクトはマージされます (ディープマージ)。

スクリプトが指定されていない場合、パネルはデータ、レイアウト、および 設定 フィールドのみを使用します。

サンキーパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

サンキーパネルにはサンキーダイアグラムが表示されます。サンキーダイアグラムは、選択したメトリクスに比例したフローの幅で、フローデータを視覚化するのに適したものです。次の図は、ソースとターゲットの 2 つのグループを含むサンキーダイアグラムを示しています。



仕組み

サンキーパネルには、フローのソースとターゲットの 2 つ以上の列のデータが必要です。クエリは、データを少なくとも 2 つのグループにグループ化する必要があります。パネルは、データポイントの最初の列からクエリの順序で最後の列へのリンクを描画します。リンクの厚さは、クエリ内のメトリクスによって割り当てられた値に比例します。

カスタマイズ

- リンク – 現在、リンクの色にはマルチまたはシングルの 2 つのオプションがあります。デフォルトでは、マルチカラーです。リンクの単一色を選択するには、[単一リンクの色のみ]オプションを切り替え、Grafana のカラーピッカーから色を選択します。
- [ノード] – [ノードの色] オプションを変更することで、長方形ノードの色を変更できます。
- [ノード幅] – ノードの幅は、[ノード幅] スライダーで調整するか、入力ボックスに数値を入力することで調整できます。数値は整数である必要があります。
- [ノードパディング] – ノード間の垂直パディングは、[ノードパディング] スライダーで調整するか、入力ボックスに数値を入力することで調整できます。数値は整数である必要があります。リンクが細すぎる場合は、この数値を調整してみてください。
- ヘッダー – 列ヘッダーは、エディタパネルの [表示名] オーバーライドを使用して変更できます。[テキスト色] で選択した色と同じ色になります

- [サンキーレイアウト] – サンキーリンクのレイアウトは、[レイアウト反復] スライダーを使用して微調整できます。この数は整数でなければならず、レイアウトの生成に使用される緩和反復の数です。

散布図パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

散布図パネルは、他のグラフパネルよりもシンプルなインターフェイスを持つテーブルデータの X/Y 散布図を示しています。グラフパネルとは異なり、散布図パネルではデータが時系列に含まれる必要はありません。散布図パネルには、データの 2 つ以上の数値列を含むテーブル形式のデータセットが必要です。

これらのいずれかを X 軸に割り当てることができます。1 つ以上を一連の Y 軸値に割り当てることができ、その結果データは一連の点としてプロットされます。各系列では、オプションで、統計に基づく複数のベストフィットのいずれかを使用して回帰線を表示することもできます。

散布図の作成

次の手順では、散布図パネルを使用して散布図を作成する方法について説明します。この例では、次の表のように、年齢別の男の子と女の子の平均身長を示す数値 `Age`、`Boys`、`Girls` の 3 つの列を持つ `HEIGHT` で呼び出されるデータがあると想定します。

年齢	男の子の身長	女の子の身長
5	109.7	109.5
6	115.6	115.4
7	121.1	120.8
8	126.3	126

年齢	男の子の身長	女の子の身長
9	131.3	131.3
10	136.2	137.1
11	141.2	143.2
12	147	148.7
13	153.6	152.6
14	159.9	155.1
15	164.4	156.7
16	167.3	157.6
17	169	158
18	170	158.3
19	170.8	158.6

散布図パネルを使用して散布図を作成するには

1. Grafana ダッシュボードで、[パネルの追加] を選択します。
2. [クエリ] には、必要なデータを返すクエリを書き込みます。この場合、`SELECT * FROM HEIGHT` などのクエリを使用します。
3. [散布図] での視覚化を選択します。

これにより、最初の列を X 軸として使用し、他の数値列を Y 軸として使用して散布図が作成されます。

設定オプション

散布図パネルには、次の 4 つのカスタム設定オプションが用意されています。

- [X 軸] – X 軸として使用するフィールド、軸の範囲とタイトル、および表示情報を選択できます。

- [Y 軸] – 各フィールドの表示オプション、軸の範囲とタイトル情報など、Y 軸に表示するフィールドを選択できます。また、各フィールドの回帰線を表示することもできます。回帰線設定の詳細については、以下の情報を参照してください。
- [凡例] – パネルの凡例をオンまたはオフにしたり、凡例内のテキストサイズを選択したりできます。
- [表示] – グリッドの色や境界スタイルなど、グラフの他のオプションを設定できます。

回帰線の設定

各 Y 軸データセットには、個々の点に加えて線を表示できます。線タイプには 5 つのオプションがあります。

- [なし] – 回帰線を表示しません。
- [単純] – データセットポイントを接続する回帰線を表示します。
- [線形] – 最小二乗法、ベストフィット方式を使用して直線を表示します。
- [指数] – 指数のベストフィット回帰線を表示します。
- [累乗] – 累乗のベストフィット回帰線を表示します。

統計パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

統計パネルの視覚化は、オプションのグラフスパークラインを含む 1 つの大きな統計値を表示します。しきい値を使用して背景または値の色を制御できます。

デフォルトでは、統計パネルには次のいずれかが表示されます。

- 単一の系列またはフィールドの値のみ。
- 複数の系列またはフィールドの値と名前の両方。

テキストモードを使用して、テキストが表示されるかどうかを制御できます。

自動レイアウト調整

パネルは、ダッシュボードで使用可能な幅と高さに応じてレイアウトを自動的に調整します。パネルが小さすぎると、グラフ (スパークライン) が自動的に非表示になります。

値オプション

次のオプションを使用して、視覚化で値を表示する方法を絞り込みます。

[表示]

Grafana がデータを表示する方法を選択します。

[計算]

すべての行を元に計算した値を表示します。

- [計算] – Grafana が多くのフィールドを 1 つの値に減らすために使用する reducer 関数を選択します。使用可能な計算のリストについては、「標準計算」を参照してください。
- [フィールド] – パネルに表示されるフィールドを選択します。

[すべての値]

行ごとに個別の統計を表示します。このオプションを選択すると、表示する行数を制限することもできます。

- [制限] – 表示する行の最大数。デフォルトは 5,000 です。
- [フィールド] – パネルに表示されるフィールドを選択します。

統計スタイル

視覚化のスタイルを変更します。

[Orientation] (向き)

積み重ね方向を選択します。

- [自動] – Grafana が最適な方向を選択します。

- [水平] – バーは水平方向に左から右に伸びます。
- [垂直] – バーは垂直に上から下に伸びます。

テキストモード

テキストモードオプションを使用して、パネルがレンダリングするテキストを制御できます。値が重要ではなく、名前と色のみが重要である場合は、[テキストモード] を [名前] に変更します。値は色を決定するために引き続き使用され、ツールチップに表示されます。

- [自動] – データに複数の系列またはフィールドが含まれている場合は、名前と値の両方を表示します。
- [値] – 値のみを表示し、名前は表示しません。代わりに、カーソルのツールチップに名前が表示されます。
- [値と名前] – 常に値と名前を表示します。
- [名前] – 値の代わりに名前を表示します。値はカーソルのツールチップに表示されます。
- [なし] – 何も表示しません (空)。名前と値は、カーソルのツールチップに表示されます。

カラーモード

カラーモードを選択します。

- [値] – 値とグラフ領域のみを色付けします。
- [Background] (背景) – 背景も色付けします。

グラフモード

グラフとスパークラインモードを選択します。

- [なし] – グラフを非表示にし、値のみを表示します。
- [エリア] – 値の下に面グラフを表示します。これには、クエリが時間列を返す必要があります。

テキストの整列

アライメントモードを選択します。

- [自動] – 値が 1 つだけ表示されている場合 (繰り返しなし)、値は中央に配置されます。複数の系列または行が表示されている場合、値は左揃えになります。

- [Center] (中央) – 統計値は中央にあります。

テキストサイズ

ゲージテキストのサイズを調整します。

- [タイトル] – ゲージタイトルサイズの数値を入力します。
- [値] – ゲージ値サイズの数値を入力します。

状態タイムラインパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

状態タイムラインパネルの視覚化では、時間の経過に伴う個別の状態の変化が表示されます。各フィールドまたは系列は、一意の水平バンドとしてレンダリングされます。状態領域は、値の有無にかかわらずレンダリングできます。このパネルは、文字列またはブール値の状態とうまく連携しますが、時系列でも使用できます。時系列で使用すると、しきい値を使用して数値を離散状態の領域に変換します。

状態タイムラインオプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[Merge equal consecutive values] (等しい連続値をマージする)

Grafana は、互いに隣接する同じ値をマージするかどうかを制御します。

[値の表示]

値が状態領域内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[値の整列]

状態領域内の値のアライメントを制御します。

[Row height] (行の高さ)

行間のスペースを制御します。1 = スペースなし = 0.5 = 50% のスペース。

[線の幅]

状態領域の線幅を制御します。

[塗りつぶしの不透明度]

状態領域の不透明度を制御します。

値のマッピング

ブール値または文字列値に色を割り当てるには、値マッピングを設定する を使用します。

しきい値を含む時系列データ

パネルは時系列データでも使用できます。この場合、しきい値を使用して時系列を個別の色付き状態領域に変換します。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

ステータス履歴パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ステータス履歴の視覚化は、時間の経過に伴う周期的な状態を表示します。各フィールドまたは系列は、横の列としてレンダリングされます。ボックスがレンダリングされ、各値を中央に配置します。

ステータス履歴の視覚化は、文字列、ブール値、数値フィールドまたは時系列で機能します。時間フィールドは必須です。値マッピングを使用して、文字列の色付け、またはテキスト値の数値範囲への割り当てができます。

表示オプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[値の表示]

値が値ボックス内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[列幅] は、ボックスの幅を制御します。1=最大、0=最小幅。

[線の幅] は、状態領域の線幅を制御します。

[塗りつぶしの不透明度] は、状態領域の塗りつぶしの不透明度を制御します。

値のマッピング

ブール値または文字列値に色を割り当てるには、値マッピングを設定する を使用します。

しきい値を含む時系列データ

パネルは時系列データでも使用できます。この場合、しきい値を使用してボックスを色付けします。グラデーションのカラースキームを使用して値を色付けすることもできます。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例値]

凡例に表示する標準計算を選択します。複数選択できます。

テーブルパネル

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

テーブルパネルの視覚化は非常に柔軟で、時系列とテーブル、注釈、未加工の JSON データにおける複数のモードをサポートしています。このパネルには、日付の書式設定、値の書式設定、色付けのためのオプションも用意されています。

列の並べ替え

列タイトルをクリックして、ソート順序をデフォルトから降順や昇順に変更します。クリックするたびに、ソート順序はサイクルの次のオプションに変わります。一度に 1 つの列のみでソートすることができます。

テーブルオプション

ヘッダーの表示

データソースからインポートされた列名を表示または非表示にします。

列幅

デフォルトでは、Grafana はテーブルサイズと最小列幅に基づいて列幅を自動的に計算します。このフィールドオプションは、設定をオーバーライドし、すべての列の幅をピクセル単位で定義できます。

例えば、`100` をフィールドに入力すると、フィールドの外部をクリックしたときにすべての列の幅が 100 ピクセルに設定されます。

[最小列幅]

デフォルトでは、テーブル列の最小幅は 150 ピクセルです。このフィールドオプションでは、デフォルトを上書きでき、テーブルパネルの新しい最小列幅をピクセル単位で定義します。

例えば、`75` をフィールドに入力すると、フィールドの外部をクリックしたときにすべての列が幅 75 ピクセル以上でスケールされます。

スマートフォンやタブレットなどのスモールスクリーンデバイスの場合、デフォルトの `150` ピクセル値を `50` に減らして、テーブルベースのパネルをダッシュボードで正しくレンダリングできます。

[列の配置]

Grafana がセルの内容を配置する方法を選択します。

- Auto (デフォルト)
- Left (左)

- Center (中央)
- Right (右)

セルタイプ

デフォルトでは、Grafana は表示設定を自動的に選択します。以下のオプションのいずれかを選択して、すべてのフィールドのデフォルトを設定することで、設定を上書きできます。一部のセルタイプでは、追加の設定を使用できます。

> **Note**
>
> フィールドタブでこれらを設定すると、時間フィールドを含むすべてのフィールドにタイプが適用されます。オーバーライドタブで設定して、変更を 1 つ以上のフィールドに適用できます。

[文字色]

しきい値が設定されている場合、フィールドテキストは適切なしきい値の色で表示されます。

Color background (gradient or solid) (背景色 (グラデーションまたは無地))

しきい値が設定されている場合、フィールドの背景は適切なしきい値の色で表示されます。

ゲージ

セルは、複数の異なるプレゼンテーションタイプでグラフィカルゲージとして表示できます。

ベーシック

基本モードでは、ゲージの色を定義するしきい値レベルを含むシンプルなゲージが表示されます。

Gradient (グラデーション)

しきい値レベルによりグラデーションが決定されます。

LCD

ゲージは小さなセルに分割され、点灯または消灯で表示されます。

JSON ビュー

コードとしてフォーマットされた値を表示します。値がオブジェクトの場合、カーソルを合わせると JSON オブジェクトのブラウジングができる JSON ビューが表示されます。

セル値の検査

テーブルセルから値の検査を有効にします。raw 値はモーダルウィンドウに表示されます。

> **Note**
>
> セル値の検査は、セル表示モードが [Auto]、[文字色]、[Color background]、または [JSON ビュー] に設定されている場合にのみ使用できます。

[Column filter] (列フィルター)

列データの表示方法を一時的に変更できます。例えば、最大から最小までの値を順序付けたり、特定の値を非表示にしたりできます。詳細については、以下の「フィルターテーブル列」を参照してください。

Pagination (ページ分割)

ページ分割を有効または無効にするには、このオプションを使用します。これは、クエリに影響を与えないフロントエンドオプションです。有効にすると、ページサイズはテーブルの高さに自動的に調整されます。

テーブル列のフィルタリング

列フィルターをオンにすると、テーブルオプションをフィルタリングできます。

列のフィルタリングを有効にするには

1. Grafana で、フィルタリングする列を含むテーブルを使用してダッシュボードに移動します。
2. フィルタリングするテーブルパネルで、パネルエディタを開きます。
3. [フィールド] タブを選択します。
4. [テーブル] オプションで、[列フィルター] オプションをオンにします。

各列のタイトルの横にフィルターアイコンが表示されます。

列の値のフィルタリング

列の値をフィルタリングするには、列タイトルの横にあるフィルター (ファネル) アイコンを選択します。Grafana にその列のフィルターオプションが表示されます。

表示する値の横にあるチェックボックスを選択します。上部の検索フィールドにテキストを入力してそれらの値をディスプレイに表示すると、スクロールして探さずに選択できます。

列フィルターのクリア

フィルターが適用された列では、タイトルの横に青い漏斗が表示されます。

フィルターを削除するには、青いファネルアイコンを選択し、[Clear filter] (フィルターのクリア) を選択します。

テーブルフッター

テーブルフッターを使用して、フィールドの計算を表示できます。

テーブルフッターを有効にしたら、[計算] を選択し、計算するフィールドを選択します。

フィールドを選択しない場合、システムはすべての数値フィールドに計算を適用します。

行のカウント

選択したフィールドの値の数ではなく、データセット内の行数を表示する場合は、[カウント] 計算を選択し、[カウント行] を有効にします。

テキストパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

テキストパネルを使用すると、ダッシュボードにテキストまたは HTML を直接含めることができます。これは、コンテキスト情報と説明の追加や、複雑な HTML を埋め込むために使用できます。

モード

モードは、埋め込みコンテンツの表示方法を決定します。これには次のオプションがあります。

- [Markdown] – このオプションはコンテンツをマークダウンとしてフォーマットします。
- [HTML] – この設定では、コンテンツをサニタイズされた HTML としてレンダリングします。
- [Code] – この設定では、読み取り専用コードエディタ内でコンテンツをレンダリングします。適切な言語を選択して、埋め込みテキストに構文強調表示を適用します。

[変数]

コンテンツの変数が展開されて表示されます。

時系列パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

時系列パネルでは、時系列を線、ドットの軌跡、または一連の棒としてレンダリングできます。このタイプのグラフは、ほぼすべての時系列データを表示するための十分な汎用性があります。

> **Note**
> グラフパネルの視覚化は時系列の視覚化に移行できます。移行するには、[パネル]タブで [時系列の視覚化] を選択します。Grafana は該当するすべての設定を転送します。

トピック

- ツールチップオプション
- 凡例オプション
- グラフスタイルオプション
- 軸オプション
- 色オプション

ツールチップオプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

グラフにカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、カーソルを合わせている系列のみが表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – ツールチップを表示しません。

凡例オプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

[凡例モード] – 凡例の表示方法を選択します。

- [リスト] – 凡例を一覧として表示します。これがデフォルトです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] – 凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例計算]

凡例に表示する計算を選択します。詳細については、「計算タイプ」を参照してください。

グラフスタイルオプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

[Graph style] (グラフスタイル)

このオプションを使用して、時系列データを表示する方法を定義します。オーバーライドを使用して、同じグラフに複数のスタイルを組み合わせることができます。3 つのスタイルオプションがあります。その他のスタイルオプションの一部は、特定のグラフスタイルにのみ適用されます。

- [折れ線] – 時系列をグラフに折れ線で表示します。
- [棒] – 時系列をグラフに一連の棒で表示し、データポイントごとに 1 つずつ表示します。
- [ポイント] – 時系列をグラフに点で表示し、データポイントごとに 1 つずつ表示します。

[棒の配置]

棒グラフの場合は、グラフ上のデータポイントが描画される場所を基準にして、棒の位置を設定します。棒には幅があるため、棒をデータポイントの前、後、または中央に配置することができます。[前]、[中央]、または[後] をこのオプションで選択します。

[線の幅]

折れ線グラフの線の太さ、または棒グラフで各棒の枠線の太さを設定します。

[塗りつぶしの不透明度]

塗りつぶし色の不透明度を設定します。塗りつぶしは、折れ線グラフの線の下にある領域を表示したり、棒グラフの棒の色として表示したりするために使用します。

[グラデーションモード]

グラデーションモードでは、系列の色に基づいてグラデーションの塗りつぶしを指定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。詳細については、「カラースキーム」を参照してください。グラデーションモードのオプションは次のとおりです。

- [なし] – グラデーションの塗りつぶしなし。
- [不透明度] – Y 軸の値が増加するにつれて、塗りつぶしの不透明度が増加する不透明度グラデーション。
- [色相] – 系列の色相に基づくグラデーション。
- [Scheme] (スキーム) – カラースキームによって定義されるカラーグラデーション。この設定は、塗りつぶしと線で使用できます。詳細については、「色オプション」を参照してください。

グラデーションの外観は、[塗りつぶしの不透明度]設定でも変更されます。

[ポイント表示]

視覚化を設定して、折れ線グラフまたは棒グラフにポイントを追加できます。[常に表示]、[常に非表示]、または [自動] を選択できます。自動を使用する場合、Grafana はデータの密度に基づいてポイントを表示するかどうかを決定します。データの密度が十分に低い場合、ポイントが表示されます。

[ポイントサイズ]

描画されたポイントのサイズを、直径 1 ~ 40 ピクセルに設定します。

[線形補間]

Grafana が系列の線形を補間する方法を選択します。選択肢は、[直線]、[平滑線]、[前のステップ]、[後のステップ] です。

[線のスタイル]

線のスタイルを設定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。

線のスタイルの外観は、[線幅] と [塗りつぶしの不透明度] の設定に影響されます。

線のスタイルの選択肢は、[実線]、[破線]、および [点線] です。

[null 値の接続]

null 値 (データ内のギャップ) をグラフに表示する方法を選択します。null 値は、連続線を形成するために接続することも、オプションで、超えるとデータ内のギャップが接続されなくなるしきい値を設定することもできます。[常に非表示] で常にデータポイントをギャップで接続しない、[常に表示] で常にデータポイントをギャップで接続する、またはデータ内のギャップを接続しなくなる [しきい値] を設定することができます。

[Stack series (系列の積み上げ)]

積み上げにより、Grafana は系列の上に別の系列を表示できます。視覚化で積み上げを使用すると、誤解を招くグラフを簡単に作成できるため、注意が必要です。積み上げが最適な手段ではない理由の詳細については、「積み上げに関する問題」を参照してください。

積み上げのオプションは次のとおりです。

- [Off] (オフ) – 系列の積み上げをオフにします。
- [普通] – 系列の上に別の系列を積み上げます。
- [100%] – パーセンテージで積み上げるため、すべての系列の合計が 100% になります。

グループ内の系列の積み上げ

積み上げ動作をオーバーライドして、グループ内で系列を積み上げることができます。オーバーライドの作成の詳細については、「フィールドのオーバーライドを設定する」を参照してください。オーバーライドを作成するときは、系列に含める積み上げグループの名前を指定します。

[下を塗りつぶす]

[下を塗りつぶす]オプションは、2 つの系列間の領域を塗りつぶします。このオプションは、系列またはフィールドのオーバーライドとしてのみ使用できます。このオプションを使用すると、系列の折れ線から 0 までではなく、2 つの系列間の領域を塗りつぶすことができます。例えば、[最大値]と[最小値]という 2 つの系列がある場合、[最大値]系列を選択し、[最小値]系列まで[下を塗りつぶす]ようオーバーライドできます。これにより、2 つの折れ線間の領域のみが塗りつぶされます。

軸オプション

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

軸カテゴリのオプションにより、X 軸と Y 軸のレンダリング方法が変わります。一部のオプションは、編集するフィールドオプションボックスの外部をクリックするまで有効になりません。または、Enter キーを押す方法もあります。

[配置]

Y 軸の配置を選択します。オプションは以下のとおりです。

- [自動] – Grafana は、Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。
- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。

各フィールドまたは系列に軸を割り当てるには、[フィールドオーバーライドの追加] をします。

ラベル

Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。

[幅]

軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。この設定により、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。

[ソフト最小値とソフト最大値]

Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて Y 軸の範囲を自動的に設定します。

ソフト最小設定とソフト最大設定により、大きな変更がない場合に小さな変更を可視化できます。標準最小フィールドオプションと最大フィールドオプションから導出されるハード最小値または最大値は、特定のポイントを超えてスパイクを切り取ることで、断続的なスパイクが有用な詳細を平坦化することを防げます。

Y 軸のハード制限を定義するために、標準の最小/最大オプションを設定できます。詳細については、「標準オプションの設定」を参照してください。

[Scale] (スケール)

Y 軸のスケール方法を設定します。選択肢は、[線形] と [対数] です。対数を選択した場合は、さらに底 2 または底 10 の対数スケールを選択できます。

[Transform] (変換)

系列をオーバーライドして、グラフ上の値に変換を適用できます (基になる値や、ツールチップ、コンテキストメニュー、凡例の値には影響しません)。変換オプションは 2 つあります。

- [負の Y 変換] – 結果を Y 軸の負の値に反転します。
- [Constant] (定数) – 最初の値を定数線として表示します。

色オプション

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

デフォルトでは、グラフは標準のカラースキームオプションを使用して系列の色を割り当てます。凡例にある系列の色アイコンをクリックすることで、凡例を使用してカラーピッカーを開くこともできます。この方法で色を設定すると、特定の系列に特定の色を設定するオーバーライドルールが自動的に作成されます。

以下は、系列の色のデフォルトを変更するために使用できる追加オプションです。

Classic palette (クラシックパレット)

最も一般的なセットアップは、グラフにクラシックパレットを使用することです。このスキームは、その順序に基づいて、各フィールドまたは系列に色を自動的に割り当てます。クエリでフィールドの順序が変更されると、色も変わります。オーバーライドルールを使用して、特定のフィールドの色を手動で設定できます。

[単一色]

このモードを使用して色を指定します。凡例の各系列の横にある色付き線アイコンをクリックして、カラーピッカーを開くこともできます。これにより、カラースキームを単一色と選択した色に設定する新しいオーバーライドが自動的に作成されます。

値別カラースキーム

[しきい値起点: 値別] や [緑黄赤: 値別] などの値別カラースキームを選択すると、[系列に使用する色] オプションが表示されます。このオプションは、系列に色を割り当てるためにどの値 (最終値、最小値、最大値) を使用するかを管理します。

スキームグラデーションモード

グラフスタイルの下にあるグラデーションモードオプションには、Scheme という名前のモードがあります。スキームを有効にすると、折れ線または棒グラフは、選択したカラースキームから定義されたグラデーション色を受け取ります。

[しきい値起点]

カラースキームがしきい値起点 (値別) に設定され、グラデーションモードがスキームに設定されている場合、折れ線または棒グラフの色は定義されたしきい値を超えると変化します。スキームで選択された正確な色のみが表示されます。

グラデーションカラースキーム

グラデーションモードをスキームに設定せずにグラデーションカラースキームを使用すると、系列内の値がしきい値セット間を移動するにつれて、選択した色間でグラデーションを形成します。

トレースパネル (ベータ)

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

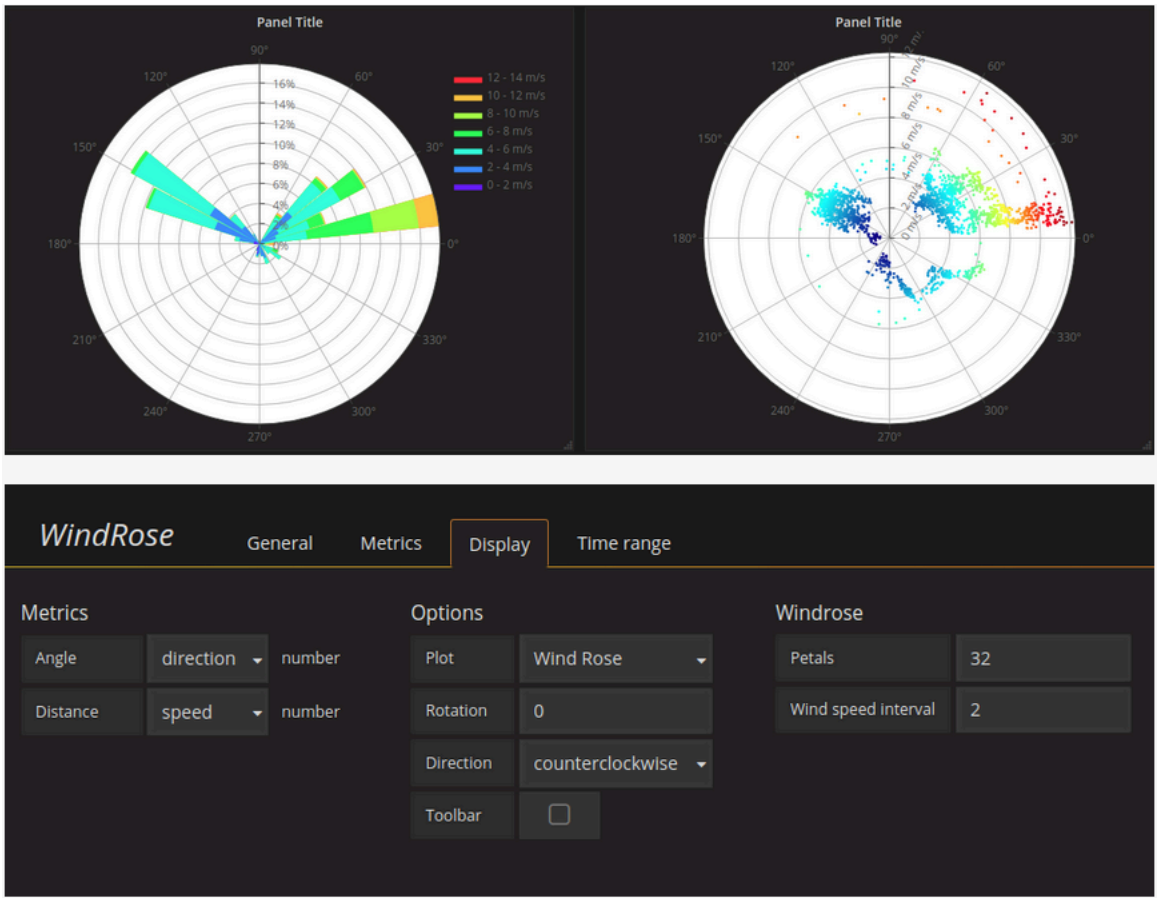
トレースは、インフラストラクチャ内のサービスを横断するリクエストを追跡および口グ記録できる視覚化です。

トレースの詳細については、「Explore でのトレース」を参照してください。

WindRose

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

WindRose パネルは、生の時系列データを受信し、データを変換して WindRose チャートにマッピングします。



オプション

WindRose パネルは、以下のオプションをサポートしています。

- 軸の頻度
- 軸のスタイル (角度またはコンパス)
- 目盛 (線形、正方形、対数)

Grafana バージョン 9 で探索する

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana のダッシュボード UI は、視覚化用のダッシュボードを構築する機能を提供します。Explore は、クエリに集中できるように、ダッシュボードとパネルのオプションを取り除きます。これにより、クエリが動作するまで反復処理を行い、クエリからダッシュボードを構築できます。

> **Note**
>
> ダッシュボードを作成せず単にデータを探索したい場合は、Explore を使用するとはるかに簡単になります。データソースがグラフとテーブルのデータをサポートしている場合、Explore は結果をグラフとテーブルの両方で表示します。これにより、データの傾向と詳細を同時に確認できます。

探索を開始する

> **Note**
>
> Explore にアクセスするには、エディタまたは管理者ロールが必要です。

Explore にアクセスするには

1. Grafana ワークスペースで、左側のメニューバーから Explore メニュー項目を選択します。

   空の Explore タブが開きます。

   または、パネル内の既存のクエリから開始するには、パネルメニューから Explore オプションを選択します。これにより、パネルからのクエリを含む Explore タブが開き、ダッシュボードの外部でクエリを微調整または反復できます。
2. 左上のドロップダウンからデータソースを選択します。Prometheus には、Explore のカスタム実装があり、他のデータソースは標準のクエリエディタを使用します。
3. クエリフィールドに、データを探索するクエリを書き込みます。クエリフィールドの横に、クリアボタン (X)、クエリ追加ボタン (+)、クエリ削除ボタン (-) の 3 つのボタンが表示されます。通常のクエリエディタと同様に、複数のクエリを追加または削除できます。

   クエリの詳細については、「データのクエリと変換」を参照してください。

分割して比較する

分割ビューを使用すると、グラフとテーブルを並べて比較したり、関連データを 1 ページでまとめて確認したりすることができます。

上部で分割ビューを開く

1. Explore ビューで、[分割] ボタンを選択して現在のクエリを複製し、ページを 2 つの並列クエリに分割します。

   > **Note**
   >
   > 新しいクエリには別のデータソースを選択できます。例えば、2 つの異なるサーバーに対して同じクエリを比較したり、ステージング環境を本番環境と比較したりできます。

   分割ビューでは、いずれかのタイムピッカーにある時刻同期ボタンを選択することで、両方のパネルのタイムピッカーを連携できます (一方を変更すると、他方も変更されます)。タイムピッカーを連携すると、分割ビュークエリの開始時刻と終了時刻が同期されます。これにより、両方の分割パネルで同じ時間間隔を確認できます。
2. 新しく作成したクエリを閉じるには、[分割を閉じる] ボタンをクリックします。

短縮リンクを共有する

短縮リンクの共有機能を使用すると、クエリパラメータで長い URL を使用する代わりに、/goto/:uid 形式のより短くてシンプルな URL を作成できます。クエリ結果への短縮リンクを作成するには、Explore ツールバーで [共有] オプションを選択します。使用されない短縮リンクは、7 日後に自動的に削除されます。

Explore でのクエリ管理

> **⚠** このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

クエリのデバッグを支援するために、Explore ではクエリリクエストとレスポンス、クエリ統計を「クエリインスペクター」を使用して調査できます。この機能は、パネルインスペクタータスク「クエリのパフォーマンス検査」、「クエリリクエストとレスポンスデータの検査」と類似しています。

クエリ履歴

クエリ履歴は、Explore で使用したクエリのリストです。履歴は Grafana データベースに保存され、他のユーザーと共有されません。履歴内のクエリの保持期間は 2 週間です。2 週間以上経過したクエリは自動的に削除されます。履歴を開いて操作するには、Explore の [クエリ履歴] ボタンを選択します。

> **Note**
>
> スター付き (お気に入り) クエリは 2 週間の保持期間の対象ではなく、削除されません。

クエリ履歴の表示

クエリ履歴を使用すると、クエリの履歴を表示できます。個々のクエリごとに、次のことができます。

- クエリを実行する
- コメントを作成および/または編集します。
- クエリをクリップボードにコピーする。
- クエリを含む短縮リンクをクリップボードにコピーする。
- クエリをスター付き (お気に入り) にする。

お気に入りのクエリを管理する

[クエリ履歴] タブにあるスター付きのすべてのクエリは、[スター付き] リストに表示されます。これにより、お気に入りのクエリにすばやくアクセスし、ゼロから入力することなくこれらのクエリを再利用できます。

[クエリ履歴の並べ替え]

デフォルトでは、クエリ履歴には最新のクエリが表示されます。履歴は日付またはデータソース名で、昇順または降順に並べ替えることができます。

クエリ履歴を並べ替えるには

1. [クエリの並べ替え順序] フィールドを選択します。
2. 以下のオプションのいずれかを選択します。
   - 新しい順
   - [古い順]

クエリ履歴のフィルタリング

クエリ履歴とスター付きタブで、クエリ履歴を特定のデータソースに対してフィルタリングできます。

データソースに対する履歴のフィルタリング

1. [特定のデータソースのクエリをフィルタリング] フィールドを選択します。
2. 履歴をフィルタリングするデータソースを選択します。複数のデータソースを選択することもできます。

[クエリ履歴] タブでは、スライダーを使用して日付でクエリをフィルタリングすることもできます。

- 垂直スライダーを使用して、クエリを日付でフィルタリングします。
- 上部のハンドルをドラッグして開始日を設定します。
- 上部のハンドルをドラッグして終了日を設定します。

クエリ履歴の検索

クエリとコメントをまたいで履歴を検索できます。クエリ履歴タブとスター付きタブでクエリを検索できます。

クエリ履歴で検索するには

1. [クエリ検索] フィールドを選択します。
2. 検索する用語を検索フィールドに入力します。

クエリ履歴設定

[設定] タブでクエリ履歴をカスタマイズできます。オプションについては、以下の表を参照してください。

設定	デフォルト値
デフォルトのアクティブタブを変更する	[クエリ履歴] タブ

> **Note**
>
> クエリ履歴設定は全体に及び、分割モードで両方のパネルに適用されます。

Prometheus 固有の機能

Explore には、Prometheus のカスタムクエリエクスペリエンスが搭載されています。クエリを実行すると、実際にはグラフ用の通常の Prometheus クエリとテーブル用のインスタントクエリの 2 つのクエリが実行されます。インスタントクエリは、グラフに表示されるデータの適切な要約を示す各時系列の最後の値を返します。

メトリクスエクスプローラー

クエリフィールドの左側で [メトリクス] を選択すると、メトリクスエクスプローラーが開きます。エクスプローラーには、メトリクスがプレフィックスごとにグループ化された階層メニューが示されます。例えば、すべてのアラートマネージャーメトリクスは `alertmanager` プレフィックスの下にグループ化されます。これは、利用可能なメトリクスを単に確認したい場合に有効です。

クエリフィールド

クエリフィールドは、メトリクス名、関数の自動補完機能に対応しており、標準の Prometheus クエリエディタとほぼ同じように動作します。Enter キーを押して新しい行を作成し、Shift+Enter キーを押してクエリを実行できます。

Ctrl+Space キーを押すとオートコンプリートメニューが開きます。オートコンプリートメニューには新たに [履歴] セクションが追加されており、最近実行されたクエリの一覧が表示されます。

クエリフィールドの下に提案が表示される場合があり、そのフィールドでクエリを選択すると、提案された変更でクエリを更新します。

- カウンター (指数関数的に増加するメトリクス) では、レート関数が提案されます。

- バケットの場合、ヒストグラム関数が候補として表示されます。
- 記録ルールの場合、ルールを展開できます。

テーブルをフィルタリング

テーブルパネルのラベル列にあるフィルターボタンを選択すると、クエリ式にフィルターを追加することができます。複数のクエリに対してフィルターを追加することもでき、フィルターはすべてのクエリに追加されます。

Explore のログ

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Explore では、メトリクスとともに、次のデータソースでログを調査できます。

- OpenSearch
- InfluxDB
- Loki

インフラストラクチャのモニタリングとインシデント対応中に、メトリクスとログをより深く掘り下げて原因を見つけることができます。Explore では、メトリクスとログを並べて表示することで、メトリクスとログを関連付けることもできます。これにより、新しいデバッグワークフローが作成されます。

1. アラートを受信します。
2. メトリクスをドリルダウンして調べます。
3. もう一度ドリルダウンし、メトリクスと時間間隔 (今後は分散トレース) に関連するログを検索します。

ログの視覚的表現

ログクエリの結果はグラフにヒストグラムとして表示され、個々のログについては以下のセクションで説明します。

データソースがフルレンジのログボリュームヒストグラムをサポートしている場合、入力されたすべてのログクエリのログ分布を含むグラフが自動的に表示されます。この機能は、現在 OpenSearch および Loki データソースでサポートされています。

> **Note**
>
> Loki では、このフルレンジのログボリュームヒストグラムをメトリクスクエリによってレンダリングします。このクエリは、クエリされた時間範囲に応じてコストがかかる可能性があります。このクエリは、小規模な Loki インストールで処理するのが特に難しい場合があります。これを軽減するには、Loki の前に nginx などのプロキシを使用して、これらのクエリにおけるユーザー定義のタイムアウト (10 秒など) を設定することをお勧めします。ログボリュームヒストグラムクエリは、値 `Source=logvolhist` の HTTP ヘッダー `X-Query-Tags` を持つクエリを検索することで識別できます。これらのヘッダーは、Grafana によってすべてのログボリュームヒストグラムクエリに追加されます。

データソースがフルレンジのログボリュームヒストグラムのロードをサポートしていない場合、ログモデルは、自動的に計算された時間間隔でバケット化されたログ行数に基づいて時系列を計算し、最初のログ行のタイムスタンプは結果からヒストグラムの開始を固定します。時系列の終了点は、タイムピッカーの To 範囲に固定されます。

ログレベル

レベルラベルが指定されているログの場合、Grafana はラベルの値を使用してログレベルを判別し、それに応じて色を更新します。ログにレベルラベルが指定されていない場合、そのコンテンツがサポートされている表現のいずれかと一致するかどうかを調べようとします (詳細については、以下を参照してください)。ログレベルは常に最初の一致によって決まります。Grafana がログレベルを判別できない場合、不明なログレベルとして視覚化されます。

> **Tip**
>
> Loki データソースを使用し、`level` がログラインにある場合は、パーサー (JSON、logfmt、regex など) を使用して、レベル情報をログレベルの決定に使用されるレベ

ルラベルに抽出できます。これにより、ヒストグラムはさまざまなログレベルを個別のバーに表示できます。

サポートされているログレベルと、ログレベルの略記と表現のマッピング

サポートされている式	ログレベル	色
emerg	critical	紫色
fatal	critical	紫色
alert	critical	紫色
crit	critical	紫色
critical	critical	紫色
err	エラー	赤
eror	エラー	赤
エラー	エラー	赤
warn	警告	黄色
警告	警告	黄色
情報	情報	緑色
information	情報	緑色
notice	情報	緑色
dbug	デバッグ	青
デバッグ	デバッグ	青
トレース	トレース	水色
*	不明	灰色

ログナビゲーション

ログ行の横にあるログナビゲーションインターフェイスを使用して、より多くのログをリクエストできます。これを行うには、ナビゲーションの下部にある [古いログ] ボタンを選択します。行制限に達し、さらにログを表示する場合は、これを使用してさらに多くのログを取得できます。各リクエストは、ナビゲーションに個別のページとして表示されます。各ページには、受信ログ行の開始と終了のタイムスタンプが表示されます。表示するページをクリックすると、以前の結果を表示できます。Explore がログナビゲーションから実行された過去 5 つのリクエストをキャッシュすることで、ページをクリックしたときに同じクエリを再実行しません。

視覚的表現オプション

ログの表示方法をカスタマイズしたり、表示する列を選択することができます。

[時間]

時間列を表示または非表示にします。これはデータソースから報告されたログ行と関連するタイムスタンプです。

[一意のラベル]

非共通ラベルのみを含む一意のラベル列を表示または非表示にします。すべての一般的なラベルが上に表示されます。

[行の折り返し]

ディスプレイで行の折り返しを使用する場合は、これを真に設定します。False に設定すると、水平スクロールになります。

Prettify JSON (JSONの整形)

すべての JSON ログを pretty print (整形表示) するには、これを `true` に設定します。この設定は、JSON 以外の形式のログには影響しません。

Deduplication (重複排除)

ログデータは非常に反復的であり、Explore は重複するログ行を非表示にすることで役立ちます。使用できる重複排除アルゴリズムはいくつかあります。

- [完全] - 完全一致は、日付フィールドを除き、行全体で行われます。

- [数値] - 数値の一致は、期間、IP アドレスなどの数値を除外した後に行で行われます。
- [署名] - 最も積極的な重複排除で、残りの空白と句読点のすべての文字、数字、一致を削除します。

[結果の順序の反転]

受信したログの順序は、デフォルトの降順 (新しい順) から昇順 (古い順) に変更できます。

ラベルと検出されたフィールド

各ログ行には、より堅牢なインタラクションのために、ラベルと検出されたフィールドを含む拡張可能な領域があります。すべてのラベルについて、選択したラベルをフィルタリングして抽出 (ポジティブフィルター)、除外 (ネガティブフィルター) する機能が追加されました。また、各フィールドやラベルには、表示されているすべてのログに関連する統計を表示する統計アイコンもあります。

改行のエスケープ

Explore は、改行 (\n、\r) やタブ (\t) など、ログ行で誤ってエスケープされたシーケンスを自動的に検出します。このようなシーケンスを検出すると、Explore は [改行のエスケープ] オプションを提供します。

Explore は、検出した誤ってエスケープされたシーケンスを自動的に修正できます

エスケープシーケンスを自動的に修正するには

1. [改行のエスケープ] を選択してシーケンスを置き換えます。
2. 置き換えを手動で確認し、その正確性を確認します。

Explore はこれらのシーケンスを置き換えます。そうすると、オプションは [改行のエスケープ] から [エスケープの削除] に変更されます。受信した入力に基づいて解析が正確ではない可能性があるため、変更を評価します。[エスケープの削除] を選択すると、置き換えを元に戻すことができます。

データリンク

データリンクを使用すると、ログメッセージの任意の部分を内部リンクまたは外部リンクに変換できます。作成されたリンクは、[ログの詳細] ビュー内の [リンク] セクションにボタンとして表示されます。

フィールドの可視性を切り替える

ログ行を展開し、目のアイコンをクリックするとフィールドを表示または非表示にできます。

Loki 固有の機能

Loki は Grafana Labs のオープンソースのログ集約システムです。Loki は、ログの内容にインデックスを作成するのではなく、ログストリームごとにラベルのセットを作成するため、費用対効果が高いように設計されています。Loki からのログは、Prometheus のラベルセレクタを使用したクエリと同様の方法でクエリを実行されます。ラベルを使用して、Prometheus ラベルと一致するよう作成されるログストリームをグループ化します。Grafana Loki の詳細については、「Grafana Loki」Github を参照してください。

詳細については、ログデータのクエリ方法に関する [Loki] を参照してください。

メトリクスからログに切り替える

Prometheus クエリからログクエリに切り替えると (まず分割してメトリクスとログを並べることができます)、ログに存在するクエリのラベルが保持され、それらを使用してログストリームに対してクエリを実行します。例えば、次の Prometheus クエリがある場合です。

```
grafana_alerting_active_alerts{job="grafana"}
```

ログデータソースに切り替えると、次のようになります。

```
{job="grafana"}
```

これにより、検索できる選択した時間範囲内のログのチャンクが返されます。

ログサンプル

選択したデータソースがログサンプルを実装し、ログクエリとメトリクスクエリの両方をサポートしている場合、メトリクスクエリの場合は、視覚化されたメトリクスに寄与したログ行のサンプルを自動的に表示できます。この機能は、現在 Loki データソースでサポートされています。

[ライブテーリング]

ライブテーリング機能を使用して、サポートされているデータソースに関するリアルタイムログを表示します。

Explore ツールバーの [Live] ボタンを選択して、Live Tail ビューに切り替えます。

Live Tail ビューでは、新しいログが画面の下部に表示され、背景が淡いコントラストになるため、最新情報を簡単に追跡できます。[一時停止] ボタンを選択するか、ログビューをスクロールしてライブテーリングを一時停止し、中断することなく以前のログを調べます。[再開] ボタンを選択してライブテーリングを再開するか、[停止] ボタンを選択してライブテーリングを終了し、標準の Explore ビューに戻ります。

Explore でのトレース

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Explore を使用すると、トレース中のデータソースからトレースを視覚化できます。

以下のデータソースがサポートされています。

- Jaeger
- Tempo
- AWS X-Ray
- Zipkin

上記のデータソースのクエリを設定する方法については、特定のデータソースのドキュメントを参照してください。

トレースビューの説明

このセクションでは、Trace View ダッシュボードの要素について説明します。

ヘッダー

トレースビューのヘッダーには、次の要素があります。

- ヘッダータイトル: ルートスパンの名前とトレース ID を表示します。

- 検索: 検索されたテキストを含むスパンを強調表示します。
- メタデータ: トレースに関するさまざまなメタデータを表示します。

ミニマップ

要約ビューまたはトレースタイムラインを表示します。ポインタをミニマップにドラッグして、より短い時間範囲にズームインできます。ズームはメインタイムラインも更新するため、短いスパンを簡単に確認できます。ミニマップにカーソルを合わせ、ズームすると、ズームをリセットする選択のリセットボタンが表示されます。

タイムライン

トレース内のスパンのリストを表示します。各スパン行は、次の要素で構成されます。

- [子を展開] ボタン: 選択したスパンのすべての子スパンを展開または折りたたみます。
- サービス名: スパンをログに記録したサービスの名前。
- オペレーション名: このスパンが表すオペレーションの名前。
- スパン期間バー: トレース内のオペレーション期間の視覚的表現。

スパンの詳細

スパン行の任意の場所をクリックすると、以下を含むスパンの詳細が表示されます。

- オペレーション名
- スパンメタデータ
- タグ: このスパンに関連付けられている任意のタグ。
- プロセスメタデータ: このスパンをログに記録したプロセスに関するメタデータ。
- ログ: このスパンおよび関連するキー値によってログに記録されたログのリスト。Zipkin ログの場合、セクションには Zipkin 注釈が表示されます。

ノードグラフ

オプションで、表示されたトレースのノードグラフを展開できます。データソースによっては、トレースのスパンをグラフ内のノードとして表示したり、現在のトレースに基づくサービスグラフなどのコンテキストを追加したりできます。

ログへのトレース

トレースビューのスパンから、そのスパンに関連するログに直接移動できます。これは、Tempo、Jaeger、および Zipkin データソースで使用できます。各データソースの設定方法については、関連するドキュメントを参照してください。

ドキュメントアイコンをクリックして、設定されたデータソースを使用して Explore で分割ビューを開き、スパンに関連するログに対してクエリを実行します。

サービスグラフビュー

Service Graph ビューは、スパンメトリクス (レート、エラーレート、期間 (RED) のデータを追跡) とサービスグラフを視覚化します。要件が設定されると、この事前設定されたビューがすぐに利用できます。

詳細については、Tempo データソースのページを参照してください。サービスグラフビューページは、「Tempo ドキュメント」でも確認できます。

Explore の Inspector

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

インスペクターは、クエリの理解とトラブルシューティングに役立ちます。未加工データを検査し、そのデータをカンマ区切り値 (CSV) ファイルにエクスポートし、ログ結果を TXT 形式でエクスポートし、クエリリクエストを表示できます。

Inspector UI

インスペクターには次のタブがあります。

- [統計タブ] – クエリにかかる時間および返される量を示します。
- [クエリタブ] – Grafana がデータソースへのクエリ実行時に送信されるサーバーへのリクエストを表示します。
- [JSON] タブ – データ JSON とデータフレーム構造 JSON を表示およびコピーできます。

- [データタブ] – クエリによって返される未加工データを表示します。
- [エラータブ] – エラーを表示します。クエリがエラーを返す場合にのみ表示されます。

Inspector タスク

Explore Inspector では、さまざまなタスクを実行できます。

Inspector を開く

検査するクエリを実行したら、[Inspector] ボタンを選択します。

画面の下部にインスペクターペインが開きます。

未加工クエリ結果の検査

クエリによって返されるデータである未加工のクエリ結果をテーブルに表示できます。

[Inspector] タブで、[Data] タブをクリックします。

複数のクエリまたは複数のノードのクエリには、追加のオプションがあります。

- [データフレームを表示する]: 表示する結果セットデータを選択します。
- [時間で結合された系列]: すべてのクエリの未加工データを一度に表示し、列ごとに結果セットを 1 つ表示します。列見出しをクリックしてデータをソートできます。

未加工のクエリ結果を CSV としてダウンロード

未加工のクエリ結果を表示すると、結果の CSV ファイルを生成できます。表示される結果の CSV ファイルを取得するため、CSV ファイルを生成する前に、必要な結果が得られるように結果を絞り込んでください。

CSV ファイルを生成するには、[Inspector] タブで [CSV のダウンロード] を選択します。

Excel 用にフォーマットされた CSV ファイルをダウンロードするには、[データオプション] を展開して [Excel 用にダウンロード] トグルをオンにしてから、[CSV のダウンロード] オプションを選択します。

ログ結果を TXT としてダウンロード

[Inspector] タブで [ログのダウンロード] を選択すると、現在表示しているログの TXT ファイルを生成できます。

トレース結果のダウンロード

データソースタイプに基づいて、Grafana は、サポートされている形式 (Jaeger、Zipkin、または OTLP 形式) のいずれかでトレース結果の JSON ファイルを生成できます。

トレースをダウンロードするには、[Inspector] タブで[トレースのダウンロード] を選択します。

クエリパフォーマンスの検査

[統計] タブには、クエリにかかる時間、送信したクエリの数、返された行数を示す統計が表示されます。この情報は、特に数値が予想外に高い場合や低い場合のクエリのトラブルシューティングに役立ちます。

統計は読み取り専用です。

[JSON モデルの表示]

データだけでなく、データフレーム JSON モデルも探索してエクスポートできます。

JSON モデルを表示するには

1. インスペクターパネルで、[JSON] タブをクリックします。
2. [ソースの選択] ドロップダウンから、以下のいずれかのオプションを選択します。
   - [データ] – Explore に返されたデータを表す JSON オブジェクトを表示します。
   - [データフレーム構造] – 未加工の結果セットを表示します。
3. JSON の一部を展開または折りたたんで、個別のセクションを表示できます。[クリップボードにコピー] オプションを選択して、JSON 本文をコピーし、別のアプリケーションに貼り付けることもできます。

データソースへの未加工リクエストとレスポンスの表示

Explore と Inspector タブを操作すると、クエリで生成する未加工のリクエストとレスポンスデータを表示できます。Inspector で [クエリ] タブを選択し、[更新] を選択して未加工データを表示します。

Grafana はクエリをサーバーに送信し、結果を表示します。クエリの特定の部分をドリルダウン、そのすべてを展開または折りたたみ、またはデータをクリップボードにコピーして他のアプリケーションで使用できます。

Grafana バージョン 9 のアラート

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana アラートからは、システムの問題が発生した直後に、サービスの中断を最小限に抑えるための強力で実用的なアラートが提供されます。

Amazon Managed Grafana には最新のアラートシステム [Grafana アラート]が含まれており、アラート情報を一元的に管理、検索できるビューが用意されています。以下に、その機能の一部を紹介します。

- Grafana アラートを一元的に作成および管理できるビュー。
- Cortex と Loki 管理のアラートを作成および管理できる単一のインターフェイス。
- Prometheus、Amazon Managed Service for Prometheus、およびその他のアラートマネージャー互換データソースからのアラート情報の表示。

Amazon Managed Grafana ワークスペースを作成するときは、Grafana アラートを使用するか、従来のダッシュボードアラート を使用するかを選択できます。このセクションでは、Grafana アラートについて説明します。

Note

Classic アラートを有効にしてワークスペースを作成し、Grafana アラートに切り替える場合は、2 つのアラートシステム間で切り替えることができます。

Grafana アラートの制約事項

- Grafana アラートシステムは、利用可能なすべての Amazon Managed Service for Prometheus、Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからはルールを取得できない場合があります。

- Prometheus ではなく Grafana で定義されたアラートルールは、複数の通知をコンタクトポイントに送信します。ネイティブ Grafana アラートを使用している場合は、新しい Grafana アラート機能を有効にせず、従来のダッシュボードアラートのまま使用することをお勧めします。Prometheus データソースで定義されたアラートを表示する場合は、Grafana アラートを有効にすることをお勧めします。これにより、Prometheus アラートマネージャーで作成されたアラートの通知が 1 つだけ送信されます。

> **Note**
>
> この制限は、Grafana v10.4 以降をサポートする Amazon Managed Grafana ワークスペースの制限ではなくなりました。

トピック

- 概要
- アラートの詳細
- アラートの設定
- 従来のダッシュボードアラートを Grafana アラートに移行する
- アラートルールの管理
- アラート通知の管理

概要

> **Warning**
>
> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

以下では、Grafana アラートの動作の概要を説明し、連携して機能し、柔軟で強力なアラートエンジンの中核を形成する主要な概念をいくつか紹介します。

1. データソース

アラートで使用するデータに接続します。このデータは、アラートの時系列データであることが多く、モニタリングおよび分析するシステムの詳細を示します。詳細については、[データソース]を参照してください。

2. アラートルール

   アラートルールは、アラートインスタンスを起動するかどうかを決定する一連の評価基準であり、データソースからデータをプルするための 1 つ以上のクエリと式、アラートの必要性を表す条件、評価の頻度、およびオプションでアラートが発せられるために条件を満たす必要がある期間で構成されます。

   Grafana マネージドアラートは、多次元アラートをサポートします。つまり、各アラートルールは複数のアラートインスタンスを作成できます。これは、1 つの式で複数のシリーズを観察する場合に非常に強力です。

3. ラベル

   アラートルールとそのインスタンスを通知ポリシーとサイレンスに一致させます。また、アラートを重要度別にグループ化するためにも使用できます。

4. 通知ポリシー

   アラートが発生したときにチームに通知するために、アラートがルーティングされる場所、タイミング、および方法を設定します。各通知ポリシーは、どのアラートを担当するかを示すラベルマッチャーのセットを指定します。通知ポリシーには、1 つ以上の通知者で構成されるコンタクトポイントが割り当てられます。

5. コンタクトポイント

   アラート発生時の通知先への通知方法を定義します。アラートがチームに適切に通知されるように、多数の ChatOps ツールをサポートしています。

機能

すべてのアラートを単一のページに

Grafana が管理するアラートと、Prometheus 互換データソースに存在するアラートの両方が 1 つの Grafana アラートページに統合されます。

多次元アラート

アラートルールは、「多次元アラート」と呼ばれるアラートルールごとに複数の個別のアラートインスタンスを作成できるため、1 つのアラートルールだけでシステム全体の可視性を得るための機能と柔軟性が得られます。

ルーティングアラート

定義したラベルに基づいて、各アラートインスタンスを特定のコンタクトポイントにルーティングします。通知ポリシーは、アラートがどこで、いつ、どのようにコンタクトポイントにルーティングされるかを定義するルールのセットです。

アラートのサイレンシング

サイレンスを使用すると、1 つ以上のアラートルールからの永続通知の受信を停止できます。また、特定の基準に基づいてアラートを部分的に一時停止することもできます。サイレンスには、整理と可視性を向上させるための独自の専用セクションがあるため、メインアラートビューを乱すことなく一時停止したアラートルールをスキャンできます。

ミュートタイミング

ミュートタイミングでは、通知を新たに生成または送信しない時間間隔を指定できます。また、メンテナンス期間など、繰り返し発生する期間中にアラート通知を停止することができます。

アラートの詳細

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafanaアラートの実装を開始または拡張するかどうかにかかわらず、アラートの作成、管理、およびアクションの実行に役立つ主要な概念と利用可能な機能の詳細を確認し、問題を迅速に解決するチームの能力を向上させます。

まず、Grafana アラートが提供するさまざまなアラートルールタイプを見てみましょう。

アラートルールタイプ

Grafana マネージドルール

Grafana マネージドルールは、最も柔軟なアラートルールの種類です。これにより、サポートされている任意のデータソースのデータに対して動作するアラートを作成できます。複数のデータソースをサポートするだけでなく、式を追加してデータを変換し、アラート条件を設定することもできます。これは、単一のルール定義で複数のデータソースからのアラートを許可する唯一のルールタイプです。

Mimir および Loki ルール

Mimir または Loki アラートを作成するには、互換性のある Prometheus または Loki データソースが必要です。データソースをテストし、Ruler API がサポートされているかどうかを観察することで、データソースが Grafana 経由でルール作成をサポートしているかどうかを確認できます。

記録ルール

記録ルールは、互換性のある Prometheus または Loki データソースでのみ使用できます。記録ルールでは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列セットとして保存できます。これは、集計データに対してアラートを実行する場合や、計算コストの高い式を繰り返しクエリするダッシュボードがある場合に便利です。

主な概念と特徴

次の表には、Grafana アラートを最大限に活用できるように設計された主要な概念、機能、およびそれらの定義の一覧を示します。

主要な概念または機能	定義
アラートに使用するデータソース	メトリクス、ログ、トレースをクエリおよび視覚化するデータソースを選択します。
アラートのプロビジョニング	アラートリソースを管理し、ファイルプロビジョニングまたは Terraform を使用して Grafana システムにプロビジョニングします。
アラートマネージャー	アラートインスタンスのルーティングとグループ化を管理します。
アラートルール	アラートルールを発動させる評価基準のセット。アラートルールは、1 つ以上のクエリと式、条件、評価の頻度、その条件の持続時間で構

主要な概念または機能	定義
	成されます。アラートルールは、複数のアラートインスタンスを生成できます。
アラートインスタンス	アラートインスタンスは、アラートルールのインスタンスです。1 次元アラートルールには 1 つのアラートインスタンスがあります。多次元アラートルールには 1 つ以上のアラートインスタンスがあります。複数の結果に一致する単一のアラートルール、たとえば 10 台の仮想マシンに対する CPU の監視は、複数 (この場合は 10) のアラートインスタンスとしてカウントされます。この数は時間の経過とともに変動する場合があります。たとえば、システム内のすべての仮想マシンの CPU 使用率を監視するアラートルールは、仮想マシンを追加するとアラートインスタンスが増加します。アラートインスタンスのクォータの詳細については、クォータ到達エラー を参照してください。
アラートグループ	アラートマネージャーは、ルート通知ポリシーのラベルを使用して、デフォルトでアラートインスタンスをグループ化します。これにより、コンタクトポイントに送信される重複排除とアラートインスタンスのグループが制御されます。
コンタクトポイント	アラートルール発生時の通知先への通知方法を定義します。
メッセージテンプレート	コンタクトポイント使用する再利用可能なカスタムテンプレートを作成します。
通知ポリシー	アラートがどこで、いつ、どのようにグループ化され、コンタクトポイントにルーティングされるかを定義するルールのセット。

主要な概念または機能	定義
ラベルとラベル照合機能	ラベルはアラートルールを識別するための一意な識別子です。ラベルによってアラートルールが通知ポリシーとサイレンスに紐づけられ、どのポリシーで処理されるか、どのアラートルールをサイレンス対象にするかを決定します。
サイレンス	1 つ以上のアラートインスタンスからの通知を停止します。サイレンスとミュートタイミングの違いは、サイレンスは指定した時間帯の間だけ有効であるのに対し、ミュートタイミングはスケジュールに沿って繰り返される点です。アラートインスタンスをサイレンス対象とするにはラベル照合機能を使用します。
ミュートタイミング	通知を新たに生成または送信しない時間間隔を指定します。また、メンテナンス期間など、繰り返し発生する期間中にアラート通知を停止することができます。既存の通知ポリシーにリンクする必要があります。

データソース

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana アラートと互換性のあるデータソースは多数あります。各データソースはプラグインでサポートされています。以下に示す組み込みデータソースのいずれかを使用できます。

これらは、Amazon Managed Grafana と互換性があり、Amazon Managed Grafana でサポートされているデータソースです。

- アラートマネージャーデータソースに接続する
- Amazon CloudWatch データソースへの接続
- Amazon OpenSearch Service データソースへの接続
- AWS IoT SiteWise データソースに接続する
- an AWS IoT TwinMaker データソースに接続する
- Amazon Managed Service for Prometheus およびオープンソースの Prometheus データソースへの接続
- Amazon Timestream データソースへの接続
- Amazon Athena データソースに接続する
- Amazon Redshift データソースに接続する
- AWS X-Ray データソースに接続する
- Azure Monitor データソースへの接続
- Google Cloud Monitoring データソースへの接続
- Graphite データソースに接続する
- InfluxDB ソースデータベースに接続するには
- Loki データソースに接続するには
- Microsoft SQL Server データソースに接続する
- MySQL データソースに接続する
- OpenTSDB データソースに接続する
- PostgreSQL データソースに接続する
- Jaeger データソースへの接続
- Zipkin データソースへの接続
- Tempo データソースへの接続
- テスト用 TestData データソースの設定

アラートルールについて

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールは、アラートインスタンスを起動するかどうかを決定する一連の評価基準です。アラートルールは、1 つ以上のクエリと式、条件、評価の頻度、およびオプションとしてその条件の持続時間で構成されます。

評価するデータセットを「クエリと式」で選択し、「条件」にアラートがアラートを発行するための基準 (しきい値) を設定します。

間隔は、アラートルールが評価される頻度を指定します。「持続時間」を設定した場合、その条件下にある状態の継続時間を示します。また、アラートルールでは、データがない場合のアラート動作も設定できます。

トピック

- アラートルールタイプ
- アラートインスタンス
- 名前空間とグループ
- 通知テンプレートの作成

アラートルールタイプ

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana は複数のアラートルールタイプをサポートしています。以下のセクションでは、そのメリットとデメリットについて説明します。この内容は、ユースケースに適したアラートタイプを選択するのに役立ちます。

Grafana マネージドルール

Grafana マネージドルールは、最も柔軟なアラートルールの種類です。これにより、既存の任意のデータソースのデータに対して動作するアラートを作成できます。

あらゆるデータソースをサポートするだけでなく、[式]を追加してデータを変換し、アラート条件を表現できます。

Mimir、Loki、Cortex ルール

Mimir、Loki、または Cortex アラートを作成するには、互換性のある Prometheus データソースが必要です。データソースが互換性があるかどうかを確認するには、データソースをテストし、Ruler API がサポートされているかどうかの詳細を確認します。

記録ルール

記録ルールは、Mimir、Loki、Cortex などの互換性のある Prometheus データソースでのみ使用できます。

記録ルールを使用すると、式の結果を新しい時系列のセットに保存できます。これは、集計データに対してアラートを実行する場合や、同じ式を繰り返しクエリするダッシュボードがある場合に便利です。

Prometheus での記録ルールの詳細については、こちらを参照してください。

アラートインスタンス

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana マネージドアラートは、多次元アラートをサポートします。各アラートルールは、複数のアラートインスタンスを作成できます。これは、1 つの式で複数のシリーズを観察する場合に強力です。

次の PromQL 式を検討してください。

```
sum by(cpu) (
  rate(node_cpu_seconds_total{mode!="idle"}[1m])
)
```

この式を使用するルールは、評価中に監視している CPU の数と同じ数のアラートインスタンスを作成し、単一のルールで各 CPU のステータスをレポートできるようにします。

名前空間とグループ

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートは、Grafana マネージドルールのフォルダと、Mimir、Loki、または Prometheus のルールとグループ名の名前空間を使用して整理できます。

名前空間

Grafana マネージドルールを作成する場合、 フォルダを使用してアクセス制御を実行し、特定のフォルダ内のすべてのルールへのアクセスを許可または拒否できます。

グループ

グループ内のすべてのルールは、同じ間隔で評価されます。

グループ内のアラートルールと記録ルールは、常に順番に評価されます。つまり、同時に出現順に評価されるルールはありません。

> ℹ Tip
> ルールを異なる間隔で同時に評価したい場合は、異なるグループに保存することを検討してください。

通知テンプレートの作成

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントを介して送信される通知は、通知テンプレートを使用して構築されます。Grafana のデフォルトテンプレートは Go テンプレートシステムに基づいており、一部のフィールドはテキストとして評価され、他のフィールドは HTML (エスケープに影響する可能性があります) として評価されます。

デフォルトのテンプレート [default_template.go] は、カスタムテンプレートの便利なリファレンスとして機能します。

コンタクトポイントに使用するほとんどのフィールドはテンプレート化できるため、再利用可能なカスタムテンプレートを作成し、複数のコンタクトポイントで使用すると良いでしょう。テンプレートを使用したカスタム通知の詳細については、「通知のカスタマイズ」を参照してください。

テンプレートのネスト

テンプレートは他のテンプレートに埋め込むことができます。

例えば、 `define` キーワードを使用してテンプレートフラグメントを定義できます。

```
{{ define "mytemplate" }}
  {{ len .Alerts.Firing }} firing. {{ len .Alerts.Resolved }} resolved.
{{ end }}
```

その後、 `template` キーワードを使用して、このフラグメント内にカスタムテンプレートを埋め込むことができます。例:

```
Alert summary:
{{ template "mytemplate" . }}
```

以下の任意のビルトインのテンプレートオプションを使用して、カスタムテンプレートを埋め込むことができます。

名前	メモ
`default.title`	全体的なステータス情報を表示します。
`default.message`	発生中および解決済みのアラートの概要をフォーマット付きで提供します。
`teams.default.message`	`default.messsage` と同様に、Microsoft Teams 用にフォーマットされています。

通知テンプレートの HTML

アラート通知テンプレートの HTML はエスケープされます。結果の通知では、HTML のレンダリングはサポートされていません。

一部の通知機能は、結果の通知のルックアンドフィールを変更する代替方法をサポートしています。例えば、Grafana は `<grafana-install-dir>/public/emails/ng_alert_notification.html` に E メールをアラートするためのベーステンプレートをインストールします。このファイルを編集して、すべてのアラートメールの外観を変更できます。

数値データのアラート

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、Grafana が時系列データではなく数値データのアラートを処理する方法について説明します。

特定のデータソースの中で、時系列ではない数値データは、サーバーサイド式 (SSE) で直接アラートしたり、渡したりすることができます。これにより、データソース内の処理と効率が向上し、ア

ラートルールを簡素化することができます。時系列データではなく数値データを基にアラートを送信する場合、ラベル付きの各時系列を 1 つの数値に集約する必要はありません。代わりに、ラベル付き番号が Grafana に返されます。

表形式のデータ

この機能は、表形式データをクエリするバックエンドデータソースでサポートされています。

- MySQL 、Postgres、MSSQL、Oracle などの SQL データソース。
- Azure Kusto ベースのサービス: Azure Monitor (Logs)、Azure Monitor (Azure Resource Graph)、および Azure Data Explorer。

Grafana 管理のアラートまたは SSE を含むクエリは、以下の場合、これらのデータソースでは数値として扱われます。

- データソースクエリの「Format AS」オプションが「テーブル」に設定されている。
- クエリから Grafana に返されるテーブルレスポンスには、1 つの数値 (int、double、float など) 列と、オプションで追加の文字列列のみが含まれている。

文字列がある場合、それらの列はラベルになります。列の名前はラベル名になり、各行の値は対応するラベルの値になります。複数の行が返された場合、各行はラベルによって一意に識別される必要があります。

例

「DiskSpace」という名前の MySQL テーブルの場合:

時間	ホスト	ディスク	PercentFree
2021 年 6 月 7 日	web1	/etc	3
2021 年 6 月 7 日	web2	/var	4
2021 年 6 月 7 日	web3	/var	8
...	...	...	...

データを時間でフィルタリングしてクエリを実行できますが、時間系列は Grafana に返す必要はありません。例えば、空き容量が 5% 未満の場合にホスト、ディスクごとにトリガーされるアラートです。

```
SELECT Host , Disk , CASE WHEN PercentFree  < 5.0 THEN PercentFree  ELSE 0 END FROM (
   SELECT
      Host,
      Disk,
      Avg(PercentFree)
   FROM DiskSpace
   Group By
      Host,
      Disk
   Where __timeFilter(Time)
```

このクエリでは、次のようなレスポンステーブルが Grafana に返されます。

ホスト	ディスク	PercentFree
web1	/etc	3
web2	/var	4
web3	/var	0

このクエリがアラートルールの条件として使用されると、ゼロ以外のクエリはアラートになります。その結果、次の 3 つのアラートインスタンスが生成されます。

ラベル	ステータス
{Host=web1,disk=/etc}	[アラート]
{Host=web2,disk=/var}	[アラート]
{Host=web3,disk=/var}	普通

ラベルと注釈

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ラベルと注釈には、アラートに関する情報が含まれています。ラベルと注釈はどちらも同じ構造であり、名前付き値のセットですが、用途は異なります。ラベルまたは同等の注釈の例は、`alertname="test"` である場合があります。

ラベルと注釈の主な違いは、ラベルはアラートを他のすべてのアラートと区別するために使用され、注釈は既存のアラートに追加情報を追加するために使用されるという点です。

例えば、2 つの高 CPU アラートを考えてみましょう。1 つは `server1` 用、もう 1 つは `server2` 用です。このような例では、最初のアラートに `server="server1"` というラベルがあり、2 番目のアラートに `server="server2"` というラベルがある、`server` というラベルがある場合があります。ただし、 `server1` や `75%` がサーバーの名前と CPU 使用率に置き換えられる `"The CPU usage for server1 is above 75%."` などの各アラートに説明を追加することもできます (これを行う方法については、ラベルと注釈のテンプレート作成 のドキュメントを参照してください)。このような説明は、注釈としてより適しています。

ラベル

ラベルには、アラートを識別する情報が含まれます。ラベルの例は `server=server1` です。各アラートには複数のラベルを含めることができ、アラートのラベルの完全なセットはラベルセットと呼ばれます。アラートを識別するのはこのラベルセットです。

例えば、アラートにラベルセット `{alertname="High CPU usage",server="server1"}` が設定されているのに対し、別のアラートにラベルセット `{alertname="High CPU usage",server="server2"}` が設定されている場合があります。`alertname` ラベルは同じですが、`server` ラベルは異なるため、これらは 2 つの別々のアラートです。

アラートのラベルセットは、データソースのラベル、アラートルールのカスタムラベル、`alertname` などの予約済みラベルの組み合わせです。

カスタムラベル

カスタムラベルは、アラートルールからの追加ラベルです。注釈と同様に、カスタムラベルには名前が必要です。その値には、アラートが発射したときに評価されるテキストコードとテンプレートコードの組み合わせを含めることができます。カスタムラベルをテンプレート化する方法については、こちらを参照してください。

テンプレートでカスタムラベルを使用する場合、多数の個別のアラートが生成されるのを避けるため、ラベルの値がアラートルールの連続した評価間で変化しないようにすることが重要です。ただし、テンプレートが異なるアラートに対して異なるラベル値を生成しても問題ありません。例えば、値が変更されるたびに新しいアラートセットが作成されてしまうため、クエリの値をカスタムラベルに入れないでください。代わりに注釈を使用します。

また、アラートのラベルセットに同じ名前のラベルが 2 つ以上ないことを確認することも重要です。カスタムラベルの名前がデータソースのラベルと同じ場合は、そのラベルが置き換えられます。ただし、カスタムラベルの名前が予約済みラベルと同じ場合、カスタムラベルはアラートから省略されます。

注釈

注釈は、既存のアラートに追加情報を追加する名前付きペアです。Grafana には、`description`、`summary`、`runbook_url`、`dashboardUId`、`panelId` など、推奨される注釈が多数あります。カスタムラベルと同様に、注釈には名前が必要です。その値には、アラートが発射したときに評価されるテキストコードとテンプレートコードの組み合わせを含めることができます。注釈にテンプレートコードが含まれている場合は、アラートが発生したときにテンプレートが 1 回評価されます。アラートが解決されても、再評価されません。注釈をテンプレートする方法については、こちらを参照してください。

ラベル一致の仕組み

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ラベルとラベルマッチャーを使用して、アラートルールを通知ポリシーとサイレンスにリンクできます。これにより、アラートインスタンスを柔軟に管理し、処理するポリシーとサイレンスするアラートを指定できます。

ラベルマッチャーは、ラベル、値、演算子 の 3 つの異なる部分で構成されます。

- [ラベル]フィールドは、一致するラベルの名前です。ラベル名と完全に一致する必要があります。
- [値]フィールドは、指定されたラベル名の対応する値と一致します。一致方法は、演算子の値によって異なります。
- [演算子] フィールドは、ラベル値と一致させる演算子です。利用できる演算子は次のとおりです。

演算子	説明
=	値と完全に等しいラベルを選択します。
!=	値と等しくないラベルを選択します。
=~	値と正規表現が一致するラベルを選択します。
!~	値と正規表現が一致しないラベルを選択します。

複数のラベルマッチャーを使用している場合は、AND 論理演算子を使用して組み合わせられます。つまり、ルールをポリシーにリンクするには、すべてのマッチャーが一致する必要があります。

シナリオの例

アラートに次のラベルセットを定義する場合：

```
{ foo=bar, baz=qux, id=12 }
```

次に：

- `foo=bar` として定義されたラベルマッチャーは、このアラートルールと一致します。
- `foo!=bar` として定義されたラベルマッチャーがこのアラートルールと一致しません。
- `id=~[0-9]+` として定義されたラベルマッチャーは、このアラートルールと一致します。
- `baz!~[0-9]+` として定義されたラベルマッチャーは、このアラートルールと一致します。

- `foo=bar`と`id=~[0-9]+`として定義された 2 つのラベルマッチャーは、このアラートルールと一致します。

Grafana アラートのラベル

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このトピックでは、ラベルがアラートの基本コンポーネントである理由について説明します。

- アラートの完全なラベルセットは、Grafana アラート内のアラートを一意に識別するものです。
- アラートマネージャーはラベルを使用して、通知ポリシーのサイレンスとアラートグループのアラートを照合します。
- アラート UI には、そのルールの評価中に生成されたすべてのアラートインスタンスのラベルが表示されます。
- コンタクトポイントには、通知を生成する際にラベルにアクセスして、そのアラートに固有の情報を含めることができます。
- アラートルールにラベルを追加できます。ラベルは手動で設定でき、テンプレート関数を使用し、他のラベルを参照できます。アラートルールに追加されたラベルは、ラベル間で競合が発生した場合に優先されます (Grafana の予約済みラベルの場合を除きます。詳細については、以下を参照してください)。

外部アラートマネージャーの互換性

Grafana の組み込みアラートマネージャーは、Unicode ラベルキーと値の両方をサポートしています。外部 Prometheus アラートマネージャーを使用している場合、ラベルキーは[データモデル]と互換性がある必要があります。つまり、ラベルキーには、ASCII 文字、数字およびアンダースコアのみが含まれ、正規表現 `[a-zA-Z_][a-zA-Z0-9_]*` と一致する必要があります。無効な文字は、Grafanaアラートエンジンによって削除または置換されてから、次の規則に従って外部アラートマネージャーに送信されます。

- `Whitespace` は削除されます。
- `ASCII characters` は、_に置き換えられます。
- `All other characters` は、小文字の 16 進数表現に置き換えられます。これが最初の文字の場合、_ というプレフィックスがつきます。

> **Note**
>
> 複数のラベルキーが同じ値にサニタイズされている場合、重複にはサフィックスとして追加された元のラベルの短いハッシュが付加されます。

Grafana 予約済みラベル

> **Note**
>
> `grafana_` のプレフィックスがついたラベルは、Grafana によって特別な用途のために予約されています。`grafana_` で始まる手動で設定されたラベルが追加されると、競合時に上書きされます。

Grafana 予約済みラベルは、手動で設定されたラベルと同じ方法で使用できます。利用可能な予約済みラベルの現在のリストは次のとおりです。

ラベル	説明
grafana_folder	アラートを含むフォルダのタイトル。

ラベルと注釈のテンプレート作成

> **⚠** このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、Prometheus と同じようにラベルと注釈をテンプレート化します。Prometheus を使用した経験がある場合は、アラートのラベルと値を含む `$labels` 変数と `$value` 変数に精通していると思います。アラートが Prometheus データソースを使用していない場合でも、Grafana で同じ変数を使用できます。以前に Prometheus を使用したことがない場合は、これらの変数のそれぞれとテンプレートの作成方法をこのページで説明するのでご安心ください。

Go のテンプレート言語

ラベルと注釈のテンプレートは、Go のテンプレート言語 text/template で記述されます。

開始と終了のタグ

テキスト/テンプレートでは、テンプレートが変数を印刷するか、if ステートメントなどのコントロール構造を実行するかに関係なく、テンプレートは {{で始まり、}} で終わります。これは、変数の出力に {{ と }} を使用し、制御構造に {% と %} を使用する Jinja などの他のテンプレート言語とは異なります。

印刷

何かの値を印刷するには、{{ と }} を使用します。関数の結果または変数の値を印刷できます。例えば、`$labels` 変数を印刷するには、次のように書き込みます。

```
{{ $labels }}
```

ラベルを反復処理する

`$labels` 内の各ラベルを反復処理するには、`range` を使用できます。ここでは、$k は名前を表し、$v は現在のラベルの値を表します。例えば、クエリがラベル `instance=test` を返した場合、$k は `instance` になり、$v は `test` になります。

```
{{ range $k, $v := $labels }}
{{ $k }}={{ $v }}
{{ end }}
```

ラベル、値、値変数

ラベル変数

`$labels` 変数には、クエリのラベルが含まれます。たとえば、インスタンスがダウンしているかどうかを確認するクエリは、停止しているインスタンスの名前を含むインスタンスラベルを返す場

合があります。例えば、インスタンスの 1 つが 5 分以上ダウンしたときに発生するアラートルールがあるとします。どのインスタンスがダウンしているかを知らせる概要をアラートに追加します。$labels 変数を使用すると、概要にインスタンスラベルを印刷する概要を作成できます。

```
Instance {{ $labels.instance }} has been down for more than 5 minutes
```

ドット付きのラベル

印刷するラベルに、テンプレート内の同じドットを使用して名前にドット (ピリオド) が含まれている場合、機能しません。

```
Instance {{ $labels.instance.name }} has been down for more than 5 minutes
```

これは、テンプレートが $labels.instance で name という既存のフィールドを使用しようとしているためです。代わりに、index 関数を使用する必要があります。この関数は、$labels 変数 instance.name にラベルを出力します。

```
Instance {{ index $labels "instance.name" }} has been down for more than 5 minutes
```

値変数

$value 変数の動作は Prometheus とは異なります。Prometheusでは、$value は式の値を含む浮動小数点数ですが、Grafana では、このアラートルールのすべてのしきい値、縮小式、数式およびクラシック条件のラベルと値を含む文字列です。クエリの結果は 10 から 10,000 行またはメトリクスのどこかに返される可能性があるため、含まれていません。

アラートの概要で $value 変数を使用する場合:

```
{{ $labels.service }} has over 5% of responses with 5xx errors: {{ $value }})
```

概要は、次のようになります。

```
api has an over 5% of responses with 5xx errors: [ var='B' labels={service=api}
 value=6.789 ]
```

ここでは、var='B' は RefID B を持つ式を指します。Grafana では、すべてのクエリと式は、アラートルール内の各クエリと式を識別する RefID によって識別されます。同様に、labels={service=api} はラベルを表し、value=6.789 は値を表します。

RefID A がないことが観察された可能性があります。これは、ほとんどのアラートルールでは RefID A がクエリを参照し、クエリが多くの行や時系列を返す可能性があるため、$value に含まれていないためです。

値変数

$value 変数に必要以上の情報が含まれている場合は、代わりに $values を使用して個々の式のラベルと値を印刷できます。$value とは異なり、$values 変数は、各式のラベルと浮動小数点値を含むオブジェクトのテーブルで、RefID によってインデックス化されます。

アラートの概要に RefID B を使用して式の値を印刷する場合：

```
{{ $labels.service }} has over 5% of responses with 5xx errors: {{ $values.B }}%
```

概要には、値のみが含まれます。

```
api has an over 5% of responses with 5xx errors: 6.789%
```

ただし、{{ $values.B }} は数字 6.789 を出力しますが、実際には RefID B のラベルと値の両方を含むオブジェクトを出力する文字列であり、B の浮動小数点値ではありません。RefID B の浮動小数点値を使用するには、$values.B の Value フィールドを使用する必要があります。アラートの概要の浮動小数点値をヒューマナイズする場合：

```
{{ $labels.service }} has over 5% of responses with 5xx errors: {{ humanize
 $values.B.Value }}%
```

データなし、ランタイムエラー、タイムアウト

アラートルールのクエリがデータなしを返す場合、またはデータソースのエラーやタイムアウトが原因で失敗した場合、そのクエリを使用するしきい値、縮小式、または数式もデータや エラーを返しません。この場合、これらの式は $values には存在しません。使用する前に RefID が存在することを確認することをお勧めします。RefID が存在しない場合、クエリがデータや エラーを返さない場合にテンプレートが壊れます。これは、if ステートメントを使用して行うことができます。

```
{{ if $values.B }}{{ $labels.service }} has over 5% of responses with 5xx errors:
 {{ humanizePercentage $values.B.Value }}{{ end }}
```

クラシック条件

ルールがしきい値、縮小式、および数式の代わりにクラシック条件を使用する場合、$values 変数はクラシック条件の条件の Ref ID と位置の両方によってインデックス化されます。例えば、RefID B に 2 つの条件を含むクラシック条件がある場合、$values には 2 つの条件 B0 と B1 が含まれます。

```
The first condition is {{ $values.B0 }}, and the second condition is {{ $values.B1 }}
```

関数

ラベルと注釈を展開するときにも、次の関数を使用できます。

args

args 関数は、オブジェクトのリストをキー arg0、arg1 などのマップに変換します。これは、複数の引数をテンプレートに渡せるようにすることを目的としています。

例

```
{{define "x"}}{{.arg0}} {{.arg1}}{{end}}{{template "x" (args 1 "2")}}
```

```
1 2
```

externalURL

externalURL 関数は、ini ファイル (複数可) で設定された Grafana サーバーの外部 URL を返します。

例

```
{{ externalURL }}
```

```
https://example.com/grafana
```

graphLink

graphLink 関数は、指定された式とデータソースの Grafana バージョン 9 で探索する のグラフィックビューへのパスを返します。

例

```
{{ graphLink "{\"expr\": \"up\", \"datasource\": \"gdev-prometheus\"}" }}
```

```
/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-
prometheus","expr":"up","instant":false,"range":true}]
```

humanize

humanize 関数は 10 進数をヒューマナイズします。

例

```
{{ humanize 1000.0 }}
```

```
1k
```

humanize1024

humanize1024 は humanize と似ていますが、1000 ではなく 1024 をベースとして使用します。

例

```
{{ humanize1024 1024.0 }}
```

```
1ki
```

humanizeDuration

humanizeDuration 関数は、期間を秒単位でヒューマナイズします。

例

```
{{ humanizeDuration 60.0 }}
```

```
1m 0s
```

humanizePercentage

humanizePercentage 関数は、比率値をパーセンテージにヒューマナイズします。

例

```
{{ humanizePercentage 0.2 }}
```

```
20%
```

humanizeTimestamp

humanizeTimestamp 関数は Unix タイムスタンプをヒューマナイズします。

例

```
{{ humanizeTimestamp 1577836800.0 }}
```

```
2020-01-01 00:00:00 +0000 UTC
```

match

match 関数は、テキストを正規表現パターンと一致させます。

例

```
{{ match "a.*" "abc" }}
```

```
true
```

pathPrefix

pathPrefix 関数は、ini ファイル (複数可) で設定された Grafana サーバーのパスを返します。

例

```
{{ pathPrefix }}
```

```
/grafana
```

tableLink

`tableLink` 関数は、指定された式とデータソースの Grafana バージョン 9 で探索する の表形式ビューへのパスを返します。

例

```
{{ tableLink "{\"expr\": \"up\", \"datasource\": \"gdev-prometheus\"}" }}
```

```
/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-
prometheus","expr":"up","instant":true,"range":false}]
```

title

`title` 関数は、各単語の最初の文字を大文字にします。

例

```
{{ title "hello, world!" }}
```

```
Hello, World!
```

toLower

`toLower` 関数はすべてのテキストを小文字で返します。

例

```
{{ toLower "Hello, world!" }}
```

```
hello, world!
```

toUpper

`toUpper` 関数は、すべてのテキストを大文字で返します。

例

```
{{ toUpper "Hello, world!" }}
```

```
HELLO, WORLD!
```

reReplaceAll

`reReplaceAll` 関数は、正規表現に一致するテキストを置き換えます。

例

```
{{ reReplaceAll "localhost:(.*)" "example.com:$1" "localhost:8080" }}
```

```
example.com:8080
```

アラートルールの状態と正常性

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールの状態と正常性は、アラートに関するいくつかの主要なステータスインジケータを理解するのに役立ちます。

アラートルールの状態 、アラートインスタンスの状態、アラートルールの正常性の 3 つの主要なコンポーネントがあります。それぞれに関連はありますが、各コンポーネントは微妙に異なる情報を伝達します。

アラートルールの状態

アラートルールは、次のいずれかの状態になります。

状態	説明
普通	評価エンジンによって返される時系列のいずれも Pending または Firing 状態ではありません。
保留中	評価エンジンから返される少なくとも 1 つの時系列は Pending です。
発射	評価エンジンから返される少なくとも 1 つの時系列は Firing です。

> **Note**
>
> アラートは最初に pending に移行し、次に firing に移行します。そのため、アラートが発生する前に少なくとも 2 つの評価サイクルが必要になります。

アラートインスタンスの状態

アラートインスタンスは、次のいずれかの状態になります。

状態	説明
普通	発射も保留中もなく、すべてが正しく機能しているアラートの状態。
保留中	設定されたしきい値期間未満でアクティブであったアラートの状態。
[アラート]	設定されたしきい値期間よりも長くアクティブであったアラートの状態。
NoData	設定された時間枠のデータを受信しない状態。
エラー	アラートルールの評価を試みたときに発生したエラー。

アラートルールの正常性

アラートルールは、次のいずれかの正常性ステータスとなります。

状態	説明
Ok	アラートルールを評価するときにエラーがありません。
エラー	アラートルールの評価中にエラーが発生しました。
NoData	ルール評価中に返される少なくとも 1 つの時系列にデータがありません。

`NoData` および `Error` [の特別なアラート]

アラートルールの評価で状態 `NoData` または `Error` が生成されると、Grafana アラートは、次の追加ラベルを含むアラートインスタンスを生成します。

ラベル	説明
AlertName	状態に応じて、`DatasourceNoData` または `DatasourceError` のいずれか。
datasource_uid	状態の原因となったデータソースの UID。

これらのアラートは、サイレンスを追加したり、コンタクトポイントにルーティングしたりすることで、通常のアラートと同じ方法で処理できます。

コンタクトポイント

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントを使用して、アラートルールが発生したときにコンタクトに通知する方法を定義します。コンタクトポイントは、E メール、Slack、ウェブフックなど、1 つ以上のコンタクトポイントタイプを持つことができます。アラートルールが発動すると、コンタクトポイントにリストされているすべてのコンタクトポイントタイプに通知が送信されます。Grafana アラートマネージャーと外部アラートマネージャーのコンタクトポイントを設定できます。

通知テンプレートを使用して、コンタクトポイントタイプの通知メッセージをカスタマイズすることもできます。

サポートされているコンタクトポイントタイプ

次の表に、Grafana でサポートされているコンタクトポイントのタイプを示します。

名前	タイプ
Amazon SNS	`sns`
OpsGenie	`opsgenie`
Pager Duty	`pagerduty`
Slack	`slack`
VictorOps	`victorops`

コンタクトポイントの詳細については、「コンタクトポイント (通知先) の使用」と「通知のカスタマイズ」を参照してください。

通知

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana はアラートマネージャーを使用して、アラートの発射と解決の通知を送信します。Grafana には、ユーザーインターフェイスで「Grafana」と呼ばれる独自のアラートマネージャーがありますが、Prometheus アラートマネージャーなどの他のアラートマネージャーからの通知の送信もサポートしています。Grafana アラートマネージャーは、通知ポリシーと問い合わせポイントを使用して、通知の送信方法と送信先、通知の送信頻度、アラートをすべて同じ通知で送信するか、ラベルのセットに基づいてグループ化された通知で送信するか、個別の通知として送信するかを設定します。

通知ポリシー

通知ポリシーは、通知の送信日時と送信場所を制御します。通知ポリシーでは、すべてのアラートを同じ通知でまとめて送信するか、ラベルのセットに基づいてグループ化された通知でアラートを送信するか、個別の通知としてアラートを送信するかを選択できます。各通知ポリシーを設定して、通知の送信頻度を制御するだけでなく、1 日の特定時刻および特定の曜日に通知を禁止するための 1 つ以上のミュートタイミングを設定することもできます。

通知ポリシーは、ツリーのルートにルートポリシーと呼ばれる通知ポリシーがあるツリー構造で構成されています。ルートポリシーは 1 つだけ設定でき、削除できません。

特定のルーティングポリシーはルートポリシーの子であり、一致するラベルのセットに基づいてすべてのアラートまたはアラートのサブセットを照合するために使用できます。通知ポリシーは、一致するラベルがアラート内のラベルと一致するときにアラートと一致します。

特定のルーティングポリシーには「ネストされたポリシー」と呼ばれる独自の子ポリシーを含めることができ、アラートの追加マッチングが可能になります。特定のルーティングポリシーの例としては、Ops チームにインフラストラクチャアラートを送信する場合があります。子ポリシーは、Pagerduty に高優先度アラートを送信し、Slack に低優先度アラートを送信することができます。

ラベルに関係なく、すべてのアラートはルートポリシーと一致します。ただし、ルートポリシーがアラートを受信すると、特定の各ルーティングポリシーが調査され、アラートに一致する最初の特定のルーティングポリシーにアラートが送信されます。特定のルーティングポリシーにさらに子ポリシーがある場合、ネストされたポリシーの 1 つとアラートとのマッチングを試行できます。ネストされたポリシーがアラートと一致しない場合、特定のルーティングポリシーが一致するポリシーになります。特定のルーティングポリシーがないか、アラートに一致する特定のルーティングポリシーがない場合、ルートポリシーは一致するポリシーです。

コンタクトポイント

コンタクトポイントには、通知を送信するための設定が含まれています。コンタクトポイントは統合のリストであり、それぞれが特定の E メールアドレス、サービス、または URL に通知を送信します。コンタクトポイントには、同じ種類の複数の統合、または異なる種類の統合の組み合わせを含めることができます。例えば、コンタクトポイントには、Pager Duty 統合、Pager Duty と Slack 統合、または Pager Duty 統合、Slack 統合、および 2 つの Amazon SNS 統合を含めることができます。統合なしでコンタクトポイントを設定することもできますが、その場合、通知は送信されません。

コンタクトポイントは、通知ポリシーに追加されるまで通知を送信できません。通知ポリシーは 1 つのコンタクトポイントにのみアラートを送信できますが、複数の通知ポリシーに同時にコンタクトポイントを追加できます。アラートが通知ポリシーと一致すると、アラートはその通知ポリシーのコンタクトポイントに送信され、その通知ポリシーはその設定の各統合に通知を送信します。

> **Note**
>
> コンタクトポイントでサポートされている統合の詳細については、「コンタクトポイント」を参照してください。

通知テンプレートの作成

テンプレートを使用して通知をカスタマイズできます。例えば、テンプレートを使用して、Slack に送信される通知のタイトルとメッセージを変更することができます。

テンプレートは個々の統合やコンタクトポイントに限定されず、同じコンタクトポイント内の多数の統合や、異なるコンタクトポイント間の統合でも使用できます。例えば、Grafana ユーザーは、`custom_subject_or_title` というテンプレートを作成し、2 つの個別のテンプレートを作成することなく、Pager Duty のテンプレート件名と Slack メッセージのタイトルの両方に使用できます。

すべての通知テンプレートはアラートページの [コンタクトポイント] タブにあり、[Go のテンプレート言語]で記述されます。

サイレンス

サイレンスを使用して、1 つ以上の発射ルールからの通知をミュートできます。サイレンスは、アラートの発射や解決を停止したり、ユーザーインターフェイスで発射アラートを非表示にしたりしません。サイレンスは、分、時間、日、月、または年で設定し、その期間だけ続きます。

アラートの設定

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートの作成と管理に必要な機能と統合を設定します。

トピック

- 外部アラートマネージャーの追加
- Grafana アラートリソースのプロビジョニング

外部アラートマネージャーの追加

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana を設定して、外部のアラートマネージャーを単一のアラートマネージャーとして使用してすべてのアラートを受信するようにできます。この外部アラートマネージャーは、Grafana 内から設定および管理できます。

アラートマネージャーを追加したら、Grafana アラート UI を使用して、サイレンス、コンタクトポイント、通知ポリシーを管理できます。これらのページのドロップダウンオプションを使用すると、アラートマネージャーを切り替えることができます。

> **Note**
>
> Grafana 9.2 以降、アラートページの [管理] タブからの外部アラートマネージャーの URL 設定は廃止されました。これは、今後のリリースで削除されます。

外部アラートマネージャーは、メインの Grafana ナビゲーションメニューから Grafana 設定を使用してデータソースとして設定する必要があります。これにより、Grafana 内から外部アラートマネージャーの連絡先と通知ポリシーを管理できるようになり、以前は URL で外部アラートマネージャーを構成するときに表示されていた HTTP 基本認証資格情報も暗号化されます。

外部アラートマネージャーを追加するには、次のステップに従います。

1. [設定]、[データソース] を続けてクリックします。
2. アラートマネージャーを検索します。
3. 実装を選択し、必要に応じてページのフィールドに入力します。

   データソースをプロビジョニングする場合は、`jsonData` フィールドの `handleGrafanaManagedAlerts` フラグを `true` に設定して、Grafana が管理するアラートをこのアラートマネージャーに送信します。

   > **Note**
   >
   > アラートマネージャーの Prometheus、Grafana Mimir、および Cortex 実装がサポートされています。Prometheus の場合、Grafana アラート UI のコンタクトポイントと通知ポリシーは読み取り専用です。

4. [保存してテスト] をクリックします。

Grafana アラートリソースのプロビジョニング

> **このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。**
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

多くの場合、アラートインフラストラクチャは複雑で、パイプラインの多くの部分が異なる場所に存在することがよくあります。これを複数のチームや組織にまたがってスケールすることは、特に難しい作業です。Grafana アラートプロビジョニングを使用すると、組織に最適な方法でアラートデータを作成、管理、維持できるため、このプロセスが容易になります。

選択できるオプションは 2 つあります。

1. アラートプロビジョニング HTTP API を使用してアラートリソースをプロビジョニングします。

> **Note**
>
> 通常、Grafana UI から API でプロビジョニングされたアラートルールを編集することはできません。
> 編集を有効にするには、 API でアラートルールを作成または編集するときに、x-disable-provenance ヘッダーを次のリクエストに追加します。
>
> ```
> POST /api/v1/provisioning/alert-rules
> PUT /api/v1/provisioning/alert-rules/{UID}
> ```

2. Terraform を使用してアラートリソースをプロビジョニングします。

> **Note**
>
> 現在、Grafana アラートのプロビジョニングでは、アラートルール、コンタクトポイント、ミュートタイミング、テンプレートがサポートされています。ファイルプロビジョニングまたは Terraform を使用してプロビジョニングされたアラートリソースは、それらを作成したソースでのみ編集でき、Grafana やその他のソース内から編集することはできません。例えば、ディスクからのファイルを使用してアラートリソースをプロビジョニングする場合、Terraform または Grafana 内からデータを編集することはできません。

トピック

- Terraform を使用したアラートリソースの作成と管理

- Grafana でのプロビジョニングされたアラートリソースの表示

Terraform を使用したアラートリソースの作成と管理

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Terraform の Grafana プロバイダーを使用してアラートリソースを管理し、Grafana システムにプロビジョニングします。Terraform プロバイダーによる Grafana アラートのサポートにより、Grafana アラートスタック全体をコードとして簡単に作成、管理、維持できます。

Terraform を使用してアラートリソースを管理する方法の詳細については、Terraform ドキュメントの [Grafana プロバイダー] ドキュメントを参照してください。

Terraform を使用してアラートリソースを作成および管理するには、次のタスクを実行します。

1. プロビジョニング用の API キーを作成します。
2. Terraform プロバイダーを設定します。
3. Terraform でアラートリソースを定義します。
4. `terraform apply` を実行して、アラートリソースをプロビジョニングします。

前提条件

- grafana/grafana [Terraform プロバイダー] 1.27.0 以降があることを確認します。
- また、Grafana 9.1 以降を使用していることを確認します。Grafana バージョン 9 で Amazon Managed Grafana インスタンスを作成した場合、これは当てはまります。

プロビジョニング用の API キーの作成

通常の Grafana API キーを作成して、Grafana で Terraform を認証できます。API キーを使用する既存のツールのほとんどは、新しい Grafana アラートサポートと自動的に連携します。Terraform で使

用するキーの作成の詳細については、[Amazon Managed Grafana オートメーションでの Terraform の使用]を参照してください。

プロビジョニング用の API キーを作成するには

1. CI パイプラインの新しいサービスアカウントを作成します。
2. ロール「アラートルールプロビジョニング API にアクセス」を割り当てます。
3. 新しいサービスアカウントトークンを作成します。
4. Terraform で使用するトークンに名前を付けて保存します。

または、基本認証を使用することもできます。サポートされているすべての認証形式を表示するには、Terraform ドキュメントの [Grafana 認証] を参照してください。

Terraform プロバイダーの設定

Grafana アラートのサポートは、[Grafana Terraform プロバイダー] の一部として含まれています。

以下は、Terraform プロバイダーの設定に使用できる例です。

```
terraform {
    required_providers {
        grafana = {
            source = "grafana/grafana"
            version = ">= 1.28.2"
        }
    }
}

provider "grafana" {
    url = <YOUR_GRAFANA_URL>
    auth = <YOUR_GRAFANA_API_KEY>
}
```

コンタクトポイントとテンプレートのプロビジョニング

コンタクトポイントは、アラートスタックを外部に接続します。これは、外部システムに接続する方法と通知の配信先を Grafana に伝えます。選択できる統合は 15 種類以上あります。この例では、Slack コンタクトポイントを使用します。

コンタクトポイントとテンプレートをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。*<slack-webhook-url>* を Slack Webhook URL (またはその他のコンタクトポイントの詳細) に置き換えます。

   この例では、アラート通知を Slack に送信するコンタクトポイントを作成します。

```
resource "grafana_contact_point" "my_slack_contact_point" {
    name = "Send to My Slack Channel"

    slack {
        url = <slack-webhook-url>
        text = <<EOT
{{ len .Alerts.Firing }} alerts are firing!

Alert summaries:
{{ range .Alerts.Firing }}
{{ template "Alert Instance Template" . }}
{{ end }}
EOT
    }
}
```

2. 通知のテキストをテキストフィールドに入力します。

   `text` フィールドは Go 形式のテンプレートをサポートします。これにより、Terraform で Grafana アラート通知テンプレートを直接管理できます。

3. `terraform apply` コマンドを実行します。
4. Grafana UI に移動し、コンタクトポイントの詳細を確認します。

   UI から Terraform 経由でプロビジョニングされたリソースを編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、コンタクトポイントが正しく機能することを確認します。

> **Note**
>
> 多くのコンタクトポイントで同じテンプレートを再利用できます。上記の例では、`{{ template "Alert Instance Template" . }}` ステートメントを使用して共有テンプレートが埋め込まれています
> その後、このフラグメントは Terraform で個別に管理できます。

```
resource "grafana_message_template" "my_alert_template" {
    name = "Alert Instance Template"

    template = <<EOT
{{ define "Alert Instance Template" }}
Firing: {{ .Labels.alertname }}
Silence: {{ .SilenceURL }}
{{ end }}
EOT
}
```

通知ポリシーとルーティングのプロビジョニング

通知ポリシーは、アラートインスタンスをどこにルーティングするかではなく、どのようにルーティングするかを Grafana に指示します。ラベルとマッチャーのシステムを使用して、発動アラートを以前に定義したコンタクトポイントに接続します。

通知ポリシーとルーティングをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。

   この例では、アラートは alertname によってグループ化されます。つまり、同じ名前のアラートから送信される通知は、同じ Slack メッセージにグループ化されます。

   特定の通知を別の方法でルーティングする場合は、サブポリシーを追加できます。サブポリシーを使用すると、ラベルマッチングに基づいて異なるアラートにルーティングを適用できます。この例では、ラベル a=b のすべてのアラートにミュートタイミングを適用します。

```
resource "grafana_notification_policy" "my_policy" {
    group_by = ["alertname"]
    contact_point = grafana_contact_point.my_slack_contact_point.name

    group_wait = "45s"
    group_interval = "6m"
    repeat_interval = "3h"

    policy {
        matcher {
            label = "a"
            match = "="
```

```
            value = "b"
        }
        group_by = ["..."]
        contact_point = grafana_contact_point.a_different_contact_point.name
        mute_timings = [grafana_mute_timing.my_mute_timing.name]

        policy {
            matcher {
                label = "sublabel"
                match = "="
                value = "subvalue"
            }
            contact_point = grafana_contact_point.a_third_contact_point.name
            group_by = ["..."]
        }
    }
}
```

2. mute_timings フィールドで、ミュートタイミングを通知ポリシーにリンクします。
3. `terraform apply` コマンドを実行します。
4. Grafana UI に移動し、通知ポリシーの詳細を確認します。

