



ユーザーガイド

# Amazon Managed Grafana



# Amazon Managed Grafana: ユーザーガイド

Copyright © 2024 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
サポートされるリージョン .....	1
開始 .....	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 .....	17
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
更新されたワークスペースに関する問題のトラブルシューティング .....	42
エンタープライズプラグイン .....	43
Enterprise プラグインへのアクセスの管理 .....	45
Grafana Labs へのリンク .....	46
FAQ AWS Marketplace エンタープライズユーザー向け .....	47
ワークスペース間でのコンテンツの移行 .....	50
Workspace ユーザーアクセス .....	51
ユーザーまたはグループにアクセス許可を付与する .....	51
アクセス許可の不一致エラー .....	54
アクセス許可の不一致に関するよくある質問 .....	54
データソースと通知のアクセス許可 .....	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 .....	57
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コントロールの設定 .....	59
Amazon VPC のデータに接続する .....	62
VPC 接続の仕組み .....	63
VPC への接続を作成する .....	64
VPC のトラブルシューティング .....	66
ワークスペースを設定する .....	70
API または の設定 AWS CLI .....	72
ワークスペースの削除 .....	74
Grafana ワークスペースを使用する .....	75
Grafana とは .....	76
メトリクスとログを調べる .....	76
アラート .....	76
注釈 .....	76
ダッシュボード変数 .....	77
ワークスペースに接続する .....	77
ユーザー、チーム、アクセス許可 .....	78
[ユーザー] .....	79
ユーザーロール .....	79
チームの管理 .....	80
許可の使用 .....	82
最初のダッシュボード .....	87
ダッシュボードの作成 .....	87
Grafana プラグイン .....	95
プラグインカタログ .....	96
プラグインの管理 .....	98
プラグインのインストールまたは削除 .....	99
プラグインを更新する .....	99
AWS データソースプラグイン .....	100
データソース .....	102
特殊なデータソース .....	103
の使用 AWS Organizations .....	104
組み込みデータソース .....	106
エンタープライズデータソース .....	290
Grafana バージョン 10 の使用 .....	373
ダッシュボード .....	374
パネルとビジュアライゼーション .....	460



探索 .....	661
相関関係 .....	688
アラート .....	694
Grafana バージョン 9 の使用 .....	817
ダッシュボード .....	818
パネルとビジュアライゼーション .....	885
探索 .....	1031
アラート .....	1047
Grafana バージョン 8 の使用 .....	1147
パネル .....	1148
ダッシュボード .....	1242
探索 .....	1270
Linking (リンク中) .....	1279
テンプレートと変数 .....	1288
Grafana アラート .....	1313
設定を変更する .....	1362
Amazon Managed Grafana プロファイルを編集する .....	1363
設定を編集する .....	1363
Amazon Managed Grafana セッションを表示する .....	1364
サポートバンドル .....	1364
サポートバンドルコンポーネント .....	1365
サポートバンドルの作成 .....	1365
Classic アラート .....	1366
アラート設定 .....	1367
クラスタリング .....	1367
通知 .....	1367
アラート実行 .....	1367
アラート通知 .....	1368
アラートの作成 .....	1373
アラートルールの一時停止 .....	1378
既存のアラートルールの表示 .....	1379
通知テンプレート .....	1380
アラートのトラブルシューティング .....	1381
Grafana API リファレンス .....	1382
トークンによる認証 .....	1383
サービスアカウント .....	1383

API キー .....	1387
アラート API .....	1391
アラートの取得 .....	1391
ID でアラートを受け取る .....	1393
ID によるアラートの一時停止 .....	1394
アラート通知チャンネル API .....	1394
すべての通知チャンネルを取得する .....	1395
すべての通知チャンネルを取得する (ルックアップ) .....	1396
UID ですべての通知チャンネルを取得する .....	1397
ID ですべての通知チャンネルを取得する .....	1398
通知チャンネルの作成 .....	1399
UID で通知チャンネルを更新する .....	1400
ID で通知チャンネルを更新する .....	1401
UID で通知チャンネルを削除する .....	1403
ID で通知チャンネルを削除する .....	1403
テスト通知チャンネル .....	1404
注釈 API .....	1405
注釈の検索 .....	1405
注釈の作成 .....	1407
グラフィック形式で注釈を作成する .....	1408
注釈の更新 .....	1409
パッチアノテーション .....	1409
ID で注釈を削除する .....	1410
認証 API .....	1411
API キーの取得 .....	1411
API キーの作成 .....	1412
API キーを削除する .....	1412
ダッシュボード API .....	1413
ダッシュボードの作成/更新 .....	1414
uid でダッシュボードを取得する .....	1419
uid でダッシュボードを削除する .....	1421
ホームダッシュボードを取得します。 .....	1421
ダッシュボードタグの取得 .....	1423
ダッシュボードアクセス許可 API .....	1423
ダッシュボードのアクセス許可を取得する .....	1424
ダッシュボードのアクセス許可を更新する .....	1426

ダッシュボードバージョン API .....	1427
すべてのダッシュボードバージョンを取得する .....	1427
ダッシュボードのバージョンを取得する .....	1429
ダッシュボードの復元 .....	1431
ダッシュボードのバージョンを比較する .....	1433
データソース API .....	1435
すべてのデータソースを取得する .....	1435
ID で単一のデータソースを取得する .....	1436
UID で単一のデータソースを取得する .....	1437
名前で単一のデータソースを取得する .....	1438
データソース ID を名前を取得する .....	1439
データソースを作成します。 .....	1439
既存のデータソースを更新する .....	1443
ID でデータソースを削除する .....	1444
UID でデータソースを削除する .....	1445
名前でデータソースを削除する .....	1445
データソースプロキシ呼び出し .....	1446
ID によるデータソースのクエリ .....	1446
データソースアクセス許可 API .....	1449
データソースのアクセス許可を有効にする .....	1449
データソースのアクセス許可を無効にする .....	1450
データソースのアクセス許可を取得する .....	1451
データソースのアクセス許可を追加する .....	1452
データソースのアクセス許可を削除する .....	1454
外部グループ同期 API .....	1454
外部グループを取得する .....	1455
外部グループを追加する .....	1455
外部グループを削除する .....	1456
フォルダ API .....	1457
フォルダの作成 .....	1458
フォルダの更新 .....	1459
すべてのフォルダを取得する .....	1461
uid でフォルダを取得する .....	1461
ID でフォルダを取得する .....	1461
uid でフォルダを削除する .....	1463
フォルダ/ダッシュボード検索 API .....	1464

フォルダとダッシュボードの検索 .....	1465
フォルダアクセス許可 API .....	1467
フォルダのアクセス許可を取得する .....	1467
フォルダのアクセス許可を更新する .....	1469
Organization API .....	1470
現在の組織を取得する .....	1471
現在の組織内のすべてのユーザーを取得する .....	1471
現在の組織内のすべてのユーザーを取得する (検索) .....	1472
指定されたユーザーを更新します。 .....	1473
現在の組織のユーザーを削除します .....	1473
現在の組織を更新する .....	1474
現在の組織にユーザーを追加する .....	1474
プレイリスト API .....	1475
プレイリストの検索 .....	1475
プレイリストを 1 つ取得する .....	1476
プレイリスト項目を取得する .....	1