> **Note**
>
> Terraform からプロビジョニングされたリソースを UI から編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、通知ポイントが正しく機能していることを確認します。

ミュートタイミングのプロビジョニング

ミュートタイミングを使用すると、定義された期間のアラート通知をミュートできます。

ミュートタイミングをプロビジョニングするには

1. このコードブロックをローカルマシンの .tf ファイルにコピーします。

   この例では、アラート通知は週末にミュートされます。

```
resource "grafana_mute_timing" "my_mute_timing" {
    name = "My Mute Timing"
```

```
    intervals {
        times {
          start = "04:56"
          end = "14:17"
        }
        weekdays = ["saturday", "sunday", "tuesday:thursday"]
        months = ["january:march", "12"]
        years = ["2025:2027"]
    }
}
```

2. `terraform apply` コマンドを実行します。
3. Grafana UI に移動し、ミュートタイミングの詳細を確認します。
4. `mute_timings` フィールドを使用して、通知ポリシーで新しく作成したミュートタイミングを参照します。これにより、通知の一部またはすべてにミュートタイミングが適用されます。

> **Note**
>
> Terraform からプロビジョニングされたリソースを UI から編集することはできません。これにより、アラートスタックは常にコードと同期したままになります。

5. [テスト] をクリックして、ミュートタイミングが正しく機能していることを確認します。

アラートルールのプロビジョニング

アラートルールを使用すると、あらゆる Grafana データソースに対してアラートを実行できます。これは、既に設定済みのデータソースを使用することも、アラートルールとともに [Terraform でデータソースを定義]することもできます。

アラートルールをプロビジョニングするには

1. クエリするデータソースと、ルールを保存するフォルダを作成します。

   この例では、テスト用 TestData データソースの設定 データソースが使用されます。

   アラートは Grafana の任意のバックエンドデータソースに対して定義できます。

```
resource "grafana_data_source" "testdata_datasource" {
    name = "TestData"
    type = "testdata"
```

```
}

resource "grafana_folder" "rule_folder" {
    title = "My Rule Folder"
}
```

2. アラートルールを定義します。

   アラートルールの詳細については、[Grafana マネージドアラートの作成方法] を参照してください。

3. 1 つ以上のルールを含むルールグループを作成します。

   この例では、grafana_rule_group リソースグループが使用されます。

```
resource "grafana_rule_group" "my_rule_group" {
    name = "My Alert Rules"
    folder_uid = grafana_folder.rule_folder.uid
    interval_seconds = 60
    org_id = 1

    rule {
        name = "My Random Walk Alert"
        condition = "C"
        for = "0s"

        // Query the datasource.
        data {
            ref_id = "A"
            relative_time_range {
                from = 600
                to = 0
            }
            datasource_uid = grafana_data_source.testdata_datasource.uid
            // `model` is a JSON blob that sends datasource-specific data.
            // It's different for every datasource. The alert's query is defined
 here.
            model = jsonencode({
                intervalMs = 1000
                maxDataPoints = 43200
                refId = "A"
            })
        }
```

```
        // The query was configured to obtain data from the last 60 seconds. Let's
 alert on the average value of that series using a Reduce stage.
        data {
            datasource_uid = "__expr__"
            // You can also create a rule in the UI, then GET that rule to obtain
 the JSON.
            // This can be helpful when using more complex reduce expressions.
            model = <<EOT
{"conditions":[{"evaluator":{"params":[0,0],"type":"gt"},"operator":
{"type":"and"},"query":{"params":["A"]},"reducer":{"params":
[],"type":"last"},"type":"avg"}],"datasource":
{"name":"Expression","type":"__expr__","uid":"__expr__"},"expression":"A","hide":false,"int
EOT
            ref_id = "B"
            relative_time_range {
                from = 0
                to = 0
            }
        }

        // Now, let's use a math expression as our threshold.
        // We want to alert when the value of stage "B" above exceeds 70.
        data {
            datasource_uid = "__expr__"
            ref_id = "C"
            relative_time_range {
                from = 0
                to = 0
            }
            model = jsonencode({
                expression = "$B > 70"
                type = "math"
                refId = "C"
            })
        }
    }
}
```

4. Grafana UI に移動し、アラートルールを確認します。

   アラートルールが発射しているかどうかを確認できます。また、アラートルールの各クエリステージを視覚化することもできます。

   アラートが発生すると、Grafana は定義したポリシーを通じて通知をルーティングします。

例えば、コンタクトポイントとして Slack を選択した場合、Grafana の埋め込みアラートマネージャーは Slack にメッセージを自動的に投稿します。

Grafana でのプロビジョニングされたアラートリソースの表示

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートリソースが Grafana で作成されたことを確認できます。

Grafana でプロビジョニングされたリソースを表示するには

1. Grafana インスタンスを開きます。
2. アラートに移動します。
3. アラートルールなど、アラートリソースフォルダをクリックします。

   プロビジョニング済みのリソースには [プロビジョニング済み] というラベルが付けられているため、手動で作成されていないことがわかります。

Note

Grafana からプロビジョニングされたリソースを編集することはできません。リソースプロパティを変更するには、プロビジョニングファイルを変更して Grafana を再起動するか、ホットリロードを実行します。これにより、ファイルが再度プロビジョニングされた場合やホットリロードが実行された場合に上書きされるリソースへの変更を防止します。

従来のダッシュボードアラートを Grafana アラートに移行する

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana アラートを使用しないワークスペースには、従来のダッシュボードアラートを利用します。新しい Grafana アラートに切り替えるには、 機能にオプトインする必要があります。

、、 AWS CLIまたは Amazon Managed Grafana API を使用して AWS Management Console、Grafana アラートを使用するように Amazon Managed Grafana インスタンスを設定できます。Grafana アラートのオン/オフの切り替えなど、Amazon Managed Grafana の設定方法の詳細については、「Amazon Managed Grafana ワークスペースを設定する」を参照してください。

Note

Grafana アラートを使用する場合、Grafana で定義されたアラートルールは、Prometheusではなく、複数の通知をコンタクトポイント (通知先) に送信します。ネイティブ Grafana アラートを使用している場合は、新しい Grafana アラート機能を有効にせず、従来のダッシュボードアラートのまま使用することをお勧めします。Prometheus データソースで定義されたアラートを表示する場合は、Grafana アラートを有効にすることをお勧めします。これにより、Prometheus アラートマネージャーで作成されたアラートの通知が 1 つだけ送信されます。
この制限は、Grafana v10.4 以降をサポートする Amazon Managed Grafana ワークスペースで削除されました。

Grafana アラートシステムへの移行

Grafana アラートを有効にすると、既存の従来のダッシュボードアラートは Grafana アラートと互換性のある形式で移行されます。移行後のアラートと新しく作成したアラートは、Grafana インスタンスのアラートページに表示されます。Grafana アラートを使用した場合、Grafana が管理するアラートルールは、一致するときに 1 つのアラートではなく、複数の通知を送信します。

従来のダッシュボードアラートと Grafana アラートへの読み取りおよび書き込みアクセス権は、それらが保存されているフォルダのアクセス権限で制御されます。移行中、従来のダッシュボードアラートアクセス許可は、次のように新しいルールアクセス許可と照合されます。

- 元のアラートが属するダッシュボードにアクセス権限が設定されている場合、移行時に元のダッシュボードの権限 (フォルダから継承された権限を含む) 設定で `Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1}` という形式の名前のフォルダが作成されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードがフォルダ内にある場合、ルールはこのフォルダにリンクされ、そのアクセス権限が継承されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードが General (全般) フォルダにある場合、ルールは General (全般) アラートフォルダにリンクされ、ルールにはデフォルトのアクセス権限が継承されます。

> **Note**
>
> Grafana のアラート設定には `NoData` に対する `Keep Last State` オプションがないため、このオプションは従来のルールを移行する際に `NoData` になります。`Error` 処理のためのオプション `Keep Last State` は、新しいオプション `Error` に移行されます。`Keep Last State` の動作に合わせるため、移行時には各アラートルールに対して 1 年間のサイレンスが Amazon Managed Grafana によって自動的に作成されます。

通知チャネルは、適切なルートと受信者が設定されたアラートマネージャー設定に移行されます。デフォルトの通知チャネルは、コンタクトポイントとしてデフォルトのルートに追加されますが、ダッシュボードアラートに関連付けられていない通知チャネルは `autogen-unlinked-channel-recv` ルートに送られます。

制限

- Grafana アラートシステムは、利用可能なすべての Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからアラートルールを取得できない場合があります。
- Grafana アラートと従来のダッシュボードアラートを交互に移行すると、どちらか一方でのみサポートされる機能のデータが失われる可能性があります。

> **Note**
>
> 従来のダッシュボードアラートに戻した場合、Grafana アラートが有効になっている間に新たに作成されたアラートルールやアラート設定に加えられたすべての変更は失われます。

アラートルールの管理

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートルールは、アラートを起動するかどうかを決定する一連の評価基準です。アラートルールは、1 つ以上のクエリと式、条件、評価の頻度、およびオプションとしてその条件の持続時間で構成されます。

評価するデータセットを「クエリと式」で選択し、「条件」にアラートがアラートを発行するための基準 (しきい値) を設定します。間隔は、アラートルールが評価される頻度を指定します。「持続時間」を設定した場合、その条件下にある状態の継続時間を示します。また、アラートルールでは、データがない場合のアラート動作も設定できます。

> **Note**
>
> Grafana マネージドアラートルールは、ルールを保存するフォルダの編集権限を持つユーザーのみが編集または削除できます。
> 外部 Grafana Mimir または Loki インスタンスのアラートルールは、エディタまたは管理者ロールを持つユーザーが編集または削除できます。

トピック

- Grafana 管理のアラートルールの作成

- Grafana Mimir または Loki 管理のアラートルールの作成
- Grafana Mimir または Loki 管理の記録ルールの作成
- Grafana Mimir または Loki ルールグループと名前空間
- アラートルールの表示と編集

Grafana 管理のアラートルールの作成

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、1 つ以上のデータソースをクエリして結果を集約または変換して、それらを相互に比較したり、固定のしきい値と比較したりするアラートルールを作成することができます。これらを実行すると、Grafana はコンタクトポイント (通知先) に通知を送信します。

Grafana 管理のルールの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [新しいアラートルール] を選択します。
3. ステップ 1 で、以下を参照してルール名、タイプ、ストレージの場所を追加します。
   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [ルールタイプ] ドロップダウンから、[Grafana 管理のアラート] を選択します。
   - [フォルダ] ドロップダウンから、ルールの保存先となるフォルダを選択します。フォルダを選択しない場合、ルールは `General` フォルダに保存されます。フォルダを作成する場合、ドロップダウンを選択し、新しいフォルダ名を入力します。
4. ステップ 2 で、評価するクエリと式を追加します。

- 名前はデフォルトのままでも構いませんが、変更する場合、カーソルを合わせて編集アイコンを選択します。
- クエリの場合、ドロップダウンからデータソースを選択します。
- 1 つ以上のクエリまたは式を追加します。
- 式ごとに [クラシック条件]を選択して単一のアラートルールを作成するか、[Math]、[削減]、[リサンプリング] オプションから選択して、各シリーズに個別のアラートを作成します。これらのオプションの詳細については、「1 次元ルールと多次元ルール」を参照してください。
- [Run queries] (クエリの実行) を選択して、クエリが正しく動作することを確認します。

5. ステップ 3 で、条件を追加します。

   - [条件] ドロップダウンから、アラートルールを発動させるクエリまたは式を選択します。
   - Evaluate every (~ごとに評価する) には、評価の頻度を指定します。10 秒の倍数である必要があります。例えば、`1m` や `30s` などです。
   - [評価対象] には、アラートが発行されるために必要な条件の継続時間を指定します。

     > **Note**
     >
     > 条件が違反すると、アラートは `Pending` 状態に入り、そのまま指定された期間条件の違反が継続するとアラートは `Firing` 状態に移行します。条件から外れた場合、`Normal` 状態に戻ります。

   - [Configure no data and error handling] (データがない場合の設定とエラー処理) で、データがない場合のアラート動作を設定します。「データがない場合やエラーが発生した場合の対処」のガイドラインを参照してください。
   - [アラートのプレビュー] を選択してルールを評価し、生成されるアラートを確認します。プレビューでは、データとエラー処理条件は除外されます。

6. ステップ 4 で、ルールに関連するメタデータを追加します。

   - アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「ラベルと注釈」を参照してください。
   - Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
   - カスタムラベルを追加します。

7. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

ルールを作成したら、ルールの通知を作成できます。通知の詳細については、「アラート通知の管理」を参照してください。

1 次元ルールと多次元ルール

Grafana 管理のアラートルールでは、従来の条件を使用してルールを作成することも、多次元ルールを作成することもできます。

1 次元ルール (従来の条件)

従来の条件式を使用して、条件が満たされたときに単一のアラートを発行するルールを作成します。複数のシリーズを返すクエリの場合、Grafana は各シリーズのアラート状態を追跡しません。そのため、複数のシリーズでアラート条件が満たされた場合でも、Grafana が送信するアラートは 1 件のみです。

式のフォーマット方法の詳細については、Grafana ドキュメントの「式」を参照してください。

多次元ルール

クエリで返されるシリーズごとに個別のアラートインスタンスを生成するには、多次元ルールを作成します。

> **Note**
>
> 多次元ルールによって生成された各アラートインスタンスは、アラートの合計クォータにカウントされます。クォータに達すると、ルールは評価されなくなります。多次元ルールのクォータの詳細については、「クォータ到達エラー」を参照してください。

1 つのルールから複数のインスタンスを作成するには、`Math`、`Reduce`、または `Resample` 式を使用して多次元ルールを作成します。例えば、以下のことが可能です。

- 各クエリに `Reduce` 式を追加すると、選択した時間範囲の値が 1 つの値に集計されます。(数値データ を使用するルールには必要ありません)。
- ルールの条件を含む `Math` 式を追加します。もしクエリや reduce 式がすでにアラートを発行すべき場合は正の数、発行しない場合に 0 を返す場合、これは必要ありません。

  例:

  - `$B > 70` は、B クエリ/式の値が 70 を超える場合にアラートを発行します。

- `$B < $C * 100` は、B の値が C の値に 100 を掛けた値より小さい場合にアラートを発行します。比較対象のクエリの結果に複数のシリーズがある場合、異なるクエリのシリーズが同じラベルを持っているか、一方が他方のサブセットである場合に一致します。

> **Note**
>
> Grafana は、テンプレート変数を使用したアラートクエリはサポートしていません。詳細については、コミュニティページ「アラートクエリではアラート設定時にテンプレート変数を使用できません」を参照してください。

多次元ルールのパフォーマンスに関する考慮事項

各アラートインスタンスは、アラートのクォータ (上限数) にカウントされます。アラートの上限数を超えるインスタンスを作成する多次元ルールは評価されず、クォータエラーが返されます。詳細については、「クォータ到達エラー」を参照してください。

多次元アラートは、Grafana ワークスペースや、Grafana がアラートルールを評価するためにクエリを実行するデータソースのパフォーマンスに大きな影響を与える可能性があります。監視システムのパフォーマンスを最適化したい場合、以下の考慮事項を参照してください。

- ルール評価の頻度 – アラートルールの [Evaluate Every] (~ごとに評価する) プロパティでルール評価の頻度を制御します。評価の頻度を許容可能な最小頻度に収めることをお勧めします。
- 結果セットのカーディナリティ – ルールで作成されるアラートインスタンスの数は直接そのパフォーマンスに影響します。例えば、すべての仮想マシン上の各 API パスごとに API 応答エラーを監視しているとします。このセットのカーディナリティは「パスの数 × 仮想マシンの数」になります。例えば、仮想マシンあたりのパスごとではなく、仮想マシンあたりの合計エラー数を監視することで、結果セットのカーディナリティを減らすことができます。
- クエリの複雑さ – データソースが迅速に処理して応答できるクエリにするだけで、リソースの消費を大幅に抑えることができます。これは他の考慮事項よりは重要度が低いものの、他の点での最適化が十分に行われている場合、個別のクエリパフォーマンスの見直しが役立つ可能性があります。また、これらのルールの評価がデータソースに与える影響にも注意する必要があります。アラートクエリは、監視データベースで処理されるクエリの大部分を占めることが多いため、Grafana インスタンスにかかる負荷要因がデータソースにも同様に影響を与えます。

クォータ到達エラー

1 つのワークスペース内に保持できるアラートインスタンスの数にはクォータ (上限) があります。この上限に達した場合、そのワークスペースにこれ以上新たにアラートルールを作成できなくなります。多次元アラートでは、アラートインスタンスの数は時間の経過とともに変動する場合があります。

以下に、アラートインスタンスを使用する際に覚えておくべき重要な点について紹介します。

- 作成するのが 1 次元ルールのみの場合、ルールからはそれぞれ 1 つのアラートインスタンスが作成されます。1 つのワークスペースに作成できるルールの上限は、アラートインスタンスのクォータ数までとなります。
- 多次元ルールからは複数のアラートインスタンスが作成されますが、その数は評価されるまでわかりません。例えば、Amazon EC2 インスタンスの CPU 使用率を追跡するアラートルールを作成した場合、最初は 50 台の EC2 インスタンスが対象となり、50 個のアラートインスタンスが作成されますが、1 週間後にさらに 10 台の EC2 インスタンスが追加された場合、次の評価時には 60 個のアラートインスタンスが存在することになります。

  アラートインスタンスの数は、多次元アラートを作成するときに評価され、アラートインスタンスのクォータをすぐに超えるインスタンスを作成することはできません。アラートインスタンスの数は変動する可能性があるため、ルール評価の都度クォータがチェックされます。
- 評価時に、ルールがアラートインスタンスのクォータを超える場合、アラートルールが更新されアラートインスタンスの合計数がサービスクォータを下回るまで、そのルールは評価されません。この場合、クォータに達したことを通知するアラート通知を受け取ります (通知は評価対象のルールの通知ポリシーを基に送信されます)。通知には、`QuotaReachedError` 値を含む `Error` 注釈が含まれています。
- `QuotaReachedError` を引き起こすルールは、評価されなくなります。評価は、更新が行われ、更新後の評価が `QuotaReachedError` を引き起こさなくなった場合にのみ再開されます。評価されていないルールは、Grafana コンソールにクォータ到達エラーが表示されます。
- アラートインスタンスの数を減らすには、アラートルールを削除するか、複数次元のアラートを編集してアラートインスタンスを減らします (例えば、仮想マシン内の API ごとに 1 個のアラートではなく、仮想マシンごとの 1 件のエラーに対して 1 個のアラートを持つようにします)。
- 評価を再開するには、アラートを更新して保存します。アラートインスタンスの数を減らすように更新することもできますし、インスタンス数を減らすために他の変更を加えた場合、変更なしで保存しても構いません。再開可能な場合、評価が再開されます。さらに別の `QuotaReachedError` を引き起こす結果となった場合、保存することはできません。

- アラートが保存され、アラートクォータを超えることなく評価が再開された場合、Grafana コンソールにはしばらくの間クォータ到達エラーが (次の評価間隔まで) 表示され続けることがありますが、アラートルール評価が開始され、ルールのしきい値に達するとアラートが送信されます。
- アラートのクォータおよびその他のクォータの詳細については、「Amazon Managed Grafana のサービスクォータ」を参照してください。

データがない場合やエラーが発生した場合の対処

データがない場合やエラーが発生した場合のアラート動作の処理方法を選択します。

データがない場合に選択できる処理方法を、以下の表に照会します。

データがない場合の選択肢	行動
データなし	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を使用したアラート `DatasourceNoData` を作成します。
[アラート]	アラートルールの状態を `Alerting` に設定します。
OK	アラートルールの状態を `Normal` に設定します。

エラーが発生した場合に選択できる処理方法を、以下の表に照会します。

エラーまたはタイムアウト時の選択肢	行動
[アラート]	アラートルールの状態を `Alerting` に設定します
OK	アラートルールの状態を `Normal` に設定します
エラー	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を

エラーまたはタイムアウト時の選択肢	行動
	使用したアラート `DatasourceError` を作成します。

Grafana Mimir または Loki 管理のアラートルールの作成

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Grafana では、外部の Grafana Mimir または Loki インスタンスのアラートルールを作成することができます。

> **Note**
>
> Grafana Mimir は、Amazon Managed Service for Prometheus および Prometheus データソースに接続できます。

前提条件

- Prometheus データソースへの書き込みアクセス権限があることを確認します。アクセス権限がない場合、Cortex 管理のアラートルールを作成または更新することはできません。
- Grafana Mimir および Loki データソースの場合、それぞれのサービスを設定して Ruler API を有効にします。
  - Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集する場合、他のストレージタイプを設定してください。
  - Grafana Mimir – `/prometheus` ではなく、レガシー `/api/prom` プレフィックスを使用します。Prometheus データソースは Grafana Mimir と Prometheus の両方に対応してお

り、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを想定しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[アラート UI 経由でアラートを管理する] チェックボックスをオフにします。

Grafana Mimir または Loki 管理のアラートルールを追加するには

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [アラートルールの作成] を選択します。
3. ステップ 1 で、次のようにルールタイプと詳細を選択します。
   - Mimir または Loki アラートを選択します。
   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [データソース]の選択ドロップダウンから、Prometheus または Loki データソースを選択します。
   - [名前空間] ドロップダウンから、既存のルール名前空間を選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。名前空間には 1 つ以上のルールグループを含めることができます。これは、組織的な目的を持たせるためのみに使用します。詳細については、「Cortex または Loki ルールグループと名前空間」を参照してください。
   - [グループ] ドロップダウンから、選択した名前空間内の既存のグループを選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。新しく作成されたルールは、グループの末尾に追加されます。グループ内のルールは、同じ評価時間で一定の間隔で順番に実行されます。
4. ステップ 2 で、評価するクエリを追加します。

   値は PromQL または LogQL 式で指定することができます。評価結果に 0 より大きい値を持つシリーズが 1 つ以上ある場合、ルールはアラートを発行します。アラートはシリーズごとに作成されます。
5. ステップ 3 で、アラート評価間隔を指定します。

条件の [For] テキストボックスに、アラートが発行されるために必要な条件の継続時間を指定します。例えば 5m を指定した場合、条件が 5 分間継続した場合にアラートが発行されます。

> **Note**
>
> 条件が満たされると、アラートは Pending 状態に入り、そのまま指定された期間条件が継続するとアラートは Firing 状態に移行します。条件から外れた場合、Normal 状態に戻ります。

6. ステップ 4 で、ルールに関連するメタデータを追加します。
   - アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「ラベルと注釈」を参照してください。
   - Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
   - カスタムラベルを追加します。
7. [アラートのプレビュー] を選択してルールを評価し、生成されるアラートを確認します。各アラートの状態と値を含むアラートのリストが表示されます。
8. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

ルールを作成したら、ルールの通知を作成できます。通知の詳細については、「アラート通知の管理」を参照してください。

Grafana Mimir または Loki 管理の記録ルールの作成

> **⚠** このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

外部 Grafana Mimir または Loki インスタンスの記録ルールを作成および管理することができます。記録ルールは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列

セットとして保存します。この新しい時系列のクエリは、特にダッシュボードでは、ダッシュボードが更新されるたびに同じ式をクエリするため、より高速になります。

前提条件

Grafana Mimir および Loki データソースの場合、それぞれのサービスを設定して Ruler API を有効にします。

- Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集する場合、他のストレージタイプを設定してください。
- Grafana Mimir – Grafana Mimir を指定するようにデータソースを設定するときは、`/prometheus` ではなくレガシー `/api/prom` プレフィックスを使用します。Prometheus データソースは Grafana Mimir と Prometheus の両方に対応しており、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを想定しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[アラート UI 経由でアラートを管理する] チェックボックスをオフにします。

Grafana Mimir または Loki 管理の記録ルールを追加するには

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [アラートルールの作成] を選択します。
3. ステップ 1 で、以下を参照してタイプ、ルール名、ストレージの場所を追加します。
   - Mimir または Loki の記録ルールオプションを選択します。
   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [データソース]の選択ドロップダウンから、Prometheus または Loki データソースを選択します。
   - [名前空間] ドロップダウンから、既存のルール名前空間を選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。名前空間には 1 つ以上のルールグループを含め

ることができます。これは、組織的な目的を持たせるためのみに使用します。詳細については、「Cortex または Loki ルールグループと名前空間」を参照してください。

- [グループ] ドロップダウンから、選択した名前空間内の既存のグループを選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。新しく作成されたルールは、グループの末尾に追加されます。グループ内のルールは、同じ評価時間で一定の間隔で順番に実行されます。

4. ステップ 2 で、評価するクエリを追加します。

   値は PromQL または LogQL 式で指定することができます。評価結果に 0 より大きい値を持つシリーズが 1 つ以上ある場合、ルールはアラートを発行します。アラートはシリーズごとに作成されます。

5. ステップ 3 で、ルールに関連するメタデータを追加します。

   - アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「アラートルールの注釈とラベル」を参照してください。
   - Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
   - カスタムラベルを追加します。

6. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

Grafana Mimir または Loki ルールグループと名前空間

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ルールは整理することができます。ルールはルールグループ内に作成され、ルールグループは名前空間に整理されます。ルールグループ内のルールは、一定の間隔で順次実行されます。デフォルトの間隔は 1 分です。また、Grafana Mimir または Loki の名前空間とルールグループの名前を変更したり、ルールグループの評価間隔を編集することができます。

ルールグループまたは名前空間の編集方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. 編集するルールグループまたは名前空間内のルールに移動します。
3. [編集] (ペン) アイコンを選択します。
4. ルールグループまたは名前空間に変更を加えます。

> **Note**
>
> 名前空間では、名前のみを編集できます。ルールグループの場合、名前、またはグループ内のルールの評価間隔を変更します。たとえば、ルールを 1 分ごとに評価する場合は `1m` を選択し、30 秒ごとに評価する場合は `30s` を選択します。

5. [変更の保存] をクリックします。

アラートルールの表示と編集

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

[アラート] ページには、アラートルールが一覧表示されます。デフォルトでは、ルールはデータソースの種類ごとにグループ化されます。Grafana セクションには Grafana によって管理されるルールが一覧表示され、Cortex/Loki セクションには Prometheus 互換データソースのルールが一覧表示されます。Prometheus 互換データソースのアラートルールは表示できますが、編集することはできません。

Mimir/Cortex/Loki ルールセクションには、Mimir、Cortex、または Loki データソースのすべてのルールが一覧表示されます。クラウドアラートルールもこのセクションに表示されます。

大量のアラートを管理する場合、拡張アラートルール検索機能を使用して、フォルダ、評価グループ、ルールをフィルタリングできます。さらに、ラベル、状態、タイプ、正常性などのプロパティでアラートルールをフィルタリングできます。

> **Note**
>
> プロビジョニングされたアラートのクエリ定義は表示することはできますが、編集はできません。これらを表示できるため、ルール定義のプロビジョニングリポジトリに戻ることなく、クエリとルール定義が正しいことを確認できます。

アラートルールの表示

Grafana アラートを使用すると、すべてのアラートを 1 ページに表示できます。

アラートの詳細を表示するには

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。デフォルトでは、ルールはデータソースの種類ごとにグループ化して表示されます。各アラートの現在の状態別に表示することもできます (詳細については後述します)。
2. [表示形式] では、グループビューと状態ビューを切り替えることができます。
3. 行の横にある矢印を選択すると、その行の詳細が表示されます。ルールの詳細には、ルールのラベル、注釈、データソース、クエリ、およびルールから生成されたアラートインスタンスのリストが含まれます。

> **Note**
>
> アラートの詳細については、「アラートルールの状態と正常性」を参照してください。

グループビュー

グループビューには、フォルダ別にグループ化された Grafana アラートルールと、`namespace + group` 別にグループ化された Loki または Prometheus アラートルールが表示されます。これは、ルールの管理を目的としたデフォルトのルールリストビューです。各グループを展開すると、そのグループ内のルールのリストが表示されます。さらにルールを展開すると、その詳細を見ることができます。また、ルールから生成されたアクションボタンとアラートを展開して、詳細を表示することもできます。

状態ビュー

状態ビューには、アラートルールが状態ごとグループ化されて表示されます。このビューを使用すると、各ルールがどの状態にあるかを一目で確認できます。各ルールは展開して詳細を表示することができます。アクションボタンとこのルールによって生成されたアラート、および各アラートをさらに展開して詳細を表示することができます。

アラートルールのフィルター

[アラート] ページに表示されるアラートルールは、いくつかの方法でフィルタリングできます。

- 特定のデータソースを使用するルールだけを表示するには [データソースの選択] でフィルタリングするデータソースを選択します。
- また、[ラベルで検索] から検索条件を選択して、ラベルでフィルタリングすることもできます。例えば、`environment=production,region=~US|EU,severity!=warning` と入力して、米国と欧州の本番稼働警告をフィルタリングできます。
- 特定の状態のルールだけを表示するには [状態でアラートをフィルタリング] で表示する状態を選択します。

アラートルールの編集または削除

Grafana管理のアラートルールの編集または削除は、ルールが保存されているフォルダに対する編集権限を持つユーザーのみが行うことができます。外部 Mimir または Loki インスタンスのアラートルールの編集または削除は、エディタまたは管理者ロールを持つユーザーが行うことができます。

ルールを編集または削除するには

1. 表示、編集、削除のルールコントロールが表示されるまでルールを展開します。
2. [編集] を選択してルールの作成ページを開きます。更新方法はルール作成方法と同じです。詳細については、「Grafana 管理のアラートルールの作成」または「Grafana Mimir または Loki 管理のアラートルールの作成」の手順を参照してください。
3. 削除する場合は [削除] を選択してルールを削除します。

アラートルールのエクスポート

Export を選択すると、Grafana ワークスペースの YAML または JSON にルールを[エクスポート]できます。これにより、新しいルールを定義してからエクスポートするオプションが提供されます。UI

を使用してルールを作成し、プロビジョニング API または terraform スクリプトで使用するために エクスポートできます。

> **Note**
>
> これは、Grafana ワークスペースとプロビジョニングインターフェイスの両方でサポートされています。

アラート通知の管理

> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラート通知を送信する方法、タイミング、および場所を選択することは、アラートシステムを設定する上で重要です。これらの決定は、問題を迅速に解決し、重要なものを見逃さない能力に直接影響します。

最初のステップとして、アラート通知の送信先となるコンタクトポイントを定義します。コンタクトポイントでは、一致する通知の宛先のセットを指定できます。通知テンプレートをコンタクトポイントに追加して、通知の再利用と一貫したメッセージングを行います。

次に、アラートがコンタクトポイントにルーティングされる場所、タイミング、方法に関する一連のルールである通知ポリシーを作成します。通知ポリシーでは、作成したコンタクトポイントのいずれかを選択して、アラート通知を送信する場所を定義します。通知ポリシーにミュートタイミングを追加します。ミュートタイミングは、あらゆる通知を送信しない定期的な時間間隔です。

アラートルールが評価されると、アラートルーラーはアラートインスタンスをアラートマネージャーに送信します。1 つのアラートルールで複数の個別のアラートインスタンスをトリガーできます。

アラートマネージャーは、これらのアラートインスタンスを受信し、ミュートタイミングを処理し、アラートをグループ化し、通知ポリシーで定義されているようにコンタクトポイントに通知を送信します。

トピック

- アラートマネージャー
- コンタクトポイント (通知先) の使用
- 通知ポリシーの使用
- 通知のカスタマイズ
- Prometheus データソースのアラート通知のサイレンス化
- ミュートタイミング
- アラートグループ別に表示およびフィルタリングする
- 通知エラーの表示

アラートマネージャー

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートマネージャーを使用すると、アラートを迅速かつ効率的に管理して対応できます。E メールや Slack など選択したチャネルを介して通知を送信することで、アラートを受信し、ミュート、抑制、グループ化、ルーティングを処理します。

Grafana では、Grafana アラートマネージャー、または外部アラートマネージャーを使用できます。複数のアラートマネージャーを実行することもできます。決定は、セットアップとアラートの生成場所によって異なります。

Grafana アラートマネージャー

Grafana アラートマネージャーは、事前設定済みで、Grafana をオンプレミスまたはオープンソースで実行している場合にデフォルトで選択できる内部アラートマネージャーです。

Grafana アラートマネージャーは Grafana からアラートを受信できますが、Mimir や Loki など、Grafana の外部からアラートを受信することはできません。

> **Note**
>
> 禁止ルールは Grafana アラートマネージャーではサポートされていません。

外部アラートマネージャー

Grafana、Loki、Mimir、Prometheus のすべてのアラートを受信するために 1 つのアラートマネージャーを使用したい場合は、外部のアラートマネージャーを使用するように Grafana を設定できます。この外部アラートマネージャーは、Grafana 内から設定および管理できます。

Grafana アラートマネージャーの代わりに、独自の外部アラートマネージャーを設定し、そこでアラートを送信する場合の例を 2 つ示します。

1. Prometheus などの他のアラートジェネレーターがあるため、独自のクラウドインフラストラクチャに既にオンプレミスのアラートマネージャーを設定しており、引き続き使用したい場合。
2. Prometheus オンプレミスとホストされた Grafana の両方を使用して、クラウドインフラストラクチャで実行されるのと同じアラートマネージャーにアラートを送信したい場合。

アラートマネージャーは、[コンタクトポイントのアラート] および [通知ポリシー] ページのドロップダウンメニューに表示されます。

データソースをプロビジョニングする場合は、`jsonData` フィールドの `handleGrafanaManagedAlerts` フラグを `true` に設定して、Grafana が管理するアラートをこのアラートマネージャーに送信します。

コンタクトポイント (通知先) の使用

> **Warning**
>
> このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

連絡窓口は、アラート発生時の通知の受け取り方の設定に使用します。コンタクトポイントには、Amazon Simple Notification Service や Slack など、1 つ以上のコンタクトポイント統合を指定で

きます。アラートが発生すると、コンタクトポイントにリストされているすべてのコンタクトポイント統合に通知が送信されます。必要に応じて、通知テンプレートを使用して、コンタクトポイントタイプの通知メッセージをカスタマイズできます。

> **Note**
>
> Grafana で管理されるアラートの通知先 (コンタクトポイント) は、作成および編集が可能です。アラートマネージャーアラートの通知先 (コンタクトポイント) は読み取り専用です。

コンタクトポイント (通知先) の使用

次の手順では、通知先 (コンタクトポイント) を追加、編集、テスト、削除する方法について説明します。

コンタクトポイントの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [コンタクトポイント] を選択し、[コンタクトポイントを追加] を選択します。
3. [アラートマネージャー] ドロップダウンから、アラートマネージャーを選択します。デフォルトでは、Grafana アラートマネージャーが選択されています。
4. コンタクトポイントの [名前] を入力します。
5. [コンタクトポイント統合] から、タイプを選択し、そのタイプに基づいて必須フィールドを選択します。例えば、Slack を選択した場合は、Slack チャネルと通知先となるユーザーを入力します。
6. 選択したコンタクトポイントで利用可能な場合は、任意のオプション設定を選択して追加の設定を指定します。
7. [通知設定]で[解決済みメッセージの無効化]を選択すると、アラートが解決した際に通知を受け取らないようにすることができます (任意)。
8. コンタクトポイントにコンタクトポイントタイプを追加したい場合は、[新しいコンタクトポイントタイプ] を選択し、必要なコンタクトポイントタイプごとに手順を繰り返します。
9. [コンタクトポイントの保存] を選択して変更を保存します。

コンタクトポイントの編集方法

1. [コンタクトポイント]を選択すると、現在設定されているコンタクトポイントのリストが表示されます。
2. 編集する連絡先を選択し、[編集] アイコン (ペン) を選択します。
3. 必要な変更を加え、[コンタクトポイントの保存] を選択して変更を保存します。

コンタクトポイントを作成したら、テスト通知を送信して、正しく設定されていることを確認します。

テスト通知の送信方法

1. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
2. テストする連絡先を選択し、[編集] アイコン (ペン) を選択します。
3. [テスト] アイコン (紙飛行機) を選択します。
4. 事前定義されたテスト通知を送信するか、[カスタム] を選択してテスト通知に独自の注釈とラベルを追加するかを選択します。
5. [テスト通知を送信]を選択して、指定されたコンタクトポイント宛にアラートをテストします。

使用されていないコンタクトポイントは、通知ポリシーで削除できます。

連絡先の削除方法

1. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
2. 削除する連絡先を選択し、[削除] アイコン (ゴミ箱) を選択します。
3. 確認ダイアログボックスで、[はい、削除] を選択します。

> **Note**
>
> コンタクトポイントが通知ポリシーで使用されている場合、コンタクトポイントを削除する前に、通知ポリシーを削除するか、別のコンタクトポイントを使用するように編集する必要があります。

サポートされている通知方法の一覧

名前	タイプ
Amazon SNS	`sns`
OpsGenie	`opsgenie`
Pager Duty	`pagerduty`
Slack	`slack`
VictorOps	`victorops`

通知ポリシーの使用

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知ポリシーでは、アラートがコンタクトポイント (通知先) にどのように配信されるかを指定します。ポリシーはツリー構造で、ポリシーごとに 1 つ以上の子ポリシーを持つことができます。ルートポリシーを除く各ポリシーは、特定のアラートラベルにも一致させることができます。各アラートはルートポリシーによって評価され、次に各子ポリシーによって評価されます。特定のポリシーに対して `Continue matching subsequent sibling nodes` オプションを有効にすると、1 つ以上の一致があっても評価が続行されます。子ポリシーのいずれとも一致しない場合、親ポリシーの設定とコンタクトポイント (通知先) の情報によってアラートの動作が制御されます。特定のポリシーと一致しない場合、ルートポリシーがアラートを制御します。

Note

Grafana 管理のアラートの通知ポリシーを作成および編集することができますが、アラートマネージャーアラートの通知ポリシーは読み取り専用となっています。

通知のグループ化

グループ化すると、似た性質のアラート通知を 1 つの funnel にまとめられます。これは、システム上で一度に複数の障害が発生して多数のアラートが同時に発生する場合、大規模な停止中にアラート通知を制御することができます。

グループ化の例

例えば、異なる環境でデータベースに接続されている100のサービスがあるとします。これらのサービスは、ラベル env=environmentname で区別され、サービスがデータベースに到達できるかどうかを監視するためのアラートルールが設定されています。このアラートルールは、alertname=DatabaseUnreachable という名前のアラートを作成します。

サービスの半分がデータベースに到達できなくなったネットワークの分断が発生すると、50 件のアラートが個別に発生しますが、影響を受けている環境のリストを含む (50 件ではなく) 1 ページの通知を受け取りたいと考えています。

group_by: [alertname]でグループ化を設定することができます(サービスごとに異なる env ラベルを使用する代わりに)。この設定を行うと、Grafana は、このアラートルールの影響を受けるすべての環境が記載された 1 通のコンパクトな通知を送信します。

特殊グループ

Grafana には 2 つの特殊グループがあります。デフォルトのグループである group_by: null は、すべてのアラートを 1 つのグループにグループ化します。... という名前の特殊ラベルを使用してすべてのラベルでアラートをグループ化することで、グループ化を無効にし、各アラートを独自のグループに送信することもできます。

通知の使用

以下に、通知ポリシーを作成および管理するための手順を紹介します。

ルート通知ポリシーの編集方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [ルートポリシー] セクションで、[編集] アイコン (ペン) を選択します。

5. [デフォルトのコンタクトポイント] で、アラートルールが特定のポリシーと一致しない場合に通知を送信するコンタクトポイント (通知先) を更新します。
6. [グループ化対象] で、アラートをグループ化するラベル (または特殊グループ) を選択します。
7. [タイミングオプション] で、以下のいずれかを選択します。
   - [グループ待機] – 同じグループのアラートをバッファリングしてから、最初の通知を送信するまでの待機時間。デフォルト値は 30 秒です。
   - [グループ間隔] – グループ内の 2 つの通知の最小時間間隔。デフォルト値は 5 分です。
   - [繰り返し間隔] – グループに新しいアラートが追加されなかった場合に、通知を再送信するまでの最小時間間隔。デフォルトは 4 時間です。
8. [保存] を選択して変更を保存します。

上位の特定のポリシーの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [特定のルーティング] セクションで、[新しい特定のポリシー] を選択します。
5. [一致するラベル] セクションで、一致するアラートラベルを 1 つ以上追加します。ラベルの一致の詳細については、このトピックの後半で説明します。
6. コンタクトポイントで、アラートがこの特定のポリシーと一致する場合に通知を送信するコンタクトポイントを追加します。ネストされたポリシーは、このコンタクトポイントを上書きします。
7. オプションで、アラートが現在のポリシーと一致した後でも、[後続の兄弟ノードを照合し続ける] を有効にして、兄弟ポリシーの照合を続行します。このオプションを有効にすると、同じアラートに対して複数の通知を取得できます。
8. 必要に応じて [グループ化を上書き] を選択して、ルートポリシーとは異なるグループ化を指定します。
9. 必要に応じて [一般的なタイミングを上書き] を選択して、グループ通知ポリシーで設定されているタイミングオプションを上書きします。
10. [ポリシーの保存] を選択して変更を保存します。

ネストしたポリシーの追加方法

1. ネストしたポリシーの作成先となる特定のポリシーを展開します。
2. [ネストしたポリシーの追加] を選択し、詳細を追加します (最上位の特定のポリシーを追加する場合と同様)。
3. [ポリシーの保存] を選択して変更を保存します。

特定のポリシーの編集方法

1. [アラート] ページから [通知ポリシー] を選択するとページを開き、現在設定されているポリシーが一覧表示されます。
2. 編集する設定ポリシーを選択したら、[編集] アイコン (ペン) を選択します。
3. 必要な変更を加えます (最上位の特定のポリシーを追加する場合と同様)。
4. [ポリシーを保存] を選択します。

ポリシーの検索

ポリシーのツリー内では、ラベルマッチャーまたはコンタクトポイントで検索できます。

- コンタクトポイントで検索するには、[コンタクトポイントで検索] フィールドにコンタクトポイントの部分的な名前または完全な名前を入力します。
- ラベルで検索するには、[ラベルで検索] 入力フィールドに有効なラベルマッチャーを入力します。複数のマッチャーをカンマで区切って入力できます。有効なマッチャー入力の例は `severity=high, region=~EMEA|NA` です。

> **Note**
>
> ラベルで検索する場合、一致するすべてのポリシーは完全一致になります。部分一致と正規表現形式の一致はサポートされていません。

ラベル一致の仕組み

ポリシーは、アラートのラベルがポリシーで指定されたすべての一致ラベルと合致する場合に、そのアラートに適用されます。

- ラベル – 一致させるラベルの名前。アラートのラベル名と完全に一致する必要があります。

- [演算子] – ラベル値と一致するラベル値の比較に使用される演算子。利用できる演算子は次のとおりです。
  - = 値が指定された文字列と完全に一致するラベルが選択されます。
  - != 値が指定された文字列と一致しないラベルが選択されます。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルが選択されます (指定された文字列は正規表現として解釈されます)。
  - != 指定された正規表現と一致しないラベルが選択されます。
- [値]) – ラベル値と一致する値。選択した演算子に応じて、文字列または正規表現として照合させることができます。

通知のカスタマイズ

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知テンプレートを使用して、通知をカスタマイズします。

通知テンプレートを使用して、通知内のメッセージのタイトル、メッセージ、形式を変更できます。

通知テンプレートは、E メールや Slack などの特定のコンタクトポイント統合には関連付けられていません。ただし、異なるコンタクトポイント統合用に個別の通知テンプレートを作成することもできます。

通知テンプレートを使用すると、次のことができます。

- 概要、説明、ラベルと注釈、値、リンクなど、通知内の情報を追加、削除、または並べ替える
- テキストを太字と斜体で書式設定し、改行を追加または削除する

通知テンプレートを次の目的には使用できません。

- Slack や Microsoft Teams などのインスタントメッセージングサービスで通知のデザインを変更する

トピック

- Go のテンプレート言語の使用
- 通知テンプレートの作成
- テンプレートリファレンス

Go のテンプレート言語の使用

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

Go のテンプレート言語である text/template で通知テンプレートを書き込みます。

このセクションでは、Go のテンプレート言語の概要と、text/template でのテンプレートの作成について説明します。

dot

text/template には dot と呼ばれる特別なカーソルがあり、. として記述されます。このカーソルは、テンプレート内の使用場所に応じて値が変化する変数と考えることができます。例えば、通知テンプレートの開始時に . は、`Alerts`、`Status`、`GroupLabels`、`CommonLabels`、`CommonAnnotations` および `ExternalURL` を含む多数のフィールドを含む `ExtendedData` オブジェクトを参照します。ただし、dot は、リスト上の `range` で使用される場合、`with` 内で使用される場合、または他のテンプレートで使用する機能テンプレートを記述する場合、他の何かを参照する場合があります。これの例は 通知テンプレートの作成 で、すべてのデータと関数は テンプレートリファレンス で確認できます。

開始と終了のタグ

テキスト/テンプレートでは、テンプレートが変数を印刷するか、if ステートメントなどのコントロール構造を実行するかに関係なく、テンプレートは {{で始まり、}} で終わります。これは、変数の出力に {{ と }} を使用し、制御構造に {% と %} を使用する Jinja などの他のテンプレート言語とは異なります。

印刷

何かの値を印刷するには、{{ と }} を使用します。dot の値、dot のフィールド、関数の結果、[変数]の値を印刷できます。例えば、dot が ExtendedData を参照する Alerts フィールドを印刷するには、次のように記述します。

```
{{ .Alerts }}
```

アラートを反復処理する

アラートに関するすべての情報ではなく、各アラートのラベルのみを印刷するには、range を使用して ExtendedData でアラートを反復処理できます。

```
{{ range .Alerts }}
{{ .Labels }}
{{ end }}
```

範囲内の dot は、ExtendedData ではなく Alert を参照します。{{ .Labels }} を使用して、各アラートのラベルを印刷できます。これは、アラートのリスト内の現在のアラートを参照するように {{ range .Alerts }} が dot を変えるために機能します。範囲が終了すると、dot は範囲の開始前に存在していた値にリセットされます。この例では ExtendedData です。

```
{{ range .Alerts }}
{{ .Labels }}
{{ end }}
{{/* does not work, .Labels does not exist here */}}
{{ .Labels }}
{{/* works, cursor was reset */}}
{{ .Status }}
```

注釈とラベルを反復処理する

各アラートのラベルを `The name of the label is $name, and the value is $value` の形式で印刷するテンプレートを書きましょう。ここで、`$name` と `$value` には各ラベルの名前と値が含まれます。

前の例と同様に、`.Alerts` の範囲を使用して のアラートを繰り返し、dot がアラートのリスト内の現在のアラートを参照するようにし、ソートされたラベルで 2 番目の範囲を使用して、dot が現在のラベルを参照するように 2 回目に更新されます。2 番目の範囲内では、 `.Name` と `.Value` を使用して各ラベルの名前と値を出力します。

```
{{ range .Alerts }}
{{ range .Labels.SortedPairs }}
The name of the label is {{ .Name }}, and the value is {{ .Value }}
{{ end }}
{{ range .Annotations.SortedPairs }}
The name of the annotation is {{ .Name }}, and the value is {{ .Value }}
{{ end }}
{{ end }}
```

If ステートメント

テンプレートでは if ステートメントを使用できます。例えば、`.Alerts` にアラートがない場合に `There are no alerts` を出力するには、次のように記述します。

```
{{ if .Alerts }}
There are alerts
{{ else }}
There are no alerts
{{ end }}
```

With

With は if ステートメントと似ていますが、if ステートメントとは異なり、`with` は dot を更新して with の値を参照します。

```
{{ with .Alerts }}
There are {{ len . }} alert(s)
{{ else }}
There are no alerts
{{ end }}
```

[変数]

text/template の変数は、テンプレート内に作成する必要があります。例えば、$variable という変数を現在の dot の値で作成するには、次のように記述します。

```
{{ $variable := . }}
```

範囲内の $variable または with を使用でき、これらは変数が定義された時点の dot の値を参照します。dot の現在の値は参照されません。

例えば、2 番目の範囲で {{ .Labels }} を使用するテンプレートを記述することはできません。ここでは、現在のアラートではなく、現在のラベルが dot で参照されるためです。

```
{{ range .Alerts }}
{{ range .Labels.SortedPairs }}
{{ .Name }} = {{ .Value }}
{{/* does not work because in the second range . is a label not an alert */}}
There are {{ len .Labels }}
{{ end }}
{{ end }}
```

これを修正するには、最初の範囲と 2 番目の範囲の前に $alert という変数を定義します。

```
{{ range .Alerts }}
{{ $alert := . }}
{{ range .Labels.SortedPairs }}
{{ .Name }} = {{ .Value }}
{{/* works because $alert refers to the value of dot inside the first range */}}
There are {{ len $alert.Labels }}
{{ end }}
{{ end }}
```

インデックスを含む範囲

範囲の開始時にインデックス変数と値変数を定義することで、範囲内の各アラートのインデックスを取得できます。

```
{{ $num_alerts := len .Alerts }}
{{ range $index, $alert := .Alerts }}
This is alert {{ $index }} out of {{ $num_alerts }}
{{ end }}
```

テンプレートの定義

define とテンプレートの名前を二重引用符で囲むことで、他のテンプレート内で使用できるテンプレートを定義できます。__subject、__text_values_list、__text_alert_list、default.title、default.messa などのデフォルトテンプレートを含め、他のテンプレートと同じ名前のテンプレートを設定しないでください。デフォルトテンプレートと同じ名前のテンプレート、または別の通知テンプレート内のテンプレートと同じ名前で作成されている場合、Grafana はいずれかのテンプレートを使用することになり、混乱が生じます。Grafana は、同じ名前のテンプレートが 2 つ以上ある場合、エラーメッセージを返しません。

```
{{ define "print_labels" }}
{{ end }}
```

テンプレートを埋め込む

template、二重引用符で囲まれたテンプレートの名前、およびテンプレートに渡されるカーソルを使用して、テンプレート内で定義されたテンプレートを埋め込むことができます。

```
{{ template "print_labels" . }}
```

テンプレートにデータを渡す

テンプレート内では、dot はテンプレートに渡される値を参照します。

例えば、テンプレートに発射アラートのリストが渡された場合、dot は発射アラートのリストを参照します。

```
{{ template "print_alerts" .Alerts }}
```

テンプレートにアラートのソート済みラベルが渡された場合、dot はソート済みラベルのリストを参照します。

```
{{ template "print_labels" .SortedLabels }}
```

これは、再利用可能なテンプレートを記述する場合に役立ちます。例えば、すべてのアラートを出力するには、次のように記述します。

```
{{ template "print_alerts" .Alerts }}
```

次に、発射アラートのみを出力するには、次のように記述します。

```
{{ template "print_alerts" .Alerts.Firing }}
```

これは、.Alerts と .Alerts.Firing の両方がアラートのリストであるために機能します。

```
{{ define "print_alerts" }}
{{ range . }}
{{ template "print_labels" .SortedLabels }}
{{ end }}
{{ end }}
```

コメント

{{/* および */}} を使用してコメントを追加できます。

```
{{/* This is a comment */}}
```

コメントが改行を追加しないようにするには、以下を使用します。

```
{{- /* This is a comment with no leading or trailing line breaks */ -}}
```

インデント

タブとスペースの両方のインデントと改行を使用して、テンプレートをより読みやすくすることができます。

```
{{ range .Alerts }}
  {{ range .Labels.SortedPairs }}
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

ただし、テンプレートのインデントはテキストにも適用されます。次に、削除する方法を説明します。

スペースと改行を削除する

text/template では、 {{- と -}} を使用して、先頭と末尾のスペースと改行を削除します。

例えば、インデントと改行を使用してテンプレートをより読みやすくする場合です。

```
{{ range .Alerts }}
  {{ range .Labels.SortedPairs }}
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

インデントと改行はテキストにも表示されます。

```
    alertname = "Test"

    grafana_folder = "Test alerts"
```

各範囲の先頭の }} を -}} に変更することで、テキストからインデントと改行を削除できます。

```
{{ range .Alerts -}}
  {{ range .Labels.SortedPairs -}}
    {{ .Name }} = {{ .Value }}
  {{ end }}
{{ end }}
```

これにより、テンプレートのインデントと改行がテキストから取り除かれます。

```
alertname = "Test"
grafana_folder = "Test alerts"
```

通知テンプレートの作成

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

コンタクトポイントに送信するため、再利用可能な通知テンプレートを作成できます。

通知テンプレートには 1 つ以上のテンプレートを追加できます。

通知テンプレート名は、一意である必要があります。同じ通知テンプレートまたは異なる通知テンプレートに、同じ名前のテンプレートを 2 つ使用することはできません。__subject、__text_values_list、__text_alert_list、default.title、default.messa などのデフォルトテンプレートと同じ名前のテンプレートを定義しないでください。

[コンタクトポイント] タブには、通知テンプレートのリストが表示されます。

通知テンプレートの作成

通知テンプレートを作成するには

1. [テンプレートの追加] をクリックします。
2. email.subject などの通知テンプレートの名前を選択します。
3. テンプレートの内容をコンテンツフィールドに書き込みます。

   例:

   ```
   {{ if .Alerts.Firing -}}
      {{ len .Alerts.Firing }} firing alerts
      {{ end }}
      {{ if .Alerts.Resolved -}}
      {{ len .Alerts.Resolved }} resolved alerts
      {{ end }}
   ```

4. 保存をクリックします。

   {{ define "email.subject" }} (email.subject はテンプレートの名前) と {{ end }} はコンテンツの先頭と末尾に自動的に追加されます。

複数のテンプレートを含む通知テンプレートを作成するには

1. [テンプレートの追加] をクリックします。
2. 通知テンプレート全体の名前を入力します。例えば、email と指定します。
3. 各テンプレートの最初と最後に {{ define "*name-of-template*" }} と {{ end }} を含む各テンプレートをコンテンツフィールドに書き込みます。email.subject や email.message など、通知テンプレート内の各テンプレートにわかりやすい名前を使用できます。この場合、上記で入力した通知テンプレートの名前は再度使用しないでください。

次のセクションでは、作成するテンプレートの詳細な例を示します。

4. 保存をクリックします。

E メールの件名テンプレートの作成

この例では、発射アラートと解決済みアラートの数を含む E メールの件名のテンプレートを作成します。

```
1 firing alerts, 0 resolved alerts
```

E メールの件名のテンプレートを作成するには

1. email.subject というテンプレートを次の内容で作成します。

   ```
   {{ define "email.subject" }}
   {{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
   {{ end }}
   ```

2. テンプレートは、キーワード template を使用して件名フィールドに配置して、コンタクトポイント統合を作成するときに使用します。

   ```
   {{ template "email.subject" . }}
   ```

E メールのメッセージ用のテンプレート作成

この例では、すべての発射アラートと解決済みアラートの概要を含む E メールのメッセージ用のテンプレートを作成します。

```
There are 2 firing alerts, and 1 resolved alerts

Firing alerts:

- alertname=Test 1 grafana_folder=GrafanaCloud has value(s) B=1
- alertname=Test 2 grafana_folder=GrafanaCloud has value(s) B=2

Resolved alerts:

- alertname=Test 3 grafana_folder=GrafanaCloud has value(s) B=0
```

E メールのメッセージ用のテンプレートを作成するには

1. email という名前の通知テンプレートを、コンテンツに email.message_alert と email.message という 2 つのテンプレートで作成します。

   email.message_alert テンプレートは、email.message テンプレートに E メールの構造が含まれている間、各発射アラートと解決済みアラートのラベルと値を印刷するために使用されます。

   ```
   {{- define "email.message_alert" -}}
   {{- range .Labels.SortedPairs }}{{ .Name }}={{ .Value }} {{ end }} has value(s)
   {{- range $k, $v := .Values }} {{ $k }}={{ $v }}{{ end }}
   {{- end -}}

   {{ define "email.message" }}
   There are {{ len .Alerts.Firing }} firing alerts, and {{ len .Alerts.Resolved }}
    resolved alerts

   {{ if .Alerts.Firing -}}
   Firing alerts:
   {{- range .Alerts.Firing }}
   - {{ template "email.message_alert" . }}
   {{- end }}
   {{- end }}

   {{ if .Alerts.Resolved -}}
   Resolved alerts:
   {{- range .Alerts.Resolved }}
   - {{ template "email.message_alert" . }}
   {{- end }}
   {{- end }}

   {{ end }}
   ```

2. テンプレートは、キーワード template を使用してテキスト本文フィールドに配置して、コンタクトポイント統合を作成するときに使用します。

   ```
   {{ template "email.message" . }}
   ```

Slack メッセージのタイトルのテンプレート作成

この例では、発射アラートと解決済みアラートの数を含む Slack メッセージのタイトルのテンプレートを作成します。

```
1 firing alerts, 0 resolved alerts
```

Slack メッセージのタイトルのテンプレートを作成するには

1. `slack.title` というテンプレートを次の内容で作成します。

   ```
   {{ define "slack.title" }}
   {{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
   {{ end }}
   ```

2. テンプレートは、キーワード `template` を使用してタイトルフィールドに配置して、コンタクトポイント統合を作成するときに使用します。

   ```
   {{ template "slack.title" . }}
   ```

Slack メッセージのコンテンツのテンプレート作成

すべての発射アラートと解決済みアラートの説明 (ラベル、注釈、ダッシュボード URL など) を含む Slack メッセージのコンテンツのテンプレートを作成します。

```
1 firing alerts:

[firing] Test1
Labels:
- alertname: Test1
- grafana_folder: GrafanaCloud
Annotations:
- description: This is a test alert
Go to dashboard: https://example.com/d/dlhdLqF4z?orgId=1

1 resolved alerts:

[firing] Test2
Labels:
- alertname: Test2
- grafana_folder: GrafanaCloud
```

```
Annotations:
- description: This is another test alert
Go to dashboard: https://example.com/d/dlhdLqF4z?orgId=1
```

Slack メッセージのコンテンツのテンプレートを作成するには

1. コンテンツに slack.print_alert とslack.message の 2 つのテンプレートを使用して、slack という名前のテンプレートを作成します。

   slack.print_alert テンプレートはラベル、注釈、および DashboardURL の印刷に使用され、slack.message テンプレートには通知の構造が含まれます。

```
{{ define "slack.print_alert" -}}
[{{.Status}}] {{ .Labels.alertname }}
Labels:
{{ range .Labels.SortedPairs -}}
- {{ .Name }}: {{ .Value }}
{{ end -}}
{{ if .Annotations -}}
Annotations:
{{ range .Annotations.SortedPairs -}}
- {{ .Name }}: {{ .Value }}
{{ end -}}
{{ end -}}
{{ if .DashboardURL -}}
  Go to dashboard: {{ .DashboardURL }}
{{- end }}
{{- end }}

{{ define "slack.message" -}}
{{ if .Alerts.Firing -}}
{{ len .Alerts.Firing }} firing alerts:
{{ range .Alerts.Firing }}
{{ template "slack.print_alert" . }}
{{ end -}}
{{ end }}
{{ if .Alerts.Resolved -}}
{{ len .Alerts.Resolved }} resolved alerts:
{{ range .Alerts.Resolved }}
{{ template "slack.print_alert" .}}
{{ end -}}
{{ end }}
```

```
{{- end }}
```

2. テンプレートは、キーワード `template` を使用してテキスト本文フィールドに配置して、コンタクトポイント統合を作成するときに使用します。

```
{{ template "slack.message" . }}
```

共有テンプレートを使用して E メールと Slack の両方にテンプレートを作成する

E メールや Slack など、コンタクトポイントごとに個別の通知テンプレートを作成する代わりに、同じテンプレートを共有できます。

例えば、この件名の E メールを送信し、このタイトル `1 firing alerts, 0 resolved alerts` の Slack メッセージを送信する場合は、共有テンプレートを作成できます。

共有テンプレートを作成するには

1. `common.subject_title` というテンプレートを次の内容で作成します。

```
{{ define "common.subject_title" }}
{{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
{{ end }}
```

2. E メールの場合は、E メールコンタクトポイント統合の件名フィールドからテンプレートを実行します。

```
{{ template "common.subject_title" . }}
```

3. Slack の場合は、Slack コンタクトポイント統合のタイトルフィールドからテンプレートを実行します。

```
{{ template "common.subject_title" . }}
```

通知テンプレートの使用

コンタクトポイントのテンプレートを使用して通知をカスタマイズします。

コンタクトポイントの作成時にテンプレートを使用するには

1. [アラート]メニューから、[コンタクトポイント] を選択して、既存のコンタクトポイントのリストを表示します。
2. [コンタクトポイントを追加] を選択します。または、編集するコンタクトポイントの横にある [編集] アイコン (ペン) を選択して、既存のコンタクトポイントを編集することもできます。
3. [Message] (メッセージ) や [Subject] (件名) などの 1 つ以上のフィールドに、使用するテンプレートを入力します。テンプレートを入力するには、`{{ template "template_name" . }}` フォームを使用し、 *template_name* を使用するテンプレートの名前に置き換えます。
4. [コンタクトポイントの保存] をクリックします。

テンプレートリファレンス

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

このセクションでは、テンプレートを作成するためのリファレンス情報を提供します。

テンプレートのデータ

メッセージテンプレートには、次のデータが渡されます。

名前	型	メモ
`Receiver`	文字列	コンタクトポイント (通知先) の名前。
`Status`	文字列	少なくとも 1 つのアラートが発生している場合は「発生中」となり、それ以外の場合「解決済」となります。

名前	型	メモ
Alerts	アラート	この通知に含まれるアラートオブジェクトのリスト (以下を参照)。
GroupLabels	KeyValue	これらのアラートがグループ化されたラベル。
CommonLabels	KeyValue	この通知に含まれるすべてのアラートに共通のラベル。
CommonAnnotations	KeyValue	この通知に含まれるすべてのアラートに共通する注釈。
ExternalURL	文字列	通知の送信元である Grafana へのリンク。外部のアラートマネージャーを使用している場合、そのアラートマネージャーへのリンク。

Alerts 型には、返されたアラートをフィルタリングするための 2 つの関数があります。

- Alerts.Firing – 発生中のアラートのリストを返します。
- Alerts.Resolved – 解決済みのアラートのリストを返します。

アラート (型)

アラート型には、次のデータがあります。

名前	型	メモ
ステータス	文字列	firing または resolved
ラベル	KeyValue	アラートに付与された一連のラベル。

名前	型	メモ
注釈	KeyValue	アラートに付与された一連の注釈。
値	KeyValue	クラシック条件を含むすべての式の値
StartsAt	time.Time	アラートが発行された時刻。
EndsAt	time.Time	アラートの終了時刻がわかっている場合にのみ設定されます。それ以外の場合、最後にアラートを受信してから設定可能なタイムアウト期間が適用されます。
GeneratorURL	文字列	Grafana または外部アラートマネージャーへのリンク。
SilenceURL	文字列	アラートをサイレンスするリンク (このアラートのラベルがあらかじめ入力されています)。Grafana 管理のアラートのみ。
DashboardURL	文字列	Grafana ダッシュボードへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。
PanelURL	文字列	Grafana ダッシュボードパネルへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。

名前	型	メモ
Fingerprint	文字列	アラートの識別に使用するフィンガープリント。
ValueString	文字列	アラート内の各削減された式のラベルと値を含む文字列。

ExtendedData

ExtendedData オブジェクトには、以下のプロパティが含まれています。

名前	[Kind] (種類)	説明	例
レシーバー	`string`	通知を送信するコンタクトポイントの名前。	`{{ .Receiver }}`
ステータス	`string`	ステータスは`firing if at least one alert is firing, otherwise resolved.`です。	`{{ .Status }}`
アラート	`[]Alert`	この通知のすべての発生アラートと解決済みアラートのリスト。	`There are {{ len .Alerts }} alerts`
発生アラート	`[]Alert`	この通知のすべての発生アラートのリスト。	`There are {{ len .Alerts.Firing }} firing alerts`
解決済みアラート	`[]Alert`	この通知で解決されたすべてのアラートのリスト。	`There are {{ len .Alerts.R`

名前	[Kind] (種類)	説明	例
			`esolved }} resolved alerts`
GroupLabels	`KeyValue`	これらのアラートをこの通知にグループ化するラベル。	`{{ .GroupLabels }}`
CommonLabels	`KeyValue`	この通知のすべてのアラートに共通するラベル。	`{{ .CommonLabels }}`
CommonAnnotations	`KeyValue`	この通知のすべてのアラートに共通する注釈。	`{{ .CommonAnnotations }}`
ExternalURL	`string`	この通知を送信した Grafana ワークスペースまたはアラートマネージャーへのリンク。	`{{ .ExternalURL }}`

KeyValue 型

`KeyValue` 型は、キー (ラベル) 値 (注釈) の文字列を組み合わせたものです。

`KeyValue` として保存されたデータに直接アクセスするメソッドだけでなく、データをソート、削除、変換するメソッドも備えています。

名前	引数	戻り値	メモ	例
SortedPairs		キーと値の文字列のソート済みのリスト		`{{ .Annotations.SortedPairs }}`
削除	[]string	KeyValue	指定したキーを除いたキー/値の	`{{ .Annotations.Remove "summary" }}`

名前	引数	戻り値	メモ	例
			マップのコピーを返します。	
名前		[]string	ラベル名のリスト	`{{ .Names }}`
値		[]string	ラベル値のリスト	`{{ .Values }}`

[時間]

時間は Go `time` パッケージからのものです。時間はさまざまな形式で印刷できます。例えば、アラートが発生した時刻を `Monday, 1st January 2022 at 10:00AM` 形式で出力するには、次のテンプレートを作成します。

```
{{ .StartsAt.Format "Monday, 2 January 2006 at 3:04PM" }}
```

Go の時間形式に関するリファレンスは、こちらを参照してください。

テンプレート関数

テンプレート関数を使用すると、ラベルと注釈を処理して動的に通知を生成できます。以下の機能を使用できます。

名前	引数の型	戻り型	説明
`humanize`	数値または文字列	文字列	メトリクスプレフィックスを使用して、数値をより読み取り可能な形式に変換します。
`humanize1024`	数値または文字列	文字列	humanize と同様に、1000 ではなく 1024 を基準にします。

名前	引数の型	戻り型	説明
humanizeDuration	数値または文字列	文字列	時間を秒単位のより読みやすい形式に変換します。
humanizePercentage	数値または文字列	文字列	比率の値を 100 分の 1 の形に変換します。
humanizeTimestamp	数値または文字列	文字列	Unix のタイムスタンプを秒単位のより読みやすい形式に変換します。
title	文字列	文字列	strings.Title を使用すると、各単語の最初の文字が大文字になります。
toUpper	文字列	文字列	strings.ToUpper を使用すると、文字列がすべて大文字に変換されます。
toLower	文字列	文字列	strings.ToLower を使用すると、文字列がすべて小文字に変換されます。
match	パターン、テキスト	ブール値	regexp.MatchString を使用すると、アンカーなしでの正規表現の一致テストを実行できます。

名前	引数の型	戻り型	説明
reReplaceAll	パターン、置換、テキスト	文字列	Regexp.ReplaceAllString を使用すると、アンカーなしでの正規表現による置換が実行できます。
graphLink	文字列 - expr および datasource フィールドを持つ JSON オブジェクト	文字列	指定された式とデータソースに対するグラフィカルビューへのパスを返します。
tableLink	文字列 - expr および datasource フィールドを持つ JSON オブジェクト	文字列	指定された式とデータソースに対するタブ形式ビューへのパスを返します。
args	[]interface{}	map[string]interface{}	オブジェクトのリストを arg0、arg1 などのキーを持つマップに変換します。複数の引数をテンプレートに渡す場合、この関数を使用します。
externalURL	なし	文字列	外部 URL を表す文字列を返します。
pathPrefix	なし	文字列	外部 URL のパスを返します。

次の表に、各関数の使用例を紹介します。

関数	TemplateString	Input	予想
humanize	{ humanize $value }	1234567.0	1.235M

関数	TemplateString	Input	予想
humanize1024	{ humanize1024 $value }	1048576.0	1Mi
humanizeDuration	{ humanizeDuration $value }	899.99	14 分 59 秒
humanizePercentage	{ humanizeP ercentage $value }	0.1234567	12.35%
humanizeTimestamp	{ humanizeTimestamp $value }	1435065584.128	2015-06-23 13:19:44. 128 +0000 UTC
title	{ $value | title }	aa bB CC	Aa Bb Cc
toUpper	{ $value | toUpper }	aa bB CC	AA BB CC
toLower	{ $value | toLower }	aa bB CC	aa bb cc
match	{ match "a+" $labels.i nstance }	aa	真
reReplaceAll	{{ reReplaceAll "localhost:(.*)" "my.domain:$1" $labels.instance }}	localhost:3000	my.domain:3000
graphLink	{{ graphLink "{\"expr\ ": \"up\", \"datasource \": \"gdev-prometheus \"}" }}		/explore?left=["no w-1h","now","gdev- prometheus",{"data source":"gdev-prom etheus","expr":"up ","instant":false, "range":true}]

関数	TemplateString	Input	予想
tableLink	{{ tableLink "{\"expr\":\"up\", \"datasource\":\"gdev-prometheus\"}" }}		/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-prometheus","expr":"up","instant":true,"range":false}]
args	{{define "x"}}{{.arg0}} {{.arg1}}{{end}}{{ template "x" (args 1 "2")}}		1 2
externalURL	{ externalURL }		http://localhost/path/prefix
pathPrefix	{ pathPrefix }		/path/prefix

Prometheus データソースのアラート通知のサイレンス化

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

外部のアラートマネージャーデータソース (Amazon Managed Service for Prometheus を含む) では、[サイレンス]機能を使用してアラート通知を停止できます。サイレンスは通知の作成を停止するだけで、アラートルールの評価やユーザーインターフェースへのアラートインスタンスの表示は引き続き行われます。アラートをサイレンスにする場合、停止する時間枠を指定します。

外部アラートマネージャーデータソースのサイレンスを設定できます。

> **Note**
>
> アラート通知を一定の時間間隔、またはその他のデータソース (定期的なメンテナンス期間中など) で停止するには、サイレンスではなく ミュートタイミング を使用します。

サイレンスの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [サイレンス] を選択して、既存の コンタクトポイント (通知先) の使用 を一覧表示するページを開きます。
3. [アラートマネージャー] ドロップダウンから外部アラートマネージャーを選択します。
4. [サイレンスの追加] を選択します。
5. [サイレンスの開始と終了] でサイレンスを有効にする開始日と終了日を選択します。

   終了時刻を設定する代わりに、[期間] でサイレンスが適用される期間を指定します。この方法を使用すると、[サイレンスの開始と終了] フィールドの終了時刻が自動的に更新されます。
6. [名前] および [値] のフィールドには、1 つ以上の一致するラベルを入力します。照合機能により、どのルールにサイレンスを適用するかが決定されます。ラベルの照合については、この手順の後に詳しく説明します。
7. 必要に応じて、コメントを追加したり、作成者を変更してサイレンスの所有者を設定することができます。
8. [作成] を選択して入力を作成します。

既存のサイレンスを編集するには、[編集] アイコン (ペン) を選択します。

アラート停止のラベル一致

サイレンスの作成時に、サイレンスの一部として一致するラベルのセットを作成します。これは、アラートが停止されるために一致する必要があるラベルに関するルールのセットです。一致するラベルは、次の 3 つの部分で構成されます。

- ラベル – 一致させるラベルの名前。アラートのラベル名と完全に一致する必要があります。
- [演算子] – ラベル値と一致するラベル値の比較に使用される演算子。利用できる演算子は次のとおりです。

  - = 値が指定された文字列と完全に一致するラベルが選択されます。
  - != 値が指定された文字列と一致しないラベルが選択されます。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルが選択されます (指定された文字列は正規表現として解釈されます)
  - != 指定された正規表現と一致しないラベルが選択されます。
- [値]) – ラベル値と一致する値。選択した演算子に応じて、文字列または正規表現として照合させることができます。

サイレンスは指定された終了日に終了しますが、いつでも手動で停止状態を解除することができます。

サイレンスを手動で終了する方法

1. [アラート] ページで、[サイレンス] を選択して既存のサイレンスのリストを表示します。
2. 終了するサイレンスを選択し、[サイレンス解除] を選択します。この操作によりアラートの停止状態は終了します。

> **Note**
>
> サイレンス解除を行うと、終了時刻が現在の時刻に設定されたものとして、アラートの停止状態が終了します。(自動または手動を問わず) 終了したサイレンスは5 日間保持され、一覧表示されます。リストからサイレンスを手動で削除することはできません。

サイレンス作成フォームへのリンクの作成

詳細が既に入力されているサイレンス作成フォームへの URL を作成できます。オペレータはこれを使用して、運用上のイベントを行う際にアラームをすばやく停止することができます。

サイレンスフォームへのリンクを作成するときは、`matchers` クエリパラメータを使用して一致するラベルを指定し、`comment` クエリパラメータを使用してコメントを指定します。`matchers` パラメータには、カンマで区切られた `[label][operator][value]` 形式の 1 つ以上の値が必要です。

URL の例

ラベル severity=critical と cluster!~europe-.* が一致し、Silencing critical EU alerts というコメントが付いたサイレンスフォームにリンクするには、次のような URL を使用します。*mygrafana* を Grafana インスタンスのホスト名に置き換えてください。

```
https://mygrafana/alerting/silence/new?matchers=severity%3Dcritical%2Ccluster!~europe-
*&comment=Silence%20critical%20EU%20alert
```

外部アラートマネージャーの新しいサイレンスページにリンクする場合、アラートマネージャーデータソース名を含む alertmanager クエリパラメータを追加します (例: alertmanager=myAlertmanagerdatasource)。

ミュートタイミング

> ⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

ミュートタイミングとは、ポリシーの新しい通知が生成されたり送信されないようにする繰り返しの時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間や繰り返し発生するのを防ぐために使用されます。

サイレンスと似ていますが、ミュートタイミングはアラートルールの評価や、ユーザーインターフェースへのアラート表示を停止せず、通知の作成のみを防ぎます。

外部のアラートマネージャーデータソースの Grafana 管理のミュートタイミングとミュートタイミングを設定できます。

サイレンスとミュートタイミングの比較

次の表に、「ミュートタイミング」と「サイレンス」の違いを示します。

ミュートタイミング	サイレンス
繰り返しの時間間隔を定義します。	開始時刻と終了時刻を固定します。

ミュートタイミング	サイレンス
作成され、通知ポリシーに追加されます。	ラベルを使用してアラートと照合し、サイレンスするかどうかが判断されます。
Grafana アラートおよび外部アラートマネージャーと連携します。	外部アラートマネージャーでのみ機能します。

ミュートタイミングの作成方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [ミュートタイミング] セクションで、[ミュートタイミングの追加] ボタンを選択します。
5. ミュートタイミングを適用したい時間間隔を選択します。
6. [送信] を選択してミュートタイミングを作成します。

通知ポリシーへのミュートタイミングの追加方法

1. ミュートタイミングを追加する通知ポリシーを選択し、[編集] ボタンを選択します。
2. [ミュートタイミング] ドロップダウンから、ポリシーに追加するミュートタイミングを選択します。

   [ポリシーを保存] ボタンを選択します。

時間間隔

時間間隔には、時間範囲を指定します。この時間間隔内にアラートが発生した場合、抑制されます。範囲には : を使用して表現することができます (例: `monday:thursday`)。ミュートタイミングには複数の時間間隔を含めることができます。時間間隔は複数のフィールド (詳細は次のリストを参照) で構成され、アラートを抑制するには、すべてのフィールドが一致する必要があります。例えば、曜日に月曜日から金曜日 `monday:friday`、時間範囲を 8:00 ~ 9:00 と指定した場合、アラートは月曜日から金曜日の 8:00 ~ 9:00 の間は抑制されますが、土曜日の 8:00 ~ 9:00 は抑制されません。

- [時間範囲] – 通知を抑制する時間帯。[開始時刻]と[終了時刻] の 2 つのサブフィールドで構成されます。時刻は 14:30 のように指定します。時刻は UTC の 24 時間表記です。
- [曜日] - 曜日。単一の日 (例: monday)、範囲 (例: monday:friday)、またはカンマで区切られた複数の日のリスト (例: monday, tuesday, wednesday) を指定できます。
- [月] - 選択する月。月は、数値で指定することも、完全な月名で指定することもできます。例えば、1 または january のどちらでも 1 月を指定できます。単一の月、月の範囲、またはカンマ区切りの月のリストを指定できます。
- [月の日付] – 1 か月内の日付。値は 1～31 の範囲で指定します。負の値は月の日を逆順で指定することを意味します。たとえば、-1 は月の最終日を表します。日付は単一の日、日付の範囲、またはカンマで区切った日付のリストを指定できます。
- [年] - 間隔の年または複数年。例えば、2023:2025。

これらの要素はそれぞれリストにすることができ、一致するには要素内の少なくとも 1 つの項目を満たす必要があります。したがって、年を 2023:2025, 2027 に設定すると、2023、2024、2025、2027 年 (2026 年を除く) に当てはまります。

フィールドを空白のままにすると、任意の時刻をフィールドと照合します。完全な時間間隔と一致させるには、ある時点がすべてのフィールドと一致している必要があります。

正確な期間を指定する場合は、その期間に必要なすべてのオプションを指定します。例えば、3 月、6 月、9 月、12 月の最初の月曜日の 12:00 から 24:00 UTC の時間間隔を作成する場合、時間間隔の指定は次のようになります。

- 時間範囲 :
  - 開始時間: 12:00
  - 終了時間: 24:00
- 曜日: monday
- 月: 3, 6, 9, 12
- 日: 1:7

アラートグループ別に表示およびフィルタリングする

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

アラートグループは、アラートマネージャーインスタンスからのグループ化されたアラートを表示します。デフォルトでは、アラートルールは通知ポリシーのルートポリシーのラベルキーによってグループ化されます。共通のアラートルールを 1 つのアラートグループにグループ化することで、重複したアラートルールが発生することを防ぎます。

アラートグループを表示し、特定の条件に一致するアラートルールをフィルター処理することもできます。

アラートグループを表示するには

1. Grafana メニューで、[アラート] (ベル) アイコンをクリックしてアラートページを開いて既存のアラートを一覧表示します。
2. [アラートグループ] をクリックして、既存のグループを一覧表示するページを開きます。
3. [アラートマネージャー] ドロップダウンから、データソースとして外部アラートマネージャーを選択します。
4. [カスタムグループ化] ドロップダウンでラベルの組み合わせを選択して、デフォルト以外のグループを表示します。これは、通知ポリシーのグループをデバッグおよび検証するのに役立ちます。

アラートにルートポリシーのグループ化またはカスタムグループ化のいずれかで指定されたラベルが含まれていない場合、アラートは `No grouping` のヘッダーを持つ [すべてのグループを取得] に追加されます。

ラベルでフィルタリングするには

- [検索] で既存のラベルを入力して、ラベルに一致するアラートを表示します。

  例えば、`environment=production,region=~US|EU,severity!=warning` と指定します。

状態でフィルタリングするには

- [状態] では、Active (アクティブ)、Suppressed (抑制)、または Unprocessed (未処理) の状態を選択すると、選択した状態に一致するアラートが表示されます。他のすべてのアラートは非表示になります。

通知エラーの表示

⚠ このドキュメントのトピックは、Grafana バージョン 9.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 8 での作業」を参照してください。

通知エラーを表示し、送信に失敗したか受信されなかった理由を確認できます。

Note

この機能は Grafana アラートマネージャーでのみサポートされています。

通知エラーを表示するには

1. Grafana メニューで、[アラート] (ベル) アイコンをクリックしてアラートページを開いて既存のアラートを一覧表示します。
2. [コンタクトポイント] を選択すると、現在設定されているコンタクトポイントのリストが表示されます。

   いずれかのコンタクトポイントが失敗した場合は、画面の右隅にメッセージが表示され、エラーが発生していることとその数がユーザーに通知されます。
3. コンタクトポイントをクリックすると、そのコンタクトポイントのエラーの詳細が表示されます。

   エラーアイコンにカーソルを合わせると、エラーの詳細が表示されます。

   コンタクトポイントに複数の統合がある場合、各統合のすべてのエラーが表示されます。

4. [正常性] 列で、通知のステータスを確認します。

   これは、[OK]、[試行なし]、または [エラー]のいずれかです。

Grafana バージョン 8 での作業

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana ワークスペースを作成するときに、使用する Grafana のバージョンを選択できます。以下のトピックでは、Grafana バージョン 8 を使用する Grafana ワークスペースの使用について説明します。

トピック

- パネル
- ダッシュボード
- Explore
- Linking (リンク中)
- テンプレートおよび変数
- Grafana アラート

パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

パネルは Grafana サーバーの基本的な視覚化構築ブロックです。いくつかの特殊用途パネルを除き、パネルは 1 つ以上のクエリの視覚的表現です。クエリは時間経過に伴うデータを表示します。これは、温度変動から、ログまたはアラートのリストに対する現在のサーバーステータスに至るまで多岐にわたります。

各パネルには、パネルで選択されたデータソース特有のクエリエディタがあります。クエリエディタを使用するとパネルに表示する視覚化を抽出できます。

データを表示するためには、ワークスペースに少なくとも 1 つのデータソースを追加する必要があります。データソースの詳細については、「データソースに接続する」を参照してください。

各パネルにはさまざまなスタイルとフォーマットのオプションがあります。パネルはドラッグ、配置変更、サイズ変更ができます。

トピック

- パネルの追加
- パネルの削除
- クエリ
- 記録済みのクエリ
- 変換
- フィールドオプションとオーバーライド
- パネルエディタ
- ライブラリパネル
- 視覚化

パネルの追加

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

パネルを使用してデータを視覚的に表示できます。このトピックでは、パネルを作成する基本的なステップについて説明します。

ダッシュボードへのグラフの追加方法

1. パネルを追加するダッシュボードを選択します。
2. [パネルを追加] アイコンを選択します。
3. [新しいラベルを追加] を選択します。

   Grafana ワークスペース上に、空のグラフパネルがデフォルトのデータソースが選択された状態で作成されます。
4. 必須ではありませんが、パネルに分かりやすいタイトルと説明を追加することをお勧めします。また、タイトルや説明には、いずれかのフィールドで定義済みの変数を使用することができますが、グローバル変数は使用できません。詳細については、「テンプレートおよび変数」を参照してください。

   - パネルタイトル – このフィールドに入力したテキストは、パネルエディタとダッシュボードのパネルの上部に表示されます。
   - 説明 – このフィールドに入力したテキストは、パネルの左上にあるツールチップに表示されます。表示するデータとパネルの説明を記入します。
5. パネルのクエリを書き込みます。視覚的表現を表示するには、各パネルに少なくとも 1 つのクエリが必要です。パネルエディタのクエリタブにクエリを書き込みます。詳細については、「クエリ」を参照してください。

   a. データソースを選択します。クエリタブの最初の行にあるドロップダウンリストを選択すると、使用可能なすべてのデータソースが表示されます。このリストには、追加したすべてのデータソースが表示されます。データソースの詳細については、「データソースに接続する」を参照してください。

   b. データソースのクエリ言語でクエリを記述または作成します。選択できる内容は、データソースによって異なります。詳細については、各データソースのドキュメントをご確認ください。
6. [パネル] タブの [視覚的表現]] セクションで、視覚的表現の種類を選択します。Grafana ワークスペースに、その視覚的表現を適用したクエリ結果のプレビューが表示されます。

7. [保存]] を選択する前に、変更内容を説明するメモを追加することをお勧めします。ダッシュボードを以前のバージョンに戻す必要が生じた際、このメモが役に立ちます。
8. ダッシュボードの保存は、画面の右上隅にある [保存]] を選択します。

パネルの削除

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana のパネルを削除するには、次のステップを実行します。

- パネルタイトルの横にあるドロップダウンを選択し、[パネルの削除] を選択します。

or

- パネルにカーソルを合わせ、キーボードを「p r」の順に押します。

クエリ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana ワークスペースのパネルは、クエリを使用してデータソースと通信し、視覚化用データを取得します。クエリは、データソースで使用するクエリ言語で記述された質問です。クエリが適切に形成

されると、データソースが応答します。パネルのデータソースオプションでは、クエリがデータソースに送信される頻度と、収集されるデータポイント数を調整できます。

Grafana ワークスペースは、パネルごとに最大 26 個のクエリをサポートします。

クエリエディタ

クエリエディタはクエリの記述に役立つフォームです。データソースによっては、クエリエディタが自動完了機能、メトリクス名、または変数の提案が用意されている場合があります。

クエリ言語の違いにより、各データソースのクエリエディタの外観が異なる場合があります。

クエリ構文

データソースがデータを求めるためのクエリ言語と構文が異なります。これらはその 2 つのクエリ例です。

[PostgreSQL]

```
SELECT hostname FROM host WHERE region IN($region)
```

PromQL

```
query_result(max_over_time(<metric>[${__range_s}s]) != <state>)
```

データソースのクエリの書き込みについては、そのデータソースのドキュメントを参照してください。データソースは データソースに接続する に一覧されています。

クエリタブ UI

[クエリ] タブは以下の要素で設定されています。

- データソースセレクター
- クエリオプション
- クエリインスペクターボタン
- クエリエディタリスト

データソースセレクター

データソースセレクターはドロップダウンリストです。追加済みのデータソースを選択するには、これを選択します。パネルを作成すると、Amazon Managed Grafana によりデフォルトのデータソースが自動的に選択されます。データソースの詳細については、「データソースに接続する」を参照してください。

Grafana ワークスペースで設定されたデータソースに加え、3 つの特別なデータソースを使用できます。

- TestDataDB – ランダムな歩行データを生成する組み込みデータソースです。Grafana データソースは視覚化のテストや実験の実行に役立ちます。
- 混合 – 同じパネル内の複数のデータソースをクエリするためのデータソースです。このデータソースを選択すると、追加する新しいクエリすべてにデータソースを選択できます。
  - 最初のクエリでは、[混合] を選択する前に選択済みのデータソースを使用します。
  - 既存のクエリを変更して混合データソースを使用することはできません。
- ダッシュボード – 同じダッシュボード内の別のパネルの一連の結果を使用するためのデータソースです。

クエリオプション

選択したデータソースの設定を表示するには、データソースセレクターの横にある [クエリオプション] を選択します。ここで行った変更はこのパネルで行われたクエリにのみ影響します。

Amazon Managed Grafana はダークグレーテキストで表示されるデフォルトを設定します。変更はホワイトテキストで表示されます。フィールドをデフォルト設定に戻すには、フィールドからホワイトテキストを削除します。

次のパネルデータソースのクエリオプションを使用できます。

- [最大データポイント数] – データソースがサポートしている場合、返される各シリーズのデータポイントの最大数を設定します。クエリが最大データポイント設定よりも多くのデータポイントを返す場合、データソースはそれらを統合します (平均、最大、または他の関数で集計することで、返されるポイントの数を減らします)。

  ポイント数を制限する主な 2 つの理由は、パフォーマンスおよびラインの平滑化です。デフォルト値はグラフの幅 (またはピクセル数) で、これにより、大きすぎてグラフパネルが表示できないデータポイントがなくなります。

ストリーミングデータでは、ローリングバッファに最大データポイント値が使用されます。(ストリーミングはデータの継続的なフローであり、バッファリングはストリームをチャンクに分割する方法です)。

- 最小間隔 – 自動計算された間隔の最小制限を設定します。これは通常、最小スクレイプ間隔です。データポイントを 15 秒ごとに保存する場合、それよりも間隔を短くする必要はありません。もう 1 つのユースケースは、スクレイプ間隔よりも高い最小値に設定して、粗く適切に機能するクエリを取得することです。
- 間隔 – データポイントを時間別に集計またはグループ化するときに使用できる時間範囲です。

  Amazon Managed Grafana は、テンプレートクエリで変数として使用できる適切な間隔を、自動的に計算します。変数は秒: `$__interval`、ミリ秒: `$__interval_ms` のいずれかです。通常、`sum` や `average` などの集計関数で使用されます。例えば、これは間隔変数 `rate(http_requests_total[$__interval])` を使用する Prometheus クエリです。

  この自動間隔は、グラフの幅に基づいて計算されます。ユーザーがズームアウトすると間隔が大きくなり、より粗視化された集計になります。ユーザーがズームインすると間隔が短くなり、より高精細な集計になります。

  詳細については、「グローバル変数」を参照してください。
- [相対時間] – 個々のパネルの相対時間範囲をオーバーライドし、これによりパネルは、ダッシュボードの右上隅にあるダッシュボードタイムピッカーで選択されている時間とは異なります。これにより同じダッシュボードに異なる期間または日数のメトリクスを表示できます。
- [タイムシフト] – 個々のパネルの時間範囲を上書きする別の方法です。この関数は相対時間範囲でのみ機能し、その時間範囲を調整できます。

  例えば、パネルの時間範囲をダッシュボードのタイムピッカーより 2 時間早くシフトできます。詳細については、「時間範囲コントロール」を参照してください。
- [キャッシュタイムアウト] – (このフィールドはデータソースで使用可能な場合にのみ表示されます)。時系列ストアにクエリキャッシュがある場合、デフォルトのキャッシュタイムアウトが上書きされます。数値 (秒単位) として指定します。

クエリインスペクターボタン

クエリインスペクター] を選択すると、パネルインスペクターの [クエリ] タブを開くことができます。[クエリ] タブには、パネルとレスポンスによって送信されたクエリリクエストが表示されます。

[更新] を選択すると、このパネルからサーバーに送信されたリクエストの全文が表示されます。

> **Note**
>
> クエリインスペクターが結果を返すようにするには、少なくとも 1 つのクエリを追加する必要があります。

パネルインスペクターの詳細については、「パネルの検査」を参照してください。

クエリエディタリスト

UI では、クエリは折りたたみ可能なクエリ行に整理されます。各クエリ行にはクエリエディタが含まれ、文字 (A、B、C など) で識別されます。

パネル間のクエリ結果の共有

Amazon Managed Grafana では、1 つのパネルのクエリ結果をダッシュボード内の他の任意のパネルで使用できます。パネル間でクエリ結果を共有すると、データソースに対して実行されるクエリの数が減少し、ダッシュボードのパフォーマンスが向上します。

ダッシュボードのデータソースを使用すると、結果を共有するクエリを含むダッシュボード内のパネルを選択できます。パネルごとに個別のクエリを送信する代わりに、Amazon Managed Grafana は 1 つのクエリを送信し、他のパネルはクエリ結果を使用して視覚化を構築します。

この戦略により、同じデータを視覚化するパネルが複数ある場合などに、実行されるクエリ数を大幅に減らすことができます。

データソースクエリを別のパネルと共有する方法

1. ダッシュボードを作成します。詳細については、「ダッシュボードの作成」を参照してください。
2. パネルを追加します。詳細については、「パネルの追加」を参照してください。
3. タイトルを「**`Source panel`**」に変更します。このパネルを他のパネルのソースとして使用します。共有するクエリ (1 つ以上) を定義します。現在利用可能なデータソースがない場合は、[Grafana] データソースを使用できます。このデータソースはテストに使用できるランダムな時系列を返します。
4. 2 番目のパネルを追加してから、クエリエディタで [ダッシュボード] データソースを選択します。
5. [パネルリストから結果を使用する] で、最初に作成したパネルを選択します。

ソースパネルで定義されたすべてのクエリが新しいパネルで使用できるようになりました。ソースパネルで行われたクエリは、複数のパネルと共有できます。

クエリを定義済みのパネルに移動するには、そのクエリを選択します。

記録済みのクエリ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

記録済みのクエリを使用して設定間隔でデータポイントのスナップショットを撮ると、時間の経過に伴う傾向を確認できます。これにより過去の傾向を把握できます。

時系列を返さないプラグインの場合は、履歴データをプロットすると便利な場合があります。例えば、[ServiceNow] をクエリするとリクエスト応答時間の履歴を表示できますが、現在時点のメトリクスのみを返すことができます。

記録済みのクエリの仕組み

記録済みのクエリは、バックエンドデータソースプラグインでのみ機能します。詳細については、「バックエンドデータソースプラグイン」を参照してください。次の 3 つのタイプのクエリを記録できます。

- 単一の行および列 - 単一の行および列を返すクエリ。
- 行カウント - カウントする意味のある行を返すクエリ。
- 式 - 任意の式。式の作成および使用の詳細については、「式」を参照してください。

記録済みのクエリを作成または有効にすると、すぐにスナップショットが作成され、設定間隔でスナップショットが引き続き作成されます。記録済みのクエリでは、スナップショットが無効化、削除された場合、または Grafana が実行されていない場合、スナップショットの作成が停止します。記録済みのクエリを実行し、各結果をリモート書き込み対応の Prometheus インスタンスに転送すると、データポイントはバックエンドに収集されます。

> **Note**
>
> 記録済みのクエリを使用するには、Prometheus データソースを設定してリモート書き込み先に関連付ける必要があります。

記録済みのクエリの作成

記録済みのクエリを作成するには、次の手順を実行します。

1. 編集パネルでダッシュボードに記録するクエリを検索または作成します。クエリは単一の行と列のみを返す必要があります。さらに多く返される場合、カウントオプションを使用して、返される結果の数を記録できます。クエリのデータソースは、バックエンドデータソースである必要があります。式は時系列クエリからのデータ集約に使用できます。式の作成および使用の詳細については、「式」を参照してください。
2. クエリエディタで [レコードクエリ] メニューを選択します。
3. 記録済みのクエリの情報を入力します。特に指示されていない限り、すべてのフィールドは必須です。
   - 名前 - 記録済みのクエリの名前。
   - 説明 - (オプション) 記録済みのクエリのリストに表示する、記録済みのクエリに関する説明を記述します。
   - 間隔 - スナップショットが取得される間隔です。間隔は、記録済みのクエリの作成時に開始され、記録済みのクエリを一時停止または削除すると停止します。記録済みのクエリの一時停止と削除の詳細については、「記録済みのクエリの管理」を参照してください。
   - 範囲 - クエリの相対時間範囲です。30m の範囲と 1h の間隔を選択すると、クエリは過去 30 分の 1 時間ごとのスナップショットを取得します。
   - クエリ結果のカウント - クエリが返した行をカウントする場合は、このオプションをオンに切り替えます。このオプションがオフの場合、クエリは 1 つの値を持つ 1 行を返します。
4. [記録済みのクエリのテスト] ボタンを選択して、記録済みのクエリをテストします。
5. [記録の開始] を選択します。

記録済みのクエリの追加

ダッシュボードで、既存の記録済みのクエリをパネルに追加できます。記録済みのクエリを追加するたび、Prometheus クエリ `generated_recorded_query_name{id="generated_id",`

name="recorded query name"} が作成されます。Prometheus から作成されたクエリは、記録済みのクエリすべてが収集されたスナップショットを返します。

1. 記録済みのクエリを追加するダッシュボードのパネルに移動します。
2. [+ 記録済みのクエリ] メニューを選択します。
3. データソースにより記録済みのクエリをフィルタリングする場合は、データソースドロップダウンメニューで、フィルターのデータソースを選択します。
4. 記録済みのクエリで [追加] メニューを選択して、パネルに追加します。

記録済みのクエリをパネルに追加すると、パネルデータソースは -- Mixed -- になります。記録済みのクエリは、記録済みのクエリの名前と一致する名前ラベルが付いた Prometheus クエリで表されます。Prometheus データソースの詳細については、「Prometheus」を参照してください。

記録済みのクエリを追加した後、Prometheus データソースではなく -- Mixed -- データソースを持つクエリが表示される場合は、記録済みのクエリに対して Prometheus リモート書き込み先が設定されていない可能性があります。「リモート書き込み先」を参照して、リモート書き込みポイントをセットアップしてください。

記録済みのクエリの使用

記録済みのクエリを使用するには、クエリを作成してダッシュボードに追加します。その後、[記録済みのクエリ] タブの [設定] で管理できます。

記録済みのクエリの管理

記録済みのクエリを一時停止またはアクティブ化すると、設定の [記録済みのクエリ] タブから削除できます。記録済みのクエリを削除すると Grafana から削除されますが、Prometheus で収集された情報はそこに残ります。記録済みのクエリを一時停止すると、再開されるまで新しいデータポイントが収集されなくなります。

リモート書き込み先

リモート書き込み先は、記録済みのクエリデータポイントが書き込まれる [Prometheus] データソースです。リモート書き込みが有効な Prometheus が必要で、この Prometheus のデータソースを作成する必要があります。

リモート書き込み先を編集するには、[設定] の [記録済みのクエリ] タブのコンソールメニューで、[リモート書き込み先の編集] を選択します。リモート書き込みが有効な [Prometheus] データソースを選択して、リモート書き込みパスを入力します。

変換

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

変換は視覚化に渡される前に一連の結果を処理します。Amazon Managed Grafana パネルエディタの [変換] タブで変換にアクセスします。

変換を使用して、フィールドの名前変更、個別の時系列の結合、クエリ間の計算などを実行できます。大きなダッシュボードや大量のクエリがある場合、あるパネルのクエリ結果を別のパネルで再利用できれば、パフォーマンスが大幅に向上する可能性があります。

Note

変換ではグラフ化できないデータが発生することがあります。この場合、Amazon Managed Grafana では視覚化に提案が表示されます。テーブル視覚化に切り替える提案を選択します。多くの場合、変換がデータに何を実行しているかを理解するのに役立ちます。

Amazon Managed Grafana は画面に一覧される順序で変換を適用します。各変換ではパイプライン内の次の変換に渡される新しい結果セットが作成されます。

この順序により結果が大きく異なる場合があります。例えば、縮小変換を使用して 1 つの列のすべての結果を 1 つの値に集約する場合、変換はその 1 つの値にのみ適用できます。

前提条件

変換を適用する前に、次のすべてが実行済みである必要があります。

- クエリを入力してデータソースからデータを返された。クエリの詳細については、「クエリ」を参照してください。
- 次のいずれかの視覚化などの、クエリをサポートする視覚化を適用した。

- 棒ゲージ
- ゲージ
- グラフ
- ヒートマップ
- ログ
- 統計
- [テーブル]

変換の適用

変換は、パネルエディタの下部ペインの、[クエリ] タブの隣の [変換] タブから使用できます。

変換の適用方法

1. 変換を追加するパネルで、パネルタイトルを選択してから、[編集] を選択します。
2. [変換] タブを選択します。
3. 変換を選択します。

   表示される変換行では、変換オプションを設定できます。
4. [変換ルールの追加] を選択すると、別の変換ルールを追加できます。次の変換は、前の変換が返した一連の結果に従って動作することに注意してください。

問題が発生した場合は、バグアイコンを選択して [変換をデバッグ] します。

変換を削除する場合、ごみ箱アイコンを選択します。

変換タイプおよびオプション

Grafana ワークスペースには次の変換が含まれます。

トピック

- 削減
- Merge (マージ)
- 名前に基づいてフィルタリングする
- クエリによるデータのフィルタリング

- フィールドの整理
- フィールドにより結合 (外部結合)
- 計算からのフィールドの追加
- フィールドへのラベル付け
- グループ化の条件
- グループ化の条件
- シリーズから行
- 値によるデータのフィルタリング
- 変換のデバッグ

削減

結果を 1 つの値に簡略化する場合は、[削減] 変換を適用します。削減では基本的に時間コンポーネントを削除します。テーブルとして視覚化する場合は、列が 1 行 (値) に減少します。

[計算] フィールドに、1 つ以上の計算タイプを入力します。計算選択肢のリストの表示を選択します。利用可能な計算の詳細については、「計算リスト」を参照してください。

少なくとも 1 つの計算を選択すると、Amazon Managed Grafana では選択した計算を使用して 1 つの値が表示されます。複数の計算を選択すると、複数の値が表示されます。

Merge (マージ)

この変換を使用して、複数クエリの結果を 1 つの結果にまとめます。これはテーブルパネルの視覚化を使用する場合に役立ちます。マージできる値は同じ行に結合されます。共有フィールドに同じデータが含まれている場合は値をマージできます。

次の例では、2 つのクエリがテーブルデータを返します。変換を適用する前、データは 2 つの個別のテーブルとして視覚化されています。

クエリ A

時間	ジョブ	アップタイム
2020-07-07 11:34:20	ノード	25260122

時間	ジョブ	アップタイム
2020-07-07 11:24:20	postgre	123001233

クエリ B

時間	ジョブ	エラー
2020-07-07 11:34:20	ノード	15
2020-07-07 11:24:20	postgre	5

[マージ] 変換を適用した後の結果は次のとおりです。

時間	ジョブ	エラー	アップタイム
2020-07-07 11:34:20	ノード	15	25260122
2020-07-07 11:24:20	postgre	5	123001233

名前に基づいてフィルタリングする

この変換を使用してクエリ結果の一部を削除します。

Amazon Managed Grafana には [識別子] フィールドが表示され、クエリが返すフィールドが続きます。

フィルターは次の 2 つのいずれかの方法で適用できます。

- 正規表現を入力します。
- フィールドを選択してフィールドのフィルタリングを切り替えます。フィルタリングされたフィールドにはダークグレーテキストが表示され、フィルタリングされていないフィールドには白いテキストが表示されます。

クエリによるデータのフィルタリング

1 つ以上のクエリを非表示にするには、複数のクエリがあるパネルでこの変換を使用します。

Amazon Managed Grafana ではクエリ識別文字がダークグレーテキストで表示されます。フィルタリングを切り替えるには、クエリ識別子を選択します。クエリ文字が白い場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示です。

フィールドの整理

この変換を使用して、クエリが返すフィールドの名前を変更、並べ替え、非表示にします。

> **Note**
>
> この変換は単一クエリを持つパネルでのみ機能します。パネルに複数のクエリがある場合は、[フィールドによる結合 (外部結合)] 変換を適用、またはさらにクエリを削除する必要があります。

Amazon Managed Grafana ではクエリが返すフィールドのリストが表示されます。以下の任意の変更を行うことができます。

- フィールド上で一時停止してフィールドの順序を変更します。カーソルが手の形になると、フィールドを新しい場所にドラッグできます。
- フィールド名の隣の目アイコンを選択して、フィールドを非表示または表示します。
- 名前変更ボックスに新しい名前を入力して、フィールドの名前を変更します。

フィールドにより結合 (外部結合)

この変換を使用してフィールドで設定された結果から複数の時系列を結合します。

この変換は、フィールドからの結果を計算できるようにするため、クエリを組み合わせる場合に役立ちます。

計算からのフィールドの追加

この変換を使用して、他の 2 つのフィールドから計算した新しいフィールドを追加します。各変換では 1 つの新しいフィールドを追加できます。

- [モード] – 以下のモードを選択します。
  - [行を減らす] – 選択したフィールドの各行に、選択済みの計算を個別に適用します。
  - バイナリオプション – 選択した 2 つのフィールドから、1 つの行の値に基本的な計算演算 (`sum`、`multiply` など) を適用します。

- フィールド名 – 新しいフィールドの計算に使用する、フィールドの名前を選択します。
- 計算 - Amazon Managed Grafana により新しいフィールドが作成されるときに使用する計算を選択します。計算選択肢のリストを表示するフィールドを選択します。利用可能な計算の詳細については、「計算リスト」を参照してください。
- [エイリアス] – (オプション) 新しいフィールドの名前を入力します。これを空白のままにすると、計算に一致するようにフィールドの名前が付けられます。
- すべてのフィールドを置き換え – (オプション) 他のすべてのフィールドを非表示にし、計算したフィールドのみを視覚化に表示する場合は、このオプションを使用します。

フィールドへのラベル付け

> **Note**
>
> この変換を適用するには、クエリがラベル付きフィールドを返す必要があります。

この変換を選択すると、Amazon Managed Grafana により、ラベル付けされたすべてのデータがフィールドに自動的に変換されます。

例えば、2 つの時系列のクエリ結果を考慮しましょう。

1: labels Server=Server A, Datacenter=EU 2: labels Server=Server B, Datacenter=EU

この変換により次の表が表示されます。

時間	[Server] (サーバー)	データセンター	値
2020-07-07 11:34:20	サーバー A	EU	1
2020-07-07 11:34:20	サーバー B	EU	2

[値フィールド名]: `Server` を [値フィールド名] として選択した場合は、`Server` ラベルの値ごとに 1 つのフィールドを取得します。

時間	データセンター	サーバー A	サーバー B
2020-07-07 11:34:20	EU	1	2

グループ化の条件

この変換は設定されたフィールドで各フレームをソートします。reverse がチェックされているとき、値は逆の順序で返されます。

グループ化の条件

この変換では、指定されたフィールド (列) 値でデータをグループ化し、各グループの計算を処理します。使用可能な計算は削減変換の場合と同じです。

オリジナルデータの例を次に示します。

時間	サーバー ID	CPU 温度	サーバーステータス
2020-07-07 11:34:20	サーバー 1	80	シャットダウン
2020-07-07 11:34:20	サーバー 3	62	OK
2020-07-07 10:32:20	サーバー 2	90	オーバーロード
2020-07-07 10:31:22	サーバー 3	55	OK
2020-07-07 09:30:57	サーバー 3	62	再起動
2020-07-07 09:30:05	サーバー 2	88	OK
2020-07-07 09:28:06	サーバー 1	80	OK
2020-07-07 09:25:05	サーバー 2	88	OK
2020-07-07 09:23:07	サーバー 1	86	OK

この変換では 2 つのステップを実行します。まず、データによりグループ化する 1 つ以上のフィールドを指定します。これにより、これらのフィールドの同じ値すべてが、ソートされたかのようにグループ化されます。例えば、Server ID フィールドに [グループ化の条件] を適用すると、データは次のようにグループ化されます。

時間	サーバー ID	CPU 温度	サーバーステータス
2020-07-07 11:34:20	サーバー 1	80	シャットダウン

時間	サーバー ID	CPU 温度	サーバーステータス
2020-07-07 09:28:06	サーバー 1	80	OK
2020-07-07 09:23:07	サーバー 1	86	OK

```
2020-07-07 10:32:20 | server 2 | 90 | Overload
2020-07-07 09:30:05 | server 2 | 88 | OK
2020-07-07 09:25:05 | server 2 | 88 | OK

2020-07-07 11:34:20 | server 3 | 62 | OK
2020-07-07 10:31:22 | server 3 | 55 | OK
2020-07-07 09:30:57 | server 3 | 62 | Rebooting
```

同じ値の Server ID を持つすべての行がグループ化されます。

データをグループ化するフィールドを選択すると、他のフィールドにさまざまな計算を追加でき、その計算は行の各グループに適用されます。例えば、これらの各サーバーの平均 CPU temperature の計算について考えましょう。CPU Temperature フィールドに適用される平均計算を追加して、以下を取得できます。

サーバー ID	CPU 温度 (平均)
サーバー 1	82
サーバー 2	88.6
サーバー 3	59.6

これらの計算を複数追加できます。例えば、次の計算を使用できます。

- フィールド Time では、[最後]の値を計算して、サーバーごとにデータポイントを最後にいつ受信したかを知ることができます。
- フィールド Server Status では、最後の値を計算して、サーバーごとに最後の状態値を知ることができます。
- フィールド Temperature では、最後の値を計算して、サーバーごとに最新の測定温度を知ることができます。

グループ化の条件変換では次の結果が生成されます。

サーバー ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最後)	サーバーステータス (最後)
サーバー 1	82	80	2020-07-07 11:34:20	シャットダウン
サーバー 2	88.6	90	2020-07-07 10:32:20	オーバーロード
サーバー 3	59.6	62	2020-07-07 11:34:20	OK

この変換を使用すると、時系列から一部のキー情報を抽出して便利な方法で表示できます。

シリーズから行

この変換を使用して、複数の時系列データクエリの結果を 1 つの結果にまとめます。これはテーブルパネルの視覚化を使用する場合に役立ちます。

この変換の結果には、`Time`、`Metric`、`Value` の 3 つの列が含まれます。`Metric` 列が追加されると、メトリクスの元となったクエリを確認できます。ソースクエリで `Label` を定義して、この値をカスタマイズします。

次の例では、2 つのクエリは時系列データを返します。変換を適用する前に、2 つの個別のテーブルとして視覚化されます。

クエリ A

時間	温度
2020-07-07 11:34:20	25
2020-07-07 10:31:22	22
2020-07-07 09:30:05	19

クエリ B

時間	湿度
2020-07-07 11:34:20	24
2020-07-07 10:32:20	29
2020-07-07 09:30:57	33

`Series to rows` 変換を適用すると次の結果が得られます。

時間	メトリクス	値
2020-07-07 11:34:20	温度	25
2020-07-07 11:34:20	湿度	22
2020-07-07 10:32:20	湿度	29
2020-07-07 10:31:22	温度	22
2020-07-07 09:30:57	湿度	33
2020-07-07 09:30:05	温度	19

値によるデータのフィルタリング

この変換により、Grafana ワークスペースでデータを直接フィルタリングして、クエリ結果から一部のデータポイントを削除できます。1 つ以上の定義済み条件と一致するデータを含めるか除外するかを選択できます。その条件は選択したフィールドに適用されます。

この変換はデータソースが値によりネイティブにフィルタリングしない場合に役立ちます。共有クエリを使用している場合、これを使用して値を絞り込んで表示することもできます。

すべてのフィールドで使用できる条件は次のとおりです。

- [Regex] – 正規表現を照合します。
- [Is Null] – 値が null の場合に照合。

- [Is Not Null] – 値が null でない場合に照合します。
- Equal – 値が指定値と等しい場合に照合します。
- Different – 値が指定された値と異なる場合に照合します。

数値フィールドで使用可能な条件は以下のとおりです。

- Greater – 値が指定値より大きい場合に照合します。
- Lower – 値が指定値よりも小さい場合に照合します。
- Greater or equal – 値が指定値以上の場合に照合します。
- Lower or equal – 値が指定値以下の場合に照合します。
- Range – 指定された最小値と最大値間の範囲を照合します。最小と最大は範囲に含まれます。

複数の条件をフィルターに追加できます。複数の条件がある場合、追加した条件のすべてまたはいずれかと一致する行に、含めたり除外したりするためのアクションを適用するかどうかを選択できます。

有効でない条件、または設定が不完全である条件は無視されます。

変換のデバッグ

変換の入力および出力結果のセットを表示するには、変換行の右側にあるバグアイコンを選択します。

Amazon Managed Grafana の変換行の下に変換デバッグビューが表示されます。

フィールドオプションとオーバーライド

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、Amazon Managed Grafana のフィールドオプションおよびフィールドオーバーライドとは何か、およびそれらの使用方法について説明します。

Grafana ワークスペースで使用されるデータモデルであるデータフレームは、時系列とテーブルのクエリ結果の両方が統合された列指向のテーブル構造です。この構造内の各列は、フィールドと呼ばれます。フィールドは、単一の時系列またはテーブル列を表すことができます。

フィールドオプションを使用すると、視覚的表現でのデータの表示方法を変更できます。オプションとオーバーライドを適用してもデータは変更されません。Amazon Managed Grafana でのデータの表示方法が変更されます。

フィールドオプション

パネルエディタの [フィールド] タブには、標準とカスタムの両方のフィールドオプションがあります。このタブで行われた変更は、すべてのフィールド (シリーズと列) に適用されます。例えば、単位をパーセンテージに変更すると、すべての数値フィールドがパーセンテージで表示されます。フィールドオプションを適用する方法については、「すべてのフィールドを設定する」で説明します。

フィールドのオーバーライド

フィールドのオーバーライドは、パネルエディタの [オーバーライド] タブに追加できます。このタブには [フィールド] タブと同じオプションを追加できますが、特定のフィールドのみの適用となります。オーバーライドを適用する方法については、「特定のフィールドを設定する」で説明します。

使用可能なフィールドオプションとオーバーライド

フィールドオプションの種類は、フィールドオプションとフィールドオーバーライドの両方に共通しています。唯一の違いは変更の適用先で、すべてのフィールド ([フィールド] タブに適用) かフィールドのサブセット ([オーバーライド] タブに適用) かが異なります。

- 標準フィールドオプション は、変換を許可するすべてのパネルの視覚的表現に適用されます。
- テーブルのフィールドオプション は、テーブルパネルの視覚的表現にのみ適用されます。

すべてのフィールドを設定する

[フィールド] タブでオプションを変更すると、すべてのフィールドのデータの表示方法を変更することができます。[オーバーライド] タブからは、特定のフィールドのフィールドオプションをオーバーライドすることができます。詳細については、「特定のフィールドを設定する」を参照してください。

例えば、小数オプションを変更すると、すべてのフィールドに表示される小数点以下の桁数が変更されます。オプションの詳細については、「標準フィールドオプション」および「テーブルのフィールドオプション」を参照してください。

フィールドオプションを変更する

オプションはいくつでも変更できます。

フィールドオプションの変更方法

1. 編集するパネル、パネルタイトル、[編集] の順に選択します。
2. [フィールド] タブを選択します。
3. 変更するオプションを見つけます。以下を定義できます。
   - 標準フィールドオプション (変換を許可するすべてのパネル視覚的表現に適用されます)
   - テーブルのフィールドオプション (テーブルパネルの視覚的表現にのみ適用されます)
4. フィールドに値を記入してオプションを追加します。オプションをデフォルト値に戻すには、フィールドの白い文字を削除します。
5. ダッシュボードの編集が完了したら、[保存] を選択します。

フィールドオプションの例

結果セットが、「時間」と「温度」の 2 つのフィールドで構成されるデータフレームである場合を例にします。

時間	温度
2020-01-02 03:04:00	45.0
2020-01-02 03:05:00	47.0
2020-01-02 03:06:00	48.0

この構造の各フィールド (列) にフィールドオプションを適用してその値の表示方法を変えることができます。例えば、単位を温度 > 摂氏に設定すると、次の表になります。

時間	温度
2020-01-02 03:04:00	45.0 °C
2020-01-02 03:05:00	47.0 °C
2020-01-02 03:06:00	48.0 °C

この例では、小数点に値が無いため不要になります。小数を auto からゼロ (0) に変更すると、次の表になります。

時間	温度
2020-01-02 03:04:00	45 °C
2020-01-02 03:05:00	47 °C
2020-01-02 03:06:00	48 °C

特定のフィールドを設定する

オーバーライドを使用して、1 つ以上のフィールドの設定を変更することができます。オーバーライドのフィールドオプションは、特定の視覚的表現で使用可能なフィールドオプションとまったく同じです。唯一の違いは、適用するフィールドを選択する点です。

例えば、すべての数値フィールドまたは列に表示される小数点以下の桁数を変更するには、型が数値のフィールドの小数点オプションを変更します。オプションの詳細については、変換を許可するすべてのパネル視覚的表現に適用される「標準フィールドオプション」およびテーブルパネルの視覚的表現のみに適用される「テーブルのフィールドオプション」を参照してください。

フィールドオーバーライドを追加する

フィールドオプションは、いくつでも上書きできます。

フィールドオーバーライドの追加方法

1. 編集するパネル、パネルタイトル、[編集] の順に選択します。
2. [オーバーライド] タブを選択します。

3. [オーバーライドの追加先] を選択します。
4. オーバーライドルールを適用するフィールドを選択します。

   - [名前付きフィールド] – 使用可能なすべてのフィールドの一覧からフィールドを選択します。このセレクタでルールに追加するプロパティは、この単一フィールドにのみ適用されます。
   - [正規表現と一致する名前付きフィールド] – 正規表現でオーバーライドするフィールドを指定します。このセレクタでルールに追加するプロパティは、フィールド名が正規表現と一致するすべてのフィールドに適用されます。
   - [タイプ別フィールド] – 文字列、数値などのフィールドをタイプ別に選択します。このセレクタでルールに追加するプロパティは、選択した型のすべてのフィールドに適用されます。
5. [オーバーライドプロパティの追加] を選択します。
6. 適用するフィールドオプションを選択します。
7. フィールドに値を追加してオプションを入力します。オプションをデフォルト値に戻すには、フィールドの白い文字を削除します。
8. [オーバーライドプロパティの追加] を選択して、このフィールドにオーバーライドを追加し続けるか、[オーバーライドの追加] を選択し、オーバーライドを追加する別のフィールドを選択できます。
9. 完了したら、[保存] を選択します。

フィールドオーバーライドの削除

1. 削除するオーバーライドがある [オーバーライド] タブを選択します。
2. オーバーライドの横にあるごみ箱アイコンを選択します。

フィールドオーバーライドの例

結果セットが、「時間」、「高温」、「低温」、「湿度」の 4 つのフィールドで構成されるデータフレームである場合を例にします。

時間	高温	低温	湿度
2020-01-02 03:04:00	45.0	30.0	67
2020-01-02 03:05:00	47.0	34.0	68
2020-01-02 03:06:00	48.0	31.0	68

フィールドオプションの例 と同様に、摂氏単位を適用して小数点以下を削除します。表は次のようになります。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67 °C
2020-01-02 03:05:00	47 °C	34 °C	68 °C
2020-01-02 03:06:00	48 °C	31 °C	68 °C

温度フィールドは適切ですが、湿度フィールドが適切ではありません。これを修正するには、湿度フィールドにフィールドオプションオーバーライドを適用し、単位を [Misc] (その他) > % (0 ~ 100) に変更します。これにより、表が適切になりました。

時間	高温	低温	湿度
2020-01-02 03:04:00	45 °C	30 °C	67%
2020-01-02 03:05:00	47 °C	34 °C	68%
2020-01-02 03:06:00	48 °C	31 °C	68%

標準フィールドオプション

このセクションでは、使用可能なフィールドオプションについて説明します。アルファベット順に紹介します。

標準フィールドオプションは、ほとんどのビルトインの Grafana ワークスペースパネルに適用できます。新しいパネルとデータモデルに更新されていない古いパネルとコミュニティパネルの一部には、これらのフィールドオプションのすべてまたは一部が欠落します。

ほとんどのフィールドオプションは、編集するフィールドオプションボックスの外部を選択するか、Enter キーを押すまで、視覚的表現には影響しません。

これらのオプションの適用の詳細については、以下のセクションを参照してください。

- すべてのフィールドを設定する

- 特定のフィールドを設定する

小数

表示する値の小数点以下の桁数を設定します。Amazon Managed Grafana がデータソースによって提供される小数点以下の桁数をそのまま使用する場合、空白のままにします。

この設定を変更するには、 フィールドに数値を入力します。

データリンク

値または視覚的表現がリンクする URL を制御します。詳細と手順については、「データリンク」を参照してください。

Display name (表示名)

すべてのフィールドの表示タイトルを設定します。フィールドタイトルには変数を使用できます。テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

複数の統計、フィールド、または系列が表示される場合、このフィールドは各統計のタイトルを制御します。式 (例: `${__field.name}`) を使用して、タイトルに系列名またはフィールド名のみを表示させることができます。

「Temp」という名前のフィールドがあり、ラベルが {"Loc"="PBI", "Sensor"="3"} である場合

式の構文	例	レンダリング先	説明
`${__fi .displ ame}`	構文と同じ	`Temp {Loc="P ", Sensor= "}`	フィールド名とラベルがある場合、{} 内が表示されます。レスポンスにラベルキーが 1 つしかない場合、ラベル部分において、Amazon Managed Grafana は括弧なしでラベルの値を表示します。
`${__fi .name}`	構文と	`Temp`	フィールドの名前を表示します (ラベルなし)。

式の構文	例	レンダリング先	説明
	同じ		
${__fi .label	構文と同じ	Loc="PE , Sensor= "	ラベルを表示します (名前なし)。
${__fi .label }	${__ .labe oc}	PBI	指定されたラベルキーの値を表示します。
${__fi .label _value	構文と同じ	PBI, 3	ラベルの値をカンマ区切りで表示します (ラベルキーなし)。

特定のフィールドに対する式をレンダリングした後に値が空の文字列である場合、デフォルトの表示方法が使用されます。

最大

パーセンテージしきい値の計算に使用される最大値を設定します。すべての系列とフィールドに基づいて自動計算を行うには、この設定を空白のままにします。

最小

パーセンテージしきい値の計算に使用される最小値を設定します。すべての系列とフィールドに基づいて自動計算を行うには、この設定を空白のままにします。

値なし

フィールド値が空または null の場合に Amazon Managed Grafana が表示する値を入力します。

[単位]

フィールドで使用される単位を指定します。[単位]フィールドを選択し、目的の単位が見つかるまでドリルダウンします。選択した単位は、時間を除くすべてのフィールドに適用されます。

ユーザー定義の単位

単位ドロップダウンを使用して、ユーザー定義の単位、接頭辞や接尾辞、日付時刻形式を指定することもできます。

ユーザー定義の単位を選択するには、単位を入力し、ドロップダウンで最後の [Custom: xxx] オプションを選択します。

- `suffix:<suffix>`: ユーザー定義の単位を値の後に表示。
- `time:<format>`: ユーザー定義の日付/時刻形式。例: `time:YYYY-MM-DD`。形式の構文とオプションについては、「表示」を参照してください。
- `si:<base scale><unit characters>`: ユーザー定義の SI 単位。例: `si: mF`。単位とソースデータスケールの両方を指定できるため、少し高度です。例えば、ソースデータがミリ (千分の一) 単位で表されている場合、単位の先頭にその SI スケール文字が付けられます。
- `count:<unit>`: ユーザー定義のカウント単位。
- `currency:<unit>`: ユーザー定義の通貨単位。

単位ピッカーにネイティブ絵文字を貼り付けて、ユーザー定義単位として選択することもできます。

[String units] (文字列の単位)

Amazon Managed Grafana は、文字列を解析して数値として表示することがあります。Amazon Managed Grafana に元の文字列を表示させるには、フィールドオーバーライドを作成し、`string` 単位の単位プロパティを追加します。

カラースキーム

フィールドの色オプションは、Amazon Managed Grafana が系列またはフィールドを色分けする方法を定義します。それぞれ動作が異なる複数のモードがあり、その有用性は、現在選択されている視覚的表現に大きく依存します。

連続色モードでは、最小値と最大値から割合を算出して色を補間します。

- 単一色 – カラーピッカーを使用して特定の色を設定します。これは、オーバーライドルールで最も便利です。

- 閾値から – 閾値から色を派生させます。これは、ゲージ、統計、テーブルの視覚的表現で使用できます。

系列別の色設定

Amazon Managed Grafana には、系列ごとに色を定義する配色機能があります。これは、グラフや円グラフなどで使用できます。

値別の色設定

Amazon Managed Grafana には、連続 (グラデーション) のカラースキーム機能もあります。これは、統計パネルやテーブルパネルなど、個々の値を色分けする視覚的表現で使用できます。

しきい値

しきい値を設定することで、値を基にフィールドの色を変更することができます。詳細と手順については、「しきい値」を参照してください。

値の配置

フィールド値または値の範囲を明示的なテキストに変換するルールを設定できます。複数の値を配置できます。

- [配置タイプ]でオプションを選択します。
  - [値] に値を入力します。フィールド値が指定した値以上の場合、テキストが表示されます。
  - [開始]] と [終了] に範囲を入力します。フィールド値が範囲内である場合、テキストが表示されます。
- [テキスト] は、フィールドで条件が満たされた場合に表示されるテキストです。このフィールドには変数を使用できます。

パネルエディタ

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このトピックでは、Amazon Managed Grafana パネルエディタの各部品について説明しています。また、詳細情報を確認するためのリンクも記載しています。

パネルエディタを開く

パネルエディタ (呼称は編集パネル画面、編集モード、またはパネル編集モードなど様々です) にアクセスする方法はいくつかあります。

- 画面上部にある [パネルの追加] アイコンを選択し、[新しいパネルの追加] を選択します。新しいパネルがパネルエディタで開きます。パネルを追加する方法については、「パネルの追加」を参照してください。
- 既存のパネルのタイトルを選択して [編集] を選択します。パネルが編集モードで開きます。
- 既存のパネルの任意の場所を選択し、キーボードで e を入力します。パネルが編集モードで開きます。

パネルエディタセクションのサイズ変更

パネルエディタのセクションのサイズは、ドラッグして変更することができます。サイドペインが狭すぎると、[パネル]、[フィールド]、[オーバーライド] タブがドロップダウンリストに変わります。

パネルエディタの部品

このセクションでは、パネルエディタ画面の部品と、各部品に関連付けられたフィールド、オプション、タスクについて説明します。

ヘッダー

ヘッダーセクションには、パネルが属するダッシュボードの名前と一部のダッシュボードコマンドが表示されます。[戻る] 矢印を選択してダッシュボードに戻ることもできます。

ヘッダーの右側には、以下のオプションがあります。

- [ダッシュボード設定 (ギア) アイコン] — ダッシュボード設定へのアクセスを選択します。
- Discard (破棄) — 選択すると、ダッシュボードを最後に保存してから行ったすべての変更が破棄されます。
- [保存] — 選択すると、パネルエディタで行ったすべての変更を含めてダッシュボードが保存されます。
- [適用] — 選択すると、行った変更が適用され、パネルエディタが閉じてダッシュボードに戻ります。また、適用された変更を保持するためにダッシュボードも保存します。

視覚的表現のプレビュー

視覚的表現のプレビューセクションには、表示オプション、時間範囲コントロール、視覚的表現のプレビュー、該当する場合はパネルタイトル、軸、凡例が含まれます。

- [塗りつぶし] – 視覚的表現のプレビュー上のプレビューパートにある塗りつぶし可能なスペースが塗りつぶされます。サイドペインの幅またはボトムペインの高さを変更すると、視覚的表現の塗りつぶし可能なスペースが塗りつぶされます。
- [塗りつぶし] – 視覚的表現のプレビューで塗りつぶし可能なスペースが塗りつぶされますが、パネルのアスペクト比は保持されます。
- [正確] – 視覚的表現のプレビューが、ダッシュボードの実際のサイズで表示されます。十分なスペースがない場合、視覚的表現はアスペクト比が保持されたまま縮小されます。
- [時間範囲コントロール] – 詳細については、「時間範囲コントロール」を参照してください。

データセクション (最下部ペイン)

データセクションには、クエリの入力、データの変換、アラートルールの作成 (該当する場合) を行うタブがあります。

- [クエリ] タブ – データソースを選択し、ここにクエリを入力します。詳細については、「クエリ」を参照してください。
- [変換 タブ] – データ変換を適用します。詳細については、「変換」を参照してください。
- [アラート] タブ – アラートルールを入力します。詳細については、「Grafana アラート」を参照してください。

パネルとフィールドのオプション (サイドペイン)

データの視覚化方法のほぼすべての設定を行うタブがこのセクションにあります。視覚的表現ごとに利用できるタブは異なります。

これらのタブの機能については、次のトピックで説明します。

- パネルの追加
- 視覚化
- フィールドオプションとオーバーライド
- パネルリンク および データリンク (視覚的表現を他のリソースに接続する際に役立つ情報を掲載しています)

ライブラリパネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ライブラリパネルを使用することで、ユーザーは再利用可能なパネルを作成できます。また、1 つのインスタンスに加えた変更が、そのパネルが使用されているすべてのダッシュボードに自動的に反映されます。これらのパネルはダッシュボードと一緒にフォルダに保存でき、複数のダッシュボード間でパネルの再利用を効率化できます。

ライブラリパネルを作成する

Note

ライブラリパネルを作成すると、元のダッシュボードのパネルもライブラリパネルに変換されます。パネルが変換されたら、元のダッシュボードを保存する必要があります。

ライブラリパネルの作成方法

1. 通常どおりにパネルを作成します。既存のパネルを使用することもできます。
2. パネルのタイトルを選択して [編集] を選択します。
3. パネル表示オプションサイドペインで、下矢印オプションを選択して視覚的表現に変更を加えます。
4. [ライブラリパネル] を選択して [新規ライブラリパネルを作成] を選択します。
5. ライブラリパネルの名前を入力し、フォルダを選択して保存します。
6. [ライブラリパネルの作成] を選択し、ダッシュボードを保存します。

また、任意のパネルの [共有] オプションを使用してライブラリパネルを作成することもできます。

作成後、そのライブラリパネルは、ライブラリパネルが表示される任意のダッシュボードを使用して変更することができます。ライブラリパネルに変更を加えて保存すると、その内容がライブラリパネルのすべてのインスタンスに反映されます。

ライブラリパネルを追加する

ダッシュボードへのライブラリパネルの追加方法

1. 左のメニューで「+」オプションを選択し、次に [作成] を選択します。
2. [パネルライブラリ からパネルを追加する] を選択します。
3. ライブラリパネルのリストをフィルタリングして、パネルを見つけます。
4. そのパネルを選択し、ダッシュボードに追加します。

ライブラリパネルのリンクを解除する

ライブラリパネルのリンクを解除することで、ライブラリパネルの他のすべてのインスタンスに影響を与えずにダッシュボードのライブラリパネルに変更を加えることができます。

ダッシュボードからライブラリパネルのリンクを解除する方法

1. 左のメニューで [ダッシュボード] を選択し、次に [ライブラリパネル を選択します。
2. ライブラリパネルを選択します。そのライブラリパネルが使用されているすべてのダッシュボードのリストが表示されます。
3. リンクを解除するパネルを選択して更新します。
4. パネルのタイトルを選択して [編集] を選択します。
5. [リンクの解除] を選択します。

ライブラリパネルを削除する

ライブラリパネルを削除する前に、そのパネルがダッシュボードで使用されていないことを確認します。

ライブラリパネルの削除方法

1. 左のメニューで [ダッシュボード] を選択し、次に [ライブラリパネル を選択します。
2. ライブラリパネルを選択します。そのライブラリパネルが使用されているすべてのダッシュボードのリストが表示されます。

3. 削除するパネルを選択します。
4. パネル名の横にある削除アイコンを選択します。

パネルエディタの部品

このセクションでは、パネルエディタ画面の部品と、各部品に関連付けられたフィールド、オプション、タスクについて説明します。

ヘッダー

ヘッダーセクションには、パネルが属するダッシュボードの名前と一部のダッシュボードコマンドが表示されます。[戻る] 矢印を選択してダッシュボードに戻ることもできます。

ヘッダーの右側には、以下のオプションがあります。

- [ダッシュボード設定 (ギア) アイコン] — ダッシュボード設定へのアクセスを選択します。
- Discard (破棄) — 選択すると、ダッシュボードを最後に保存してから行ったすべての変更が破棄されます。
- [保存] — 選択すると、パネルエディタで行ったすべての変更を含めてダッシュボードが保存されます。
- [適用] — 選択すると、行った変更が適用され、パネルエディタが閉じてダッシュボードに戻ります。また、適用された変更を保持するためにダッシュボードも保存します。

視覚的表現のプレビュー

視覚的表現のプレビューセクションには、表示オプション、時間範囲コントロール、視覚的表現のプレビュー、該当する場合はパネルタイトル、軸、凡例が含まれます。

- [塗りつぶし] – 視覚的表現のプレビュー上のプレビューパートにある塗りつぶし可能なスペースが塗りつぶされます。サイドペインの幅またはボトムペインの高さを変更すると、視覚的表現の塗りつぶし可能なスペースが塗りつぶされます。
- [塗りつぶし] – 視覚的表現のプレビューで塗りつぶし可能なスペースが塗りつぶされますが、パネルのアスペクト比は保持されます。
- [正確] – 視覚的表現のプレビューが、ダッシュボードの実際のサイズで表示されます。十分なスペースがない場合、視覚的表現はアスペクト比が保持されたまま縮小されます。
- [時間範囲コントロール] – 詳細については、「時間範囲コントロール」を参照してください。

データセクション (最下部ペイン)

データセクションには、クエリの入力、データの変換、アラートルールの作成 (該当する場合) を行うタブがあります。

- [クエリ] タブ – データソースを選択し、ここにクエリを入力します。詳細については、「クエリ」を参照してください。
- [変換 タブ] – データ変換を適用します。詳細については、「変換」を参照してください。
- [アラート] タブ – アラートルールを入力します。詳細については、「Grafana アラート」を参照してください。

パネルとフィールドのオプション (サイドペイン)

データの視覚化方法のほぼすべての設定を行うタブがこのセクションにあります。視覚的表現ごとに利用できるタブは異なります。

これらのタブの機能については、次のトピックで説明します。

- パネルの追加
- 視覚化
- フィールドオプションとオーバーライド
- パネルリンク および データリンク (視覚的表現を他のリソースに接続する際に役立つ情報を掲載しています)

視覚化

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Amazon Managed Grafana には、さまざまなユースケースに合わせたさまざまな視覚的表現が提供されてます。以下のセクションでは、Amazon Managed Grafana で使用できる視覚的表現とその表示設定について紹介します。

トピック

- アラートリストパネル
- 棒グラフパネル
- 棒ゲージパネル
- 時計パネル
- ダッシュボードリストパネル
- ゲージパネル
- ジオマップパネル
- グラフパネル
- ヒートマップ
- ヒストグラムパネル
- ログパネル
- ニュースパネル
- ノードグラフパネル (ベータ)
- 円グラフパネル
- Plotly パネル
- サンキーパネル
- 散布図パネル
- 統計パネル
- 状態タイムラインパネル
- ステータス履歴パネル
- テーブルパネル
- テキストパネル
- 時系列パネル
- しきい値
- WindRose
- パネルの検査

- 計算リスト

アラートリストパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

アラートリストパネルには、ダッシュボードのアラートが表示されます。このリストは、現在の状態を表示、または最近の状態の変更を表示するように設定することができます。アラートの詳細については、「Grafana アラート」を参照してください。

この設定で、視覚化する内容を絞り込みます。

オプション

- [表示] — パネルに「現在のアラートの状態」または「最近のアラート状態の変更」のどちらを表示するかを選択します。
- [最大項目数] – 一覧表示するアラートの最大数を設定します。
- [ソート順序] – アラートをどのような順番で表示するかを選択します。
  - アルファベット (昇順) – アルファベット順
  - アルファベット (降順) – 逆アルファベット順
  - [重要性] – 以下を参照。1 が最も高い。
    - アラート中: 1
    - no_data (データなし): 2
    - 保留中: 3
    - OK: 4
    - 一時停止中: 5
- [このダッシュボードからのアラート] – アラートリストが含まれているダッシュボードからのみアラートを表示します。

フィルター

選択したクエリ、フォルダ、またはタグでアラートをフィルタリングする場合、次のオプションを使用できます。

- [アラート名] – アラート名クエリを入力します。
- ダッシュボードのタイトル – ダッシュボードのタイトルクエリを入力します。
- [フォルダ] – フォルダを選択します。選択したフォルダ内のダッシュボードからのアラートのみが表示されます。
- [ダッシュボードタグ] - 1 つ以上のタグを選択します。1 つ以上のタグを持つダッシュボードからのアラートのみが表示されます。

状態フィルター

このパネルに表示するアラートの状態を選択します。

- OK
- 一時停止中
- データなし
- 実行エラー
- [アラート]
- 保留中

棒グラフパネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このパネル表示では、カテゴリデータをグラフ化できます。

サポートされているデータ形式

サポートされるデータフレームは 1 つだけで、X 軸または Y 軸の項目として使用する文字列フィールドが少なくとも 1 つ、および数値フィールドが 1 つ以上必要です。以下は、データ形式の例です。

ブラウザ	市場シェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合は、パネルにグループ化されたバーが表示されます。

時系列または複数の結果セットの視覚化

複数の時系列またはテーブルがある場合は、まず結合を使用して結合するか、変換を減らす必要があります。例えば、複数の時系列があり、その最終値と最大値を比較する場合は、変換の [削減] を追加し、[計算] で [最大] と [Last] をオプションとして指定します。

棒グラフのオプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[Orientation] (向き)

- [自動] – Grafana は、パネルのディメンションに基づいてバーの向きを決定します。
- [水平] – X 軸をカテゴリ軸にします。
- [垂直] – Y 軸をカテゴリ軸にします。

[値の表示]

値がバーの上部または左側に表示されるかどうかを制御します。

- [自動] – スペースがある場合は値が表示されます。
- [常に表示] – 常に値を表示します。
- [常に非表示] – 常に値を表示しません。

[グループの幅] は、グループの幅を制御します。0=最小、1=最大幅です。

[Bar width] は、バーの幅を制御します。0=最小、1=最大幅です。

[線の幅]: 棒の線幅を制御します。

[塗りつぶしの不透明度]: 棒の塗りつぶし不透明度を制御します。

[グラデーションモード]: グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーション塗りつぶしなし。これはデフォルト設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。
- [色相] – グラデーション色は線の色相に基づいて生成されます。

[ツールチップモード] – 視覚化にカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、グラフ上でカーソルを合わせている系列が 1 つだけ表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – グラフを操作するときにツールチップを表示しません。

> **Note**
>
> オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] - 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。

- [右側] – グラフの右側。

[凡例計算] - 凡例に表示する標準計算を選択します。複数選択できます。

[テキストサイズ] – 棒グラフのテキストのサイズを変更する値を入力します。

[軸] – 次のフィールド設定を使用して、軸の表示方法を絞り込みます。一部のフィールドオプションは、編集するフィールドオプションボックスの外部をクリックするか、Enter キーを押すまで、視覚的表現には影響しません。

- [配置] – Y 軸の配置を設定します。
- [自動] – Grafana は、Y 軸を系列に自動的に割り当てます。異なるユニットを持つ系列が 2 つ以上ある場合、Grafana は左軸を最初のユニットに、右軸を次のユニットに割り当てます。
- [左] – 左側にすべての Y 軸を表示します。
- [右側] – 右側にすべての Y 軸を表示します。
- [非表示] – すべての Y 軸を非表示にします。
- [ラベル] – Y 軸テキストラベルを設定します。複数の Y 軸がある場合は、オーバーライドで異なるラベルを割り当てることができます。
- [幅] – 軸の固定幅を設定します。デフォルトでは、Grafana は軸の幅を動的に計算します。

  軸の幅を設定することで、異なる軸タイプのデータが同じ表示比率を共有できます。これにより、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。
- [ソフト最小値とソフト最大値] – Y 軸の制限をより適切に制御するために、ソフト最小値またはソフト最大値オプションを設定します。デフォルトでは、Grafana はデータセットに基づいて Y 軸の範囲を自動的に設定します。

  ソフト最小値とソフト最大値の設定により、データがほぼフラットなときに、小さな変動が大きな変動に見えるのを防ぎます。また、標準の最小値と最大値のフィールドオプションから派生したハード最小値または最大値により、定義されたポイントを超えたスパイクを切り取ることで、断続的なスパイクが有用な詳細を平坦化することを防げます。

  標準最小/最大値オプションを設定して、Y 軸のハード制限を定義できます。

棒ゲージパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

棒ゲージは、各フィールドのデータを 1 つの値にまとめて表示することで情報を簡素化します。Amazon Managed Grafana では、どのようにその値を計算するかを選択できます。

このパネルには、クエリの結果によって、1 つまたは複数の棒ゲージを表示できます (データが複数の系列、行、または列で構成されている場合など)。

データオプションとフィールドオプション

棒ゲージの可視化表現を使用すると、次のオプションを適用できます。

- 変換
- フィールドオプションとオーバーライド
- しきい値

表示オプション

次のオプションを使用して、視覚化する表示内容を絞り込みます。

- [表示] – Amazon Managed Grafana でのデータ表示方法を選択します。
  - [計算]– すべての行を元に計算した値を表示します。使用可能な計算のリストについては、「計算リスト」を参照してください。
  - [すべての値] - 行ごとに個別の統計を表示します。このオプションを選択すると、[制限]、または表示する最大行数も選択できます。
- [値] – Amazon Managed Grafana が多くのフィールドを 1 つの値にまとめるための reducer 関数を選択します。Value (値)リストを選択すると、関数と簡単な説明が表示されます。
- [Orientation] (向き) – 積み重ね方向を選択します。
  - [自動] – Amazon Managed Grafana が最適な方向を選択します。

  - [水平] – バーは水平方向に左から右に伸びます。
  - [垂直] – バーは垂直に上から下に伸びます。
- Display mode (表示モード) – 表示モードを選択します。
  - Gradient (グラデーション) – グラデーションを定義するしきい値レベルを選択します。
  - Retro LCD (レトロ LCD) – 指標を小さなセルに分け、点灯または消灯させて表示します。
  - [基本] – 指定したしきい値に基づいて 1 色で表示します。
- [塗りつぶされていない領域を表示] – バーの塗りつぶされていない領域を濃い灰色で表示する場合は、このオプションを選択します。このオプションは、レトロ LCD 表示モードでは使用できません。

時計パネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

時計パネルは、現在の時刻またはカウントダウンを表示します。毎秒更新されます。

- [モード] (モード) – デフォルトは時間です。カウントダウンを選択した場合、カウントダウンの期限を設定してカウントダウンを開始します。
- [12 時間または 24 時間表記] – 時間は 12 時間形式と 24 時間形式のいずれかを選択できます。
- [タイムゾーン] – タイムゾーンは、moment timezone のライブラリから提供されます。デフォルトでは、使用しているコンピュータのタイムゾーンが使用されます。
- [カウントダウン期限] – [モード]を[カウントダウン]に設定した場合、カウントダウンが終わる日時を指定します。
- [カウントダウン終了テキスト] – カウントダウンが終了するタイミングで表示されるテキストを指定します。
- 日付/時刻の書式設定オプション – フォントサイズ、フォントの太さ、日付/時刻の書式設定をカスタマイズします。カウントダウンを表示していて、秒単位のカウントダウンを表示したくない場合は、24 時間表記の場合は `hh:mm`、12 時間時表記の場合は `h:mm A` に時間形式を変更してください。オプションの完全な一覧については、「表示」を参照してください。

- [背景色] – 時計の背景色を選択します。

ダッシュボードリストパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードリストパネルには、他のダッシュボードへの動的リンクが表示されます。リストは、スター付きダッシュボード、最近表示されたダッシュボード、検索クエリ、ダッシュボードタグを使用できるよう設定が可能です。

ダッシュボードのロードごとに、このパネルはダッシュボードリストに対してクエリを実行し、常に最新の結果を表示します。

オプション

次のオプションを使用して、視覚化する表示内容を絞り込みます。

- [Starred] (スター付き) – スター付きダッシュボードをアルファベット順に表示します。
- [最近表示] – 最近表示したダッシュボードをアルファベット順に表示します。
- [検索] – 検索クエリまたはタグでダッシュボードを表示します。これを選択した場合、[クエリ] または [タグ] に少なくとも 1 つの値を入力する必要があります。
- [見出しの表示] – 選択したリスト選択 (スター付き、最近表示、検索) が見出しとして表示されます。
- [最大項目数] – セクションごとに一覧表示する項目の最大数を設定します。例えば、デフォルト値の 10 で、「スター付きダッシュボード」と「最近表示されたダッシュボード」を表示する場合、パネルには合計で最大 20 個のダッシュボードが表示され、各セクションに 10 個ずつ表示されます。

[検索]

以下のオプションは、検索オプションが選択されている場合にのみ適用されます。

- [Query] (クエリ) – 検索するクエリを入力します。クエリでは大文字と小文字が区別されず、部分的な値も受け入れられます。
- [フォルダ] – 表示するダッシュボードフォルダを選択します。
- [タグ] – 検索するタグを入力します。なお、既存のタグは入力時に表示されず、大文字と小文字が区別されます。

> **Note**
>
> 複数のタグと文字列が表示されると、ダッシュボードリストにすべての条件に一致するタグと文字列が表示されます。

ゲージパネル

> このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ゲージは、各シリーズ、列、または行に対してゲージを繰り返すことができる単一値のパネルです。

データオプションとフィールドオプション

ゲージの可視化を使用すると、次のオプションを適用できます。

- 変換
- フィールドオプションとオーバーライド
- しきい値

表示オプション

可視化を改善するには、次のオプションを使用します。

- [表示] – Amazon Managed Grafana でのデータ表示方法を選択します。

  - [計算]– すべての行を元に計算した値を表示します。使用可能な計算のリストについては、「計算リスト」を参照してください。
  - [すべての値] - 行ごとに個別の統計を表示します。このオプションを選択すると、[制限]、または表示する最大行数も選択できます。
- [Orientation] (向き) – 積み重ね方向を選択します。
  - 自動 – Amazon Managed Grafana は最適な方向であると考えるものを選択します。
  - [水平] – バーは水平方向に左から右に伸びます。
  - [垂直] – バーは垂直に上から下に伸びます。
- しきい値ラベルを表示 – しきい値を表示するかどうかを選択します。
- しきい値マーカを表示 – 内部ゲージ値幅の外にしきい値幅を表示するかどうかを選択します。

ジオマップパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ジオマップパネルの視覚的表現により、地理空間データを用いて世界地図を表示、カスタマイズできます。さまざまなオーバーレイスタイルとマップビュー設定を設定して、データの重要な位置情報に注目しやすくすることができます。

データレイヤー

Geomap プラグインのデータレイヤーを使用して、ベースマップ上にどのように地理空間データを表示するかを決定します。

場所

ジオマップパネルには、地理的データの情報源が必要です。このデータには 4 つのマッピングオプションがあり、データベースクエリを使用して取得します。

- [自動] でロケーションデータを自動的に検索します。クエリがデータフィールドの次のいずれかの名前に基づいている場合は、このオプションを使用します。

  - ジオハッシュ:「geohash」
  - 緯度:「latitude」、「lat」
  - 経度:「longitude」、「lng」、「lon」
  - ルックアップ:「lookup」
- Coords (座標): クエリに座標データが含めることを指定します。データベースクエリから、緯度と経度の数値データフィールドを選択するよう求められます。
- Geohash (ジオハッシュ): クエリがジオハッシュデータが含めることを指定します。データベースクエリから、ジオハッシュの文字列データフィールドを選択するよう求められます。
- ルックアップ: クエリに値にマッピングする必要がある口ケーション名データが含めることを指定します。データベースクエリと `gazetteer` からルックアップフィールドを選択するよう求められます。 `gazetteer` は、クエリされたデータを地理的ポイントにマッピングするために使用されるディレクトリです。

マーカーレイヤー

マーカーレイヤーを使用すると、データポイントを円、正方形、三角形、星などのさまざまな形のマーカーとして表示することができます。

- [マーカーの色] で、マーカーの色を設定します。デフォルトの「Fixed (固定)」サイズではすべてのポイントが 1 色になります。しきい値セクションで設定された「データポイント値」と「しきい値」に応じて、複数の色を使用できる代替オプションがあります。
- マーカーサイズ: マーカーのサイズを設定します。デフォルトは `Fixed size` で、すべてのマーカーサイズはデータポイントに関係なく同じになります。ただし、対応するデータポイントに円を拡大縮小するオプションもあります。マーカーレイヤーの拡大縮小の範囲を指定するため、マーカーサイズの `Min` と `Max` を設定する必要があります。
- Marker Shape (マーカーの形状): データポイントをさまざまな形で表示することができます。
  - [Circle] (円形)
  - [Square] (平方)
  - [Triangle] (三角形)
  - [Cross] (十字)
  - X
- [塗りつぶしの透明度]: 各マーカーの透明度を設定します。

ヒートマップレイヤー

ヒートマップレイヤーは、さまざまなデータポイントをクラスター化して、密度の異なる場所を視覚化します。ヒートマップレイヤーを追加するには、[データレイヤー] で [ヒートマップ] を選択します。

マーカーと同様に、どのデータポイントをどのように視覚化するかを決定するためのさまざまなオプションが用意されています。

- [重み値]: ヒートマップクラスターの強度を設定します。「固定値」を選択すると、すべてのデータポイントに対して一定の重みが適用されます。この値は 0～1 の範囲で設定します。マーカーと同様に、ドロップダウンには、データ値に応じて重み値を自動的にスケールする代替オプションがあります。
- [Radius] (半径): ヒートマップクラスターのサイズを設定します。
- [Blur] (ぼかし) では、各クラスターのぼかしの量を設定します。

グラフパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

グラフパネルは、線、ドットの軌跡、または一連のバーとしてレンダリングできます。このタイプのグラフは、ほぼすべての時系列データを表示するための十分な汎用性があります。

データオプションとフィールドオプション

グラフ視覚化を使用すると、次のオプションを適用できます。

- 変換
- アラート。これはアラートを設定できる唯一の視覚化タイプです。詳細については、「Grafana アラート」を参照してください。

- しきい値

表示オプション

視覚化を絞り込むためには、次の設定を使用します。

- [棒] – 値を棒グラフとして表示します。
- [線] – 値を折れ線グラフとして表示します。
- [線の幅] – 系列の線幅を指定します。デフォルトは 1 です。
- [階段] – 隣接するポイントを階段として描画します。
- [領域塗りつぶし] – シリーズの色塗りつぶし量を指定します。デフォルトは 1 で、0 はなしです。
- [塗りつぶしグラデーション] – エリア塗りつぶしのグラデーションの程度を指定します。デフォルトは 0 で、グラデーションはありません。10 は急なグラデーションです。
- [ポイント] – 値のポイントを表示します。
- [ポイント半径] – ポイントの大きさを制御します。
- [アラートしきい値] – アラートしきい値とリージョンをパネルに表示します。

スタッキングおよび null 値

- [スタック] – 各シリーズが別のシリーズの上にスタックされます。
- [パーセント] – 各系列が全系列の合計のパーセンテージとして描画されます。このオプションは [スタック] が選択されている場合に使用できます。
- [Null 値] – null 値の表示方法を指定します。これは重要な設定です。以下の注記を参照してください。
  - [接続] – 系列にギャップがある場合、つまり 1 つ以上の null 値がある場合、線はギャップをスキップして次の null 以外の値に接続します。
  - [null] - 系列にギャップがある場合、つまり null 値の場合、グラフの線は分割されてギャップが表示されます。これはデフォルトの設定です。
  - [ゼロとしての null] – シリーズにギャップがある場合、つまり null 値は、グラフパネルにゼロ値として表示されます。

> **Important**
>
> サーバーの CPU 負荷をモニタリングしているときに負荷が 100% に達すると、サーバーはロックアップされ、統計情報を送信するエージェントは負荷統計を収集できなくなりま

す。これによりメトリクスにギャップが生じ、デフォルトの null 設定を使用すると Amazon Managed Grafana にはギャップが表示され、何かが間違っていることを示します。これが[接続]に設定されていると、このシグナルを見逃しやすくなります。

ホバー時のツールヒント

これらの設定を使用して、グラフの視覚化上にカーソルを置いたときに表示されるツールヒントの外観を変更します。

- [モード] - カーソルのツールチップに表示される系列数を決定します。
  - [すべての系列] – カーソルのツールチップには、グラフ内の系列がすべて表示されます。Grafana ワークスペースは、ツールヒント内のシリーズリストで、一時停止しているシリーズを太字で強調表示します。
  - [単独] – ツールチップには、カーソルを合わせている系列が 1 つだけ表示されます。
- [ソート順序] – [すべての系列] モードを選択した場合、カーソルのツールチップにおいて系列の順序を並べ替えます。グラフで一時停止すると、Amazon Managed Grafana は線に関連付けられた値を表示します。一般的に、ユーザーは最大値や最小値に最も関心があります。これらの値をソートすると、目的のデータを簡単に見つけることができます。
  - [なし] – ツールチップ内のシリーズ順序は、クエリのソート順序によって決まります。例えば、シリーズ名によりシリーズをアルファベット順にソートできます。
  - [増加] – ホバー時のツールチップ内のシリーズは値が昇順にソートされ、リストの上部には最小値が表示されます。
  - [減少] – ホバー時のツールチップ内のシリーズは値が降順にソートされ、リストの上部に最大値が表示されます。

シリーズオーバーライド

シリーズオーバーライドを使用すると、グラフパネル内の系列を他の系列とは異なるレンダリングにできます。表示オプションは、シリーズごとに、または正規表現ルールを使用してカスタマイズできます。例えば、1 つのシリーズで線の幅を太くして目立つようにしたり、右の Y 軸に移動したりできます。

シリーズオーバーライドは複数追加できます。

シリーズオーバーライドの追加方法

1. [シリーズオーバーライドの追加] を選択します。

2. [エイリアスまたは正規表現] で、シリーズを入力または選択します。フィールドを選択すると使用可能なシリーズのリストが表示されます。

   例えば、`/Network.*/` は `Network out` および `Network in` という名前の 2 つのシリーズと一致します。

3. [+] を選択してからシリーズに適用するスタイルを選択します。各エントリには複数のスタイルを追加できます。

- [棒] – 系列を棒グラフとして表示します。
- [折れ線] – シリーズを折れ線グラフとして表示します。
- [折れ線塗りつぶし] – 領域塗りつぶしで折れ線グラフを表示します。
- [塗りつぶしグラデーション] – 領域塗りつぶしグラデーション量を指定します。
- [線の幅] – 線幅を設定します。
- [Null ポイントモード] – このオプションを使用して null 値を無視、またはゼロに置き換えます。これはデータ内ののギャップを無視する場合に重要です。
- [下を塗りつぶし] – 2 つのシリーズ間の領域を塗りつぶします。
- [階段線] — シリーズを階段線として表示します。
- [ダッシュ] – ダッシュで線を表示します。
- [シリーズを非表示] – シリーズを非表示にします。
- [ダッシュ長さ] – 線内のダッシュの長さを設定します。
- [ダッシュスペース] – 線内のダッシュ間のスペースの長さを設定します。
- [ポイント] – 個別のポイントとして系列を表示します。
- [ポイント半径] – ポイントのレンダリング半径を設定します。
- [スタック] – シリーズのスタックグループを設定します。
- [色] – シリーズの色を設定します。
- [Y 軸] – 系列の Y 軸を設定します。
- [Z インデックス] – 系列の z インデックス (レンダリング順序) を設定します。このオプションは、棒グラフや面グラフなどの、さまざまなスタイルをオーバーレイする場合に重要です。
- [変換] – 負の値に変換して Y 軸の下にレンダリングします。
- [凡例] – 凡例内にシリーズを表示するかどうかを制御します。
- [ツールヒントで非表示] – グラフのツールヒントにシリーズを表示するかどうかを制御します。

[Axes] (軸)

これらのオプションを使用して、視覚化内の軸表示を制御します。

[Left Y]/[Right Y]

オプションは両方の Y 軸で同じです。

- [表示] – 軸の表示または非表示を選択します。
- [単位] – Y 値の表示単位を選択します。
- [スケール] – Y 値に使用するスケールを、[線形]、または [対数] から選択します。デフォルトは [線形] です。
- [Y 最小] – Y の最小値です。デフォルトは [自動] です。
- [Y 最大] – Y の最大値です。デフォルトは [自動] です。
- [小数] – Y 値に表示される小数点以下の桁数を定義します。デフォルトは [自動] です。
- [ラベル] – Y 軸のラベルを指定します。デフォルトは [“",] です

Y 軸

- [配置] – 左右の Y 軸を値で配置します。デフォルトは [未チェック/偽] です。
- [レベル] – Y=0 から始まる、左右の Y 軸の配置に使用する値を入力します。デフォルトは 0 です。このオプションは、[配置] が選択されている場合に使用できます。

X 軸

- [表示] – 軸の表示または非表示を選択します。
- [モード] – 表示モードではグラフパネルの視覚化が完全に変更されます。3 つのパネルを 1 つにしたようなものです。メインモードは、X 軸の時間による時系列モードです。他の 2 つのモードは、時間とヒストグラムのモードではなく、X 軸上のシリーズ使用する基本的な棒グラフモードです。
  - [時間] (デフォルト) – X 軸は時間を表し、データは時間 (時間、分など) でグループ化されます。
  - [系列] – データは時間ではなく系列でグループ化されます。Y 軸は引き続き値を表します。
    - [値] – これは値のために使用する集計タイプです。デフォルトは [合計] (値を合計) です。
  - [ヒストグラム] – このオプションではグラフをヒストグラムに変換します。ヒストグラムは数値を範囲にグループ化する棒グラフの一種で、バケットやビンとも呼ばれます。背の高い棒は、より多くのデータがその範囲内にあることを示します。

ヒストグラムの詳細については、「ヒストグラムおよびヒートマップの概要」を参照してください。

- [バケット] – 値をグループ化するためのバケットの数を設定します。空にすると、Amazon Managed Grafana は適切な数のバケットを計算しようとします。
- [X 最小] – この下限未満のヒストグラムから値をフィルタリングします。
- [X 最大] – この上限を超える値をフィルタリングします。

凡例

これらの設定を使用して、グラフ内の凡例の表示方法を改善します。

オプション

- [表示] – クリアすると凡例は非表示になります。デフォルトが選択されます (真)。
- [テーブルとして] – 選択すると凡例がテーブルに表示されます。デフォルトがチェックされます (真)。
- [右側] - 選択すると右側に凡例が表示されます。
- [幅] – 凡例の最小幅をピクセル単位で入力します。このオプションは、[右側] を選択した場合に使用できます。

値

凡例名の横に追加の値を表示できます。

- [最小] – メトリクスクエリから返される最小値です。
- [最大] – メトリクスクエリから返される最大値です。
- 平均 – メトリクスクエリから返される平均値です。
- [現在] – メトリクスクエリから返された最後の値です。
- [合計] – メトリクスクエリから返されるすべての値の合計です。
- [小数] – 凡例値およびグラフカーソルのツールチップに表示される小数点以下の桁数です。

Amazon Managed Grafana はクライアント側の凡例値を計算します。これらの凡例値は、メトリクスクエリが使用している集計またはポイント統合のタイプによって異なります。上記の凡例値がすべて同時に正確になることはありません。

例えば、おそらく平均をアグリゲーターとして使用する、リクエスト/秒の割合などをプロットする場合、凡例の合計はリクエストの合計数を表しません。これは Amazon Managed Grafana が受信したすべてのデータポイントの合計にすぎません。

シリーズの非表示

メトリクスクエリからのシリーズの値すべてが特定の値であるときに、シリーズを非表示にします。

- [null のみを使用] – Value=null (デフォルトは未チェック)
- [ゼロのみを使用] – Value=ゼロ (デフォルトは未チェック)

時間リージョン

週末、営業時間、勤務時間外など、グラフ上の特定の時間リージョンを強調表示して見やすくできます。設定されたすべての時間リージョンは UTC 時間です。

ヒートマップ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ヒートマップパネルの視覚化により、ヒストグラムを経時的に表示できます。ヒストグラムの詳細については、「ヒストグラムおよびヒートマップの概要」を参照してください。

軸オプション

これらの設定を使用して、視覚化での軸の表示方法を設定します。

Y 軸

- [単位] – Y 軸の値の表示単位
- [Scale] – Y 軸の値に使用する目盛
  - [線形] – 線形目盛。
  - [log (base 2)] – 底が 2 の対数目盛

  - [log (base 10)] – 底が 10 の対数目盛
  - [log (base 32)] – 底が 32 の対数目盛
  - [log (base 1024)] – 底が 1024 の対数目盛
- [Y 最小] – y の最小値 (デフォルトは自動)
- [Y 最大] – y の最大値 (デフォルトは自動)
- [小数] – Y 軸の値をレンダリングする際の小数点以下の桁数 (デフォルトは自動)

バケット

> **Note**
>
> データ形式が時系列バケットの場合、このセクションは使用できません。

- [Y 軸バケット] – Y 軸が分割されるバケットの数。
- [サイズ] – 各 y 軸バケットのサイズ ([目盛]が線形の場合にのみ表示)。このオプションは、Y 軸バケットよりも優先されます。
- [分割係数] – (目盛が 対数 (底 2) 以上の場合にのみ表示)。デフォルトでは、Amazon Managed Grafana は対数の底で y 値を分割します。このオプションを使用すると、各デフォルトバケットを指定された数のバケットに分割できます。
- [X 軸バケット] – X 軸が分割されるバケットの数。
- [サイズ] – 各 X 軸バケットのサイズ。数または時間間隔 (10 秒、5 分、1 時間など)。サポートされている間隔: ms、s、m、h、d、w、M、y。このオプションは、X 軸バケットよりも優先されます。

バケット境界

データ形式が時系列バケットの場合、データソースはバケット境界を表す名前の系列を返します。ただし、データソースによっては、境界を上下にすることができます。このオプションを使用して、境界線のタイプを調整できます。自動設定にされている場合、パネルのデータソースタイプに基づいて境界オプションが選択されます。

[バケットサイズ]

バケット数とサイズオプションは、Amazon Managed Grafana がヒートマップ内の各セルのサイズを計算するために使用されます。バケットサイズは、バケット数 (最初の入力ボックス) またはサイ

ズ間隔を指定することで定義できます。Y 軸の場合、サイズ間隔は単なる値です。X バケットの場合、サイズ入力で時間間隔を指定できます。例えば、時間範囲を1hに設定することができます。これにより、X 軸でセルの幅が 1 時間になります。

データ形式

[フォーマット] リストでオプションを選択します。

- [時系列] – Amazon Managed Grafana は、すべての時系列値を調べることでバケット化を実行します。バケットのサイズと間隔は、[バケット]オプションで設定されます。
- [時系列バケット] – 各時系列は元々、Y 軸バケットを表します。時系列名 (エイリアス) は、バケットの上限間隔または下限間隔を表す数値である必要があります。Grafana ワークスペースはバケット化を行わないため、バケットサイズオプションは非表示になります。

表示オプション

この設定で、視覚化する内容を絞り込みます。

カラー

色のスペクトルは、値の数 (各バケット内) と各バケットに割り当てられた色間とのマッピングを制御します。スペクトルの左端の色は最低数を表し、右端の色は最大数を表します。一部のカラースキームは、ライトテーマを使用するときに自動的に反転されます。

また、カラーモードを [不透明度] に変更することもできます。この場合、色は変わりませんが、不透明度はバケット数に応じて変化します。

- モード
  - [不透明度] – セルの不透明度で表されるバケット値。不透明セルは最大値を意味します。
    - [Color] – セルのベース色。
    - [Scale] (目盛) – バケット値を不透明度にマッピングするための目盛。
      - [線形] – 線形目盛。バケット値は不透明度に対して直線的にマッピングされます。
      - [sqrt] (平方根) – 対数目盛。セルの不透明度は `value ^ k` として計算され、k は設定された指数値です。指数が 1 より小さい場合、対数スケールが表示されます。指数が 1 より大きい場合、指数スケールが表示されます。1 の場合、目盛は線形と同じになります。
    - [Exponent] – 指数の値 (`0` より大きい場合)。
  - [spectrum] (スペクトル) – セルの色で表されるバケット値。
    - [Scheme] – モードがスペクトルの場合、カラースキームを選択します。

カラースケール

デフォルトでは、Amazon Managed Grafana は最小および最大バケット値に基づいてセルの色を計算します。最小値と最大値を設定することで、これらの値を上書きできます。バケット値は z 軸、最小値と最大値はそれぞれ Z の最小値と Z の最大値と考えてください。

- [最小] – セルの色の計算に使用される最小値。バケット値が最小値より小さい場合、最小値の色にマッピングされます。デフォルトは `series min value` です。
- [最大] – セルの色の計算に使用される最大値。バケット値が最大値より大きい場合、最大値の色にマッピングされます。デフォルトは `series max value` です。

凡例

視覚化にヒートマップの凡例を表示するかどうかを選択します。

バケット

- [Hide zero] – 値がゼロのセルは表示しません。
- [Space] – セル間のスペースをピクセル単位で設定します。デフォルトは 1 ピクセルです。
- [Round] – セルの真円度をピクセル単位で設定します。デフォルトは 0 です。

Tooltip (ツールチップ)

- [Show tooltip] - ヒートマップのツールチップを表示します。
- [Histogram] – ツールチップに Y 軸のヒストグラムを表示します。ヒストグラムは、特定のタイムスタンプにおけるバケット値の分布を表します。
- [小数] – バケット値をレンダリングする際の小数点以下の桁数を設定します (デフォルトは自動)。

ヒストグラムパネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ヒストグラムの視覚化は値の分布を計算して棒グラフとして表示します。Y 軸と各バーの高さは各括弧に含まれる値の数を表し、X 軸は値の範囲を表します。

ヒストグラムの視覚化では、1 つ以上の数値フィールドを持つ時系列とテーブル結果をサポートしています。

表示オプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[バケットサイズ]

バケットのサイズです。自動バケットサイズ設定 (全範囲の 10% まで) では、この値は空のままにします。

[バケットオフセット]

最初のバケットを 0 で始めない場合です。ゼロ以外のオフセットでは集計ウィンドウがシフトします。例えば、デフォルトの 0 オフセットの 0～5、5～10、10～15 の 5 サイズのバケットは、オフセット 2 の 2～7、7～12、12～17 になる場合があります。この場合、オフセット 0、5、10 であれば事実上何も実行されません。通常、このオプションは自動ではなく、明示的に定義されたバケットサイズで使用します。この設定が作用するためには、オフセット量を 0 より大きく、バケットサイズより小さくする必要があります。この範囲外の値には、この範囲内の値と同じ効果があります。

[系列の結合]

これにより、すべての系列とフィールドが結合されたヒストグラムにマージされます。

[線の幅]: 棒の線幅を制御します。

[塗りつぶしの不透明度]: 棒の塗りつぶし不透明度を制御します。

[グラデーションモード]: グラデーション塗りつぶしモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。グラデーションの外観は[塗りつぶしの不透明度]設定の影響を受けます。

- [なし] – グラデーション塗りつぶしなし。これはデフォルト設定です。
- [不透明度] – グラデーションの透明度は Y 軸の値に基づいて計算されます。Y 軸の値とともに塗りつぶしの不透明度が増加します。
- [色相] – グラデーション色は線の色相に基づいて生成されます。

[ツールチップモード] グラフにカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、カーソルを合わせている系列のみが表示されます。
- [すべて] – ホバー時のツールチップには視覚化のすべてのシリーズが表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – ツールチップを表示しません。

> **Note**
>
> オーバーライドを使用して、ツールチップから個々の系列を非表示にします。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

[凡例計算]

凡例に表示する計算を選択します。詳細については、「計算リスト」を参照してください。

ログパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ログパネルの視覚化では、Elastic、Influx、Loki などのログをサポートするデータソースのログ行が表示されます。通常、グラフパネルの横のこのパネルを使用して、関連するプロセスのログ出力を表示します。

ログパネルには、[Query] タブに入力されたクエリの結果が表示されます。複数のクエリ結果がマージされ、時間別にソートされます。表示可能な行よりも多くの行がデータソースから返される場合は、パネル内でスクロールできます。

レンダリングされる行数を制限するには、[クエリオプション]で [最大データポイント数] の設定を使用できます。設定しない場合、データソースは通常デフォルトの制限を適用します。

表示オプション

以下の設定を使用して、視覚化する内容を絞り込みます。

- [時間] – 時間列を表示または非表示にします。これはデータソースから報告されたログ行と関連するタイムスタンプです。
- [一意のラベル] – 一意のラベル列を表示または非表示にします。この列には、共通ではないラベルのみが表示されます。
- [行の折り返し] – 行の折り返しを切り替えます。
- [Order] (順序) – 結果を降順または昇順で表示します。デフォルトは [降順] で、最新のログが最初に表示されます。[昇順] に設定すると最も古いログ行が最初に表示されます。

ニュースパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このパネルには RSS フィードが表示されます。デフォルトでは、Grafana Labs のブログ記事が表示されます。

[表示] セクションの [URL] フィールドに、RSS フィードの URL を入力します。このパネルタイプは他のクエリを受け入れません。

ノードグラフパネル (ベータ)

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ノードグラフパネルは有向グラフまたはネットワークを視覚化します。有向フォースレイアウトを使用してノードを効果的に配置し、複雑なインフラストラクチャマップ、階層、または実行図の表示に役立ちます。

データ要件

ノードグラフのパネルでは、ノードとエッジを表示するためにデータの特定の形状が必要です。このパネルでは、すべてのデータソースやクエリを視覚化できるわけではありません。

ノードグラフの視覚化は、ノードおよびエッジで構成されます。

- あるノードは円として表示されます。あるノードは、アプリケーション、サービス、またはアプリケーションの観点から関連するその他のものを表す場合があります。
- あるエッジは、2 つのノードを接続する線として表示されます。接続は、リクエスト、実行、または 2 つのノード間の他の関係である場合があります。

ノード

通常、ノードには、ノード内の 2 つの統計値とノードのすぐ下の 2 つの識別子、通常は名前とタイプが表示されます。ノードでは、ノードの周囲に一連の別の値を色付きの円として表示することもできます。異なる色のセクションは、合計が 1 になる異なる値を表します。例えば、円の赤い部分によってエラーの割合を表すことができます。

ノードを選択すると表示されるコンテキストメニューには、さらに詳細を表示できます。Grafana ワークスペースの他の部分または外部リンクをターゲットにできる追加のリンクを、コンテキストメニューに含めることもできます。

エッジ

エッジでは、エッジにカーソルを合わせると統計を表示することもできます。ノードと同様に、エッジを選択すると、追加の詳細とリンクを含むコンテキストメニューを開けます。

この視覚化をサポートする最初のデータソースは、サービスマップ機能の AWS X-Ray データソースです。詳細については、「AWS X-Ray データソースに接続する」を参照してください。

ノードを選択すると表示されるコンテキストメニューには、さらに詳細を表示できます。Grafana ワークスペースの他の部分または外部リンクをターゲットにできる追加のリンクを、コンテキストメニューに含めることもできます。

ノードグラフの操作方法

ノードグラフ内でパンするには、ノードまたはエッジの外部を選択してマウスをドラッグします。

ノードグラフの左上隅にあるボタンを使用するとズームできます。

円グラフパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

円グラフには、1 つ以上のクエリから誘導されたシリーズ、またはシリーズ内の値が、円グラフのスライスで相互に関連付けられて表示されます。スライスの円弧長、面積、中心角度はすべての値の合

計と関係するため、すべてスライス値に比例します。この種類のグラフは、一連の小規模な値を見た目がきれいな形ですばやく比較する場合に最適です。

円グラフの可視化を使用すると、次のオプションを適用できます。

- 変換.
- フィールドオプションとオーバーライド.
- しきい値.

オプション

次のオプションを使用して、視覚化を絞り込みます。

- 表示 – 表示する情報の量を選択します。[計算] 各値を系列ごとに単一の値に減らします。[すべての値] 1 つの系列におけるすべての値を表示します。
- [計算] - を選択すると、各系列を削減する[計算]を選択できます。利用可能な計算の詳細については、「計算リスト」を参照してください。
- 制限 – 1 つのシリーズのすべての値を表示する場合、表示される値の数が制限されます。
- フィールド – 視覚化に表示するフィールドを選択します。
  - 数値フィールド – 数値を持つすべてのフィールド。
  - [すべてのフィールド] – 変換によって削除されないすべてのフィールド。
  - [時間] – 時間値を持つすべてのフィールド。

ラベル

円グラフに表示するラベルを選択します。複数選べます。

- [名前] – 系列名またはフィールド名。
- [パーセント] – 全体の割合。
- [Value] – 未加工の数値。

ラベルはグラフ部分の上に白で表示されます。グラフの色を濃く選択して、より見やすくする必要がある場合があります。長い名前や数字は文字切れする可能性があります。

凡例の配置および値

凡例を表示する場所を選択します。

- 下部 – グラフの下。
- 右 – グラフの右側。

凡例に表示する値は、複数選択できます。[パーセント] は全体のパーセンテージ、[値] は未加工の数値です。

Plotly パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Plotly パネルは、オープンソースの javascript グラフ作成ライブラリである Plotly を使用してグラフをレンダリングします。

データ、レイアウト、および設定 フィールドは、「Plotly ドキュメント」で説明されている一般的なパラメータと一致します。これらは JSON 形式である必要があります。

データソースによって提供されるデータは、Plotly グラフに挿入される前に、ユーザー定義スクリプトを介して変換できます。スクリプトには 2 つの引数が含まれます。

- `data` – データソースから返されるデータ。
- `variables` – 現在のダッシュボードに Grafana 変数を含むオブジェクト (ユーザー変数と、グローバル変数である `__from`、`__to`、`__interval`、`__interval_ms`)。

スクリプトは、`data`、`layout`、`config`、`frames` のプロパティから 1 つ以上を有するオブジェクトを返す必要があります。以下に例を示します。

```
let x = data.series[0].fields[0].values;
let y = data.series[0].fields[1].values;
let series = {
    x: x,
    y: y,
```

```
    name: variables.name, // where 'name' is the name of a Grafana dashboard variable
};

return {
    data: [series],
    config: {
        displayModeBar: false,
    },
};
```

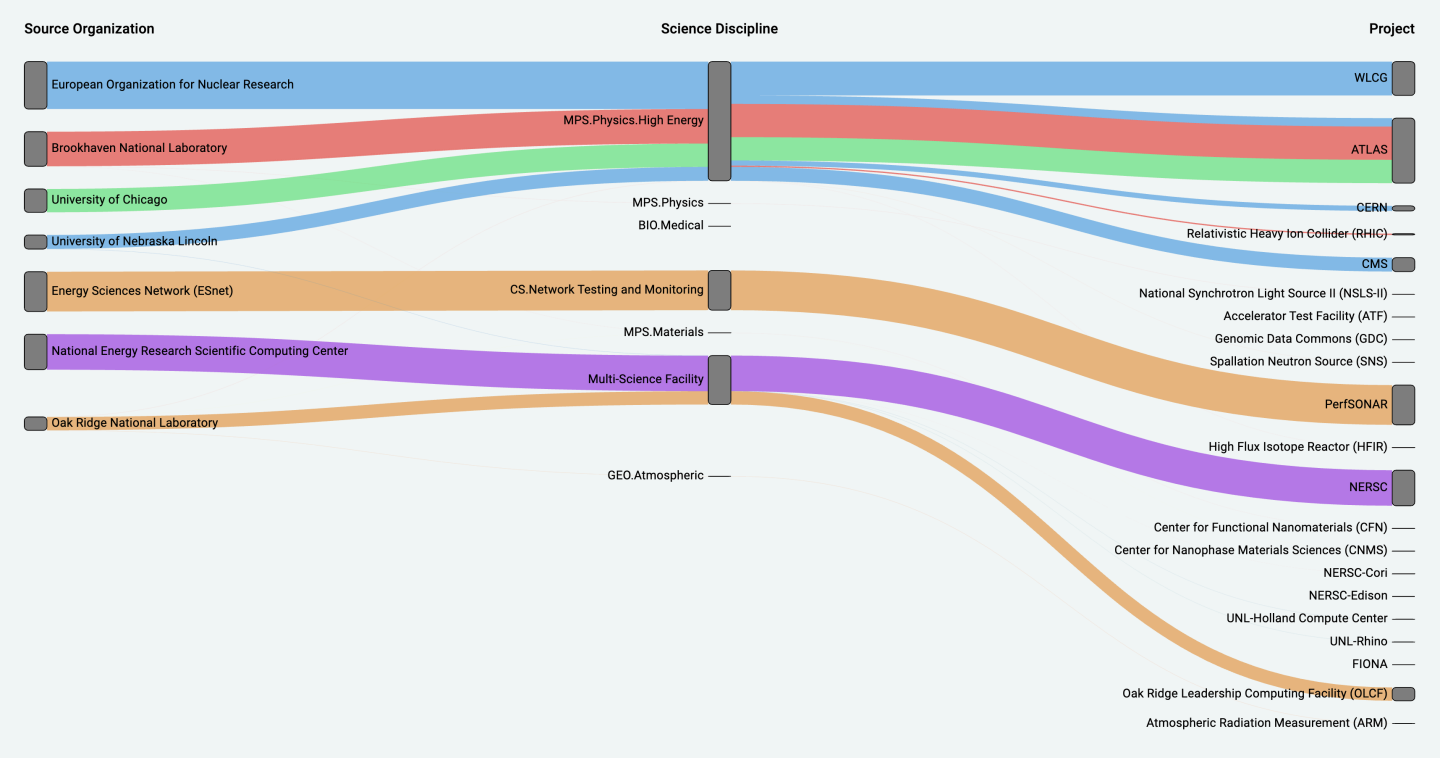
データ、レイアウト、および 設定 フィールドで指定されたスクリプトと JSON によって返されるオブジェクトはマージされます (ディープマージ)。

スクリプトが指定されていない場合、パネルはデータ、レイアウト、および 設定 フィールドのみを使用します。

サンキーパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

サンキーパネルにはサンキーダイアグラムが表示されます。サンキーダイアグラムは、選択したメトリクスに比例したフローの幅で、フローデータを視覚化するのに適したものです。次の図は、ソースとターゲットの 2 つのグループを含むサンキーダイアグラムを示しています。



仕組み

サンキーパネルには、フローのソースとターゲットの 2 つ以上の列のデータが必要です。クエリは、データを少なくとも 2 つのグループにグループ化する必要があります。パネルは、データポイントの最初の列からクエリの順序で最後の列へのリンクを描画します。リンクの厚さは、クエリ内のメトリクスによって割り当てられた値に比例します。

カスタマイズ

- リンク – 現在、リンクの色にはマルチまたはシングルの 2 つのオプションがあります。デフォルトでは、マルチカラーです。リンクの単一色を選択するには、[単一リンクの色のみ]オプションを切り替え、Grafana のカラーピッカーから色を選択します。
- [ノード] – [ノードの色] オプションを変更することで、長方形ノードの色を変更できます。
- [ノード幅] – ノードの幅は、[ノード幅] スライダーで調整するか、入力ボックスに数値を入力することで調整できます。数値は整数である必要があります。
- [ノードパディング] – ノード間の垂直パディングは、[ノードパディング] スライダーで調整するか、入力ボックスに数値を入力することで調整できます。数値は整数である必要があります。リンクが細すぎる場合は、この数値を調整してみてください。
- ヘッダー – 列ヘッダーは、エディタパネルの [表示名] オーバーライドを使用して変更できます。[テキスト色] で選択した色と同じ色になります

- [サンキーレイアウト] – サンキーリンクのレイアウトは、[レイアウト反復] スライダーを使用して微調整できます。この数は整数でなければならず、レイアウトの生成に使用される緩和反復の数です。

散布図パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

散布図パネルは、他のグラフパネルよりもシンプルなインターフェイスを持つテーブルデータの X/Y 散布図を示しています。グラフパネルとは異なり、散布図パネルではデータが時系列に含まれる必要はありません。散布図パネルには、データの 2 つ以上の数値列を含むテーブル形式のデータセットが必要です。

これらのいずれかを X 軸に割り当てることができます。1 つ以上を一連の Y 軸値に割り当てることができ、その結果データは一連の点としてプロットされます。各系列では、オプションで、統計に基づく複数のベストフィットのいずれかを使用して回帰線を表示することもできます。

散布図の作成

次の手順では、散布図パネルを使用して散布図を作成する方法について説明します。この例では、次の表のように、年齢別の男の子と女の子の平均身長を示す数値 `Age`、`Boys`、`Girls` の 3 つの列を持つ `HEIGHT` で呼び出されるデータがあると想定します。

年齢	男の子の身長	女の子の身長
5	109.7	109.5
6	115.6	115.4
7	121.1	120.8
8	126.3	126

年齢	男の子の身長	女の子の身長
9	131.3	131.3
10	136.2	137.1
11	141.2	143.2
12	147	148.7
13	153.6	152.6
14	159.9	155.1
15	164.4	156.7
16	167.3	157.6
17	169	158
18	170	158.3
19	170.8	158.6

散布図パネルを使用して散布図を作成するには

1. Grafana ダッシュボードで、[パネルの追加] を選択します。パネルの追加の詳細については、「パネルの追加」を参照してください。
2. [クエリ] には、必要なデータを返すクエリを書き込みます。この場合、`SELECT * FROM HEIGHT` などのクエリを使用します。
3. [散布図] での視覚化を選択します。

これにより、最初の列を X 軸として使用し、他の数値列を Y 軸として使用して散布図が作成されます。

設定オプション

散布図パネルには、次の 4 つのカスタム設定オプションが用意されています。

- [X 軸] – X 軸として使用するフィールド、軸の範囲とタイトル、および表示情報を選択できます。

- [Y 軸] – 各フィールドの表示オプション、軸の範囲とタイトル情報など、Y 軸に表示するフィールドを選択できます。また、各フィールドの回帰線を表示することもできます。回帰線設定の詳細については、以下の情報を参照してください。
- [凡例] – パネルの凡例をオンまたはオフにしたり、凡例内のテキストサイズを選択したりできます。
- [表示] – グリッドの色や境界スタイルなど、グラフの他のオプションを設定できます。

回帰線の設定

各 Y 軸データセットには、個々の点に加えて線を表示できます。線タイプには 5 つのオプションがあります。

- [なし] – 回帰線を表示しません。
- [単純] – データセットポイントを接続する回帰線を表示します。
- [線形] – 最小二乗法、ベストフィット方式を使用して直線を表示します。
- [指数] – 指数のベストフィット回帰線を表示します。
- [累乗] – 累乗のベストフィット回帰線を表示します。

統計パネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

統計パネルにはオプションのグラフスパークラインを含む 1 つの大きな統計値が表示されます。しきい値を使用すると背景または値の色を制御できます。

デフォルトでは、統計パネルに次のいずれかが表示されます。

- 単一のシリーズまたはフィールドの値のみ。
- 複数の系列またはフィールドの値と名前の両方。

テキストモードオプションを使用すると、テキストを表示するかどうかを制御できます。

データオプションとフィールドオプション

統計の可視化を使用すると、次のオプションを適用できます。

- 変換.
- フィールドオプションとオーバーライド.
- しきい値.

自動レイアウト調整

パネルは、ダッシュボードで使用可能な幅と高さに応じてレイアウトを自動的に調整します。パネルが小さすぎると、グラフ (スパークライン) が自動的に非表示になります。

表示オプション

次のオプションを使用して、視覚化する表示内容を絞り込みます。

- [表示] – Amazon Managed Grafana でのデータ表示方法を選択します。
  - [計算] – すべての行を元に計算した値を表示します。
    - 計算 - 適用する計算を選択します。利用可能な計算の詳細については、「計算リスト」を参照してください。
  - すべての値 — 行ごとに個別の統計を表示します。
    - [制限] – 表示する行の最大数を指定します。
- [フィールド] – このパネルに含めるフィールド名またはフィールドタイプ ([すべてのフィールド] または [数値フィールド] を含む) を選択します。
- [値] – Amazon Managed Grafana が多くのフィールドを 1 つの値にまとめるための reducer 関数を選択します。Value (値)リストを選択すると、関数と簡単な説明が表示されます。
- [Orientation] (向き) – 積み重ね方向を選択します。
  - 自動 – Amazon Managed Grafana は最適な方向であると考えるものを選択します。
  - [水平] – バーは水平方向に左から右に伸びます。
  - [垂直] – バーは垂直に上から下に伸びます。
- テキストモード – [テキストモード] オプションを使用すると、パネルが表示するテキストを制御できます。名前と色のみが重要で、値が重要でない場合は、[テキストモード] を [名前] に変更します。値が色を決定するために使用され、ツールヒントに表示されます。

  - [自動] – データに複数のシリーズまたはフィールドが含まれている場合は、名前と値の両方を表示します。
  - [値] – 値のみを表示し、名前は表示しません。名前がツールヒントに表示されます。
  - 値と名前 – 常に値と名前を表示します。
  - 名前 – 値の代わりに名前を表示します。値がツールヒントに表示されます。
  - [なし] – 何も表示しません (空)。名前と値がツールヒントに表示されます。
- カラーモード – カラーモードを選択します。
  - [値] – 値とグラフ領域のみを色付けします。
  - [Background] (背景) – 背景も色付けします。
- グラフモード – グラフモードを選択します。
  - [なし] – グラフを非表示にし値のみを表示します。
  - [エリア] – 値の下に面グラフを表示します。このオプションではクエリが時間列を返す必要があります。
- アライメントモード – アライメントモードを選択します。
  - 自動 – 値が 1 つだけ表示されている場合 (繰り返しなし)、値は中央に配置されます。複数の系列または行が表示されている場合、値は左揃えになります。
  - [Center] (中央) – 統計値は中央にあります。

状態タイムラインパネル

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

状態タイムラインパネルの視覚化では、時間の経過に伴う個別の状態の変化が表示されます。各フィールドまたは系列は、一意の水平バンドとしてレンダリングされます。状態領域は、値の有無にかかわらずレンダリングできます。このパネルは、文字列またはブール値の状態とうまく連携しますが、時系列でも使用できます。時系列で使用すると、しきい値を使用して数値を離散状態の領域に変換します。

状態タイムラインオプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[Merge equal consecutive values] (等しい連続値をマージする)

Grafana は、互いに隣接する同じ値をマージするかどうかを制御します。

[値の表示]

値が状態領域内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[値の整列]

状態領域内の値のアライメントを制御します。

[Row height] (行の高さ)

行間のスペースを制御します。1 = スペースなし = 0.5 = 50% のスペース。

[線の幅]

状態領域の線幅を制御します。

[塗りつぶしの不透明度]

状態領域の不透明度を制御します。

値のマッピング

ブール値または文字列値に色を割り当てるには、値の配置 を使用します。

しきい値を含む時系列データ

パネルは時系列データでも使用できます。この場合、しきい値を使用して時系列を個別の色付き状態領域に変換します。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

ステータス履歴パネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ステータス履歴の視覚化は、時間の経過に伴う周期的な状態を表示します。各フィールドまたは系列は、横の列としてレンダリングされます。ボックスがレンダリングされ、各値を中央に配置します。

ステータス履歴の視覚化は、文字列、ブール値、数値フィールドまたは時系列で機能します。時間フィールドは必須です。値マッピングを使用して、文字列の色付け、またはテキスト値の数値範囲への割り当てができます。

表示オプション

これらのオプションで、視覚的表現の内容を絞り込みます。

[値の表示]

値が値ボックス内でレンダリングされるかどうかを制御します。[自動] は、十分なスペースがある場合に値をレンダリングします。

[列幅] は、ボックスの幅を制御します。1=最大、0=最小幅。

[線の幅] は、状態領域の線幅を制御します。

[塗りつぶしの不透明度] は、状態領域の塗りつぶしの不透明度を制御します。

値のマッピング

ブール値または文字列値に色を割り当てるには、値の配置 を使用します。

しきい値を含む時系列データ

パネルは時系列データでも使用できます。この場合、しきい値を使用してボックスを色付けします。グラデーションのカラースキームを使用して値を色付けすることもできます。

凡例オプション

凡例オプションを有効にすると、値のマッピングまたはしきい値の括弧のいずれかを表示できます。凡例に値のマッピングを表示するには、[標準] オプションの [カラースキーム] オプションを [単一色] または [クラシックパレット] に設定することが重要です。凡例にしきい値の括弧を表示するには、カラースキームを [しきい値起点] に設定します。

[凡例モード] – これらの設定を使用して視覚化内の凡例の表示方法を改善します。

- [リスト] – 凡例を一覧として表示します。これは凡例のデフォルトの表示モードです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

[凡例の配置] 凡例を配置する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

テーブルパネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

テーブルパネルは、時系列、テーブル、注釈、未加工 JSON データのための複数のモードをサポートします。このパネルには、日付の書式設定、値の書式設定、色付けのためのオプションも用意されています。

データオプションとフィールドオプション

テーブルの視覚化を使用すると、次のオプションを適用できます。

- 変換.
- フィールドオプションとオーバーライド.
- しきい値.

表示オプション

- ヘッダーの表示 – データソースからインポートされた列名を表示または非表示にします。
- [昇順/降順のソート] — 列タイトルを選択してソート順序をデフォルトから降順に変更します。選択するたび、ソート順序はサイクルの次のオプションに変わります。一度に 1 つの列のみでソートすることができます。
- テーブルのフィールドオプション – 列幅、配置、セル表示モードなどのフィールドオプションに変わります。
- テーブル列のフィルタリング – 列データの表示方法が一時的に変わります。例えば、最大から最小までの値を順序付けたり、特定の値を非表示にしたりできます。

注釈のサポート

注釈は現在新しいテーブルパネルではサポートされていません。

テーブルのフィールドオプション

このセクションでは、使用可能なすべてのテーブルフィールドのオプションについて説明します。オプションは、Amazon Managed Grafana と同じ順序で一覧されます。このトピックに一覧されるオプションは、テーブルパネルの視覚化にのみ適用されます。

ほとんどのフィールドオプションは、編集するフィールドオプションボックスの外部を選択するか、Enter キーを押すまで、視覚化には影響しません。

これらのオプションの適用の詳細については、「すべてのフィールドを設定する」および「特定のフィールドを設定する」を参照してください。

[列の配置]

Amazon Managed Grafana がセルの内容を配置する方法を選択します。

- Auto (デフォルト)
- Left (左)
- Center (中央)
- Right (右)

列幅

デフォルトでは、Amazon Managed Grafana はセルの内容に基づいて列の幅を自動的に計算します。このフィールドオプションでは、設定を上書きして、すべての列の幅をピクセル単位で定義できます。

例えば、`100` をフィールドに入力すると、フィールドの外部を選択したときにすべての列の幅が 100 ピクセルに設定されます。

セル表示モード

デフォルトでは、Amazon Managed Grafana は自動的に表示設定を選択します。以下のオプションのいずれかを選択してすべてのフィールドを変更すると、設定を上書きできます。

> **Note**
>
> [フィールド] タブでこれらを設定すると、表示モードが時間フィールドを含むすべてのフィールドに適用されます。[オーバーライド] タブで設定すると、多くのオプションが最適に機能します。

文字色

しきい値が設定されている場合、フィールドのテキストは適切なしきい値の色で表示されます。

色付背景

しきい値が設定されている場合、フィールドの背景は適切なしきい値の色で表示されます。

グラデーションゲージ

しきい値レベルによりグラデーションが決定されます。

LCD ゲージ

ゲージは小さなセルに分割され、点灯または消灯で表示されます。

JSON ビュー

値はコードの形式で表示されます。値がオブジェクトの場合、値の上にカーソルを置くと JSON オブジェクトに移動できる JSON ビューが表示されます。

[Column filter] (列フィルター)

テーブル列のフィルタリング

テーブルオプションで [列フィルター] をオンにすると、テーブルオプションをフィルタリングできます。詳細については、「テーブルのフィールドオプション」を参照してください。

列フィルタリングを有効にする

1. Amazon Managed Grafana で、フィルタリングする列を含むテーブルを表示するダッシュボードを選択します。
2. フィルタリングするテーブルパネルで、パネルエディタを開く を実行します。
3. [フィールド] タブを選択します。
4. [テーブル] オプションで、[列フィルター] オプションをオンにします。

各列のタイトルの横にフィルターアイコンが表示されます。

列の値のフィルタリング

列の値をフィルタリングするには、列タイトルの横にあるフィルター (ファネル) アイコンを選択します。Grafana ワークスペースにその列のフィルターオプションが表示されます。

表示する値の横にあるチェックボックスを選択します。上部の検索フィールドにテキストを入力してそれらの値をディスプレイに表示すると、スクロールして探さずに選択できます。

列フィルターのクリア

フィルターが適用された列では、タイトルの横に青い漏斗が表示されます。

フィルターを削除するには、青いファネルアイコンを選択してから、[フィルターをクリア] を選択します。

テキストパネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

テキストパネルを使用すると、ダッシュボードの情報パネルと説明パネルを作成できます。

[モード]で、マークダウンまたは HTML を使用してテキストのスタイルを設定するかどうかを選択し、以下のボックスにコンテンツを入力します。Grafana ワークスペースには使用開始に役立つタイトルと段落が含まれています。別の エディタからコンテンツを貼り付けることもできます。

時系列パネル

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

時系列パネルでは、時系列を線、ドットの軌跡、または一連の棒としてレンダリングできます。このタイプのグラフは、ほぼすべての時系列データを表示するための十分な汎用性があります。

Note
グラフパネルの視覚化は時系列の視覚化に移行できます。移行するには、[パネル]タブで [時系列の視覚化] を選択します。Grafana は該当するすべての設定を転送します。

時系列の視覚化を使用すると、次のオプションを適用できます。

- 変換
- フィールドオプションとオーバーライド
- しきい値

また、フィールドオプションを使用してさまざまなタイプのグラフを作成、または軸を調整できます。

この設定で、視覚化する内容を絞り込みます。

ツールチップモード

グラフにカーソルを合わせると、Grafana はツールチップを表示できます。ツールチップの動作を選択します。

- [単独] – ツールチップには、カーソルを合わせている系列のみが表示されます。
- [すべて] – ツールチップには、グラフ内のすべての系列が表示されます。Grafana は、ツールチップの系列一覧で、カーソルを合わせている系列を太字で強調表示します。
- [非表示] – ツールチップを表示しません。

凡例モードと配置

凡例の表示方法を選択します。

- [リスト] – 凡例を一覧として表示します。これがデフォルトです。
- [テーブル] – 凡例をテーブルとして表示します。
- [非表示] – 凡例を非表示にします。

凡例を表示する場所を選択します。

- [Bottom] – グラフの下。
- [右側] – グラフの右側。

凡例計算

凡例に表示する計算を選択します。詳細については、「計算リスト」を参照してください。

線による時系列のグラフ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、時系列フィールドオプションを使用して時系列データを線として視覚化する方法と、オプションの動作について説明します。

パネルの作成

1. [時系列] の視覚化を選択して、パネルを作成します。詳細については、「パネルの追加」を参照してください。
2. [パネルエディタ] で、[フィールド] を選択します。
3. [スタイル] で、[線] を選択します。

線のスタイル設定

線のスタイリングにはさまざまなオプションがあります。

- [線形補間] — Grafana がシリーズの線を補間する方法を選択します。選択肢は、[直線]、[平滑線]、[前のステップ]、[後のステップ] です。
- 線幅 – 線の厚さを 0 ~ 10 ピクセルに設定します。
- [塗りつぶしの不透明度] – シリーズの塗りつぶしの不透明度を 0 ~ 100% に設定します。
- [グラデーションモード] – グラデーション塗りつぶしのモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。

  グラデーションの外観は [塗りつぶし不透明度] の設定に影響されます。

  グラデーションフィルの選択肢は、[なし]、[不透明度]、[色相] です。[不透明度] では、グラデーションの透明度が y 軸の値に基づいて計算されます。y 軸の値とともに塗りつぶしの不透明度が増加します。[色相] では、グラデーションの色が線の色の色相に基づいて生成されます。

- [線のスタイル] – 線のスタイルを設定します。色を変更するには、標準のカラースキームフィールドオプションを使用します。

  線のスタイルの外観は、[線幅] と [塗りつぶしの不透明度] の設定に影響されます。

  線のスタイルの選択肢は、[実線]、[破線]、および [点線] です。
- [Null 値] – データ内のギャップの表示方法を選択します。null 値は、連続線を形成するために接続することも、オプションで、超えるとデータ内のギャップが接続されなくなるしきい値を設定することもできます。[常に非表示] で常にデータポイントをギャップで接続しない、[常に表示] で常にデータポイントをギャップで接続する、またはデータ内のギャップを接続しなくなる [しきい値] を設定することができます。
- ポイントの表示 – ポイントがグラフに表示されるタイミングを選択します。選択肢は、[自動]、[常に]、[なし] です。

[下を塗りつぶす]

このオプションは、オーバーライドタブでのみ使用できます。

2 つのシリーズ間の領域を埋める方法

1. 下を塗りつぶすフィールドを選択します。
2. [オーバーライドプロパティの追加] で、[下を塗りつぶす] を選択します。
3. 塗りつぶしを停止するシリーズを選択します。

棒による時系列のグラフ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、時系列フィールドオプションを使用して時系列データを棒として視覚化する方法と、オプションの動作について説明します。

パネルの作成

1. [時系列] の視覚化を選択して、パネルを作成します。詳細については、「パネルの追加」を参照してください。
2. [パネルエディタ] で、[フィールド] を選択します。
3. [スタイル] で、[棒] を選択します。

棒のスタイル設定

棒のスタイリングにはさまざまなオプションがあります。

- 棒の配置 – データポイントに対する棒の位置を設定します。選択肢は、[前]、[中央]、[後] です。
- 線幅 – 棒の輪郭の厚さを 0 ~ 10 ピクセルに設定します。
- [塗りつぶしの不透明度] – 棒の塗りつぶしの不透明度を 0 ~ 100% に設定します。
- [グラデーションモード] – グラデーション塗りつぶしのモードを設定します。塗りつぶしグラデーションは線の色を基準にします。色を変更するには、標準のカラースキームフィールドオプションを使用します。

  グラデーションの外観は [塗りつぶし不透明度] の設定に影響されます。

  グラデーションフィルの選択肢は、[なし]、[不透明度]、[色相] です。[不透明度] では、グラデーションの透明度が y 軸の値に基づいて計算されます。y 軸の値とともに塗りつぶしの不透明度が増加します。[色相] では、グラデーションの色が線の色の色相に基づいて生成されます。
- ポイントの表示 – ポイントがグラフに表示されるタイミングを選択します。選択肢は、[自動]、[常に]、[なし] です。

ポイントによる時系列のグラフ

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、時系列フィールドオプションを使用して時系列データをポイントとして視覚化する方法と、オプションの動作について説明します。

パネルの作成

1. [時系列] の視覚化を選択して、パネルを作成します。詳細については、「パネルの追加」を参照してください。
2. [パネルエディタ] で、[フィールド] を選択します。
3. [スタイル] で、[ポイント] を選択します。

ポイントのスタイル設定

ポイントとしてグラフ化するときは、ポイントサイズを選択できます。

- ポイントサイズ – 直径 1 ~ 40 ピクセルからポイントサイズを選択します。

軸表示の変更

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、時系列フィールドオプションを使用して視覚化内の軸表示を制御する方法を説明し、軸オプションの動作を示します。

軸にはさまざまなオプションがあります。

- Y 軸の配置 – Y 軸の配置を設定します。選択肢は、[左]、[右]、[非表示] です。
- Y 軸のラベル – Y 軸のテキストラベルを設定します。Y 軸が複数ある場合は、[上書き] タブを使用して異なるラベルを割り当てることができます。
- 幅 – 軸の固定幅を設定します。デフォルトでは、Grafana ワークスペースは軸の幅を動的に計算します。軸の幅を設定することで、軸タイプが異なるデータでも同じ表示比率を共有できます。これ

により、軸が互いに視覚的に近接して移動または伸張されないため、複数のグラフのデータの価値を簡単に比較できます。

- ソフト最小値とソフト最大値 — Y 軸の制限をより適切に制御するために、[ソフト最小値] または [ソフト最大値] を設定します。デフォルトでは、Grafana ワークスペースはデータに基づいて y 軸の範囲を自動的に設定します。

  [ソフト最小値] または [ソフト最大値] の設定をすると、データがほぼフラットなときに小さな変動が大きな変動に見えるのを防ぎます。また、標準の最小値および最大値のフィールドオプションから派生したハード最小値または最大値により、定義されたポイントを超えたスパイクを切り取ることで、断続的なスパイクが有用な詳細を平坦化することを防げます。
- スケール – Y 軸の値に使用するスケールを設定します。選択肢は、[線形] と [対数] です。

積み上げ時系列のグラフ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、時系列パネルフィールドのオプションを使用してシリーズの積み上げを制御する方法と、積み上げオプションの動作について説明します。積み上げにより、Grafana は系列の上に別の系列を表示できます。視覚化で積み上げを使用すると、誤解を招くグラフを簡単に作成できるため、注意が必要です。積み上げが最適な手段ではない理由の詳細については、「積み上げに関する問題」を参照してください。

グループ内の系列の積み上げ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

積み上げグループオプションは、オーバーライドとしてのみ使用できます。

同じグループにシリーズをスタックする方法

1. オーバーライドセクションで、[シリーズの積み上げ] オプションのフィールドオーバーライドを作成します。
2. [普通] スタックモードを選択します。
3. シリーズを表示する積み上げグループに名前を付けます。積み上げグループ名オプションは、オーバーライドを作成する場合にのみ使用できます。

しきい値

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

しきい値により、定義した条件に応じて値のテキストまたは背景の、いずれかの色が設定されます。

しきい値は次の 2 つの方法のいずれかで定義できます。

- 絶対しきい値は、1 ~ 150 のスケールで 80 のように、数値に基づいて定義されます。
- パーセントしきい値は、最小値または最大値に対して、80% などのように定義されます。

次の視覚化に対してしきい値を適用できます。

- 棒ゲージパネル
- ゲージパネル
- グラフパネル

- 統計パネル
- テーブルパネル

デフォルトのしきい値

これをサポートする視覚化で、Amazon Managed Grafana は、80 = 赤、ベース = 緑、モード = 絶対の、デフォルトのしきい値を設定します。

[Base] (ベース) 値は、マイナス無限大を表します。一般的に良い色です。

しきい値の追加

パネルにはお好みの数のしきい値を追加できます。Grafana ワークスペースはしきい値を高い順に自動的にソートします。

> **Note**
>
> これらの手順は、統計、ゲージ、棒ゲージ、テーブルの視覚化にのみ適用されます。

1. しきい値を追加するパネルを選択します。
2. [フィールド] タブを選択します。
3. [しきい値の追加] を選択します。

   Amazon Managed Grafana は推奨される数値と色値を含むしきい値を追加します。
4. 推奨を受け入れるか新しいしきい値を編集します。

   - 色の編集 – 変更する色ドットを選択してから新しい色を選択します。
   - 数の編集 – 変更する数を選択し、新しい数を入力します。
   - しきい値モード – このパネルのすべてのしきい値に対して変更するモードを選択します。
5. [保存] を選択し、この変更をダッシュボードに保存します。

グラフパネルへのしきい値の追加

グラフパネルの視覚化では、しきい値を使用してグラフに任意の線またはセクションを追加すると、グラフが特定のしきい値を超えるタイミングを簡単に確認できます。

1. しきい値を追加するグラフパネルを選択します。

2. [パネル] タブで、[しきい値] を選択します。
3. [しきい値の追加] を選択します。
4. お好みの数のフィールドに入力します。T1 フィールドのみが必須です。
   - T1 – しきい値を表示するために両方の値が必要です。
     - lt または gt – しきい値の適用対象示すため、より小さい場合は lt、より大きい場合は gt を選択します。
     - [値] – しきい値を入力します。Grafana ワークスペースでは、その値の y 軸に沿ってしきい値線が描画されます。
   - Color – 色に対応する条件を選択、または独自の色を定義します。
     - [カスタム] – 塗りつぶしの色と線の色を定義します。
     - [重大] - 塗りつぶしの色と線の色は赤色です。
     - [警告} – 塗りつぶしの色と線の色は黄色です。
     - OK – 塗りつぶしの色と線の色は緑色です。
   - 塗りつぶし – しきい値の塗りつぶしを表示するかどうかを選択します。
   - 線 – しきい値の線を表示するかどうかを選択します。
   - [Y 軸] – 左または右を選択します。
5. [保存] を選択し、この変更をダッシュボードに保存します。

しきい値の削除

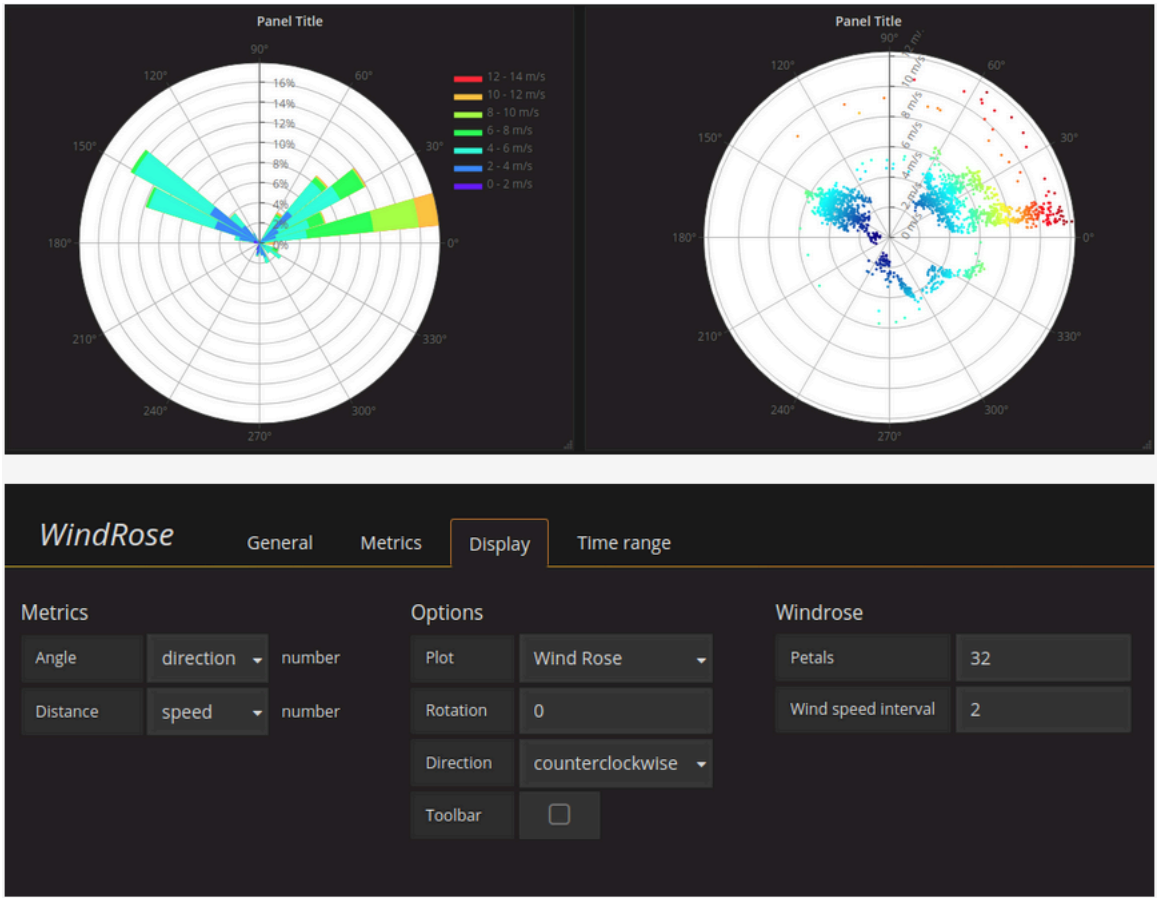
1. しきい値を削除するパネルを選択します。
2. [フィールド] タブを選択します。(グラフパネルの場合は、[パネル] タブを選択します)
3. 削除するしきい値の横にある、ごみ箱アイコンを選択します。
4. [保存] を選択し、この変更をダッシュボードに保存します。

WindRose

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。

Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

WindRose パネルは、生の時系列データを受信し、データを変換して WindRose チャートにマッピングします。



オプション

WindRose パネルは、以下のオプションをサポートしています。

- 軸の頻度
- 軸のスタイル (角度またはコンパス)
- 目盛 (線形、正方形、対数)

パネルの検査

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

パネルインスペクターは、パネルの理解およびトラブルシューティングに役立ちます。Grafana ワークスペースパネルの未加工データの検査、データの CSV (comma-separated values: カンマ区切り) ファイルへのエクスポート、クエリリクエストの表示、JSON 形式のパネルとデータのエクスポートが可能です。

パネルインスペクター UI

パネルインスペクターでは、ペインの上部に [Inspect: <NameOfPanelBeingInspected >] が表示されます。右上隅の矢印を選択してペインを展開または縮小します。

パネルインスペクターは次の 4 つのタブで構成されます。

- [データ タブ] – 変換が適用されたクエリによって返される未加工データを表示します。オーバーライドや値マッピングなどのフィールドオプションは、デフォルトで適用されません。
- [統計タブ] – クエリにかかる時間および返される量を示します。
- [JSON] タブ – パネル JSON、パネルデータ JSON、およびデータフレーム構造 JSON を、表示およびコピーできます。これは Amazon Managed Grafana のプロビジョニングまたは管理に役立ちます。
- クエリタブ – Amazon Managed Grafana がデータソースをクエリするときに送信されるサーバーへのリクエストを表示します。

Note

すべてのパネルタイプに 4 つのタブがすべてあるわけではありません。例えば、ダッシュボードリストパネルには検査する未加工データがないため、[統計]、[データ]、[クエリ] タブは表示されません。

パネルインスペクタータスク

パネルインスペクターでは、パネルの検査、未加工のクエリ結果の検査およびダウンロード、クエリのパフォーマンスの検査を実行し、パネル JSON モデルを表示し、データソースに対する未加工のリクエストおよびレスポンスを表示できます。

パネルインスペクターを開く

表示できるすべてのパネルを検査できます。

1. Grafana ワークスペースコンソールで、検査するパネルが含まれるダッシュボードを選択します。
2. 検査するパネルのタイトルを選択して、 [検査] を選択します。または、パネルタイトルを一時停止してから i を押します。

画面の右側にパネルインスペクターペインが表示されます。

未加工クエリ結果の検査

テーブル内の未加工クエリ結果を確認します。これらは、、クエリに変換が適用され、パネルにフィールドオプションまたはフィールドオプションオーバーライドを適用する前に、クエリによって返されるデータです。

1. パネルインスペクターを開いてから [データ] タブを選択します。または、パネルメニューで、[検査]、[データ] を選択します。
2. パネルに複数のクエリがある、または複数のノードにクエリを実行する場合は、追加のオプションがあります。

- [結果の選択] – 表示する結果セットデータを選択します。
- [データを変換する]
  - [時間で結合]: すべてのクエリの生データを同時に表示し、列ごとに結果セットを 1 つ表示します。列見出しを選択して、データ順序を変更します。

フィールドオプションとオプションオーバーライドを適用済みのテーブルに未加工のクエリ結果を表示します。

1. パネルインスペクターで [データ] タブを開きます。
2. テーブルの上方にある [データ表示オプション] を選択します。

3. [フィールド設定の適用] トグルボタンを選択します。

未加工のクエリ結果を CSV ファイルとしてダウンロード

Amazon Managed Grafana は、ブラウザのデフォルトのダウンロード場所に CSV ファイルを生成します。お好みのビューワーで開くことができます。

1. パネルインスペクターを開きます。
2. 上記のように未加工のクエリ結果を検査します。エクスポートする未加工データが表示されるまで、設定を調整します。
3. [CSV のダウンロード] を選択します。

Excel 用にフォーマットされた CSV ファイルをダウンロードするには、[データオプション] パネルを展開して [Excel 用ダウンロード] オプションをオンにしてから、[CSV のダウンロード] を選択します。

クエリパフォーマンスの検査

[統計] タブには、クエリにかかる時間、送信したクエリの数、返された行数を示す統計が表示されます。この情報は、特に数値が予想外に高い場合や低い場合のクエリのトラブルシューティングに役立ちます。

1. パネルインスペクターを開きます。
2. [ステータス] タブを選択します。

統計が読み取り専用形式で表示されます。

パネル JSON モデルの表示

パネル、パネルデータ、データフレーム JSON モデルを検索してエクスポートします。

1. パネルインスペクターを開いてから、[JSON] タブを選択します。または、パネルメニューで、[検査]、[パネル JSON] を選択します。
2. [ソースの選択] で、次のオプションのいずれかを選択します。
   - Panel JSON – パネルを表す JSON オブジェクトを表示します。
   - [Panel data] – パネルに渡されるデータを表す JSON オブジェクトを表示します。

- [データフレーム構造] – 変換、フィールド設定、オーバーライドが適用された、未加工の結果セットを表示します。

3. JSON の一部を展開または折りたたんで調べるか、[クリップボードにコピー] を選択して別のアプリケーションの JSON に貼り付けることができます。

データソースへの未加工リクエストとレスポンスの表示

1. パネルインスペクターを開いてから [クエリ] タブを選択します。または、パネルメニューで、[検査]、[クエリ] を選択します。
2. [更新] を選択します。

Amazon Managed Grafana は情報を収集するためのクエリをサーバーに送信してから、結果を表示します。クエリの特定の部分をドリルダウン、そのすべてを展開または折りたたみ、またはデータをクリップボードにコピーして他のアプリケーションで使用できます。

計算リスト

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このトピックでは Amazon Managed Grafana で使用される計算を一覧表示して定義します。

特に、これらの計算は、[変換] タブと棒ゲージ、ゲージ、および統計の視覚化に使用します。

計算	説明
All nulls	すべての値が null のとき真
All zeros	すべての値が 0 のとき真
Change count	フィールドの値が変更される回数

計算	説明
Count	フィールド内の値の数
Delta	値の累積変化
Difference	フィールドの最初の値と最後の値の差
Distinct count	フィールド内の一意の値の数
最初 (null ではない)	最初、null でないフィールド内の値
最大	フィールドの最大値
Mean	フィールド内のすべての値の平均値
最小	フィールドの最小値
最小 (above zero)	フィールドの最小値、正の値
Range	フィールドの最大値と最小値との差
Step	フィールドの値間の最小間隔
Total	フィールド内のすべての値の合計

ダッシュボード

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードは、1 つ以上の行に編成および配置された 1 つ以上のパネルのセットです。Amazon Managed Grafana には、さまざまなパネルが用意されています。Amazon Managed Grafana では、

クエリを簡単に作成し、表示プロパティをカスタマイズして、お好みのダッシュボードを作成することができます。各パネルは、設定された任意のデータソースからのデータを操作できます。

ダッシュボードの管理

ダッシュボードの期間の制御は、ダッシュボードの右上にある 時間範囲コントロール を使用します。

ダッシュボードは、テンプレートや変数を使用することで、より動的でインタラクティブにすることができます。詳細については、「テンプレートおよび変数」を参照してください。

ダッシュボードは、注釈 を使用してパネル間でイベントデータを表示できます。これにより、パネル内の時系列データを他のイベントと関連付けることができます。

ダッシュボードはさまざまな方法で簡単に共有できます。詳細については、「ダッシュボードの共有」を参照してください。

ダッシュボードにはタグを付けることができます。ダッシュボードピッカーを使用することで、特定の組織内のすべてのダッシュボードを簡単に検索してアクセスできるようになります。

[Rows] (行)

行は、ダッシュボード内を論理的に区切るための要素です。行を使用してパネルをグループ化することができます。

行の幅は常に 12 ユニットです。このユニットは、ブラウザの水平解像度に従って自動的にスケーリングされます。特定の幅を設定することで、行内のパネルの相対的な幅を制御できます。

Amazon Managed Grafana は、画面サイズに応じて最適な表示を実現するために、ユニットの抽象化を使用しています。

> **Note**
>
> 最大データポイント数 機能を使用すると、Amazon Managed Grafana は解像度や時間範囲に関係なく、必要な数のデータポイントを表示できます。

行を折りたたむには、行タイトルを選択します。行が折りたたまれた状態でダッシュボードを保存するとその状態で保存され、行を展開するまでグラフは読み込まれません。

繰り返し行機能を使用すると、選択したテンプレート変数に基づいて、パネルの行全体を動的に作成または削除することができます。

注釈

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

注釈は、リッチイベントでグラフ上のポイントをマークする方法を提供します。注釈にカーソルを合わせると、イベントの説明とイベントタグを見ることができます。テキストフィールドには、より詳細な他のシステムへのリンクを含めることができます。

ネイティブ注釈

Amazon Managed Grafana には、ネイティブ注釈ストアと、グラフパネルから直接注釈イベントを追加する機能が付属しています。

注釈の追加

注釈を追加するには、[Ctrl] または [Cmd] を押して、注釈を追加する場所を選択します。注釈にタグを追加することで、他のダッシュボードから注釈を検索できるようになります。

リージョン注釈の追加

リージョンの注釈を作成するには、リージョンを選択しながら [Ctrl] または [Cmd] を押します。

ビルトインクエリクエリ

追加した注釈は表示されたままになります。これは、ビルトインの注釈クエリがすべてのダッシュボードに存在するためです。この注釈クエリは、現在のダッシュボードから発生するすべての注釈イベントを取得し、それらが作成されたパネルに表示されます。これには、アラート状態履歴の注釈が含まれます。注釈の取得と表示を停止するには、[ダッシュボード設定] (歯車) アイコン、[注釈] の順に選択し、`Annotations & Alerts (Built-in)` という名前のクエリを変更します。

[名前を付けて保存] 機能を使用してダッシュボードをコピーすると、新しいダッシュボードに新しいダッシュボード ID が割り当てられるため、コピー元のダッシュボードで作成された注釈はコピーに表示されません。コピー元のダッシュボードの注釈にフィルタリングするタグがある場合は、新しい[注釈クエリ]を追加し、タグでフィルタリングすることで、コピーに注釈を表示できます。

タグによるクエリ

`-- Grafana --` データソースを使用して [~ でフィルタリング] を タグ に設定することで、ネイティブ注釈ストアから注釈を取得する新しい注釈クエリを作成できます。少なくとも 1 つのタグを指定します。例えば、`outages` という名前の注釈クエリ名を作成し、`outage` という名前のタグを指定します。このクエリは、(ダッシュボードまたは API 経由) で作成した `outage` タグを含むすべての注釈を表示します。

デフォルトでは、注釈クエリに複数のタグを追加すると、Amazon Managed Grafana は指定したすべてのタグを持つ注釈のみを表示します。指定したタグのうち、いずれか1つを含む注釈を表示するには、[任意の一致] をオンにします。

Amazon Managed Grafana では、タグクエリでテンプレート変数を使用できます。例えば、さまざまなサービスの統計を表示するダッシュボードと、表示するサービスを制御するテンプレート変数がある場合は、注釈クエリで同じテンプレート変数を使用して、それらのサービスの注釈のみを表示できます。

その他のデータソースのクエリ

注釈イベントは、注釈クエリを使用して取得されます。ダッシュボードに新しい注釈クエリを追加するには、[ダッシュボード設定] (歯車) アイコン、[Annotations]、[新規] の順に選択します。

注釈クエリの名前を指定します。この名前は、このクエリの注釈イベントの表示または非表示を切り替えるチェックボックスに表示されます。例えば、2 つの注釈クエリ(`Deploys` と `Outages`) があるとします。チェックボックスのオンとオフを切り替えることで、表示する注釈を指定できます。

注釈クエリの詳細

注釈クエリオプションは、データソースごとに異なります。

- Graphite クエリエディタを使用した注釈
- OpenSearch データソースを使用した注釈
- Prometheus を使用した注釈
- [MySQL を使用した注釈]

- [PostgreSQL を使用した注釈]

ダッシュボードのフォルダ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

フォルダを使用して、ダッシュボードを整理してグループ化することができます。これは、ダッシュボードが数多くある場合や、複数のチームが同じ Grafana ワークスペースを使用している場合に重宝します。

フォルダの作成

フォルダを作成するには、以下のいずれかの操作を実行します。

- サイドメニューの [+] アイコンから [フォルダの作成] リンクを選択します。
- [ダッシュボードの管理] ページで、[フォルダの作成] ボタンを選択します。

[フォルダの作成] ページで、フォルダの一意の名前を入力して [作成] を選択します。

ダッシュボードの保存先には、既存のフォルダを選択するか、新しいフォルダを作成して選択することができます。

ダッシュボードの管理

[ダッシュボードの管理] ページでは、さまざまなタスクを実行できます。

- フォルダを作成します。
- ダッシュボードを作成します。
- ダッシュボードをフォルダに移動します。
- 複数のダッシュボードを削除します。

- フォルダページに移動します (フォルダまたはそのダッシュボードのアクセス権限を設定できます)。

ダッシュボードフォルダページ

ダッシュボードフォルダページを開くには、[検索結果] または [ダッシュボードの管理] ページのダッシュボード一覧のフォルダにカーソルを合わせると表示される歯車アイコンを選択します。

ダッシュボードフォルダページは、ダッシュボードの管理ページと似ています。ダッシュボードフォルダページで、以下の操作を行うことができます。

- フォルダ内のダッシュボードを移動または削除する。
- フォルダの名前の変更 ([設定] タブで)。
- フォルダのアクセス権限の設定 (フォルダ内のダッシュボードによって継承されます)。

アクセス許可

アクセス許可はフォルダに割り当てることができ、そのフォルダに含まれるダッシュボードがその権限を継承します。アクセスコントロールリスト (ACL) は、組織ロール 、チーム、個人ユーザーに権限を割り当てるために使用します。詳細については、「ダッシュボードおよびフォルダのアクセス権限」を参照してください。

プレイリスト

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

プレイリストは、ダッシュボードを順番に表示するためのリストです。プレイリストを使用して状況認識を高めたり、チームや訪問者にメトリクスを提示することができます。

Amazon Managed Grafana は、解像度に合わせてダッシュボードを自動的に拡大縮小し、大画面にも対応します。

プレイリスト機能には、サイドメニューのダッシュボードサブメニューからアクセスできます。

プレイリストの作成

プレイリストは、ダッシュボードを順番に、一定の時間間隔で表示します。

1. プレイリスト機能へのアクセスは、サイドメニューにカーソルを合わせます。
2. [プレイリスト] を選択します。
3. [新しいプレイリスト] を選択します。
4. [名前] ボックスに、プレイリスト名を入力します。
5. [間隔] テキストボックスに時間間隔を入力します。

   時間間隔は、Amazon Managed Grafana が順番に表示するプレイリストの各ダッシュボードを表示したままにする時間です。
6. プレイリストに追加したいダッシュボードの横にある、[プレイリストに追加] を選択します。
7. [作成] を選択します。

プレイリストの編集

プレイリストは、作成中または保存後に編集できます。

1. プレイリスト機能へのアクセスは、サイドメニューにカーソルを合わせます。
2. [プレイリスト] を選択します。
3. 編集するプレイリストを選択します。

プレイリスト名の編集

1. [名前] テキストボックスを選択します。
2. 名前を編集します。
3. [保存] を選択して変更を保存します。

プレイリストの間隔の編集

1. [間隔] テキストボックスを選択します。
2. 間隔を編集します。

3. [保存] を選択して変更を保存します。

プレイリストにダッシュボードを追加する

1. プレイリストに追加したい各ダッシュボードの横にある、[プレイリストに追加] を選択します。
2. [保存] を選択して変更を保存します。

追加するダッシュボードの検索

1. [ダッシュボードの追加] で、[名前でダッシュボードを検索] テキストボックスを選択します。
2. 名前または正規表現を入力します。
3. 必要に応じて、スター付きの状態やタグで結果を絞り込むことができます。デフォルトでは、スター付きのダッシュボードがプレイリストに追加できるオプションとして表示されます。
4. [保存] を選択して変更を保存します。

ダッシュボードの順序の並べ替え

1. 移動させるダッシュボードの横にある上矢印または下矢印を選択します。
2. [保存] を選択して変更を保存します。

ダッシュボードの削除

1. プレイリストからダッシュボードを削除するには、x アイコンを選択します。
2. [保存] を選択して変更を保存します。

プレイリストの削除

1. [プレイリスト] を選択します。
2. 削除したいプレイリストの横にある x アイコンを選択します。

プレイリストの保存

プレイリストを保存すると、プレイリストページに追加され、そこで再生を開始できます。プレイリストを保存する前に、表示させたいすべてのダッシュボードが追加されていることを確認してください。

1. プレイリスト機能へのアクセスは、サイドメニューにカーソルを合わせます。
2. [プレイリスト] を選択します。
3. プレイリストを選択します。
4. プレイリストを編集します。

   プレイリストに名前、間隔が設定されており、少なくとも 1 つのダッシュボードが追加されていることを確認します。
5. [保存] を選択します。

プレイリストの開始

プレイリストの開始は 5 種類のビューモードから選択できます。モードによって、ダッシュボードへのメニューとナビゲーションバーの表示方法が異なります。

デフォルトでは、各ダッシュボードは間隔フィールド (プレイリストの作成時または編集時に設定可能) に入力された時間表示されます。プレイリスト開始後は、画面上部のナビバーで制御します。

1. [ダッシュボード]]メニューで、[プレイリスト]] を選択します。
2. 開始するプレイリストの横にある [プレイリストの開始] を選択します。
3. ドロップダウンリストから、次のいずれかの表示モードを選択します。

   - [通常モード]
     - サイドメニューは表示されたままです。
     - ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。
   - TV モード
     - サイドメニューは非表示 (削除) されます。
     - ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。
     - 1 分間操作しないと、TV モードが自動的にオンになります。
     - TV モードを手動でオンにするには、ショートカットとして d v を順番に押すか、ダッシュボード URL に パラメータ `?inactive` を追加します。
     - マウスを動かしたり何かキーボードのキーを押すことで、TV モードを無効にできます。
   - TV モード (自動フィットパネル付き)
     - サイドメニューは非表示 (削除) されます。
     - ナビゲーションバー、行、パネルのコントロールが画面の上部に表示されます。

    - ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。
- キオスクモード
    - サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。
    - プレイリストの起動後、 ショートカットとして d v を順番に押すと、手動で Kiosk モードをオンにできます。
    - このショートカットを使用して、手動で Kiosk モードをオフにできます。
- キオスクモード (自動フィットパネル付き)
    - サイドメニュー、ナビゲーションバー、行、およびパネルコントロールが、完全に非表示 (表示から削除) になります。
    - ダッシュボードパネルは、画面上のスペースを最適化するために自動的に調整されます。

プレイリストの制御

プレイリストの開始後は、画面上部のナビゲーションバーを使用して、通常モードまたはテレビモードでプレイリストを制御できます。

ボタン	結果
次へ (二重右矢印)	次のダッシュボードに進みます。
戻る (左矢印)	前のダッシュボードに戻ります。
停止 (四角)	プレイリストを終了し、現在のダッシュボードに戻ります。
サイクルビューモード (モニターアイコン)	ダッシュボードの表示を現在とは違うビューモードに変更します。
時間範囲	指定した時間範囲内のデータを表示します。ドロップダウン矢印を使用して、過去 5 分から 5 年前、または表示するカスタム時間範囲を設定します。
更新 (円矢印)	ダッシュボードを再読み込みして、最新のデータを表示します。ドロップダウン矢印を使用し

ボタン	結果
	て、5 秒ごとに自動的にリロードするように設定することもできます。

キーボードの Esc でもプレイリストを停止できます。

ビューモードでプレイリストを共有する

お好みのビューモードで URL をコピーし、宛先に URL を貼り付けることで、プレイリストを共有できます。

1. [ダッシュボード]]メニューから、[プレイリスト] を選択します。
2. 共有するプレイリストの横にある [プレイリストの開始] を選択し、お好みのビューモードを選択します。
3. [リンクアドレスをコピー] を選択すると、リンクがクリップボードにコピーされます。

   例えば、Grafana Play サイトのキオスクモードのプレイリスト URL は `https://play.grafana.org/d/000000010/annotations?orgId=1&kiosk` のようになります
4. 送信先に URL を貼り付けます。

ダッシュボードの検索

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードは、ダッシュボード名で検索することができます。また、タグ (複数可) やスター付きの状態でフィルタリングすることができます。ダッシュボードの検索は、ダッシュボード上部のナビゲーションバーにあるダッシュボードピッカーから行うことができます。ダッシュボードの検索は、ショートカット「F」でも開きます。

キーボードのみを使用する場合、キーボードの矢印キーで目的のダッシュボードまで検索結果を移動して「Enter」キーを押して開くことができます。

ダッシュボード名でフィルタリング

検索バーにダッシュボード名の一部を入力します。入力すると、文字列の部分一致の検索結果がリアルタイムで返されます。

ダッシュボードの検索機能:

- リアルタイム
- 大文字と小文字は区別されません。
- 保存済みダッシュボードとファイルベースのダッシュボード全体

タグでフィルタリング

ダッシュボードの数が増えるほど、タグをつけることで効果的に整理することができます。タグはダッシュボードの [設定] で追加および管理できます。

ダッシュボードの一覧をタグでフィルタリングするには、右側の列に表示されるタグを選択します。選択するタグを増やすことで、一覧をさらにフィルタリングできます。

[タグでフィルタリング]ドロップダウンメニューを選択すると、使用可能なすべてのタグの一覧を見ることができます。タグを選択すると、ダッシュボードの検索結果がリアルタイムにフィルタリングされます。

キーボードのみを使用する場合、「Tab」を押してタグリンクにカーソルを移動し、下矢印キーを押してタグを検索し、「Enter」を押してタグを選択します。

> **Note**
>
> 複数のタグを選択すると、Amazon Managed Grafana は選択したすべてのタグを含むダッシュボードを表示します。

ダッシュボードの共有

> このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。

Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードを共有するには、上部のナビゲーションバーで [ダッシュボードの共有] (共有アイコン) を選択します。[共有] ダイアログボックスが開き、現在選択されている時間範囲とテンプレート変数を含む現在のダッシュボードへのリンクを取得できます。ダッシュボードに変更を加えた場合は、リンクをコピーする前に、必ずこれらの変更を保存してください。

ダッシュボードスナップショット

ダッシュボードスナップショットは、インタラクティブダッシュボードを即時にパブリックに共有する方法です。スナップショットを作成すると、Amazon Managed Grafana はクエリ (メトリクス、テンプレート、注釈) やパネルリンクなどの機密データを消去し、ダッシュボードに埋め込まれた可視メトリクスデータとシリーズ名のみが残されます。ダッシュボードスナップショットには、リンクがあり URL にアクセスできるすべてのユーザーがアクセスできます。

スナップショットの発行

スナップショットはローカルインスタンスに発行できます。

パネルの共有

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

パネルタイトルを選択してパネルメニューを開き、パネルメニューで [共有] を選択してパネル共有ダイアログボックスを開きます。コピーしたリンクには現在の時間範囲と選択したテンプレート変数が設定されるため、このパネルに正確に移動できます。

時間範囲コントロール

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Amazon Managed Grafana では、視覚化されるデータの時間範囲を制御できるいくつかの方法がダッシュボードレベルとパネルレベルの両方に用意されています。

このトピックでは、サポートされている時間単位と相対範囲、一般的な時間コントロール、ダッシュボード全体の時間設定、パネル固有の時間設定について説明します。

Note

時間コントロールの使用には、データに時間列が含まれている必要があります。時間列を含める方法の詳細については、特定の「データソース」のドキュメントを参照してください。

時間単位と相対範囲

以下の時間単位がサポートされています。

- `s` (秒)
- `m` (分)
- `h` (時間)
- `d` (日数)
- `w` (週数)
- `M` (月数)
- `y` (年数)

マイナス演算子を使用すると、現在の時間を基準に過去を指定することができます。単位の全期間 (日、週、月など) を表示するには、`/<time unit>` を追加します。

プラス演算子を使用すると、現在の時間を基準に未来を指定することができます。この機能を使用して、将来の予測データを確認できます。

次に例を示します。

相対範囲の例	From	To
過去 5 分間	now-5m	now
今日のこれまでのところ	now/d	now
今週	now/w	now/w
今週始めから現在まで	now/w	now
前月	now-1M/M	now-1M/M

一般的な時間範囲コントロール

ダッシュボードとパネルの時間コントロールには、以下のオプションを含む共通のユーザーインターフェイスがあります。

現在の時間範囲

現在の時間範囲 (別名タイムピッカー)は、そのダッシュボードまたはパネルに現在表示されている時間範囲を示します。

フィールドにカーソルを合わせると、その範囲内の正確なタイムスタンプとソース (ローカルブラウザなど) が表示されます。

時間範囲を変更するには、現在の時間範囲を選択します。現在の時間は、過去 15 分などの相対時間範囲や、`2020-05-14 00:00:00 to 2020-05-15 23:59:59` のような絶対時間範囲で変更できます。

相対時間範囲

[相対時間範囲] リストから相対時間範囲を選択します。相対時間範囲の例をいくつか示します。

- 過去 30 分
- 過去 12 時間

- 過去 7 日間
- 過去 2 年間
- 昨日
- 一昨日
- 先週のこの日
- 現在まで
- 今週始めから現在まで
- 今月始めから現在まで

絶対時間範囲

絶対時間範囲は、次の 2 つの方法のいずれかで設定します。

- [開始] フィールドと [終了] フィールドに値を入力します。正確な時間値または now-24h などの相対値を入力し、[時間範囲の適用] を選択します。
- [開始] または [終了] フィールドを選択します。Amazon Managed Grafana にカレンダーが表示されます。現在の時間範囲として使用したい日付を選択し、[時間範囲の適用] を選択します。

Amazon Managed Grafana には、最近使用した絶対範囲も表示されます。

ズームアウト (Cmd+Z または Ctrl+Z)

ダッシュボードまたはパネルの視覚的表現でより大きな時間範囲を表示するには、[時間範囲のズームアウト] アイコンを選択します。

ズームイン (グラフに対してのみ)

グラフ上でドラッグして表示する時間範囲を選択します。

ダッシュボードの更新

[ダッシュボードの更新] アイコンを選択すると、ダッシュボード上のすべてのクエリが即座に実行され、視覚的表現が更新されます。Amazon Managed Grafana は、新しい更新が開始されると、保留中のリクエストをすべてキャンセルします。

デフォルトでは、Amazon Managed Grafana はダッシュボードを自動的に更新しません。クエリは、パネル設定に従って独自のスケジュールで実行されます。[ダッシュボードの更新] アイコンの横

にある下矢印を選択し、更新間隔を選択することで、ダッシュボードを定期的に更新することができます。

ダッシュボードの時間設定

時間設定はダッシュボードごとに保存されます。

画面上部のダッシュボード設定 (歯車) アイコンを選択すると、ダッシュボードの時間設定が開きます。設定は、[全般] タブの「時間オプション」セクションで行います。

- [タイムゾーン] – モニタリングするサービスまたはシステムのローカルタイムゾーン。これは、複数のタイムゾーンで動作するシステムまたはサービスを監視する場合に役立ちます。
  - [デフォルト] – ユーザープロファイル、チーム、または組織にデフォルトで設定されているタイムゾーン。ユーザープロファイル、ユーザーが所属するチーム、または組織にタイムゾーンが指定されていない場合、Amazon Managed Grafana はローカルブラウザの時間を使用します。
  - [ブラウザの時間] – 使用中のブラウザに設定されたタイムゾーン。これは通常、コンピュータに設定されているタイムゾーンです。
  - [協定世界時] – UTC を含む標準の ISO 8601 タイムゾーン。詳細については、「タイムゾーン一覧」を参照してください。
- [自動更新] – 相対時間と自動更新設定 (カスタマイズ可能オプション)。入力はカンマで区切りで、いずれも有効な時間単位である必要があります。
- 現在時刻の遅延- - now 値をオーバーライドする時間遅延値。通常、この機能は、データ集約の既知の遅延に対応するために null 値を回避するために使用されます。
- [タイムピッカーを非表示] – タイムピッカーを非表示にします。

パネル時間のオーバーライドと時間のシフト

クエリオプション では、個々のパネルに対して相対時間範囲を上書きして、ダッシュボードの上部ナビゲーションバーにあるタイムピッカーで選択されているものと異なる時間範囲に設定することができます。これにより、異なる期間または日数のメトリクスを同時に表示することができます。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、ダッシュボードの時間範囲を制御できます。

- `from` – エポックミリ秒で指定された時間範囲の下限、または相対時間を定義します。詳細については、「相対時間範囲」を参照してください。

- to – エポックミリ秒で指定された時間範囲の上限、または相対時間を定義します。詳細については、「相対時間範囲」を参照してください。
- time および time.window — time-time.window/2 から time+time.window/2 までの時間範囲を定義します。両方のパラメータは、ミリ秒単位で指定する必要があります。例えば、?time=1500000000000&time.window=10000 は 1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードのエクスポートとインポート

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Amazon Managed Grafana のダッシュボードは、UI または [HTTP API] から簡単にエクスポートおよびインポートすることができます。

ダッシュボードのエクスポート

ダッシュボードは、後でダッシュボードをインポートするために必要なレイアウト、変数、スタイル、データソース、クエリなどすべてを含めた Amazon Managed Grafana JSON 形式でエクスポートされます。

エクスポート機能は、ダッシュボードメニューの [共有] ボタンを選択して開いた共有ウィンドウから利用できます。

ダッシュボードをポータブルにする

他のユーザーが使用できるようにダッシュボードをエクスポートする場合、メトリクスプレフィックス (定数変数を使用) やサーバー名などのテンプレート変数を追加できます。

Constant タイプのテンプレート変数は、ダッシュボードで自動的に非表示になります。ダッシュボードがインポートされる際にも、必要な入力として追加されます。テンプレート作成とテンプレート変数の詳細については、「テンプレートおよび変数」を参照してください。

ダッシュボードのインポート

ダッシュボードをインポートするには、サイドメニューの [+] アイコン、[インポート] の順に選択します。

ダッシュボードの JSON ファイルをアップロードしたり、ダッシュボードの URL を貼り付けたり、ダッシュボードの JSON テキストをテキスト領域に直接貼り付けたりすることができます。

インポートプロセスのステップ 2 では、ダッシュボードの名前を変更したり、ダッシュボードで使用するデータソースを指定したり、メトリクスプレフィックス (ダッシュボードが使用している場合) を指定することができます。

Grafana.com でダッシュボードを探す

Grafana.com/dashboards で一般的なサーバーアプリケーションのダッシュボードを検索します。

ダッシュボードのバージョン履歴

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードのバージョンを保存するたびに、以前のバージョンのダッシュボードが失われないように、そのバージョンのコピーが保存されます。これらのバージョンの一覧は、[ダッシュボード設定] を選択し、左側のメニューで [バージョン] を選択することで確認できます。

ダッシュボードのバージョン履歴機能では、以前に保存したダッシュボードバージョンを比較して復元できます。

2 つのダッシュボードバージョンの比較

2 つのダッシュボードバージョンを比較するには、比較するリストから 2 つのバージョンを選択します。2 つのバージョンを選択後、[バージョンを比較] を選択すると差分ビューが開きます。デフォルトでは、次の画像のように、変更の概要がテキストで表示されます。

ダッシュボードを表す未加工の JSON の差分を表示するには、[JSON の差分を表示] を選択します。

比較対象の以前のバージョンに復元するには、[バージョン <x> に復元] を選択します。

以前に保存したダッシュボードバージョンへの復元

以前に保存したダッシュボードバージョンに復元する必要がある場合、ダッシュボードバージョンリストの行の右側にある [復元] ボタンを選択するか、差分ビューに表示される [バージョン<x>に復元] を選択します。復元を選択すると、復元を確認するポップアップボックスが表示されます。

以前のバージョンに復元すると、以前のバージョンとまったく同じデータを含みますが、バージョン番号が異なる新しいバージョンが作成されます。これは [Notes] (注記) 列に表示されます。これにより、以前のダッシュボードバージョンが変更の影響を受けなくなります。

キーボードショートカット

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Amazon Managed Grafana では多数のキーボードショートカットを利用できます。ご使用の Amazon Managed Grafana のバージョンで使用可能なすべてのキーボードショートカットを表示するには、キーボードで Shift + ? を押します。

Amazon Managed Grafana には次の一般的なショートカットが含まれています。

- Ctrl+S では現在のダッシュボードを保存します。
- Ctrl+F ではダッシュボードファインダー/検索が開きます。
- Ctrl+H ではすべてのコントロールが非表示になります (コントロールの非表示は TV のディスプレイでうまく機能します)。
- Escape では全画面モードまたは編集モードのときにはグラフを終了します。

ダッシュボード JSON モデル

このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Amazon Managed Grafana のダッシュボードは、ダッシュボードのメタデータを保存する JSON オブジェクトによって表されます。ダッシュボードメタデータには、ダッシュボードプロパティ、パネルからのメタデータ、テンプレート変数、パネルクエリが含まれます。

ダッシュボードの JSON を表示する方法

1. ダッシュボードを開きます。
2. 上部のナビゲーションバーで、ダッシュボードの管理を選択します。
3. ドロップダウンメニューから [JSON の表示] を選択します。

JSON フィールド

ユーザーが新しいダッシュボードを作成すると、新しいダッシュボード JSON オブジェクトが次のフィールドで初期化されます。

Note

次の JSON では、id は null として表示されます。これは、ダッシュボードが保存されるまで割り当てられるデフォルト値です。ダッシュボードを保存すると、整数値が id フィールドに割り当てられます。

```
{
  "id": null,
  "uid": "cLV5GDCkz",
```

```
  "title": "New dashboard",
  "tags": [],
  "style": "dark",
  "timezone": "browser",
  "editable": true,
  "hideControls": false,
  "graphTooltip": 1,
  "panels": [],
  "time": {
    "from": "now-6h",
    "to": "now"
  },
  "timepicker": {
    "time_options": [],
    "refresh_intervals": []
  },
  "templating": {
    "list": []
  },
  "annotations": {
    "list": []
  },
  "refresh": "5s",
  "schemaVersion": 17,
  "version": 0,
  "links": []
}
```

次の表に、ダッシュボード JSON の各フィールドの使用方法の詳細を示します。

名前	使用方法
id	ダッシュボードの一意の数値識別子 (データベースによって生成)
uid	誰でも生成できる一意のダッシュボード識別子。uid は 8～40 文字の文字列です。
title	ダッシュボードの現在のタイトル。
[タグ]	ダッシュボードに関連付けられているタグ。JSON では、タグは文字列の配列です。
style	ダッシュボードのテーマ (dark や light など)。

名前	使用方法
timezone	ダッシュボードのタイムゾーン (utc または browser)。
editable	ダッシュボード編集の可否。
graphTooltip	ツールチップのスタイル。 • 0: 共有クロスヘアまたはツールチップなし (デフォルト) • 1: 共有クロスヘア • 2: 共有クロスヘアと共有ツールチップ
time	ダッシュボードの時間範囲 (過去 6 時間、過去 7 日など)。
timepicker	タイムピッカーメタデータ。詳細については、「タイムピッカー」を参照してください。
templating	テンプレートメタデータ。詳細については、「テンプレートおよび変数」を参照してください。
annotations	注釈メタデータ。詳細については、「注釈」を参照してください。
refresh	自動更新間隔。
schemaVersion	JSON スキーマのバージョン (整数)。Amazon Managed Grafana 更新でスキーマが変更されるたびに増加します。
version	ダッシュボードのバージョン (整数)。ダッシュボードが更新されるたびに増加します。
panels	パネル配列。詳細については、「パネル」を参照してください。

パネル

パネルはダッシュボードの構成要素であり、データソースクエリ、グラフのタイプ、エイリアス、その他のデータなどで構成されます。パネル JSON は、JSON オブジェクトの配列で構成され、それぞれが異なるパネルを表します。ほとんどのフィールドはすべてのパネルで共通ですが、一部の

フィールドはパネルタイプによって異なります。以下は、テキストパネルのパネル JSON の例です。

```
"panels": [
  {
    "type": "text",
    "title": "Panel Title",
    "gridPos": {
      "x": 0,
      "y": 0,
      "w": 12,
      "h": 9
    },
    "id": 4,
    "mode": "markdown",
    "content": "# title"
  }
```

パネルのサイズと位置

gridPos プロパティは、グリッド座標のパネルサイズと位置を記述します。

- w – 1 ~ 24。ダッシュボードの幅は 24 列に分割されます。
- h – グリッドの高さの単位。各グリッド高さの単位では 30 ピクセルを表します。
- x – x 位置。x 位置は、w と同じ列単位を使用します。
- y – Y 位置。y 位置は、h と同じグリッド高単位を使用します。

グリッドには負の重力があり、パネルの上に空のスペースがある場合にパネルを上に移動します。

タイムピッカー

timepicker のオプションの例を以下に示します。

```
"timepicker": {
    "collapse": false,
    "enable": true,
    "notice": false,
    "now": true,
    "refresh_intervals": [
```

```
      "5s",
      "10s",
      "30s",
      "1m",
      "5m",
      "15m",
      "30m",
      "1h",
      "2h",
      "1d"
    ],
    "status": "Stable",
    "type": "timepicker"
  }
```

timepicker の使用方法の詳細を以下の表に示します。

名前	使用方法
collapse	timepicker が折りたたまれているかどうか
enable	timepicker がアクティブ化されているかどうか
notice	TODO
now	TODO
refresh_intervals	TODO
status	TODO
type	TODO

テンプレート作成

templating フィールドには、保存された値やその他のメタデータを含むテンプレート変数の配列が格納されています。templating のイベントの例を、以下に示します。

```
 "templating": {
```

```
    "enable": true,
    "list": [
      {
        "allFormat": "wildcard",
        "current": {
          "tags": [],
          "text": "prod",
          "value": "prod"
        },
        "datasource": null,
        "includeAll": true,
        "name": "env",
        "options": [
          {
            "selected": false,
            "text": "All",
            "value": "*"
          },
          {
            "selected": false,
            "text": "stage",
            "value": "stage"
          },
          {
            "selected": false,
            "text": "test",
            "value": "test"
          }
        ],
        "query": "tag_values(cpu.utilization.average,env)",
        "refresh": false,
        "type": "query"
      },
      {
        "allFormat": "wildcard",
        "current": {
          "text": "apache",
          "value": "apache"
        },
        "datasource": null,
        "includeAll": false,
        "multi": false,
        "multiFormat": "glob",
        "name": "app",
```

```
        "options": [
          {
            "selected": true,
            "text": "tomcat",
            "value": "tomcat"
          },
          {
            "selected": false,
            "text": "cassandra",
            "value": "cassandra"
          }
        ],
        "query": "tag_values(cpu.utilization.average,app)",
        "refresh": false,
        "regex": "",
        "type": "query"
      }
    ]
  }
```

テンプレートセクションの使用方法の詳細を以下の表に示します。

名前	使用方法
enable	テンプレートが有効になっているかどうか。
list	1 つのテンプレート変数を表すオブジェクトの配列
allFormat	データソースからすべての値を取得する際に使用する形式 (例: `wildcard`、`glob`、`regex`、`pipe`)。
current	ダッシュボードに現在選択されている変数テキスト/値を表示します
データソース	変数のデータソースを表示します
includeAll	すべての値オプションが使用可能かどうか
multi	変数値リストから複数の値を選択できるかどうか

名前	使用方法
multiFormat	データソースから時系列を取得する際に使用する形式
[名前]	変数の名前。
options:	ダッシュボードで選択できる可変テキスト/値のペアの配列
query	変数の値の取得に使用されるデータソースクエリ
[更新]	TODO
[正規表現]	TODO
type	変数のタイプ (`custom`、`query`、または `interval`)。

スクリプト化ダッシュボード

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

⚠ Warning

この機能は廃止されており、将来のリリースでは削除されます。

定義されたパターンで変更されるメトリクス名 (新しいサーバーなど) が多数ある場合、常に新しいダッシュボードを作成するには時間がかかる場合があります。

スクリプト化ダッシュボードでは、JavaScript を使用してダッシュボードを動的に作成できます。Grafana のインストールフォルダの `public/dashboards/` の下に、`scripted.js` という

名前のファイルがあります。このファイルにはスクリプト化ダッシュボードの例が含まれています。URL `http://grafana_url/dashboard/script/scripted.js?rows=3&name=myName` を使用してアクセスできます。

`scripted.js` を開くと、ARGS 変数から URL パラメータを読み取ってから行とパネルを追加する方法を確認できます。

例: scripted.js

```
var seriesName = 'argName';

if (!_.isUndefined(ARGS.name)) {
  seriesName = ARGS.name;
}

dashboard.panels.push({
  title: 'Events',
  type: 'graph',
  fill: 1,
  linewidth: 2,
  gridPos: {
    h: 10,
    w: 24,
    x: 0,
    y: 10,
  },
  targets: [
    {
      target: "randomWalk('" + seriesName + "')",
    },
    {
      target: "randomWalk('random walk2')",
    },
  ],
});

return dashboard;
```

その他の の例

その他の例は、Grafana インストールの `public/dashboards/` ディレクトリに収められています。

Explore

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana ワークスペースでは、ダッシュボード UI に視覚的表現用のダッシュボードを構築するためのツールが提供されています。Explore では、クエリに集中できるように、ダッシュボードとパネルのオプションが削除されます。クエリが機能するまで繰り返し調整し、その後ダッシュボードを計画して構築します。

インフラの監視やインシデント対応のために障害をデバッグする際に、他のツールに切り替える必要がなくなりました。Explore により、メトリクスとログをより深く掘り下げて原因を特定することができます。

Explore では、ダッシュボードを作成せずにデータを簡単に表示することができます。データソースがグラフとテーブルのデータをサポートしている場合、Explore には結果をグラフとテーブルの両方が表示されます。これにより、データの傾向と詳細を同時に確認できます。

探索を開始する

Note

デフォルトでは、閲覧者ロールが付与されたユーザーは編集できず、Explore へのアクセス権を持ちません。

左側のメニューの [Explore] アイコンをクリックすると、空の Explore タブが開きます。

パネル内の既存のクエリから開始するには、[パネル] メニューから [Explore] オプションを選択します。選択すると、パネルからのクエリを含む Explore タブが開きます。その後、ダッシュボードの外部でクエリを調整または反復処理することができます。

左上のドロップダウンからデータソースを選択します。Prometheus にはカスタム Explore 実装があります。他のデータソースでは、標準のクエリエディタを使用します。

クエリフィールドでは、クエリを記述し、データを探索できます。クエリフィールドの横に、クリアボタン (X)、クエリ追加ボタン (+)、クエリ削除ボタン (-) の 3 つのボタンが表示されます。パネルクエリエディタと同様に、複数のクエリを追加または削除できます。

分割と比較

分割ビューを使用すると、グラフと表を並べて比較したり、関連するデータを 1 ページでまとめて表示することができます。[分割] を選択して現在のクエリを複製し、ページを 2 つのクエリを並列に分割します。複製したクエリに、別のデータソースを選択するオプションが表示されます。この機能を使用して、同じクエリで 2 つサーバーを比較したり、ステージング環境と本番環境を比較することができます。

分割ビューのどちらかで、タイムピッカーに付いている [時刻の同期] ボタンを選択すると、両方のパネルのタイムピッカーをリンクすることができます (一方を変更すると、他方の他のパネルも変更されます)。タイムピッカーをリンクすると、分割ビュークエリの開始時刻と終了時刻が同期されるため、両方の分割パネルで同じ時間間隔を確認できます。

複製した方のクエリを閉じるには、[分割を閉じる] を選択します。

短縮リンクの共有

短縮リンクの共有機能を使用すると、複雑なクエリパラメータを含む長い URL を共有する代わりに、`/goto/:uid` 形式のより短いシンプルな URL を作成することができます。短縮リンクを作成するには、Explore ツールバーで [共有] オプションを選択します。使用されていない短縮リンクは、7 日後に自動的に削除されます。

クエリ履歴

クエリ履歴は、Explore で使用したクエリの一覧です。履歴は使用しているブラウザにのみ有効であり、共有されません。履歴の表示と操作は、Explore で [クエリ履歴] を選択します。

クエリ履歴の表示

クエリ履歴では、以下の操作が可能です。

- クエリを実行する
- コメントを作成または編集する。
- クエリをクリップボードにコピーする。
- クエリを含む短縮リンクをクリップボードにコピーする。

- クエリをスター付けする。

お気に入りのクエリの管理

[クエリ履歴] タブにあるスター付きのすべてのクエリは、[スター付き] タブに表示されます。お気に入りのクエリにすばやくアクセスし、それらのクエリを再入力することなく再利用できます。

クエリ履歴の並べ替え

デフォルトでは、クエリ履歴には最新のクエリが表示されます。履歴は日付またはデータソース名で、昇順または降順に並べ替えることができます。

クエリ履歴の右側にあるドロップダウンリストで、フィールドオプションから次のいずれかを選択します。

- 新しい順
- [古い順]
- データソース A→Z
- データソース Z→A

> **Note**
>
> 分割ビューの場合、ソートモードはアクティブなパネルにのみ適用されます。

クエリ履歴のフィルタリング

[クエリ履歴] と [スター付き] タブでは、データソース名でクエリ履歴をフィルタリングすることができます。

1. [特定のデータソース (複数可) のクエリをフィルタリング] を選択します。
2. 履歴のフィルタリングに使用するデータソースを選択します。複数のデータソースを選択することもできます。

[クエリ履歴] タブでは、垂直スライダーを使用してクエリを日付でフィルタリングできます。

- 下部のハンドルをドラッグして、開始日を調整します。

- 上部のハンドルをドラッグして、終了日を調整します。

> **Note**
>
> 分割ビューの場合、フィルターはアクティブなパネルにのみ適用されます。

クエリ履歴での検索

クエリとコメントをまたいで履歴を検索できます。[クエリ履歴] タブと [スター付き] タブのクエリを検索できます。

1. [検索クエリ] フィールドを選択します。
2. 検索フィールドに、検索語句を入力します。

クエリ履歴設定

[設定] タブでクエリ履歴をカスタマイズできます。以下の表に、使用できるオプションの一覧を紹介します。

設定	デフォルト値
Grafana にクエリ履歴が保存される期間を指定します。	1 週間
デフォルトでアクティブになるタブを変更します。	[クエリ履歴] タブ
Explore で現在アクティブなデータソースに対してのみクエリを表示します。	真
クエリ履歴をクリアします。	([クエリ履歴をクリア] を選択すると、保存されたすべてのクエリが完全に削除されます。)

> **Note**
>
> クエリ履歴設定は全体に及び、分割モードで両方のパネルに適用されます。

Prometheus 固有の機能

Explore の最初のバージョンには、Prometheus のカスタムクエリエクスペリエンスが含まれています。クエリを実行すると、Grafana ではグラフ用の通常の Prometheus クエリとテーブル用のインスタントクエリの 2 つのクエリが実行されます。インスタントクエリは、グラフに表示されるデータの適切な要約を示す各時系列の最後の値を返します。

メトリクスエクスプローラー

クエリフィールドの左側で [メトリクス] を選択すると、メトリクスエクスプローラーが開きます。エクスプローラーには、メトリクスがプレフィックスごとにグループ化された階層メニューが示されます。例えば、すべてのアラートマネージャーメトリクスは `alertmanager` プレフィックスの下にグループ化されます。これにより、利用可能なメトリクスの確認が容易になります。

クエリフィールド

クエリフィールドは、メトリクス名、関数の自動補完機能がサポートされており、標準の Prometheus クエリエディタとほぼ同じように動作します。[Enter] を押すとクエリが実行されます。

Ctrl+Space キーを押すとオートコンプリートメニューが開きます。オートコンプリートメニューには新たに [履歴] セクションが追加されており、最近実行されたクエリの一覧が表示されます。

クエリフィールドの下に候補が表示されます。候補を選択すると、クエリがその内容で更新されます。

- カウンター (指数関数的に増加するメトリクス) には、レート関数が候補として表示されます。
- バケットのには、ヒストグラム関数が候補として表示されます。
- 記録ルールの場合、ルールを展開できます。

テーブルをフィルタリング

テーブルパネルの [ラベル] 列にある [フィルター] ボタンを選択すると、クエリ式にフィルターを追加することができます。フィルターは複数のクエリに追加することもできます。フィルターはすべてのクエリに追加されます。

ログの統合

Explore では、ログの調査に以下のデータソースを使用することもできます。

- InfluxDB
- Elasticsearch

ログの視覚的表現

ログクエリの結果はグラフにヒストグラムとして表示され、個々のログはその下に表示されます。データソースから要求した時間範囲のヒストグラムデータが送信されない場合、ログモデルは、自動的に計算された時間間隔でバケット化されたログ行数を基に時系列を計算します。ヒストグラムの開始点は、結果の最初のログ行のタイムスタンプで固定されます。時系列の終了点は、タイムピッカーの To 範囲に固定されます。

ログレベル

レベルラベルが指定されているログの場合、Grafana はラベルの値を使用してログレベルを判別し、それに応じて色を更新します。ログにレベルラベルが指定されていない場合、Grafana はログを解析して、その内容がサポートされている式のいずれかと一致するかどうかを確認します。ログレベルは常に最初の一致によって決まります。Grafana がログレベルを判別できない場合、不明なログレベルとして視覚化されます。以下に、ログレベルとログレベルの略語と式の対応表を示します。

サポートされている式	ログレベル	色
emerg	critical	紫色
fatal	critical	紫色
alert	critical	紫色
crit	critical	紫色
critical	critical	紫色
err	エラー	赤
eror	エラー	赤
エラー	エラー	赤
warn	警告	黄色
警告	警告	黄色
情報	情報	緑色
information	情報	緑色

サポートされている式	ログレベル	色
notice	情報	緑色
dbug	デバッグ	青
デバッグ	デバッグ	青
トレース	トレース	水色
*	不明	灰色

視覚的表現オプション

ログの表示方法をカスタマイズしたり、表示する列を選択することができます。

[時間]

時間列の表示と非表示を切り替えます。これはデータソースから報告されたログ行に関連付けられているタイムスタンプです。

一意のラベル

非一般的なラベルのみを含む一意のラベル列の表示と非表示を切り替えます。すべての一般的なラベルが上に表示されます。

行の折り返し

行を折り返して表示する場合、[真] に設定します。[偽] に設定すると、水平スクロールになります。

重複排除

ログデータは非常に繰り返しが多いことがあります。Explore では、重複するログ行を非表示にすることができます。さまざまな重複排除アルゴリズムを選択できます。

- [完全] - 完全一致は、日付フィールドを除き、行全体で行われます。
- [数値] - の一致は、期間、IP アドレスなどの数値を除外した後に行で行われます。
- 署名 – 最も積極的な重複排除で、すべての文字と数字を除きます。一致は残りの空白文字と句読点に対して行われます。

結果の順序を反転する

受信したログの順序は、デフォルトの降順 (新しい順) から昇順 (古い順) に変更できます。

ラベルと検出されたフィールド

各ログ行には、より堅牢なインタラクションのために、ラベルと検出されたフィールドを含む拡張可能な領域があります。すべてのラベルについて、選択したラベルを含める (ポジティブフィルター) または除外する (ネガティブフィルター) フィルターを適用することができます。各フィールドまたはラベルには、表示されているすべてのログに関連する統計を1回で表示する統計アイコンがあります。

検出されたフィールドの切り替え

ログが JSON または logfmt で構造化されている場合、検出されたフィールドを表示または非表示にできます。ログ行を展開し、目のアイコンを選択してフィールドの表示または非表示を切り替えます。

{{< docs-imagebox img="/img/docs/explore/parsed-fields-7-2.gif" max-width="800px" caption="Explore の検出されたフィールドの切り替え" >}}

トレース統合

Explore では、データソースのトレースからのトレースを視覚化できます。以下は現在サポートされているデータソースです。

- Jaeger データソースへの接続
- Tempo データソースへの接続
- AWS X-Ray データソースに接続する
- Zipkin データソースへの接続

クエリエディタの使用については、該当するデータソースのドキュメントを参照してください。

ヘッダー

このヘッダーには、次の項目が含まれます。

- ヘッダータイトル (ルートスパンの名前とトレース ID が表示されます)
- 検索 (検索されたテキストを含む時間範囲が強調表示されます)

- トレースに関するメタデータ

ミニマップ

ミニマップには、要約ビューまたはトレースタイムラインが表示されます。マウスをミニマップ上でドラッグすると、より短い時間範囲にズームインすることができます。ズームするとメインタイムラインも更新されるため、短い時間範囲も簡単に確認することができます。ズームされたミニマップ上にカーソルを合わせて表示される [選択のリセット] ボタンを押すと、ズームがリセットされます。

タイムライン

タイムラインには、トレース内の時間範囲のリストが表示されます。各スパン行は以下で構成されます。

- [子を展開] ボタン: 選択したスパンのすべての子スパンを展開または折りたたみます
- サービス名: スパンをログに記録したサービスの名前
- オペレーション名: このスパンが表すオペレーションの名前
- スパン期間バー: トレース内のオペレーション期間の視覚的表現

スパン行の任意の場所を選択すると、スパンの詳細が表示されます。

スパンの詳細

スパンの詳細には、次の項目が含まれます。

- オペレーション名
- スパンメタデータ
- タグ (このスパンに関連付けられているタグ)
- プロセスメタデータ (このスパンをログに記録したプロセスに関するメタデータ)
- ログ: このスパンおよび関連するキー値によってログに記録されたログのリスト。Zipkin ログの場合、セクションには Zipkin 注釈が表示されます。

Explore とダッシュボード間の移動

Explore とダッシュボード間の頻繁な行き来を効率化するため、パネルのドロップダウンから Explore に移動した後、元のダッシュボードに戻る機能を追加しました。

Explore に移動後、ツールバーに [戻る] ボタンが表示されます。

ボタンを選択するだけで元のダッシュボードに戻ることができます。また、Explore で行った変更をダッシュボードにも反映させたい場合、ボタン横の矢印をクリックして、[変更を反映してパネルに戻る] メニュー項目を選択します。

クエリインスペクター

クエリのデバッグを支援するために、Explore ではクエリリクエストとレスポンス、クエリ統計を「クエリインスペクター」を使用して調査できます。この機能は、パネルインスペクターの [統計] タブと [クエリ] タブと似たものです。詳細については、「クエリパフォーマンスの検査」および「データソースへの未加工リクエストとレスポンスの表示」を参照してください。

Linking (リンク中)

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

リンクを使用すると、よく使用されるダッシュボード間を移動、または他のユーザーを視覚化に接続できます。リンクでは、他のダッシュボード、パネル、さらには外部ウェブサイトへのショートカットを作成できます。

Amazon Managed Grafana では、ダッシュボードリンク、パネルリンク、データリンクをサポートしています。ダッシュボードリンクはダッシュボードの上部に表示されます。パネルリンクにアクセスするためには、パネルの左上隅にあるアイコンを選択します。

どのリンクを使用する必要がありますか。

最初にダッシュボード間の現在の移動方法を確認します。頻繁に一連のダッシュボード間でジャンプし、各ダッシュボードで同じコンテキストを見つけるのに苦労している場合、ワークフローの最適化にリンクが役立ちます。

次のステップでは、ご使用のワークフローに適したリンクタイプを特定します。Grafana のすべてのリンクタイプは他のダッシュボードや外部ウェブサイトへのショートカットの作成に使用されますが、それらはさまざまなコンテキストで動作します。

- ダッシュボード内のパネルの大部分またはすべてに関連するリンクを追加するには、ダッシュボードリンク を使用します。
- 特定のパネルにドリルダウンするには、パネルリンク を使用します。
- 外部サイトへのリンクには、ダッシュボードリンクまたはパネルリンクのいずれかを使用できます。
- 特定のシリーズ、または 1 つの測定までドリルダウンするには、データリンク を使用します。

URL の使用による時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定すると、パネルまたはダッシュボードの時間範囲を制御できます。

- `from`は、エポックミリ秒で指定した、時間範囲の下限を定義します。
- `to`は、エポックミリ秒で指定された時間範囲の上限を定義します。
- `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータは、ミリ秒単位で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は 1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

ダッシュボードリンク

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

ダッシュボードリンクを作成するときに、時間範囲と現在のテンプレート変数を含めることで、別のダッシュボードの同じコンテキストに直接ジャンプできます。これにより、リンクを送信した相手が適切なデータを確認できるようになります。その他のタイプのリンクについては、「データリンク変数」を参照してください。

追加したダッシュボードリンクは、ダッシュボードの右上隅に表示されます。

ダッシュボードへのリンクの追加

現在のダッシュボードの上部にある他のダッシュボードへのリンクを追加します。

1. リンクを追加するダッシュボードを表示している状態で、画面上部の歯車アイコンを選択して [ダッシュボード設定] を開きます。
2. [リンク] を選択し、[ダッシュボードリンクの追加]] または 新規] を選択します。
3. [タイプ] で、[ダッシュボード] を選択します。
4. リンクオプションを選択します。

   - [タグあり] – タグを入力して、リンクされたダッシュボードを、入力したタグを持つダッシュボードのみに制限します。それ以外の場合、Grafana ワークスペースには他のすべてのダッシュボードへのリンクが含まれます。
   - [As dropdown] (ドロップダウン) - 多くのダッシュボードにリンクする場合はこれを選択して、ドロップダウンリストの選択肢となるタイトルを追加します。これを選択しなかった場合、ダッシュボードのリンクは Grafana ワークスペースのダッシュボードの上部に並べて表示されます。
   - [時間範囲] – このオプションを選択すると、ダッシュボードの時間範囲がリンクに含まれます。ユーザーがリンクを選択すると、指定された時間範囲が既に設定された状態でリンクされたダッシュボードが開きます。
   - [変数値] – このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクを選択すると、リンクされたダッシュボード内の一致するテンプレートがリンクの値に設定されます。
   - [新しいタブで開く] – ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。
5. [Add] (追加) を選択します。

ダッシュボードへの URL リンクの追加

現在のダッシュボードの上部にある URL へのリンクを追加します。ダッシュボード、パネル、外部サイトなど、使用可能なあらゆる URL にリンクできます。ユーザーが Grafana ワークスペースの特定のデータを表示できるように、時間範囲を制御できます。

1. リンクするダッシュボードを表示している状態で、画面上部の歯車アイコンを選択して [ダッシュボード設定] を開きます。
2. [リンク] を選択し、[ダッシュボードリンクの追加]] または 新規] を選択します。

3. [タイプ] で [リンク] を選択します。
4. リンクオプションを選択します。

   - URL – リンク先の URL を入力します。ターゲットによっては、フィールド値を含めることもできます。
   - [タイトル] — リンクを表示するタイトルを入力します。
   - Tooltip (ツールヒント) – リンクにカーソルを合わせたときに表示するツールヒントを入力します。
   - Icon (アイコン) – リンクとともに表示するアイコンを選択します。
   - [時間範囲] – このオプションを選択すると、ダッシュボードの時間範囲がリンクに含まれます。ユーザーがリンクを選択すると、指定された時間範囲が既に設定された状態でリンクされたダッシュボードが開きます。
     - `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。
     - `to` は、エポックミリ秒で指定した、時間範囲の上限を定義します。
     - `time` および `time.window` は、`time-time.window/2` から `time+time.window/2` までの時間範囲を定義します。両方のパラメータは、ミリ秒単位で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は 1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。
   - [変数値] – このオプションを選択すると、リンクにクエリパラメータとして現在使用されているテンプレート変数を含めることができます。ユーザーがリンクを選択すると、リンク先のダッシュボードに一致するテンプレートがリンクの値に設定されます。例: https://play.grafana.org/d/000000074/alerting?var-app=backend&var-server=backend_01&var-server=backend_03&var-interval=1h
   - [新しいタブで開く] – ダッシュボードリンクを新しいタブまたはウィンドウで開く場合は、このオプションを選択します。
5. [Add] (追加) を選択します。

ダッシュボードリンクの更新

既存のダッシュボードリンクを変更または更新するには、次の手順に従います。

1. [ダッシュボード設定], の[リンク] タブで、編集する既存のリンクを選択します。
2. 設定を変更し、[更新] を選択します。

ダッシュボードリンクの複製

既存のダッシュボードリンクを複製するには、複製する既存のリンクの横にある複製アイコンを選択します。

ダッシュボードリンクの削除

既存のダッシュボードリンクを削除するには、削除するリンクのゴミ箱アイコンを選択します。

パネルリンク

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

各パネルには、パネルの左上隅に表示する一連の独自のリンクを含めることができます。ダッシュボード、パネル、外部サイトなどの、使用可能な URL にリンクできます。ユーザーが Grafana ワークスペースの特定のデータを表示できるように、時間範囲を制御できます。

パネルの左上隅にあるアイコンを選択すると使用可能なパネルリンクが表示されます。

パネルリンクの追加

1. リンクを追加するパネルで一時停止して、e を押します。または、パネルタイトルの横にあるドロップダウン矢印を選択して、[編集] を選択します。
2. [パネル] タブで、[リンク] セクションまでスクロールダウンします。
3. [リンク] を展開して、[リンクの追加] を選択します。
4. リンクの [タイトル] を入力します。そのタイトルは UI に表示されます。
5. リンクする [URL] を入力します。ダッシュボードで定義済みのテンプレート変数のいずれかを含めることができます。Ctrl+スペース または Cmd+スペース を押してから、URL フィールドを選択して使用可能な変数を表示します。パネルリンクにテンプレート変数を追加すると、リンクは関連する変数を設定済みの、適切なコンテキストを開きます。時間変数も使用できます。

   - `from` は、エポックミリ秒で指定した、時間範囲の下限を定義します。

- to は、エポックミリ秒で指定した、時間範囲の上限を定義します。
- time および time.window は、time-time.window/2 から time+time.window/2 までの時間範囲を定義します。両方のパラメータは、ミリ秒単位で指定する必要があります。例えば、?time=1500000000000&time.window=10000 は 1499999995000 から 1500000005000 までの 10 秒の時間範囲になります。

6. 新しいタブで開くには、[新しいタブで開く] を選択します。
7. [保存] を選択して変更を保存し、ウィンドウを閉じます。
8. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

パネルリンクの更新

1. [パネル] タブで、変更するリンクを見つけます。
2. [編集] (鉛筆) アイコンを選択して、[リンクの編集] ウィンドウを開きます。
3. 必要に応じて変更します。
4. [保存] を選択して変更を保存し、ウィンドウを閉じます。
5. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

パネルリンクの削除

1. [パネル] タブで、削除するリンクを見つけます。
2. 削除するキーの横にある [X] アイコンを選択します。
3. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

データリンク

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

データリンクは、リンクにより詳細なコンテキストを提供します。系列名または値を含むリンクを作成できます。例えば、視覚的表現で 4 つのサーバーが表示されている場合は、そのうちの 1 つまたは 2 つにデータリンクを追加できます。

リンク自体には、視覚的表現に応じてさまざまな方法でアクセスできます。グラフパネルでは、データポイントまたは線を選択する必要があります。統計、ゲージ、棒ゲージなどのパネルでは、視覚的表現の任意の場所を選択してコンテキストメニューを開くことができます。

データリンクに変数を使用すると、データフィルターが設定された状態の詳細なダッシュボードにユーザーを移動させることができます。例えば、 変数を使用して、時間範囲、系列、または変数の選択を指定できます。詳細については、「データリンク変数」を参照してください。

入力候補機能

データリンクを作成または更新する際に、キーボードで [Ctrl+Space] または [Cmd+Space] を押すと入力候補が表示され、URL に変数をより簡単に追加できます。

データリンクの追加

1. リンクを追加するパネルで一時停止して、e を押します。または、パネルタイトルの横にあるドロップダウン矢印を選択して、[編集] を選択します。
2. [フィールド] タブで、[データリンク] セクションまでスクロールダウンします。
3. [データリンク] を展開して、[リンクの追加] を選択します。
4. リンクの [タイトル] を入力します。そのタイトルは UI に表示されます。
5. リンクする [URL] を入力します。

   ダッシュボードで定義済みのテンプレート変数のいずれかを追加することができます。URL フィールドを選択し、$ と入力するか、Ctrl+Space または Cmd+Space を押すと、使用可能な変数の一覧が用事されます。パネルリンクにテンプレート変数を追加すると、リンクは関連する変数を設定済みの、適切なコンテキストを開きます。詳細については、「データリンク変数」を参照してください。
6. 新しいタブで開くには、[新しいタブで開く] を選択します。
7. [保存] を選択して変更を保存し、ウィンドウを閉じます。
8. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

データリンクの更新

1. [フィールド] タブで、変更するリンクを見つけます。

2. [編集] (鉛筆) アイコンを選択して、[リンクの編集] ウィンドウを開きます。
3. 必要に応じて変更します。
4. [保存] を選択して変更を保存し、ウィンドウを閉じます。
5. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

データリンクの削除

1. [フィールド] タブで、削除するリンクを見つけます。
2. 削除するキーの横にある [X] アイコンを選択します。
3. 右上の [保存] を選択し、この変更をダッシュボードに保存します。

データリンク変数

データリンクで変数を使用すると、シリーズフィールド、ラベル、値を表示できます。データリンクの詳細については、「データリンク」を参照してください。

使用可能な変数のリストを表示するには、データリンクの URL フィールドに $ を入力します。

データリンク URL ではテンプレート変数も使用できます。詳細については、「テンプレートおよび変数」を参照してください。

時間範囲パネル変数

次の変数を使用すると、データリンク URL に現在の時間範囲を含めることができます。

- `__url_time_range` – 現在のダッシュボードの時間範囲。例: `?from=now-6h&to=now`
- `$__from and $__to` – 詳細については、「[グローバル変数]({{< relref "../variables/variable-types/global-variables.md#__from-and-__to" >}})」を参照してください。

系列変数

シリーズ固有の変数は、`__series` 名前空間で使用できます。

- `__series.name` — URL へのシリーズ名の追加
- `__series.labels.<LABEL>` – ラベルの値を URL に追加します。ラベルにドットが含まれている場合は、`__series.labels["<LABEL>"]` 構文を使用します。

フィールド変数

フィールド固有の変数は、__field 名前空間で使用できます。

- __field.name – フィールドの名前

値変数

値固有の変数は、__value 名前空間で使用できます。

- __value.time – URL への値のタイムスタンプ (Unix ms エポック)。例えば ?time=1560268814105
- __value.raw – 未加工値。
- __value.numeric – 値の数値表現
- __value.text – 値のテキスト表現
- __value.calc – 値が計算結果である場合の計算名

テンプレート変数

テンプレート変数を使用する別のダッシュボードにリンクするときは、リンクを選択した人の変数値を選択します。

var-myvar=${myvar} を使用します。ここで、myvar は使用する現在のダッシュボードのテンプレート変数の名前です。

現在のダッシュボードのすべての変数を URL に追加するには、__all_variables を使用します。

テンプレートおよび変数

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

変数は、値のプレースホルダーです。メトリクスクエリとパネルタイトルで変数を使用できます。変数を使用するとよりインタラクティブで動的なダッシュボードを作成できます。メトリクスクエリでサーバー、アプリケーション、センサー名などをハードコーディングする代わりに、変数を使用できます。

変数は、ダッシュボードの上部にドロップダウンリストとして表示されます。ダッシュボードの上部にあるドロップダウンリストを使用して値を変更すると、パネルのメトリクスクエリには新しい値が反映されます。

これらは、ビューワーが視覚化をすばやく調整できるようにしたいが、完全な編集アクセス許可を付与したくない管理者にとって特に便利です。Grafana ビューワーは変数を使用できます。

変数とテンプレートを使用すると、単一ソースのダッシュボードにすることができます。同じデータソースまたはサーバーが複数ある場合は、1 つのダッシュボードを作成し、変数を使用して表示内容を変更できます。これにより、メンテナンスと維持が簡素化されます。

サポートされている変数タイプのリスト、各変数タイプの追加手順については、「変数タイプ」を参照してください。

テンプレート

[テンプレート]は、変数を含むすべてのクエリです。

例えば、複数サーバーをモニタリングするためにダッシュボードを管理している場合は、サーバーごとにダッシュボードを作成できます。または、次の例に示すように、1 つのダッシュボードを作成して、テンプレートクエリでパネルを使用できます。

```
wmi_system_threads{instance=~"$server"}
```

変数値は構文 `var-<varname>=value` を使用して常に URL に同期されます。

変数のベストプラクティス

変数のドロップダウンリストは、[ダッシュボード設定] の変数リストに一覧されている順序で表示されます。

頻繁に変更する変数を一番上に置くと、ダッシュボードの左端に最初に表示されるようになります。

変数構文

パネルタイトルとメトリクスクエリは、次の 2 つの異なる構文を使用して変数を参照できます。

- `$varname` これは次の例のように読みやすい構文です。`apps.frontend.$server.requests.count`ただし、単語の途中で変数を使用することができません。
- `${var_name}` この構文は式の途中で変数を補間する場合に使用します。
- `${var_name:<format>}` この形式により、Grafana が値を解釈する方法をより細かく制御できます。詳細については、「高度な変数フォーマットオプション」を参照してください。

クエリがデータソースに送信される前に、クエリは[補間されます]。つまり、変数は現在の値に置き換えられます。補間中、変数値は、クエリ言語の構文と使用される場所に準拠するようにエスケープされる可能性があります。例えば、Prometheus クエリの正規表現で使用される変数は、正規表現からエスケープされます。補間中の値のエスケープの詳細については、データソース固有のドキュメントトピックを参照してください。

データソースのデフォルトのフォーマットを上書きする高度な構文については、「高度な変数フォーマットオプション」を参照してください。

変数タイプ

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana は、変数のタイプをいくつか使用します。

変数タイプ	説明
クエリ	メトリクス名、サーバー名、センサー ID、データセンターなど、クエリによって生成された値のリストです。詳細については、「クエリ変数の追加」を参照してください。

変数タイプ	説明
カスタム	カンマ区切りリストを使用して変数オプションを手動で定義します。詳細については、「カスタム変数の追加」を参照してください。
テキストボックス	任意のデフォルト値を使用してテキスト入力フィールドを表示します。詳細については、「テキストボックス変数の追加」を参照してください。
定数	非表示の定数を定義します。詳細については、「定数変数の追加」を参照してください。
データソース	ダッシュボード全体のデータソースをすばやく変更します。詳細については、「データソース変数の追加」を参照してください。
間隔	間隔変数は時間範囲を表します。詳細については、「間隔変数の追加」を参照してください。
アドホックフィルター	データソースのすべてのメトリクスクエリに自動的に追加されるキー/値のフィルターを指します (InfluxDB 、Prometheus、OpenSearch のみ)。詳細については、「アドホックフィルターの追加」を参照してください。
グローバル変数	クエリエディタの式で使用できる組み込み変数です。詳細については、「グローバル変数」を参照してください。
連鎖変数	変数クエリには、他の変数を含めることができます。詳細については、「連鎖変数」を参照してください。

クエリ変数の追加

クエリ変数を使用すると、メトリクス名、タグ値、またはキーのリストを返すデータソースクエリを記述できます。例えば、クエリ変数は、サーバー名、センサー ID、またはデータセンターのリストを返す場合があります。変数値は、データソースクエリでオプションを動的に取得するにつれて変化します。

クエリ式には、他の変数への参照を含めることができ、実際にはリンクされた変数を作成できます。Grafana はこれを検出し、リンクされた変数のいずれかが変更されると変数を自動的に更新します。

クエリ式

クエリ式はデータソースごとに異なります。詳細については、データソースに接続するのデータソースに関するドキュメントを参照してください。

一般的なオプションの入力

クエリ変数の一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の[名前]を入力します。
4. [タイプ] 一覧で、[クエリ] を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
   - [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

クエリオプションの入力

クエリ変数のクエリオプションを入力するには

1. [データソース] リストで、クエリのターゲットデータソースを選択します。データソースの詳細については、「データソースに接続する」を参照してください。
2. [更新] リストで、変数がオプションを更新するタイミングを選択します。
   - [なし] - 変数クエリをキャッシュし、値は更新されません。これは、値が変更されない場合は問題ありませんが、動的で大きく変更される場合は問題になります。
   - [ダッシュボード読み込み時] - ダッシュボードが読み込まれるたびにデータソースにクエリを実行します。これにより、ダッシュボードの読み込みが遅くなります。ダッシュボードを初期化する前に変数クエリを完了する必要があるためです。
   - [時間範囲の変更時] - ダッシュボードの時間範囲が変更されたときにデータソースにクエリを実行します。このオプションは、変数のクエリオプションに時間範囲フィルターが含まれているか、ダッシュボードの時間範囲に依存している場合にのみ使用します。
3. [クエリ] フィールドにクエリを入力します。
   - クエリフィールドはデータソースによって異なります。一部のデータソースには、カスタムクエリエディタがあります。
   - 1 つの入力フィールドクエリエディタにより多くのスペースが必要な場合は、フィールドの右下隅にある線で一時停止し、下にドラッグして展開します。
4. (オプション) [正規表現] フィールドに正規表現を入力して、データソースクエリによって返される名前の特定部分をフィルタリングまたはキャプチャします。例については「正規表現での変数のフィルタリング」を参照してください。
5. [ソート] リストで、ドロップダウンリストに表示される値のソート順序を選択します。デフォルトのオプション [無効] では、データソースクエリによって返されるオプションの順序が使用されます。
6. (オプション) [選択オプション] を入力します。詳細については、「変数選択オプションの入力」を参照してください。
7. [値のプレビュー] では、Grafana ワークスペースに現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
8. [追加] を選択してダッシュボードに変数を追加します。

カスタム変数の追加

変更されない値にはカスタム変数を使用します。これは、数値、文字列、またはその他の変数である場合があります。

例えば、サーバー名やリージョン名が変更されない場合は、クエリ変数ではなくカスタム変数として作成できます。これらは変更されないため、他のクエリ変数ではなく、連鎖変数で使用できます。これにより、連鎖変数が更新されたときに Grafana が送信する必要があるクエリの数が減少します。連鎖変数の詳細については、「連鎖変数」を参照してください。

一般的なオプションの入力

カスタム変数のクエリオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで [カスタム] を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。
   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
   - [ラベル] – 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

カスタムオプションの入力

カスタム変数のカスタムオプションを入力するには

1. [カンマで区切られた値] リストで、この変数の値をカンマ区切りリストに入力します。コロンで区切られた数値、文字列、その他の変数、またはキーと値のペアを含めることができます。
2. (オプション) [選択オプション] を入力します。詳細については、「変数選択オプションの入力」を参照してください。

3. [値のプレビュー] では、Grafana ワークスペースに現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

テキストボックス変数の追加

テキストボックス変数は、任意のデフォルト値を使用してテキスト入力フィールドを表示します。これは、任意の値を入力できるため、最も柔軟な変数です。カーディナリティの高いメトリクスがある場合や、ダッシュボード内の複数のパネルを同時に更新する場合は、このタイプの変数を使用します。

一般的なオプションの入力

テキストボックス変数の一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで、[テキストボックス] を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
   - [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

テキストオプションの入力

テキストボックス変数のテキストオプションを入力するには

1. (オプション) [デフォルト値] フィールドで、変数のデフォルト値を選択します。このフィールドに何も入力しない場合、Grafana はテキストを入力できる空のテキストボックスを表示します。
2. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。

3. [追加] を選択してダッシュボードに変数を追加します。

定数変数の追加

非表示の定数を定義するには、定数変数を使用します。定数変数は、共有するダッシュボードに対するメトリックパスの接頭辞に有用です。ダッシュボードをエクスポートすると、定数変数がインポートオプションに変換されます。

定数変数は複数設定できません。各定数変数は 1 つの値のみを保持します。更新するには、変数設定を更新する必要があります。

定数変数は、クエリに含める必要がある複素数値があっても、単一クエリすべてに再入力したくない場合に便利です。例えば、「i-0b6a61efe2ab843gg」というサーバーパスがある場合、それを「$path_gg」という変数に置き換えることができます。

一般的なオプションの入力

定数変数の一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで [Constant] (定数) を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。これがデフォルトです。
   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。
   - [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。

定数オプションの入力

定数変数の定数オプションを入力するには

1. [値] フィールドに変数値を入力します。英数字と記号を入力できます。高度な変数の形式オプションを使用する場合は、ワイルドカードを使用することもできます。詳細については、「高度な変数フォーマットオプション」を参照してください。
2. [値のプレビュー] では、Grafana ワークスペースに現在の変数値が表示されます。これを確認して、期待どおりであることを確認します。
3. [追加] を選択してダッシュボードに変数を追加します。

データソース変数の追加

ダッシュボード全体のデータソースをすばやく変更するには、データソース変数を使用できます。データソースのインスタンスが複数ある場合、おそらく異なる環境にある場合に便利です。

一般的なオプションの入力

データソース変数の一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで、[Datasource] を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
   - [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

データソースオプションの入力

データソース変数のデータソースオプションを入力するには

1. [タイプ] リストで、変数のターゲットデータソースを選択します。データソースの詳細については、「データソースに接続する」を参照してください。
2. (オプション) [インスタンス名フィルター] には、変数値のドロップダウンリストからデータソースインスタンスを選択する正規表現フィルターを入力します。このフィールドは空のままにして、すべてのインスタンスを表示します。
3. (オプション) [選択オプション] を入力します。詳細については、「変数選択オプションの入力」を参照してください。
4. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
5. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の追加

間隔変数を使用して、1m、1h、1dなどの時間範囲を表します。これらはダッシュボード全体の group-by-time コマンドと考えることができます。間隔変数は、視覚化でのデータのグループ化方法を変更します。また、[自動] オプションを使用して、時間範囲ごとに複数のデータポイントを返すこともできます。

間隔変数を時間 (InfluxDB の場合)、日付ヒストグラム間隔 (OpenSearch の場合) でグループ化するパラメータとして、または要約関数パラメータ (Graphite の場合) として使用できます。

一般的なオプションの入力

間隔変数の一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで [間隔] を選択します。
5. (オプション) [ラベル]には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

- 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
- [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
- [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

間隔オプションの入力

間隔変数の間隔オプションを入力するには

1. [値] フィールドに、変数ドロップダウンリストに表示する時間間隔を入力します。次の時間単位に対応しています: `s`（seconds）、`m`（minutes）、`h`（hours）、`d`（days）、`w`（weeks）、`M`（months）、`y`（years）。次のデフォルト値を使用または編集することもできます: `1m,10m,30m,1h,6h,12h,1d,7d,14d,30d`。
2. (オプション) `auto`オプションをリストに追加する場合は、[Auto Option] を(自動オプション) オンにします。このオプションを使用して、現在の時間範囲を分割して現在の`auto`時間範囲を計算する回数を指定します。オンにすると、さらに次の 2 つのオプションが表示されます。
   - [ステップ数] – [最大データポイント]クエリオプションと同様に、現在の時間範囲を分類して値を計算する回数を選択します。例えば、現在の表示時間範囲が 30 分の場合、`auto` 間隔はデータを 1 分単位× 30 でグループ化します。デフォルト値は 30 ステップです。
   - [最小間隔] – ステップ数の間隔が時間を分類しない最小しきい値です。引き続き 30 分の例を挙げる場合、最小間隔を `2m` に設定すると、Grafana はデータを 2 分単位× 15 でグループ化します。
3. [値のプレビュー] には現在の変数値の一覧が表示されます。これらを確認して、期待どおりであることを確認します。
4. [追加] を選択してダッシュボードに変数を追加します。

間隔変数の例

Graphite 関数でテンプレート変数 `myinterval` を使用する例:

```
summarize($myinterval, sum, false)
```

より複雑な Graphite の例:

```
groupByNode(summarize(movingAverage(apps.$app.$server.counters.requests.count, 5),
 '$interval', 'sum', false), 2, 'sum')
```

アドホックフィルターの追加

1 回限りのフィルターまたはアドホックフィルターを使用して、指定されたデータソースを使用するすべてのメトリクスクエリに自動的に追加されるキー/値フィルターを追加できます。他の変数とは異なり、クエリでは 1 回限りのフィルターを使用しません。代わりに、アドホックフィルターを使用して既存のクエリのフィルターを書き込みます。

> **Note**
>
> 注: 1 回限り、またはアドホックのフィルター変数は、InfluxDB 、Prometheus、および OpenSearch データソースでのみ機能します。

一般的なオプションの入力

アドホックフィルターの一般的なオプションを入力するには

1. 変数を作成するダッシュボードに移動し、ページ上部の [ダッシュボード設定] (歯車) アイコンを選択します。
2. [変数] タブで、[新規] を選択します。
3. 変数の名前を入力します。
4. [タイプ] リストで、[Ad hoc filters] (アドホックフィルター) を選択します。
5. (オプション) [ラベル] には、変数ドロップダウンリストの表示名を入力します。表示名を入力しない場合、ドロップダウンラベルは変数名になります。
6. [非表示] オプションを次から選択します。

   - 選択なし (空白) – 変数ドロップダウンリストには、変数の名前またはラベル値が表示されます。これがデフォルトです。
   - [ラベル]– 変数ドロップダウンリストには、選択した変数値と下矢印のみが表示されます。
   - [変数] – ダッシュボードには変数ドロップダウンリストが表示されません。

オプションの入力

アドホックフィルターのオプションを入力するには

1. [データソース] リストで、ターゲットデータソースを選択します。データソースの詳細については、「データソースに接続する」を参照してください。
2. [追加] を選択してダッシュボードに変数を追加します。

アドホックフィルターの作成

アドホックフィルターは、利用可能な最も複雑で変更可能な変数オプションの 1 つです。変数オプションの通常リストの代わりに、この変数によりダッシュボード全体のアドホッククエリを構築できます。この方法で適用するフィルターは、ダッシュボード上のすべてのパネルに適用されます。

連鎖変数

連鎖変数は、リンクされた変数またはネストされた変数とも呼ばれ、変数クエリ内において 1 つ以上の他の変数を持つクエリ変数です。このセクションでは、連鎖変数がどのように機能するかを説明し、連鎖変数を使用するダッシュボード例へのリンクを共有します。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべて同様です。連鎖変数クエリは、それらをサポートする任意のデータソースで使用できます。

複雑なリンクされたテンプレートダッシュボードを 5 階層または 10 階層深く構築できます。技術的には、どれだけ深く、複雑にできるかに制限はありませんが、リンクが多いほどクエリの負荷が大きくなります。

ベストプラクティスとヒント

以下のプラクティスにより、ダッシュボードと変数の使用が容易になります。

新しい連鎖変数の作成

- 連鎖変数は親子の依存関係を作成します。ラダーまたはツリー構造として想定できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数の基盤となる変数をコピーすることです。変数リストで、変数エントリの右側にある [Duplicate variable] (変数の複製) アイコンを選択してコピーを作成します。その後、親変数のクエリに追加できます。
- この方法で作成した新しい連鎖変数は、リストの下部に表示されます。リストに論理的順序を付けるには、変数をリスト内の別の位置にドラッグします。

変数の順序

ダッシュボードの変数リストにある変数の順序を変更するには、各エントリの右側にある上下の矢印を選択します。Grafana ワークスペースは、このリストに従って左から右に変数ドロップダウンリストを一覧表示し、左端のリストの上部に変数を表示します。

- 子変数の前に、親変数と依存関係のない変数のリストを作成します。
- 各変数は、依存している変数に従う必要があります。
- UI は、依存関係を持つ変数を示しません。変数を論理的な順序で一覧表示して、エンドユーザー (および自分自身) に明確にします。

複雑さに関する考慮事項

変数の依存関係のレイヤーが多いほど、変数を変更した後にダッシュボードを更新するのに時間がかかります。

例えば、4 項目がリンクされた一連の変数 (国、地域、サーバー、メトリック) があり、ルート変数値 (国) を変更する場合、Grafana ワークスペースはダッシュボード内の視覚化を更新する前に、すべての従属変数に対してクエリを実行する必要があります。

グローバル変数

Grafana には、クエリエディタの式で使用できるグローバル組み込み変数があります。このトピックでは、それらをアルファベット順に一覧表示し、定義します。これらの変数は、クエリ、ダッシュボードリンク、パネルリンク、データリンクに役立ちます。

$__dashboard

この変数は、現在のダッシュボードの名前です。

$__from および $__to

Grafana には、「`$__from`」と「`$__to`」の 2 つの組み込み時間範囲変数があります。これらは現在、デフォルトでは常にエポックミリ秒として補間されていますが、日付の書式設定をユーザーが管理できます。

構文	結果の例	説明
`${__from}`	1594671549254	Unix エポックミリ秒

構文	結果の例	説明
`${__from:date}`	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
`${__from:date:iso}`	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
`${__from:date:seconds}`	1594671549	Unix エポック秒
`${__from:date:YYYY-MM}`	2020-07	任意のカスタムデータ形式。詳細については、「表示」を参照してください。

上記の構文は「`${__to}`」を使用しても機能します。

この変数は URL でも使用できます。例えば、6 時間前から現在までの時間範囲を示すダッシュボードにエンドユーザーを送信するには、次の URL を使用します。https://play.grafana.org/d/000000012/grafana-play-home?viewPanel=2&orgId=1?from=now-6h&to=now

$__interval

`$__interval` 変数を時間 (InfluxDB 、Myself、Postgres、MSSQL の場合)、日付ヒストグラム間隔 (OpenSearch の場合) でグループ化するパラメータとして、または[要約]関数パラメータ (Graphite の場合) として使用できます。

Grafana ワークスペースは、クエリで時間別にグループ化するために使用できる間隔を自動的に計算します。グラフに表示されるデータポイントよりも多い場合、クエリはより長い間隔でグループ化することで、より効率的になります。例えば、3 か月分のデータを調べる場合、1 日でグループ化する方が 10 秒でグループ化するよりも効率的です。グラフは同じように見え、クエリは高速になります。`$__interval` は、時間範囲とグラフの幅 (ピクセル数) を使用して計算されます。

概算: `(from - to) / resolution`

例えば、時間範囲が 1 時間でグラフが全画面表示の場合、間隔は 2m で計算され、ポイントは 2 分間隔でグループ化されます。時間範囲が 6 か月で、グラフが全画面表示の場合、間隔は 1d (1 日) で、ポイントは日ごとにグループ化されます。

InfluxDB データソースでは、レガシー変数 $interval は同じ変数です。代わりに $__interval を使用します。

InfluxDB データソースと OpenSearch データソースには、間隔をハードコード化したり、> 構文 -> Group by time interval を使用して $__interval 変数の最小制限を設定したりするために使用される >10m フィールドがあります。

$__interval_ms

この変数はミリ秒単位の $__interval 変数であり、時間間隔形式の文字列ではありません。例えば、$__interval が 20m の場合、$__interval_ms は 1200000 です。

$__name

この変数は Singlestat パネルでのみ使用でき、[オプション] タブのプレフィックスまたはサフィックスフィールドで使用できます。変数は系列名またはエイリアスに置き換えられます。

$__org

この変数は、現在の組織 ID です。変数 ${__org.name} は、現在の組織名称です。

$__user

変数 ${__user.id} は現在のユーザー ID です。変数 ${__user.login} は、現在のユーザーのログインハンドルです。変数 ${__user.email} は、現在のユーザーの E メールアドレスです。

$__range

この変数は現在、Prometheus データソースでのみサポートされています。この変数は、現在のダッシュボードの範囲を表します。to - from で計算されます。ミリ秒は $__range_ms、秒は $__range_s で表示されます。

$timeFilter または $__timeFilter

$timeFilter 変数は、現在選択されている時間範囲を式として返します。例えば、時間間隔「Last 7 days」の式は time > now() - 7d です。

この変数は、次のようないくつかの場所で使用されます。

- InfluxDB データソースの WHERE 句。クエリエディタ では、Grafana は InfluxDB クエリに自動的に追加します。テキストエディタモードでは、`WHERE $timeFilter` を手動で追加できます。
- Log Analytics は、Azure Monitor データソースでクエリを実行します。
- SQL は、MySQL 、Postgres、MSSQL でクエリを実行します。
- `$__timeFilter` 変数は MySQL データソースで使用されます。

その他の変数オプション

このセクションでは、他の使用可能な変数オプションについて説明します。

変数選択オプションの入力

選択オプションを使用すると可変オプションの選択を管理できます。すべての選択オプションは任意で使用でき、デフォルトではオフになっています。

複数値

このオプションをオンにすると、変数のドロップダウンリストで複数のオプションを同時に選択することができます。詳細については、「複数値の変数のフォーマット」を参照してください。

すべてを含めるオプション

Grafana ワークスペースは、変数ドロップダウンリストに `All` オプションを追加します。エンドユーザーがこのオプションを選択すると、すべての変数オプションが選択されます。

すべての値のカスタム化

このオプションは、[すべてのオプションを含める] が選択されている場合にのみ表示されます。

`All` オプションの値を定義するには、[すべての値のカスタム化] フィールドに regex、glob、または Lucene 構文を入力します。

デフォルトでは、`All` 値には結合式内のすべてのオプションが含まれます。これは非常に長くなり、パフォーマンスの問題が発生する可能性があります。ワイルドカード正規表現など、すべての値のカスタム化を指定する方がよい場合があります。

すべての値のカスタム化オプションでカスタム正規表現、glob、または Lucene 構文を使用する場合はエスケープされないため、データソースに有効な値は何かを考慮する必要があります。

高度な変数フォーマットオプション

変数補間のフォーマットはデータソースによって異なりますが、デフォルトの形式を変更する場合があります。

例えば、MySQL データソースのデフォルトは、'server01','server02' などのように、複数の値を引用符でカンマ区切りで結合することです。場合によっては、server01,server02 など、引用符なしでカンマ区切りの文字列を指定することもできます。これを実行するには、次の高度な可変フォーマットオプションを使用します。

一般的な構文

構文: ${var_name:option}

無効なフォーマットオプションが指定されている場合、glob はデフォルト、またはフォールバックのオプションです。

CSV

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:csv}'
Interpolation result: 'test1,test2'
```

分散 - OpenTSDB

OpenTSDB のカスタム形式の複数の値を持つ変数をフォーマットします。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:distributed}'
Interpolation result: 'test1,servers=test2'
```

二重引用符

単一値変数と複数値の変数をカンマ区切りの文字列にフォーマットして、\" によって各値の " を工スケープし、" で引用符で囲みます。

```
servers = ['test1', 'test2']
```

```
String to interpolate: '${servers:doublequote}'
Interpolation result: '"test1","test2"'
```

Glob - Graphite

複数の値を持つ変数を glob にフォーマットします (Graphite クエリの場合)。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:glob}'
Interpolation result: '{test1,test2}'
```

JSON

複数の値をカンマ区切り文字列として変数をフォーマットします。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:json}'
Interpolation result: '["test1", "test2"]'
```

Lucene - OpenSearch

OpenSearch の Lucene 形式の複数の値を持つ変数をフォーマットします。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:lucene}'
Interpolation result: '("test1" OR "test2")'
```

Percentencode

URL パラメータで使用する単一値の変数と複数値の変数をフォーマットします。

```
servers = ['foo()bar BAZ', 'test2']
String to interpolate: '${servers:percentencode}'
Interpolation result: 'foo%28%29bar%20BAZ%2Ctest2'
```

パイプ

複数の値を持つ変数をパイプ区切り文字列にフォーマットします。

```
servers = ['test1.', 'test2']
String to interpolate: '${servers:pipe}'
Interpolation result: 'test1.|test2'
```

Raw

SQL クエリにおける単一引用符など、データソース固有のフォーマットをオフにします。

```
servers = ['test1.', 'test2']
String to interpolate: '${var_name:raw}'
Interpolation result: '{test.1,test2}'
```

[正規表現]

複数の値を持つ変数を正規表現文字列にフォーマットします。

```
servers = ['test1.', 'test2']
String to interpolate: '${servers:regex}'
Interpolation result: '(test1\.|test2)'
```

一重引用符

単一値の変数と複数値の変数をカンマ区切り文字列にフォーマットし、\' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = ['test1', 'test2']
String to interpolate: '${servers:singlequote}'
Interpolation result: "'test1','test2'"
```

Sqlstring

単一値の変数と複数値の変数をカンマ区切り文字列にフォーマットし、'' によって各値の ' をエスケープし、' で各値を引用符で囲みます。

```
servers = ["test'1", "test2"]
String to interpolate: '${servers:sqlstring}'
Interpolation result: "'test''1','test2'"
```

テキスト

単一値の変数と複数値の変数をテキスト表現にフォーマットします。単一値の変数の場合、テキスト表現のみを返します。複数値の変数の場合、テキスト表現は + と組み合わせて返されます。

```
servers = ["test1", "test2"]
String to interpolate: '${servers:text}'
Interpolation result: "test1 + test2"
```

複数値の変数のフォーマット

複数値が選択された変数を補間するのは、変数が使用されている特定のコンテキストで有効な文字列に複数の値をフォーマットする方法が単純ではないため、難しくなります。Grafana は、各データソースプラグインがテンプレート補間エンジンに、複数値に使用する形式を通知できるようにすることで、この問題を解決しようと試みます。

> **Note**
>
> Grafana がすべての値を 1 つの文字列にフォーマットするには、 変数の [すべての値をカスタム化] オプションを空白にする必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (一緒に加える) します。`value1,value2,value3` のようになります。カスタム `all` 値を使用する場合、代わりに `*` または `all` のような値になります。

Graphite データソースを使用した複数値変数

Graphite は glob 式を使用します。この場合、複数の値を持つ変数は、現在の変数値が host1、host2、host3 であった場合は、`{host1,host2,host3}` として補間されます。

Prometheus または InfluxDB データソースを使用した複数値変数

InfluxDB と Prometheus は正規表現を使用するため、同じ変数は `(host1|host2|host3)` として補間されます。すべての値も正規表現エスケープされます。そうでない場合、正規表現コントロール文字を持つ値は正規表現を破ります。

Elastic データソースを使用した複数値変数

Amazon OpenSearch は Lucene クエリ構文を使用するため、同じ変数は (`"host1" OR "host2" OR "host3"`) としてフォーマットされます。この場合、すべての値が エスケープされ、値には Lucene 制御語と引用符のみが含まれます。

フォーマットのトラブルシューティング

自動エスケープとフォーマットは問題を引き起こす場合があります。特に InfluxDB と Prometheus では正規表現構文を使用するため、変数を正規表現演算子コンテキストで使用する必要があり、問題の背後にあるロジックの把握が難しい場合があります。

Grafana でこの自動正規表現エスケープとフォーマットを実行したくない場合は、次のいずれかを実行する必要があります。

- [複数値] [すべてのオプションを含める] オプションをオフにします。
- [raw variable format]({{< relref "advanced-variable-format-options.md#raw" >}}) を使用します。

正規表現での変数のフィルタリング

正規表現クエリオプションを使用すると、変数クエリが返すオプションのリストをフィルタリング、または返すオプションを変更できます。

このセクションでは、正規表現を使用して変数ドロップダウンリスト内の値をフィルタリングおよび変更する方法を示します。

[正規表現クエリ] オプションを使用して、変数クエリによって返されるオプションの一覧をフィルタリングするか、返されるオプションを変更します。詳細については、「正規表現」を参照してください。

次のオプションリストでのフィルタリングの例:

```
backend_01
backend_02
backend_03
backend_04
```

`01` または **`02`** で終わるオプションのみが返されるようにフィルタリング

[Regex] (正規表現)

```
/.*[01|02]/
```

結果:

```
backend_01
backend_02
```

正規表現キャプチャグループを使用してテキストの一部を返すオプションのフィルタリングおよび変更

正規表現:

```
/.*(01|02)/
```

結果:

```
01
02
```

フィルタリングと変更 - Prometheus 例

オプションのリスト:

```
up{instance="demo.robustperception.io:9090",job="prometheus"} 1 1521630638000
up{instance="demo.robustperception.io:9093",job="alertmanager"} 1 1521630638000
up{instance="demo.robustperception.io:9100",job="node"} 1 1521630638000
```

正規表現:

```
/.*instance="([^"]*).*/
```

結果:

```
demo.robustperception.io:9090
demo.robustperception.io:9093
demo.robustperception.io:9100
```

名前付きテキストおよび値のキャプチャグループを使用したフィルタリングと変更

名前付きキャプチャグループを使用すると、変数クエリが返すオプションから「テキスト」と「値」の部分を別々にキャプチャできます。変数ドロップダウンリストには、選択できる各値に対するわかりやすい名前を含めることができます。

例えば、node_hwmon_chip_names Prometheus メトリクスをクエリする場合、chip_name は chip 値よりもわかりやすくなります。次の変数クエリ結果で開始します。

```
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_0",chip_name="enp216s0f0np0"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_1",chip_name="enp216s0f0np1"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_2",chip_name="enp216s0f0np2"} 1
node_hwmon_chip_names{chip="0000:d7:00_0_0000:d8:00_3",chip_name="enp216s0f0np3"} 1
```

これを次の正規表現に渡します。

```
/chip_name="(?<text>[^"]+)|chip="(?<value>[^"]+)/g
```

次のドロップダウンリストが生成されます。

```
Display Name          Value
------------          -------------------------
enp216s0f0np0         0000:d7:00_0_0000:d8:00_0
enp216s0f0np1         0000:d7:00_0_0000:d8:00_1
enp216s0f0np2         0000:d7:00_0_0000:d8:00_2
enp216s0f0np3         0000:d7:00_0_0000:d8:00_3
```

注: text および value のキャプチャグループ名のみがサポートされています。

パネルまたは行の繰り返し

テンプレート変数を使用すると動的ダッシュボードを作成できます。クエリ内のすべての変数は、クエリがデータベースに送信される前に変数の現在の値に拡張されます。変数を使用すると、すべてのサービスで 1 つのダッシュボードを再利用できます。

ダッシュボード全体でクエリを動的に変更する際は、テンプレート変数が非常に役立ちます。選択した値に基づいて Grafana で新しいパネルまたは行を動的に作成させる場合は、繰り返し機能を使用できます。

パネルの繰り返し

`Multi-value` または `Include all value` オプションがオンの変数がある場合は、1 つのパネルを選択して、Grafana に選択した値ごとにそのパネルを繰り返させることができます。繰り返し機能はパネル編集モードの全般タブにあります。

`direction` はパネルの配置方法を制御します。

`horizontal` を選択するとパネルは並んで配置されます。Grafana は繰り返しパネルの幅を自動的に調整して行全体がいっぱいになるようにします。現在、行上の他のパネルと繰り返しパネルを混在して使うことはできません。

`Max per row` を設定して、行あたりの最大パネル数を Grafana に指示します。デフォルトは 4 です。

`vertical` を選択すると、パネルは列の上から下に配置されます。繰り返しパネルの幅は、繰り返される最初の繰り返しパネル (元のテンプレート) の幅と同じです。

最初のパネル (元のテンプレート) のみ変更します。すべてのパネルに変更が適用されるためには、動的ダッシュボードの再構築を開始する必要があります。このためには変数値を変更 (つまり、繰り返しの基礎)、またはダッシュボードを再ロードします。

> **Note**
>
> パネルを繰り返すには、変数に対して 1 つ以上の項目を選択する必要があります。非表示にするためにパネルを 0 回繰り返すことはできません。

行の繰り返し

パネルに関して前述したように、変数に `Multi-value` または `Include all value` 選択オプションを設定している場合は、行を繰り返すこともできます。

この機能を有効にするには、まず [パネルの追加] メニューを使用して新しい行を追加する必要があります。次に、行タイトルの上にカーソルを置いて歯車ボタンを選択し、`Row Options` 設定パネルにアクセスします。行を繰り返す変数を選択できるようになります。

ベストプラクティスは行タイトルにも変数を使用することです。

Grafana アラート

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana アラートからは、システムの問題が発生した直後に、サービスの中断を最小限に抑えるための強力で実用的なアラートが提供されます。

Amazon Managed Grafana には最新のアラートシステム [Grafana アラート]が含まれており、アラート情報を一元的に管理、検索できるビューが用意されています。以下に、その機能の一部を紹介します。

- Grafana アラートを一元的に作成および管理できるビュー。
- Cortex と Loki 管理のアラートを作成および管理できる単一のインターフェイス。詳細については、「アラートルールの管理」を参照してください。
- Prometheus、Amazon Managed Service for Prometheus、およびその他のアラートマネージャー互換データソースからのアラート情報の表示。
- 複数のアラートインスタンスを 1 つのアラートルールから作成。詳細については、「1 次元ルールと多次元ルール」を参照してください。
- アラートリソースの管理が可能な terraform またはプロビジョニング API。詳細については、「Grafana アラートリソースのプロビジョニング」を参照してください。

既存の Amazon Managed Grafana ワークスペースの場合、デフォルトでは 従来のダッシュボードアラート が使用されます。Grafana アラートに移行するには、Grafana アラートに移行する必要があります。

Grafana アラートの詳細については、「Grafana アラートの新機能」を参照してください。

Grafana アラートには 4 つの主要な構成要素があります。

- アラートルール - アラートを発行するかどうかを判断する評価基準。1 つ以上のクエリと式、条件、評価の頻度、およびオプションとしてその条件の持続時間で構成されます。
- コンタクトポイント - アラートルールの条件が満たされたときに通知の送信先となるチャネル。
- 通知ポリシー - 通知の頻度を決定するために使用される一連の一致基準とグループ化基準。
- サイレンス - 通知のサイレンスに使用される日付と一致する基準。

Grafana アラートを有効にすると、次のことができます。

- Grafana マネージドアラートルールを作成する
- Cortex または Loki管理のアラートルールを作成する
- 既存のアラートルールを表示し、現在の状態を管理する
- アラートルールの状態と正常性を表示する
- アラートのコンタクトポイント (通知先) を追加または編集する
- 通知ポリシーを追加または編集する
- サイレンスを追加または編集する

制限

- Grafana アラートシステムは、利用可能なすべての Amazon Managed Service for Prometheus、Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからはルールを取得できない場合があります。
- Prometheus ではなく Grafana で定義されたアラートルールは、複数の通知をコンタクトポイントに送信します。ネイティブ Grafana アラートを使用している場合は、新しい Grafana アラート機能を有効にせず、従来のダッシュボードアラートのまま使用することをお勧めします。Prometheus データソースで定義されたアラートを表示する場合は、Grafana アラートを有効にすることをお勧めします。これにより、Prometheus アラートマネージャーで作成されたアラートの通知が 1 つだけ送信されます。

> **Note**
>
> この制限は、Grafana v10.4 以降をサポートする Amazon Managed Grafana ワークスペースの制限ではなくなりました。

トピック

- Grafana アラートの新機能
- 従来のダッシュボードアラートを Grafana アラートに移行する
- アラートの基本
- Grafana アラートルールの作成と管理
- アラートグループ
- Prometheus データソースのアラート通知のサイレンス化
- コンタクトポイント (通知先) の使用
- メッセージテンプレートの使用
- 通知ポリシーの使用

Grafana アラートの新機能

このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

Grafana アラートは、従来のダッシュボードアラートよりもいくつかの機能強化が行われています。

多次元アラートの作成

1 つのアラートルールでシステム全体の可視性を提供できるようになりました。このルールは 1 つのアラートルールから複数のアラートインスタンスが生成されます。例えば、1 つのルールで 1 台のホスト上に存在する複数のマウントポイントのディスク使用量をモニタリングするルールを作成できます。評価エンジンは 1 つのクエリから複数の時系列を返し、各時系列データはラベルセットで識別されます。

Note

各アラートインスタンスは、アラートのクォータ (上限数) にカウントされます。アラートの上限数を超えるインスタンスを作成する多次元ルールは評価されず、クォータエラーが返されます。詳細については、「クォータ到達エラー」を参照してください。

ダッシュボード外のアラートの作成

従来のダッシュボードアラートとは異なり、Grafanaのアラート機能では複数のデータソースからのデータを独自に組み合わせたクエリや式を作成できます。また、ID を使用してダッシュボードとパネルをアラートルールにリンクし、観察中のシステムのトラブルシューティングをすばやく行うことができます。

統合アラートはパネルクエリに直接関連付けられなくなるため、通知メールに画像やクエリ値は含まれませんが、カスタマイズされた通知テンプレートを使用して、クエリ値を表示できます。

Loki および Cortex のアラートルールの作成

Grafana アラートでは、Grafana 管理のアラートと同じ UI と API を使用して Loki および Cortex のアラートルールを管理することができます。

Amazon Managed Service for Prometheus およびその他の Prometheus 互換データソースからのアラートの表示と検索

Amazon Managed Service for Prometheus および Prometheus 互換データソースのアラートがアラートインターフェイスに一覧表示されるようになりました。これにより、複数のデータソースでラベルを検索して、関連するアラートをすばやく見つけることができます。

特別な (NoData、Error アラート状態) アラート

Grafana アラートは、アラートルールの評価で状態が `NoData` または `Error` のものを生成した場合、次のラベルを持つ特別なアラートを生成します。

- `alertname` (状態によって `DatasourceNoData` または `DatasourceError` の値を持ちます)。
- `rulename` (特別なアラートが属するアラートルールの名前を持ちます)。
- `datasource_uid` (状態の原因となったデータソースの UID を持ちます)。
- 元のルールのすべてのラベルと注釈。

これらのアラートは、例えば、サイレンスを追加したり、コンタクトポイントにルーティングしたりするなど、通常のアラートと同じように処理できます。

> **Note**
>
> ルールが複数のデータソースを使用し、そのうち 1 つ以上 がデータを返さない場合、アラート状態の原因となったデータソースごとに特別なアラートが作成されます。

従来のダッシュボードアラートを Grafana アラートに移行する

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

既存のワークスペースや、Grafana アラートを使用しないワークスペースには、従来のダッシュボードアラートを利用します。新しい Grafana アラートに移行するには、この機能を利用することに同意する必要があります。

、、 AWS CLIまたは Amazon Managed Grafana API を使用して AWS Management Console、Grafana アラートを使用するように Amazon Managed Grafana インスタンスを設定できます。Grafana アラートのオン/オフの切り替えなど、Amazon Managed Grafana の設定方法の詳細については、「Amazon Managed Grafana ワークスペースを設定する」を参照してください。

Note

Grafana アラートを使用する場合、Grafana で定義されたアラートルールは、Prometheusではなく、複数の通知をコンタクトポイント (通知先) に送信します。ネイティブ Grafana アラートを使用している場合は、新しい Grafana アラート機能を有効にせず、従来のダッシュボードアラートのまま使用することをお勧めします。Prometheus データソースで定義されたアラートを表示する場合は、Grafana アラートを有効にすることをお勧めします。これにより、Prometheus アラートマネージャーで作成されたアラートの通知が 1 つだけ送信されます。
この制限は、Grafana v10.4 以降をサポートする Amazon Managed Grafana ワークスペースで削除されます。

Grafana アラートシステムへの移行

Grafana アラートを有効にすると、既存の従来のダッシュボードアラートは Grafana アラートと互換性のある形式で移行されます。移行後のアラートと新しく作成したアラートは、Grafana インス

タンスのアラートページに表示されます。Grafana アラートを使用した場合、Grafana が管理するアラートルールは、一致するときに 1 つのアラートではなく、複数の通知を送信します。

従来のダッシュボードアラートと Grafana アラートへの読み取りおよび書き込みアクセス権は、それらが保存されているフォルダのアクセス権限で制御されます。移行中、従来のダッシュボードアラートアクセス許可は、次のように新しいルールアクセス許可と照合されます。

- 元のアラートが属するダッシュボードにアクセス権限が設定されている場合、移行時に元のダッシュボードの権限 (フォルダから継承された権限を含む) 設定で `Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1}` という形式の名前のフォルダが作成されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードがフォルダ内にある場合、ルールはこのフォルダにリンクされ、そのアクセス権限が継承されます。
- ダッシュボードにアクセス権限が設定されておらず、ダッシュボードが General (全般) フォルダにある場合、ルールは General (全般) アラートフォルダにリンクされ、ルールにはデフォルトのアクセス権限が継承されます。

> **Note**
>
> Grafana のアラート設定には `NoData` に対する `Keep Last State` オプションがないため、このオプションは従来のルールを移行する際に `NoData` になります。`Error` 処理のためのオプション `Keep Last State` は、新しいオプション `Error` に移行されます。`Keep Last State` の動作に合わせるため、移行時には各アラートルールに対して 1 年間のサイレンスが Amazon Managed Grafana によって自動的に作成されます。

通知チャネルは、適切なルートと受信者が設定されたアラートマネージャー設定に移行されます。デフォルトの通知チャネルは、コンタクトポイントとしてデフォルトのルートに追加されますが、ダッシュボードアラートに関連付けられていない通知チャネルは `autogen-unlinked-channel-recv` ルートに送られます。

制限

- Grafana アラートシステムは、利用可能なすべての Prometheus、Loki、およびアラートマネージャーデータソースからルールを取得できますが、他のサポートされているデータソースからアラートルールを取得できない場合があります。
- Grafana アラートと従来のダッシュボードアラートを交互に移行すると、どちらか一方でのみサポートされる機能のデータが失われる可能性があります。

> **Note**
>
> 従来のダッシュボードアラートに戻した場合、Grafana アラートが有効になっている間に新たに作成されたアラートルールやアラート設定に加えられたすべての変更は失われます。

アラートの基本

> このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

このセクションでは、Grafana アラートの基本概念について説明します。

アラートの概念

以下の表では、Grafana アラート機能の主要な概念について説明しています。

主要な概念または機能	定義
アラートに使用するデータソース	メトリクス、ログ、トレースをクエリおよび視覚化するデータソースを選択します。
スケジューラー	アラートルールを評価するコンポーネントで、データソースに対して定期的にクエリを実行します。これは Grafana が管理するルールにのみ適用されます。
アラートマネージャー	アラートインスタンスのルーティングとグループ化を管理します。

主要な概念または機能	定義
アラートルール	アラートルールを発動させる評価基準のセット。アラートルールは、1 つ以上のクエリと式、条件、評価の頻度、その条件の持続時間で構成されます。アラートルールは、複数のアラートインスタンスを生成できます。
アラートインスタンス	アラートインスタンスは、アラートルールのインスタンスです。1 次元アラートルールには 1 つのアラートインスタンスがあります。多次元アラートルールには 1 つ以上のアラートインスタンスがあります。複数の結果に一致する単一のアラートルール、たとえば 10 台の仮想マシンに対する CPU の監視は、複数 (この場合は 10) のアラートインスタンスとしてカウントされます。この数は時間の経過とともに変動する場合があります。たとえば、システム内のすべての仮想マシンの CPU 使用率を監視するアラートルールは、仮想マシンを追加するとアラートインスタンスが増加します。アラートインスタンスのクォータの詳細については、クォータ到達エラー を参照してください。
アラートグループ	アラートマネージャーは、ルート通知ポリシーのラベルを使用して、デフォルトでアラートインスタンスをグループ化します。これにより、コンタクトポイントに送信される重複排除とアラートインスタンスのグループが制御されます。
コンタクトポイント	アラートルール発生時の通知先への通知方法を定義します。
メッセージテンプレート	コンタクトポイント使用する再利用可能なカスタムテンプレートを作成します。

主要な概念または機能	定義
通知ポリシー	アラートがどこで、いつ、どのようにグループ化され、コンタクトポイントにルーティングされるかを定義するルールのセット。
ラベルとラベル照合機能	ラベルはアラートルールを識別するための一意な識別子です。ラベルによってアラートルールが通知ポリシーとサイレンスに紐づけられ、どのポリシーで処理されるか、どのアラートルールをサイレンス対象にするかを決定します。
サイレンス	1 つ以上のアラートインスタンスからの通知を停止します。サイレンスとミュートタイミングの違いは、サイレンスは指定した時間帯の間だけ有効であるのに対し、ミュートタイミングは繰り返しのスケジュールに基づいて発生する点です。アラートインスタンスをサイレンス対象とするにはラベル照合機能を使用します。
ミュートタイミング	通知を新たに生成または送信しない時間間隔を指定します。メンテナンス期間など、繰り返し発生する期間中にアラート通知を停止することができます。既存の通知ポリシーにリンクする必要があります。

アラートのデータソース

Grafana 管理のアラートは、アラートが有効になっている以下のバックエンドデータソースをクエリします。

- ビルトイン、または Grafana によって開発、維持管理されているデータソース: `Alertmanager`、`Graphite`、`Prometheus` (Amazon Managed Service for Prometheus を含む)、`Loki`、`InfluxDB`、`Amazon OpenSearch Service`、`Google Cloud Monitoring`、`Amazon CloudWatch`、`Azure Monitor`、`MySQL`、`PostgreSQL`、`MSSQL`、`OpenTSDB`、`Oracle`、`Azure Monitor`。

数値データのアラート

時系列形式でない数値データは、サーバー側の式に渡したり、直接アラートすることができます。これにより、データソース内の処理と効率が向上し、アラートルールを簡素化することができます。時系列データではなく数値データを基にアラートを送信する場合、ラベル付きの各時系列を 1 つの数値に集約する必要はありません。代わりに、ラベル付き番号が Grafana に返されます。

表形式のデータ

この機能は、MySQL 、Postgres、MSSQL、Oracle などの SQL データソースを含む表形式のデータをクエリするバックエンドデータソースでサポートされています。

Grafana 管理のアラートまたはサーバー側の式を含むクエリは、これらのデータソースでは数値として扱われます。

- データソースのクエリで `Format AS` オプションが `Table` に設定されている場合。
- クエリから Grafana に返されたレスポンステーブルの数値列 (int、double、float など) が 1 列だけの場合 (その他文字列が返されている場合も含みます)。

文字列がある場合、それらの列はラベルになります。列の名前はラベル名になり、各行の値は対応するラベルの値になります。複数の行が返された場合、各行はラベルによって一意に識別される必要があります。

例

例えば、次のような MySQL のテーブル「Diskspace」があるとします。

時間	ホスト	ディスク	PercentFree
2021 年 6 月 7 日	web1	/etc	3
2021 年 6 月 7 日	web2	/var	4
2021 年 6 月 7 日	web3	/var	8
…	…	…	…

データを時間でフィルタリングしてクエリを実行できますが、時間系列は Grafana に返す必要はありません。例えば、空き容量が 5% 未満の場合、ホスト、ディスクごとにアラートを発生させる場合は次のようになります。

```
SELECT Host, Disk, CASE WHEN PercentFree < 5.0 THEN PercentFree ELSE 0 END FROM (
  SELECT
      Host,
      Disk,
      Avg(PercentFree)
  FROM DiskSpace
  Group By
    Host,
    Disk
  Where __timeFilter(Time)
```

このクエリでは、次のようなレスポンステーブルが返されます。

ホスト	ディスク	PercentFree
web1	/etc	3
web2	/var	4
web3	/var	0

このクエリがアラートルールの[条件]として使用されると、値がゼロでない場合にアラートが発生します。その結果、次の表のように 3 つのアラートインスタンスが生成されます。

ラベル	ステータス
{Host=web1,disk=/etc}	[アラート]
{Host=web2,disk=/var}	[アラート]
{Host=web3,disk=/var}	普通

アラートマネージャー

Grafana にはビルトインで Prometheus アラートマネージャーのサポートが含まれています。アラートマネージャーはアラートルールをグループ化し管理する役割を果たし、アラートエンジンの上にオーケストレーション機能を追加します。デフォルトでは、Grafana 管理のアラートの通知は、Grafana の基本機能の一部として組み込まれているアラートマネージャーによって処理されます。アラートマネージャードロップダウンから Grafana オプションを選択することで、Grafana アラート UI からアラートマネージャーのコンタクトポイント、通知ポリシー、テンプレートを設定できます。

Grafana のアラート機能は、外部アラートマネージャー設定をサポートしています (外部データソースとしてのアラートマネージャーの詳細については、「アラートマネージャーデータソースに接続する」を参照してください)。外部アラートマネージャーを追加すると、アラートマネージャードロップダウンに使用可能な外部アラートマネージャーデータソースのリストが表示されます。データソースを選択して、スタンドアロンの Cortex または Loki データソースのアラートを作成および管理します。

アラートルールの状態と正常性

アラートルールの状態と正常性は、アラートに関するいくつかの主要なステータスインジケータを理解するのに役立ちます。主な構成内容は、「アラートの状態」、「アラートルール状態」、「アラートルールの正常性」の 3 つです。これらは関連していますが、それぞれ異なる情報を伝達します。

アラートルールの状態

- [普通] – 評価エンジンから返された時系列のいずれも `Pending` または `Firing` の状態でない。
- [保留中]– 評価エンジンから返された時系列の少なくとも 1 つに `Pending` がある。
- [発動中] — 評価エンジンから返された時系列の少なくとも 1 つに `Firing` がある。

アラートの状態

- [普通] – 評価エンジンから返されたすべての時系列に対して、アラートルールの条件が [偽] である。
- [アラート] – 評価エンジンから返された時系列のうち、1 つ以上のアラートルールの条件が [真] である。アラートが発生するために必要な、条件がである持続時間が設定されていて、その持続時間を経過した、または超えている。

- [保留中] – 評価エンジンから返された時系列のうち、1 つ以上のアラートルールの条件が [真] である。アラートが発生するために必要な、条件がである持続時間が設定されていて、その持続時間を経過していない。
- [NoData] – アラートルールが時系列を返していない、時系列のすべての値が null、または時系列のすべての値がゼロ。
- [エラー] - アラートルールを評価しようとしてエラーが発生した。

アラートルールの正常性

- OK – アラートルールの評価時にエラーがなかった。
- [エラー] – アラートルールの評価時にエラーがあった。
- [NoData] – ルール評価中に返された少なくとも 1 つの時系列にデータが存在しない。

Grafana アラートルールの作成と管理

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

アラートルールは、アラートを発行するかどうかを判断する評価基準のセットです。アラートルールは、1 つ以上のクエリと式、条件、評価の頻度、およびオプションとしてその条件の持続時間で構成されます。

評価するデータセットを「クエリと式」で選択し、「条件」にアラートがアラートを発行するための基準 (しきい値) を設定します。間隔は、アラートルールが評価される頻度を指定します。「持続時間」を設定した場合、その条件下にある状態の継続時間を示します。また、データがない場合のアラート動作も設定できます。

以下のセクションでは、さまざまな種類の Grafana アラートルールの作成と管理について説明します。

トピック

- Cortex または Loki 管理のアラートルールの作成
- Cortex または Loki 管理のレコードルールの作成
- Grafana 管理のアラートルールの作成
- アラートルールの注釈とラベル
- アラートルールの管理
- Cortex または Loki ルールグループと名前空間

Cortex または Loki 管理のアラートルールの作成

Grafana では、外部の Cortex または Loki インスタンスのアラートルールを作成することができます。

> **Note**
>
> Cortex は、Amazon Managed Service for Prometheus および Prometheus データソースで使用される時系列データベースです。

前提条件

- Prometheus データソースへの書き込みアクセス権限があることを確認します。アクセス権限がない場合、Cortex 管理のアラートルールを作成または更新することはできません。
- Cortex と Loki のデータソースでそれぞれのサービスを設定して Ruler API を有効にします。
  - Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集する場合、他のストレージタイプを設定してください。
  - Cortex – `/prometheus` ではなく、旧式の `/api/prom` プレフィックスを使用します。Prometheus データソースは Cortex と Prometheus の両方に対応しており、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを期待しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[アラート UI 経由でアラートを管理する] チェックボックスをオフにします。

Cortex または Loki 管理のアラートルールの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [新しいアラートルール] を選択します。
3. ステップ 1 で、以下を参照してルール名、タイプ、ストレージの場所を追加します。

   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [ルールタイプ] ドロップダウンから、[Cortex/Loki 管理アラート]を選択します。
   - [データソース]の選択ドロップダウンから、Prometheus または Loki データソースを選択します。
   - [名前空間] ドロップダウンから、既存のルール名前空間を選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。名前空間には 1 つ以上のルールグループを含めることができます。これは、組織的な目的を持たせるためのみに使用します。詳細については、「Cortex または Loki ルールグループと名前空間」を参照してください。
   - [グループ] ドロップダウンから、選択した名前空間内の既存のグループを選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。新しく作成されたルールは、グループの末尾に追加されます。グループ内のルールは、同じ評価時間で一定の間隔で順番に実行されます。
4. ステップ 2 で、評価するクエリを追加します。

   値は PromQL または LogQL 式で指定することができます。評価結果に 0 より大きい値を持つシリーズが 1 つ以上ある場合、ルールはアラートを発行します。アラートはシリーズごとに作成されます。
5. ステップ 3 で、条件を追加します。

   条件の [For] テキストボックスに、アラートが発行されるために必要な条件の継続時間を指定します。例えば `5m` を指定した場合、条件が 5 分間継続した場合にアラートが発行されます。

> **Note**
>
> 条件が満たされると、アラートは `Pending` 状態に入り、そのまま指定された期間条件が継続するとアラートは `Firing` 状態に移行します。条件から外れた場合、`Normal` 状態に戻ります。

6. ステップ 4 で、ルールに関連するメタデータを追加します。
   - アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「アラートルールの注釈とラベル」を参照してください。
   - Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
   - カスタムラベルを追加します。
7. [アラートのプレビュー] を選択してルールを評価し、生成されるアラートを確認します。各アラートの状態と値を含むアラートのリストが表示されます。
8. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

Cortex または Loki 管理のレコードルールの作成

外部 Cortex または Loki インスタンスのレコードルールを作成および管理することができます。記録ルールは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列セットとして保存します。この新しい時系列のクエリは、特にダッシュボードでは、ダッシュボードが更新されるたびに同じ式をクエリするため、より高速になります。

前提条件

Cortex と Loki のデータソースでそれぞれのサービスを設定して Ruler API を有効にします。

- Loki - Loki データソースのデフォルトである `local` ルールストレージタイプは、ルールの表示のみをサポートしています。ルールを編集する場合、他のストレージタイプを設定してください。
- Cortex – Cortex を指すように Grafana Prometheus データソースを設定するときは、`/prometheus` ではなく旧式の `/api/prom` プレフィックスを使用します。Prometheus データソースは Cortex と Prometheus の両方に対応しており、Grafana はクエリ API と Ruler API の両方が同じ URL にあることを期待しています。Ruler API に別の URL を指定することはできません。

> **Note**
>
> 特定の Loki または Prometheus データソースのアラートルールを管理しない場合は、その設定に移動し、[アラート UI 経由でアラートを管理する] チェックボックスをオフにします。

Cortex または Loki 管理のレコードルールの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [新しいアラートルール] を選択します。
3. ステップ 1 で、以下を参照してルール名、タイプ、ストレージの場所を追加します。

   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [ルールタイプ] ドロップダウンから、[Cortex/Loki 管理アラート]を選択します。
   - [データソース]の選択ドロップダウンから、Prometheus または Loki データソースを選択します。
   - [名前空間] ドロップダウンから、既存のルール名前空間を選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。名前空間には 1 つ以上のルールグループを含めることができます。これは、組織的な目的を持たせるためのみに使用します。詳細については、「Cortex または Loki ルールグループと名前空間」を参照してください。
   - [グループ] ドロップダウンから、選択した名前空間内の既存のグループを選択します。それ以外の場合は、[新規追加] を選択し、名前を入力します。新しく作成されたルールは、グループの末尾に追加されます。グループ内のルールは、同じ評価時間で一定の間隔で順番に実行されます。
4. ステップ 2 で、評価するクエリを追加します。

   値は PromQL または LogQL 式で指定することができます。評価結果に 0 より大きい値を持つシリーズが 1 つ以上ある場合、ルールはアラートを発行します。アラートはシリーズごとに作成されます。
5. ステップ 3 で、ルールに関連するメタデータを追加します。

   - アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「アラートルールの注釈とラベル」を参照してください。
   - Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
   - カスタムラベルを追加します。
6. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

Grafana 管理のアラートルールの作成

Grafana では、1 つ以上のデータソースをクエリして結果を集約または変換して、それらを相互に比較したり、固定のしきい値と比較したりするアラートルールを作成することができます。これらが処理された場合、Grafana はコンタクトポイント (通知先) に通知を送信します。

> **Note**
>
> Grafana アラートの使用中に Grafana 管理のアラートルールを作成すると、ルールが一致した際に複数の通知が送信されます。一部のコンタクトポイントとなるプロバイダーには、重複した通知を避けるための設定可能なオプションがある場合があります。

Grafana 管理のルールの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開いて既存のアラートを一覧表示します。
2. [新しいアラートルール] を選択します。
3. ステップ 1 で、以下を参照してルール名、タイプ、ストレージの場所を追加します。
   - [ルール名] にわかりやすい名前を追加します。この名前はアラートルールリストに表示されます。またこの名前は、このルールから作成されるすべてのアラートインスタンスの `alertname` ラベルにもなります。
   - [ルールタイプ] ドロップダウンから、[Grafana 管理のアラート] を選択します。
   - [フォルダ] ドロップダウンから、ルールの保存先となるフォルダを選択します。フォルダを選択しない場合、ルールは `General` フォルダに保存されます。フォルダを作成する場合、ドロップダウンを選択し、新しいフォルダ名を入力します。
4. ステップ 2 で、評価するクエリと式を追加します。
   - 名前はデフォルトのままでも構いませんが、変更する場合、カーソルを合わせて編集アイコンを選択します。
   - クエリの場合、ドロップダウンからデータソースを選択します。
   - 1 つ以上のクエリまたは式を追加します (式の詳細については、Grafana ドキュメントの「式」を参照してください)。
   - 式ごとに [クラシック条件]を選択して単一のアラートルールを作成するか、[Math]、[削減]、[リサンプリング] オプションから選択して、各シリーズに個別のアラートを作成しま

す。これらのオプションの詳細については、「1 次元ルールと多次元ルール」を参照してください。

- [Run queries] (クエリの実行) を選択して、クエリが正しく動作することを確認します。

5. ステップ 3 で、条件を追加します。

- [条件] ドロップダウンから、アラートルールを発動させるクエリまたは式を選択します。
- Evaluate every (~ごとに評価する) には、評価の頻度を指定します。10 秒の倍数である必要があります。例えば、`1m` や `30s` などです。
- [評価対象] には、アラートが発行されるために必要な条件の継続時間を指定します。

> **Note**
>
> 条件が違反すると、アラートは `Pending` 状態に入り、そのまま指定された期間条件の違反が継続するとアラートは `Firing` 状態に移行します。条件から外れた場合、`Normal` 状態に戻ります。

- [Configure no data and error handling] (データがない場合の設定とエラー処理) で、データがない場合のアラート動作を設定します。「データがない場合やエラーが発生した場合の対処」のガイドラインを参照してください。
- [アラートのプレビュー] を選択してルールを評価し、生成されるアラートを確認します。プレビューでは、データとエラー処理条件は除外されます。

6. ステップ 4 で、ルールに関連するメタデータを追加します。

- アラートメッセージのカスタマイズ時に使用する説明と概要を追加します。ガイドライン「アラートルールの注釈とラベル」を参照してください。
- Runbook URL、パネル、ダッシュボード、アラート ID を追加します。
- カスタムラベルを追加します。

7. [保存] を選択してルールを保存するか、[保存して終了] を選択してルールを保存し、[アラート] ページに戻ります。

1 次元ルールと多次元ルール

Grafana 管理のアラートルールでは、従来の条件を使用してルールを作成することも、多次元ルールを作成することもできます。

1 次元ルール (従来の条件)

従来の条件式を使用して、条件が満たされたときに単一のアラートを発行するルールを作成します。複数のシリーズを返すクエリの場合、Grafana は各シリーズのアラート状態を追跡しません。そのため、複数のシリーズでアラート条件が満たされた場合でも、Grafana が送信するアラートは 1 件のみです。

式のフォーマット方法の詳細については、Grafana ドキュメントの「式」を参照してください。

多次元ルール

クエリで返されるシリーズごとに個別のアラートインスタンスを生成するには、多次元ルールを作成します。

> **Note**
>
> 多次元ルールによって生成された各アラートインスタンスは、アラートの合計クォータにカウントされます。クォータに達すると、ルールは評価されなくなります。多次元ルールのクォータの詳細については、「クォータ到達エラー」を参照してください。

1 つのルールから複数のインスタンスを作成するには、`Math`、`Reduce`、または `Resample` 式を使用して多次元ルールを作成します。例えば、以下のことが可能です。

- 各クエリに `Reduce` 式を追加すると、選択した時間範囲の値が 1 つの値に集計されます。(数値データ を使用するルールには必要ありません)。
- ルールの条件を含む `Math` 式を追加します。もしクエリや reduce 式がすでにアラートを発行すべき場合は正の数、発行しない場合に 0 を返す場合、これは必要ありません。

  例:

  - `$B > 70` は、B クエリ/式の値が 70 を超える場合にアラートを発行します。
  - `$B < $C * 100` は、B の値が C の値に 100 を掛けた値より小さい場合にアラートを発行します。比較対象のクエリの結果に複数のシリーズがある場合、異なるクエリのシリーズが同じラベルを持っているか、一方が他方のサブセットである場合に一致します。

> **Note**
>
> Grafana は、テンプレート変数を使用したアラートクエリはサポートしていません。詳細については、コミュニティページ「アラートクエリではアラート設定時にテンプレート変数を使用できません」を参照してください。

多次元ルールのパフォーマンスに関する考慮事項

各アラートインスタンスは、アラートのクォータ (上限数) にカウントされます。アラートの上限数を超えるインスタンスを作成する多次元ルールは評価されず、クォータエラーが返されます。詳細については、「クォータ到達エラー」を参照してください。

多次元アラートは、Grafana ワークスペースや、Grafana がアラートルールを評価するためにクエリを実行するデータソースのパフォーマンスに大きな影響を与える可能性があります。監視システムのパフォーマンスを最適化したい場合、以下の考慮事項を参照してください。

- ルール評価の頻度 – アラートルールの [Evaluate Every] (~ ごとに評価する) プロパティでルール評価の頻度を制御します。評価の頻度を許容可能な最小頻度に収めることをお勧めします。
- 結果セットのカーディナリティ – ルールで作成されるアラートインスタンスの数は直接そのパフォーマンスに影響します。例えば、すべての仮想マシン上の各 API パスごとに API 応答エラーを監視しているとします。このセットのカーディナリティは「パスの数 × 仮想マシンの数」になります。例えば、仮想マシンあたりのパスごとではなく、仮想マシンあたりの合計エラー数を監視することで、結果セットのカーディナリティを減らすことができます。
- クエリの複雑さ – データソースが迅速に処理して応答できるクエリにするだけで、リソースの消費を大幅に抑えることができます。これは他の考慮事項よりは重要度が低いものの、他の点での最適化が十分に行われている場合、個別のクエリパフォーマンスの見直しが役立つ可能性があります。また、これらのルールの評価がデータソースに与える影響にも注意する必要があります。アラートクエリは、監視データベースで処理されるクエリの大部分を占めることが多いため、Grafana インスタンスにかかる負荷要因がデータソースにも同様に影響を与えます。

クォータ到達エラー

1 つのワークスペース内に保持できるアラートインスタンスの数にはクォータ (上限) があります。この上限に達した場合、そのワークスペースにこれ以上新たにアラートルールを作成できなくなります。多次元アラートでは、アラートインスタンスの数は時間の経過とともに変動する場合があります。

以下に、アラートインスタンスを使用する際に覚えておくべき重要な点について紹介します。

- 作成するのが 1 次元ルールのみの場合、ルールからはそれぞれ 1 つのアラートインスタンスが作成されます。1 つのワークスペースに作成できるルールの上限は、アラートインスタンスのクォータ数までとなります。
- 多次元ルールからは複数のアラートインスタンスが作成されますが、その数は評価されるまでわかりません。例えば、Amazon EC2 インスタンスの CPU 使用率を追跡するアラートルールを作成した場合、最初は 50 台の EC2 インスタンスが対象となり、50 個のアラートインスタンスが作成されますが、1 週間後にさらに 10 台の EC2 インスタンスが追加された場合、次の評価時には 60 個のアラートインスタンスが存在することになります。

  アラートインスタンスの数は、多次元アラートを作成するときに評価され、アラートインスタンスのクォータをすぐに超えるインスタンスを作成することはできません。アラートインスタンスの数は変動する可能性があるため、ルール評価の都度クォータがチェックされます。
- 評価時に、ルールがアラートインスタンスのクォータを超える場合、アラートルールが更新されアラートインスタンスの合計数がサービスクォータを下回るまで、そのルールは評価されません。この場合、クォータに達したことを通知するアラート通知を受け取ります (通知は評価対象のルールの通知ポリシーを基に送信されます)。通知には、`QuotaReachedError` 値を含む `Error` 注釈が含まれています。
- `QuotaReachedError` を引き起こすルールは、評価されなくなります。評価は、更新が行われ、更新後の評価が `QuotaReachedError` を引き起こさなくなった場合にのみ再開されます。評価されていないルールは、Grafana コンソールにクォータ到達エラーが表示されます。
- アラートインスタンスの数を減らすには、アラートルールを削除するか、複数次元のアラートを編集してアラートインスタンスを減らします (例えば、仮想マシン内の API ごとに 1 個のアラートではなく、仮想マシンごとの 1 件のエラーに対して 1 個のアラートを持つようにします)。
- 評価を再開するには、アラートを更新して保存します。アラートインスタンスの数を減らすように更新することもできますし、インスタンス数を減らすために他の変更を加えた場合、変更なしで保存しても構いません。再開可能な場合、評価が再開されます。さらに別の `QuotaReachedError` を引き起こす結果となった場合、保存することはできません。
- アラートが保存され、アラートクォータを超えることなく評価が再開された場合、Grafana コンソールにはしばらくの間クォータ到達エラーが (次の評価間隔まで) 表示され続けることがありますが、アラートルール評価が開始され、ルールのしきい値に達するとアラートが送信されます。
- アラートのクォータおよびその他のクォータの詳細については、「Amazon Managed Grafana のサービスクォータ」を参照してください。

データがない場合やエラーが発生した場合の対処

データがない場合やエラーが発生した場合のアラート動作の処理方法を選択します。

データがない場合に選択できる処理方法を、以下の表に照会します。

データがない場合の選択肢	行動
データなし	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を使用したアラート DatasourceNoData を作成します。
[アラート]	アラートルールの状態を Alerting に設定します。
OK	アラートルールの状態を Normal に設定します。

エラーが発生した場合に選択できる処理方法を、以下の表に照会します。

エラーまたはタイムアウト時の選択肢	行動
[アラート]	アラートルールの状態を Alerting に設定します
OK	アラートルールの状態を Normal に設定します
エラー	アラートルールの名前と UID、およびラベルとしてデータを返さないデータソースの UID を使用したアラート DatasourceError を作成します。

アラートルールの注釈とラベル

注釈とラベルは、アラートルール、データソースのレスポンス、およびアラートルール評価の結果として発生するアラートに関連付けられたキーと値のペアです。これらは、アラート通知の内容として

直接使用することも、テンプレートやテンプレート関数で使用して、通知内容を動的に作成することもできます。

注釈

注釈は、アラートに関する追加情報を提供するキーと値のペアです。description、summary、runbook_url、alertId、dashboardUid、panelId の注釈を使用できます。これらは UI のルールとアラートの詳細に表示され、コンタクトポイントのメッセージテンプレートで使用できます。

ラベル

ラベルは、アラートに関する情報を含むキーと値のペアです。アラートのラベルセットは、アラート評価および通知プロセス全体で生成および追加されます。これらは以下の方法で使用されます。

- アラートの完全なラベルセットを使用することで、Grafana Alerts 内でそのアラートが一意に識別されます。
- アラートマネージャーはラベルを使用して、[通知ポリシー]の[サイレンス]と[アラートグループ]のアラートを照合します。
- アラート UI には、そのルールの評価によって生成されたすべてのアラートインスタンスのラベルが表示されます。
- コンタクトポイントには、通知を生成する際にラベルにアクセスして、そのアラートに固有の情報を含めることができます。
- ラベルはアラートルール に追加できます。これらの手動で設定されたラベルは、テンプレート関数を使用したり、他のラベルを参照することができます。ここでアラートルールに追加されたラベルは、ラベル間で衝突が発生した場合に優先されます。

以下に、注釈とラベルを展開する際に使用できる変数を紹介します。

名前	説明
$labels	クエリまたは条件からのラベル。例えば、{{ $labels.instance }} と {{ $labels.job }} です。これは、ルールが従来の条件を使用している場合には使用できません。

名前	説明
`$values`	このアラートルールで評価されたすべての reduce と math 式の値。たとえば、`{{ $values.A }}` 、`{{ $values.A.Labels }}` 、`{{ $values.A.Value }}` などです。式の `refID` は、`A` にあたります。これは、ルールが従来の条件を使用している場合には使用できません。
`$value`	アラートインスタンスの値の文字列。例えば、`[ var='A' labels={instance=foo} value=10 ]` 。

アラートルールの管理

[アラート] ページには、アラートルールが一覧表示されます。デフォルトでは、ルールはデータソースの種類ごとにグループ化されます。Grafana セクションには Grafana によって管理されるルールが一覧表示され、Cortex/Loki セクションには Prometheus 互換データソースのルールが一覧表示されます。Prometheus 互換データソースのアラートルールは表示できますが、編集することはできません。

アラートルールの表示

Grafana アラートを使用すると、すべてのアラートを 1 ページに表示できます。

アラートの詳細を表示するには

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。デフォルトでは、ルールはデータソースの種類ごとにグループ化して表示されます。各アラートの現在の状態別に表示することもできます (詳細については後述します)。
2. [表示形式] では、グループビューと状態ビューを切り替えることができます。
3. 行の横にある矢印を選択すると、その行の詳細が表示されます。ルールの詳細には、ルールのラベル、注釈、データソース、クエリ、およびルールから生成されたアラートインスタンスのリストが含まれます。

グループビュー

グループビューには、フォルダ別にグループ化された Grafana アラートルールと、namespace + group 別にグループ化された Loki または Prometheus アラートルールが表示されます。これは、ルールの管理を目的としたデフォルトのルールリストビューです。各グループを展開すると、そのグループ内のルールのリストが表示されます。さらにルールを展開すると、その詳細を見ることができます。また、ルールから生成されたアクションボタンとアラートを展開して、詳細を表示することもできます。

状態ビュー

状態ビューには、アラートルールが状態ごとグループ化されて表示されます。このビューを使用すると、各ルールがどの状態にあるかを一目で確認できます。各ルールは展開して詳細を表示することができます。アクションボタンとこのルールによって生成されたアラート、および各アラートをさらに展開して詳細を表示することができます。

アラートルールのフィルター

[アラート] ページに表示されるアラートルールは、いくつかの方法でフィルタリングできます。

- 特定のデータソースを使用するルールだけを表示するには [データソースの選択] でフィルタリングするデータソースを選択します。
- また、[ラベルで検索] から検索条件を選択して、ラベルでフィルタリングすることもできます。例えば、`environment=production`、`region=~US|EU`、`severity!=warning` などの条件で検索できます。
- 特定の状態のルールだけを表示するには [状態でアラートをフィルタリング] で表示する状態を選択します。

アラートルールの編集または削除

Grafana管理のアラートルールの編集または削除は、ルールが保存されているフォルダに対する編集権限を持つユーザーのみが行うことができます。外部 Cortex または Loki インスタンスのアラートルールの編集または削除は、エディタまたは管理者ロールを持つユーザーが行うことができます。

ルールを編集または削除するには

1. 表示、編集、削除のルールコントロールが表示されるまでルールを展開します。
2. [編集] を選択してルールの作成ページを開きます。更新方法はルール作成方法と同じです。詳細については、「Grafana 管理のアラートルールの作成」または「Cortex または Loki 管理のアラートルールの作成」の手順を参照してください。
3. 削除する場合は [削除] を選択してルールを削除します。

Cortex または Loki ルールグループと名前空間

ルールは整理することができます。ルールはルールグループ内に作成され、ルールグループは名前空間に整理されます。ルールグループ内のルールは、一定の間隔で順次実行されます。デフォルトの間隔は 1 分です。また、Cortex または Loki の名前空間とルールグループの名前を変更したり、ルールグループの評価間隔を編集することができます。

ルールグループまたは名前空間の編集方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. 編集するルールグループまたは名前空間内のルールに移動します。
3. [編集] (ペン) アイコンを選択します。
4. ルールグループまたは名前空間に変更を加えます。

> **Note**
>
> 名前空間では、名前のみを編集できます。ルールグループの場合、名前、またはグループ内のルールの評価間隔を変更します。たとえば、ルールを 1 分ごとに評価する場合は `1m` を選択し、30 秒ごとに評価する場合は `30s` を選択します。

5. [変更の保存] をクリックします。

アラートグループ

> このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

アラートグループは、アラートマネージャーインスタンスからのグループ化されたアラートを表示します。デフォルトでは、アラートは 通知ポリシーの使用 のルートポリシーのラベルキーでグループ化されます。共通のアラートを 1 つのアラートグループにグループ化することで、重複したアラートが発生することを防ぎます。

アラートグループの表示方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択し、[アラートグループ化] 項目を選択すると、現在設定されているグループの一覧が表示されるページが開きます。
2. [アラートマネージャー] ドロップダウンから、データソースとして外部アラートマネージャーを選択します。デフォルトでは、Grafana アラートマネージャーが選択されています。
3. [カスタムグループ化対象] ドロップダウンから、グループ化を表示するラベルの組み合わせを選択します。このビューを使用して、通知ポリシーのグループ化をデバッグしたり検証することができます。

ルートポリシーのグループ化またはカスタムグループ化で指定されたラベルのないアラートは、No grouping のヘッダーのグループに追加されます。

アラートのフィルタリング

アラートは、次のフィルターを使用して、特定の条件に一致するものだけを表示することができます。

- [ラベルで検索] – [検索] に既存のラベルを入力すると、ラベルに一致するアラートが表示されます。例えば、environment=production、region=~US|EU、severity!=warning などです。
- [状態別にアラートをフィルタ] – [状態]] で、Active、Suppressed、Unprocessed のいずれかを選択すると、その状態のアラートが表示されます。

Prometheus データソースのアラート通知のサイレンス化

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

外部のアラートマネージャーデータソース (Amazon Managed Service for Prometheus を含む) では、[サイレンス]機能を使用してアラート通知を停止できます。サイレンスは通知の作成を停止するだけで、アラートルールの評価やユーザーインターフェースへのアラートインスタンスの表示は引き続き行われます。アラートをサイレンスにする場合、停止する時間枠を指定します。

外部アラートマネージャーデータソースのサイレンスを設定できます。

> **Note**
>
> アラート通知を一定の時間間隔 (定期的なメンテナンス期間中など) で停止するには、サイレンスではなく ミュートタイミング を使用します。

サイレンスの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [サイレンス] を選択するとページが開き、既存のコンタクトポイントが一覧表示されます。
3. [アラートマネージャー] ドロップダウンから外部アラートマネージャーを選択します。
4. [新規サイレンス] を選択します。
5. [サイレンスの開始と終了] でサイレンスを有効にする開始日と終了日を選択します。

   終了時刻を設定する代わりに、[期間] でサイレンスが適用される期間を指定します。この方法を使用すると、[サイレンスの開始と終了] フィールドの終了時刻が自動的に更新されます。
6. [名前] および [値] のフィールドには、1 つ以上の一致するラベルを入力します。照合機能により、どのルールにサイレンスを適用するかが決定されます。ラベルの照合については、この手順の後に詳しく説明します。
7. 必要に応じて、コメントを追加したり、作成者を変更してサイレンスの所有者を設定することができます。

アラート停止のラベル一致

サイレンスの作成時に、サイレンスの一部として一致するラベルのセットを作成します。これは、アラートが停止されるために一致する必要があるラベルに関するルールのセットです。一致するラベルは、次の 3 つの部分で構成されます。

- ラベル – 一致させるラベルの名前。アラートのラベル名と完全に一致する必要があります。

- [演算子] – ラベル値と一致するラベル値の比較に使用される演算子。利用できる演算子は次のとおりです。
  - = 値が指定された文字列と完全に一致するラベルが選択されます。
  - != 値が指定された文字列と一致しないラベルが選択されます。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルが選択されます (指定された文字列は正規表現として解釈されます)
  - != 指定された正規表現と一致しないラベルが選択されます。
- [値]) – ラベル値と一致する値。選択した演算子に応じて、文字列または正規表現として照合させることができます。

サイレンスは指定された終了日に終了しますが、いつでも手動で停止状態を解除することができます。

サイレンスを手動で終了する方法

1. [アラート] ページで、[サイレンス] を選択して既存のサイレンスのリストを表示します。
2. 終了するサイレンスを選択し、[サイレンス解除] を選択します。この操作によりアラートの停止状態は終了します。

> **Note**
>
> サイレンス解除を行うと、終了時刻が現在の時刻に設定されたものとして、アラートの停止状態が終了します。(自動または手動を問わず) 終了したサイレンスは5 日間保持され、一覧表示されます。リストからサイレンスを手動で削除することはできません。

サイレンス作成フォームへのリンクの作成

詳細が既に入力されているサイレンス作成フォームへの URL を作成できます。オペレータはこれを使用して、運用上のイベントを行う際にアラームをすばやく停止することができます。

サイレンスフォームへのリンクを作成するときは、`matchers` クエリパラメータを使用して一致するラベルを指定し、`comment` クエリパラメータを使用してコメントを指定します。`matchers` パラメータには、カンマで区切られた `[label][operator][value]` 形式の 1 つ以上の値が必要です。

URL の例

ラベル `severity=critical` と `cluster!~europe-.*` が一致し、`Silencing critical EU alerts` というコメントが付いたサイレンスフォームにリンクするには、次のような URL を使用します。*`mygrafana`* を Grafana インスタンスのホスト名に置き換えてください。

```
https://mygrafana/alerting/silence/new?matchers=severity%3Dcritical%2Ccluster!~europe-
*&comment=Silence%20critical%20EU%20alert
```

外部アラートマネージャーの新しいサイレンスページにリンクする場合、アラートマネージャーデータソース名を含む `alertmanager` クエリパラメータを追加します (例: `alertmanager=myAlertmanagerdatasource`)。

コンタクトポイント (通知先) の使用

> このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

連絡窓口は、アラート発生時の通知の受け取り方の設定に使用します。コンタクトポイントには、Amazon Simple Notification Service や Slack など、1 つ以上の通知方法を指定できます。アラートが発生すると、コンタクトポイントにリストされているすべての連絡窓口タイプに通知が送信されます。必要に応じて、メッセージテンプレートの使用 を使用して通知メッセージをカスタマイズすることができます。

> **Note**
>
> Grafana で管理されるアラートの通知先 (コンタクトポイント) は、作成および編集が可能です。アラートマネージャーアラートの通知先 (コンタクトポイント) は読み取り専用です。

コンタクトポイント (通知先) の使用

次の手順では、通知先 (コンタクトポイント) を追加、編集、テスト、削除する方法について説明します。

コンタクトポイントの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [コンタクトポイント]を選択し、[新しいコンタクトポイント]を選択します。
3. [アラートマネージャー] ドロップダウンから、アラートマネージャーを選択します。デフォルトでは、Grafana アラートマネージャーが選択されています。
4. コンタクトポイントの [名前] を入力します。
5. [コンタクトポイントタイプ] から、タイプを選択し、そのタイプの必須フィールドを選択します。例えば、Slack を選択した場合は、Slack チャネルと通知先となるユーザーを入力します。
6. 選択したコンタクトポイントで利用可能な場合は、オプションの設定を選択して追加の設定を指定します。
7. [通知設定]で[解決済みメッセージの無効化]を選択すると、アラートが解決した際に通知を受け取らないようにすることができます (任意)。
8. コンタクトポイントにコンタクトポイントタイプを追加したい場合は、新しいコンタクトポイントタイプを選択し、必要なコンタクトポイントタイプごとに同じ手順を繰り返します。
9. [コンタクトポイントの保存] を選択して変更を保存します。

コンタクトポイントの編集方法

1. [コンタクトポイント]を選択すると、現在設定されているコンタクトポイントのリストが表示されます。
2. 編集する連絡先を選択し、[編集] アイコン (ペン) を選択します。
3. 必要な変更を加え、[コンタクトポイントの保存] を選択して変更を保存します。

コンタクトポイントを作成したら、テスト通知を送信して、正しく設定されていることを確認します。

テスト通知の送信方法

1. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
2. テストする連絡先を選択し、[編集] アイコン (ペン) を選択します。
3. [テスト] アイコン (紙飛行機) を選択します。

4. 事前定義されたテスト通知を送信するか、[カスタム] を選択してテスト通知に独自の注釈とラベルを追加するかを選択します。
5. [テスト通知を送信]を選択して、指定されたコンタクトポイント宛にアラートをテストします。

使用されていないコンタクトポイントは、通知ポリシーで削除できます。

連絡先の削除方法

1. [コンタクトポイント] を選択して、現在設定されているコンタクトポイントのリストを開きます。
2. 削除する連絡先を選択し、[削除] アイコン (ゴミ箱) を選択します。
3. 確認ダイアログボックスで、[はい、削除] を選択します。

> **Note**
>
> コンタクトポイントが通知ポリシーで使用されている場合、コンタクトポイントを削除する前に、通知ポリシーを削除するか、別のコンタクトポイントを使用するように編集する必要があります。

サポートされている通知方法の一覧

名前	タイプ
Amazon SNS	`sns`
OpsGenie	`opsgenie`
Pager Duty	`pagerduty`
Slack	`slack`
VictorOps	`victorops`

メッセージテンプレートの使用

⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

コンタクトポイント (通知先) の使用 経由で送信される通知は、メッセージングテンプレートを使用して構築されます。Grafana のデフォルトテンプレートは Go テンプレートシステムに基づいており、一部のフィールドはテキストとして評価され、他のフィールドは HTML (エスケープに影響する可能性があります) として評価されます。

コンタクトポイントに使用するほとんどのフィールドはテンプレート化できるため、再利用可能なカスタムテンプレートを作成し、複数のコンタクトポイントで使用すると良いでしょう。この テンプレートのデータ トピックでは、テンプレートに使用できる変数一覧を紹介します。

テンプレートの使用

テンプレートはメッセージを作成するために使用されます。例えば、Slack のアラートメッセージでは、コンタクトポイントにタイトルと本文を設定することができます。次の例では、デフォルトのテンプレートを使用して、発行されたアラートの件数と解決済みのアラートの件数を含むタイトルと、アラートとそのステータスを一覧表示する本文を作成する方法を紹介します。

- タイトル:

```
{{ len .Alerts.Firing }} firing, {{ len .Alerts.Resolved }} resolved
```

- 本文テキスト:

```
{{ range .Alerts }}{{ .Status }}: {{ .Labels.alertname }}
{{end }}
```

カスタムテンプレートは次のように作成します。

- タイトル:

```
{{ template "slack.default.title" .}}
```

- 本文テキスト:

```
{{ template "mymessage" .}}
```

以下はサンプルテンプレートです。

```
{{ define "myalert" }}
  [{{.Status}}] {{ .Labels.alertname }}

  Labels:
  {{ range .Labels.SortedPairs }}
    {{ .Name }}: {{ .Value }}
  {{ end }}

  {{ if gt (len .Annotations) 0 }}
  Annotations:
  {{ range .Annotations.SortedPairs }}
    {{ .Name }}: {{ .Value }}
  {{ end }}
  {{ end }}

  {{ if gt (len .SilenceURL ) 0 }}
    Silence alert: {{ .SilenceURL }}
  {{ end }}
  {{ if gt (len .DashboardURL ) 0 }}
    Go to dashboard: {{ .DashboardURL }}
  {{ end }}
{{ end }}
```

以下に、カスタムメッセージテンプレートを作成、編集、削除する手順を紹介します。

メッセージテンプレートの作成方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [コンタクトポイント] を選択します。

3. [アラートマネージャー] ドロップダウンから、メッセージテンプレートを作成するアラートマネージャーインスタンスを選択します。デフォルトは Grafana アラートマネージャーです。
4. テンプレートを追加を選択します。
5. わかりやすい名前を付けます。
6. テンプレートの内容を追加します。例:

```
{{ define "mymessage" }}
  {{ range .Alerts }}
    [{{ .Status }}] {{ range .Labels }} {{ .Name }}={{.Value }}{{end}}
  {{ end }}
{{ end }}
```

   コンテンツセクションの `define` タグは、テンプレート名を割り当てます。このタグはオプションです。省略した場合、テンプレート名は名前フィールドから取得されます。両方を指定する場合、一致させておくことが推奨されます。
7. [テンプレートを保存] を選択します。

> **Note**
>
> アラートメッセージテンプレートの HTML はテキストとして表示され、制御文字はエスケープされます。Grafana では、通知での HTML 形式の表示はサポートされていません。

メッセージテンプレートを編集するには

1. アラートページで、[コンタクトポイント]を選択してコンタクトポイントのリストを開きます。
2. テンプレートテーブルの編集するテンプレートで編集アイコン (ペン) を選択します。
3. 変更後、[テンプレートを保存] を選択します。

メッセージテンプレートを削除するには

1. アラートページで、[コンタクトポイント]を選択してコンタクトポイントのリストを開きます。
2. テンプレートテーブルの削除するテンプレートで削除アイコン (ゴミ箱) を選択します。
3. [はい、削除します] を選択してテンプレートを削除します。

テンプレートのネスト

テンプレートは他のテンプレートに埋め込むことができます。

例えば、define キーワードを使用してテンプレートの一部を定義することができます。

```
{{ define "mytemplate" }}
  {{ len .Alerts.Firing }} firing. {{ len .Alerts.Resolved }} resolved.
{{ end }}
```

その後、 template キーワードを使用して、このフラグメント内にカスタムテンプレートを埋め込むことができます。例:

```
Alert summary:
{{ template "mytemplate" . }}
```

以下のビルトインのテンプレートオプションを使用して、カスタムテンプレートを埋め込むことができます。

名前	メモ
default.title	全体的なステータス情報を表示します。
default.message	発生中および解決済みのアラートの概要をフォーマット付きで提供します。

カスタムテンプレートの例

以下にカスタムテンプレートの使用方法の例を紹介します。

単一のアラートを表示するテンプレート:

```
{{ define "myalert" }}
  [{{.Status}}] {{ .Labels.alertname }}

  Labels:
  {{ range .Labels.SortedPairs }}
    {{ .Name }}: {{ .Value }}
  {{ end }}

  {{ if gt (len .Annotations) 0 }}
```

```
  Annotations:
  {{ range .Annotations.SortedPairs }}
    {{ .Name }}: {{ .Value }}
  {{ end }}
  {{ end }}

  {{ if gt (len .SilenceURL ) 0 }}
    Silence alert: {{ .SilenceURL }}
  {{ end }}
  {{ if gt (len .DashboardURL ) 0 }}
    Go to dashboard: {{ .DashboardURL }}
  {{ end }}
{{ end }}
```

通知メッセージ全体を表示するテンプレート:

```
{{ define "mymessage" }}
  {{ if gt (len .Alerts.Firing) 0 }}
    {{ len .Alerts.Firing }} firing:
    {{ range .Alerts.Firing }} {{ template "myalert" .}} {{ end }}
  {{ end }}
  {{ if gt (len .Alerts.Resolved) 0 }}
    {{ len .Alerts.Resolved }} resolved:
    {{ range .Alerts.Resolved }} {{ template "myalert" .}} {{ end }}
  {{ end }}
{{ end }}
```

テンプレートのデータ

メッセージテンプレートには、次のデータが渡されます。

名前	型	メモ
Receiver	文字列	コンタクトポイント (通知先) の名前。
Status	文字列	少なくとも 1 つのアラートが発生している場合は「発生中」となり、それ以外の場合「解決済」となります。

名前	型	メモ
Alerts	アラート	この通知に含まれるアラートオブジェクトのリスト (以下を参照)。
GroupLabels	KeyValue	これらのアラートがグループ化されたラベル。
CommonLabels	KeyValue	この通知に含まれるすべてのアラートに共通のラベル。
CommonAnnotations	KeyValue	この通知に含まれるすべてのアラートに共通する注釈。
ExternalURL	文字列	通知の送信元である Grafana へのリンク。外部のアラートマネージャーを使用している場合、そのアラートマネージャーへのリンク。

Alerts 型には、返されたアラートをフィルタリングするための 2 つの関数があります。

- Alerts.Firing – 発生中のアラートのリストを返します。
- Alerts.Resolved – 解決済みのアラートのリストを返します。

アラート (型)

アラート型には、次のデータがあります。

名前	型	メモ
ステータス	文字列	firing または resolved
ラベル	KeyValue	アラートに付与された一連のラベル。

名前	型	メモ
注釈	KeyValue	アラートに付与された一連の注釈。
StartsAt	time.Time	アラートが発行された時刻。
EndsAt	time.Time	アラートの終了時刻がわかっている場合にのみ設定されます。それ以外の場合、最後にアラートを受信してから設定可能なタイムアウト期間が適用されます。
GeneratorURL	文字列	Grafana または外部アラートマネージャーへのリンク。
SilenceURL	文字列	このアラートのラベルが事前に入力された grafana のサイレンスへのリンク。Grafana 管理のアラートのみ。
DashboardURL	文字列	Grafana ダッシュボードへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。
PanelURL	文字列	Grafana ダッシュボードパネルへのリンク (アラートルールが Grafana に属している場合)。Grafana 管理のアラートのみ。
Fingerprint	文字列	アラートの識別に使用するフィンガープリント。
ValueString	文字列	アラート内の各削減された式のラベルと値を含む文字列。

KeyValue 型

KeyValue 型は、キー (ラベル) 値 (注釈) の文字列を組み合わせたものです。

KeyValue として保存されたデータに直接アクセスするメソッドだけでなく、データをソート、削除、変換するメソッドも備えています。

名前	引数	戻り値	メモ
SortedPairs		キーと値の文字列のソート済みのリスト	
削除	[]string	KeyValue	指定したキーを除いたキー/値のマップのコピーを返します。
名前		[]string	ラベル名のリスト
値		[]string	ラベル値のリスト

テンプレート関数

テンプレート関数を使用すると、ラベルと注釈を処理して動的に通知を生成できます。以下の機能を使用できます。

名前	引数の型	戻り型	説明
humanize	数値または文字列	文字列	メトリクスプレフィックスを使用して、数値をより読み取り可能な形式に変換します。
humanize1024	数値または文字列	文字列	humanize と同様に、1000 ではなく 1024 を基準にします。

名前	引数の型	戻り型	説明
humanizeDuration	数値または文字列	文字列	時間を秒単位のより読みやすい形式に変換します。
humanizePercentage	数値または文字列	文字列	比率の値を 100 分の 1 の形に変換します。
humanizeTimestamp	数値または文字列	文字列	Unix のタイムスタンプを秒単位のより読みやすい形式に変換します。
title	文字列	文字列	strings.Title を使用すると、各単語の最初の文字が大文字になります。
toUpper	文字列	文字列	strings.ToUpper を使用すると、文字列がすべて大文字に変換されます。
toLower	文字列	文字列	strings.ToLower を使用すると、文字列がすべて小文字に変換されます。
match	パターン、テキスト	boolean	regexp.MatchString を使用すると、アンカーなしでの正規表現の一致テストを実行できます。

名前	引数の型	戻り型	説明
`reReplaceAll`	パターン、置換、テキスト	文字列	Regexp.ReplaceAllString を使用すると、アンカーなしでの正規表現による置換が実行できます。
`graphLink`	文字列 - `expr` および `datasource` フィールドを持つ JSON オブジェクト	文字列	指定された式とデータソースに対するグラフィカルビューへのパスを返します。
`tableLink`	文字列 - `expr` および `datasource` フィールドを持つ JSON オブジェクト	文字列	指定された式とデータソースに対するタブ形式ビューへのパスを返します。
`args`	[]interface{}	map[string]interface{}	オブジェクトのリストを arg0、arg1 などのキーを持つマップに変換します。複数の引数をテンプレートに渡す場合、この関数を使用します。
`externalURL`	なし	文字列	外部 URL を表す文字列を返します。
`pathPrefix`	なし	文字列	外部 URL のパスを返します。

次の表に、各関数の使用例を紹介します。

関数	TemplateString	Input	予想
humanize	{ humanize $value }	1234567.0	1.235M

関数	TemplateString	Input	予想
humanize1024	{ humanize1024 $value }	1048576.0	1Mi
humanizeDuration	{ humanizeDuration $value }	899.99	14 分 59 秒
humanizePercentage	{ humanizeP ercentage $value }	0.1234567	12.35%
humanizeTimestamp	{ humanizeTimestamp $value }	1435065584.128	2015-06-23 13:19:44. 128 +0000 UTC
title	{ $value | title }	aa bB CC	Aa Bb Cc
toUpper	{ $value | toUpper }	aa bB CC	AA BB CC
toLower	{ $value | toLower }	aa bB CC	aa bb cc
match	{ match "a+" $labels.i nstance }	aa	真
reReplaceAll	{{ reReplaceAll "localhost:(.*)" "my.domain:$1" $labels.instance }}	localhost:3000	my.domain:3000
graphLink	{{ graphLink "{\"expr\ ": \"up\", \"datasource \": \"gdev-prometheus \"}" }}		/explore?left=["no w-1h","now","gdev- prometheus",{"data source":"gdev-prom etheus","expr":"up ","instant":false, "range":true}]

関数	TemplateString	Input	予想
tableLink	{{ tableLink "{\"expr\":\"up\", \"datasource\":\"gdev-prometheus\"}" }}		/explore?left=["now-1h","now","gdev-prometheus",{"datasource":"gdev-prometheus","expr":"up","instant":true,"range":false}]
args	{{define "x"}}{{.arg0}} {{.arg1}}{{end}}{{ template "x" (args 1 "2")}}		1 2
externalURL	{ externalURL }		http://localhost/path/prefix
pathPrefix	{ pathPrefix }		/path/prefix

通知ポリシーの使用

> ⚠ このドキュメントのトピックは、Grafana バージョン 8.x をサポートする Grafana ワークスペース向けです。
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 での作業」を参照してください。
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 での作業」を参照してください。

通知ポリシーでは、アラートがコンタクトポイント (通知先) にどのように配信されるかを指定します。ポリシーはツリー構造で、ポリシーごとに 1 つ以上の子ポリシーを持つことができます。ルートポリシーを除く各ポリシーは、特定のアラートラベルにも一致させることができます。各アラートはルートポリシーによって評価され、次に各子ポリシーによって評価されます。特定のポリシーに対して `Continue matching subsequent sibling nodes` オプションを有効にすると、1 つ以上の一致があっても評価が続行されます。子ポリシーのいずれとも一致しない場合、親ポリシーの設定

とコンタクトポイント (通知先) の情報によってアラートの動作が制御されます。特定のポリシーと一致しない場合、ルートポリシーがアラートを制御します。

> **Note**
>
> Grafana 管理のアラートの通知ポリシーを作成および編集することができますが、アラートマネージャーアラートの通知ポリシーは読み取り専用となっています。

通知のグループ化

グループ化すると、似た性質のアラート通知を 1 つの funnel にまとめられます。これは、システム上で一度に複数の障害が発生して多数のアラートが同時に発生する場合、大規模な停止中にアラート通知を制御することができます。

グループ化の例

例えば、異なる環境でデータベースに接続されている100のサービスがあるとします。これらのサービスは、ラベル `env=environmentname` で区別され、サービスがデータベースに到達できるかどうかを監視するためのアラートルールが設定されています。このアラートルールは、`alertname=DatabaseUnreachable` という名前のアラートを作成します。

サービスの半分がデータベースに到達できなくなったネットワークの分断が発生すると、50 件のアラートが個別に発生しますが、影響を受けている環境のリストを含む (50 件ではなく) 1 ページの通知を受け取りたいと考えています。

`group_by: [alertname]`でグループ化を設定することができます(サービスごとに異なる `env` ラベルを使用する代わりに)。この設定を行うと、Grafana は、このアラートルールの影響を受けるすべての環境が記載された 1 通のコンパクトな通知を送信します。

特殊グループ

Grafana には 2 つの特殊グループがあります。デフォルトのグループである `group_by: null` は、すべてのアラートを 1 つのグループにグループ化します。`...` という名前の特殊ラベルを使用してすべてのラベルでアラートをグループ化することで、グループ化を無効にし、各アラートを独自のグループに送信することもできます。

通知の使用

以下に、通知ポリシーを作成および管理するための手順を紹介します。

ルート通知ポリシーの編集方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [ルートポリシー] セクションで、[編集] アイコン (ペン) を選択します。
5. [デフォルトのコンタクトポイント] で、アラートルールが特定のポリシーと一致しない場合に通知を送信するコンタクトポイント (通知先) を更新します。
6. [グループ化対象] で、アラートをグループ化するラベル (または特殊グループ) を選択します。
7. [タイミングオプション] で、以下のいずれかを選択します。
   - [グループ待機] – 同じグループのアラートをバッファリングしてから、最初の通知を送信するまでの待機時間。デフォルト値は 30 秒です。
   - [グループ間隔] – グループ内の 2 つの通知の最小時間間隔。デフォルト値は 5 分です。
   - [繰り返し間隔] – グループに新しいアラートが追加されなかった場合に、通知を再送信するまでの最小時間間隔。デフォルトは 4 時間です。
8. [保存] を選択して変更を保存します。

上位の特定のポリシーの追加方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [特定のルーティング] セクションで、[新しい特定のポリシー] を選択します。
5. [一致するラベル] セクションで、一致するアラートラベルを 1 つ以上追加します。ラベルの一致の詳細については、このトピックの後半で説明します。
6. コンタクトポイントで、アラートがこの特定のポリシーと一致する場合に通知を送信するコンタクトポイントを追加します。ネストされたポリシーは、このコンタクトポイントを上書きします。
7. 必要に応じて [グループ化を上書き] を選択して、ルートポリシーとは異なるグループ化を指定します。

8. 必要に応じて [一般的なタイミングを上書き] を選択して、グループ通知ポリシーで設定されているタイミングオプションを上書きします。
9. [ポリシーの保存] を選択して変更を保存します。

ネストしたポリシーの追加方法

1. ネストしたポリシーの作成先となる特定のポリシーを展開します。
2. [ネストしたポリシーの追加] を選択し、詳細を追加します (最上位の特定のポリシーを追加する場合と同様)。
3. [ポリシーの保存] を選択して変更を保存します。

特定のポリシーの編集方法

1. [アラート] ページから [通知ポリシー] を選択するとページを開き、現在設定されているポリシーが一覧表示されます。
2. 編集する設定ポリシーを選択したら、[編集] アイコン (ペン) を選択します。
3. 必要な変更を加えます (最上位の特定のポリシーを追加する場合と同様)。
4. [ポリシーを保存] を選択します。

ラベル一致の仕組み

ポリシーは、アラートのラベルがポリシーで指定されたすべての一致ラベルと合致する場合に、そのアラートに適用されます。

- ラベル – 一致させるラベルの名前。アラートのラベル名と完全に一致する必要があります。
- [演算子] – ラベル値と一致するラベル値の比較に使用される演算子。利用できる演算子は次のとおりです。
  - = 値が指定された文字列と完全に一致するラベルが選択されます。
  - != 値が指定された文字列と一致しないラベルが選択されます。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルが選択されます (指定された文字列は正規表現として解釈されます)。
  - != 指定された正規表現と一致しないラベルが選択されます。
- [値]) – ラベル値と一致する値。選択した演算子に応じて、文字列または正規表現として照合させることができます。

ミュートタイミング

ミュートタイミングとは、ポリシーの新しい通知が生成されたり送信されないようにする繰り返しの時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間や繰り返し発生するのを防ぐために使用されます。

サイレンスと似ていますが、ミュートタイミングはアラートルールの評価や、ユーザーインターフェースへのアラート表示を停止せず、通知の作成のみを防ぎます。

外部のアラートマネージャーデータソースの Grafana 管理のミュートタイミングとミュートタイミングを設定できます。

サイレンスとミュートタイミングの比較

次の表に、「ミュートタイミング」と「サイレンス」の違いを示します。

ミュートタイミング	サイレンス
繰り返しの時間間隔を定義します	開始時刻と終了時刻を固定します
作成され、通知ポリシーに追加されます	ラベルを使用してアラートと照合し、サイレンスするかどうかが判断されます

ミュートタイミングの作成方法

1. Grafana コンソールの Grafana メニューで、[アラート] (ベル) アイコンを選択して [アラート] ページを開きます。
2. [通知ポリシー] を選択します。
3. [アラートマネージャー] ドロップダウンから、編集するアラートマネージャーを選択します。
4. [ミュートタイミング] セクションで、[ミュートタイミングの追加] ボタンを選択します。
5. ミュートタイミングを適用したい時間間隔を選択します。
6. [送信] を選択してミュートタイミングを作成します。

通知ポリシーへのミュートタイミングの追加方法

1. ミュートタイミングを追加する通知ポリシーを選択し、[編集] ボタンを選択します。
2. [ミュートタイミング] ドロップダウンから、ポリシーに追加するミュートタイミングを選択します。

[ポリシーを保存] ボタンを選択します。

時間間隔

時間間隔には、時間範囲を指定します。この時間間隔内にアラートが発生した場合、抑制されます。範囲には : を使用して表現することができます (例: monday:thursday)。ミュートタイミングには複数の時間間隔を含めることができます。時間間隔は複数のフィールド (詳細は次のリストを参照) で構成され、アラートを抑制するには、すべてのフィールドが一致する必要があります。例えば、曜日に月曜日から金曜日 monday:friday、時間範囲を 8:00 ~ 9:00 と指定した場合、アラートは月曜日から金曜日の 8:00 ~ 9:00 の間は抑制されますが、土曜日の 8:00 ~ 9:00 は抑制されません。

- [時間範囲] – 通知を抑制する時間帯。[開始時刻]と[終了時刻] の 2 つのサブフィールドで構成されます。時刻は 14:30 のように指定します。時刻は UTC の 24 時間表記です。
- [曜日] - 曜日。単一の日 (例: monday)、範囲 (例: monday:friday)、またはカンマで区切られた複数の日のリスト (例: monday, tuesday, wednesday) を指定できます。
- [月] - 選択する月。月は、数値で指定することも、完全な月名で指定することもできます。例えば、1 または january のどちらでも 1 月を指定できます。単一の月、月の範囲、またはカンマ区切りの月のリストを指定できます。
- [月の日付] – 1 か月内の日付。値は 1 ~ 31 の範囲で指定します。負の値は月の日を逆順で指定することを意味します。たとえば、-1 は月の最終日を表します。日付は単一の日、日付の範囲、またはカンマで区切った日付のリストを指定できます。

設定を変更する

[設定] タブでは、さまざまな作業を行うことができます。プロファイルの編集、Amazon Managed Grafana の設定の変更、プロファイルと Amazon Managed Grafana の使用に関する情報の表示を行うことができます。

> **Note**
>
> ワークスペースの設定変更については、「Amazon Managed Grafana ワークスペースを設定する」を参照してください。

Amazon Managed Grafana プロファイルを編集する

プロファイルには、名前、ユーザー名、メールアドレスなどを設定します。

プロファイルの編集方法

1. 画面の左下隅にあるユーザーアイコンを選択し、[Preferences] (設定) を選択します。
2. [プロファイルの編集] セクションで編集できる内容には、以下のものがあります。
   - [[名前] – プロファイルに関連付けられた表示名を変更するには、このフィールドを編集します。
   - [Email] (メールアドレス) – プロファイルに関連付けられたメールアドレスを変更するには、このフィールドを編集します。
   - [ユーザー名] – ユーザー名を変更するには、このフィールドを編集します。
3. [保存] を選択します。

設定を編集する

設定には、ダークテーマまたはライトテーマの使用の選択、ホームダッシュボード、タイムゾーンなどがあります。

> **Note**
> パーソナルインスタンスの設定は、管理者がインスタンスまたはチームレベルで行った設定よりも優先されます。

設定の変更方法

1. 画面の左下隅にあるユーザーアイコンを選択し、[Preferences] (設定) を選択します。
2. [Preferences] (設定) セクションで編集できる内容には、以下のものがあります。
   - [UI Theme] (UI のテーマ) – テーマは、ダークまたはライトから選択できます。デフォルトを選択した場合、ダークテーマまたは Grafana 管理者によって選択されたテーマのいずれかが適用されます。
   - [ホームダッシュボード]

- [タイムゾーン] – タイムゾーンの一覧から 1 つ選択します。[デフォルト] を選択した場合、ブラウザのローカルタイムゾーンまたは Grafana 管理者によって選択されたタイムゾーンのいずれかが適用されます。詳細については、「」を参照してください時間範囲コントロール。

3. [保存] を選択します。

Amazon Managed Grafana セッションを表示する

Amazon Managed Grafana は、各 Grafana ワークスペースにセッションを記録します。Amazon Managed Grafana 認証情報が不正使用された疑いがある場合は、このセクションを確認してください。

セッションに関する情報の表示方法

1. 画面の左下隅にあるユーザーアイコンを選択し、[Preferences] (設定) を選択します。
2. [Sessions] (セッション) セクションまで下にスクロールします。以下の内容が表示されます。
   - [最終アクセス] — 最後にログオンしてからの時間。
   - [ログオン日] – 現在の Grafana インスタンスにログオンした日付。
   - [IP アドレス] – ログオン元の IP アドレス。
   - [Browser & OS] (ブラウザと OS) – Grafana へのログオンに使用したウェブブラウザとオペレーティングシステム。
   - この画面を表示しているユーザーがそのインスタンスの Grafana 管理者である場合、セッション行の赤いサインアウトアイコンを選択してセッションを取り消すことができます。

サポート用情報の収集

サポートバンドルは、ユーザーインターフェイスを使用て Grafana ワークスペースに関する情報を収集するための簡単な方法を提供します。Grafana ワークスペースで問題が発生する場合は、次のようなワークスペースに関する情報を含むサポートバンドルを製品サポートに送信できます。

- Grafana バージョン
- インストール済みのプラグイン
- Grafana の設定
- データベースの情報と移行

> **Note**
>
> サポートバンドルは Grafana バージョン 10 以降と互換性のあるワークスペースでのみ使用できます。

サポートバンドルのコンポーネント

サポートバンドルには次のいずれかのコンポーネントを含めることができます。

- 使用統計 – Grafana ワークスペースの使用統計。
- ユーザー情報 – Grafana ワークスペースのユーザーのリスト。
- データベースと移行の情報 – データベースの情報、および移行ログ。
- プラグイン情報 – ワークスペースにインストール済みのプラグインに関する情報。
- 基本情報 – バージョン、メモリ使用量などの、Grafana ワークスペースに関する基本情報。
- 設定 – Grafana ワークスペースの設定。
- SAML – SAML のヘルスチェック接続とメタデータ (SAML が有効になっている場合にのみ表示)。
- LDAP – LDAP のヘルスチェック接続とメタデータ (LDAP が有効になっている場合にのみ表示)。
- OAuth2 – 各 OAuth2 プロバイダーのヘルスチェック接続とメタデータ (OAuth プロバイダーが有効になっている場合にのみ表示)。

サポートバンドルの作成

以下の手順に従って、サポートバンドルを作成します。

> **Note**
>
> この手順では、ワークスペースに `admin` アクセス許可が必要です。

サポートバンドルの作成方法

1. Grafana ワークスペースにサインインします。
2. ヘルプアイコンを選択します。
3. ヘルプメニューから、[サポートバンドル] を選択します。

4. [新規サポートバンドル] を選択します。
5. サポートバンドルに含めるコンポーネントを選択します。
6. [作成] を選択します。
7. サポートバンドルの準備ができたら、[ダウンロード] を選択します。

   Grafana はサポートバンドルをアーカイブ (`tar.gz`) ファイルにダウンロードします。

ファイルを開くとサポートバンドルの内容を表示できます。必要に応じて、 ファイルをサポートに直接送信できます。バンドルに個人情報が含まれていて、プライベートではないチャンネル経由で送信する必要がある場合は、暗号化を検討してください。送信する前に、age などのツールを使用してファイルを暗号化できます。

従来のダッシュボードアラート

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

Note

このセクションでは Grafana の従来のダッシュボードアラートシステムについて説明します。Prometheus アラートマネージャーアラートと Grafana アラートを表示するように設計された、新しい Grafana アラートへの移行と使用については、「Grafana バージョン 10 のアラート」を参照してください。
GrafanaLabs は、Grafana のバージョン 11 で従来のダッシュボードアラートを削除することを発表しました。

従来のダッシュボードアラートは次の 2 つの部分で構成されます。

- アラートルール – アラートがトリガーされるとき。アラートルールは、Grafana が定期的に評価する 1 つ以上の条件によって定義されます。
- 通知チャネル – アラート配信方法。アラートルールの条件が満たされると、Grafana はそのアラートのために設定されたチャネルに通知します。

現在アラートをサポートしているのはグラフパネルの視覚化のみです。

アラート設定

アラートは Amazon Managed Grafana ワークスペースで設定できます。

- アラート通知チャネルを追加または編集します。詳細については、「通知」を参照してください。
- アラートルールを作成します。詳細については、「アラートの作成」を参照してください。
- 既存のアラートルールと、現在の状態を表示します。詳細については、「既存のアラートルールの表示」を参照してください。
- アラートルールをテストしてトラブルシューティングします。詳細については、「アラートのトラブルシューティング」を参照してください。

クラスタリング

現在サポートされているアラートは、限られた形式の高可用性です。複数のワークスペースを実行すると、アラート通知の重複が排除されます。つまり、すべてのアラートはすべてのサーバーで実行されますが、重複排除ロジックのおかげで重複するアラート通知は送信されません。

通知

問題を解決する方法、ランブックへのリンクなど、詳細なメッセージを含むアラートルールを作成できます。

実際の通知が設定され、複数のアラート間で共有されます。

アラートの実行

アラートルールは、 Amazon Managed Grafana のスケジューラーとクエリ実行エンジンで評価されます。

アラート通知

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

アラートの状態が変化すると、通知が送信されます。各アラートルールには、複数の通知を含めることができます。アラートルールに通知を追加するには、まず notification チャネルを追加および設定する必要があります。

これは通知チャネルページから実行します。

通知チャネルの追加

1. サイドバーで、アラート (ベル) アイコンを一時停止してから、通知チャネルを選択します。
2. [チャンネルの追加] を選択します。
3. フィールドに記入、または以下のセクションで説明されているオプションを選択します。

新しい通知チャネルのフィールド

デフォルト (すべてのアラートで送信)

- 名前 - チャネルの名前を入力します。ユーザーがアラートルールに通知を追加するときに表示されます。
- タイプ – チャンネルのタイプを選択します。詳細については、「サポートされている通知方法の一覧」を参照してください。
- デフォルト (すべてのアラートで送信) – このオプションを選択すると、すべてのアラートルールでこのチャネルの通知が送信されます。

- 解決済みメッセージの無効化 – このオプションを選択すると、アラート状態が偽に戻ったときに送信される解決メッセージである [OK] が無効になります。
- リマインダーの送信 - このオプションを選択すると、アラートで追加の通知 (リマインダー) が送信されます。リマインダーの送信頻度は、秒 (s)、分 (m)、時間 (h) などを使用して、`30s`、`3m`、`5m`、`1h` のように指定できます。

> **Important**
>
> アラートリマインダーは、ルールの評価後に送信されます。そのため、設定済みのアラートルール評価間隔よりも頻繁にリマインダーを送信することはできません。

次の例は、トリガーされたアラートに対してリマインダーが送信される、頻度とタイミングを示しています。

アラートルール評価間隔	リマインダーの送信間隔	リマインダーの送信間隔 (前回のアラート通知後)
`30s`	`15s`	30 秒以内
`1m`	`5m`	5 分以内
`5m`	`15m`	15 分以内
`6m`	`20m`	24 分以内
`1h`	`15m`	1 時間以内
`1h`	`2h`	2 時間以内

サポートされている通知方法の一覧

名前	型	イメージをサポート	アラートルールタグをサポート
Amazon Simple Notification Service	sns	いいえ	あり

名前	型	イメージをサポート	アラートルールタグをサポート
OpsGenie	`opsgenie`	いいえ	あり
PagerDuty	`pagerduty`	なし	あり
Slack	`slack`	いいえ	いいえ
VictorOps	`victorops`	いいえ	いいえ

Amazon Simple Notification Service

サービスマネージドアクセス許可を有効にし、ワークスペースの通知チャネルに Amazon SNS を含めると、SNS トピック ARN の提供が通知チャネルの作成時にのみになります。[名前] フィールドに、作成した SNS トピックの名前を入力します。サービスマネージドアクセス許可を使用してワークスペースを作成する場合に、通知をトピックに正常に発行するには、SNS トピック名に `grafana` のプレフィックスを付ける必要があります。ワークスペースの作成時にカスタマーマネージドアクセス許可を選択する場合は、SNS トピック名に `grafana` のプレフィックスを付ける必要はありません。

[トピック] フィールドに、SNS トピックの ARN をコピーして貼り付けます。[メッセージ本文の形式] では、JSON またはテキストオプションのいずれかを選択できます。

オプションの AWS SNS 設定 フィールドで、メッセージ内のすべてのタグを含める チェックボックスをオンにして、メッセージ本文内のすべての Grafana タグを表示します。

ワークプレイスでカスタマーマネージドアクセス許可を使用する場合は、指定する IAM ロールに SNS トピックの SNS 公開権限を含める必要があります。

Slack

Slack を設定するためには、受信 Slack ウェブフック URL を設定する必要があります。詳細については、「Sending messages using incoming webhooks」を参照してください。

Slack ボット統合の設定の詳細については、「Follow Slack's guide to set up a bot integration」を参照してください。「xoxb」で始まる、提供されたトークンを使用します。

設定	説明
Url	Slack 受信ウェブフック URL、または Chat.post Message Slack API エンドポイント。
ユーザーネーム	ボットメッセージのユーザー名を設定します。
受取人	これを使用して Slack の受信者を上書きします。チャネル Slack ID、ユーザー Slack ID、ユーザー名リファレンス (@<user>、すべて小文字、空白なし)、またはチャネルリファレンス (#<channel>、すべて小文字、空白なし) のいずれかを指定する必要があります。chat.postMessage Slack API エンドポイントを使用する場合はこれが必要です。
アイコンの絵文字	ボットメッセージのアイコンとして使用する絵文字を指定します。例えば、:smile:
アイコンの URL	ボットメッセージのアイコンとして使用する画像の URL を指定します。
ユーザーの言及	必要に応じて、Grafana が送信する Slack 通知で 1 人以上のユーザーについて言及します。カンマで区切られたユーザーを表示するには、対応する Slack ID を使用して、各ユーザーの Slack プロファイルのオーバーフローボタンを選択します。
グループの言及	必要に応じて、Grafana が送信する Slack 通知で 1 人以上のグループについて言及します。対応する Slack ID (各グループの Slack プロファイル URL から取得できます) を使用して、カンマで区切られたグループを表示できます。
チャネルの言及	必要に応じて、すべてのチャネルメンバー、またはアクティブなチャネルメンバーのみのいずれかに言及します。

設定	説明
トークン	指定すると、Amazon Managed Grafana は、外部イメージの送信先ではなく、Slack file.upload API オペレーションを介して生成されたイメージをアップロードします。`chat.postMessage` Slack API エンドポイントを使用する場合はこれが必要です。

Slack ボットにトークンを使用する場合は、通知を送信するチャネルにボットを招待する必要があります。受信者フィールドにチャネルを追加します。

PagerDuty

PagerDuty を設定するため、統合キーを指定します。

設定	説明
統合キー	PagerDuty の統合キー。
緊急度	動的通知のレベルで、デフォルトは `critical` (1) です。
インシデントの自動解決	アラートが OK に戻った後に PagerDuty でインシデントを解決します。
メッセージの詳細	PD のサマリーフィールドからアラートメッセージを削除し、代わりにカスタム詳細に配置します (2)。

> **Note**
>
> タグ `Severity`、`Class`、`Group`、`dedup_key`、`Component` は [PagerDuty 共通イベント形式 – PD-CEF] で特別な役割があります。アラートパネルによりこれらのタグキーが定義されると、PagerDuty に送信されたイベントのルートに変換されます。つまり PagerDuty UI およびフィルタリングツール内でこれらのタグを使用できます。アラートに設定された重要

度タグが有効なレベルである場合、通知チャネルに設定されたグローバル重要度が上書きされます。

> **Note**
>
> メッセージの詳細を使用すると、PagerDuty イベントの `custom_details` フィールドの構造が変更されます。`payload.custom_details` のフィールドに依存すると、PagerDuty ルールのカスタムイベントルールが壊れる可能性があります。`custom_details.myMetric` を使用する既存のルールを `custom_details.queries.myMetric` に移動します。

> **Note**
>
> `dedup_key` タグを使用すると、カスタムキーにより Grafana が生成した `dedup_key` が上書きされます。

アラート通知から Grafana に戻るリンクの設定

すべてのアラート通知には、Grafana ワークスペースでトリガーされたアラートに戻るリンクが含まれています。

アラートの作成

> **Warning**
>
> このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
>
> Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
>
> Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
>
> Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

Amazon Managed Grafana アラートを使用すると、ダッシュボードパネルにルールをアタッチできます。ダッシュボードを保存すると、Amazon Managed Grafana はアラートルールを別のアラートルールストレージに抽出して、評価をスケジュールします。

グラフパネルの [アラート] タブで、アラートルールを評価する頻度と、アラートの状態を変更して通知を開始するために満たす必要がある条件を設定できます。

現在アラートをサポートしているのはグラフパネルのみです。

アラートルールの追加または編集

1. アラートルールの追加または編集用のパネルに移動して、タイトルを選択し、[編集] を選択します。
2. [アラート] タブで、[アラームを作成] を選択します。このパネルにアラートが既に存在する場合は、[アラート] タブでフィールドを編集できます。
3. フィールドに入力します。詳細については、「アラートルールフィールド」を参照してください。
4. ルールの作成が終了したら、右上隅にある [保存] を選択して、アラートルールとダッシュボードを保存します。
5. (オプションですが推奨) ルールが期待どおりの結果を確実に返すようにするため、[ルールをテスト] を選択します。

アラートルールの削除

アラートを削除するには、アラートの一番下までスクロールして、[削除] を選択します。

アラートルールフィールド

このセクションではアラート作成で入力するフィールドについて説明します。

ルール

- 名前 - わかりやすい名前を入力します。その名前は [メンバー] リストに表示されます。
- 評価頻度 — スケジューラーがアラートルールを評価する頻度を指定します。これは評価間隔と呼ばれます。
- 長さ — 設定済みのしきい値をクエリが超えてから、アラート通知がトリガーされるまでの長さを指定ます。

> **Warning**
>
> If no data or all values are null 設定を No Data にしたままで For を使用しないでください。No Data のトリガーがすぐにトリガーされ、For が考慮されません。これにより、アラートが No Data -> Pending -> OK から移行しても OK 通知が送信されない場合もあります。

アラートルールに For が設定されていて、設定されたしきい値をクエリが超えた場合、まず OK から Pending に移行します。OK から Pending に移行すると、Amazon Managed Grafana では通知が送信されません。For より長い期間アラートルールが発生すると、Alerting に変更されてアラート通知が送信されます。

通常は、この設定を使用することをお勧めします。アラート通知の開始を数分待つより、誤検出の発生の方が悪いことが多いためです。Alert list または Alert list panels を見ると、保留状態のアラートを確認できます。

条件

現在、既存の条件タイプは、クエリ文字、時間範囲、集計関数を指定できる Query 条件のみです。

クエリ条件例

```
avg() OF query(A, 15m, now) IS BELOW 14
```

- avg() は、しきい値と比較できる値にするために各シリーズの値を減らす方法を制御します。関数を選択して別の集計関数に変更します。
- query(A, 15m, now) 文字はメトリクスタブから実行するクエリを定義します。2 番目の 2 つのパラメータは時間範囲を定義します。15m, now は、15 分前から現在までを意味します。また、10m, now-2m を使用して、10 分前から 2 分前の時間範囲を定義できます。これは過去 2 分間のデータを無視する場合に役立ちます。
- IS BELOW 14 はしきい値のタイプおよびしきい値を定義します。しきい値のタイプを変更するには IS BELOW を選択します。

アラートルールで使用されるクエリにテンプレート変数を含めることはできません。現在、条件間の OR と AND 演算子のみがサポートされており、これらは連続的に実行されま

す。例えば、condition:A(evaluates to: TRUE) OR condition:B(evaluates to: FALSE) AND condition:C(evaluates to: TRUE) の 3 つの条件の順序では、((TRUE OR FALSE) AND TRUE) = TRUE として計算されます。

複数のシリーズ

クエリが複数のシリーズを返す場合は、集計関数としきい値チェックがシリーズごとに評価されます。現在、Amazon Managed Grafana はシリーズごとにはアラートルール状態を追跡しません。この結果は、次のシナリオで説明されます。

- server1 と server2 の 2 つのシリーズを返すクエリを含むアラート条件。
- server1 シリーズでは、アラートルールが発生して`Alerting`状態に切り替わります。
- 通知は load peaking (server1) というメッセージとともに送信されます。
- 同じアラートルールのその後の評価では、server2 シリーズによりアラートルールが発生されます。
- アラートルールが既に `Alerting` 状態であるため、新しい通知は送信されません。

前のシナリオからわかるように、ルールが既に `Alerting` 状態にある場合、Grafana は他のシリーズでアラートが発生しても通知を送信しません。

> **Note**
>
> トリガーされたアラートを送信するリマインダーを設定できます。これにより、アラートが引き続き発生するときに追加の通知が送信されます。他のシリーズ (前の例の server2 など) でもアラートルールが発生する場合は、リマインダー通知に含められます。使用する通知チャネルによっては、この機能を利用して、アラートが発生している新規または既存のシリーズを特定できる場合があります。

データなしまたはエラー処理

次の表は、ルール評価エンジンがデータなしまたは null 値のみを返すクエリを処理する方法を制御する条件を示しています。

データがない場合の選択肢	説明
データなし	アラートルールの状態を NoData に設定します。
[アラート]	アラートルールの状態を Alerting に設定します。
最後の状態を保持する	現在のアラートルール状態を、そのままにしておきます。
OK	サポートされますが、通常は役に立ちません。

実行エラーまたはタイムアウト

以下のオプションは、Amazon Managed Grafana が実行エラーまたはタイムアウトエラーを処理する方法を指定します。

エラーまたはタイムアウト時の選択肢	説明
[アラート]	アラートルールの状態を Alerting に設定します。
最後の状態を保持する	現在のアラートルール状態を、そのままにしておきます。

クエリがタイムアウトまたはランダムに失敗する信頼性の低い時系列ストアがある場合は、基本的に無視するようにこのオプションを Keep Last State に設定することができます。

通知

[アラート] タブでは、アラートルールの通知と、アラートルールに関する詳細なメッセージも指定できます。メッセージには、問題の解決方法、ランブックへのリンクなど、任意の情報を含めることができます。

実際の通知が設定され、複数のアラート間で共有されます。通知の設定およびセットアップ方法については、「アラート通知」を参照してください。

- 送信先 – アラート通知チャネルを設定済みの場合は、そのチャネルを選択します。
- メッセージ – 通知チャネルで送信するテキストメッセージを入力します。一部のアラート通知では HTML またはその他のリッチ形式へのテキスト変換をサポートしています。
- タグ – 通知に含めるタグのリスト (キー値) を指定します。一部の通知子でのみサポートされています。

アラートの状態履歴と注釈

アラート状態の変更は、Amazon Managed Grafana データベース内の内部注釈テーブルに記録されます。状態の変更はアラートルールのグラフパネルで注釈として視覚化されます。[アラート] タブの `State history` サブメニューに移動して、状態履歴を表示およびクリアすることもできます。

アラートルールの一時停止

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

アラートルール評価の一時停止が便利な場合があります。例えば、メンテナンスウィンドウ中にアラートルールを一時停止すると、アラートのフラッド発生を防止できる場合があります。

1. Grafana のサイドバーで、[アラート] (ベル) アイコンを一時停止してから [アラートルール] を選択します。設定済みのすべてのアラートルールが、現在の状態とともに一覧表示されます。
2. リスト内でアラートを検索して、右の一時停止アイコンを選択します。一時停止アイコンは再生アイコンに変わります。
3. アラート評価を再開するには、再生アイコンを選択します。

既存のアラートルールの表示

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

Amazon Managed Grafana では個々のアラートルールは定義済みパネルに保存されますが、既存のすべてのアラートルールとその現在の状態のリストを表示することもできます。

Grafana のサイドバーで、[アラート] (ベル) アイコンを一時停止してから [アラートルール] を選択します。設定済みのすべてのアラートルールが、現在の状態とともに一覧表示されます。

アラートの表示中には、以下を実行できます。

- 名前によるアラートのフィルタリング – アラート名を [検索] フィールドに入力します。
- 状態によるアラートのフィルタリング – [状態] で、表示するアラート状態を選択します。他のすべては非表示になります。
- アラートの一時停止または再開 – アラートの横にある一時停止または再生アイコンを選択して、評価を一時停止または再開します。
- アラートルール設定へのアクセス — アラート名または[アラートルールの編集] (歯車) アイコンを選択します。Amazon Managed Grafana は、アラートルールが定義されているパネルの [アラート] タブを開きます。これはアラートが発生していても、どのパネルで定義されているかわからない場合に役立ちます。

通知テンプレートの作成

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

アラート通知テンプレート作成を使用すると、アラートクエリからラベル値を取得してアラート通知に挿入できます。

アラートクエリの評価に存在するラベルは、アラートルール名とアラート通知メッセージフィールドで使用できます。アラートがアラート状態にあるとき、アラートラベルデータは通知フィールドに投入されます。同じラベルに複数の一意の値がある場合、値はカンマで区切られます。

アラート通知にアラートラベルデータを追加する方法

1. アラートルールを追加または編集するパネルに移動します。
2. パネルタイトルを選択して [編集] を選択します。
3. [アラート] タブで、[アラームを作成] を選択します。このパネルにアラートがすでに存在する場合、直接それを編集できます。
4. `${Label}` 構文を使用して、アラートルール名またはアラート通知メッセージフィールドの、アラートクエリラベルを参照します。アラートクエリラベルの詳細については、Grafana ドキュメント「メッセージテンプレート」を参照してください。
5. 右上隅の [保存] を選択します。

アラートのトラブルシューティング

⚠ このドキュメントのトピックでは、Grafana での従来のアラートについて説明します。これは、Amazon Managed Grafana の将来のバージョンではサポートされません。Grafana アラートに移行すると、最新のアラート機能を使用できます。詳細については、次のトピックのいずれかを参照してください。
Grafana バージョン 10.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 10 のアラート」を参照してください。
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「Grafana バージョン 9 のアラート」を参照してください。
Grafana バージョン 8.x をサポートする Grafana ワークスペースについては、「Grafana アラート」を参照してください。

アラートが想定どおりに動作しない場合、次の手順はトラブルシューティングおよび問題の特定に役立ちます。

トラブルシューティングで実行できる最初のレベルは、テストルール を選択することです。クエリから返された未加工データが表示される時点まで、結果を展開できます。

Grafana HTTP API の使用

Amazon Managed Grafana ワークスペースでは、Grafana HTTP API を使用できます。以下のセクションでは、API の使用方法について説明し、サポートされている Grafana API の一覧を示します。

> **Note**
>
> Amazon Managed Grafana には AWS APIs も用意されています。 AWS APIs「Amazon Managed Grafana API リファレンス」を参照してください。

トピック

- トークンを使用した認証
- アラート API
- アラート通知チャネル API
- 注釈 API
- 認証 API
- ダッシュボード API
- ダッシュボードアクセス許可 API
- ダッシュボードバージョン API
- データソース API
- データソースアクセス許可 API
- 外部グループ同期 API
- フォルダ API
- フォルダ/ダッシュボード検索 API
- フォルダアクセス許可 API
- 組織 API
- プレイリスト API
- プラグイン API
- 設定 API
- スナップショット API

- チーム API
- ユーザー API

トークンを使用した認証

Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana トークンが必要です。このトークンは、API呼び出し元の認証と認可を行います。トークンを作成する方法は 2 つあります。

- サービスアカウント – サービスアカウントは、プロビジョニング、設定、レポート生成など、Grafana で自動処理の実行に使用されます。サービスアカウントには、専用のトークンを作成することができます。サービスアカウントは、バージョン 9.x と互換性のある Grafana ワークスペースで利用可能で、Grafana API と対話するアプリケーションを認証する主要な方法として API キーに代わるものとして使用される予定です。
- API キー – API キー (API トークンとも呼ばれます) は、API外部システムが Grafana HTTP API と対話するために使用できるランダムに生成された文字列です。API キーは Grafana ワークスペースのバージョン 8、9、および 10 で利用できますが、GrafanaLabs は将来のリリースでそれらが廃止予定であることを発表しています。

トピック

- サービスアカウントを使用した Grafana HTTP API の認証
- Grafana HTTP API での認証に対する API キーの使用

サービスアカウントを使用した Grafana HTTP API の認証

サービスアカウントを使用すると、ダッシュボードのプロビジョニング、設定、レポート生成などの自動ワークロードを Grafana で実行できます。Grafana コンソールまたは Amazon Managed Grafana API を使用して、Terraform などのアプリケーションを認証するサービスアカウントおよびトークンを作成します。

> **Note**
>
> サービスアカウントは Grafana 9.x 以降で利用可能です。これは、Grafana とやり取りするアプリケーションを認証する主な方法として、API キーに代わるものです。

サービスアカウントを作成する一般的なユースケースは、自動タスクまたはトリガーされたタスクに対してオペレーションを実行することです。サービスアカウントを使用すると、次のことを実行できます。

- Grafana で使用するシステムのアラートを定義する
- ユーザーとしてサインインせずに Grafana とやり取りする

> **Note**
>
> 各サービスアカウントは、請求対象のユーザーと見なされます。

サービスアカウントトークン

サービスアカウントトークンは、Grafana の HTTP API で認証するためのキーとして機能する、生成済みの文字列です。

サービスアカウントを作成するときに、1 つ以上のアクセストークンに関連付けることができます。サービスアクセストークンは、Grafana HTTP API にプログラムでアクセスするなど、API キーと同じ方法で使用できます。

同じサービスアカウントに対して複数のトークンを作成できます。次のケースでは複数作成する場合があります。

- 複数のアプリケーションが同じアクセス許可を使用するが、それらのアクションを個別に監査または管理したい。
- 漏洩したトークンをローテーションまたは置き換える必要がある。

サービスアカウントアクセストークンは、サービスアカウントのアクセス許可を継承します。

Amazon Managed Grafana には、一度に使用できるサービスアカウントトークン数のクォータがあります。これにはアクティブトークンと期限切れトークンが含まれます。トークンを削除してご使用のクォータから削除する必要があります。

サービスアカウントの利点

API キーに追加されたサービスアカウントの利点は次のとおりです。

- サービスアカウントは Grafana ユーザーと似ており、有効化/無効化、特定のアクセス許可を付与、および削除または無効化されるまでアクティブにできます。API キーが有効なのはその有効期限までです。
- サービスアカウントは複数のトークンに関連付けることができます。
- API キーとは異なり、サービスアカウントトークンは特定のユーザーに関連付けられません。つまり、Grafana ユーザーが削除されてもアプリケーションを認証できます。
- ユーザーにアクセス許可を付与するのと同じ方法で、サービスアカウントにアクセス許可を付与できます。

  権限の詳細については、「許可の使用」を参照してください。

サービスアカウントの作成

> **Note**
>
> サービスアカウントを作成するユーザーは、作成済みのサービスアカウントと、そのサービスアカウントに関連付けられたアクセス許可を、読み取り、更新、削除できます。

前提条件

サービスアカウントの作成および編集用アクセス許可があることを確認します。デフォルトでは、サービスアカウントの作成および編集には組織管理者ロールが必要です。権限の詳細については、「許可の使用」を参照してください。

サービスアカウントの作成方法

1. Amazon Managed Grafana ワークスペースにサインインし、左側のメニューから [管理] を選択します。
2. [サービスアカウント] を選択します。
3. [サービスアカウントの追加] を選択します。
4. [表示名] を入力します。
5. サービスアカウントに関連付けられた ID が表示名により判断されるため、表示名は一意である必要があります。

- サービスアカウントに名前を付けるときは、一貫した命名規則を使用することをお勧めします。一貫した命名規則は、将来的にサービスアカウントを維持してスケールするのに役立ちます。
- 表示名はいつでも変更できます。

6. [作成] を選択します。

> **Note**
>
> Amazon Managed AWS APIs を使用してサービスアカウントを作成することもできます。CreateWorkspaceServiceAccount を使用してプログラムでサービスアカウントを作成します。

サービスアカウントへのトークンの追加

サービスアカウントトークンは、Grafana の HTTP API で認証するときにパスワードの代替として機能する、生成済みのランダム文字列です。

前提条件

サービスアカウントの作成および編集用アクセス許可があることを確認します。デフォルトでは、サービスアカウントの作成および編集には組織管理者ロールが必要です。権限の詳細については、「許可の使用」を参照してください。

サービスアカウントへのトークンの追加方法

1. Grafana ワークスペースにサインインし、左側のメニューで [管理] を選択します。
2. [ユーザーとアクセス] メニューを展開します。
3. [サービスアカウント] を選択します。
4. トークンを追加するサービスアカウントを選択します。
5. [サービスアカウントトークンの追加] を選択します。
6. トークンの名前を入力します。
7. [有効期限の設定] を選択してトークンの有効期限を入力します。
   - 有効期限にはキーを有効にする期間を指定します。

- 有効期限は最大 30 日先まで設定できます。
- 有効期限が不明な場合は、数時間以内などの短期間でトークンの有効期限が切れるように設定することをお勧めします。これにより長期間有効なトークンに関連するリスクが制限されます。

8. [Generate token] を選択します。

> **Note**
>
> Amazon Managed Grafana AWS API を使用してサービスアカウントトークンを作成することもできます。CreateWorkspaceServiceAccountToken を使用して、プログラムでサービスアカウントトークンを作成します。

サービストークンの削除

サービストークンの使用が終了したら、トークンを削除してワークスペースから削除する必要があります。有効期限が切れても削除されていないトークンは、サービスアカウントトークンのクォータにカウントされます。

前提条件

サービスアカウントの作成および編集用アクセス許可があることを確認します。デフォルトでは、サービスアカウントの作成および編集には組織管理者ロールが必要です。権限の詳細については、「許可の使用」を参照してください。

サービスアカウントへのトークンを削除する方法

1. Grafana ワークスペースにサインインし、左側のメニューで [管理] を選択します。
2. [ユーザーとアクセス] メニューを展開します。
3. [サービスアカウント] を選択します。
4. トークンを削除するサービスアカウントを選択します。
5. トークンのリストで、削除するサービスアカウントトークンの隣の [x] が付いた赤いアイコンを選択します。
6. [削除] を選択します。

トークンが削除されます。

> **Note**
>
> Amazon Managed Grafana AWS API を使用してサービスアカウントトークンを削除することもできます。DeleteWorkspaceServiceAccountToken を使用して、サービスアカウントトークンをプログラムで削除します。

サービスアカウントへのロールの割り当て

Grafana サービスアカウントにロールを割り当てると、関連付けられたサービスアカウントトークンへのアクセスを制御できます。Grafana UI または API を使用して、サービスアカウントにロールを割り当てることができます。

前提条件

サービスアカウントの作成および編集用アクセス許可があることを確認します。デフォルトでは、サービスアカウントの作成および編集には組織管理者ロールが必要です。権限の詳細については、「許可の使用」を参照してください。

ロールをサービスアカウントに割り当てる方法

1. Grafana にサインインし、左側のメニューで [管理] を選択します。
2. [サービスアカウント] を選択します。
3. ロールを割り当てるサービスアカウントを選択します。別の方法として、リストビューでサービスアカウントを検索します。
4. 更新するロールピッカーを使用してロールを割り当てます。

Grafana HTTP API での認証に対する API キーの使用

Grafana API にアクセスする 1 つの方法は、API キーを使用することです。これは API トークンとも呼ばれます。PPI キーを作成するには、次のいずれかの手順に従います。API キーは、作成時に指定した期間の有効で、有効期間は最大 30 日です。

トピック

- ワークスペースでの Grafana API で使用する Grafana API キーの作成 (コンソール)
- AWS CLIを使用した Amazon Managed Grafana ワークスペース API キーの作成

> **Note**
>
> バージョン 9 以降では、API キーの代わりにサービスアカウントの使用をお勧めします。サービスアカウントは、Grafana API とやり取りするアプリケーションを認証するための主な方法として API に代わるものです。Grafana Labs は今後のリリースで API キーが削除されることを発表しました。

API キーを作成するとき、キーのロールを指定します。ロールによりキーのユーザーが持つ管理能力レベルが決定します。

次の表は、管理、エディタ、表示者の各ロールに付与されたアクセス許可を示しています。最初の表は全般的な組織のアクセス許可を示しています。このテーブルでは、[フル] はアクセス許可を表示、編集、追加、削除する機能を意味します。Explore 列には、ロールが Explore ビューを使用できるかどうかが表示されます。その他のアクセス許可列には、ロールにユーザー、チーム、プラグイン、組織の設定を管理するアクセス許可があるかどうかが示されます。

ロール	ダッシュボード	プレイリスト	フォルダ	Explore	データソース	その他のアクセス許可
表示者	ビュー	ビュー	いいえ	いいえ	いいえ	いいえ
[Editor] (エディタ)	フル	フル	フル	あり	いいえ	いいえ
Admin	フル	フル	フル	あり	フル	フル

次の表は、設定可能な追加のダッシュボードレベルおよびフォルダレベルのアクセス許可を示しています。これらは、管理、エディタ、表示者のロールとは異なります。

ロール	ダッシュボード	フォルダ	許可の変更
表示	ビュー	ビュー	いいえ
編集	作成、編集	ビュー	いいえ
Admin	作成、編集、削除	作成、編集、削除	あり

> **Note**
>
> より多い権限を持つ一般的なルールが存在する場合、より低い権限レベルの範囲にある権限は効果を持ちません。例えば、あるユーザーに組織 [エディタ] ロールを付与してから、そのユーザーにダッシュボードの [表示] アクセス許可のみを割り当てると、ユーザーは [エディタ] ロールを持つため完全な [編集] アクセス権限を持ち、より制限の厳しい [表示] アクセス許可は効果を持ちません。

ワークスペースでの Grafana API で使用する Grafana API キーの作成 (コンソール)

> **Note**
>
> Grafana バージョン 10 以降と互換性のある Amazon Managed Grafana ワークスペースでは、ワークスペースで API キーを作成する機能が削除されました。ワークスペースが Grafana バージョン 10 のワークスペースである場合は、 AWS CLI または API を介してのみ API キーを作成できます。
> Grafana Labs は今後のリリースで API キーが削除されることを発生しました。代わりにサービスアカウントの使用をお勧めします。

ワークスペースコンソールでの Grafana API で使用する Grafana API キーの作成方法

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. ページの左上隅にあるメニューアイコンを選択し、[すべてのワークスペース] を選択します。
3. Amazon Managed Grafana ワークスペースの名前を選択します。
4. ワークスペースの詳細ページで、[Grafana ワークスペース URL] の下に表示される URL を選択します。
5. Grafana コンソールのサイドメニューで、カーソルを [設定] (歯車) アイコンに置き、[API キー] を選択します。
6. [新しい API キー] を選択します。
7. キーの一意の名前を入力します。
8. [ロール] では、キーに付与するアクセスレベルを選択します。[管理] を選択すると、このキーを持つユーザーが、最も広範で強力な管理レベルで API を使用できるようになります。[エディタ] または [表示者] を選択して、それらのパワーレベルにキーのユーザーを制限します。詳細については、前のテーブルを参照してください。

9. [有効期限] では、キーの有効期間を指定します。最大は 30 日 (1 か月) です。数字と文字を入力します。有効な文字は、秒の [s]、分の [m]、時間の [h]、日の [d]、週の [w]、月の [M] です。例えば、[12h] は 12 時間、[1M] は 1 か月 (30 日) です。

   短時間 (数時間以下など) に設定することを強くお勧めします。これにより、長期間有効な API キーを持つ場合と比較してリスクが大幅に軽減されます。

10. [追加] を選択します。

11. (オプション) API キーの作成は、Terraform を使用した [API キーの作成] API で自動化できます。Terraform を使用した API キー作成の自動化の詳細については、[Terraform を使用した Grafana API キーの作成] を参照してください。

AWS CLIを使用した Amazon Managed Grafana ワークスペース API キーの作成

AWS CLIを使用した Amazon Managed Grafana ワークスペース API キーの作成方法

次の例では、*key_name*、*key_role*、*seconds_to_live*、*Workspace_id* を独自の情報に置き換えます。key-name、key-role、secon-to-live の形式については、 API ガイド「https://docs.aws.amazon.com/grafana/latest/APIReference/API_CreateWorkspaceApiKey.html」を参照してください。

```
aws grafana create-workspace-api-key --key-name "key_name" --key-role "key_role" --
seconds-to-live seconds_to_live --workspace-id "workspace_id"
```

CLI レスポンスの例を次に示します。

create-workspace-api-key 出力の例

ワークスペースの *Workspace_id* は、次のコマンドを実行して確認できます。

```
aws grafana list-workspaces
```

アラート API

> **Note**
>
> このセクションは従来のアラートにのみ適用されます。詳細については、「Grafana アラート」を参照してください。

プリファレンス API を使用して、従来のダッシュボードアラートとその状態に関する情報を取得します。ただし、この API を使用してもアラートは変更できません。新しいアラートを作成したり、それを変更するためには、アラートを含むダッシュボード JSON を更新する必要があります。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

アラートの取得

```
GET /api/alerts
```

リクエストの例

```
GET /api/alerts HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリ文字列パラメータ:

これらのパラメータはクエリ文字列パラメータとして使用されます。例: `/api/alerts?dashboardId=1`

- dashboardId — 指定されたダッシュボード値のアラートにレスポンスを制限します。ダッシュボードは複数指定できます。例: `dashboardId=23&dashboardId=35`

- panelId — ダッシュボード上の指定パネルのアラートにレスポンスを制限します。
- query — この値のような名前のアラートにレスポンスを制限します。
- state — ALL、alerting、ok、no_data、paused、pending のアラート状態が 1 つ以上あるアラートを返します。複数の状態を指定するには、次の形式を使用します: ? state=paused&state=alerting
- [制限] — X アラート数にレスポンスを制限します。
- folderId — 指定フォルダ内のダッシュボードのアラートにレスポンスを制限します。フォルダは複数指定できます。例: folderId=23&folderId=35
- dashboardQuery — この値のようなダッシュボード名を持つアラートにレスポンスを制限します。
- dashboardTag — 指定タグを持つダッシュボードのアラートにレスポンスを制限します。複数のタグで「AND」のフィルタリングを実行するには、タグパラメータを複数回指定します。例えば、dashboardTag=tag1&dashboardTag=tag2。これらはタグではなく Grafana AWS タグであることに注意してください。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "dashboardId": 1,
    "dashboardUId": "ABcdEFghij"
    "dashboardSlug": "sensors",
    "panelId": 1,
    "name": "fire place sensor",
    "state": "alerting",
    "newStateDate": "2018-05-14T05:55:20+02:00",
    "evalDate": "0001-01-01T00:00:00Z",
    "evalData": null,
    "executionError": "",
    "url": "http://grafana.com/dashboard/db/sensors"
  }
]
```

ID によるアラート取得

```
GET /api/alerts/:id
```

リクエストの例

```
GET /api/alerts/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "dashboardId": 1,
  "dashboardUId": "ABcdEFghij"
  "dashboardSlug": "sensors",
  "panelId": 1,
  "name": "fire place sensor",
  "state": "alerting",
  "message": "Someone is trying to break in through the fire place",
  "newStateDate": "2018-05-14T05:55:20+02:00",
  "evalDate": "0001-01-01T00:00:00Z",
  "evalData": "evalMatches": [
    {
      "metric": "movement",
      "tags": {
        "name": "fireplace_chimney"
      },
      "value": 98.765
    }
  ],
  "executionError": "",
  "url": "http://grafana.com/dashboard/db/sensors"
}
```

> **Important**
>
> evalMatches データはアラート状態が変更された場合にのみデータベースにキャッシュされます。1 つのサーバーのデータによって最初にアラートがトリガーされ、そのサーバーがアラート状態から離れるのが確認される前に 2 番目のサーバーもアラートをトリガーする状態に入ると、2 番目のサーバーは evalMatches データに表示されません。

ID によるアラート一時停止

```
POST /api/alerts/:id/pause
```

リクエストの例

```
POST /api/alerts/1/pause HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "paused": true
}
```

:id クエリパラメータは、一時停止または一時停止解除するアラートの ID です。paused が true の場合はアラートを一時停止し、false の場合はアラートの一時停止を解除します。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "alertId": 1,
  "state":   "Paused",
  "message": "alert paused"
}
```

アラート通知チャネル API

アラート通知チャネル API を使用して通知チャネルを作成、更新、削除、取得します。

通知チャネルの識別子 (id) は自動増分数値で、ワークスペースごとに一意です。通知チャネルの一意の識別子 (uid) は、複数のワークスペース間のフォルダを一意に識別するために使用できます。通知チャネルを作成するときに指定しない場合は、自動的に生成されます。uid を使用すると、通知チャネルにアクセスしたり、複数の Amazon Managed Grafana ワークスペース間の通知チャネルを同期したりするための、一貫した URL を設定できます。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

すべての通知チャネルの取得

認証されたユーザーが表示アクセス許可を持つ、すべての通知チャネルを返します。

```
GET /api/alert-notifications
```

リクエストの例

```
GET /api/alert-notifications HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "uid": "sns-uid",
    "name": "test",
    "type": "sns",
    "isDefault": false,
    "sendReminder": false,
```

```
    "disableResolveMessage": false,
    "frequency": "",
    "created": "2023-09-08T19:57:56Z",
    "updated": "2023-09-08T19:57:56Z",
    "settings": {
      "authProvider": "default",
      "autoResolve": true,
      "httpMethod": "POST",
      "messageFormat": "json",
      "severity": "critical",
      "topic": "<SNS-TOPIC-ARN>",
      "uploadImage": false
    },
    "secureFields": {}
  }
]
```

すべての通知チャネルの取得 (ルックアップ)

すべての通知チャネルを返しますが、詳細情報が少なくなります。認証されたユーザーがアクセスでき、主にアラートルールを設定するときに、Grafana ワークスペースコンソール UI でアラート通知チャネルを提供するために使用します。

```
GET /api/alert-notifications/lookup
```

リクエストの例

```
GET /api/alert-notifications/lookup HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "uid": "sns-uid",
```

```
    "name": "test",
    "type": "sns",
    "isDefault": false
  },
  {
    "id": 2,
    "uid": "slack-uid",
    "name": "Slack",
    "type": "slack",
    "isDefault": false
  }
]
```

UID によるすべての通知チャネルの取得

```
GET /api/alert-notifications/uid/:uid
```

リクエストの例

```
GET /api/alert-notifications/uid/sns-uid HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "sns-uid",
  "name": "test",
  "type": "sns",
  "isDefault": false,
  "sendReminder": false,
  "disableResolveMessage": false,
  "frequency": "",
  "created": "2023-09-08T19:57:56Z",
  "updated": "2023-09-08T19:57:56Z",
  "settings": {
    "authProvider": "default",
```

```
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  },
  "secureFields": {}
}
```

Id によるすべての通知チャネルの取得

```
GET /api/alert-notifications/:id
```

リクエストの例

```
GET /api/alert-notifications/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "sns-uid",
  "name": "test",
  "type": "sns",
  "isDefault": false,
  "sendReminder": false,
  "disableResolveMessage": false,
  "frequency": "",
  "created": "2023-09-08T19:57:56Z",
  "updated": "2023-09-08T19:57:56Z",
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
```

```
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  },
  "secureFields": {}
}
```

通知チャネルの作成

Amazon Managed Grafana でサポートされている通知チャネルを確認するには、「コンタクトポイント (通知先) の使用」でサポート対象の通知方法のリストを参照してください。

```
POST /api/alert-notifications
```

リクエストの例

```
POST /api/alert-notifications HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890

{
  "uid": "new-sns-uid", // optional
  "name": "sns alert notification",  //Required
  "type":  "sns", //Required
  "isDefault": false,
  "sendReminder": false,
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  }
}
```

レスポンスの例

```
HTTP/1.1 200
```

```
Content-Type: application/json

{
  "id": 1,
  "uid": "new-sns-uid",
  "name": "sns alert notification",
  "type": "sns",
  "isDefault": false,
  "sendReminder": false,
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  },
  "created": "2018-04-23T14:44:09+02:00",
  "updated": "2018-08-20T15:47:49+02:00"
}
```

UID による通知チャネルの更新

```
PUT /api/alert-notifications/uid/:uid
```

リクエストの例

```
PUT /api/alert-notifications/uid/sns-uid HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890

{
  "uid": "sns-uid", // optional
  "name": "sns alert notification",  //Required
  "type":  "sns", //Required
  "isDefault": false,
  "sendReminder": true,
  "frequency": "15m",
  "settings": {
    "authProvider": "default",
```

```
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  }
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "sns-uid",
  "name": "sns alert notification",
  "type": "sns",
  "isDefault": false,
  "sendReminder": true,
  "frequency": "15m",
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  },
  "created": "2017-01-01 12:34",
  "updated": "2017-01-01 12:34"
}
```

Id による通知チャネルの更新

```
PUT /api/alert-notifications/:id
```

リクエストの例

```
PUT /api/alert-notifications/1 HTTP/1.1
```

```
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890

{
  "id": 1,
  "uid": "sns-uid", // optional
  "name": "sns alert notification",  //Required
  "type":  "sns", //Required
  "isDefault": false,
  "sendReminder": true,
  "frequency": "15m",
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  }
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "sns-uid",
  "name": "sns alert notification",
  "type": "sns",
  "isDefault": false,
  "sendReminder": true,
  "frequency": "15m",
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
```

```
    "uploadImage": false
  },
  "created": "2017-01-01 12:34",
  "updated": "2017-01-01 12:34"
}
```

UID による通知チャネルの削除

```
DELETE /api/alert-notifications/uid/:uid
```

リクエストの例

```
DELETE /api/alert-notifications/uid/sns-uid HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "message": "Notification deleted"
}
```

Id による通知チャネルの削除

```
DELETE /api/alert-notifications/:id
```

リクエストの例

```
DELETE /api/alert-notifications/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "message": "Notification deleted"
}
```

テスト通知チャネル

指定された通知チャネルタイプおよび設定のテスト通知メッセージを送信します。

```
POST /api/alert-notifications/test
```

リクエストの例

```
POST /api/alert-notifications/test HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer 1234abcd567exampleToken890

{
  "type":  "sns",
  "settings": {
    "authProvider": "default",
    "autoResolve": true,
    "httpMethod": "POST",
    "messageFormat": "json",
    "severity": "critical",
    "topic": "<SNS-TOPIC-ARN>",
    "uploadImage": false
  }
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "message": "Test notification sent"
}
```

注釈 API

注釈 API を使用して、Amazon Managed Grafana ワークスペースの注釈を作成、更新、削除、操作します。

注釈は、ワークスペースの Grafana データベース (sqlite、mysql、postgres) に保存されます。注釈は、注釈データソースを設定することにより任意のダッシュボードに表示できる、グローバル注釈にできます。注釈はタグによりフィルタリングされます。または、ダッシュボード上のパネルに紐付けして、そのパネルにのみ表示できます。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

注釈の検索

```
GET /api/annotations?from=1506676478816&to=1507281278816&tags=tag1&tags=tag2&limit=100
```

リクエストの例

```
GET /api/annotations?from=1506676478816&to=1507281278816&tags=tag1&tags=tag2&limit=100
 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリパラメータ:

- [開始] — (オプション) ミリ秒単位のエポック日時。
- to — (オプション) ミリ秒単位のエポック日時。
- [制限] — (オプション) 返される最大結果数。デフォルトは 100 です。
- alertid — (オプション) 指定アラートの注釈を検索します。
- dashboardId — (オプション) 指定ダッシュボードに限定されている注釈を検索します。

- panelId — (オプション) 指定パネルに限定されている注釈を検索します。
- userId — (オプション) 指定ユーザーが作成した注釈を検索します。
- [タイプ] — (オプション) アラートまたはユーザーが作成した注釈を返すために指定します。有効な値は、alert、および annotation です。
- [タグ] — (オプション) グローバル注釈をフィルタリングするために使用します。グローバル注釈は、ダッシュボードやパネルに明確に接続されていない注釈データソースからの注釈です。複数のタグで「AND」のフィルタリングを実行するには、タグパラメータを複数回指定します。例えば、tags=tag1&tags=tag2。これらは Grafana タグであり、 AWS タグではありません。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
[
    {
        "id": 1124,
        "alertId": 0,
        "dashboardId": 468,
        "panelId": 2,
        "userId": 1,
        "userName": "",
        "newState": "",
        "prevState": "",
        "time": 1507266395000,
        "timeEnd": 1507266395000,
        "text": "test",
        "metric": "",
        "tags": [
            "tag1",
            "tag2"
        ],
        "data": {}
    },
    {
        "id": 1123,
        "alertId": 0,
        "dashboardId": 468,
        "panelId": 2,
        "userId": 1,
        "userName": "",
        "newState": "",
```

```
        "prevState": "",
        "time": 1507265111000,
        "text": "test",
        "metric": "",
        "tags": [
            "tag1",
            "tag2"
        ],
        "data": {}
    }
]
```

注釈の作成

```
POST /api/annotations
```

ワークスペースの Grafana データベースで注釈を作成します。dashboardId および panelId フィールドはオプションです。これらのフィールドを指定しない場合、グローバル注釈が作成され、Grafana 注釈データソースを追加するダッシュボードでクエリを実行できます。リージョン注釈を作成するときは、必ず timeEnd プロパティを含めてください。

time と timeEnd の形式は、ミリ秒単位のエポック番号にする必要があります。

リクエストの例

```
POST /api/annotations HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "dashboardId":468,
  "panelId":1,
  "time":1507037197339,
  "timeEnd":1507180805056,
  "tags":["tag1","tag2"],
  "text":"Annotation Description"
}
```

レスポンスの例

```
HTTP/1.1 200
```

```
Content-Type: application/json

{
    "message":"Annotation added",
    "id": 1,
}
```

グラファイト形式の注釈の作成

```
POST /api/annotations/graphite
```

Graphite 互換のイベント形式を使用して注釈を作成します。when および data フィールドはオプションです。when を指定しない場合は、現在の時刻が注釈のタイムスタンプとして使用されます。tags フィールドは、Graphite 0.10.0 以前の形式 (スペースで区切られた複数のタグを含む文字列) にすることもできます。

リクエストの例

```
POST /api/annotations/graphite HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "what": "Event - deploy",
  "tags": ["deploy", "production"],
  "when": 1467844481,
  "data": "deploy of master branch happened at Wed Jul 6 22:34:41 UTC 2016"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
    "message":"Graphite annotation added",
    "id": 1
}
```

注釈の更新

```
PUT /api/annotations/:id
```

指定 id と一致する注釈のすべてのプロパティを更新します。特定のプロパティのみを更新するためには、パッチ注釈オペレーションを使用します。

リクエストの例

```
PUT /api/annotations/1141 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
Content-Type: application/json

{
  "time":1507037197339,
  "timeEnd":1507180805056,
  "text":"Annotation Description",
  "tags":["tag3","tag4","tag5"]
}
```

レスポンスの例:

```
HTTP/1.1 200
Content-Type: application/json

{
    "message":"Annotation updated"
}
```

パッチ注釈

```
PATCH /api/annotations/:id
```

指定 id と一致する 1 つ以上の注釈のプロパティを更新します。このオペレーションは現在、text、tags、time、および timeEnd プロパティの更新をサポートしています。

リクエストの例:

```
PATCH /api/annotations/1145 HTTP/1.1
```

```
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
Content-Type: application/json

{
   "text":"New Annotation Description",
   "tags":["tag6","tag7","tag8"]
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
    "message":"Annotation patched"
}
```

Id による注釈の削除

```
DELETE /api/annotations/:id
```

指定 Id と一致する注釈を削除します。

リクエストの例

```
DELETE /api/annotations/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
    "message":"Annotation deleted"
}
```

認証 API

認証 API を使用して Amazon Managed Grafana ワークスペースの認証キーを操作します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

API キーの取得

```
GET /api/auth/keys
```

リクエストの例

```
GET /api/auth/keys HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリパラメーター:

- includeExpired — (オプション) 返される結果に期限切れキーを含めるかどうかを指定する、ブールパラメータです。デフォルトは `false` です。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {"id": 3,"name": "API","role": "Admin"},
  {"id": 1,"name": "TestAdmin","role": "Admin","expiration":
 "2019-06-26T10:52:03+03:00"}
]
```

API キーの作成

```
POST /api/auth/keys
```

リクエストの例

```
POST /api/auth/keys HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "mykey",
  "role": "Admin",
  "secondsToLive": 86400
}
```

JSON ボディスキーマ:

- [名前] — キーの名前です。
- role — キーのアクセスレベル (Grafana ロール) を設定します。有効な値は Admin、Editor、または Viewer です。
- secondsToLive — キーの有効期限が切れるまでの合計時間を設定します。2592000 (30 日) 以下にする必要があります。

レスポンスの例

```
{"name":"mykey","key":"eyJrIjoiWHZiSWd3NzdCYUZnNUtibE9obUpESmE3bzJYNDRIc0UiLCJuIjoibXlrZXkiLCJp
```

エラーステータス:

- 400 — secondsToLive が 2592000 より大きい
- 500 — データベースにキーを保存できませんでした。

API キーの削除

```
DELETE /api/auth/keys/:id
```

リクエストの例

```
DELETE /api/auth/keys/3 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"API key deleted"}
```

ダッシュボード API

ダッシュボード API を使用して、Amazon Managed Grafana ワークスペースのダッシュボードを作成、更新、削除、操作します。

ダッシュボードの識別子 (id) は自動増分数値で、ワークスペースごとに一意です。ダッシュボードの一意の識別子 (uid) は、複数の Amazon Managed Grafana ワークスペース間のダッシュボードを一意に識別するために使用できます。ダッシュボードの作成時に指定しない場合は自動的に生成されます。uid を使用すると、ダッシュボードにアクセスしたり、複数のワークスペース間でダッシュボードを同期したりするための、一貫した URL を設定できます。uid を使用すると、ダッシュボードのタイトルを変更しても、そのダッシュボードにブックマークされたリンクが破損することはありません。

uid の最大長は 40 文字までです。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

ダッシュボードの作成/更新

```
POST /api/dashboards/db
```

新しいダッシュボードを作成または既存のダッシュボードを更新します。

新しいダッシュボードを作成するためのリクエスト例

```
POST /api/dashboards/db HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "dashboard": {
    "id": null,
    "uid": null,
    "title": "Production Overview",
    "tags": [ "templated" ],
    "timezone": "browser",
    "schemaVersion": 16,
    "version": 0,
    "refresh": "25s"
  },
  "folderId": 0,
  "folderUid": "l3KqBxCMz",
  "message": "Made changes to xyz",
  "overwrite": false
}
```

JSON ボディスキーマ:

- dashboard — 完全なダッシュボードモデルです。新しいダッシュボードの作成には null を使用します。
- dashboard.id — 新しいダッシュボードの作成には null を使用します。
- dashboard.uid — これを使用して新しいダッシュボードを作成するときの、一意のオプション識別子です。null の場合は新しい uid が生成されます。
- folderid — ダッシュボードが保存されるフォルダの ID。
- folderUid — ダッシュボードが保存されるフォルダの Uid。`folderid` の値を上書きします

- overwrite — `true` を指定すると、既存のダッシュボードが、新しいバージョン、フォルダ内の同じダッシュボードタイトル、または同じダッシュボード uid で上書きされます。
- message — バージョン履歴のコミットメッセージを設定します。
- [更新] — ダッシュボード更新間隔を設定します。これが最小更新間隔より小さい場合は、無視されて最小更新間隔が使用されます。

ダッシュボードパネルのアラートルールを追加または更新するには、`dashboard.panels.alert` ブロックを宣言します。

ダッシュボードアラートルールを更新するリクエスト例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 78

{
 "dashboard":  {
        "id": 104,
        "panels": [
            {
                "alert": {
                    "alertRuleTags": {},
                    "conditions": [
                        {
                            "evaluator": {
                                "params": [
                                    25
                                ],
                                "type": "gt"
                            },
                            "operator": {
                                "type": "and"
                            },
                            "query": {
                                "params": [
                                    "A",
                                    "5m",
                                    "now"
                                ]
                            },
                            "reducer": {
```

```
                        "params": [],
                        "type": "avg"
                    },
                    "type": "query"
                }
            ],
            "executionErrorState": "alerting",
            "for": "5m",
            "frequency": "1m",
            "handler": 1,
            "name": "Panel Title alert",
            "noDataState": "no_data",
            "notifications": []
        },
        "aliasColors": {},
        "bars": false,
        "dashLength": 10,
        "dashes": false,
        "datasource": null,
        "fieldConfig": {
            "defaults": {
                "custom": {}
            },
            "overrides": []
        },
        "fill": 1,
        "fillGradient": 0,
        "gridPos": {
            "h": 9,
            "w": 12,
            "x": 0,
            "y": 0
        },
        "hiddenSeries": false,
        "id": 2,
        "legend": {
            "avg": false,
            "current": false,
            "max": false,
            "min": false,
            "show": true,
            "total": false,
            "values": false
        },
```

```
                "lines": true,
                "linewidth": 1,
                "nullPointMode": "null",
                "options": {
                    "dataLinks": []
                },
                "percentage": false,
                "pointradius": 2,
                "points": false,
                "renderer": "flot",
                "seriesOverrides": [],
                "spaceLength": 10,
                "stack": false,
                "steppedLine": false,
                "targets": [
                    {
                        "refId": "A",
                        "scenarioId": "random_walk"
                    }
                ],
                "thresholds": [
                    {
                        "colorMode": "critical",
                        "fill": true,
                        "line": true,
                        "op": "gt",
                        "value": 50
                    }
                ],
                "timeFrom": null,
                "timeRegions": [],
                "timeShift": null,
                "title": "Panel Title",
                "tooltip": {
                    "shared": true,
                    "sort": 0,
                    "value_type": "individual"
                },
                "type": "graph",
                "xaxis": {
                    "buckets": null,
                    "mode": "time",
                    "name": null,
                    "show": true,
```

```
                    "values": []
                },
                "yaxes": [
                    {
                        "format": "short",
                        "label": null,
                        "logBase": 1,
                        "max": null,
                        "min": null,
                        "show": true
                    },
                    {
                        "format": "short",
                        "label": null,
                        "logBase": 1,
                        "max": null,
                        "min": null,
                        "show": true
                    }
                ],
                "yaxis": {
                    "align": false,
                    "alignLevel": null
                }
            }
        ],
        "title": "Update alert rule via API",
        "uid": "dHEquNzGz",
        "version": 1
    }
}
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 78

{
  "id":      1,
  "uid":     "cIBgcSjkk",
  "url":     "/d/cIBgcSjkk/production-overview",
  "status":  "success",
```

```
  "version": 1,
  "slug":    "production-overview" //deprecated in Grafana v5.0
}
```

ステータスコード:

- 200 — 作成済み
- 400 — 無効な JSON、無効なフィールド、欠落したフィールドなどのエラー
- 401 — 未許可
- 403 — アクセス拒否
- 412 — 前提条件失敗

412 ステータスコードはダッシュボードを作成できない理由の説明に使用されます。

- ダッシュボードが他のユーザーによって変更された `status=version-mismatch`
- フォルダに同じ名前のダッシュボードが既に存在する `status=name-exists`
- 同じ uid のダッシュボードが既に存在する `status=name-exists`
- ダッシュボードはプラグイン `plugin  title` `status=plugin-dashboard` に属している

レスポンス本文には次のプロパティがあります。別のダッシュボードに同じタイトルがある場合、`status` 値は `name-exists` です。

```
HTTP/1.1 412 Precondition Failed
Content-Type: application/json; charset=UTF-8
Content-Length: 97

{
  "message": "The dashboard has been changed by someone else",
  "status": "version-mismatch"
}
```

uid によるダッシュボードの取得

```
GET /api/dashboards/uid/:uid
```

uid と一致するダッシュボードを返します。返されるメタデータには、ダッシュボードを含むフォルダの UID の情報が含まれている場合があります。

リクエストの例

```
GET /api/dashboards/uid/cIBgcSjkk HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "dashboard": {
    "id": 1,
    "uid": "cIBgcSjkk",
    "title": "Production Overview",
    "tags": [ "templated" ],
    "timezone": "browser",
    "schemaVersion": 16,
    "version": 0
  },
  "meta": {
    "isStarred": false,
    "url": "/d/cIBgcSjkk/production-overview",
    "folderId": 2,
    "folderUid": "l3KqBxCMz",
    "slug": "production-overview" //deprecated in Grafana v5.0
  }
}
```

ステータスコード:

- 200 — 検出済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — 未検出

uid によるダッシュボードの削除

```
DELETE /api/dashboards/uid/:uid
```

uid と一致するダッシュボードを削除します。

リクエストの例

```
DELETE /api/dashboards/uid/cIBgcSjkk HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "title": "Production Overview",
  "message": "Dashboard Production Overview deleted",
  "id": 2
}
```

ステータスコード:

- 200 — 削除済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — 未検出

ホームダッシュボードの取得

```
GET /api/dashboards/home
```

ホームダッシュボードを返します。

リクエストの例

```
GET /api/dashboards/home HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "dashboard": {
    "editable":false,
    "hideControls":true,
    "nav":[
      {
        "enable":false,
        "type":"timepicker"
      }
    ],
    "style":"dark",
    "tags":[],
    "templating":{
      "list":[
      ]
    },
    "time":{
    },
    "timezone":"browser",
    "title":"Home",
    "version":5
  },
  "meta": {
    "isHome":true,
    "canSave":false,
    "canEdit":false,
    "canStar":false,
    "url":"",
    "expires":"0001-01-01T00:00:00Z",
    "created":"0001-01-01T00:00:00Z"
  }
}
```

ダッシュボードタグの取得

```
GET /api/dashboards/tags
```

ダッシュボードのタグすべてを返します。

リクエストの例

```
GET /api/dashboards/tags HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "term":"tag1",
    "count":1
  },
  {
    "term":"tag2",
    "count":4
  }
]
```

ダッシュボードアクセス許可 API

Dashboard Permissions API を使用してダッシュボードのアクセス許可を更新または取得します。

dashboardId=-1 のアクセス許可は、ビューワーロールとエディタロールを持つユーザーのデフォルトのアクセス許可です。アクセス許可は、ユーザー、チーム、またはロール (ビューワーまたはエディタ) に設定できます。管理者にアクセス許可を設定することはできません。管理者は常にすべてにアクセスできます。

permission フィールドのアクセス許可レベルは、次のとおりです。

- 1 = 表示
- 2 = 編集
- 4 = 管理者

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの Authorization フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

ダッシュボードのアクセス許可の取得

```
GET /api/dashboards/id/:dashboardId/permissions
```

指定された dashboardId を使用して、ダッシュボードの既存のアクセス許可をすべて取得します。

リクエストの例

```
GET /api/dashboards/id/1/permissions HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 551

[
  {
    "id": 1,
    "dashboardId": -1,
    "created": "2017-06-20T02:00:00+02:00",
    "updated": "2017-06-20T02:00:00+02:00",
```

```
    "userId": 0,
    "userLogin": "",
    "userEmail": "",
    "teamId": 0,
    "team": "",
    "role": "Viewer",
    "permission": 1,
    "permissionName": "View",
    "uid": "",
    "title": "",
    "slug": "",
    "isFolder": false,
    "url": ""
  },
  {
    "id": 2,
    "dashboardId": -1,
    "created": "2017-06-20T02:00:00+02:00",
    "updated": "2017-06-20T02:00:00+02:00",
    "userId": 0,
    "userLogin": "",
    "userEmail": "",
    "teamId": 0,
    "team": "",
    "role": "Editor",
    "permission": 2,
    "permissionName": "Edit",
    "uid": "",
    "title": "",
    "slug": "",
    "isFolder": false,
    "url": ""
  }
]
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 403 — アクセス拒否
- 404 — ダッシュボード未検出

ダッシュボードのアクセス許可の更新

```
POST /api/dashboards/id/:dashboardId/permissions
```

ダッシュボードのアクセス許可を更新します。このオペレーションでは、既存のアクセス許可がリクエストに含まれていない場合、それらのアクセス許可を削除します。

リクエストの例

```
POST /api/dashboards/id/1/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "items": [
    {
      "role": "Viewer",
      "permission": 1
    },
    {
      "role": "Editor",
      "permission": 2
    },
    {
      "teamId": 1,
      "permission": 1
    },
    {
      "userId": 11,
      "permission": 4
    }
  ]
}
```

JSON ボディスキーマ:

- items — 追加または更新するアクセス許可の項目です。リストから省略された既存の項目が削除されます。

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Dashboard permissions updated"}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — ダッシュボード未検出

ダッシュボードバージョン API

ダッシュボードバージョン API を使用してダッシュボードバージョンを取得し、指定バージョンにダッシュボードを復元します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

すべてのダッシュボードバージョンの取得

```
GET /api/dashboards/id/:dashboardId/versions
```

指定された `dashboardId` を使用して、ダッシュボードの既存のダッシュボードバージョンすべてを取得します。

クエリパラメータ:

- [制限] — 返される結果の最大数。
- [開始] — クエリを返すときから開始するバージョン。

リクエストの例

```
GET /api/dashboards/id/1/versions?limit=2?start=0 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 428

[
  {
    "id": 2,
    "dashboardId": 1,
    "parentVersion": 1,
    "restoredFrom": 0,
    "version": 2,
    "created": "2017-06-08T17:24:33-04:00",
    "createdBy": "admin",
    "message": "Updated panel title"
  },
  {
    "id": 1,
    "dashboardId": 1,
    "parentVersion": 0,
    "restoredFrom": 0,
    "version": 1,
    "created": "2017-06-08T17:23:33-04:00",
    "createdBy": "admin",
    "message": "Initial save"
  }
]
```

ステータスコード:

- 200 — OK
- 400 — エラー
- 401 — 未許可

- 404 — ダッシュボードバージョン未検出

ダッシュボードバージョンの取得

```
GET /api/dashboards/id/:dashboardId/versions/:id
```

特定の dashboardId を持つダッシュボードの、特定の id を持つダッシュボードバージョンを取得します。

リクエストの例

```
GET /api/dashboards/id/1/versions/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 1300

{
  "id": 1,
  "dashboardId": 1,
  "parentVersion": 0,
  "restoredFrom": 0,
  "version": 1,
  "created": "2017-04-26T17:18:38-04:00",
  "message": "Initial save",
  "data": {
    "annotations": {
      "list": [

      ]
    },
    "editable": true,
    "gnetId": null,
    "graphTooltip": 0,
    "hideControls": false,
    "id": 1,
```

```
    "links": [

    ],
    "rows": [
      {
        "collapse": false,
        "height": "250px",
        "panels": [

        ],
        "repeat": null,
        "repeatIteration": null,
        "repeatRowId": null,
        "showTitle": false,
        "title": "Dashboard Row",
        "titleSize": "h6"
      }
    ],
    "schemaVersion": 14,
    "style": "dark",
    "tags": [

    ],
    "templating": {
      "list": [

      ]
    },
    "time": {
      "from": "now-6h",
      "to": "now"
    },
    "timepicker": {
      "refresh_intervals": [
        "5s",
        "10s",
        "30s",
        "1m",
        "5m",
        "15m",
        "30m",
        "1h",
        "2h",
        "1d"
```

```
      ],
      "time_options": [
        "5m",
        "15m",
        "1h",
        "6h",
        "12h",
        "24h",
        "2d",
        "7d",
        "30d"
      ]
    },
    "timezone": "browser",
    "title": "test",
    "version": 1
  },
  "createdBy": "admin"
}
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 404 — ダッシュボードバージョン未検出

ダッシュボードの復元

```
POST /api/dashboards/id/:dashboardId/restore
```

指定したダッシュボードバージョンにダッシュボードを復元します。

リクエストの例

```
POST /api/dashboards/id/1/restore
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
```

```
  "version": 1
}
```

JSON ボディスキーマ:

- version — 復元するダッシュボードバージョン。

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 67

{
  "slug": "my-dashboard",
  "status": "success",
  "version": 3
}
```

JSON レスポンスボディスキーマ:

- slug — ダッシュボードのタイトルの URL フレンドリーなスラグ。
- status — 復元が成功したかどうか。
- version — 復元後の新しいダッシュボードバージョン。

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 404 — ダッシュボードまたはダッシュボードバージョンが未検出
- 500 — 内部サーバーエラー (データベースからダッシュボードタグの取得に関する問題を示します)

エラー応答のサンプル:

```
HTTP/1.1 404 Not Found
Content-Type: application/json; charset=UTF-8
```

```
Content-Length: 46

{
  "message": "Dashboard version not found"
}
```

JSON レスポンスボディスキーマ:

- message — 障害の理由を説明するメッセージ。

ダッシュボードバージョンの比較

```
POST /api/dashboards/calculate-diff
```

2 つのダッシュボードバージョンは、それらの JSON 差分を計算して比較します。

リクエストの例

```
POST /api/dashboards/calculate-diff HTTP/1.1
Accept: text/html
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "base": {
    "dashboardId": 1,
    "version": 1
  },
  "new": {
    "dashboardId": 1,
    "version": 2
  },
  "diffType": "json"
}
```

JSON ボディスキーマ:

- base — 基本ダッシュボードバージョンを表すオブジェクト。
- [新規] — 新しいダッシュボードバージョンを表すオブジェクト。
- difftype — 返される差分のタイプ。有効な値は、json および basic です。

レスポンス例 (JSON 差分)

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8

<p id="l1" class="diff-line diff-json-same">
  <!-- Diff omitted -->
</p>
```

レスポンスは差分のテキスト表現で、GitHub や GitLab などのサイトで見られる差分と同様に、JSON で記述したダッシュボード値です。

ステータスコード:

- 200 — OK
- 200 — 不正なリクエスト、無効な JSON の送信
- 401 — 未許可
- 404 — 未検出

レスポンス例 (基本差分)

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8

<div class="diff-group">
  <!-- Diff omitted -->
</div>
```

レスポンスは 2 つの JSON オブジェクト間の差分から派生した、変更の概要です。

ステータスコード:

- 200 — OK
- 200 — 不正なリクエスト、無効な JSON の送信
- 401 — 未許可
- 404 — 未検出

データソース API

Data Source API を使用して、Amazon Managed Grafana ワークスペースのデータソースを作成、更新、削除、操作します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

すべてのデータソースの取得

```
GET /api/datasources
```

リクエストの例

```
GET /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
   {
     "id": 1,
     "orgId": 1,
     "uid": "H8joYFVGz"
     "name": "datasource_elastic",
     "type": "elasticsearch",
     "typeLogoUrl": "public/app/plugins/datasource/elasticsearch/img/
elasticsearch.svg",
     "access": "proxy",
     "url": "http://mydatasource.com",
```

```
    "password": "",
    "user": "",
    "database": "grafana-dash",
    "basicAuth": false,
    "isDefault": false,
    "jsonData": {
        "esVersion": 5,
        "logLevelField": "",
        "logMessageField": "",
        "maxConcurrentShardRequests": 256,
        "timeField": "@timestamp"
    },
    "readOnly": false
  }
]
```

id による単一データソースの取得

```
GET /api/datasources/:datasourceId
```

リクエストの例

```
GET /api/datasources/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "kLtEtcRGk",
  "orgId": 1,
  "name": "test_datasource",
  "type": "graphite",
  "typeLogoUrl": "",
  "access": "proxy",
  "url": "http://mydatasource.com",
  "password": "",
```

```
  "user": "",
  "database": "",
  "basicAuth": false,
  "basicAuthUser": "",
  "basicAuthPassword": "",
  "withCredentials": false,
  "isDefault": false,
  "jsonData": {
    "graphiteType": "default",
    "graphiteVersion": "1.1"
  },
  "secureJsonFields": {},
  "version": 1,
  "readOnly": false
}
```

UID による単一データソースの取得

```
GET /api/datasources/uid/:uid
```

リクエストの例

```
GET /api/datasources/uid/kLtEtcRGk HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "kLtEtcRGk",
  "orgId": 1,
  "name": "test_datasource",
  "type": "graphite",
  "typeLogoUrl": "",
  "access": "proxy",
  "url": "http://mydatasource.com",
  "password": "",
```

```
  "user": "",
  "database": "",
  "basicAuth": false,
  "basicAuthUser": "",
  "basicAuthPassword": "",
  "withCredentials": false,
  "isDefault": false,
  "jsonData": {
    "graphiteType": "default",
    "graphiteVersion": "1.1"
  },
  "secureJsonFields": {},
  "version": 1,
  "readOnly": false
}
```

名前による単一データソースの取得

```
GET /api/datasources/name/:name
```

リクエストの例

```
GET /api/datasources/name/test_datasource HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "uid": "kLtEtcRGk",
  "orgId": 1,
  "name": "test_datasource",
  "type": "graphite",
  "typeLogoUrl": "",
  "access": "proxy",
  "url": "http://mydatasource.com",
  "password": "",
```

```
  "user": "",
  "database": "",
  "basicAuth": false,
  "basicAuthUser": "",
  "basicAuthPassword": "",
  "withCredentials": false,
  "isDefault": false,
  "jsonData": {
    "graphiteType": "default",
    "graphiteVersion": "1.1"
  },
  "secureJsonFields": {},
  "version": 1,
  "readOnly": false
}
```

名前によるデータソース Id の取得

```
GET /api/datasources/id/:name
```

リクエストの例

```
GET /api/datasources/id/test_datasource HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1
}
```

データソースを作成します。

```
POST /api/datasources
```

Graphite のリクエスト例

```
POST /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name":"test_datasource",
  "type":"graphite",
  "url":"http://mydatasource.com",
  "access":"proxy",
  "basicAuth":false
}
```

Graphite のレスポンス例

```
HTTP/1.1 200
Content-Type: application/json

{
  "datasource": {
    "id": 1,
    "orgId": 1,
    "name": "test_datasource",
    "type": "graphite",
    "typeLogoUrl": "",
    "access": "proxy",
    "url": "http://mydatasource.com",
    "password": "",
    "user": "",
    "database": "",
    "basicAuth": false,
    "basicAuthUser": "",
    "basicAuthPassword": "",
    "withCredentials": false,
    "isDefault": false,
    "jsonData": {},
    "secureJsonFields": {},
    "version": 1,
    "readOnly": false
  },
  "id": 1,
  "message": "Datasource added",
  "name": "test_datasource"
```

```
}
```

> **Note**
>
> `secureJsonData` 内で `password` および `basicAuthPassword` を定義すると、Amazon Managed Grafana により暗号化された blob として、データベース内で安全に暗号化されます。その後、レスポンスにより `secureJsonFields` の暗号化フィールドが一覧表示されます。

基本的な認証が有効な Graphite のリクエスト例

```
POST /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "test_datasource",
  "type": "graphite",
  "url": "http://mydatasource.com",
  "access": "proxy",
  "basicAuth": true,
  "basicAuthUser": "basicuser",
  "secureJsonData": {
    "basicAuthPassword": "basicpassword"
  }
}
```

基本的な認証が有効なレスポンス例

```
HTTP/1.1 200
Content-Type: application/json

{
  "datasource": {
    "id": 1,
    "orgId": 1,
    "name": "test_datasource",
    "type": "graphite",
    "typeLogoUrl": "",
```

```
    "access": "proxy",
    "url": "http://mydatasource.com",
    "password": "",
    "user": "",
    "database": "",
    "basicAuth": true,
    "basicAuthUser": "basicuser",
    "basicAuthPassword": "",
    "withCredentials": false,
    "isDefault": false,
    "jsonData": {},
    "secureJsonFields": {
      "basicAuthPassword": true
    },
    "version": 1,
    "readOnly": false
  },
  "id": 102,
  "message": "Datasource added",
  "name": "test_datasource"
}
```

CloudWatch のリクエスト例

```
POST /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "test_datasource",
  "type": "cloudwatch",
  "url": "http://monitoring.us-west-1.amazonaws.com",
  "access": "proxy",
  "jsonData": {
    "authType": "keys",
    "defaultRegion": "us-west-1"
  },
  "secureJsonData": {
    "accessKey": "Ol4pIDpeKSA6XikgOl4p",
    "secretKey": "dGVzdCBrZXkgYmxlYXNlIGRvbid0IHN0ZWFs"
  }
}
```

既存のデータソースの削除

```
PUT /api/datasources/:datasourceId
```

リクエストの例

```
PUT /api/datasources/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "id":1,
  "orgId":1,
  "name":"test_datasource",
  "type":"graphite",
  "access":"proxy",
  "url":"http://mydatasource.com",
  "password":"",
  "user":"",
  "database":"",
  "basicAuth":true,
  "basicAuthUser":"basicuser",
  "secureJsonData": {
    "basicAuthPassword": "basicpassword"
  },
  "isDefault":false,
  "jsonData":null
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "datasource": {
    "id": 1,
    "orgId": 1,
    "name": "test_datasource",
    "type": "graphite",
    "typeLogoUrl": "",
```

```
    "access": "proxy",
    "url": "http://mydatasource.com",
    "password": "",
    "user": "",
    "database": "",
    "basicAuth": true,
    "basicAuthUser": "basicuser",
    "basicAuthPassword": "",
    "withCredentials": false,
    "isDefault": false,
    "jsonData": {},
    "secureJsonFields": {
      "basicAuthPassword": true
    },
    "version": 1,
    "readOnly": false
  },
  "id": 102,
  "message": "Datasource updated",
  "name": "test_datasource"
}
```

> **Note**
>
> secureJsonData 内で password および basicAuthPassword を定義して、暗号化された blob としてデータベースに安全に保存することをお勧めします。その後、レスポンスにより secureJsonFields の暗号化フィールドが一覧表示されます。

Id によるデータソースの削除

```
DELETE /api/datasources/:datasourceId
```

リクエストの例

```
DELETE /api/datasources/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Data source deleted"}
```

UID によるデータソースの削除

```
DELETE /api/datasources/uid/:uid
```

リクエストの例

```
DELETE /api/datasources/uid/kLtEtcRGk HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Data source deleted"}
```

名前によるデータソースの削除

```
DELETE /api/datasources/name/:datasourceName
```

リクエストの例

```
DELETE /api/datasources/name/test_datasource HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
```

```
{
  "message":"Data source deleted",
  "id": 1
}
```

データソースのプロキシコール

```
GET /api/datasources/proxy/:datasourceId/*
```

実際のデータソースへのすべてのコールをプロキシします。

データソースのクエリ

```
POST /api/ds/query
```

バックエンド実装を持つデータソースをクエリします。大半の組み込みデータソースにはバックエンド実装があります。

リクエストの例

```
POST /api/ds/query HTTP/1.1
Accept: application/json
Content-Type: application/json

{
   "queries":[
      {
         "refId":"A",
         "scenarioId":"csv_metric_values",
         "datasource":{
            "uid":"PD8C576611E62080A"
         },
         "format": "table",
         "maxDataPoints":1848,
         "intervalMs":200,
         "stringInput":"1,20,90,30,5,0"
      }
   ],
   "from":"now-5m",
   "to":"now"
}
```

JSON ボディスキーマ:

- from/to — クエリの時間範囲を指定します。時間は、エポックタイムスタンプをミリ秒単位にするか、Grafana 時間単位を使用して相対的にすることができます。例えば、now-5m。
- queries — 1 つ以上のクエリを指定します。少なくとも 1 を含める必要があります。
- queries.datasource.uid — クエリするデータソースの UID を指定します。リクエスト内の各クエリには、一意の datasource が必要です。
- queries.refId — クエリの識別子を指定します。デフォルトは「A」です。
- queries.format — データが返される形式を指定します。有効なオプションは、データソースtableに応じて time_seriesまたは です。
- queries.maxDataPoints — ダッシュボードパネルがレンダリングできるデータポイントの最大量を指定します。デフォルトは 100 です。
- queries.intervalMs — 時系列の時間間隔をミリ秒単位で指定します。デフォルトは 1000 です。

さらに、各データソースの特定のプロパティをリクエストに追加する必要があります (上記のリクエストに示すように、Querys.stringInput など) 。特定のデータソースのクエリを作成する方法をよりよく理解するには、選択したブラウザで デベロッパーツールを使用し、 に対して行われる HTTP リクエストを調べます/api/ds/query。

テストデータソースの時系列クエリレスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "results": {
    "A": {
      "frames": [
        {
          "schema": {
            "refId": "A",
            "fields": [
              {
                "name": "time",
                "type": "time",
                "typeInfo": {
                  "frame": "time.Time"
                }
              },
```

```
              {
                "name": "A-series",
                "type": "number",
                "typeInfo": {
                  "frame": "int64",
                  "nullable": true
                }
              }
            ]
          },
          "data": {
            "values": [
              [1644488152084, 1644488212084, 1644488272084, 1644488332084,
 1644488392084, 1644488452084],
              [1, 20, 90, 30, 5, 0]
            ]
          }
        }
      ]
    }
  }
}
```

ID によるデータソースのクエリ

```
POST /api/tsdb/query
```

> **Important**
>
> バージョン 9 以降、 `/api/tsdb/query`はサポートされていません。データソースのクエリを使用します。

バックエンド実装されるデータソースをクエリします。大半の組み込みデータソースにはバックエンド実装があります。

リクエストの例

```
POST /api/tsdb/query HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
{
  "from": "1420066800000",
  "to": "1575845999999",
  "queries": [
    {
      "refId": "A",
      "intervalMs": 86400000,
      "maxDataPoints": 1092,
      "datasourceId": 86,
      "rawSql": "SELECT 1 as valueOne, 2 as valueTwo",
      "format": "table"
    }
  ]
}
```

> **Note**
>
> from、to、queries のプロパティが必要です。

JSON ボディスキーマ:

- 開始/終了 — ミリ秒単位のエポックタイムスタンプの絶対値、または Grafana 時間単位を使用する相対値のいずれかである必要があります。例えば、now-1h。
- queries.refId — (オプション) クエリのための識別子を指定します。デフォルトは A です。
- queries.datasourceId — クエリを実行するデータソースを指定します。リクエスト内の各クエリには、一意の datasourceId が必要です。
- queries.maxDataPoints — (オプション) ダッシュボードパネルがレンダリングできる、データポイントの最大量を指定します。デフォルトは 100 です。
- queries.intervalIMs — (オプション) ミリ秒単位で時系列の時間間隔を指定します。デフォルトは 1000 です

MySQL データソースのリクエスト例:

```
POST /api/tsdb/query HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
{
  "from": "1420066800000",
  "to": "1575845999999",
  "queries": [
    {
      "refId": "A",
      "intervalMs": 86400000,
      "maxDataPoints": 1092,
      "datasourceId": 86,
      "rawSql": "SELECT\n  time,\n  sum(opened) AS \"Opened\",\n  sum(closed) AS
 \"Closed\"\nFROM\n  issues_activity\nWHERE\n  $__unixEpochFilter(time) AND\n  period
 = 'm' AND\n  repo IN('grafana/grafana') AND\n  opened_by IN('Contributor','Grafana
 Labs')\nGROUP BY 1\nORDER BY 1\n",
      "format": "time_series"
    }
  ]
}
```

MySQL データソースリクエストのレスポンス例:

```
HTTP/1.1 200
Content-Type: application/json

{
  "results": {
    "A": {
      "refId": "A",
      "meta": {
        "rowCount": 0,
        "sql": "SELECT\n  time,\n  sum(opened) AS \"Opened\",\n  sum(closed) AS
 \"Closed\"\nFROM\n  issues_activity\nWHERE\n  time <= 1420066800 AND time >=
 1575845999 AND\n  period = 'm' AND\n  repo IN('grafana/grafana') AND\n  opened_by
 IN('Contributor','Grafana Labs')\nGROUP BY 1\nORDER BY 1\n"
      },
      "series": [
        {
          "name": "Opened",
          "points": [
            [
              109,
              1420070400000
            ],
            [
```

```
              122,
              1422748800000
            ]
          ]
        },
        {
          "name": "Closed",
          "points": [
            [
              89,
              1420070400000
            ]
          ]
        }
      ]
    }
  }
}
```

データソースアクセス許可 API

データソースアクセス許可 API を使用して、データソースのアクセス許可を有効化、無効化、一覧表示、追加、削除します。

ユーザーまたはチームにアクセス許可を設定できます。管理者は常にすべてにアクセスできるため、管理者にはアクセス許可を設定できません。

アクセス許可フィールドのアクセス許可レベルは、次のとおりです。

- 1 = クエリ

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

データソースのアクセス許可の有効化

```
POST /api/datasources/:id/enable-permissions
```

指定 id を持つデータソースのアクセス許可を有効にします。特定のユーザーまたはチームにデータソースのクエリのためのアクセス許可が追加されるまで、組織管理者以外の誰もデータソースをクエリできません。

リクエストの例

```
POST /api/datasources/1/enable-permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permissions enabled"}
```

ステータスコード:

- 200 — 作成済み
- 400 — アクセス許可を有効にできません。詳細については、レスポンス本文を参照してください。
- 401 — 未許可
- 403 — アクセス拒否
- 404 — データソースが未検出

データソースのアクセス許可の無効化

```
POST /api/datasources/:id/disable-permissions
```

指定 id を持つデータソースのアクセス許可を無効にします。既存のアクセス許可がすべて削除され、誰でもデータソースをクエリできます。

リクエストの例

```
POST /api/datasources/1/disable-permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{}
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permissions disabled"}
```

ステータスコード:

- 200 — OK
- 400 — アクセス許可を無効にできません。詳細については、レスポンス本文を参照してください。
- 401 — 未許可
- 403 — アクセス拒否
- 404 — データソースが未検出

データソースのアクセス許可の取得

```
GET /api/datasources/:id/permissions
```

指定された id を使用して、データソースの既存のアクセス許可をすべて取得します。

リクエストの例

```
GET /api/datasources/1/permissions HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 551

{
  "datasourceId": 1,
  "enabled": true,
  "permissions":
  [
    {
      "id": 1,
      "datasourceId": 1,
      "userId": 1,
      "userLogin": "user",
      "userEmail": "user@test.com",
      "userAvatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae",
      "permission": 1,
      "permissionName": "Query",
      "created": "2017-06-20T02:00:00+02:00",
      "updated": "2017-06-20T02:00:00+02:00",
    },
    {
      "id": 2,
      "datasourceId": 1,
      "teamId": 1,
      "team": "A Team",
      "teamAvatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae",
      "permission": 1,
      "permissionName": "Query",
      "created": "2017-06-20T02:00:00+02:00",
      "updated": "2017-06-20T02:00:00+02:00",
    }
  ]
}
```

ステータスコード:

- 200 — OK
- 401 — 未許可

- 403 — アクセス拒否
- 404 — データソースが未検出

データソースのアクセス許可の追加

```
POST /api/datasources/:id/permissions
```

指定 id を持つデータソースのユーザーアクセス許可を追加します。

ユーザーアクセス許可を追加するリクエスト例

```
POST /api/datasources/1/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "userId": 1,
  "permission": 1
}
```

ユーザーアクセス許可を追加するレスポンス例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permission added"}
```

チームアクセス許可を追加するリクエスト例

```
POST /api/datasources/1/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "teamId": 1,
  "permission": 1
```

```
}
```

チームアクセス許可を追加するレスポンス例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permission added"}
```

ステータスコード:

- 200 — OK
- 400 — アクセス許可を追加できません。詳細については、レスポンス本文を参照してください。
- 401 — 未許可
- 403 — アクセス拒否
- 404 — データソースが未検出

データソースのアクセス許可の削除

```
DELETE /api/datasources/:id/permissions/:permissionId
```

指定 id のデータソースに対する指定 permissionId のアクセス許可を削除します。

リクエストの例

```
DELETE /api/datasources/1/permissions/2
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35
```

```
{"message":"Datasource permission removed"}
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 403 — アクセス拒否
- 404 — データソースが未検出またはアクセス許可が未検出

外部グループ同期 API

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

外部グループの取得

```
GET /api/teams/:teamId/groups
```

リクエストの例

```
GET /api/teams/1/groups HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk]
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
```

```
    "orgId": 1,
    "teamId": 1,
    "groupId": "cn=editors,ou=groups,dc=grafana,dc=org"
  }
]
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 403 — アクセス拒否

外部グループの追加

```
POST /api/teams/:teamId/groups
```

リクエストの例

```
POST /api/teams/1/members HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk]

{
  "groupId": "cn=editors,ou=groups,dc=grafana,dc=org"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Group added to Team"}
```

ステータスコード:

- 200 — OK
- 400 — グループが既にこのチームに追加済み

- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チーム未検出

外部グループの削除

```
DELETE /api/teams/:teamId/groups/:groupId
```

リクエストの例

```
DELETE /api/teams/1/groups/cn=editors,ou=groups,dc=grafana,dc=org HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk]
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Team Group removed"}
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 403— アクセス許可拒否
- 404 - チームが未検出またはグループが未検出

フォルダ API

フォルダ API を使用して Amazon Managed Grafana ワークスペースのフォルダを操作します。

フォルダの識別子 (id) は自動増分数値あり、ワークスペースごとに一意です。フォルダの一意の識別子 (uid) は、複数のワークスペース間のフォルダを一意に識別ために使用できます。フォルダの作成時に指定しない場合は自動的に生成されます。uid を使用すると、フォルダにアクセスしたり、複数

の Amazon Managed Grafana ワークスペース間のフォルダを同期したりするための、一貫した URL を使用できます。uid を使用すると、フォルダのタイトルを変更しても、そのフォルダにブックマークされたリンクが破損することはありません。

uid の最大長は 40 文字までです。

フォルダのネストはできません。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

`id` が 0 の [全般] フォルダは、フォルダ API の一部ではありません。フォルダ API を使用して全般フォルダの情報を取得することはできません。

フォルダの作成

```
POST /api/folders
```

新しいフォルダを作成します。

リクエストの例

```
POST /api/folders HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "uid": "nErXDvCkzz",
  "title": "Department ABC"
}
```

JSON ボディスキーマ:

- uid — オプションの、一意の ID。null の場合は新しい uid が生成されます。

- タイトル — フォルダのタイトルです。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

ステータスコード:

- 200 — 作成済み
- 400 — 無効な JSON、無効なフィールド、欠落したフィールドなどのエラー
- 401 — 未許可
- 403 — アクセス拒否

フォルダの更新

```
PUT /api/folders/:uid
```

uid と一致する既存のフォルダを更新します。

リクエストの例

```
PUT /api/folders/nErXDvCkzz HTTP/1.1
```

```
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "title":"Department DEF",
  "version": 1
}
```

JSON ボディスキーマ:

- uid — 指定されると、一意の ID を変更します。
- タイトル — フォルダのタイトルです。
- version — フォルダの上書きができるように、現在のバージョンを指定します。overwrite=true の場合は不要です。
- overwrite — true にセットすると、既存のフォルダが新しいバージョンに上書きされます。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department DEF",
  "url": "/dashboards/f/nErXDvCkzz/department-def",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

ステータスコード:

- 200 — 作成済み

- 400 — 無効な JSON、無効なフィールド、欠落したフィールドなどのエラー
- 401 — 未許可
- 403 — アクセス拒否
- 404 — フォルダが未検出
- 412 — 前提条件失敗

412 ステータスコードは、フォルダを更新できない理由の説明に使用されます。

- フォルダが他のユーザーによって変更された `status=version-mismatch`

レスポンス本文には次のプロパティがあります:

```
HTTP/1.1 412 Precondition Failed
Content-Type: application/json; charset=UTF-8
Content-Length: 97

{
  "message": "The folder has been changed by someone else",
  "status": "version-mismatch"
}
```

すべてのフォルダの取得

```
GET /api/folders
```

表示アクセス許可を持つすべてのフォルダを返します。`limit` クエリパラメータを使用すると、返されるフォルダの最大数を制御できます。デフォルトは 1000 です。

リクエストの例

```
GET /api/folders?limit=10 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
```

```
Content-Type: application/json

[
  {
    "id":1,
    "uid": "nErXDvCkzz",
    "title": "Department ABC"
  },
  {
    "id":2,
    "uid": "k3S1cklGk",
    "title": "Department RND"
  }
]
```

uid によるフォルダの取得

```
GET /api/folders/:uid
```

指定 uid と一致するすべてのフォルダを返します。

リクエストの例

```
GET /api/folders/nErXDvCkzzh HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
```

```
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

ステータスコード:

- 200 — 検出済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — 未検出

id によるフォルダの取得

```
GET /api/folders/id/:id
```

指定 id と一致するすべてのフォルダを返します。

リクエストの例

```
GET /api/folders/id/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
```

```
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

ステータスコード:

- 200 — 検出済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — 未検出

uid によるフォルダの削除

```
DELETE /api/folders/:uid
```

uid と一致するフォルダを削除し、フォルダに保存されているすべてのダッシュボードも削除します。このオペレーションを元に戻すことはできません。

リクエストの例

```
DELETE /api/folders/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "message":"Folder deleted",
  "id": 2
```

```
}
```

ステータスコード:

- 200 — 削除済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — 未検出

フォルダ/ダッシュボード検索 API

フォルダ/ダッシュボード検索 API を使用して、Amazon Managed Grafana ワークスペース内のフォルダとダッシュボードを検索します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

フォルダとダッシュボードの検索

```
GET /api/search/
```

クエリパラメータ:

- query — クエリの検索
- tag — 検索するタグのリストです。これらは Grafana タグであり、 AWS タグではありません。
- [タイプ] — 検索する、`dash-folder` または `dash-db` のいずれかのタイプです。
- dashboardIds — 検索するダッシュボード Id のリストです。
- folderIds — ダッシュボード内で検索するダッシュボード Id のリストです。
- starred — スター付きダッシュボードのみを返すように指定するフラグです。
- [制限] — 返される結果の数の制限 (最大は 5000 です)。

- [ページ] — このパラメータを使用して制限を超えるヒットにアクセスします。1 から番号が始まります。limit パラメータはページサイズの役目を果たします。

全般フォルダのフォルダとダッシュボードを取得するためのリクエスト例

```
GET /api/search?folderIds=0&query=&starred=false HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

全般フォルダのフォルダとダッシュボードを取得するためのレスポンス例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 163,
    "uid": "000000163",
    "title": "Folder",
    "url": "/dashboards/f/000000163/folder",
    "type": "dash-folder",
    "tags": [],
    "isStarred": false,
    "uri":"db/folder" // deprecated in Grafana v5.0
  },
  {
    "id":1,
    "uid": "cIBgcSjkk",
    "title":"Production Overview",
    "url": "/d/cIBgcSjkk/production-overview",
    "type":"dash-db",
    "tags":[prod],
    "isStarred":true,
    "uri":"db/production-overview" // deprecated in Grafana v5.0
  }
]
```

スター付きダッシュボードの検索のリクエスト例

```
GET /api/search?query=Production%20Overview&starred=true&tag=prod HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

スター付きダッシュボードの検索のレスポンス例

```
HTTP/1.1 200
Content-Type: application/json

[HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":1,
    "uid": "cIBgcSjkk",
    "title":"Production Overview",
    "url": "/d/cIBgcSjkk/production-overview",
    "type":"dash-db",
    "tags":[prod],
    "isStarred":true,
    "folderId": 2,
    "folderUid": "000000163",
    "folderTitle": "Folder",
    "folderUrl": "/dashboards/f/000000163/folder",
    "uri":"db/production-overview" // deprecated in Grafana v5.0
  }
]
```

フォルダアクセス許可 API

フォルダ API を使用してフォルダのアクセス許可を更新または取得します。

`folderId=-1` のアクセス許可は、ビューワーロールとエディタロールを持つユーザーのデフォルトのアクセス許可です。アクセス許可は、ユーザー、チーム、またはロール (ビューワーまたはエディタ) に設定できます。管理者にアクセス許可を設定することはできません。管理者は常にすべてにアクセスできます。

`permission` フィールドのアクセス許可レベルは、次のとおりです。

- 1 = 表示
- 2 = 編集

- 4 = 管理者

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの Authorization フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

フォルダのアクセス許可の取得

```
GET /api/folders/:uid/permissions
```

指定された uid を使用して、フォルダの既存のアクセス許可をすべて取得します。

リクエストの例

```
GET /api/folders/nErXDvCkzz/permissions HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 551

[
  {
    "id": 1,
    "folderId": -1,
    "created": "2017-06-20T02:00:00+02:00",
    "updated": "2017-06-20T02:00:00+02:00",
    "userId": 0,
    "userLogin": "",
    "userEmail": "",
    "teamId": 0,
    "team": "",
```

```
    "role": "Viewer",
    "permission": 1,
    "permissionName": "View",
    "uid": "nErXDvCkzz",
    "title": "",
    "slug": "",
    "isFolder": false,
    "url": ""
  },
  {
    "id": 2,
    "dashboardId": -1,
    "created": "2017-06-20T02:00:00+02:00",
    "updated": "2017-06-20T02:00:00+02:00",
    "userId": 0,
    "userLogin": "",
    "userEmail": "",
    "teamId": 0,
    "team": "",
    "role": "Editor",
    "permission": 2,
    "permissionName": "Edit",
    "uid": "",
    "title": "",
    "slug": "",
    "isFolder": false,
    "url": ""
  }
]
```

ステータスコード:

- 200 — OK
- 401 — 未許可
- 403 — アクセス拒否
- 404 — フォルダが未検出

フォルダに対するアクセス許可の更新

```
POST /api/folders/:uid/permissions
```

フォルダのアクセス許可を更新します。このオペレーションでは、既存のアクセス許可がリクエストに含まれていない場合、それらのアクセス許可を削除します。

リクエストの例

```
POST /api/folders/nErXDvCkzz/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
{
  "items": [
    {
      "role": "Viewer",
      "permission": 1
    },
    {
      "role": "Editor",
      "permission": 2
    },
    {
      "teamId": 1,
      "permission": 1
    },
    {
      "userId": 11,
      "permission": 4
    }
  ]
}
```

JSON ボディスキーマ:

- items — 追加または更新するアクセス許可の項目です。リストから省略された既存の項目が削除されます。

レスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Folder permissions updated","id":1,"title":"Department ABC"}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403 — アクセス拒否
- 404 — ダッシュボード未検出

組織 API

Organization API を使用して、Amazon Managed Grafana ワークスペース内の組織を使用します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

現在の組織の取得

```
GET /api/org/
```

リクエストの例

```
GET /api/org/ HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
```

```
  "name":"Main Org."
}
```

現在の組織内のすべてのユーザーの取得

```
GET /api/org/users
```

必要なアクセス許可: 範囲が users:* の org.users:read アクション

リクエストの例

```
GET /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId": 1,
    "userId": 1,
    "email": "admin@localhost",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae",
    "login": "admin",
    "role": "Admin",
    "lastSeenAt": "2019-08-09T11:02:49+02:00",
    "lastSeenAtAge": "< 1m"
  }
]
```

現在の組織内のすべてのユーザーの取得 (ルックアップ)

```
GET /api/org/users/lookup
```

現在の組織内のすべてのユーザーを返しますが、詳細情報が少なくなります。組織の管理者ロール、任意のフォルダの管理者、または任意のチームの管理者であるユーザーがアクセスできます。主に

チームメンバーを追加するとき、およびフォルダ/ダッシュボードのアクセス許可を編集するときに、Grafana UI がユーザーリストを提供するために使用します。

リクエストの例

```
GET /api/org/users/lookup HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "userId": 1,
    "login": "admin",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae"
  }
]
```

指定ユーザーの更新

```
PATCH /api/org/users/:userId
```

必要なアクセス許可: 範囲が `users:*` の `org.users.role:update` アクション

リクエストの例

```
PATCH /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Viewer",
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization user updated"}
```

現在の組織のユーザーの削除

```
DELETE /api/org/users/:userId
```

必要なアクセス許可: 範囲が users:* の org.users:remove アクション

リクエストの例

```
DELETE /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"User removed from organization"}
```

現在の組織の更新

```
PUT /api/org
```

リクエストの例

```
PUT /api/org HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name":"Main Org."
```

```
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization updated"}
```

現在の組織へのユーザーの追加

```
POST /api/org/users
```

必要なアクセス許可: 範囲が `users:*` の `org.users:add` アクション

リクエストの例

```
POST /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Admin",
  "loginOrEmail": "admin"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"User added to organization","userId":1}
```

プレイリスト API

プレイリスト API を使用して Amazon Managed Grafana ワークスペースのプレイリストを操作します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

プレイリストの検索

```
GET /api/playlists
```

ページ分割を使用して、現在の Amazon Managed Grafana ワークスペースのプレイリストすべてを返します。

リクエストの例

```
GET /api/playlists HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリ文字列パラメータ:

- query — この値のような名前のプレイリストにレスポンスを制限します。
- [制限] — X 個のプレイリストにレスポンスを制限します。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id": 1,
    "name": "my playlist",
    "interval": "5m"
  }
]
```

プレイリストを 1 つ取得

```
GET /api/playlists/:id
```

リクエストの例

```
GET /api/playlists/1 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "id" : 1,
  "name": "my playlist",
  "interval": "5m",
  "orgId": "my org",
  "items": [
    {
      "id": 1,
      "playlistId": 1,
      "type": "dashboard_by_id",
      "value": "3",
      "order": 1,
      "title":"my third dashboard"
    },
    {
      "id": 2,
      "playlistId": 1,
      "type": "dashboard_by_tag",
      "value": "myTag",
      "order": 2,
      "title":"my other dashboard"
    }
  ]
}
```

プレイリスト項目の取得

```
GET /api/playlists/:id/items
```

リクエストの例

```
GET /api/playlists/1/items HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id": 1,
    "playlistId": 1,
    "type": "dashboard_by_id",
    "value": "3",
    "order": 1,
    "title":"my third dashboard"
  },
  {
    "id": 2,
    "playlistId": 1,
    "type": "dashboard_by_tag",
    "value": "myTag",
    "order": 2,
    "title":"my other dashboard"
  }
]
```

プレイリストダッシュボードの取得

```
GET /api/playlists/:id/dashboards
```

リクエストの例

```
GET /api/playlists/1/dashboards HTTP/1.1
Accept: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id": 3,
    "title": "my third dashboard",
    "order": 1,
  },
  {
    "id": 5,
    "title":"my other dashboard"
    "order": 2,

  }
]
```

プレイリストの作成

```
POST /api/playlists/
```

リクエストの例

```
PUT /api/playlists/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
  {
    "name": "my playlist",
    "interval": "5m",
    "items": [
      {
        "type": "dashboard_by_id",
        "value": "3",
        "order": 1,
        "title":"my third dashboard"
      },
      {
        "type": "dashboard_by_tag",
```

```
        "value": "myTag",
        "order": 2,
        "title":"my other dashboard"
      }
    ]
  }
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
  {
    "id": 1,
    "name": "my playlist",
    "interval": "5m"
  }
```

プレイリストの更新

```
PUT /api/playlists/:id
```

リクエストの例

```
PUT /api/playlists/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
  {
    "name": "my playlist",
    "interval": "5m",
    "items": [
      {
        "playlistId": 1,
        "type": "dashboard_by_id",
        "value": "3",
        "order": 1,
        "title":"my third dashboard"
      },
      {
        "playlistId": 1,
        "type": "dashboard_by_tag",
        "value": "myTag",
```

```
        "order": 2,
        "title":"my other dashboard"
      }
    ]
  }
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "id" : 1,
  "name": "my playlist",
  "interval": "5m",
  "orgId": "my org",
  "items": [
    {
      "id": 1,
      "playlistId": 1,
      "type": "dashboard_by_id",
      "value": "3",
      "order": 1,
      "title":"my third dashboard"
    },
    {
      "id": 2,
      "playlistId": 1,
      "type": "dashboard_by_tag",
      "value": "myTag",
      "order": 2,
      "title":"my other dashboard"
    }
  ]
}
```

プレイリストの削除

```
DELETE /api/playlists/:id
```

リクエストの例

```
DELETE /api/playlists/1 HTTP/1.1
```

```
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{}
```

プラグイン API

プラグイン API を使用して、Amazon Managed Grafana ワークスペースでプラグインを管理します。この API を使用してプラグインを変更するには、ご使用のワークスペースでプラグイン管理を有効にする必要があります。Grafana API キーで定義されたユーザーは、Amazon Managed Grafana ワークスペースの管理者でもある必要があります。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

プラグインのインストール

```
POST /api/plugins/:id/install
```

リクエストの例

```
POST /api/plugins/grafana-athena-datasource/install HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "version": "2.12.0" # optional, uses the latest compatible version if not provided
}
```

レスポンスの例

```
HTTP/1.1 200
```

プラグインのアンインストール

```
POST /api/plugins/:id/uninstall
```

リクエストの例

```
POST /api/plugins/grafana-athena-datasource/uninstall HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "version": "2.12.0" # optional, uninstalls whatever is installed if not provided
}
```

レスポンスの例

```
HTTP/1.1 200
```

すべてのプラグインの取得

```
GET /api/gnet/plugins
```

リクエストの例

```
GET /api/gnet/plugins HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "items": [
```

```
    {
      "status": "active",
      "id": 74,
      "typeId": 1,
      "typeName": "Application",
      "typeCode": "app",
      "slug": "alexanderzobnin-zabbix-app",
      "name": "Zabbix",
      "description": "Zabbix plugin for Grafana",
      "version": "4.4.3",
      "versionStatus": "active",
      "versionSignatureType": "grafana",
      "versionSignedByOrg": "grafana",
      "versionSignedByOrgName": "Grafana Labs",
      "userId": 0,
      "orgId": 13056,
      "orgName": "Alexander Zobnin",
      "orgSlug": "alexanderzobnin",
      "orgUrl": "https://github.com/alexanderzobnin",
      "url": "https://github.com/grafana/grafana-zabbix/",
      "createdAt": "2016-04-06T20:23:41.000Z",
      "updatedAt": "2023-10-10T12:53:51.000Z",
      "downloads": 90788771,
      "verified": false,
      "featured": 180,
      "internal": false,
      "downloadSlug": "alexanderzobnin-zabbix-app",
      "popularity": 0.2485,
      "signatureType": "grafana",
      "packages": {
        "linux-amd64": {
          "md5": "baa06e8f26731f99748c58522cd4ffb6",
          "sha256": "a4a108f2e04a2114810c7b60419b4b04bf80d3377e2394b0586e2dc96b5a929c",
          "packageName": "linux-amd64",
          "downloadUrl": "/api/plugins/alexanderzobnin-zabbix-app/versions/4.4.3/
download?os=linux&arch=amd64"
        },
        <... further packages>
      },
      "links": [
        {
          "rel": "self",
          "href": "/plugins/alexanderzobnin-zabbix-app"
        },
```

```
        <... further links>
      ],
      "angularDetected": false
    },
    <... further plugins>
  ],
  "orderBy": "weight",
  "direction": "asc",
  "links": [
    {
      "rel": "self",
      "href": "/plugins"
    }
  ]
}
```

プラグインの取得

```
GET /api/gnet/plugins/:id
```

リクエストの例

```
GET /api/gnet/plugins/grafana-athena-datasource HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "status": "active",
  "id": 764,
  "typeId": 2,
  "typeName": "Data Source",
  "typeCode": "datasource",
  "slug": "grafana-athena-datasource",
  "name": "Amazon Athena",
  "description": "Use Amazon Athena with Grafana",
  "version": "2.13.0",
  "versionStatus": "active",
```

```
  "versionSignatureType": "grafana",
  "versionSignedByOrg": "grafana",
  "versionSignedByOrgName": "Grafana Labs",
  "userId": 0,
  "orgId": 5000,
  "orgName": "Grafana Labs",
  "orgSlug": "grafana",
  "orgUrl": "https://grafana.org",
  "url": "https://github.com/grafana/athena-datasource/",
  "createdAt": "2021-11-24T08:55:41.000Z",
  "updatedAt": "2023-10-31T17:20:32.000Z",
  "json": {
    "$schema": "https://raw.githubusercontent.com/grafana/grafana/master/docs/sources/
developers/plugins/plugin.schema.json",
    "alerting": true,
    "annotations": true,
    "backend": true,
    "dependencies": {
      "grafanaDependency": ">=8.0.0",
      "plugins": []
    },
    "executable": "gpx_athena",
    "id": "grafana-athena-datasource",
    "includes": [
      {
        "name": "Cost Usage Report Monitoring",
        "path": "dashboards/cur-monitoring.json",
        "type": "dashboard"
      },
      {
        "name": "Amazon VPC Flow Logs",
        "path": "dashboards/vpc-flow-logs.json",
        "type": "dashboard"
      }
    ],
    "info": {
      "author": {
        "name": "Grafana Labs",
        "url": "https://grafana.com"
      },
      "build": {
        "time": 1698764559022,
        "repo": "https://github.com/grafana/athena-datasource",
        "branch": "main",
```

```
        "hash": "25cc131300f1ed22593bc3ba08b2bef7d23fbcd01",
        "build": 1462
      },
      "description": "Use Amazon Athena with Grafana",
      "keywords": [
        "datasource",
        "athena"
      ],
      "links": [
        {
          "name": "Website",
          "url": "https://github.com/grafana/athena-datasource"
        },
        {
          "name": "License",
          "url": "https://github.com/grafana/athena-datasource/blob/master/LICENSE"
        }
      ],
      "logos": {
        "large": "img/logo.svg",
        "small": "img/logo.svg"
      },
      "screenshots": [],
      "updated": "2023-10-31",
      "version": "2.13.0"
    },
    "metrics": true,
    "name": "Amazon Athena",
    "type": "datasource"
  },
  "readme": "<... full HTML readme>",
  "statusContext": "",
  "downloads": 2505825,
  "verified": false,
  "featured": 0,
  "internal": false,
  "downloadSlug": "grafana-athena-datasource",
  "popularity": 0.0594,
  "signatureType": "grafana",
  "grafanaDependency": ">=8.0.0",
  "packages": {
    "linux-amd64": {
      "md5": "7efef359bf917b4ca6b149de42a1282d",
      "sha256": "cd2fc5737c321dc3d8bbe2852c801c01adb64eacc9f60420bd21dc18bee43531",
```

```
      "packageName": "linux-amd64",
      "downloadUrl": "/api/plugins/grafana-athena-datasource/versions/2.13.0/download?
os=linux&arch=amd64"
    },
    <... other packages>
  },
  "links": [
    {
      "rel": "self",
      "href": "/plugins/grafana-athena-datasource"
    },
    <... other links>
  ],
  "angularDetected": false
}
```

プラグインバージョンの取得

```
POST /api/gnet/plugins/:id/versions
```

リクエストの例

```
GET /api/gnet/plugins/grafana-athena-datasource/versions HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "items": [
    {
      "id": 5306,
      "pluginId": 764,
      "pluginSlug": "grafana-athena-datasource",
      "version": "2.13.0",
      "url": "https://github.com/grafana/athena-datasource/",
      "commit": "",
      "description": "Use Amazon Athena with Grafana",
      "createdAt": "2023-10-31T17:20:31.000Z",
```

```
      "updatedAt": null,
      "downloads": 33790,
      "verified": false,
      "status": "active",
      "statusContext": "",
      "downloadSlug": "grafana-athena-datasource",
      "packages": {},
      "links": [
        {
          "rel": "self",
          "href": "/plugins/grafana-athena-datasource/versions/2.13.0"
        },
        {
          "rel": "images",
          "href": "/plugins/grafana-athena-datasource/versions/2.13.0/images"
        },
        {
          "rel": "thumbnails",
          "href": "/plugins/grafana-athena-datasource/versions/2.13.0/thumbnails"
        },
        {
          "rel": "plugin",
          "href": "/plugins/grafana-athena-datasource"
        },
        {
          "rel": "download",
          "href": "/plugins/grafana-athena-datasource/versions/2.13.0/download"
        }
      ],
      "grafanaDependency": ">=8.0.0",
      "angularDetected": false
    },
    {
      "id": 5244,
      "pluginId": 764,
      "pluginSlug": "grafana-athena-datasource",
      "version": "2.12.0",
      "url": "https://github.com/grafana/athena-datasource/",
      "commit": "",
      "description": "Use Amazon Athena with Grafana",
      "createdAt": "2023-10-17T12:42:13.000Z",
      "updatedAt": null,
      "downloads": 60742,
      "verified": false,
```

```
      "status": "active",
      "statusContext": "",
      "downloadSlug": "grafana-athena-datasource",
      "packages": {},
      "links": [
        {
          "rel": "self",
          "href": "/plugins/grafana-athena-datasource/versions/2.12.0"
        },
        {
          "rel": "images",
          "href": "/plugins/grafana-athena-datasource/versions/2.12.0/images"
        },
        {
          "rel": "thumbnails",
          "href": "/plugins/grafana-athena-datasource/versions/2.12.0/thumbnails"
        },
        {
          "rel": "plugin",
          "href": "/plugins/grafana-athena-datasource"
        },
        {
          "rel": "download",
          "href": "/plugins/grafana-athena-datasource/versions/2.12.0/download"
        }
      ],
      "grafanaDependency": ">=8.0.0",
      "angularDetected": false
    },
    <... other versions>
  ]
}
```

設定 API

設定 API を使用して、Amazon Managed Grafana ワークスペースでユーザー設定を使用します。

キー:

- theme — 有効な値は `light`、`dark`、またはデフォルトテーマを使用する空の文字列です。
- homeDashboardId — お気に入りのダッシュボードの数値の `:id` です。デフォルトは 0 です。
- [タイムゾーン] — 有効な値は `utc`、`browser`、またはデフォルトを使用する空の文字列です。

キーを省略すると、現在の値はシステムのデフォルト値に置き換えられます。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの Authorization フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

現在のユーザー設定の取得

```
GET /api/user/preferences
```

リクエストの例

```
GET /api/user/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"theme":"","homeDashboardId":0,"timezone":""}
```

現在のユーザー設定の更新

```
PUT /api/user/preferences
```

リクエストの例

```
PUT /api/user/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "theme": "",
  "homeDashboardId":0,
  "timezone":"utc"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{"message":"Preferences updated"}
```

現在の組織設定の取得

```
GET /api/org/preferences
```

リクエストの例

```
GET /api/org/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"theme":"","homeDashboardId":0,"timezone":""}
```

現在の組織設定の更新

```
PUT /api/org/preferences
```

リクエストの例

```
PUT /api/org/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "",
  "homeDashboardId":0,
  "timezone":"utc"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{"message":"Preferences updated"}
```

スナップショット API

スナップショット API を使用して、Amazon Managed Grafana ワークスペースのスナップショットを使用します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

新しいシェープショットの作成

```
POST /api/snapshots
```

リクエストの例

```
POST /api/snapshots HTTP/1.1
    Accept: application/json
    Content-Type: application/json
    Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "dashboard": {
    "editable":false,
    "hideControls":true,
    "nav":[
    {
      "enable":false,
    "type":"timepicker"
    }
    ],
    "rows": [
      {

      }
    ],
    "style":"dark",
    "tags":[],
    "templating":{
      "list":[
      ]
    },
    "time":{
    },
    "timezone":"browser",
    "title":"Home",
    "version":5
    },
  "expires": 3600
}
```

JSON ボディスキーマ:

- dashboard — (必須) 完全なダッシュボードモデルです。
- [名前] — (オプション) スナップショットの名前です。
- expires — (オプション) スナップショットの有効期限が切れるタイミングを、秒単位で指定します。デフォルトは、有効期限なしです。
- external — (オプション) スナップショットをローカルではなく、外部サーバーに保存します。デフォルトは偽です。
- key — (`external` が `true` の場合は必須) 一意のキーを定義します。
- deletekey — (`external` が `true` の場合は必須) スナップショットの削除に使用する一意のキーです。これは `key` とは異なり、作成者のみがスナップショットを削除できます。

> **Note**
>
> API を使用してスナップショットを作成するときは、スナップショットデータを含むダッシュボードペイロード全体を用意する必要があります。このエンドポイントは Grafana UI 用です。

レスポンスの例

```
HTTP/1.1 200
    Content-Type: application/json
    {
      "deleteKey":"XXXXXXX",
      "deleteUrl":"myurl/api/snapshots-delete/XXXXXXX",
      "key":"YYYYYYY",
      "url":"myurl/dashboard/snapshot/YYYYYYY",
      "id": 1,
    }
```

キー:

- deleteKey — スナップショットの削除に使用するために生成されたキーです。
- key — ダッシュボードを共有するために生成されたキーです。

スナップショットのリストの取得

```
GET /api/dashboard/snapshots
```

クエリパラメータ:

- query — クエリの検索
- [制限] — 返される結果数の制限

リクエストの例

```
GET /api/dashboard/snapshots HTTP/1.1
Accept: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":8,
    "name":"Home",
    "key":"YYYYYYY",
    "orgId":1,
    "userId":1,
    "external":false,
    "externalUrl":"",
    "expires":"2200-13-32T25:23:23+02:00",
    "created":"2200-13-32T28:24:23+02:00",
    "updated":"2200-13-32T28:24:23+02:00"
  }
]
```

key によるスナップショットの取得

```
GET /api/snapshots/:key
```

リクエストの例

```
GET /api/snapshots/YYYYYYY HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "meta":{
    "isSnapshot":true,
    "type":"snapshot",
```

```
    "canSave":false,
    "canEdit":false,
    "canStar":false,
    "slug":"",
    "expires":"2200-13-32T25:23:23+02:00",
    "created":"2200-13-32T28:24:23+02:00"
  },
  "dashboard": {
    "editable":false,
    "hideControls":true,
    "nav": [
      {
        "enable":false,
        "type":"timepicker"
      }
    ],
    "rows": [
      {

      }
    ],
    "style":"dark",
    "tags":[],
    "templating":{
      "list":[
      ]
    },
    "time":{
    },
    "timezone":"browser",
    "title":"Home",
    "version":5
  }
}
```

key によるスナップショットの削除

```
DELETE /api/snapshots/:key
```

リクエストの例

```
DELETE /api/snapshots/YYYYYYY HTTP/1.1
```

```
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Snapshot deleted. It might take an hour before it's cleared from any CDN
 caches.", "id": 1}
```

deleteKey によるスナップショットの削除

この API コールは、スナップショットのシークレット削除キーを使用すると認証なしで使用できます。

```
GET /api/snapshots-delete/:deleteKey
```

リクエストの例

```
GET /api/snapshots-delete/XXXXXXX HTTP/1.1
Accept: application/json
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Snapshot deleted. It might take an hour before it's cleared from any CDN
 caches.", "id": 1}
```

チーム API

チーム API を使用して、Amazon Managed Grafana ワークスペース内のチームを使用します。この API のすべてのアクションには管理者ロールが必要です。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに

含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

ページ分割によるチーム検索

```
GET /api/teams/search?perpage=50&page=1&query=myteam
```

or

```
GET /api/teams/search?name=myteam
```

リクエストの例

```
GET /api/teams/search?perpage=10&page=1&query=myteam HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリパラメータの使用

perpage パラメータのデフォルト値は 1000 で、page パラメータのデフォルト値は 1 です。

レスポンスの totalCount フィールドはチームリストのページ分割に使用できます。例えば、totalCount が 100 チームで perpage パラメータに 10 が設定されている場合、10 ページのチームがあります。

query パラメータはオプションで、クエリ値が nameフィールドに含まれている場合に結果を返します。クエリ値にスペースが含まれる場合は URL エンコードする必要があります。例えば、query=my%20team。

name パラメータの使用

パラメータが name フィールドと一致する場合、name パラメータは単一のチームを返します。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
```

```
{
  "totalCount": 1,
  "teams": [
    {
      "id": 1,
      "orgId": 1,
      "name": "MyTestTeam",
      "email": "",
      "avatarUrl": "\/avatar\/3f49c15916554246daa714b9bd0ee39",
      "memberCount": 1
    }
  ],
  "page": 1,
  "perPage": 1000
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チームが未検出 (名前により検索する場合)

Id によるチームの取得

```
GET /api/teams/:id
```

リクエストの例

```
GET /api/teams/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HHTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
```

```
  "orgId": 1,
  "name": "MyTestTeam",
  "email": "",
  "created": "2017-12-15T10:40:45+01:00",
  "updated": "2017-12-15T10:40:45+01:00"
}
```

チームの追加

チームの name は一意である必要があります。name フィールドは必須で、email および orgId フィールドはオプションです。

```
POST /api/teams
```

リクエストの例

```
POST /api/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "MyTestTeam",
  "email": "email@test.com",
  "orgId": 2
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Team created","teamId":2}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否
- 409 — チーム名が既に存在

チームの更新

```
PUT /api/teams/:id
```

name および email フィールドのみ更新できます。

リクエストの例

```
PUT /api/teams/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "MyTestTeam",
  "email": "email@test.com"
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Team updated"}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チーム未検出
- 409 — チーム名が既に存在

Id によるチームの削除

```
DELETE /api/teams/:id
```

リクエストの例

```
DELETE /api/teams/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Team deleted"}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チーム未検出

チームメンバーの取得

```
GET /api/teams/:teamId/members
```

リクエストの例

```
GET /api/teams/1/members HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId": 1,
```

```
    "teamId": 1,
    "userId": 3,
    "email": "user1@email.com",
    "login": "user1",
    "avatarUrl": "\/avatar\/1b3c32f6386b0185c40d359cdc733a7"
  },
  {
    "orgId": 1,
    "teamId": 1,
    "userId": 2,
    "email": "user2@email.com",
    "login": "user2",
    "avatarUrl": "\/avatar\/cad3c68da76e45d10269e8ef02f8e7"
  }
]
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否

チームメンバーを追加する

```
POST /api/teams/:teamId/members
```

リクエストの例

```
POST /api/teams/1/members HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "userId": 2
}
```

レスポンスの例

```
HTTP/1.1 200
```

```
Content-Type: application/json

{"message":"Member added to Team"}
```

ステータスコード:

- 200 — 作成済み
- 400 — チーム内にユーザーが既に存在
- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チーム未検出

チームのメンバーの削除

```
DELETE /api/teams/:teamId/members/:userId
```

リクエストの例

```
DELETE /api/teams/2/members/3 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Team Member removed"}
```

ステータスコード:

- 200 — 作成済み
- 401 — 未許可
- 403— アクセス許可拒否
- 404 — チームが未検出/チームメンバーが未検出

チーム設定の取得

```
GET /api/teams/:teamId/preferences
```

リクエストの例

```
GET /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "theme": "",
  "homeDashboardId": 0,
  "timezone": ""
}
```

チーム設定の更新

```
PUT /api/teams/:teamId/preferences
```

リクエストの例

```
PUT /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark",
  "homeDashboardId": 39,
  "timezone": "utc"
}
```

JSON ボディスキーマ:

- theme — `light`、`dark` のにずれか、またはデフォルトテーマを使用する空の文字列を指定します。
- homeDashboardId — ダッシュボードの数値の `:id` です。デフォルトは 0 です。
- [タイムゾーン] — `utc`、`browser` のにずれか、またはデフォルトを使用する空の文字列を指定します。

パラメータを省略すると、現在の値はシステムのデフォルト値に置き換えられます。

レスポンスの例

```
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{
  "message":"Preferences updated"
}
```

ユーザー API

ユーザー API を使用して、Amazon Managed Grafana ワークスペース内のユーザーを使用します。

> **Note**
>
> Amazon Managed Grafana ワークスペースで Grafana API を使用するには、有効な Grafana API トークンが必要です。このトークンは API リクエストの `Authorization` フィールドに含めます。API コールを認証するトークンの作成方法については、「トークンを使用した認証」を参照してください。

ユーザーがメンバーであるチームの取得

```
GET /api/user/teams
```

リクエストの例

```
GET /api/user/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "orgId": 1,
    "name": "MyTestTeam",
    "email": "",
    "avatarUrl": "\/avatar\/3f49c15916554246daa714b9bd0ee3",
    "memberCount": 1
  }
]
```

スナップショットのリストの取得

実際のユーザーの指定ダッシュボードにスター付けします。

```
POST /api/user/stars/dashboard/:dashboardId
```

リクエストの例

```
POST /api/user/stars/dashboard/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Dashboard starred!"}
```

ダッシュボードのスター解除

実際のユーザーの指定ダッシュボードのスターを解除します。

```
DELETE /api/user/stars/dashboard/:dashboardId
```

リクエストの例

```
DELETE /api/user/stars/dashboard/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Dashboard unstarred"}
```

実際のユーザーの認証トークンの取得

```
GET /api/user/auth-tokens
```

リクエストの例

```
GET /api/user/auth-tokens HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 361,
    "isActive": true,
    "clientIp": "127.0.0.1",
    "browser": "Chrome",
    "browserVersion": "72.0",
    "os": "Linux",
```

```
    "osVersion": "",
    "device": "Other",
    "createdAt": "2019-03-05T21:22:54+01:00",
    "seenAt": "2019-03-06T19:41:06+01:00"
  },
  {
    "id": 364,
    "isActive": false,
    "clientIp": "127.0.0.1",
    "browser": "Mobile Safari",
    "browserVersion": "11.0",
    "os": "iOS",
    "osVersion": "11.0",
    "device": "iPhone",
    "createdAt": "2019-03-06T19:41:19+01:00",
    "seenAt": "2019-03-06T19:41:21+01:00"
  }
]
```

実際のユーザーの認証トークンの取り消し

```
POST /api/user/revoke-auth-token
```

実際のユーザーの指定認証トークン (デバイス) を取り消します。発行済み認証トークン (デバイス) のユーザーはログインしなくなり、次のアクティビティで再度認証を受ける必要があります。

リクエストの例

```
POST /api/user/revoke-auth-token HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "authTokenId": 364
}
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
```

```
{
  "message": "User auth token revoked"
}
```

オブザーバビリティソリューション

Amazon Managed Grafana を使用して、システムまたはアプリケーションをモニタリングできます。 AWS は、さまざまな種類のコンピューティングシステムの基本モニタリング設定を作成するのに役立つソリューションを提供します。これらのソリューションは、Amazon Managed Grafana の Amazon Managed Service for Prometheus を使用して、アプリケーションまたはサービスに関するインサイトを提供する Amazon EKS ソリューションをモニタリングします。

Amazon EKS インフラストラクチャ、Amazon EKS で実行されている Java 仮想マシン (JVM) アプリケーション、Amazon EKS の JVM で実行されている Apache Kafka アプリケーションをモニタリングするためのソリューションがあります。

オブザーバビリティソリューション

- Amazon Managed Grafana で Amazon EKS インフラストラクチャをモニタリングするためのソリューション
- Amazon Managed Grafana で JVM アプリケーションをモニタリングするためのソリューション
- Amazon Managed Grafana で Kafka アプリケーションをモニタリングするためのソリューション

Amazon Managed Grafana で Amazon EKS インフラストラクチャをモニタリングするためのソリューション

Amazon Elastic Kubernetes Service インフラストラクチャのモニタリングは、Amazon Managed Grafana を使用する最も一般的なシナリオの 1 つです。このページでは、このシナリオのソリューションを提供するテンプレートについて説明します。ソリューションは [AWS Cloud Development Kit (AWS CDK)] または [Terraform] を使用してインストールできます。

このソリューションは以下を設定します。

- Amazon Managed Service for Prometheus ワークスペースで Amazon EKS クラスターからのメトリクスを保存し、メトリクスをスクレイピングしてそのワークスペースにプッシュするマネージドコレクターを作成します。詳細については、AWS 「マネージドコレクターによるメトリクスの取り込み」を参照してください。
- CloudWatch エージェントを使用して、Amazon EKS クラスターからログを収集します。ログは CloudWatch に格納され、Amazon Managed Grafana によってクエリされます。詳細については、「Amazon EKS の口グ記録」を参照してください。

- Amazon Managed Grafana ワークスペースでこれらのログとメトリクスをプルして、クラスターのモニタリングに役立つダッシュボードとアラートを作成します。

このソリューションを適用すると、以下のようなダッシュボードとアラートが作成されます。

- Amazon EKS クラスターの全体的な健全性を評価します。
- Amazon EKS コントロールプレーンの健全性とパフォーマンスを表示します。
- Amazon EKS データプレーンの健全性とパフォーマンスを表示します。
- Kubernetes 名前空間全体の Amazon EKS ワークロードのインサイトを表示します。
- CPU、メモリ、ディスク、ネットワーク使用量などの、名前空間全体のリソース使用量を表示します。

このソリューションについて

このソリューションは、Amazon EKS クラスターのためにメトリクスを提供するように Amazon Managed Grafana ワークスペースを設定します。メトリクスはダッシュボードとアラートを生成するために使用されます。

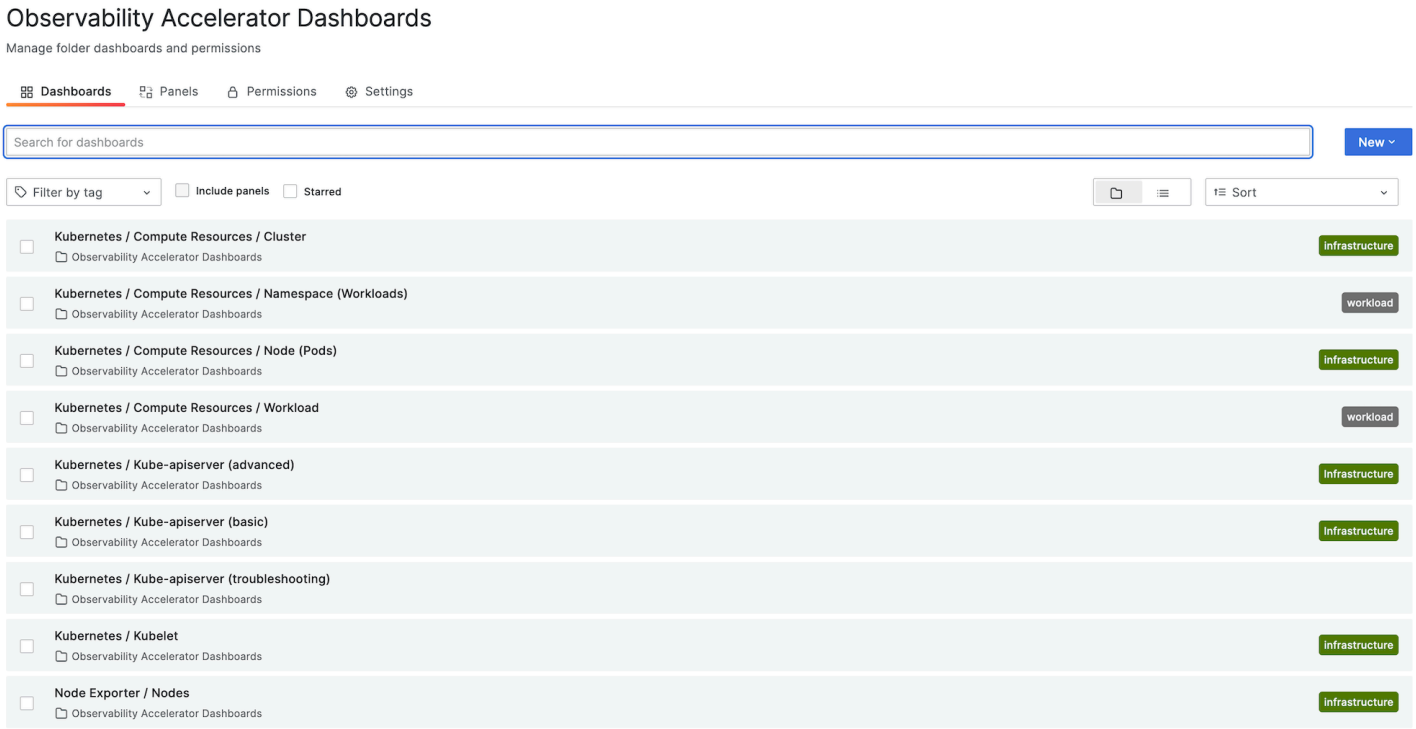
このメトリクスは、Kubernetes コントロールとデータプレーンのヘルスとパフォーマンスに関するインサイトを提供するため、Amazon EKS クラスターをより効果的に運用するのに役立ちます。Amazon EKS クラスターに関して、リソース使用状況の詳細なモニタリングなど、ノードレベルからポッド、Kubernetes レベルまで把握できます。

このソリューションには予測機能と修正機能の両方が用意されています。

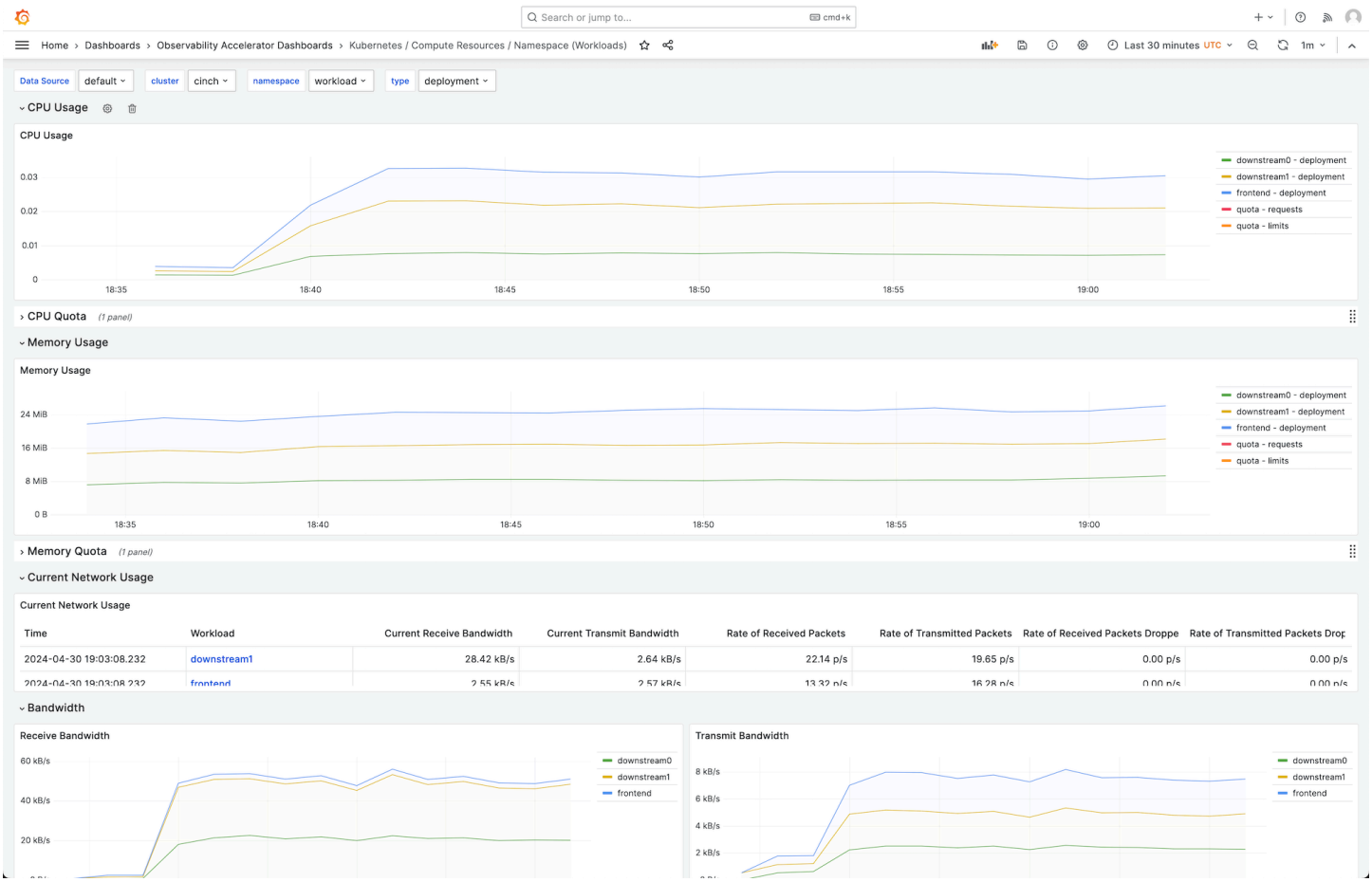
- 予測機能には以下が含まれます。
  - スケジューリングの決定を推進してリソース効率を管理します。例えば、Amazon EKS クラスターの内部ユーザーにパフォーマンスと信頼性の SLA を提供するには、過去の使用状況の追跡に基づいて、ワークロードに十分な CPU リソースとメモリリソースを割り当てることができます。
  - 使用状況の予測: ノード、[Amazon EBS でサポートされる永続的ボリューム]、[Application Load Balancer] などの Amazon EKS クラスターリソースの現在の使用状況に基づいて、同様の需要を持つ新製品やプロジェクトなどを事前に計画できます。
  - 潜在的な問題の早期検出: 例えば、Kubernetes 名前空間レベルでリソース消費の傾向を分析することで、ワークロードの使用状況の季節性を理解できます。
- 修正機能には以下が含まれます。

- インフラストラクチャと Kubernetes ワークロードレベルでの問題の平均検出時間 (MTTD) が短縮されます。例えば、トラブルシューティングダッシュボードを見ると、何が問題だったかに関する仮説をすぐにテストして排除できます。
- スタック内のどこで問題が発生しているか判断します。例えば、Amazon EKS コントロールプレーンは によって完全に管理 AWS されており、API サーバーが過負荷になったり、接続が影響を受けたりすると、Kubernetes デプロイの更新などの特定のオペレーションが失敗する可能性があります。

次の図はソリューションのダッシュボードフォルダのサンプルを示しています。

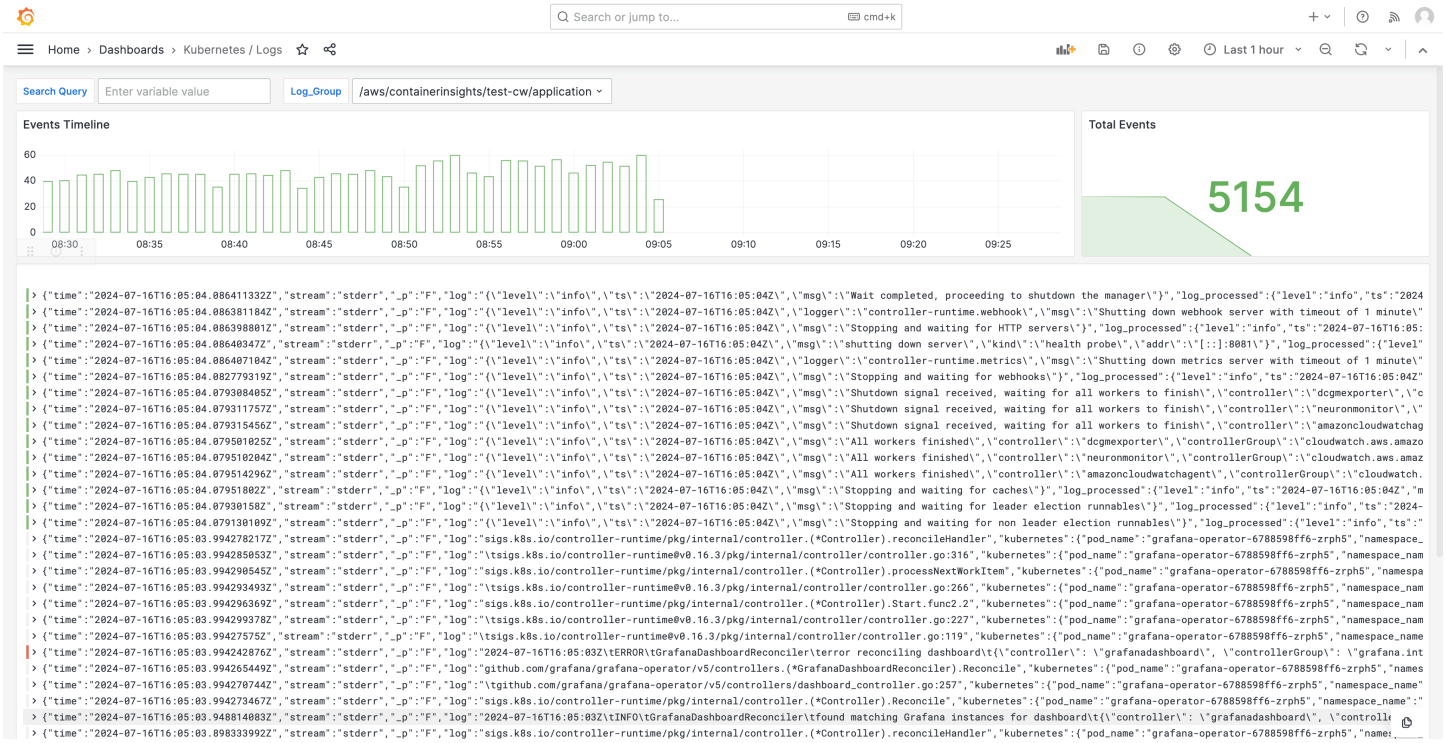


ダッシュボードを選択して詳細を表示できます。例えば、ワークロードのコンピューティングリソースの表示を選択すると、次のイメージに示すようなダッシュボードが表示されます。



メトリクスは 1 分間のスクレイプ間隔でスクレイプされます。ダッシュボードには、特定のメトリクスに基づいて、1 分、5 分、またはそれ以上に集約されたメトリクスが表示されます。

ログはダッシュボードにも表示されるため、ログをクエリおよび分析して、問題の根本原因を見つけることができます。次の画像は、ログダッシュボードの例を示しています。



このソリューションで追跡されるメトリクスのリストについては、「追跡されるメトリクスのリスト」を参照してください。

ソリューションによって作成されたアラートのリストについては、「作成済みアラートのリスト」を参照してください。

コスト

このソリューションは、ワークスペース内にリソースを作成して使用します。作成したリソースの標準使用量に対して課金されますが、これには以下が含まれます。

- ユーザーによる Amazon Managed Grafana ワークスペースアクセス。料金に関する詳細については、「Amazon Managed Grafana の料金」を参照してください。
- Amazon Managed Service for Prometheus エージェントレスコレクターの使用、メトリクス分析 (クエリサンプル処理) などの、Amazon Managed Service for Prometheus メトリクスの取り込みとストレージ。このソリューションで使用されるメトリクス数は、Amazon EKS クラスターの設定と使用状況によって異なります。

  CloudWatch を使用して、Amazon Managed Service for Prometheus で取り込みメトリクスとストレージメトリクスを表示できます。詳細については、Amazon Managed Service for Prometheus ユーザーガイドの [CloudWatch メトリクス] を参照してください。

Amazon Managed Service for Prometheus の料金ページでは、料金計算ツールを使用してコストを見積もることができます。メトリクス数は、クラスター内のノード数と、アプリケーションが生成するメトリクスによって異なります。

- CloudWatch Logs の取り込み、ストレージ、分析。デフォルトでは、ログ保持は有効期限が切れないように設定されています。これは CloudWatch 内で調整できます。料金の詳細については、「「Amazon CloudWatch 料金表」を参照してください。
- ネットワークコスト。クロスアベイラビリティーゾーン、リージョン、またはその他のトラフィックに対して標準 AWS ネットワーク料金が発生する場合があります。

各製品の料金ページから利用できる料金計算ツールは、ソリューションの潜在的コストを理解するのに役立ちます。以下の情報は、Amazon EKS クラスターと同じアベイラビリティーゾーンで実行されているソリューションの、基本コストを取得するのに役立ちます。

製品	計算メトリクス	値
Amazon Managed Service for Prometheus	アクティブなシリーズ	8000 (ベース) 15,000 (ノードあたり)
	平均収集間隔	60 (秒)
Amazon Managed Service for Prometheus (マネージドコレクター)	コレクター数	1
	サンプル数	15 (ベース) 150 (ノードあたり)
	ルールの数	161
	平均ルール抽出間隔	60 (秒)
Amazon Managed Grafana	アクティブなエディタ/管理者の数	1 (またはユーザーに基づき、さらに多く)
CloudWatch (Logs)	標準 Logs: データインジェスト	24.5 GB (ベース)

製品	計算メトリクス	値
		0.5 GB (ノードあたり)
	ログストレージ/アーカイブ (標準 Logs と Vended Logs)	ログを保存する: 1 か月の保持を前提
	スキャンされた予想ログデータ	Grafana からの各ログインサイトクエリは、指定期間のグループからのすべてのログコンテンツをスキャンします。

これらの数値は、追加ソフトウェアなしで EKS を実行するソリューションのベース番号です。これにより基本コストの見積もりが得られます。また、ネットワーク使用コストも削減されます。これは、Amazon Managed Grafana ワークスペース、Amazon Managed Service for Prometheus ワークスペース、Amazon EKS クラスターが同じアベイラビリティーゾーン、 AWS リージョンおよび VPN にあるかどうかによって異なります。

> **Note**
>
> このテーブルの項目に (base) 値とリソースあたりの値 (例: (per node)) が含まれている場合は、ベース値にリソースあたりの値を加えて、そのリソースの数を乗じる必要があります。例えば、[平均アクティブ時系列] の場合、8000 + the number of nodes in your cluster * 15,000 の数値を入力します。ノードが 2 つある場合は、38,000 と入力し、これは 8000 + (2 * 15,000) です。

前提条件

このソリューションを使用する前に、次のことを行う必要があります。

1. モニタリングする [Amazon Elastic Kubernetes Service クラスターを作成する]、またはそのクラスターがあるいる必要があります。そのクラスターには少なくとも 1 つのノードが必要です。クラスターには、プライベートアクセスを含めるように設定された API サーバーエンドポイントアクセスが必要です (パブリックアクセスを許可することもできます)。

   認証モードに API アクセスを含める必要があります (API または API_AND_CONFIG_MAP に設定)。これによりソリューションデプロイでアクセスエントリを使用できます。

クラスターに以下をインストールする必要があります (コンソール経由でクラスターを作成する場合はデフォルトで true、 AWS API または を使用してクラスターを作成する場合は追加する必要があります AWS CLI) 。 AWS CNI、CoreDNS、Kube-proxy AddOns。

クラスター名を保存して後で指定します。これは Amazon EKS コンソールのクラスター詳細にあります。

> **Note**
>
> Amazon EKS クラスターの作成方法の詳細については、「Amazon EKS の使用方法」を参照してください。

2. Amazon EKS クラスター AWS アカウント と同じ に Amazon Managed Service for Prometheus ワークスペースを作成する必要があります。詳細については、[Amazon Managed Service for Prometheus ユーザーガイド] の [ワークスペースの作成] を参照してください。

   Amazon Managed Service for Prometheus ワークスペース ARN を保存して後で指定します。

3. Grafana バージョン 9 以降の Amazon Managed Grafana ワークスペースは、Amazon EKS クラスターと同じ に作成する必要があります。 AWS リージョン 新しいワークスペースの作成の詳細については、「Amazon Managed Grafana ワークスペースを作成する」を参照してください。

   ワークスペースロールには、Amazon Managed Service for Prometheus および Amazon CloudWatch API にアクセスするためのアクセス許可が必要です。この許可を付与する最も簡単な方法は、[サービス管理のアクセス許可]を使用して、Amazon Managed Service for Prometheus と CloudWatch を選択することです。AmazonPrometheusQueryAccess ポリシーと AmazonGrafanaCloudWatchAccess ポリシーをワークスペース IAM ロールに手動で追加することもできます。

   Amazon Managed Grafana ワークスペース ID とエンドポイントを保存して後で指定します。ID は `g-123example` の形式になります。ID とエンドポイントは Amazon Managed Grafana コンソールにあります。エンドポイントはワークスペースの URL で、ID が含まれます。例えば、`https://g-123example.grafana-workspace.<region>.amazonaws.com/`。

4. Terraform でソリューションをデプロイするには、アカウントからアクセスできる [Amazon S3 バケット] を作成する必要があります。これはデプロイの Terraform 状態ファイルを保存するために使用されます。

   Amazon S3 バケット ID を保存して後で指定します。

5. Amazon Managed Service for Prometheus のアラートルールを表示するには、Amazon Managed Grafana ワークスペースの [Grafana アラート]を有効にする必要があります。

   さらに、Amazon Managed Grafana には Prometheus リソースに対する次のアクセス許可が必要です。これらのアクセス許可は、AWS データソースの Amazon Managed Grafana アクセス許可とポリシー で説明されているサービス管理ポリシーとカスタマー管理ポリシーのいずれかに追加する必要があります。

   - `aps:ListRules`
   - `aps:ListAlertManagerSilences`
   - `aps:ListAlertManagerAlerts`
   - `aps:GetAlertManagerStatus`
   - `aps:ListAlertManagerAlertGroups`
   - `aps:PutAlertManagerSilences`
   - `aps:DeleteAlertManagerSilence`

> **Note**
>
> ソリューションのセットアップが必ず必要なわけではありませんが、作成されたダッシュボードにアクセスするには、Amazon Managed Grafana ワークスペースでユーザー認証を設定する必要があります。詳細については、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。

このソリューションの使用

このソリューションは、Amazon EKS クラスターからのメトリクスのレポートとモニタリングをサポートする AWS ようにインフラストラクチャを設定します。AWS Cloud Development Kit (AWS CDK) または Terraform を使用してインストールできます。

Using AWS CDK

このソリューションが提供される方法の 1 つは、 AWS CDK アプリケーションとして提供されることです。使用するリソースに関する情報を提供すると、ソリューションはスクレイパー、ログ、ダッシュボードを作成します。

> **Note**
>
> ここでのステップでは、 AWS CLI、、および Node.js AWS CDKと NPM の両方がインストールされている環境があることを前提としています。make と を使用してbrew、ビルドやその他の一般的なアクションを簡素化します。

このソリューションを使用して で Amazon EKS クラスターをモニタリングするには AWS CDK

1. 前提条件のすべてのステップを完了済みであることを確認してください。
2. Amazon S3 からソリューションのファイルすべてをダウンロードします。ファイルは s3://aws-observability-solutions/EKS/OSS/CDK/v3.0.0/iac にあり、次の Amazon S3 コマンドを使用してダウンロードできます。コマンドライン環境のフォルダから、このコマンドを実行します。

   ```
   aws s3 sync s3://aws-observability-solutions/EKS/OSS/CDK/v3.0.0/iac/ .
   ```

   これらのファイルを修正する必要はありません。
3. コマンドライン環境で (ソリューションファイルをダウンロードしたフォルダから)、次のコマンドを実行します。

   必要な環境変数を設定します。*REGION*、*AMG_ENDPOINT*、*EKS_CLUSTER*、*AMP_ARN* を、 、Amazon Managed Grafana ワークスペースエンドポイント (の形式http://g-123example.grafana-workspace.us-east-1.amazonaws.com) AWS リージョン、Amazon EKS クラスター名、Amazon Managed Service for Prometheus ワークスペース ARN に置き換えます。

   ```
   export AWS_REGION=REGION
   export AMG_ENDPOINT=AMG_ENDPOINT
   export EKS_CLUSTER_NAME=EKS_CLUSTER
   export AMP_WS_ARN=AMP_ARN
   ```
4. Grafana HTTP API を呼び出すには、ADMIN アクセス許可を持つサービスアカウントトークンを作成する必要があります。詳細については、「サービスアカウントを使用した Grafana HTTP API の認証」を参照してください。トークンを作成するには、次のコマンド AWS CLI で を使用できます。*GRAFANA_ID* を ご使用の Grafana ワークスペースの ID に置き換える必要があります (g-123example 形式)。このキーは 7,200 秒または 2 時間後に期限が切れ

ます。必要に応じて、時間 (seconds-to-live) を変更できます。デプロイにかかる時間は 1 時間未満です。

```
GRAFANA_SA_ID=$(aws grafana create-workspace-service-account \
  --workspace-id GRAFANA_ID \
  --grafana-role ADMIN \
  --name grafana-operator-key \
  --query 'id' \
  --output text)

# creates a new token for calling APIs
export AMG_API_KEY=$(aws grafana create-workspace-service-account-token \
  --workspace-id $managed_grafana_workspace_id \
  --name "grafana-operator-key-$(date +%s)" \
  --seconds-to-live 7200 \
  --service-account-id $GRAFANA_SA_ID \
  --query 'serviceAccountToken.key' \
  --output text)
```

次のコマンドを使用して API キーを に追加 AWS CDK することで、 API キー AWS Systems Manager を で使用できるようにします。*AWS_REGION* を、ソリューションを実行するリージョンに置き換えます (us-east-1 形式)。

```
aws ssm put-parameter --name "/observability-aws-solution-eks-infra/grafana-api-
key" \
    --type "SecureString" \
    --value $AMG_API_KEY \
    --region AWS_REGION \
    --overwrite
```

5. 次の make コマンドを実行して、プロジェクトの他の依存関係をインストールします。

```
make deps
```

6. 最後に、 AWS CDK プロジェクトを実行します。

```
make build && make pattern aws-observability-solution-eks-infra-
$EKS_CLUSTER_NAME deploy
```

7. (オプション) スタックの作成が完了したら、他の各前提条件 (個別の Amazon Managed Grafana ワークスペースと Amazon Managed Service for Prometheus ワークスペースを含

む) を満たす限り、同じ環境を使用して、同じリージョン内の他の Amazon EKS クラスターにスタックのインスタンスをさらに作成できます。新しいパラメータで `export` コマンドを再定義する必要があります。

スタックの作成が完了すると、Amazon Managed Grafana ワークスペースにデータが入力されて、ダッシュボードに Amazon EKS クラスターのメトリクスが表示されます。スクレイパーがメトリクスの収集を開始するため、メトリクスが表示されるまでに数分かかります。

Using Terraform

このソリューションを提供する方法の 1 つは、Terraform ソリューションを使用することです。使用するリソースに関する情報を提供すると、ソリューションはスクレイパー、ログ、ダッシュボードを作成します。

このソリューションを使用して Terraform で Amazon EKS クラスターをモニタリングする方法

1. 前提条件のすべてのステップを完了済みであることを確認してください。
2. Amazon S3 からソリューションのファイルすべてをダウンロードします。ファイルは `s3://aws-observability-solutions/EKS/OSS/Terraform/v3.0.0/` にあり、次の Amazon S3 コマンドを使用してダウンロードできます。コマンドライン環境内のフォルダからこのコマンドを実行して、デプロイするフォルダにディレクトリを変更します。

   ```
   aws s3 sync s3://aws-observability-solutions/EKS/OSS/Terraform/v3.0.0/ .
   cd eks-monitoring
   ```

   これらのファイルを修正する必要はありません。
3. コマンドライン環境で (ソリューションファイルをダウンロードしたフォルダから)、次のコマンドを実行します。

   必要な環境変数を設定します。*REGION*、*AMG_ENDPOINT*、*EKS_CLUSTER*、*AMP_ARN*、および *S3_ID* を、新しいリソースをデプロイ AWS リージョン する (などus-east-1) 、Amazon Managed Grafana ワークスペースエンドポイント (の形式http://g-123example.grafana-workspace.us-east-1.amazonaws.com) 、Amazon EKS クラスター名、Amazon Managed Service for Prometheus ワークスペース ARN、および Amazon S3 バケット ID に置き換えます。

   ```
   export TF_VAR_aws_region=REGION
   export TF_VAR_amg_endpoint=AMG_ENDPOINT
   export TF_VAR_eks_cluster_name=EKS_CLUSTER
   ```

```
export TF_VAR_amp_ws_arn=AMP_ARN
export TF_VAR_s3_bucket_id=S3_ID
```

4. Grafana HTTP API を呼び出すには、ADMIN アクセス許可を持つサービスアカウントトークンを作成する必要があります。詳細については、「サービスアカウントを使用した Grafana HTTP API の認証」を参照してください。トークンを作成するには、次のコマンド AWS CLI で を使用できます。*GRAFANA_ID* を ご使用の Grafana ワークスペースの ID に置き換える必要があります (g-123example 形式)。このキーは 7,200 秒または 2 時間後に期限が切れます。必要に応じて、時間 (seconds-to-live) を変更できます。デプロイにかかる時間は 1 時間未満です。

```
GRAFANA_SA_ID=$(aws grafana create-workspace-service-account \
  --workspace-id GRAFANA_ID \
  --grafana-role ADMIN \
  --name grafana-operator-key \
  --query 'id' \
  --output text)

# creates a new token for running Terraform
export TF_VAR_grafana_api_key=$(aws grafana create-workspace-service-account-
token \
  --workspace-id $managed_grafana_workspace_id \
  --name "grafana-operator-key-$(date +%s)" \
  --seconds-to-live 7200 \
  --service-account-id $GRAFANA_SA_ID \
  --query 'serviceAccountToken.key' \
  --output text)
```

> **Note**
>
> 上記の最初のステップでは、すでにサービスアカウントをお持ちの場合はワークスペースのサービスアカウントを作成する必要はありません。この場合、*$GRAFANA_SA_ID* をご使用のサービスアカウントの ID に置き換えます。

5. 次の terraform コマンドを実行して、ソリューションで Terraform を初期化します。

```
terraform init -reconfigure \
-backend-config="bucket=${TF_VAR_s3_bucket_id}" \
-backend-config="region=${TF_VAR_aws_region}" \
-backend-config="key=state/${TF_VAR_eks_cluster_name}/terraform.tfstate"
```

6. 最後に、Terraform プロジェクトをデプロイします。

```
terraform apply
```

ソリューションの作成が完了すると、Amazon Managed Grafana ワークスペースにデータが入力されて、ダッシュボードに Amazon EKS クラスターのメトリクスが表示されます。スクレイパーがメトリクスの収集を開始するため、メトリクスが表示されるまでに数分かかります。

追跡されるメトリクスのリスト

このソリューションは Amazon EKS クラスターからメトリクスを収集するスクレイパーを作成します。これらのメトリクスは Amazon Managed Service for Prometheus に格納され、Amazon Managed Grafana ダッシュボードに表示されます。デフォルトでは、スクレイパーはクラスターが公開するすべての Prometheus 互換メトリクスを収集します。より多くのメトリクスを生成するクラスターにソフトウェアをインストールすると、収集されるメトリクスが増加します。必要に応じて、メトリクスをフィルタリングする設定でスクレイパーを更新して、メトリクス数を減らすことができます。

このソリューションでは、追加ソフトウェアをインストールしないベース Amazon EKS クラスター設定で、次のメトリクスを追跡します。

メトリクス	説明/目的
aggregator_unavailable_apiservice	APIService 名で分類された、使用不可としてマークされた APIServices のゲージ。
apiserver_admission_webhook_admission_duration_seconds_bucket	名前で識別され、各オペレーションと API リソースとタイプ (検証または承認) ごとに分割された、秒単位のアドミッションウェブフックのレイテンシーヒストグラム。
apiserver_current_inflight_requests	過去 1 秒のリクエストタイプごとの、現在使用されているこの apiserver の保留リクエスト制限の最大数。
apiserver_envelope_encryption_dek_cache_fill_percent	キャッシュされた DEK が現在占めているキャッシュスロットのパーセント。

メトリクス	説明/目的
apiserver_flowcontrol_current_executing_requests	API Priority and Fairness サブシステムの初期 (WATCH の場合) または任意の (WATCH 以外の場合) 実行ステージのリクエスト数。
apiserver_flowcontrol_rejected_requests_total	拒否された API Priority and Fairness サブシステムの初期 (WATCH の場合) または任意の (WATCH 以外の場合) 実行ステージのリクエスト数。
apiserver_flowcontrol_request_concurrency_limit	各優先度レベルに設定されている、実行シートの公称数。
apiserver_flowcontrol_request_execution_seconds_bucket	API Priority and Fairness サブシステムにおけるリクエスト実行の初期ステージ (WATCH の場合) または任意の (WATCH 以外の場合) ステージの持続期間の、バケット化ヒストグラム。
apiserver_flowcontrol_request_queue_length_after_enqueue_count	API Priority and Fairness サブシステムにおけるリクエスト実行の初期ステージ (WATCH の場合) または任意の (WATCH 以外の場合) ステージのカウント。
apiserver_request	API サーバーリクエストを示します。
apiserver_requested_deprecated_apis	API グループ、バージョン、リソース、サブリソース、removed_release 別に分類された、リクエストされた非推奨 API のゲージ。
apiserver_request_duration_seconds	verb、dry run 値、グループ、バージョン、リソース、サブリソース、スコープ、コンポーネントごとの、秒単位の応答遅延分布。

メトリクス	説明/目的
apiserver_request_duration_seconds_bucket	verb、dry run 値、グループ、バージョン、リソース、サブリソース、スコープ、コンポーネントごとの、秒単位の応答遅延分布のバケット化ヒストグラム。
apiserver_request_slo_duration_seconds	verb、dry run 値、グループ、バージョン、リソース、サブリソース、スコープ、コンポーネントごとの、サービスレベル目標 (SLO) の秒単位の応答遅延分布。
apiserver_request_terminations_total	apiserver が自己防衛で終了したリクエスト数。
apiserver_request_total	verb、dry run 値、グループ、バージョン、リソース、スコープ、コンポーネント、HTTP レスポンスコードごとに分割された apiserver リクエストのカウンター。
container_cpu_usage_seconds_total	cpu 累積消費時間。
container_fs_reads_bytes_total	読み取られたバイトの累積カウント。
container_fs_reads_total	完了した読み取りの累積カウント。
container_fs_writes_bytes_total	書き込まれたバイトの累積カウント。
container_fs_writes_total	完了した書き込みの累積カウント。
container_memory_cache	ページキャッシュメモリの合計。
container_memory_rss	RSS のサイズ。
container_memory_swap	コンテナスワップの使用状況。
container_memory_working_set_bytes	現在のワーキングセット。

メトリクス	説明/目的
container_network_receive_bytes_total	受信したバイトの累積カウント。
container_network_receive_packets_dropped_total	受信中にドロップされたパケットの累積カウント。
container_network_receive_packets_total	受信したパケットの累積カウント。
container_network_transmit_bytes_total	送信されたバイトの累積カウント。
container_network_transmit_packets_dropped_total	送信中にドロップされたパケットの累積カウント。
container_network_transmit_packets_total	送信されたパケットの累積カウント。
etcd_request_duration_seconds_bucket	オペレーションとオブジェクトタイプごとの、秒単位の etcd リクエストのレイテンシーのバケット化ヒストグラム。
go_goroutines	現在存在する goroutine 数。
go_threads	作成された OS スレッド数。
kubelet_cgroup_manager_duration_seconds_bucket	cgroup マネージャーオペレーションの、秒単位の持続期間のバケット化ヒストグラム。メソッドにより分類。
kubelet_cgroup_manager_duration_seconds_count	cgroup マネージャーオペレーションの秒単位の持続期間。メソッドにより分類。
kubelet_node_config_error	ノードで設定関連のエラーが発生した場合はこのメトリクスが真 (1) となり、それ以外の場合は偽 (0) となります。
kubelet_node_name	ノードの名前。カウントは常に 1 です。

メトリクス	説明/目的
kubelet_pleg_relist_duration_seconds_bucket	PLEG でポッドを再び一覧表示するための、秒単位の持続時間のバケット化ヒストグラム。
kubelet_pleg_relist_duration_seconds_count	PLEG でポッドを再び一覧表示するための、秒単位の持続時間のカウント。
kubelet_pleg_relist_interval_seconds_bucket	PLEG で再び一覧表示するまでの間隔の、バケット化ヒストグラム。
kubelet_pod_start_duration_seconds_count	kubelet がポッドを初めて確認したときからポッドの実行が開始されるまでの、秒単位の持続時間カウント。
kubelet_pod_worker_duration_seconds_bucket	1 つのポッドを同期するための、秒単位の持続時間のバケット化ヒストグラム。作成、更新、同期のオペレーションタイプにより分類。
kubelet_pod_worker_duration_seconds_count	1 つのポッドを同期する、秒単位の持続時間カウント。作成、更新、同期のオペレーションタイプにより分類。
kubelet_running_containers	現在実行中のコンテナ数。
kubelet_running_pods	実行中のポッドサンドボックスを持つポッド数。
kubelet_runtime_operations_duration_seconds_bucket	ランタイムオペレーションの、秒単位の持続時間のバケット化ヒストグラム。オペレーションタイプにより分類。
kubelet_runtime_operations_errors_total	オペレーションタイプごとのランタイムオペレーションエラーの累積数。
kubelet_runtime_operations_total	オペレーションタイプごとのランタイムオペレーションの累積数。

メトリクス	説明/目的
kube_node_status_allocatable	ポッドに割り当て可能なリソースの量 (システムデーモン用に一部を予約後)。
kube_node_status_capacity	ノードで使用できるリソースの合計量。
kube_pod_container_resource_limits (CPU)	コンテナがリクエストした制限リソース数。
kube_pod_container_resource_limits (Memory)	コンテナがリクエストした制限リソース数。
kube_pod_container_resource_requests (CPU)	コンテナがリクエストしたリクエストリソース数。
kube_pod_container_resource_requests (Memory)	コンテナがリクエストしたリクエストリソース数。
kube_pod_owner	ポッドの所有者に関する情報。
kube_resourcequota	Kubernetes のリソースクォータでは、名前空間内の CPU、メモリ、ストレージなどのリソースに使用制限を適用します。
node_cpu	コアあたりの使用量と合計使用量を含む、ノードの CPU 使用量メトリクス。
node_cpu_seconds_total	各モードで消費された CPU の秒数。
node_disk_io_time_seconds	ノードがディスクで I/O オペレーションを実行した累積合計時間。
node_disk_io_time_seconds_total	ノードがディスクで I/O オペレーションを実行した合計時間。
node_disk_read_bytes_total	ノードがディスクから読み込んだ合計バイト数。
node_disk_written_bytes_total	ノードがディスクに書き込んだ合計バイト数。

メトリクス	説明/目的
node_filesystem_avail_bytes	Kubernetes クラスター内のノードのファイルシステムで使用可能なスペースのバイト量。
node_filesystem_size_bytes	ノード上のファイルシステムの合計サイズ。
node_load1	ノードの CPU 使用率の 1 分間の負荷平均。
node_load15	ノードの CPU 使用率の 15 分間の負荷平均。
node_load5	ノードの CPU 使用率の 5 分間の負荷平均。
node_memory_Buffers_bytes	ノードのオペレーティングシステムがバッファキャッシュに使用したメモリ量。
node_memory_Cached_bytes,	ノードのオペレーティングシステムがディスクキャッシュに使用したメモリ量。
node_memory_MemAvailable_bytes	アプリケーションとキャッシュで使用できるメモリ量。
node_memory_MemFree_bytes	ノードで使用可能な空きメモリ量。
node_memory_MemTotal_bytes	ノードで使用可能な物理メモリ合計量。
node_network_receive_bytes_total	ノードによって、ネットワーク経由で受信されている合計バイト数。
node_network_transmit_bytes_total	ノードによって、ネットワーク経由で送信されている合計バイト数。
process_cpu_seconds_total	ユーザー合計数および秒単位のシステム CPU 消費時間合計。
process_resident_memory_bytes	バイト単位の常駐メモリサイズ。
rest_client_requests_total	ステータスコード、メソッド、ホストでパーティション分割された、HTTP リクエスト数。

メトリクス	説明/目的
rest_client_request_duration_seconds_bucket	秒単位のリクエストレイテンシーのバケット化ヒストグラム。verb とホストで分類。
storage_operation_duration_seconds_bucket	ストレージオペレーションの持続期間のバケット化ヒストグラム。
storage_operation_duration_seconds_count	ストレージオペレーションの持続期間カウント。
storage_operation_errors_total	ストレージオペレーション中の累積エラー数。
up	モニタリング対象のターゲット (ノードなど) が稼働しているかどうかを示すメトリクス。
volume_manager_total_volumes	ボリュームマネージャーにより管理されるボリューム合計数。
workqueue_adds_total	workqueue が処理する合計追加数。
workqueue_depth	現在の workqueue の深さ。
workqueue_queue_duration_seconds_bucket	リクエストされるまでに項目が workqueue に留まる時間を秒単位で示す、バケット化ヒストグラム。
workqueue_work_duration_seconds_bucket	workqueue の項目の処理にかかる時間を秒単位で示す、バケット化ヒストグラム。

作成済みアラートのリスト

次の表はこのソリューションが作成するアラートの一覧です。Amazon Managed Service for Prometheus ではアラートがルールとして作成され、Amazon Managed Grafana ワークスペースに表示されます。

Amazon Managed Service for Prometheus ワークスペースで[ルール設定ファイルの編集]を実行すると、ルールを追加または削除してルールを変更できます。

これらの 2 つのアラートは、一般的なアラートとは少し異なる処理される特別なアラートです。問題を警告する代わりに、システムのモニタリングに使用する情報を提供します。説明にはこれらのアラートの使用方法に関する詳細が含まれています。

アラート	説明および使用状況
`Watchdog`	これはアラートパイプライン全体が機能することを保証するためのアラートです。このアラートは常に発せられるため、常にアラートマネージャーで発し、常に受信者に対して発する必要があります。これを通知メカニズムと統合して、このアラートが発していないときに通知を送信できます。例えば、PagerDuty では DeadMansSnitch 統合を使用できます。
`InfoInhibitor`	これは情報アラートを抑制するために使用するアラートです。それ自体では、情報レベルのアラートはノイズが非常に多い場合がありますが、他のアラートと組み合わせると現実的になります。このアラートは `severity=info` アラートがあるたびに発生し、重要度が `warning` または `critical` のアラートが同じ名前空間で発生し始めると、発生が停止します。このアラートは null レシーバーにルーティングされ、`severity=info` でアラートを抑制するように設定する必要があります。

次のアラートはシステムに関する情報または警告を提供します。

アラート	緊急度	説明
`NodeNetworkInterfaceFlapping`	`warning`	ネットワークインターフェイスが頻繁にステータスを変更しています。

アラート	緊急度	説明
NodeFilesystemSpaceFillingUp	warning	ファイルシステムのスペースが今後 24 時間以内に不足すると予測されます。
NodeFilesystemSpaceFillingUp	critical	ファイルシステムのスペースが今後 4 時間以内に不足すると予測されます。
NodeFilesystemAlmostOutOfSpace	warning	ファイルシステムの残りのスペースが 5% 未満です。
NodeFilesystemAlmostOutOfSpace	critical	ファイルシステムの残りのスペースが 3% 未満です。
NodeFilesystemFilesFillingUp	warning	ファイルシステムの inode が今後 24 時間以内に不足すると予測されます。
NodeFilesystemFilesFillingUp	critical	ファイルシステムの inode が今後 4 時間以内に不足すると予測されます。
NodeFilesystemAlmostOutOfFiles	warning	ファイルシステムの残りの inode が 5% 未満です。
NodeFilesystemAlmostOutOfFiles	critical	ファイルシステムの残りの inode が 3% 未満です。
NodeNetworkReceiveErrs	warning	ネットワークインターフェイスが多くの受信エラーを報告しています。
NodeNetworkTransmitErrs	warning	ネットワークインターフェイスが多くの送信エラーを報告しています。

アラート	緊急度	説明
NodeHighNumberConntrackEntriesUsed	warning	conntrack エントリ数が制限に近づいています。
NodeTextFileCollectorScrapeError	warning	Node Exporter のテキストファイルコレクターがスクレイピングに失敗しました。
NodeClockSkewDetected	warning	クロックスキューが検出されました。
NodeClockNotSynchronizing	warning	クロックが同期していません。
NodeRAIDDegraded	critical	RAID アレイが劣化
NodeRAIDDiskFailure	warning	RAID アレイが失敗したデバイス
NodeFileDescriptorLimit	warning	カーネルはまもなくファイル記述子制限を使い果たすと予測されます。
NodeFileDescriptorLimit	critical	カーネルはまもなくファイル記述子制限を使い果たすと予測されます。
KubeNodeNotReady	warning	ノードの準備ができていません。
KubeNodeUnreachable	warning	ノードに到達できません。
KubeletTooManyPods	info	Kubelet は最大能力で実行されています。
KubeNodeReadinessFlapping	warning	ノードの準備状況ステータスがフラッピングしています。

アラート	緊急度	説明
KubeletPlegDurationHigh	warning	Kubelet ポッドライフサイクルイベントジェネレータの再リストに時間がかかりすぎています。
KubeletPodStartUpLatencyHigh	warning	Kubelet ポッドスタートアップレイテンシーが高すぎます。
KubeletClientCertificateExpiration	warning	Kubelet クライアント証明書の有効期限が近づいています。
KubeletClientCertificateExpiration	critical	Kubelet クライアント証明書の有効期限が近づいています。
KubeletServerCertificateExpiration	warning	Kubelet サーバー証明書の有効期限が近づいています。
KubeletServerCertificateExpiration	critical	Kubelet サーバー証明書の有効期限が近づいています。
KubeletClientCertificateRenewalErrors	warning	Kubelet はクライアント証明書の更新に失敗しました。
KubeletServerCertificateRenewalErrors	warning	Kubelet はサーバー証明書の更新に失敗しました。
KubeletDown	critical	Prometheus のターゲット検出からターゲットが消えました。
KubeVersionMismatch	warning	異なるセマンティックバージョンの Kubernetes コンポーネントが実行中です。
KubeClientErrors	warning	Kubernetes API サーバークライアントでエラーが発生しています。

アラート	緊急度	説明
KubeClientCertificateExpiration	warning	クライアント証明書の有効期限が近づいています。
KubeClientCertificateExpiration	critical	クライアント証明書の有効期限が近づいています。
KubeAggregatedAPIErrors	warning	Kubernetes 集計 API でエラーが報告されました。
KubeAggregatedAPIDown	warning	Kubernetes 集計 API がダウンしています。
KubeAPIDown	critical	Prometheus のターゲット検出からターゲットが消えました。
KubeAPITerminatedRequests	warning	kubernetes apiserver が受信リクエストの {{ $value | humanizePercentage }} を終了しました。
KubePersistentVolumeFillingUp	critical	永続ボリュームがいっぱいです。
KubePersistentVolumeFillingUp	warning	永続ボリュームがいっぱいです。
KubePersistentVolumeInodesFillingUp	critical	永続ボリュームの Inode がいっぱいです。
KubePersistentVolumeInodesFillingUp	warning	永続ボリュームの Inode がいっぱいです。
KubePersistentVolumeErrors	critical	永続ボリュームのプロビジョニングに問題があります。

アラート	緊急度	説明
KubeCPUOvercommit	warning	クラスターが CPU リソースリクエストをオーバーコミットしました。
KubeMemoryOvercommit	warning	クラスターがメモリリソースリクエストをオーバーコミットしました。
KubeCPUQuotaOvercommit	warning	クラスターが CPU リソースリクエストをオーバーコミットしました。
KubeMemoryQuotaOvercommit	warning	クラスターがメモリリソースリクエストをオーバーコミットしました。
KubeQuotaAlmostFull	info	名前空間のクォータがいっぱいになります。
KubeQuotaFullyUsed	info	名前空間のクォータが完全に使用されています。
KubeQuotaExceeded	warning	名前空間のクォータが制限を超えました。
CPUThrottlingHigh	info	プロセスが昇格 CPU スロットリングを発見しました。
KubePodCrashLooping	warning	ポッドがクラッシュループです。
KubePodNotReady	warning	ポッドが 15 分以上準備状態になっていません。
KubeDeploymentGenerationMismatch	warning	ロールバックの可能性に起因するデプロイ生成不一致

アラート	緊急度	説明
KubeDeploymentReplicasMismatch	warning	デプロイが予想レプリカ数と一致しません。
KubeStatefulSetReplicasMismatch	warning	StatefulSet が予想レプリカ数と一致していません。
KubeStatefulSetGenerationMismatch	warning	ロールバックの可能性に起因する StatefulSet 生成不一致
KubeStatefulSetUpdateNotRolledOut	warning	StatefulSet 更新がロールアウトしません。
KubeDaemonSetRolloutStuck	warning	DaemonSet のロールアウトがスタックしています。
KubeContainerWaiting	warning	ポッドコンテナが 1 時間以上待機
KubeDaemonSetNotScheduled	warning	DaemonSet ポッドがスケジュールされていません。
KubeDaemonSetMisScheduled	warning	DaemonSet ポッドのスケジュールが間違っています。
KubeJobNotCompleted	warning	ジョブが時間内に完了しませんでした
KubeJobFailed	warning	ジョブの完了に失敗しました。
KubeHpaReplicasMismatch	warning	HPA が希望レプリカ数と一致しません。
KubeHpaMaxedOut	warning	HPA が最大レプリカで実行されています

アラート	緊急度	説明
KubeStateMetricsListErrors	critical	kube-state-metrics でリストオペレーションのエラーが発生しています。
KubeStateMetricsWatchErrors	critical	kube-state-metrics で監視オペレーションのエラーが発生しています。
KubeStateMetricsShardingMismatch	critical	kube-state-metrics のシャード設定が間違っています。
KubeStateMetricsShardsMissing	critical	kube-state-metrics のシャードがありません。
KubeAPIErrorBudgetBurn	critical	API サーバーで過剰なエラーバジェットを消費しています。
KubeAPIErrorBudgetBurn	critical	API サーバーで過剰なエラーバジェットを消費しています。
KubeAPIErrorBudgetBurn	warning	API サーバーで過剰なエラーバジェットを消費しています。
KubeAPIErrorBudgetBurn	warning	API サーバーで過剰なエラーバジェットを消費しています。
TargetDown	warning	1 つ以上のターゲットがダウンしています。
etcdInsufficientMembers	critical	Etcd クラスターのメンバーが不十分です。

アラート	緊急度	説明
etcdHighNumberOfLeaderChanges	warning	Etcd クラスターでリーダーの変更数が多い。
etcdNoLeader	critical	Etcd クラスターにリーダーがありません。
etcdHighNumberOfFailedGRPCRequests	warning	Etcd クラスターで gRPC リクエストの失敗数が多い。
etcdGRPCRequestsSlow	critical	Etcd クラスターで gRPC リクエストが遅い。
etcdMemberCommunicationSlow	warning	Etcd クラスターメンバーの通信が遅い。
etcdHighNumberOfFailedProposals	warning	Etcd クラスターのプロポーザルの失敗数が多い。
etcdHighFsyncDurations	warning	Etcd クラスターの同期期間が長い。
etcdHighCommitDurations	warning	Etcd クラスターが予想コミット期間より長い。
etcdHighNumberOfFailedHTTPRequests	warning	Etcd クラスターが HTTP リクエストに失敗しました。
etcdHighNumberOfFailedHTTPRequests	critical	Etcd クラスターで HTTP リクエストの失敗数が多い。
etcdHTTPRequestsSlow	warning	Etcd クラスターで HTTP リクエストが遅い。
HostClockNotSynchronizing	warning	ホストクロックが同期していません。
HostOomKillDetected	warning	ホスト OOM 強制終了が検出されました。

トラブルシューティング

プロジェクトのセットアップが失敗する原因がいくつかあります。必ず以下の内容を確認してください。

- ソリューションのインストール前にすべての [前提条件] を完了する必要があります。
- ソリューションの作成やメトリクスへのアクセスを試みる前に、クラスターに少なくとも 1 つのノードを含める必要があります。
- Amazon EKS クラスターには、AWS CNI、CoreDNS、kube-proxy アドオンをインストールする必要があります。インストールしないとソリューションが正しく動作しません。これらは、コンソールを介してクラスターを作成するときにデフォルトでインストールされます。クラスターが AWS SDK を介して作成された場合は、インストールが必要になる場合があります。
- Amazon EKS ポッドのインストールがタイムアウトしました。これは、使用可能なノード容量が十分でないときに発生する場合があります。これらの問題には、次のような複数の原因があります。
  - Amazon EKS クラスターが Amazon EC2 ではなく Fargate で初期化されました。このプロジェクトでは Amazon EC2 が必須です。
  - ノードが汚染されているため使用できません。

    `kubectl describe node NODENAME | grep Taints` を使用するとテイントを確認できます。次に `kubectl taint node NODENAME TAINT_NAME-` でテイントを削除します。テイント名の後に必ず - を含めてください。
  - ノードが容量制限に達しました。この場合は新しいノードを作成、または容量を増やすことができます。
- Grafana にダッシュボードが表示されません: 間違った Grafana ワークスペース ID を使用しています。

  以下のコマンドを実行して Grafana に関する情報を取得してください。

  ```
  kubectl describe grafanas external-grafana -n grafana-operator
  ```

  正しいワークスペース URL の結果を確認できます。予想しているものでない場合は、正しいワークスペース ID で再デプロイします。

  ```
  Spec:
    External:
      API Key:
  ```

```
      Key:   GF_SECURITY_ADMIN_APIKEY
      Name:  grafana-admin-credentials
    URL:     https://g-123example.grafana-workspace.aws-region.amazonaws.com
Status:
  Admin URL:  https://g-123example.grafana-workspace.aws-region.amazonaws.com
  Dashboards:
    ...
```

- Grafana にダッシュボードが表示されません: 期限切れの API キーを使用しています。

  この場合の検索では、grafana 演算子を取得して、ログにエラーがないことを確認する必要があります。次のコマンドを使用して Grafana 演算子の名前を取得します。

  ```
  kubectl get pods -n grafana-operator
  ```

  次のような演算子名が返されます。

  ```
  NAME                                READY   STATUS    RESTARTS   AGE
  grafana-operator-1234abcd5678ef90   1/1     Running   0          1h2m
  ```

  次のコマンドで演算子名を使用します。

  ```
  kubectl logs grafana-operator-1234abcd5678ef90 -n grafana-operator
  ```

  次のようなエラーメッセージは、API キーの有効期限が切れたことを示しています。

  ```
  ERROR   error reconciling datasource    {"controller": "grafanadatasource",
   "controllerGroup": "grafana.integreatly.org", "controllerKind": "GrafanaDatasource",
   "GrafanaDatasource": {"name":"grafanadatasource-sample-amp","namespace":"grafana-
  operator"}, "namespace": "grafana-operator", "name": "grafanadatasource-sample-
  amp", "reconcileID": "72cfd60c-a255-44a1-bfbd-88b0cbc4f90c", "datasource":
   "grafanadatasource-sample-amp", "grafana": "external-grafana", "error": "status:
   401, body: {\"message\":\"Expired API key\"}\n"}
  github.com/grafana-operator/grafana-operator/controllers.
  (*GrafanaDatasourceReconciler).Reconcile
  ```

  この場合は、新しい API キーを作成してソリューションを再度デプロイします。問題が解決しない場合は、再びデプロイする前に次のコマンドを使用して強制的に同期できます。

  ```
  kubectl delete externalsecret/external-secrets-sm -n grafana-operator
  ```

- CDK インストール — SSM パラメータが欠けています。次のようなエラーが表示される場合は、cdk bootstrap を実行して再試行してください。

```
Deployment failed: Error: aws-observability-solution-eks-infra-$EKS_CLUSTER_NAME:
 SSM
parameter /cdk-bootstrap/xxxxxxx/version not found. Has the environment been
bootstrapped? Please run 'cdk bootstrap' (see https://docs.aws.amazon.com/cdk/latest/
guide/bootstrapping.html)
```

- OIDC プロバイダーが既に存在する場合はデプロイが失敗する可能性があります。次のような エラーが表示されます (この場合は、CDK インストール)。

```
| CREATE_FAILED | Custom::AWSCDKOpenIdConnectProvider | OIDCProvider/Resource/Default
Received response status [FAILED] from custom resource. Message returned:
EntityAlreadyExistsException: Provider with url https://
oidc.eks.REGION.amazonaws.com/id/PROVIDER ID already exists.
```

  この場合は、IAM ポータルに移動して OIDC プロバイダーを削除し、再試行してください。

- [Terraform のインストール] — `cluster-secretstore-sm failed to create kubernetes rest client for update of resource` と `failed to create kubernetes rest client for update of resource` を含むエラーメッセージが表示されます。

  このエラーは通常、External Secrets Operator が Kubernetes クラスターに未インストールまたは有効化されていないことを示します。これはソリューションのデプロイ時にインストールされますが、ソリューションが必要なときに準備できていない場合があります。

  以下のコマンドを使用すると、KFP SDK がインストールされたことを確認できます。

```
kubectl get deployments -n external-secrets
```

  インストールされている場合、オペレーターを完全に使用できるようになるまでに時間がかかることがあります。次のコマンドを実行すると、必要なカスタムリソース定義 (CRD) のステータスを確認できます。

```
kubectl get crds|grep external-secrets
```

  このコマンドは、`clustersecretstores.external-secrets.io` や `externalsecrets.external-secrets.io` など、外部シークレット演算子に関連する CRD を一覧表示するはずです。リストにない場合は、数分待ってからもう一度確認してください。

CRD が登録されたら、`terraform apply` を再度実行してソリューションをデプロイできます。

Amazon Managed Grafana で JVM アプリケーションをモニタリングするためのソリューション

Java 仮想マシン (JVM) で構築されたアプリケーションには、特殊なモニタリングニーズがあります。このページでは、Amazon EKS クラスターで実行されている JVM ベースのアプリケーションをモニタリングするためのソリューションを提供するテンプレートについて説明します。このソリューションは、 を使用してインストールできますAWS Cloud Development Kit (AWS CDK)。

> **Note**
>
> このソリューションは、JVM アプリケーションのモニタリングを提供します。JVM アプリケーションが特に Apache Kafka アプリケーションの場合は、代わりに、JVM と Kafka の両方のモニタリングを含む Kafka モニタリングソリューションを使用することを選択できます。

このソリューションは以下を設定します。

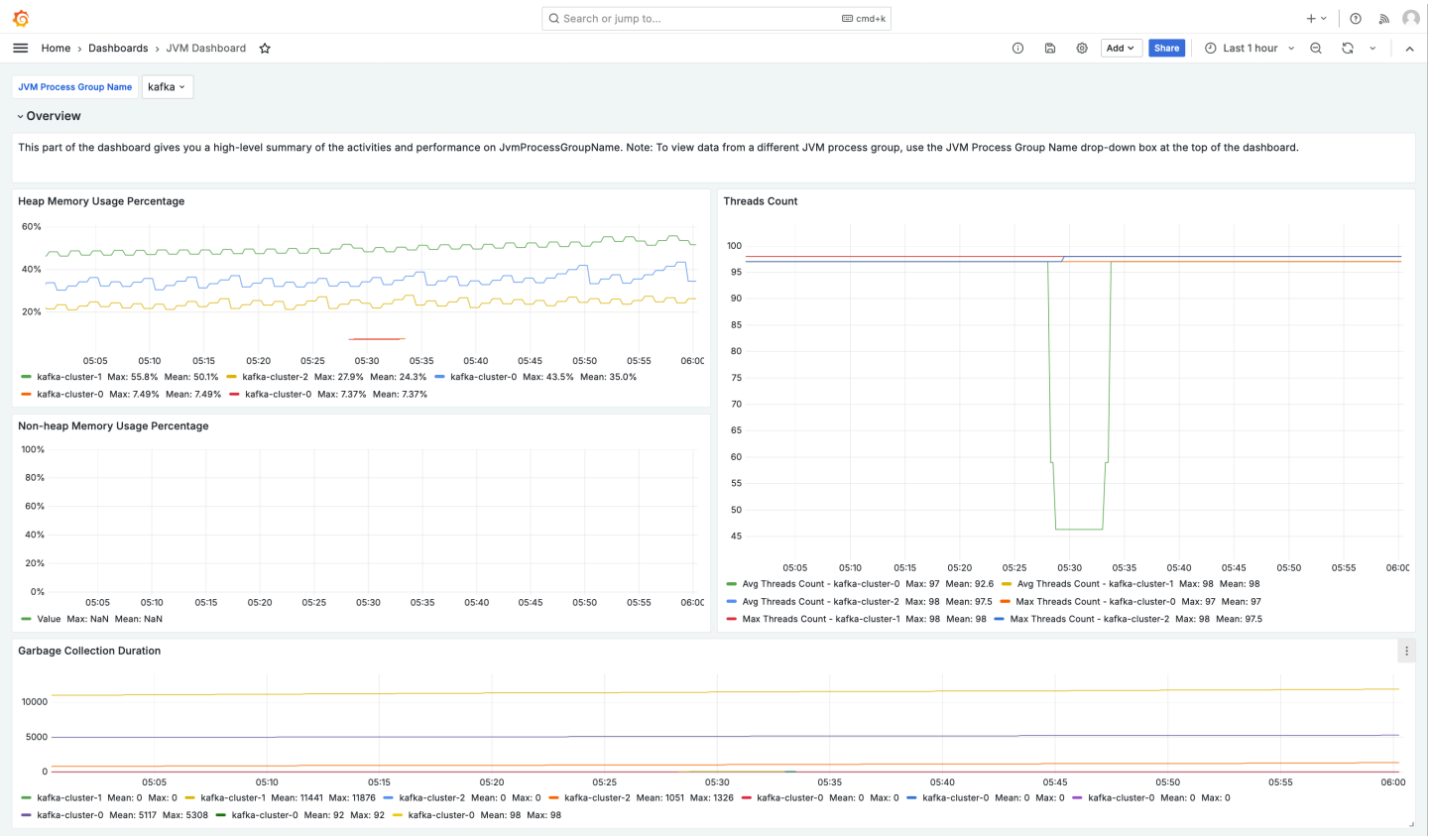
- Amazon EKS クラスターの Java 仮想マシン (JVM) メトリクスを保存する Amazon Managed Service for Prometheus ワークスペース。
- CloudWatch エージェントと CloudWatch エージェントアドオンを使用して特定の JVM メトリクスを収集します。メトリクスは、Amazon Managed Service for Prometheus ワークスペースに送信するように設定されています。
- これらのメトリクスをプルし、クラスターのモニタリングに役立つダッシュボードを作成するための Amazon Managed Grafana ワークスペース。

> **Note**
>
> このソリューションは、Amazon EKS で実行されているアプリケーションの JVM メトリクスを提供しますが、Amazon EKS メトリクスは含まれません。さらに、オブザーバビリティソリューションを使用して Amazon EKS をモニタリングし、Amazon EKS クラスターのメトリクスとアラートを確認することもできます。

このソリューションについて

このソリューションは、Java 仮想マシン (JVM) アプリケーションのメトリクスを提供するように Amazon Managed Grafana ワークスペースを設定します。メトリクスは、アプリケーションの正常性とパフォーマンスに関するインサイトを提供することで、アプリケーションの運用をより効果的に行うのに役立つダッシュボードを生成するために使用されます。

次の図は、このソリューションによって作成されたダッシュボードのサンプルを示しています。



メトリクスは 1 分間のスクレイプ間隔でスクレイプされます。ダッシュボードには、特定のメトリクスに基づいて、1 分、5 分、またはそれ以上に集約されたメトリクスが表示されます。

このソリューションで追跡されるメトリクスのリストについては、「追跡されるメトリクスのリスト」を参照してください。

コスト

このソリューションは、ワークスペース内にリソースを作成して使用します。作成したリソースの標準使用量に対して課金されますが、これには以下が含まれます。

- ユーザーによる Amazon Managed Grafana ワークスペースアクセス。料金に関する詳細については、「Amazon Managed Grafana の料金」を参照してください。
- Amazon Managed Service for Prometheus メトリクスの取り込みとストレージ、およびメトリクス分析 (クエリサンプル処理) 。このソリューションで使用されるメトリクスの数は、アプリケーションの設定と使用状況によって異なります。

  CloudWatch を使用して、Amazon Managed Service for Prometheus で取り込みメトリクスとストレージメトリクスを表示できます。詳細については、Amazon Managed Service for Prometheus ユーザーガイドの [CloudWatch メトリクス] を参照してください。

  Amazon Managed Service for Prometheus の料金ページでは、料金計算ツールを使用してコストを見積もることができます。メトリクス数は、クラスター内のノード数と、アプリケーションが生成するメトリクスによって異なります。
- ネットワークコスト。クロスアベイラビリティーゾーン、リージョン、またはその他のトラフィックに対して標準 AWS ネットワーク料金が発生する場合があります。

各製品の料金ページから利用できる料金計算ツールは、ソリューションの潜在的コストを理解するのに役立ちます。以下の情報は、Amazon EKS クラスターと同じアベイラビリティーゾーンで実行されているソリューションの、基本コストを取得するのに役立ちます。

製品	計算メトリクス	値
Amazon Managed Service for Prometheus	アクティブなシリーズ	50 (アプリケーションポッドあたり)
	平均収集間隔	60 (秒)
Amazon Managed Grafana	アクティブなエディタ/管理者の数	1 (またはユーザーに基づき、さらに多く)

これらの番号は、Amazon EKS で実行されている JVM アプリケーションのベース番号です。これにより基本コストの見積もりが得られます。アプリケーションにポッドを追加すると、次に示すようにコストが増加します。これらのコストにより、ネットワーク使用コストが削減されます。これは、Amazon Managed Grafana ワークスペース、Amazon Managed Service for Prometheus ワークスペース、Amazon EKS クラスターが同じアベイラビリティーゾーン AWS リージョン、および VPN にあるかどうかによって異なります。

前提条件

このソリューションを使用する前に、次のことを行う必要があります。

1. モニタリングする [Amazon Elastic Kubernetes Service クラスターを作成する]、またはそのクラスターがあるいる必要があります。そのクラスターには少なくとも 1 つのノードが必要です。クラスターには、プライベートアクセスを含めるように設定された API サーバーエンドポイントアクセスが必要です (パブリックアクセスを許可することもできます)。

   認証モードに API アクセスを含める必要があります (`API` または `API_AND_CONFIG_MAP` に設定)。これによりソリューションデプロイでアクセスエントリを使用できます。

   クラスターには、以下をインストールする必要があります (コンソール経由でクラスターを作成する場合はデフォルトで true、 AWS API または を使用してクラスターを作成する場合は追加する必要があります AWS CLI) 。Amazon EKS Pod Identity Agent、 AWS CNI、CoreDNS、Kube-proxy、Amazon EBS CSI Driver AddOns (Amazon EBS CSI Driver AddOn は、ソリューションに技術的には必要ありませんが、一部の JVM アプリケーションに必要です) 。

   クラスター名を保存して後で指定します。これは Amazon EKS コンソールのクラスター詳細にあります。

   > **Note**
   >
   > Amazon EKS クラスターの作成方法の詳細については、「Amazon EKS の使用方法」を参照してください。

2. Amazon EKS クラスターの Java 仮想マシンでアプリケーションを実行している必要があります。
3. Amazon EKS クラスターと同じ に Amazon Managed Service for Prometheus ワークスペースを作成する必要があります。 AWS アカウント 詳細については、[Amazon Managed Service for Prometheus ユーザーガイド] の [ワークスペースの作成] を参照してください。

   Amazon Managed Service for Prometheus ワークスペース ARN を保存して後で指定します。
4. Grafana バージョン 9 以降の Amazon Managed Grafana ワークスペースは、Amazon EKS クラスターと同じ に作成する必要があります。 AWS リージョン 新しいワークスペースの作成の詳細については、「Amazon Managed Grafana ワークスペースを作成する」を参照してください。

   ワークスペースロールには、Amazon Managed Service for Prometheus および Amazon CloudWatch API にアクセスするためのアクセス許可が必要です。この許可を付与する最も

簡単な方法は、[サービス管理のアクセス許可]を使用して、Amazon Managed Service for Prometheus と CloudWatch を選択することです。AmazonPrometheusQueryAccess ポリシーと AmazonGrafanaCloudWatchAccess ポリシーをワークスペース IAM ロールに手動で追加することもできます。

Amazon Managed Grafana ワークスペース ID とエンドポイントを保存して後で指定します。ID は `g-123example` の形式になります。ID とエンドポイントは Amazon Managed Grafana コンソールにあります。エンドポイントはワークスペースの URL で、ID が含まれます。例えば、`https://g-123example.grafana-workspace.<region>.amazonaws.com/` と指定します。

> **Note**
>
> ソリューションのセットアップが必ず必要なわけではありませんが、作成されたダッシュボードにアクセスするには、Amazon Managed Grafana ワークスペースでユーザー認証を設定する必要があります。詳細については、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。

このソリューションの使用

このソリューションは、Amazon EKS クラスターで実行されている Java 仮想マシン (JVM) アプリケーションからのメトリクスのレポートとモニタリングをサポートする AWS ようにインフラストラクチャを設定します。を使用してインストールできますAWS Cloud Development Kit (AWS CDK)。

> **Note**
>
> ここでのステップでは、 AWS CLI、、および Node.js AWS CDKと NPM の両方がインストールされている環境があることを前提としています。`make` と を使用して`brew`、ビルドやその他の一般的なアクションを簡素化します。

このソリューションを使用して で Amazon EKS クラスターをモニタリングするには AWS CDK

1. 前提条件のすべてのステップを完了済みであることを確認してください。
2. Amazon S3 からソリューションのファイルすべてをダウンロードします。ファイルは `s3://aws-observability-solutions/JVM_EKS/OSS/CDK/v1.0.0/iac` にあり、次の Amazon

S3 コマンドを使用してダウンロードできます。コマンドライン環境のフォルダから、このコマンドを実行します。

```
aws s3 sync s3://aws-observability-solutions/JVM_EKS/OSS/CDK/v1.0.0/iac/ .
```

これらのファイルを修正する必要はありません。

3. コマンドライン環境で (ソリューションファイルをダウンロードしたフォルダから)、次のコマンドを実行します。

   必要な環境変数を設定します。*REGION*、*AMG_ENDPOINT*、*EKS_CLUSTER*、*AMP_ARN* を、Amazon Managed Grafana ワークスペースエンドポイント (の形式`http://g-123example.grafana-workspace.us-east-1.amazonaws.com`) AWS リージョン、Amazon EKS クラスター名、Amazon Managed Service for Prometheus ワークスペース ARN に置き換えます。

   ```
   export AWS_REGION=REGION
   export AMG_ENDPOINT=AMG_ENDPOINT
   export EKS_CLUSTER_NAME=EKS_CLUSTER
   export AMP_WS_ARN=AMP_ARN
   ```

4. ソリューションで使用できる注釈を作成します。名前空間、デプロイ、ステートフルセット、デーモンセット、またはポッドに直接注釈を付けることができます。JSM ソリューションには 2 つの注釈が必要です。を使用して`kubectl`、次のコマンドでリソースに注釈を付けます。

   ```
   kubectl annotate <resource-type> <resource-value> instrumentation.opentelemetry.io/inject-java=true
   kubectl annotate <resource-type> <resource-value> cloudwatch.aws.amazon.com/inject-jmx-jvm=true
   ```

   <resource-type> と *<resource-value>* をシステムに適した値に置き換えます。たとえば、`foo`デプロイに注釈を付けるには、最初のコマンドは次のようになります。

   ```
   kubectl annotate deployment foo instrumentation.opentelemetry.io/inject-java=true
   ```

5. Grafana HTTP APIs を呼び出すための ADMIN アクセスを持つサービスアカウントトークンを作成します。詳細については、「サービスアカウントを使用した Grafana HTTP API の認証」を参照してください。トークンを作成するには、次のコマンド AWS CLI で を使用できます。*GRAFANA_ID* を ご使用の Grafana ワークスペースの ID に置き換える必要があります

(g-123example 形式)。このキーは 7,200 秒または 2 時間後に期限が切れます。必要に応じて、時間 (seconds-to-live) を変更できます。デプロイにかかる時間は 1 時間未満です。

```
# creates a new service account (optional: you can use an existing account)
GRAFANA_SA_ID=$(aws grafana create-workspace-service-account \
  --workspace-id GRAFANA_ID \
  --grafana-role ADMIN \
  --name grafana-operator-key \
  --query 'id' \
  --output text)

# creates a new token for calling APIs
export AMG_API_KEY=$(aws grafana create-workspace-service-account-token \
  --workspace-id $managed_grafana_workspace_id \
  --name "grafana-operator-key-$(date +%s)" \
  --seconds-to-live 7200 \
  --service-account-id $GRAFANA_SA_ID \
  --query 'serviceAccountToken.key' \
  --output text)
```

次のコマンドを使用して API キーを に追加 AWS CDK することで、 AWS Systems Manager で API キーを使用できるようにします。*AWS_REGION* を、ソリューションを実行するリージョンに置き換えます (us-east-1 形式)。

```
aws ssm put-parameter --name "/observability-aws-solution-jvm-eks/grafana-api-key"
 \
  --type "SecureString" \
  --value $AMG_API_KEY \
  --region AWS_REGION \
  --overwrite
```

6. 次の make コマンドを実行して、プロジェクトの他の依存関係をインストールします。

```
make deps
```

7. 最後に、 AWS CDK プロジェクトを実行します。

```
make build && make pattern aws-observability-solution-jvm-eks-$EKS_CLUSTER_NAME
 deploy
```

8. 〔オプション] スタックの作成が完了したら、それぞれの他の前提条件 (個別の Amazon Managed Grafana ワークスペースと Amazon Managed Service for Prometheus ワークスペースを含む) を満たしている限り、同じ環境を使用して、同じリージョンの Amazon EKS クラスターで実行されている他の JVM アプリケーションのスタックのインスタンスをさらに作成できます。新しいパラメータで `export` コマンドを再定義する必要があります。

スタックの作成が完了すると、Amazon Managed Grafana ワークスペースに、アプリケーションと Amazon EKS クラスターのメトリクスを示すダッシュボードが表示されます。メトリクスが収集されるため、メトリクスが表示されるまでに数分かかります。

追跡されるメトリクスのリスト

このソリューションは、JVM ベースのアプリケーションからメトリクスを収集します。これらのメトリクスは Amazon Managed Service for Prometheus に格納され、Amazon Managed Grafana ダッシュボードに表示されます。

このソリューションでは、次のメトリクスが追跡されます。

- jvm.classes.loaded
- jvm.gc.collections.count
- jvm.gc.collections.elapsed
- jvm.memory.heap.init
- jvm.memory.heap.max
- jvm.memory.heap.used
- jvm.memory.heap.committed
- jvm.memory.nonheap.init
- jvm.memory.nonheap.max
- jvm.memory.nonheap.used
- jvm.memory.nonheap.committed
- jvm.memory.pool.init
- jvm.memory.pool.max
- jvm.memory.pool.used
- jvm.memory.pool.committed
- jvm.threads.count

トラブルシューティング

プロジェクトのセットアップが失敗する原因がいくつかあります。必ず以下の内容を確認してください。

- ソリューションのインストール前にすべての [前提条件] を完了する必要があります。
- ソリューションの作成やメトリクスへのアクセスを試みる前に、クラスターに少なくとも 1 つのノードを含める必要があります。
- Amazon EKS クラスターには、AWS CNI、CoreDNS、kube-proxy アドオンをインストールする必要があります。インストールしないとソリューションが正しく動作しません。これらは、コンソールを介してクラスターを作成するときにデフォルトでインストールされます。クラスターが AWS SDK を使用して作成された場合は、インストールが必要になる場合があります。
- Amazon EKS ポッドのインストールがタイムアウトしました。これは、使用可能なノード容量が十分でないときに発生する場合があります。これらの問題には、次のような複数の原因があります。
  - Amazon EKS クラスターが Amazon EC2 ではなく Fargate で初期化されました。このプロジェクトでは Amazon EC2 が必須です。
  - ノードが汚染されているため使用できません。

    `kubectl describe node NODENAME | grep Taints` を使用するとテイントを確認できます。次に `kubectl taint node NODENAME TAINT_NAME-` でテイントを削除します。テイント名の後に必ず - を含めてください。
  - ノードが容量制限に達しました。この場合は新しいノードを作成、または容量を増やすことができます。
- Grafana にダッシュボードが表示されません: 間違った Grafana ワークスペース ID を使用しています。

  以下のコマンドを実行して Grafana に関する情報を取得してください。

  ```
  kubectl describe grafanas external-grafana -n grafana-operator
  ```

  正しいワークスペース URL の結果を確認できます。予想しているものでない場合は、正しいワークスペース ID で再デプロイします。

  ```
  Spec:
    External:
      API Key:
  ```

```
      Key:   GF_SECURITY_ADMIN_APIKEY
      Name:  grafana-admin-credentials
    URL:     https://g-123example.grafana-workspace.aws-region.amazonaws.com
Status:
  Admin URL:  https://g-123example.grafana-workspace.aws-region.amazonaws.com
  Dashboards:
    ...
```

- Grafana にダッシュボードが表示されません: 期限切れの API キーを使用しています。

  この場合の検索では、grafana 演算子を取得して、ログにエラーがないことを確認する必要があります。次のコマンドを使用して Grafana 演算子の名前を取得します。

  ```
  kubectl get pods -n grafana-operator
  ```

  次のような演算子名が返されます。

  ```
  NAME                                READY   STATUS    RESTARTS   AGE
  grafana-operator-1234abcd5678ef90   1/1     Running   0          1h2m
  ```

  次のコマンドで演算子名を使用します。

  ```
  kubectl logs grafana-operator-1234abcd5678ef90 -n grafana-operator
  ```

  次のようなエラーメッセージは、API キーの有効期限が切れたことを示しています。

  ```
  ERROR   error reconciling datasource    {"controller": "grafanadatasource",
   "controllerGroup": "grafana.integreatly.org", "controllerKind": "GrafanaDatasource",
   "GrafanaDatasource": {"name":"grafanadatasource-sample-amp","namespace":"grafana-
  operator"}, "namespace": "grafana-operator", "name": "grafanadatasource-sample-
  amp", "reconcileID": "72cfd60c-a255-44a1-bfbd-88b0cbc4f90c", "datasource":
   "grafanadatasource-sample-amp", "grafana": "external-grafana", "error": "status:
   401, body: {\"message\":\"Expired API key\"}\n"}
  github.com/grafana-operator/grafana-operator/controllers.
  (*GrafanaDatasourceReconciler).Reconcile
  ```

  この場合は、新しい API キーを作成してソリューションを再度デプロイします。問題が解決しない場合は、再びデプロイする前に次のコマンドを使用して強制的に同期できます。

  ```
  kubectl delete externalsecret/external-secrets-sm -n grafana-operator
  ```

- SSM パラメータがありません。次のようなエラーが表示される場合は、`cdk bootstrap` を実行して再試行してください。

```
Deployment failed: Error: aws-observability-solution-jvm-eks-$EKS_CLUSTER_NAME: SSM
parameter /cdk-bootstrap/xxxxxxx/version not found. Has the environment been
bootstrapped? Please run 'cdk bootstrap' (see https://docs.aws.amazon.com/cdk/latest/
guide/bootstrapping.html)
```

Amazon Managed Grafana で Kafka アプリケーションをモニタリングするためのソリューション

Apache Kafka 上に構築されたアプリケーションには、特殊なモニタリングニーズがあります。このページでは、Amazon EKS クラスターの Java 仮想マシンで実行されている Kafka アプリケーションをモニタリングするためのソリューションを提供するテンプレートについて説明します。このソリューションは、 を使用してインストールできますAWS Cloud Development Kit (AWS CDK)。

> **Note**
>
> このソリューションは、Amazon Managed Streaming for Apache Kafka アプリケーションのモニタリングをサポートしていません。Amazon MSK アプリケーションのモニタリングの詳細については、「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「Amazon MSK クラスターのモニタリング」を参照してください。

このソリューションは以下を設定します。

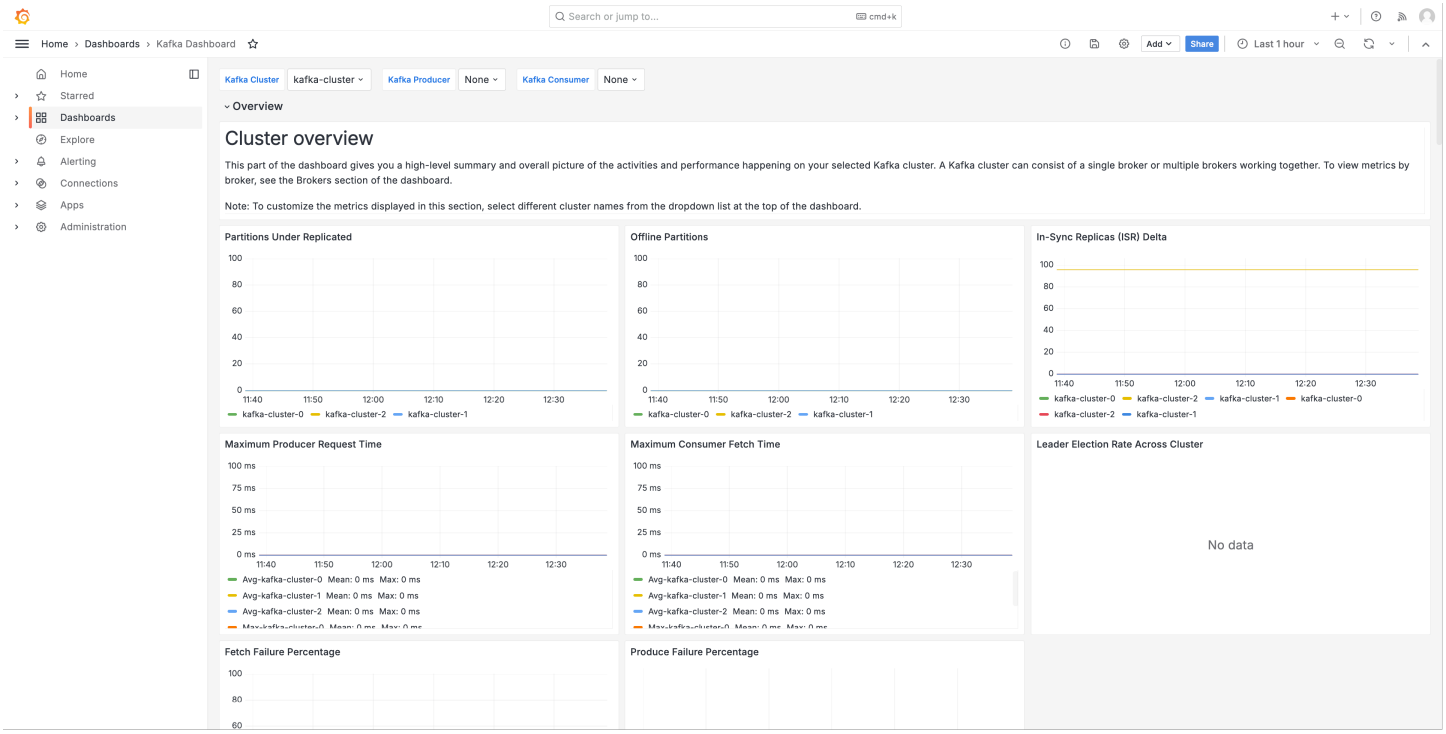
- Amazon EKS クラスターから Kafka および Java 仮想マシン (JVM) メトリクスを保存する Amazon Managed Service for Prometheus ワークスペース。
- CloudWatch エージェントと CloudWatch エージェントアドオンを使用して、特定の Kafka および JVM CloudWatch メトリクスを収集します。メトリクスは、Amazon Managed Service for Prometheus ワークスペースに送信するように設定されています。
- Amazon Managed Grafana ワークスペースでこれらのメトリクスをプルし、クラスターのモニタリングに役立つダッシュボードを作成します。

> **Note**
>
> このソリューションは、Amazon EKS で実行されているアプリケーションの JVM および Kafka メトリクスを提供しますが、Amazon EKS メトリクスは含まれません。オブザーバビリティソリューションを使用して Amazon EKS をモニタリングし、Amazon EKS クラスターのメトリクスとアラートを確認できます。

このソリューションについて

このソリューションは、Apache Kafka アプリケーションのメトリクスを提供するように Amazon Managed Grafana ワークスペースを設定します。メトリクスは、Kafka アプリケーションのパフォーマンスとワークロードに関するインサイトを提供することで、アプリケーションの運用をより効果的に行うのに役立つダッシュボードを生成するために使用されます。

次の図は、このソリューションによって作成されたダッシュボードのサンプルを示しています。



メトリクスは 1 分間のスクレイプ間隔でスクレイプされます。ダッシュボードには、特定のメトリクスに基づいて、1 分、5 分、またはそれ以上に集約されたメトリクスが表示されます。

このソリューションで追跡されるメトリクスのリストについては、「追跡されるメトリクスのリスト」を参照してください。

コスト

このソリューションは、ワークスペース内にリソースを作成して使用します。作成したリソースの標準使用量に対して課金されますが、これには以下が含まれます。

- ユーザーによる Amazon Managed Grafana ワークスペースアクセス。料金に関する詳細については、「Amazon Managed Grafana の料金」を参照してください。
- Amazon Managed Service for Prometheus メトリクスの取り込みとストレージ、およびメトリクス分析 (クエリサンプル処理) 。このソリューションで使用されるメトリクスの数は、アプリケーションの設定と使用状況によって異なります。

  CloudWatch を使用して、Amazon Managed Service for Prometheus で取り込みメトリクスとストレージメトリクスを表示できます。詳細については、Amazon Managed Service for Prometheus ユーザーガイドの [CloudWatch メトリクス] を参照してください。

  Amazon Managed Service for Prometheus の料金ページでは、料金計算ツールを使用してコストを見積もることができます。メトリクス数は、クラスター内のノード数と、アプリケーションが生成するメトリクスによって異なります。
- ネットワークコスト。クロスアベイラビリティーゾーン、リージョン、またはその他のトラフィックに対して標準 AWS ネットワーク料金が発生する場合があります。

各製品の料金ページから利用できる料金計算ツールは、ソリューションの潜在的コストを理解するのに役立ちます。以下の情報は、Amazon EKS クラスターと同じアベイラビリティーゾーンで実行されているソリューションの、基本コストを取得するのに役立ちます。

製品	計算メトリクス	値
Amazon Managed Service for Prometheus	アクティブなシリーズ	95 (Kafka ポッドあたり)
	平均収集間隔	60 (秒)
Amazon Managed Grafana	アクティブなエディタ/管理者の数	1 (またはユーザーに基づき、さらに多く)

これらの数値は、Amazon EKS で Kafka を実行するソリューションのベース番号です。これにより基本コストの見積もりが得られます。Kafka ポッドをアプリケーションに追加すると、次に示すよ

うにコストが増加します。これらのコストにより、ネットワーク使用コストが削減されます。これは、Amazon Managed Grafana ワークスペース、Amazon Managed Service for Prometheus ワークスペース、Amazon EKS クラスターが同じアベイラビリティーゾーン AWS リージョン、および VPN にあるかどうかによって異なります。

前提条件

このソリューションを使用する前に、次のことを行う必要があります。

1. モニタリングする [Amazon Elastic Kubernetes Service クラスターを作成する]、またはそのクラスターがあるいる必要があります。そのクラスターには少なくとも 1 つのノードが必要です。クラスターには、プライベートアクセスを含めるように設定された API サーバーエンドポイントアクセスが必要です (パブリックアクセスを許可することもできます)。

   認証モードに API アクセスを含める必要があります (`API` または `API_AND_CONFIG_MAP` に設定)。これによりソリューションデプロイでアクセスエントリを使用できます。

   クラスターには、以下をインストールする必要があります (コンソール経由でクラスターを作成する場合はデフォルトでは true、 AWS API または を使用してクラスターを作成する場合は追加する必要があります AWS CLI) 。Amazon EKS Pod Identity Agent、 AWS CNI、CoreDNS、Kube-proxy、Amazon EBS CSI Driver AddOns (Amazon EBS CSI Driver AddOn は、ソリューションに技術的には必要ありませんが、ほとんどの Kafka アプリケーションに必要です) 。

   クラスター名を保存して後で指定します。これは Amazon EKS コンソールのクラスター詳細にあります。

   > **Note**
   >
   > Amazon EKS クラスターの作成方法の詳細については、「Amazon EKS の使用方法」を参照してください。

2. Amazon EKS クラスターの Java 仮想マシンで Apache Kafka アプリケーションを実行している必要があります。
3. Amazon EKS クラスターと同じ に AWS アカウント Amazon Managed Service for Prometheus ワークスペースを作成する必要があります。詳細については、[Amazon Managed Service for Prometheus ユーザーガイド] の [ワークスペースの作成] を参照してください。

   Amazon Managed Service for Prometheus ワークスペース ARN を保存して後で指定します。

4. Grafana バージョン 9 以降の Amazon Managed Grafana ワークスペースは、Amazon EKS クラスターと同じ に作成する必要があります。 AWS リージョン 新しいワークスペースの作成の詳細については、「Amazon Managed Grafana ワークスペースを作成する」を参照してください。

   ワークスペースロールには、Amazon Managed Service for Prometheus および Amazon CloudWatch API にアクセスするためのアクセス許可が必要です。この許可を付与する最も簡単な方法は、[サービス管理のアクセス許可]を使用して、Amazon Managed Service for Prometheus と CloudWatch を選択することです。AmazonPrometheusQueryAccess ポリシーと AmazonGrafanaCloudWatchAccess ポリシーをワークスペース IAM ロールに手動で追加することもできます。

   Amazon Managed Grafana ワークスペース ID とエンドポイントを保存して後で指定します。ID は `g-123example` の形式になります。ID とエンドポイントは Amazon Managed Grafana コンソールにあります。エンドポイントはワークスペースの URL で、ID が含まれます。例えば、`https://g-123example.grafana-workspace.<region>.amazonaws.com/`。

> **Note**
>
> ソリューションのセットアップが必ず必要なわけではありませんが、作成されたダッシュボードにアクセスするには、Amazon Managed Grafana ワークスペースでユーザー認証を設定する必要があります。詳細については、「Amazon Managed Grafana ワークスペースでユーザーを認証する」を参照してください。

このソリューションの使用

このソリューションは、Amazon EKS クラスターで実行されている Kafka アプリケーションからのメトリクスのレポートとモニタリングをサポートする AWS ようにインフラストラクチャを設定します。を使用してインストールできますAWS Cloud Development Kit (AWS CDK)。

> **Note**
>
> ここでのステップでは、 AWS CLI、、および Node.js AWS CDKと NPM の両方がインストールされている環境があることを前提としています。`make` および を使用して`brew`、ビルドやその他の一般的なアクションを簡素化します。

このソリューションを使用して で Amazon EKS クラスターをモニタリングするには AWS CDK

1. 前提条件のすべてのステップを完了済みであることを確認してください。
2. Amazon S3 からソリューションのファイルすべてをダウンロードします。ファイルは s3://aws-observability-solutions/Kafka_EKS/OSS/CDK/v1.0.0/iac にあり、次の Amazon S3 コマンドを使用してダウンロードできます。コマンドライン環境のフォルダから、このコマンドを実行します。

   ```
   aws s3 sync s3://aws-observability-solutions/Kafka_EKS/OSS/CDK/v1.0.0/iac/ .
   ```

   これらのファイルを修正する必要はありません。
3. コマンドライン環境で (ソリューションファイルをダウンロードしたフォルダから)、次のコマンドを実行します。

   必要な環境変数を設定します。*REGION*、*AMG_ENDPOINT*、*EKS_CLUSTER*、*AMP_ARN* を、 AWS リージョンの Amazon Managed Grafana ワークスペースエンドポイント (の形式http://g-123example.grafana-workspace.us-east-1.amazonaws.com) 、Amazon EKS クラスター名、Amazon Managed Service for Prometheus ワークスペース ARN に置き換えます。

   ```
   export AWS_REGION=REGION
   export AMG_ENDPOINT=AMG_ENDPOINT
   export EKS_CLUSTER_NAME=EKS_CLUSTER
   export AMP_WS_ARN=AMP_ARN
   ```
4. デプロイで使用できる注釈を作成する必要があります。名前空間、デプロイ、ステートフルセット、デーモンセット、またはポッドに直接注釈を付けることができます。Kafka ソリューションには 5 つの注釈が必要です。を使用してkubectl、次のコマンドでリソースに注釈を付けます。

   ```
   kubectl annotate <resource-type> <resource-value> instrumentation.opentelemetry.io/inject-java=true
   kubectl annotate <resource-type> <resource-value> cloudwatch.aws.amazon.com/inject-jmx-jvm=true
   kubectl annotate <resource-type> <resource-value> cloudwatch.aws.amazon.com/inject-jmx-kafka=true
   kubectl annotate <resource-type> <resource-value> cloudwatch.aws.amazon.com/inject-jmx-kafka-producer=true
   ```

```
kubectl annotate <resource-type> <resource-value> cloudwatch.aws.amazon.com/inject-jmx-kafka-consumer=true
```

<resource-type> と *<resource-value>* をシステムに適した値に置き換えます。たとえば、fooデプロイに注釈を付けるには、最初のコマンドは次のとおりです。

```
kubectl annotate deployment foo instrumentation.opentelemetry.io/inject-java=true
```

5. Grafana HTTP APIs を呼び出すための ADMIN アクセスを持つサービスアカウントトークンを作成します。詳細については、「サービスアカウントを使用した Grafana HTTP API の認証」を参照してください。トークンを作成するには、次のコマンド AWS CLI で を使用できます。*GRAFANA_ID* を ご使用の Grafana ワークスペースの ID に置き換える必要があります (g-123example 形式)。このキーは 7,200 秒または 2 時間後に期限が切れます。必要に応じて、時間 (seconds-to-live) を変更できます。デプロイにかかる時間は 1 時間未満です。

```
# creates a new service account (optional: you can use an existing account)
GRAFANA_SA_ID=$(aws grafana create-workspace-service-account \
  --workspace-id GRAFANA_ID \
  --grafana-role ADMIN \
  --name grafana-operator-key \
  --query 'id' \
  --output text)

# creates a new token for calling APIs
export AMG_API_KEY=$(aws grafana create-workspace-service-account-token \
  --workspace-id $managed_grafana_workspace_id \
  --name "grafana-operator-key-$(date +%s)" \
  --seconds-to-live 7200 \
  --service-account-id $GRAFANA_SA_ID \
  --query 'serviceAccountToken.key' \
  --output text)
```

次のコマンドを使用して、 API キーを に追加 AWS CDK することで AWS Systems Manager 、で使用できるようにします。*AWS_REGION* を、ソリューションを実行するリージョンに置き換えます (us-east-1 形式)。

```
aws ssm put-parameter --name "/observability-aws-solution-kafka-eks/grafana-api-key" \
  --type "SecureString" \
  --value $AMG_API_KEY \
```

```
  --region AWS_REGION \
  --overwrite
```

6. 次の make コマンドを実行して、プロジェクトの他の依存関係をインストールします。

```
make deps
```

7. 最後に、 AWS CDK プロジェクトを実行します。

```
make build && make pattern aws-observability-solution-kafka-eks-$EKS_CLUSTER_NAME
 deploy
```

8. 〔オプション] スタックの作成が完了したら、それぞれの他の前提条件 (個別の Amazon Managed Grafana ワークスペースと Amazon Managed Service for Prometheus ワークスペースを含む) を満たしている限り、同じ環境を使用して、同じリージョンの Amazon EKS クラスターで実行されている他の Kafka アプリケーションのスタックのインスタンスをさらに作成できます。新しいパラメータで export コマンドを再定義する必要があります。

スタックの作成が完了すると、Amazon Managed Grafana ワークスペースに、アプリケーションと Amazon EKS クラスターのメトリクスを示すダッシュボードが入力されます。メトリクスが収集されるため、メトリクスが表示されるまでに数分かかります。

追跡されるメトリクスのリスト

このソリューションは、JVM ベースの Kafka アプリケーションからメトリクスを収集します。これらのメトリクスは Amazon Managed Service for Prometheus に格納され、Amazon Managed Grafana ダッシュボードに表示されます。

このソリューションでは、次のメトリクスを追跡します。

- jvm.classes.loaded
- jvm.gc.collections.count
- jvm.gc.collections.elapsed
- jvm.memory.heap.init
- jvm.memory.heap.max
- jvm.memory.heap.used
- jvm.memory.heap.committed
- jvm.memory.nonheap.init

- jvm.memory.nonheap.max
- jvm.memory.nonheap.used
- jvm.memory.nonheap.committed
- jvm.memory.pool.init
- jvm.memory.pool.max
- jvm.memory.pool.used
- jvm.memory.pool.committed
- jvm.threads.count
- kafka.message.count
- kafka.request.count
- kafka.request.failed
- kafka.request.time.total
- kafka.request.time.50p
- kafka.request.time.99p
- kafka.request.time.avg
- kafka.network.io
- kafka.purgatory.size
- kafka.partition.count
- kafka.partition.offline
- kafka.partition.under_replicated
- kafka.isr.operation.count
- kafka.max.lag
- kafka.controller.active.count
- kafka.leader.election.rate
- kafka.unclean.election.rate
- kafka.request.queue
- kafka.logs.flush.time.count
- kafka.logs.flush.time.median
- kafka.logs.flush.time.99p
- kafka.consumer.fetch-rate

- kafka.consumer.records-lag-max
- kafka.consumer.total.bytes-consumed-rate
- kafka.consumer.total.fetch-size-avg
- kafka.consumer.total.records-consumed-rate
- kafka.consumer.bytes-consumed-rate
- kafka.consumer.fetch-size-avg
- kafka.consumer.records-consumed-rate
- kafka.producer.io-wait-time-ns-avg
- kafka.producer.outgoing-byte-rate
- kafka.producer.request-latency-avg
- kafka.producer.request-rate
- kafka.producer.response-rate
- kafka.producer.byte-rate
- kafka.producer.compression-rate
- kafka.producer.record-error-rate
- kafka.producer.record-retry-rate
- kafka.producer.record-send-rate

トラブルシューティング

プロジェクトのセットアップが失敗する原因がいくつかあります。必ず以下の内容を確認してください。

- ソリューションのインストール前にすべての [前提条件] を完了する必要があります。
- ソリューションの作成やメトリクスへのアクセスを試みる前に、クラスターに少なくとも 1 つのノードを含める必要があります。
- Amazon EKS クラスターには、`AWS CNI`、`CoreDNS`、`kube-proxy` アドオンをインストールする必要があります。インストールしないとソリューションが正しく動作しません。これらは、コンソールを介してクラスターを作成するときにデフォルトでインストールされます。クラスターが AWS SDK を介して作成された場合は、インストールが必要になる場合があります。
- Amazon EKS ポッドのインストールがタイムアウトしました。これは、使用可能なノード容量が十分でないときに発生する場合があります。これらの問題には、次のような複数の原因があります。

- Amazon EKS クラスターが Amazon EC2 ではなく Fargate で初期化されました。このプロジェクトでは Amazon EC2 が必須です。
- ノードが汚染されているため使用できません。

  `kubectl describe node NODENAME | grep Taints` を使用するとテイントを確認できます。次に `kubectl taint node NODENAME TAINT_NAME-` でテイントを削除します。テイント名の後に必ず - を含めてください。
- ノードが容量制限に達しました。この場合は新しいノードを作成、または容量を増やすことができます。

- Grafana にダッシュボードが表示されません: 間違った Grafana ワークスペース ID を使用しています。

  以下のコマンドを実行して Grafana に関する情報を取得してください。

```
kubectl describe grafanas external-grafana -n grafana-operator
```

  正しいワークスペース URL の結果を確認できます。予想しているものでない場合は、正しいワークスペース ID で再デプロイします。

```
Spec:
  External:
    API Key:
      Key:   GF_SECURITY_ADMIN_APIKEY
      Name:  grafana-admin-credentials
    URL:     https://g-123example.grafana-workspace.aws-region.amazonaws.com
Status:
  Admin URL:  https://g-123example.grafana-workspace.aws-region.amazonaws.com
  Dashboards:
    ...
```

- Grafana にダッシュボードが表示されません: 期限切れの API キーを使用しています。

  この場合の検索では、grafana 演算子を取得して、ログにエラーがないことを確認する必要があります。次のコマンドを使用して Grafana 演算子の名前を取得します。

```
kubectl get pods -n grafana-operator
```

  次のような演算子名が返されます。

```
NAME                                  READY   STATUS    RESTARTS   AGE
grafana-operator-1234abcd5678ef90   1/1     Running   0          1h2m
```

次のコマンドで演算子名を使用します。

```
kubectl logs grafana-operator-1234abcd5678ef90 -n grafana-operator
```

次のような エラーメッセージは、API キーの有効期限が切れたことを示しています。

```
ERROR   error reconciling datasource    {"controller": "grafanadatasource",
 "controllerGroup": "grafana.integreatly.org", "controllerKind": "GrafanaDatasource",
 "GrafanaDatasource": {"name":"grafanadatasource-sample-amp","namespace":"grafana-
operator"}, "namespace": "grafana-operator", "name": "grafanadatasource-sample-
amp", "reconcileID": "72cfd60c-a255-44a1-bfbd-88b0cbc4f90c", "datasource":
 "grafanadatasource-sample-amp", "grafana": "external-grafana", "error": "status:
 401, body: {\"message\":\"Expired API key\"}\n"}
github.com/grafana-operator/grafana-operator/controllers.
(*GrafanaDatasourceReconciler).Reconcile
```

この場合は、新しい API キーを作成してソリューションを再度デプロイします。問題が解決しない場合は、再びデプロイする前に次のコマンドを使用して強制的に同期できます。

```
kubectl delete externalsecret/external-secrets-sm -n grafana-operator
```

- SSM パラメータがありません。次のような エラーが表示される場合は、cdk bootstrap を実行して再試行してください。

```
Deployment failed: Error: aws-observability-solution-kafka-eks-$EKS_CLUSTER_NAME:
 SSM
parameter /cdk-bootstrap/xxxxxxx/version not found. Has the environment been
bootstrapped? Please run 'cdk bootstrap' (see https://docs.aws.amazon.com/cdk/latest/
guide/bootstrapping.html)
```

Amazon Managed Grafana リソースへのタグの追加

タグは、ユーザーまたは AWS が AWS リソースに割り当てるカスタム属性ラベルです。各 AWS タグには 2 つの部分があります。

- タグキー (CostCenter、Environment、Project、Secret など)。タグキーでは、大文字と小文字が区別されます。
- タグ値と呼ばれるオプションのフィールド (111122223333、Production、チーム名など)。タグ値を省略すると、空の文字列を使用した場合と同じになります。タグキーと同様に、タグ値では大文字と小文字が区別されます。

これらを合わせて、キーと値のペアと呼ばれます。各ワークスペースには、最大 50 個のタグを割り当てることができます。

タグは、 AWS リソースの識別と整理に役立ちます。多くの AWS サービスはタグ付けをサポートしているため、異なる サービスのリソースに同じタグを割り当てて、リソースが関連していることを示すことができます。例えば、関連する Amazon Managed Grafana ワークスペースに、同じタグを割り当てることができます。タグ付け戦略の詳細については、「 AWS 全般のリファレンスガイド」の「 リソースのタグ付け AWS」を参照してください。

Amazon Managed Grafana はワークスペースのタグ付けをサポートしています。コンソール、CLI、APIs、または SDKs AWS を使用して、これらのリソースのタグを追加、管理、削除できます。タグは、ワークスペースの識別、整理、追跡に使用できるほか、IAM ポリシーで使用すると、Amazon Managed Grafana リソースを表示および操作できるユーザーを制御するために役立ちます。タグベースのアクセスコントロールの詳細については、「IAM ユーザーガイド」の「タグを使用した AWS リソースへのアクセスの制御」を参照してください。

タグの制限

タグには以下のベーシックな制限があります。

- 各リソースには、最大 50 個のタグを設定できます。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- タグキーの最大長は UTF-8 で 128 Unicode 文字です。
- タグ値の最大長は UTF-8 で 256 Unicode 文字です。

- タグ付けスキーマが複数の AWS サービスおよびリソースで使用されている場合は、他のサービスで許可される文字に制限がある可能性があることに注意してください。一般的に使用できる文字は、UTF-8 で表現できる英字、数字、スペースと、. : + = @ _ / - (ハイフン) です。
- タグのキーと値では、大文字と小文字が区別されます。ベストプラクティスとして、タグでの大文字の使用方針を決定し、その方針をすべてのリソースタイプで一貫して実装することをお勧めします。例えば、Costcenter、costcenter、CostCenter のいずれを使用するかを決定し、すべてのタグに同じ規則を使用します。大文字と小文字の扱いについて、同様のタグに整合性のない規則を使用することは避けてください。
- キーまたは値のプレフィックスとして、aws:、AWS:、またはキーワード AWS の大文字と小文字の組み合わせを変えたものは使用しないでください。これらは AWS 専用に予約されています。このプレフィックスを持つタグのキーや値を編集または削除することはできません。このプレフィックスを持つタグは、リソースごとのタグ数の制限にはカウントされません。

タグ付けの制限の詳細については、「 AWS 全般のリファレンスガイド」の「 リソースのタグ付け AWS」を参照してください。

Amazon Managed Grafana ワークスペースのタグを付け

Amazon Managed Grafana ワークスペースのタグを使用するには、このセクションの手順を使用します。

ワークスペースへのタグの追加

Amazon Managed Grafana ワークスペースにタグ (キーと値のペア) を追加すると、 AWS リソースの識別と整理に役立ちます。最初に、ワークスペースに 1 つ以上のタグを追加します。このタグを基にワークスペースへのアクセスを管理する IAM ポリシーを作成することができます。コンソールまたは CLI AWS を使用して、Amazon Managed Grafana ワークスペースにタグを追加できます。

> **Important**
>
> ワークスペースにタグを追加すると、そのワークスペースへのアクセスに影響が生じる可能性があります。ワークスペースにタグを追加する前に、タグを使用してリソースへのアクセスを制御している可能性のある IAM ポリシーをすべて確認してください。

Amazon Managed Grafana ワークスペースにタグを追加する方法については、「Amazon Managed Grafana User Guide」の「Amazon Managed Grafana ワークスペースを作成する」を参照してください。

Console

コンソールから Amazon Managed Grafana ワークスペースにタグ (複数可) を追加することができます。

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側にあるナビゲーションペインで、[メニュー] アイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースのワークスペース ID を選択します。
5. [タグ] タブを選択します。
6. [タグを管理] を選択します。
7. [キー] フィールドにタグ名を入力します。[値] では、必要に応じて値にタグを追加することができます。
8. (オプション) 別のタグを追加するには、[タグを追加] を選択します。
9. タグの追加が完了したら、[変更を保存] を選択します。

CLI

CLI AWS を使用して Amazon Managed Grafana ワークスペースに 1 つ以上のタグを追加するには、次の手順に従います。

- ターミナルまたはコマンドラインで、`tag-resource` コマンドを実行して、タグを追加するワークスペースの Amazon リソースネーム (ARN) と、追加するタグのキーおよび値を指定します。ワークスペースには 1 つ以上のタグを追加できます。以下に、Grafana ワークスペース *My-Workspace* に *[##]*という名前のタグキーを *my_key_value* のタグ値でタグ付けする場合に実行するコマンドの例を示します:

```
aws grafana tag-resource --resource-arn arn:aws:grafana:us-west-2:123456789012:/
workspace/My-Workspace --tags "Name=my_key_value"
```

ワークスペースのタグの表示

Amazon Managed Grafana ワークスペースに関連付けられたタグを表示できます。タグ付け戦略の詳細については、「 AWS 全般のリファレンス」の「 AWS リソースのタグ付け」を参照してください。

Console

コンソールを使用して、Amazon Managed Grafana ワークスペースに関連付けられているタグを表示できます。

1. Amazon Managed Grafana コンソール (https://console.aws.amazon.com/grafana/) を開きます。
2. 左側にあるナビゲーションペインで、[メニュー] アイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースのワークスペース ID を選択します。
5. [タグ] タブを選択します。

CLI

CLI AWS を使用して Amazon Managed Grafana ワークスペースの AWS タグを表示するには、次の手順に従います。タグが追加されていない場合、返されるリストは空になります。

ターミナルまたはコマンドラインで、list-tags-for-resource コマンドを実行します。例えば、ワークスペースのタグキーとタグ値の一覧を表示するには、次のコマンドを実行します。

```
aws grafana list-tags-for-resource --resoure-arn arn:aws:grafana:us-west-2:/
workspace/workspace-IDstring
```

成功した場合、このコマンドは次のような情報を返します。

```
{
    "tags": {
        "Status": "Secret",
        "Team": "My-Team"
    }
}
```

ワークスペースのタグの編集

TagResource API を使用すると、ワークスペースに関連付けられているタグの値を 1 回の呼び出しで変更できます。既存のタグのキーを更新する場合、UntagResource と TagResource API を組み合わせて使用します。

> **Important**
>
> Amazon Managed Grafana ワークスペースのタグを編集すると、そのワークスペースへのアクセス権に影響が及ぶ可能性があります。ワークスペースのタグを編集する前に、タグのキーや値を使用してリポジトリなどのリソースへのアクセスを制御する可能性のある IAM ポリシーを必ず確認してください。

Console

コンソールを使用して、Amazon Managed Grafana ワークスペースに関連付けられているタグを編集できます。

1. Grafana コンソールで Amazon Managed Grafana コンソールを開きます。
2. 左側にあるナビゲーションペインで、[メニュー] アイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースのワークスペース ID を選択します。
5. [タグ] タブを選択します。
6. [タグを管理] を選択します。
7. [キー] フィールドにタグ名を入力します。[値] では、必要に応じて値にタグを追加することができます。
8. タグの編集が完了したら、[変更を保存] を選択します。

CLI

を使用してワークスペースのタグ AWS CLI を更新するには、次の手順に従います。既存のキーの値を変更したり、別のキーを追加することができます。

ターミナルまたはコマンドラインで、tag-resource コマンドを実行して、タグを更新する Amazon Managed Grafana ワークスペースの Amazon リソースネーム (ARN) と、タグキーおよびタグ値を指定します。

例えば、タグの値を新しい値「*Key_value_new*」に変更するには、次のコマンドを実行します。

```
aws grafana tag-resource \
    --resource-arn arn:aws:grafana:us-west-2:123456789012:/workspace/workspace-IDstring \
    --tags "Name=Key_value_new"
```

タグのキーを新しい名前「*Name_new* 」に変更するには、次のコマンドを実行します。

```
aws grafana untag-resource --resource-arn arn:aws:grafana:us-west-2:123456789012:/workspace/workspace-IDstring --tag-keys Items=Name
aws grafana tag-resource --resource-arn arn:aws:grafana:us-west-2:123456789012:/workspace/workspace-IDstring --tags "Name_new=Key_value_old"
```

ワークスペースからのタグの削除

ワークスペースに関連付けられている 1 つ以上のタグを削除できます。タグを削除しても、そのタグに関連付けられている他の AWS リソースからタグは削除されません。

> **Important**
>
> Amazon Managed Grafana ワークスペースからタグを削除すると、そのワークスペースへのアクセスに影響が及ぶ可能性があります。ワークスペースからタグを削除する前に、タグのキーや値を使用してワークスペースなどのリソースへのアクセスを制御する可能性のある IAM ポリシーを必ず確認してください。

Console

コンソールを使用して、タグとワークスペースとの関連付けを解除できます。

1. Grafana コンソールで Amazon Managed Grafana コンソールを開きます。
2. 左側にあるナビゲーションペインで、[メニュー] アイコンを選択します。
3. [すべてのワークスペース] を選択します。
4. 管理するワークスペースのワークスペース ID を選択します。
5. [タグ] タブを選択します。

6. [タグを管理] を選択します。
7. 削除するタグを見つけて、[削除] を選択します。
8. タグの削除を終了したら、[変更を保存] を選択します。

CLI

を使用してワークスペースからタグ AWS CLI を削除するには、次の手順に従います。タグを削除しても他のリソースからそのタグがなくなるわけではありません。タグとワークスペースとの関連付けが解除されるだけです。

> **Note**
>
> Amazon Managed Grafana ワークスペースを削除すると、削除されたワークスペースからタグの関連付けがすべて解除されます。ワークスペースを削除する前にタグを削除する必要はありません。

ターミナルまたはコマンドラインで、[untag-resource] コマンドを実行して、タグを削除するワークスペースの Amazon リソースネーム (ARN) と、削除するタグのタグキーを指定します。例えば、*workspace-IDstring* という名前のワークスペースでタグキー *##* を使用してタグを削除するには、次のコマンドを実行します。

```
aws grafana untag-resource --resoure-arn arn:aws:grafana:us-west-2:/
workspaces/workspace-IDstring --tag-keys Items=Name
```

正常に実行されると、このコマンドは空のレスポンスを返します。ワークスペースに関連付けられたタグが削除されていることを確認するには、[list-tags-for-resource] コマンドを実行します。

Amazon Managed Grafana のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、お客様と AWS お客様の間の責任共有です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – クラウドで AWS AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。 AWS また、 では、安全に使用できるサービスも提供しています。サードパーティーの監査者は、AWS コンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。Amazon Managed Grafana に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウドのセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Amazon Managed Grafana 使用時における責任共有モデルの適用法を理解するのに役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目的を満たすように Amazon Managed Grafana を設定する方法について説明します。また、Amazon Managed Grafana リソースのモニタリングと保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- でのデータ保護 AWS
- Amazon Managed Grafana 用の Identity and Access Management
- AWS データソースの Amazon Managed Grafana アクセス許可とポリシー
- IAM 許可
- Amazon Managed Grafana のコンプライアンス検証
- Amazon Managed Grafana の耐障害性
- Amazon Managed Grafana のインフラストラクチャセキュリティ
- AWS CloudTrailを使用した Amazon Managed Grafana API コールのログ記録
- セキュリティに関するベストプラクティス

- インターフェイス VPC エンドポイント

でのデータ保護 AWS

Amazon Managed Grafana でのデータ保護には、 AWS 責任共有モデル責任共有モデルが適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。また、使用する「 AWS のサービス 」のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、データプライバシーに関するよくある質問を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された AWS 責任共有モデルおよび GDPR のブログ記事を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント 、 AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「 AWS CloudTrail ユーザーガイド」のCloudTrail 証跡の使用」を参照してください。
- AWS 暗号化ソリューションと、その中のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度な管理されたセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-3 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、タグ、または [名前] フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これは、コンソール、API、または SDK を使用して Amazon Managed Grafana AWS CLIまたは他の AWS のサービス を使用する場合も同様です。 AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力し

たデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

Amazon Managed Grafana でのデータ保護

Amazon Managed Grafana では、次のタイプのデータが収集、保存されます。

- Grafana ワークスペースに対して顧客が行ったダッシュボード設定とアラート設定。
- ワークスペースに保存されている Grafana ダッシュボードスナップショット。
- AWS IAM Identity Center ユーザーのユーザー名と E メールアドレスを含む、Grafana ワークスペースへのアクセスが許可されているユーザーのリスト。

Amazon Managed Grafana が保存するデータは暗号化されます AWS Key Management Service。転送中のデータは、Secure Sockets Layer (SSL) で自動的に暗号化されます。

Amazon Managed Grafana 用の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰に Amazon Managed Grafana リソースの使用を承認 (許可を付与) するかを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

トピック

- 対象者
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- Amazon Managed Grafana と IAM の連携方法
- Amazon Managed Grafana のアイデンティティベースのポリシー例
- AWS Amazon Managed Grafana の マネージドポリシー
- Amazon Managed Grafana アイデンティティとアクセスのトラブルシューティング
- サービス間の混乱した代理の防止

- Amazon Managed Grafana のサービスリンクロールの使用

対象者

AWS Identity and Access Management (IAM) の使用方法は、Amazon Managed Grafana で行う作業によって異なります。

サービスユーザー – 業務を行うために Amazon Managed Grafana サービスを使用する場合は、管理者から必要な認証情報と許可が提供されます。多くの Amazon Managed Grafana 機能を使用して作業を行う場合は、追加の権限が必要になる場合があります。アクセスの管理方法を理解すると、管理者に適切なアクセス許可をリクエストするのに役に立ちます。Amazon Managed Grafana の機能にアクセスできない場合は、「Amazon Managed Grafana アイデンティティとアクセスのトラブルシューティング」を参照してください。

サービス管理者 – 社内の Amazon Managed Grafana リソースを担当している場合は、通常、Amazon Managed Grafana へのフルアクセスがあります。サービスのユーザーがどの Amazon Managed Grafana 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社で Amazon Managed Grafana と IAM を併用する方法の詳細については、「Amazon Managed Grafana と IAM の連携方法」を参照してください。

IAM 管理者 – 管理者は、Amazon Managed Grafana へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる Amazon Managed Grafana アイデンティティベースのポリシーの例を表示するには、「Amazon Managed Grafana のアイデンティティベースのポリシー例」を参照してください。

アイデンティティを使用した認証

認証は、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けることによって、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook 認証情報は、フェデレーション ID の例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 AWS サインイン ユーザーガイド」の「 へのサインイン AWS アカウント方法」を参照してください。

AWS プログラムで にアクセスする場合、 はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストを暗号化して署名します。AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。リクエストに自分で署名する推奨方法の使用については、「IAM ユーザーガイド」の「API リクエストに対するAWS Signature Version 4」を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。たとえば、 では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「多要素認証」および「IAM ユーザーガイド」の「IAM のAWS 多要素認証」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して一時的な認証情報 AWS のサービス を使用して にアクセスすることを要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを介して提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッド ID がにアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成するか、独自の ID ソースのユーザーとグループのセッ

トに接続して同期し、すべての AWS アカウント とアプリケーションで使用できます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内の ID です。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。で IAM ロールを一時的に引き受けるには AWS Management Console、ユーザーから IAM ロール (コンソール) に切り替えることができます。ロールを引き受けるには、 または AWS API オペレーションを AWS CLI 呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロール

については、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する」を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center User Guide」の「Permission sets」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。
- クロスサービスアクセス – 一部の は他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする と組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除することができます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに許可を委任するロールを作成する」を参照してください。
  - サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカ

ウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。EC2 インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは AWS 、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデン

ティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「管理ポリシーとインラインポリシーのいずれかを選択する」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、および Amazon VPC は AWS WAF、ACLs。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的でない追加のポリシータイプをサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPsは、 の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー (SCP)」を参照してください。
- リソースコントロールポリシー (RCP) – RCP は、所有する各リソースにアタッチされた IAM ポリシーを更新することなく、アカウント内のリソースに利用可能な最大数のアクセス許可を設定するために使用できる JSON ポリシーです。RCP は、メンバーアカウントのリソースのアクセス許可を制限し、組織に属しているかどうかにかかわらず AWS アカウントのルートユーザー、 を含む ID の有効なアクセス許可に影響を与える可能性があります。RCP をサポートする のリストを含む Organizations と RCP の詳細については、AWS Organizations RCPs「リソースコントロールポリシー (RCPs」を参照してください。 AWS のサービス
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「ポリシー評価ロジック」を参照してください。

Amazon Managed Grafana と IAM の連携方法

IAM を使用して Amazon Managed Grafana へのアクセスを管理する前に、Amazon Managed Grafana で使用できる IAM 機能について理解しておく必要があります。

Amazon Managed Grafana で使用できる IAM の機能

IAM 機能	Amazon Managed Grafana のサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	いいえ
ACL	いいえ
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	あり
転送アクセスセッション (FAS)	あり
サービスロール	はい
サービスリンクロール	はい

Amazon Managed Grafana およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

Amazon Managed Grafana のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

Amazon Managed Grafana のアイデンティティベースのポリシー例

Amazon Managed Grafana でのアイデンティティベースのポリシーの例は、「Amazon Managed Grafana のアイデンティティベースのポリシー例」を参照してください。

Amazon Managed Grafana 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必

要はありません。詳細については、「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」を参照してください。

Amazon Managed Grafana のポリシーアクション

ポリシーアクションのサポート:あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

Amazon Managed Grafana アクションの一覧については、「サービス認可リファレンス」の「Amazon Managed Grafana によって定義されるアクション」を参照してください。

Amazon Managed Grafana のポリシーアクションは、アクションの前に以下のプレフィックスを使用します。

```
grafana
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
      "grafana:action1",
      "grafana:action2"
         ]
```

Amazon Managed Grafana でのアイデンティティベースのポリシーの例は、「Amazon Managed Grafana のアイデンティティベースのポリシー例」を参照してください。

Amazon Managed Grafana のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ステートメントにはResource または NotResource 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

Amazon Managed Grafana リソースのタイプとその ARN のリストを確認するには、「サービス認可リファレンス」の「Amazon Managed Grafana で定義されるリソース」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「Amazon Managed Grafana で定義されるアクション」を参照してください。

Amazon Managed Grafana でのアイデンティティベースのポリシーの例は、「Amazon Managed Grafana のアイデンティティベースのポリシー例」を参照してください。

Amazon Managed Grafana のポリシー条件キー

サービス固有のポリシー条件キーへのサポート: なし

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「 グローバル条件コンテキストキー」を参照してください。

Amazon Managed Grafana のアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon Managed Grafana での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: あり

Amazon Managed Grafana はリソースおよびアイデンティティベースのタグ付けをサポートしています。

Amazon Managed Grafana リソースのタグ付けの詳細については、「Amazon Managed Grafana ワークスペースのタグを付け」を参照してください。

リソースのタグに基づいてリソースへのアクセスを制限するためのアイデンティティベースポリシーの例を表示するには、「AWS Amazon Managed Grafana の マネージドポリシー」を参照してください。

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/*key-name*、aws:RequestTag/*key-name*、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

Amazon Managed Grafana での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する機能などの詳細については、AWS のサービス「IAM ユーザーガイド」の「IAM と連携する」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

Amazon Managed Grafana のフォワードアクセスセッション

転送アクセスセッション (FAS) のサポート: あり

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクショ

ンがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストをリクエストする を使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Amazon Managed Grafana のサービスロール

サービスロールのサポート: あり

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに許可を委任するロールを作成する」を参照してください。

> **Warning**
>
> サービスロールのアクセス許可を変更すると、Amazon Managed Grafana が機能しなくなる可能性があります。Amazon Managed Grafana が指示するときにのみ、サービスロールを編集してください。

Amazon Managed Grafana のサービスリンクロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

Amazon Managed Grafana でのサービスリンクロールの作成または管理の詳細については、「Amazon Managed Grafana のサービスリンクロールの使用」を参照してください。

Amazon Managed Grafana のアイデンティティベースのポリシー例

デフォルトでは、ユーザーおよびロールには Amazon Managed Grafana リソースを作成または変更するアクセス許可がありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理

者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

Amazon Managed Grafana が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「Amazon Managed Grafana のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- Amazon Managed Grafana コンソールの使用
- Amazon Managed Grafana のサンプルポリシー

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウント内で誰かが Amazon Managed Grafana リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「ジョブ機能のAWS マネージドポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「IAM でのポリシーとアクセス許可」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエ

ストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素:条件」を参照してください。

- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「IAM Access Analyzer でポリシーを検証する」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「MFA を使用した安全な API アクセス」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「IAM でのセキュリティのベストプラクティス」を参照してください。

Amazon Managed Grafana コンソールの使用

コンソールにアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、 AWS アカウントのリソースの詳細をリストおよび表示できます。最小限の必要なアクセス許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しなくなります。

Amazon Managed Grafana のサンプルポリシー

このセクションでは、Amazon Managed Grafana のいくつかのシナリオに役立つ、ID ベースのポリシーについて説明します。

SAML を使用する Grafana 管理者

ユーザー認証に SAML を使用する場合、Amazon Managed Grafana を作成および管理する管理者には、次のポリシーが必要です。

- AWSGrafanaAccountAdministrator または Amazon Managed Grafana ワークスペースを作成および管理する同等のアクセス許可。

- Amazon Managed Grafana ワークスペースを Grafana Enterprise にアップグレードする場合は、AWSMarketplaceManageSubscriptions ポリシーまたは同等のアクセス許可。

IAM Identity Center を使用する管理アカウントの Grafana 管理者

組織全体で Amazon Managed Grafana ワークスペースを作成および管理するためのアクセス許可を付与し、IAM Identity Center などの依存関係を有効にするには、[AWSGrafanaAccountAdministrator]、[AWSSSOMasterAccountAdministrator] および [AWSSSODirectoryAdministrator] ポリシーをユーザーに割り当てます。さらに、Amazon Managed Grafana ワークスペースを Grafana Enterprise にアップグレードするには、ユーザーに [AWSMarketplaceManageSubscriptions] IAM ポリシーまたは同等のアクセス許可が必要です。

Amazon Managed Grafana ワークスペースの作成時にサービス管理のアクセス許可を使用する場合、ワークスペースを作成するユーザーに `iam:CreateRole`、`iam:CreatePolicy`、および `iam:AttachRolePolicy` 権限も必要です。これらの権限は、 AWS CloudFormation StackSets を使用して、組織のアカウント内のデータソースを読み取ることを許可するポリシーを展開するために必要です。

> **Important**
>
> ユーザーに、`iam:CreateRole`、`iam:CreatePolicy`、および `iam:AttachRolePolicy` アクセス許可を付与すると、そのユーザーには、 AWS アカウントへの完全な管理アクセス許可が与えられます。たとえば、これらのアクセス許可を持つユーザーは、すべてのリソースに対する完全なアクセス許可を持つポリシーを作成し、そのポリシーを任意のロールにアタッチできます。これらのアクセス許可を付与するユーザーには十分注意してください。

AWSGrafanaAccountAdministrator に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaAccountAdministrator」を参照してください。

IAM Identity Center を使用するメンバーアカウントの Grafana 管理者

組織のメンバーアカウントで Amazon Managed Grafana ワークスペースを作成および管理するためのアクセス許可を付与するには、[AWSGrafanaAccountAdministrator]、[AWSSSOMemberAccountAdministrator] および [AWSSSODirectoryAdministrator] ポリシーをユーザーに割り当てます。さらに、Amazon Managed Grafana ワークスペースを Grafana Enterprise にアップグレードするには、ユーザーに [AWSMarketplaceManageSubscriptions] IAM ポリシーまたは同等のアクセス許可が必要です。

Amazon Managed Grafana ワークスペースの作成時にサービス管理のアクセス許可を使用する場合、ワークスペースを作成するユーザーに iam:CreateRole、iam:CreatePolicy、および iam:AttachRolePolicy 権限も必要です。これらはユーザーがアカウント内のデータソースを読み取るために必要です。

> **Important**
>
> ユーザーに、iam:CreateRole、iam:CreatePolicy、および iam:AttachRolePolicy アクセス許可を付与すると、そのユーザーには、 AWS アカウントへの完全な管理アクセス許可が与えられます。たとえば、これらのアクセス許可を持つユーザーは、すべてのリソースに対する完全なアクセス許可を持つポリシーを作成し、そのポリシーを任意のロールにアタッチできます。これらのアクセス許可を付与するユーザーには十分注意してください。

AWSGrafanaAccountAdministrator に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaAccountAdministrator」を参照してください。

Amazon Managed Grafana ワークスペースとユーザーを IAM Identity Center を使用する単一スタンドアロンアカウントで作成および管理する

スタンドアロン AWS アカウントは、まだ組織のメンバーではないアカウントです。Organizations の詳細については、「AWS Organizationsとは」を参照してください。

スタンドアロンアカウントで Amazon Managed Grafana ワークスペースとユーザーを作成および管理するためのアクセス許可を付与するには、[AWSGrafanaAccountAdministrator]、[AWSSSOMasterAccountAdministrator]、[AWSOrganizationsFullA ポリシーをユーザーに割り当てます。さらに、Amazon Managed Grafana ワークスペースを Grafana Enterprise にアップグレードするには、ユーザーに [AWSMarketplaceManageSubscriptions] IAM ポリシーまたは同等のアクセス許可が必要です。

> **Important**
>
> ユーザーに、iam:CreateRole、iam:CreatePolicy、および iam:AttachRolePolicy アクセス許可を付与すると、そのユーザーには、 AWS アカウントへの完全な管理アクセス許可が与えられます。たとえば、これらのアクセス許可を持つユーザーは、すべてのリソースに対する完全なアクセス許可を持つポリシーを作成し、そのポリシーを任意のロールにアタッチできます。これらのアクセス許可を付与するユーザーには十分注意してください。

AWSGrafanaAccountAdministrator に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaAccountAdministrator」を参照してください。

Amazon Managed Grafana に対するアクセス許可のユーザーへの割り当ておよび割り当て解除

ワークスペースのユーザーに Grafana 管理者アクセス許可を付与するなど、アカウント内の Amazon Managed Grafana ワークスペースへの他のユーザーのアクセスを管理するアクセス許可を付与するには、[AWSGrafanaWorkspacePermissionManagementV2] ポリシーをそのユーザーに割り当てます。IAM Identity Center を使用してこのワークスペース内のユーザーを管理する場合、ユーザーには [AWSSSOReadOnly] ポリシーと [AWSSSODirectoryReadOnly] ポリシーも必要です。

AWSGrafanaWorkspacePermissionManagementV2 に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaWorkspacePermissionManagementV2 」を参照してください。

Amazon Managed Grafana の読み取り専用アクセス許可

ワークスペースの一覧および表示、Grafana ワークスペースコンソールを開くなどの読み取りアクションのアクセス許可を付与するには、[AWSGrafanaConsoleReadOnlyAccess]、[AWSSSOReadOnly] および [AWSSSODirectoryReadOnly] ポリシーをユーザーまたは IAM ロールに割り当てます。

AWSGrafanaConsoleReadOnlyAccess に付与された権限を確認するには、「AWS マネージドポリシー: AWSGrafanaConsoleReadOnlyAccess 」を参照してください。

AWS Amazon Managed Grafana の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: AWSGrafanaAccountAdministrator

AWSGrafanaAccountAdministrator ポリシーは、組織全体のワークスペースを作成および管理するための Amazon Managed Grafana 内のアクセスを提供すします。

ご使用の IAM エンティティに、AWSGrafanaAccountAdministrator をアタッチすることはできません。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `iam` – 管理者がロールをワークスペースに関連付けるだけでなく、ロールを Amazon Managed Grafana サービスに渡すように、プリンシパルが IAM ロールを一覧表示して取得できるようにします。
- `Amazon Managed Grafana` – プリンシパルがすべての Amazon Managed Grafana API に読み取りおよび書き込みアクセスできるようにします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSGrafanaOrganizationAdmin",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Sid": "GrafanaIAMGetRolePermission",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/*"
        },
        {
            "Sid": "AWSGrafanaPermissions",
            "Effect": "Allow",
            "Action": [
                "grafana:*"
```

```
            ],
            "Resource": "*"
        },
        {
            "Sid": "GrafanaIAMPassRolePermission",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "grafana.amazonaws.com"
                }
            }
        }
    ]
}
```

AWS マネージドポリシー: AWSGrafanaWorkspacePermissionManagement (廃止)

このポリシーは廃止されました。このポリシーを他のユーザー、グループ、ロールに適用しないでください。

Amazon Managed Grafana は、このポリシーに代わる新しいポリシー AWSGrafanaWorkspacePermissionManagementV2 を追加しました。この新しい管理ポリシーは、より制限の厳しいアクセス許可セットを提供し、ワークスペースのセキュリティが向上されます。

AWS マネージドポリシー: AWSGrafanaWorkspacePermissionManagementV2

AWSGrafanaWorkspacePermissionManagementV2 ポリシーは、Amazon Managed Grafana ワークスペースのユーザーのアクセス許可とグループのアクセス許可を更新する機能のみを提供します。

ご使用の IAM エンティティに、AWSGrafanaWorkspacePermissionManagementV2 をアタッチすることはできません。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `Amazon Managed Grafana` – プリンシパルが Amazon Managed Grafana ワークスペースのユーザーのアクセス許可およびグループのアクセス許可を読み取り更新できるようにします。

- `IAM Identity Center` – プリンシパルに IAM Identity Center のエンティティの読み取りを許可します。これはプリンシパルを Amazon Managed Grafana アプリケーションに関連付けるために欠かせない要素ですが、以下のポリシーリストの後に説明されている追加のステップも必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Sid": "AWSGrafanaPermissions",
            "Effect": "Allow",
            "Action": [
                "grafana:DescribeWorkspace",
                "grafana:DescribeWorkspaceAuthentication",
                "grafana:UpdatePermissions",
                "grafana:ListPermissions",
                "grafana:ListWorkspaces"
            ],
            "Resource": "arn:aws:grafana:*:*:/workspaces*"
        },
        {
            "Sid": "IAMIdentityCenterPermissions",
            "Effect": "Allow",
            "Action": [
                "sso:DescribeRegisteredRegions",
                "sso:GetSharedSsoConfiguration",
                "sso:ListDirectoryAssociations",
                "sso:GetManagedApplicationInstance",
                "sso:ListProfiles",
                "sso:GetProfile",
                "sso:ListProfileAssociations",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup"
            ],
            "Resource": "*"
        }
    ]
}
```

必要な追加ポリシー

権限の割り当てをユーザーに完全に許可するには、AWSGrafanaWorkspacePermissionManagementV2 ポリシーに加えて、IAM Identity Center のアプリケーション割り当てへのアクセスを提供するポリシーも割り当てる必要があります。

このポリシーの作成には、まずワークスペースの Grafana アプリケーション ARN を収集する必要があります。

1. IAM Identity Center コンソールを開きます。
2. 左のメニューで、[アプリケーション] をクリックします。
3. [AWS マネージド] タブで、[Amazon Grafana-workspace-name] という名前のアプリケーションを見つけます。workspace-name はワークスペースの名前です。アプリケーションの名前を選択します。
4. ワークスペースの Amazon Managed Grafana が管理する IAM Identity Center アプリケーションが表示されます。このアプリケーションの ARN は詳細ページに表示されます。これは arn:aws:sso::*owner-account-id*:application/ssoins-*unique-id*/apl-*unique-id* 形式です。

作成するポリシーは、次のようになります。*grafana-application-arn* を前のステップで見つけた ARN で置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sso:CreateApplicationAssignment",
                "sso:DeleteApplicationAssignment"
            ],
            "Resource": [
                "grafana-application-arn"
            ]
        }
    ]
}
```

ポリシーを作成して適用する方法の詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM アイデンティティの許可の追加および削除」を参照してください。

AWS マネージドポリシー: AWSGrafanaConsoleReadOnlyAccess

AWSGrafanaConsoleReadOnlyAccess ポリシーは、Amazon Managed Grafana の読み取り専用オペレーションへのアクセスを付与します。

ご使用の IAM エンティティに、AWSGrafanaConsoleReadOnlyAccess をアタッチすることはできません。

アクセス許可の詳細

このポリシーには、次の許可が含まれています。

- `Amazon Managed Grafana` – プリンシパルが Amazon Managed Grafana API に読み取り専用アクセスできるようにします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSGrafanaConsoleReadOnlyAccess",
      "Effect": "Allow",
      "Action": ["grafana:Describe*", "grafana:List*"],
      "Resource": "*"
    }
  ]
}
```

AWS マネージドポリシー: AmazonGrafanaRedshiftAccess

このポリシーは、Amazon Redshift と、Amazon Managed Grafana で Amazon Redshift プラグインを使用するために必要な依存関係へのスコープ付きアクセスを付与します。AmazonGrafanaRedshiftAccess ポリシーでは、ユーザーまたは IAM ロールが Grafana の Amazon Redshift データソースプラグインを使用できるようにします。Amazon Redshift データベースの一時認証情報はデータベースユーザー `redshift_data_api_user` を対象とし、シークレットにキー `RedshiftQueryOwner` がタグ付けされている場合、Secrets Manager の認証情報を取得できます。このポリシーは、`GrafanaDataSource` でタグ付けされた Amazon Redshift クラスターへのアクセスを許可します。カスタマー管理ポリシーを作成するときは、タグベースの認証はオプションです。

IAM エンティティに AmazonGrafanaRedshiftAccess をアタッチできます。Amazon Managed Grafana では、このポリシーはユーザーに代わってアクションを実行できることを Amazon Managed Grafana に許可するためのサービスロールにも添付されます。

アクセス許可の詳細

このポリシーには、次の許可が含まれています。

- Amazon Redshift – プリンシパルがクラスターを記述し、redshift_data_api_user という名前のデータベースユーザーの一時認証情報を取得できるようにします。
- Amazon Redshift-data – プリンシパルが GrafanaDataSource としてタグ付けされたクラスターでクエリを実行できるようにします。
- Secrets Manager – プリンシパルがシークレットを一覧表示し、RedshiftQueryOwner としてタグ付けされたシークレットのシークレット値を読み取ることを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "redshift:DescribeClusters",
        "redshift-data:GetStatementResult",
        "redshift-data:DescribeStatement",
        "secretsmanager:ListSecrets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "redshift-data:DescribeTable",
        "redshift-data:ExecuteStatement",
        "redshift-data:ListTables",
        "redshift-data:ListSchemas"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/GrafanaDataSource": "false"
        }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": "redshift:GetClusterCredentials",
      "Resource": [
        "arn:aws:redshift:*:*:dbname:*/*",
        "arn:aws:redshift:*:*:dbuser:*/redshift_data_api_user"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "secretsmanager:ResourceTag/RedshiftQueryOwner": "false"
        }
      }
    }
  ]
}
```

AWS マネージドポリシー: AmazonGrafanaAthenaAccess

このポリシーは、Athena と、Amazon Managed Grafana の Athena プラグインからクエリを実行して Amazon S3 に結果を書き込むために必要な依存関係へのアクセス権を付与します。AmazonGrafanaAthenaAccess ポリシーでは、ユーザーまたは IAM ロールが Grafana の Athena データソースプラグインを使用できるようにします。Athena ワークグループにアクセスするには、`GrafanaDataSource` でタグ付けする必要があります。このポリシーには、`grafana-athena-query-results-` というプレフィックスが付いた名前の Amazon S3 バケットにクエリ結果を書き込むためのアクセス許可が含まれています。Athena クエリの基盤となるデータソースにアクセスするための Amazon S3 のアクセス許可は、このポリシーに含まれません。

ご使用の IAM エンティティに、AWSGrafanaAthenaAccess をアタッチすることはできません。Amazon Managed Grafana では、このポリシーはユーザーに代わってアクションを実行できることを Amazon Managed Grafana に許可するためのサービスロールにも添付されます。

アクセス許可の詳細

このポリシーには、次の許可が含まれています。

- Athena – プリンシパルが GrafanaDataSource とタグ付けされたワークグループ内の Athena リソースに対してクエリを実行できるようにします。
- Amazon S3 – プリンシパルが、grafana-athena-query-results- でプレフィックスが付けられたバケットにクエリ結果を読み取りおよび書き込みできるようにします。
- AWS Glue – プリンシパルに AWS Glue データベース、テーブル、パーティションへのアクセスを許可します。これは、プリンシパルが Athena で AWS Glue Data Catalog を使用するのに必要です。

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
         "Effect": "Allow",
         "Action": [
             "athena:GetDatabase",
             "athena:GetDataCatalog",
             "athena:GetTableMetadata",
             "athena:ListDatabases",
             "athena:ListDataCatalogs",
             "athena:ListTableMetadata",
             "athena:ListWorkGroups"
         ],
         "Resource": [
             "*"
         ]
     },
     {
         "Effect": "Allow",
         "Action": [
             "athena:GetQueryExecution",
             "athena:GetQueryResults",
             "athena:GetWorkGroup",
             "athena:StartQueryExecution",
             "athena:StopQueryExecution"
         ],
         "Resource": [
             "*"
         ],
         "Condition": {
```

```
            "Null": {
                "aws:ResourceTag/GrafanaDataSource": "false"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "glue:GetDatabase",
            "glue:GetDatabases",
            "glue:GetTable",
            "glue:GetTables",
            "glue:GetPartition",
            "glue:GetPartitions",
            "glue:BatchGetPartition"
        ],
        "Resource": [
            "*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts",
            "s3:AbortMultipartUpload",
            "s3:CreateBucket",
            "s3:PutObject",
            "s3:PutBucketPublicAccessBlock"
        ],
        "Resource": [
            "arn:aws:s3:::grafana-athena-query-results-*"
        ]
    }
 ]
}
```

AWS マネージドポリシー: AmazonGrafanaCloudWatchAccess

このポリシーは、Amazon CloudWatch と、CloudWatch を Amazon Managed Grafana 内のデータソースとして使用するために必要な依存関係へのアクセス権を付与します。

ご使用の IAM エンティティに、AWSGrafanaCloudWatchAccess をアタッチすることはできません。Amazon Managed Grafana では、このポリシーはユーザーに代わってアクションを実行できることを Amazon Managed Grafana に許可するためのサービスロールにも添付されます。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `CloudWatch` — プリンシパルが Amazon CloudWatch のメトリクスデータとログを一覧表示および取得できるようにします。また、CloudWatch クロスアカウントオブザーバビリティで、ソースアカウントから共有されたデータを表示することもできます。
- `Amazon EC2` – モニタリング中のリソースに関する詳細をプリンシパルが取得できるようにします。
- `Tags` – プリンシパルがリソースのタグにアクセスして、CloudWatch メトリクスクエリをフィルタリングできるようにします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmsForMetric",
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:GetMetricData",
        "cloudwatch:GetInsightRuleReport"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "logs:DescribeLogGroups",
        "logs:GetLogGroupFields",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:GetQueryResults",
        "logs:GetLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeRegions"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "tag:GetResources",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "oam:ListSinks",
        "oam:ListAttachedLinks"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS マネージドポリシーに対する Amazon Managed Grafana の更新

このサービスがこれらの変更の追跡を開始してからの Amazon Managed Grafana の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動通知を入手するには、Amazon Managed Grafana ドキュメントの履歴ページから、RSS フィードにサブスクライブしてください。

変更	説明	日付
AWSGrafanaWorkspacePermissionManagement – 廃止	このポリシーは AWSGrafanaWorkspacePermissionManagementV2 に置き換えられました。 このポリシーは廃止と見なされ、更新されなくなります。新しいポリシーでは、より制限の厳しいアクセス許可セットが提供されるので、ワークスペースのセキュリティが向上されます。	2024 年 1 月 5 日
AWSGrafanaWorkspacePermissionManagementV2 – 新しいポリシー	Amazon Managed Grafana は、古い [AWSGrafanaWorkspacePermissionManagement] ポリシーに代わる新しいポリシー [AWSGrafanaWorkspacePermissionManagementV2] を追加しました。この新しい管理ポリシーは、より制限の厳しいアクセス許可セットを提供し、ワークスペースのセキュリティが向上されます。	2024 年 1 月 5 日
AmazonGrafanaCloudWatchAccess – 新しいポリシー	Amazon Managed Grafana が新しいポリシー [AmazonGrafanaCloudWatchAccess] を追加しました。	2023 年 3 月 24 日
AWSGrafanaWorkspacePermissionManagement – 既存のポリシーの更新	Amazon Managed Grafana は、Active Directory の IAM Identity Center ユーザーとグループを Grafana ワー	2023 年 3 月 14 日

変更	説明	日付
	クスペースに関連付けることができるようにするため、新しいアクセス許可 [AWSGrafanaWorkspacePermissionManagement] を追加しました。 アクセス許可 `sso-directory:DescribeUser` 、`sso-directory:DescribeGroup` が追加されました。	

変更	説明	日付
AWSGrafanaWorkspacePermissionManagement – 既存のポリシーの更新	Amazon Managed Grafana は、IAM Identity Center ユーザーとグループを Grafana ワークスペースに関連付けることができるようにするため、新しいアクセス許可 [AWSGrafanaWorkspacePermissionManagement] を追加しました。 アクセス許可 `sso:DescribeRegisteredRegions` 、`sso:GetSharedSsoConfiguration` 、`sso:ListDirectoryAssociations` 、`sso:GetManagedApplicationInstance` 、`sso:ListProfiles` 、`sso:AssociateProfile` 、`sso:DisassociateProfile` 、`sso:GetProfile` 、`sso:ListProfileAssociations` が追加されました。	2022 年 12 月 20 日
AmazonGrafanaServiceLinkedRolePolicy – 新しい SLR ポリシー	Amazon Managed Grafana は、Grafana サービスにリンクされたロール [AmazonGrafanaServiceLinkedRolePolicy] の新しいポリシーを追加しました。	2022 年 11 月 18 日

変更	説明	日付
AWSGrafanaAccountAdministrator、AWSGrafanaConsoleReadOnlyAccess	すべての Amazon Managed Grafana リソースへのアクセスの許可	2022 年 2 月 17 日
AmazonGrafanaRedshiftAccess – 新しいポリシー	Amazon Managed Grafana が新しいポリシー [AmazonGrafanaRedshiftAccess] を追加しました。	2021 年 11 月 26 日
AmazonGrafanaAthenaAccess – 新しいポリシー	Amazon Managed Grafana が新しいポリシー [AmazonGrafanaAthenaAccess] を追加しました。	2021 年 11 月 22 日
AWSGrafanaAccountAdministrator - 既存ポリシーへの更新。	Amazon Managed Grafana は [AWSGrafanaAccountAdministrator] からアクセス許可を削除しました。 `sso.amazonaws.com` サービスにスコープされた `iam:CreateServiceLinkedRole` アクセス許可が削除されました。代わりに、[AWSSSOMasterAccountAdministrator] ポリシーをアタッチして、このアクセス許可をユーザーに付与することをお勧めします。	2021 年 10 月 13 日

変更	説明	日付
AWSGrafanaWorkspacePermissionManagement – 既存ポリシーへの更新	Amazon Managed Grafana は、このポリシーを持つユーザーがワークスペースに関連付けられた認証方法を表示できるように、[AWSGrafanaWorkspacePermissionManagement] に新しいアクセス許可を追加しました。 `grafana:DescribeWorkspaceAuthentication` アクセス許可が追加されました。	2021 年 9 月 21 日
AWSGrafanaConsoleReadOnlyAccess – 既存ポリシーへの更新	Amazon Managed Grafana は、このポリシーを持つユーザーがワークスペースに関連付けられた認証方法を表示できるように、[AWSGrafanaConsoleReadOnlyAccess] に新しいアクセス許可を追加しました。 `grafana:Describe*` および `grafana:List*` のアクセス許可がポリシーに追加され、以前の狭いアクセス許可 `grafana:DescribeWorkspace` 、`grafana:ListPermissions` 、`grafana:ListWorkspaces` が置き換えられます。	2021 年 9 月 21 日

変更	説明	日付
Amazon Managed Grafana が変更の追跡を開始	Amazon Managed Grafana は AWS 、管理ポリシーの変更の追跡を開始しました。	2021 年 9 月 9 日

Amazon Managed Grafana アイデンティティとアクセスのトラブルシューティング

次の情報は、Amazon Managed Grafana と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- Amazon Managed Grafana でアクションを実行する権限がない
- iam:PassRole を実行する権限がありません
- AWS アカウント外のユーザーに Amazon Managed Grafana リソースへのアクセスを許可したい

Amazon Managed Grafana でアクションを実行する権限がない

アクションを実行する権限がないというエラーが表示された場合は、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要なgrafana:*GetWidget* アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 grafana:GetWidget on resource: my-example-widget
```

この場合、grafana:*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

iam:PassRole を実行する権限がありません

`iam:PassRole` アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新してAmazon Managed Grafana にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、`marymajor` という IAM ユーザーがコンソールを使用して Amazon Managed Grafana でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

AWS アカウント外のユーザーに Amazon Managed Grafana リソースへのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Amazon Managed Grafana がこれらの機能をサポートしているかどうかを確認するには、「Amazon Managed Grafana と IAM の連携方法」を参照してください。
- 所有 AWS アカウント している 全体のリソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。

- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、「IAM ユーザーガイド」の「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、「IAM ユーザーガイド」の「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

サービス間の混乱した代理の防止

混乱した代理問題は、アクションを実行するためのアクセス許可を持たないエンティティが、より特権のあるエンティティにアクションの実行を強制できてしまう場合に生じる、セキュリティ上の問題です。では AWS、サービス間のなりすましにより、混乱した代理問題が発生する可能性があります。サービス間でのなりすましは、1 つのサービス (呼び出し元サービス) が、別のサービス (呼び出し対象サービス) を呼び出すときに発生する可能性があります。呼び出し元サービスは、本来ならアクセスすることが許可されるべきではない方法でその許可を使用して、別のお客様のリソースに対する処理を実行するように操作される場合があります。これを防ぐため、 AWS では、アカウントのリソースへのアクセス権が付与されたサービスプリンシパルで、すべてのサービスのデータを保護するために役立つツールを提供しています。

リソースポリシー内では `aws:SourceArn` および `aws:SourceAccount` のグローバル条件コンテキストキーを使用して、Amazon Managed Grafana が別のサービスに付与する、リソースへのアクセス許可を制限することをお勧めします。`aws:SourceArn` の値に Amazon S3 バケット ARN などのアカウント ID が含まれていない場合は、両方のグローバル条件コンテキストキーを使用して、アクセス許可を制限する必要があります。同じポリシーステートメントでこれらのグローバル条件コンテキストキーの両方を使用し、アカウント ID に`aws:SourceArn` の値が含まれていない場合、`aws:SourceAccount` 値と `aws:SourceArn` 値の中のアカウントには、同じアカウント ID を使用する必要があります。クロスサービスのアクセスにリソースを 1 つだけ関連付けたい場合は、`aws:SourceArn` を使用します。そのアカウント内のリソースをクロスサービスの使用に関連付けることを許可する場合は、`aws:SourceAccount` を使用します。

`aws:SourceArn` の値は、Amazon Managed Grafana ワークスペースの ARN である必要があります。

混乱した代理問題から保護するための最も効果的な方法は、リソースの完全な ARN を指定して `aws:SourceArn` グローバル条件コンテキストキーを使用することです。リソースの完全

な ARN が不明な場合や、複数のリソースを指定する場合は、aws:SourceArn グローバルコンテキスト条件キーを使用して、ARN の未知部分をワイルドカード (*) で表します。例えば、arn:aws:grafana:*:*123456789012*:* と指定します。

次の例は、Amazon Managed Grafana の IAM ロールの信頼ポリシーで aws:SourceArn および aws:SourceAccount グローバル条件コンテキストキーを使用して、「混乱した代理」問題を回避する方法を示します。

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Principal": {
       "Service": "grafana.amazonaws.com"
     },
     "Action": "sts:AssumeRole",
     "Condition": {
       "StringEquals": {
         "aws:SourceAccount": "accountId",
         "aws:SourceArn": "arn:aws:grafana:region:accountId:/workspaces/workspaceId"
       }
     }
   }
 ]
}
```

Amazon Managed Grafana のサービスリンクロールの使用

Amazon Managed Grafana は AWS Identity and Access Management 、(IAM) サービスにリンクされたロールを使用します。サービスにリンクされたロールは、Amazon Managed Grafana に直接リンクされた特殊な IAM ロールです。サービスにリンクされたロールは Amazon Managed Grafana によって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

必要な許可を手動で追加する必要がないため、サービスリンクロールは Amazon Managed Grafana のセットアップを容易にします。サービスリンクロールの許可は Amazon Managed Grafana が定義し、別段の定義がない限り、Amazon Managed Grafana のみがそのロールを引き受けることがで

きます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これは、リソースにアクセスするための許可を誤って削除できないため、Amazon Managed Grafana リソースを保護します。

サービスにリンクされたロールをサポートする他のサービスの詳細については、AWS「IAM と連携するサービス」を参照し、「サービスにリンクされたロール」列で「はい」を持つサービスを探します。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[あり] (はい) リンクを選択します。

Amazon Managed Grafana 向けのサービスリンクロールのアクセス許可

Amazon Managed Grafana は、AmazonManagedGrafana という名前のサービスリンクロールを使用します。Amazon Managed Grafana は、このロールを使用して、カスタマーアカウント内で ENIsや Secrets Manager シークレットなどのリソースを作成および設定します。AmazonManagedGrafana サービスリンクロールは、ロールの引き受けについて以下のサービスを信頼します。

- `grafana.amazonaws.com`

AmazonManagedGrafana サービスにリンクされたロールがAmazonGrafanaServiceLinkedRolePolicyポリシーにアタッチされます。このポリシーの更新については、「AWS マネージドポリシーに対する Amazon Managed Grafana の更新」を参照してください

ロールのアクセス許可ポリシーは、指定したリソースに対して以下のアクションを実行することを Amazon Managed Grafana に許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterface",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "AmazonGrafanaManaged"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
                "Condition": {
                    "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface"
                },
                "Null": {
                    "aws:RequestTag/AmazonGrafanaManaged": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteNetworkInterface",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "ec2:ResourceTag/AmazonGrafanaManaged": "false"
                }
            }
        }
    ]
}
```

サービスリンク役割の作成、編集、削除を IAM エンティティ (ユーザー、グループ、役割など) に許可するにはアクセス許可を設定する必要があります。詳細については、「IAM User Guide(IAM ユーザーガイド)」 の「Service-linked role permissions 」 を参照してください。

Amazon Managed Grafana のサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。 AWS Management Console、、 AWS CLIまたは AWS API で VpcConfiguration を使用して CreateWorkspace を呼び出すと、Amazon Managed Grafana によってサービスにリンクされたロールが作成されます。

> **Important**
>
> このサービスリンク役割はこの役割でサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。また、サービスにリンクされたロールのサポートが開始された 2022 年 11 月 30 日より前に Amazon Managed Grafana サービスを使用していた場合、Amazon Managed Grafana はアカウントにおいて AmazonManagedGrafana ロールを作成しました。詳細については、「IAM アカウントに新しいロールが表示される」を参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ手順でアカウントにロールを再作成できます。VpcConfiguration を使用して CreateWorkspace を呼び出すと、Amazon Managed Grafana はサービスリンクロールを再度作成します。

Grafana ユースケースでサービスリンクロールを作成するには、IAM コンソールも使用できます。AWS CLI または AWS API で、サービス名を使用して`grafana.amazonaws.com`サービスにリンクされたロールを作成します。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

Amazon Managed Grafana のサービスリンクロールの編集

Amazon Managed Grafana では、AmazonManagedGrafana サービスリンクロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」を参照してください。

Amazon Managed Grafana のサービスリンクロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、積極的にモニタリングまたは保守されていない未使用のエンティティを排除できます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

> **Note**
>
> リソースを削除しようとしているときに Amazon Managed Grafana サービスがロールを使用している場合は、削除が失敗する可能性があります。失敗した場合は数分待ってから操作を再試行してください。

AmazonManagedGrafana で使用される Amazon Managed Grafana リソースを削除するには

1. `Region` AWS コンソールで のすべてのワークスペースビューに移動します。
2. `Region` 内のすべてのワークスペースを削除します。各ワークスペースのラジオボタンを確認し、[すべてのワークスペース] ビューの右上にある [削除] ボタンを選択する必要があります。すべてのワークスペースが `Region` から削除されるまで、各ワークスペースの削除を繰り返します。Amazon Managed Grafana でワークスペースを削除する方法の詳細については、このユーザーガイドの「ワークスペースの削除」を参照してください。

> **Note**
>
> ワークスペース AWS リージョン がある ごとにこの手順を繰り返します。ただし、サービスリンクロールを削除する前に、[すべてのリージョンの]すべてのワークスペースを削除する必要があります。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、 AWS CLI、または AWS API を使用して、AmazonManagedGrafana サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

Amazon Managed Grafana サービスリンクロールがサポートされるリージョン

Amazon Managed Grafana は、サービスを利用できるすべてのリージョンで、サービスにリンクされたロールの使用をサポートします。詳細については、「AWS リージョンとエンドポイント」を参照してください。

AWS データソースの Amazon Managed Grafana アクセス許可とポリシー

Amazon Managed Grafana には、次の 3 つのアクセス許可モードがあります。

- 現在のアカウントの「サービス管理」のアクセス許可
- 組織の「サービス管理」のアクセス許可
- 「カスタマー管理」のアクセス許可

ワークスペースの作成時に、使用するアクセス許可モードを選択します。ここで選択した内容は後で変更できます。

どちらのサービスマネージドアクセス許可モードでも、Amazon Managed Grafana は、アカウントまたは組織の AWS データソースにアクセスして検出するために必要なロールとポリシーを作成します。作成後、これらのポリシーは IAM コンソールで編集できます。

サービス管理のアクセス許可 (単一アカウント用)

このモードを選択した場合、Amazon Managed Grafana に AmazonGrafanaServiceRole -***random-id*** というロールが作成されます。次に、Amazon Managed Grafana は、Amazon Managed Grafana ワークスペースからアクセスするために選択した AWS サービスごとに、このロールにポリシーをアタッチします。

CloudWatch

Amazon Managed Grafana は、 AWS 管理ポリシー AmazonGrafanaCloudWatchAccess をアタッチします。

> **Note**
>
> AmazonGrafanaCloudWatchAccess 管理のポリシーが作成される前に CloudWatch を使用していたワークスペースには、AmazonGrafanaCloudWatchPolicy -***random-id*** という名前のカスタマー管理のポリシーが作成されました。

Amazon OpenSearch Service

Amazon Managed Grafana によって、AmazonGrafanaOpenSearchPolicy -***random-id*** という名前のカスタマー管理のポリシーが作成されます。データソースへのアクセスには、Get/Post アクセス許可が必要です。List/Describe アクセス許可は、データソースの検出に Amazon Managed Grafana によって使用されますが、データソースプラグインの動作には必須ではありません。このポリシーの内容は次のとおりです。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "es:ESHttpGet",
                "es:DescribeElasticsearchDomains",
                "es:ListDomainNames"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttpPost",
            "Resource": [
                "arn:aws:es:*:*:domain/*/_msearch*",
                "arn:aws:es:*:*:domain/*/_opendistro/_ppl"
            ]
        }
    ]
}
```

AWS IoT SiteWise

Amazon Managed Grafana は、 AWS 管理ポリシー AWSIoTSiteWiseReadOnlyAccess をアタッチします。

Amazon Redshift

Amazon Managed Grafana は、 AWS 管理ポリシー AmazonGrafanaRedshiftAccess をアタッチします。

Amazon Athena

Amazon Managed Grafana は、 AWS 管理ポリシー AmazonGrafanaAthenaAccess をアタッチします。

Amazon Managed Service for Prometheus

Amazon Managed Grafana によって、AmazonGrafanaPrometheusPolicy-***random-id*** という名前のカスタマー管理のポリシーが作成されます。List/Describe アクセス許可は、データソースの検出に Amazon Managed Grafana によって使用されますが、プラグインの動作には必須ではありません。このポリシーの内容は次のとおりです。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:ListWorkspaces",
                "aps:DescribeWorkspace",
                "aps:QueryMetrics",
                "aps:GetLabels",
                "aps:GetSeries",
                "aps:GetMetricMetadata"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon SNS

Amazon Managed Grafana によって、AmazonGrafanaSNSPolicy-***random-id*** という名前のカスタマー管理のポリシーが作成されます。このポリシーにより、 アカウント内で使用できる SNS

トピックが文字列 grafana で始まるものに制限されます。独自のポリシーを作成する場合、これはは必要ありません。このポリシーの内容は次のとおりです。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:*:accountId:grafana*"
            ]
        }
    ]
}
```

Timestream

Amazon Managed Grafana は、 AWS 管理ポリシー AmazonTimestreamReadOnlyAccess をアタッチします。

X-Ray

Amazon Managed Grafana は、 AWS 管理ポリシー AWSXrayReadOnlyAccess をアタッチします。

組織の「サービス管理」のアクセス許可

このモードは、組織内の管理アカウントまたは被委任管理者アカウントで作成されたワークスペースにのみ対応しています。被委任管理者アカウントでは、組織のスタックセットの作成や管理が可能です。被委任管理者の追加の詳細については、「被委任管理者アカウントの登録」を参照してください。

> **Note**
>
> 組織の管理アカウントに Amazon Managed Grafana ワークスペースなどのリソースを作成することは、 AWS セキュリティのベストプラクティスに違反します。

このモードでは、Amazon Managed Grafana は、 AWS 組織内の他のアカウントのリソースにアクセス AWS するために必要なすべての IAM ロールを作成します。Amazon Managed Grafana によって、選択した組織単位の各アカウントに AmazonGrafanaOrgMemberRole -***random-id*** というロールが作成されます。このロールの作成は、 AWS CloudFormation StackSets との統合を通じて実行されます。

このロールには、ワークスペースで使用するように選択した AWS データソースごとにポリシーがアタッチされています。これらのデータポリシーの内容については、「サービス管理のアクセス許可 (単一アカウント用) 」を参照してください。

Amazon Managed Grafana によって、組織の管理アカウントに AmazonGrafanaOrgAdminRole -***random-id*** というロールも作成されます。このロールは、Amazon Managed Grafana ワークスペースが組織内の他のアカウントにアクセスすることを許可します。 AWS サービス通知チャネルポリシーもこのロールにアタッチされます。ワークスペースの [AWS データソース] メニューを使用すると、ワークスペースがアクセスできる各アカウントのデータソースをすばやくプロビジョニングされます

このモードを使用するには、組織内の AWS 信頼されたサービスとして AWS CloudFormation Stacksets を有効にする必要があります。詳細については、「 で信頼されたアクセスを有効にする AWS Organizations」を参照してください。

以下は、AmazonGrafanaStackSet-***random-id*** スタックセットの内容です。

```
Parameters:
  IncludePrometheusPolicy:
    Description: Whether to include Amazon Prometheus access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeAESPolicy:
    Description: Whether to include Amazon Elasticsearch access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeCloudWatchPolicy:
    Description: Whether to include CloudWatch access in the role
    Type: String
    AllowedValues:
```

```
      - true
      - false
    Default: false
  IncludeTimestreamPolicy:
    Description: Whether to include Amazon Timestream access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeXrayPolicy:
    Description: Whether to include AWS X-Ray access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeSitewisePolicy:
    Description: Whether to include AWS IoT SiteWise access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeRedshiftPolicy:
    Description: Whether to include Amazon Redshift access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  IncludeAthenaPolicy:
    Description: Whether to include Amazon Athena access in the role
    Type: String
    AllowedValues:
      - true
      - false
    Default: false
  RoleName:
    Description: Name of the role to create
    Type: String
  AdminAccountId:
    Description: Account ID of the Amazon Grafana org admin
    Type: String
```

```
Conditions:
  addPrometheus: !Equals [!Ref IncludePrometheusPolicy, true]
  addAES: !Equals [!Ref IncludeAESPolicy, true]
  addCloudWatch: !Equals [!Ref IncludeCloudWatchPolicy, true]
  addTimestream: !Equals [!Ref IncludeTimestreamPolicy, true]
  addXray: !Equals [!Ref IncludeXrayPolicy, true]
  addSitewise: !Equals [!Ref IncludeSitewisePolicy, true]
  addRedshift: !Equals [!Ref IncludeRedshiftPolicy, true]
  addAthena: !Equals [!Ref IncludeAthenaPolicy, true]

Resources:
  PrometheusPolicy:
    Type: AWS::IAM::Policy
    Condition: addPrometheus
    Properties:
      Roles:
       - !Ref GrafanaMemberServiceRole
      PolicyName: AmazonGrafanaPrometheusPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - aps:QueryMetrics
              - aps:GetLabels
              - aps:GetSeries
              - aps:GetMetricMetadata
              - aps:ListWorkspaces
              - aps:DescribeWorkspace
            Resource: '*'

  AESPolicy:
    Type: AWS::IAM::Policy
    Condition: addAES
    Properties:
      Roles:
       - !Ref GrafanaMemberServiceRole
      PolicyName: AmazonGrafanaElasticsearchPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: AllowReadingESDomains
            Effect: Allow
            Action:
```

```
              - es:ESHttpGet
              - es:ESHttpPost
              - es:ListDomainNames
              - es:DescribeElasticsearchDomains
            Resource: '*'

  CloudWatchPolicy:
    Type: AWS::IAM::Policy
    Condition: addCloudWatch
    Properties:
      Roles:
       - !Ref GrafanaMemberServiceRole
      PolicyName: AmazonGrafanaCloudWatchPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: AllowReadingMetricsFromCloudWatch
            Effect: Allow
            Action:
              - cloudwatch:DescribeAlarmsForMetric
              - cloudwatch:DescribeAlarmHistory
              - cloudwatch:DescribeAlarms
              - cloudwatch:ListMetrics
              - cloudwatch:GetMetricStatistics
              - cloudwatch:GetMetricData
              - cloudwatch:GetInsightRuleReport
            Resource: "*"
          - Sid: AllowReadingLogsFromCloudWatch
            Effect: Allow
            Action:
              - logs:DescribeLogGroups
              - logs:GetLogGroupFields
              - logs:StartQuery
              - logs:StopQuery
              - logs:GetQueryResults
              - logs:GetLogEvents
            Resource: "*"
          - Sid: AllowReadingTagsInstancesRegionsFromEC2
            Effect: Allow
            Action:
              - ec2:DescribeTags
              - ec2:DescribeInstances
              - ec2:DescribeRegions
            Resource: "*"
```

```
          - Sid: AllowReadingResourcesForTags
            Effect: Allow
            Action:
              - tag:GetResources
            Resource: "*"
  GrafanaMemberServiceRole:
    Type: 'AWS::IAM::Role'
    Properties:
      RoleName: !Ref RoleName
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              AWS: !Sub arn:aws:iam::${AdminAccountId}:root
            Action:
              - 'sts:AssumeRole'
      Path: /service-role/
      ManagedPolicyArns:
        - !If [addTimestream, arn:aws:iam::aws:policy/AmazonTimestreamReadOnlyAccess, !
Ref AWS::NoValue]
        - !If [addXray, arn:aws:iam::aws:policy/AWSXrayReadOnlyAccess, !Ref
 AWS::NoValue]
        - !If [addSitewise, arn:aws:iam::aws:policy/AWSIoTSiteWiseReadOnlyAccess, !Ref
 AWS::NoValue]
        - !If [addRedshift, arn:aws:iam::aws:policy/service-role/
AmazonGrafanaRedshiftAccess, !Ref AWS::NoValue]
        - !If [addAthena, arn:aws:iam::aws:policy/service-role/
AmazonGrafanaAthenaAccess, !Ref AWS::NoValue]
```

以下は AmazonGrafanaOrgAdminPolicy-***random-id*** の内容です。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "organizations:ListAccountsForParent",
            "organizations:ListOrganizationalUnitsForParent"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
```

```
                "aws:PrincipalOrgID": "o-organizationId"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "sts:AssumeRole"
        ],
        "Resource": "arn:aws:iam::*:role/service-role/
AmazonGrafanaOrgMemberRole-random-Id"
    }]
}
```

「カスタマー管理」のアクセス許可

カスタマー管理のアクセス許可を使用する場合、Amazon Managed Grafana ワークスペースを作成する際に、アカウントで既存の IAM ロールを指定します。ロールには、`grafana.amazonaws.com` を信頼する信頼ポリシーが必要です。

以下に、このようなポリシーの例を示します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "grafana.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

そのロールがそのアカウントの AWS データソースまたは通知チャネルにアクセスするには、このセクションで前述したポリシーのアクセス許可が必要です。例えば、CloudWatch データソースを使用するには、サービス管理のアクセス許可 (単一アカウント用) に記載されている CloudWatch ポリシーのアクセス許可が必要です。

サービス管理のアクセス許可 (単一アカウント用) で挙げた Amazon OpenSearch Service および Amazon Managed Service for Prometheus のポリシーの `List` および `Describe` アクセス許可は、

データソースの検出とプロビジョニングが正しく機能するためにのみ必要です。これらのデータソースを手動で設定するだけの場合は必要ありません。

クロスアカウントアクセス

アカウント 111111111111 でワークスペースを作成する場合、アカウント 1111111111111 のロールを指定する必要があります。この例では、このロール WorkspaceRole を呼び出します。アカウント 999999999999 のデータにアクセスするには、アカウント 999999999999 にロールを作成する必要があります。その DataSourceRole を呼び出します。次に、WorkspaceRole と DataSourceRole の間に信頼関係を確立する必要があります。2 つのロール間の信頼を確立する方法の詳細については、「IAM チュートリアル: IAM ロールを使用して AWS アカウント間のアクセスを委任する」を参照してください。

DataSourceRole には、使用する各データソースについて、このセクションの冒頭に挙げたポリシーステートメントを含める必要があります。信頼関係が確立されたら、ワークスペース AWS 内の任意のデータソースのデータソース設定ページのロール ARN の引き受けフィールドで DataSourceRole (arn:aws:iam::999999999999:role:DataSourceRole) の ARN を指定できます。 その後、データソースは DataSourceRole] で定義されているアクセス許可でアカウント 999999999999 にアクセスします。

IAM 許可

Amazon Managed Grafana のアクションとデータにアクセスするには、認証情報が必要です。これらの認証情報には、クラウド AWS リソースに関する Amazon Managed Grafana データの取得など、アクションを実行し、リソースにアクセスするためのアクセス許可が必要です。以下のセクションでは、 AWS Identity and Access Management と Amazon Managed Grafana を使用して、リソースにアクセスできるユーザーを制御することでリソースを保護する方法について詳しく説明します。詳細については、 「IAM のポリシーとアクセス許可」を参照してください。

Amazon Managed Grafana のアクセス許可

次の表は、Amazon Managed Grafana で実行される可能性のあるアクションと、そのアクションに必要なアクセス許可を示しています。

アクション	必要なアクセス許可
Amazon Managed Grafana ワークスペースを作成する。ワークスペースは、メトリクス、ログ、トレースを作成および視	`grafana:CreateWorkspace`

アクション	必要なアクセス許可
覚化するために使用される論理的に分離された Grafana サーバーです。	
Amazon Managed Grafana ワークスペースを削除する。	`grafana:DeleteWorkspace`
Amazon Managed Grafana ワークスペースに関する詳細情報を取得する。	`grafana:DescribeWorkspace`
ワークスペースに関連付けられた認証設定を取得する。	`grafana:DescribeWorkspaceAuthentication`
ワークスペースのユーザーとグループに関連付けられたアクセス許可のリストを取得する。	`grafana:ListPermissions`
アカウント内に存在する Amazon Managed Grafana ワークスペースのリストを取得する。	`grafana:ListWorkspaces`
ワークスペースのユーザーとグループに関連付けられたアクセス許可を更新する。	`grafana:UpdatePermissions`
Amazon Managed Grafana ワークスペースを更新する。	`grafana:UpdateWorkspace`
ワークスペースに関連付けられた認証設定を更新する。	`grafana:UpdateWorkspaceAuthentication`
Grafana エンタープライズライセンスをワークスペースに関連付ける。	`grafana:AssociateLicense`

Amazon Managed Grafana のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、AWS のサービス 「コンプライアンスプログラムによる範囲内」を参照して、関心のある

コンプライアンスプログラムを選択します。一般的な情報については、AWS 「 Compliance ProgramsAssurance」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 は、コンプライアンスに役立つ以下のリソース AWS を提供します。

- セキュリティのコンプライアンスとガバナンス – これらのソリューション実装ガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスの機能をデプロイする手順を示します。
- HIPAA 対応サービスのリファレンス – HIPAA 対応サービスの一覧が提供されています。すべての AWS のサービス が HIPAA の対象となるわけではありません。
- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド – コンプライアンスの観点から責任共有モデルを理解します。このガイドは、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールを保護し、そのガイダンスに AWS のサービス マッピングするためのベストプラクティスをまとめたものです。
- 「 デベロッパーガイド」の「ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が内部プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に把握できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールの一覧については、Security Hub のコントロールリファレンスを参照してください。
- Amazon GuardDuty – 不審なアクティビティや悪意のあるアクティビティがないか環境をモニタリングすることで AWS アカウント、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。GuardDuty を使用すると、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件に対応できます。

- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Amazon Managed Grafana の耐障害性

AWS グローバルインフラストラクチャは、 AWS リージョンとアベイラビリティーゾーンを中心に構築されています。 AWS リージョンは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーンを提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョンとアベイラビリティーゾーンの詳細については、AWS 「 グローバルインフラストラクチャ」を参照してください。

Amazon Managed Grafana は、 AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズをサポートするのに役立ついくつかの機能を提供しています。

Amazon Managed Grafana のインフラストラクチャセキュリティ

マネージドサービスである Amazon Managed Grafana は、 AWS グローバルネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

AWS 公開された API コールを使用して、ネットワーク経由で Amazon Managed Grafana にアクセスします。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードはJava 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストにはアクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token ServiceAWS STSを使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

AWS CloudTrailを使用した Amazon Managed Grafana API コールのログ記録

Amazon Managed Grafana は、ユーザー、ロール、または AWS のサービスが実行したアクションの記録を提供するサービスである AWS CloudTrail と統合されています。Amazon Managed Grafana へのすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Amazon Managed Grafana コンソールからの呼び出しと、Amazon Managed Grafana API オペレーションへのコード呼び出しが含まれます。

Amazon Managed Grafana は Grafana API を使用する一部の呼び出しもキャプチャします。キャプチャされる呼び出しは、リソースを作成、更新、または削除する呼び出しなどのような、データを変更する呼び出しです。Amazon Managed Grafana でサポートされている Grafana API の詳細については、「Grafana HTTP API の使用」を参照してください。

CloudTrail で収集された情報を使用して、Amazon Managed Grafana に対するリクエスト、リクエスト元の IP アドレス、リクエストの作成日時、その他の詳細を確認できます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか。
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

CloudTrail は、アカウント AWS アカウント を作成すると でアクティブになり、CloudTrail イベント履歴に自動的にアクセスできます。CloudTrail の [イベント履歴] では、 AWS リージョンで過去 90 日間に記録された 管理イベントの表示、検索、およびダウンロードが可能で、変更不可能な記録を確認できます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。[イベント履歴] の閲覧には CloudTrail の料金はかかりません。

AWS アカウント 過去 90 日間のイベントの継続的な記録については、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。を使用して作成された証跡はすべてマルチリージョン AWS Management Console です。 AWS CLIを使用する際は、単一リージョンまたは複数リージョンの証跡を作成できます。 AWS リージョン アカウントのすべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成すると、進行中の管理イベントのコピーを 1 つ無料で CloudTrail から Amazon S3 バケットに配信できますが、Amazon S3 ストレージには料金がかかります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

[CloudTrail Lake] を使用すると、イベントに対して SQL ベースのクエリを実行できます。CloudTrail Lake は、行ベースの JSON 形式の既存のイベントを Apache ORC 形式に変換します。ORC は、データを高速に取得するために最適化された単票ストレージ形式です。イベントは、イベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基づいた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。CloudTrail Lake の詳細については、「 AWS CloudTrail ユーザーガイド」の「Working with AWS CloudTrail Lake」を参照してください。

CloudTrail Lake のイベントデータストアとクエリにはコストがかかります。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプションによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。

CloudTrail の Amazon Managed Grafana 管理イベント

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します AWS アカウント。これらのイベントは、コントロールプレーンオペレーションとも呼ばれます。CloudTrail は、デフォルトで管理イベントをログ記録します。

Amazon Managed Grafana では、すべての Amazon Managed Grafana コントロールプレーンオペレーションが管理イベントとしてログに記録されます。Amazon Managed Grafana が CloudTrail にログ記録する Amazon Managed Grafana コントロールプレーンオペレーションのリストについては、「Amazon Managed Grafana API リファレンス」を参照してください。

Amazon Managed Grafana イベント例

各イベントは任意の送信元からの単一のリクエストを表し、リクエストされた API オペレーション、オペレーションの日時、リクエストパラメータなどに関する情報を含みます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、イベントは特定の順序で表示されません。

次の例は、CreateWorkspace の CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "ANPAJ2UCCR6DPCEXAMPLE:sdbt-example",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/sdbt-example",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "ANPAJ2UCCR6DPCEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-11-26T20:59:21Z"
            }
        }
```

```
    },
    "eventTime": "2020-11-26T21:10:48Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "CreateWorkspace",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.179",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:82.0) Gecko/20100101
 Firefox/82.0",
    "requestParameters": {
        "permissionType": "Service Managed",
        "workspaceNotificationDestinations": [
            "SNS"
        ],
        "workspaceDescription": "",
        "clientToken": "12345678-abcd-1234-5678-111122223333",
        "workspaceDataSources": [
            "SITEWISE",
            "XRAY",
            "CLOUDWATCH",
            "ELASTICSEARCH",
            "PROMETHEUS",
            "TIMESTREAM"
        ],
        "accountAccessType": "CURRENT_ACCOUNT",
        "workspaceName": "CloudTrailTest",
        "workspaceRoleArn": "arn:aws:iam::123456789012:role/service-role/
AmazonGrafanaServiceRole-2705976ol"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
        "workspace": {
            "accountAccessType": "CURRENT_ACCOUNT",
            "created": 1606425045.22,
            "dataSources": [
                "SITEWISE",
                "XRAY",
                "CLOUDWATCH",
                "ELASTICSEARCH",
                "PROMETHEUS",
                "TIMESTREAM"
            ],
            "description": "",
            "grafanaVersion": "7.3.1",
```

```
            "id": "g-a187c473d3",
            "modified": 1606425045.22,
            "name": "CloudTrailTest",
            "notificationDestinations": [
                "SNS"
            ],
            "permissionType": "Service Managed",
            "status": "CREATING",
            "workspaceRoleArn": "arn:aws:iam::123456789012:role/service-role/
AmazonGrafanaServiceRole-2705976ol"
        }
    },
    "requestID": "12345678-5533-4e10-b486-e9c7b219f2fd",
    "eventID": "12345678-2710-4359-ad90-b902dbfb606b",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

次の例は UpdateWorkspaceAuthentication アクションの CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId":
 "AROAU2UJBF3NRO35YZ3GV:CODETEST_Series_GrafanaApiTestHydraCanary12-
o6aeXqaXS_1090259374",
        "arn": "arn:aws:sts::332073610971:assumed-role/
HydraInvocationRole-4912743f1277b7c3c67cb29518f8bc413ae/
CODETEST_Series_GrafanaApiTestHydraCanary12-o6aeXqaXS_1090259374",
        "accountId": "111122223333",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAU2UJBF3NRO35YZ3GV",
                "arn": "arn:aws:iam::111122223333:role/
HydraInvocationRole-4912743f1277b7c3c67cb29518f8bc413ae",
                "accountId": "332073610971",
                "userName": "TestInvocationRole-4912743f1277b7c3c67cb29518f8bc413ae"
```

```
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2021-08-04T20:50:24Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-08-04T21:29:25Z",
    "eventSource": "gamma-grafana.amazonaws.com",
    "eventName": "UpdateWorkspaceAuthentication",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "34.215.72.249",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.1030
 Linux/4.14.231-180.360.amzn2.x86_64 OpenJDK_64-Bit_Server_VM/11.0.11+9-LTS
 java/11.0.11 vendor/Amazon.com_Inc. cfg/retry-mode/legacy exec-env/AWS_Lambda_java11",
    "requestParameters": {
        "authenticationProviders": [
            "AWS_SSO",
            "SAML"
        ],
        "samlConfiguration": {
            "idpMetadata": {
                "url": "https://portal.sso.us-east-1.amazonaws.com/saml/metadata/
NjMwMDg2NDc4OTA3X2lucy1jY2E2ZGU3ZDlmYjdiM2Vh"
            }
        },
        "workspaceId": "g-84ea23c1b4"
    },
    "responseElements": {
        "authentication": {
            "awsSso": {
                "ssoClientId": "gAROcWGs9-LoqCMIQ56XyEXAMPLE"
            },
            "providers": [
                "AWS_SSO",
                "SAML"
            ],
            "saml": {
                "configuration": {
                    "idpMetadata": {
                        "url": "https://portal.sso.us-east-1.amazonaws.com/saml/
metadata/NjMwMDg2NDc4OTA3X2lucy1jY2E2ZGU3ZDlmYjdiM2Vh"
                    },
```

```
                    "loginValidityDuration": 60
                },
                "status": "CONFIGURED"
            }
        }
    },
    "requestID": "96adb1de-7fa5-487e-b6c6-6b0d4495cb71",
    "eventID": "406bc825-bc52-475c-9c91-4c0d8a07c1fa",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CloudTrail レコードの内容については、「AWS CloudTrail ユーザーガイド」の「CloudTrail record contents」を参照してください。

Grafana API イベント例

Amazon Managed Grafana は CloudTrail の一部の Grafana API コールもログに記録します。キャプチャされる呼び出しは、リソースを作成、更新、または削除する呼び出しなどのような、データを変更する呼び出しです。Amazon Managed Grafana でサポートされている Grafana API の詳細については、「Grafana HTTP API の使用」を参照してください。

AWS IAM Identity Centerを使用して Amazon Managed Grafana ワークスペースにサインインするユーザー

```
{
    "Records": [
        {
            "eventVersion": "1.08",
            "userIdentity": {
                "type": "SAMLUser",
                "userName": "johndoe"
            },
            "eventTime": "2021-07-09T02:31:59Z",
            "eventSource": "grafana.amazonaws.com",
            "eventName": "login-auth.sso",
            "awsRegion": "us-west-2",
            "sourceIPAddress": "192.0.2.0,198.51.100.0",
```

```
            "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36",
            "requestParameters": null,
            "responseElements": null,
            "eventID": "176bf326-0302-4190-8dbf-dfdf481d8198",
            "readOnly": false,
            "eventType": "AwsServiceEvent",
            "managementEvent": true,
            "eventCategory": "Management",
            "recipientAccountId": "111122223333",
            "serviceEventDetails": {
                "timestamp": "2021-07-09T02:31:59.045984031Z",
                "user": {
                    "userId": 1,
                    "orgId": 1,
                    "name": "johndoe",
                    "isAnonymous": false
                },
                "action": "login-auth.sso",
                "requestUri": "",
                "request": {
                    "query": {
                        "code": [

 "eyJraWQiOiJrZXktMTU2Njk2ODEyMSIsImFsZyI6IkhTMzg0In0.eyJwbGFpbnRleHQiOiJZUzEwYWtaWHpBZUowTDlQc
                        ],
                        "state": [

 "QUFBQURtdGxlUzB4TlRZNE9UVTFOekkyM2RUWUFUaHZHYXcyOU9ULUVaWHhNUXAwX184N25RVGVWMmd0enFpVE1iWlRPV
                        ]
                    }
                },
                "result": {
                    "statusType": "failure"
                },
                "ipAddress": "192.0.2.0,198.51.100.0",
                "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.114 Safari/537.36",
                "grafanaVersion": "7.5.7",
                "additionalData": {
                    "GiraffeCustomerAccount": "111122223333",
                    "GiraffeWorkspaceId": "g-123EXAMPLE",
                    "extUserInfo": "{\"OAuthToken\":null,\"AuthModule\":\"auth.sso\",
\"AuthId\":\"92670be4c1-e524608b-82f2-452d-a707-161c1e5f4706\",\"UserId\":0,\"Email\":
```

```
\"\",\"Login\":\"johndoe\",\"Name\":\"johndoe\",\"Groups\":null,\"OrgRoles\":{\"1\":
\"Admin\"},\"IsGrafanaAdmin\":false,\"IsDisabled\":false}"
                }
            }
        }
    ]
}
```

Grafana API POST /api/auth/keys

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:32Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "create",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.1",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "157bbf19-6ba4-4704-bc3b-d3e334b3a2b8",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:32.419795511Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "create",
        "resources": [
            {
```

```
                "ID": 0,
                "type": "api-key"
            }
        ],
        "requestUri": "",
        "request": {
            "body": "{\"name\":\"keyname\",\"role\":\"Admin\",\"secondsToLive\":60}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.1",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API DELETE /api/auth/keys/:id

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:33Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "delete",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.2",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "df1aafb3-28c6-4836-a64b-4d34538edc51",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
```

```
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:33.045041594Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "delete",
        "resources": [
            {
                "ID": 0,
                "type": "api-key"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":id": "24"
            }
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.2",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API POST /api/alerts/:id/pause

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:40Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "pause",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.3",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "d533a7ba-f193-45ac-a88c-75ed0594509b",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:40.261226856Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "pause",
        "resources": [
            {
                "ID": 0,
                "type": "alert"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":alertId": "1"
            },
            "body": "{\"paused\":true}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
```

```
        },
        "ipAddress": "192.0.2.0,198.51.100.3",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana POST /api/alerts/test

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:39Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "test",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,10.0.42.208",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "400",
    "errorMessage": "The dashboard needs to be saved at least once before you can test
 an alert rule",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "7094644d-8230-4774-a092-8a128eb6dec9",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:39.622607860Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
```

```
            "isAnonymous": false
        },
        "action": "test",
        "resources": [
            {
                "ID": 0,
                "type": "panel"
            }
        ],
        "requestUri": "",
        "request": {},
        "result": {
            "statusType": "failure",
            "statusCode": "400",
            "failureMessage": "The dashboard needs to be saved at least once before you
 test an alert rule"
        },
        "ipAddress": "192.0.2.0, 10.0.42.208",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API POST /api/alert-notifications

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:40Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "create",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.0",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
```

```
    "responseElements": null,
    "eventID": "1ce099b3-c427-4338-9f42-d38d1ef64efe",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:40.888295790Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "create",
        "resources": [
            {
                "ID": 0,
                "type": "alert-notification"
            }
        ],
        "requestUri": "",
        "request": {
            "body": "{\"name\":\"alert notification name\",\"type\":\"Slack\"}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.0",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```


Grafana API PUT /api/alert-notifications/uid/:uid

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:42Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "update",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.3",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cebfeb38-5007-495c-bd29-c8077797acac",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:42.792652648Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "update",
        "resources": [
            {
                "ID": 0,
                "type": "alert-notification"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":uid": "WvDWDSinz"
            },
```

```
            "body": "{\"name\":\"DIFFERENT alert notification name\",\"type\":\"AWS SNS
\"}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.3",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API POST /api/annotations

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:45Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "create",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.1",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "13bf3bef-966c-4913-a760-ade365a4a08f",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:45.394513179Z",
        "user": {
```

```
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "create",
        "resources": [
            {
                "ID": 0,
                "type": "annotation"
            }
        ],
        "requestUri": "",
        "request": {
            "body": "{\"dashboardId\":36,\"panelId\":2,\"tags\":[\"tag1\",\"tag2\"],
\"what\":\"Event Name\"}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.1",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API DELETE /api/dashboards/uid/:uid

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:17:09Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "delete",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.7",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "d6ad9134-5fbc-403c-a76d-4ed9a81065b6",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:17:09.200112003Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "delete",
        "resources": [
            {
                "ID": 0,
                "type": "dashboard"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":uid": "GLzWvIi7z"
            }
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.7",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
```

```
        }
    }
}
```

Grafana API PUT /api/datasources/:datasourceId

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:36Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "update",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,10.0.108.94",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "92877483-bdf6-44f5-803e-1ac8ad997113",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:36.918660585Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "update",
        "resources": [
            {
                "ID": 0,
                "type": "datasource"
            }
        ],
```

```
        "requestUri": "",
        "request": {
            "params": {
                ":id": "108"
            },
            "body": "{\"access\":\"proxy\",\"basicAuth\":false,\"name\":
\"test_amp_datasource_NEW_name\",\"type\":\"Amazon Managed Prometheus\",\"url\":
\"http://amp.amazonaws.com\"}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,10.0.108.94",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API DELETE /api/teams/:teamId/groups/:groupId

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:17:07Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "delete",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.2",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "b41d3967-daab-44d1-994a-a437556add82",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
```

```
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:17:07.296142539Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "delete",
        "resources": [
            {
                "ID": 0,
                "type": "team"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":groupId": "cn=editors,ou=groups,dc=grafana,dc=org",
                ":teamId": "35"
            }
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,198.51.100.2",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```


Grafana API PUT /api/folders/:uid


```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:16:56Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "update",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,198.51.100.1",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "412",
    "errorMessage": "the folder has been changed by someone else",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "414c98c8-aa53-45e4-940d-bea55716eaf6",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:16:56.382646826Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "update",
        "resources": [
            {
                "ID": 0,
                "type": "folder"
            }
        ],
        "requestUri": "",
        "request": {
            "params": {
                ":uid": "lnsZvSi7z"
            },
            "body": "{\"title\":\"NEW Folder Name\"}"
        },
```

```
        "result": {
            "statusType": "failure",
            "statusCode": "412",
            "failureMessage": "the folder has been changed by someone else"
        },
        "ipAddress": "192.0.2.0,198.51.100.1",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

Grafana API POST /api/teams

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "userName": "api_key"
    },
    "eventTime": "2021-07-09T02:17:02Z",
    "eventSource": "grafana.amazonaws.com",
    "eventName": "create",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0,10.0.40.206",
    "userAgent": "python-requests/2.24.0",
    "errorCode": "200",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "8d40bd79-76a8-490c-b7bb-74205253b707",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "timestamp": "2021-07-09T02:17:02.845022379Z",
        "user": {
            "orgId": 1,
            "orgRole": "Admin",
```

```
            "name": "api_key",
            "apiKeyId": "23",
            "isAnonymous": false
        },
        "action": "create",
        "resources": [
            {
                "ID": 0,
                "type": "team"
            }
        ],
        "requestUri": "",
        "request": {
            "body": "{\"name\":\"TeamName\"}"
        },
        "result": {
            "statusType": "success",
            "statusCode": "200"
        },
        "ipAddress": "192.0.2.0,10.0.40.206",
        "userAgent": "python-requests/2.24.0",
        "grafanaVersion": "7.5.7",
        "additionalData": {
            "GiraffeCustomerAccount": "111122223333",
            "GiraffeWorkspaceId": "g-123EXAMPLE"
        }
    }
}
```

セキュリティに関するベストプラクティス

このセクションのトピックでは、Amazon Managed Grafana デプロイのセキュリティを最も適切に維持するために従うべきベストプラクティスについて説明します。

有効期間の短い API キーを使用する

Amazon Managed Grafana ワークスペースで Grafana API を使用するには、まず認証に使用する API キーを作成する必要があります。キーを作成する際にキーの [有効期限] を指定しますが、これはキーが有効である期間 (最大 30 日間) を定義します。短時間 (数時間以下など) に設定することを強くお勧めします。これにより、長期間有効な API キーを持つ場合と比較してリスクが大幅に軽減されます。

また、API キーのセキュリティ保護の観点から、API キーはパスワードと同じように扱うことをお勧めします。例えば、プレーンテキストで保存しないでください。

自己管理型 Grafana からの移行

このセクションは、既存の自己管理型の Grafana または Grafana Enterprise デプロイを Amazon Managed Grafana に移行する場合に関連します。これは、オンプレミスの Grafana と AWS、独自のアカウントの 上の Grafana デプロイの両方に適用されます。

オンプレミスまたは独自の AWS アカウントで Grafana を実行している場合は、アクセスを管理するためにユーザーとチーム、場合によっては組織ロールを定義している可能性があります。Amazon Managed Grafana では、ユーザーとグループは Amazon Managed Grafana の外部から管理されるか、IAM Identity Center を使用または SAML 2.0 統合を介して ID プロバイダー (IdP) から直接管理されます。Amazon Managed Grafana では、ダッシュボードの閲覧権限など、タスクの実行に必要な特定の権限を割り当てることができます。Amazon Managed Grafana でのユーザー管理の詳細については、「Amazon Managed Grafana でワークスペース、ユーザー、ポリシーを管理する」を参照してください。

さらに、オンプレミスの Grafana を実行する場合、データソースにアクセスするために、長期間有効なキーまたはシークレット認証情報を使用しています。Amazon Managed Grafana に移行する際は、これらの IAM ユーザーを IAM ロールに置き換えることを強くお勧めします。例については、「CloudWatch をデータソースとして手動で追加する」を参照してください。

インターフェイス VPC エンドポイント

Amazon VPC と Amazon Managed Grafana の間で AWS PrivateLink サポートを提供しています。Amazon VPC エンドポイントの IAM リソースポリシーを添付することで、仮想プライベートクラウド (VPC) のエンドポイントから Amazon Managed Grafana サービスへのアクセスを制御できます。

Amazon Managed Grafana は、2 つの異なる種類の VPC エンドポイントをサポートしています。Amazon Managed Grafana サービスに接続し、Amazon Managed Grafana API にアクセスしてワークスペースを管理できます。または、特定のワークスペースに VPC エンドポイントを作成できます。

インターフェイス Amazon VPC エンドポイントを使った Amazon Managed Grafana の使用

Amazon Managed Grafana でインターフェイス VPC エンドポイントを使用する方法は 2 つあります。VPC エンドポイントを使用して、Amazon EC2 インスタンスなどの AWS リソースが Amazon Managed Grafana API にアクセスしてリソースを管理できるようにするか、Amazon Managed Grafana ワークスペースへのネットワークアクセスの制限の一部として VPC エンドポイントを使用できます。

- Amazon VPC を使用して AWS リソースをホストしている場合は、`com.amazonaws.region.grafana`サービス名エンドポイントを使用して VPC と Amazon Managed Grafana API の間にプライベート接続を確立できます。
- ネットワークアクセスコントロールを使用して Amazon Managed Grafana ワークスペースにセキュリティを追加しようとする場合は、`com.amazonaws.region.grafana-workspace`サービス名エンドポイントを使用して、VPC と Grafana ワークスペースエンドポイントとの間にプライベート接続を確立できます。

Amazon VPC AWS のサービス は、定義した仮想ネットワークで AWS リソースを起動するために使用できる です。VPC を使用すると、IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどのネットワーク設定を制御できます。VPC をお使いの Amazon Managed Grafana API に接続するには、インターフェイス VPC エンドポイントを定義します。このエンドポイントは、インターネットゲートウェイ、ネットワークアドレス変換 (NAT) インスタンス、または VPN 接続を必要とすることなく、Amazon Managed Grafana へのスケーラブルで信頼性の高い接続を提供します。詳細については、「Amazon VPC ユーザーガイド」の「Amazon VPC とは」を参照してください。

インターフェイス VPC エンドポイントは AWS PrivateLink、プライベート IP アドレスを持つ Elastic Network Interface AWS のサービス を使用して 間のプライベート通信を可能にする AWS テクノロジーである を利用しています。詳細については、「New – AWS PrivateLink for AWS Services」を参照してください。

Amazon VPC での開始方法の詳細については、「Amazon VPC ユーザーガイド」の「開始方法」を参照してください。

Amazon Managed Grafana に AWS PrivateLink 接続するための VPC エンドポイントの作成

次のいずれかのサービス名エンドポイントを使用して、Amazon Managed Grafana へのインターフェイス VPC エンドポイントを作成します。

- Amazon Managed Grafana API に接続してワークスペースを管理するには、以下を選択します。

  `com.amazonaws.region.grafana`.
- Amazon Managed Grafana ワークスペースに接続するには (Grafana API を使用するなど)、以下を選択します。

  `com.amazonaws.region.grafana-workspace`

インターフェイス VPC エンドポイントの作成に関する詳細については、「Amazon VPC ユーザーガイド」の「インターフェイスエンドポイントの作成」を参照してください。

Grafana API 呼び出すには、Amazon VPC ユーザーガイド の手順に従って、VPC エンドポイントのプライベート DNS も有効にする必要があります。これにより、`*.grafana-workspace.region.amazonaws.com` フォーム内の URL のローカル解決が可能になります。

ネットワークアクセスコントロールを使用して Grafana ワークスペースへのアクセスを制限する

特定の Grafana ワークスペースへのアクセスに使用できる IP アドレスまたは VPC エンドポイントを制限する場合は、そのワークスペースへのネットワークアクセスコントロールを設定できます。

ワークスペースへのアクセスを許可する VPC エンドポイントに対してエンドポイントのセキュリティグループを設定することで、より詳細なアクセス制限が可能になります。詳細については、「Amazon VPC ドキュメント」の「セキュリティグループの関連付け」と「セキュリティグループのルール」を参照してください。

エンドポイントポリシーを使用して Amazon Managed Grafana API VPC エンドポイントへのアクセスを制御する

(`com.amazonaws.region.grafana` を使用して) Amazon Managed Grafana API に接続されている VPC エンドポイントの場合、VPC エンドポイントポリシーを追加してサービスへのアクセスを制限できます。

> **Note**
>
> (com.amazonaws.*region*.grafana-workspace を使用して) ワークスペースに接続されている VPC エンドポイントは、VPC エンドポイントポリシーをサポートしていません。

VPC エンドポイントポリシーは、エンドポイントの作成時または変更時にエンドポイントに加える IAM リソースポリシーです。エンドポイントの作成時にポリシーをアタッチしない場合、サービスへのフルアクセスを許可するデフォルトのポリシーが Amazon VPC によって自動的にアタッチされます。エンドポイントポリシーは、IAM アイデンティティベースのポリシーやサービス固有のポリシーを上書きしたり置き換えたりするものではありません。これは、評価項目から指定されたサービスへのアクセスを制御するための別のポリシーです。

エンドポイントのポリシーは、JSON 形式で記述する必要があります。

詳細については、「Amazon VPC ユーザーガイド」の「VPC エンドポイントによるサービスへのアクセスのコントロール」を参照してください。

Amazon Managed Grafana のエンドポイントポリシーの例を次に示します。このポリシーにより、VPC 経由で Amazon Managed Grafana に接続するユーザーは、Amazon Managed Grafana サービスにデータを送信できます。また、他の Amazon Managed Grafana アクションを実行できなくなります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSGrafanaPermissions",
            "Effect": "Allow",
            "Action": [
                "grafana:DescribeWorkspace",
                "grafana:UpdatePermissions",
                "grafana:ListPermissions",
                "grafana:ListWorkspaces"
            ],
            "Resource": "arn:aws:grafana:*:*:/workspaces*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:root"
                ]
```

```
            }
        }
    ]
}
```

Grafana の VPC エンドポイントポリシーを編集するには

1. [VPC console] で Amazon VPC コンソールを開きます。
2. ナビゲーションペインで、[エンドポイント] を選択します。
3. エンドポイントをまだ作成していない場合は、[Create Endpoin] (エンドポイントの作成) を選択します。
4. `com.amazonaws.region.grafana` エンドポイントを選択し、[Policy] (ポリシー) タブを選択します。
5. [ポリシーの編集] を選択し、変更を加えます。

Amazon Managed Grafana のサービスクォータ

Amazon Managed Grafana には、以下のクォータがあります。ワークスペース数のクォータ (上限数) の引き上げをリクエストできます。

名前	デフォルト	引き上げ可能	説明
ワークスペースの数	サポートされている各リージョン：5	あり	このアカウントで現在のリージョンにおいて保持できるワークスペースの最大数。
AssociateLicense リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの AssociateLicense リクエストの最大数。
CreateWorkspace リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの CreateWorkspace リクエストの最大数。
DeleteWorkspace リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの DeleteWorkspace リクエストの最大数。

名前	デフォルト	引き上げ可能	説明
DescribeWorkspace リクエストのレート	サポートされている各リージョン: 5/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの DescribeWorkspace リクエストの最大数。
DescribeWorkspaceAuthentication リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの DescribeWorkspaceAuthentication リクエストの最大数。
DisassociateLicense リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの DisassociateLicense リクエストの最大数。
ListPermissions リクエストのレート	サポートされている各リージョン : 10/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの ListPermissions リクエストの最大数。
ListWorkspaces リクエストのレート	サポートされている各リージョン: 5/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの ListWorkspaces リクエストの最大数。

名前	デフォルト	引き上げ可能	説明
UpdatePermissions リクエストのレート	サポートされている各リージョン：10/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの UpdatePermissions リクエストの最大数。
UpdateWorkspace リクエストのレート	サポートされている各リージョン：10/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの UpdateWorkspace リクエストの最大数。
UpdateWorkspaceAuthentication リクエストのレート	サポートされている各リージョン: 1/秒	いいえ	このアカウントで現在のリージョンにおいて実行できる 1 秒あたりの UpdateWorkspaceAuthentication リクエストの最大数。

さらに、Amazon Managed Grafana の各ワークスペースには以下のクォータがあります。

リソース	調整可能	デフォルトのクォータ
アラート 従来のアラートに設定できるワークスペースあたりのルールの数、または Grafana アラートに設定できるワークスペースあ	いいえ	ワークスペースあたり 100。

リソース	調整可能	デフォルトのクォータ
たりのルールインスタンスの数。		
ダッシュボード	いいえ	ワークスペースあたり 2,000。
データソース	いいえ	ワークスペースあたり 2,000。
[ユーザー]	いいえ	10,000 がプロビジョニングされ、ワークスペースあたり 500 が同時実行されました。
API キー	いいえ	ワークスペースあたり 100。
サービスアカウント	いいえ	ワークスペースあたり 100。
サービスアカウントトークン アクティブなトークンと期限切れトークンは、このクォータにカウントされます。トークンを削除してクォータから削除します。	いいえ	ワークスペースあたり 100。
ネットワークアクセスコントロール: プレフィックスリスト	いいえ	ワークスペースあたり 5。
ネットワークアクセスコントロール: IP アドレスの範囲	いいえ	プレフィックスリストあたり 100。

リソース	調整可能	デフォルトのクォータ
ネットワークアクセスコントロール: VPC エンドポイント	いいえ	ワークスペースあたり 5。

ユーザーガイドのドキュメント履歴

次の表に、Amazon Managed Grafana の前回のリリース以後に行われたドキュメントの重要な変更を示します。このドキュメントの更新に関する通知を受け取るには、RSS フィードにサブスクライブできます。

変更	説明	日付
Amazon Managed Grafana の Amazon EKS クラスターのモニタリングソリューションを更新	Amazon Managed Grafana は、 を使用してダッシュボードで Amazon EKS クラスターを自動的にモニタリングするようにソリューションを更新 AWS CDKし、デプロイを簡素化します。	2024 年 6 月 10 日
Amazon Managed Grafana に Grafana バージョン 10 のサポートを追加	Amazon Managed Grafana に Grafana バージョン 10 のサポートが追加されました。Grafana バージョン 8 と 9 も引き続き使用できます。	2024 年 5 月 15 日
Amazon Managed Grafana に Amazon EKS クラスターのモニタリングソリューションを追加	Amazon Managed Grafana でのソリューション追加により、 AWS CloudFormationを介して作成されたダッシュボードを使用して Amazon EKS クラスターを自動的にモニタリングできるようになります。	2024 年 4 月 30 日
Amazon Managed Grafana の古い管理ポリシーを置換	Amazon Managed Grafana に、古い「AWSGrafanaWorkspacePermissionManagement」マネージドポリシーを置き換える新しいマネージドポリシー	2024 年 1 月 5 日

	「AWSGrafanaWorkspacePermissionManagementV2」が追加されます。この新しい管理ポリシーは、より制限の厳しいアクセス許可セットを提供し、ワークスペースのセキュリティが向上されます。	
Amazon Managed Grafana にコミュニティプラグインの使用のサポートを追加	Amazon Managed Grafana に、Grafana バージョン 9 と互換性のあるワークスペースでコミュニティプラグインを使用するためのサポートが追加されます。	2023 年 11 月 15 日
Amazon Managed Grafana に、既存のワークスペースのバージョンを更新するためのサポートを追加	Amazon Managed Grafana に、既存のワークスペースを最新バージョンに更新するためのサポートが追加されます。例えば、Grafana バージョン 8 と互換性のあるワークスペースを更新して、バージョン 9 と互換性を持たせることができます。	2023 年 7 月 19 日
Amazon Managed Grafana に OpenSearch データソースでのトレース分析のサポートを追加	Amazon Managed Grafana に、OpenSearch データソースのトレース一覧を表示する機能が追加されます (バージョン 9 以降をサポートするワークスペースを使用する場合)。	2023 年 6 月 22 日
Amazon Managed Grafana に Grafana バージョン 9 のサポートを追加	Amazon Managed Grafana に Grafana バージョン 9 のサポートが追加されました。Grafana バージョン 8 も引き続き使用できます。	2023 年 4 月 28 日

Amazon Managed Grafana に新しいマネージドポリシーを追加	Amazon Managed Grafana に、Amazon Managed Grafana が CloudWatch のメトリクスにアクセスできるようにする新しいマネージドポリシー「AmazonGrafanaCloudWatchAccess」が追加されました。	2023 年 3 月 24 日
Amazon Managed Grafana に新しい権限を追加	Amazon Managed Grafana の AWSGrafanaWorkspacePermissionManagement に、Microsoft Active Directory または Active Directory Connector で管理されている IAM Identity Center のユーザーとグループを Amazon Managed Grafana ワークスペースに関連付け、または解除できるようにする新しい権限が追加されました。	2023 年 3 月 22 日
アラートインスタンスを明確化	Grafana アラートでは、1 つのアラートルールで複数のアラートインスタンスを作成でき、これはすぐにクォータ (上限) に達してしまう要因となります。ドキュメントの更新に伴い、詳細が追加されています。	2023 年 3 月 20 日
Amazon Managed Grafana に新しいデータソースを追加	Amazon Managed Grafana に、Databricks、Google BigQuery 用の新しいデータソースプラグインが追加されます。	2023 年 3 月 14 日

Amazon Managed Grafana にワークスペースへのネットワークアクセスコントロールを追加	Amazon Managed Grafana に、指定された IP アドレスまたは VPC エンドポイントのみがワークスペースにアクセスできるようにするネットワークアクセスコントロールが追加されます。	2023 年 2 月 16 日
Amazon Managed Grafana に新しい権限を追加	Amazon Managed Grafana の AWSGrafanaWorkspacePermissionManagement に、IAM Identity Center のユーザーとグループを Grafana ワークスペースに関連付けできるようにする新しい権限が追加されました。	2022 年 12 月 20 日
Grafana サービス向けの新しい SLR ポリシー	`AmazonGrafanaServiceLinkedRolePolicy` AWS 管理ポリシーからアクセス許可を受け取るサービスにリンクされたロールAmazonManagedGrafana を追加しました。	2022 年 11 月 23 日
Amazon VPC データソースとの接続	Amazon VPC のデータソースへの接続が追加されました。	2022 年 11 月 23 日
Amazon Managed Grafana にワークスペース設定が追加されました	Amazon Managed Grafana に、設定変更を Grafana ワークスペースインスタンスごとに行えるサポートが追加されました	2022 年 11 月 23 日

Amazon Managed Grafana に Grafana アラートのサポートを追加	Amazon Managed Grafana に、Amazon Managed Service for Prometheus インスタンスと Prometheus インスタンスからのアラートの統合など、最新の Grafana アラート機能を使用するためのサポートが追加されました。	2022 年 11 月 23 日
Amazon Managed Grafana に 3 つの新しい視覚的表現を追加	Amazon Managed Grafana に、Plotly、Sankey、Scatter パネルの視覚的表現が追加されます。	2022 年 11 月 17 日
AWS SSO の IAM アイデンティティセンターへのブランド変更	AWS SSO は IAM Identity Center にブランド変更されました。	2022 年 7 月 26 日
Amazon Managed Grafana の機能強化	Amazon Managed Grafana に、Grafana バージョン 8.4、Pixie、GitHub、および Moogsoft データソース、Recorded クエリ、WindRose パネルの視覚的表現のサポートが追加されます。	2022 年 5 月 13 日
Amazon Managed Grafana の機能強化	Amazon Managed Grafana にタグ付けのサポートが追加されます。	2022 年 3 月 31 日
Amazon Managed Grafana の機能強化	Amazon Managed Grafana に VPC Privatelink のサポートが追加されました。	2022 年 1 月 7 日

Amazon Managed Grafana の機能強化	Amazon Managed Grafana に、Amazon Redshift データソース、Amazon Athena データソース、Zabbix、Cloudflare のサポートが追加されます。Amazon Managed Grafana では、Geomap パネルの視覚的表現と Grafana バージョン 8.2 をサポートしています。	2021 年 11 月 24 日
Amazon Managed Grafana のプレビュー機能の強化	Amazon Managed Grafana のプレビューでは、Grafana バージョン 7.5 がサポートされ、 AWS Marketplace 統合による Grafana Enterprise へのアップグレードがサポートされています。Amazon Elasticsearch Service データソースも、Open Distro for Elasticsearch をサポートするためにアップグレードされました。	2021 年 4 月 16 日
Amazon Managed Grafana プレビューがリリースされました。	Amazon Managed Grafana のプレビューがリリースされました。	2020 年 12 月 15 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。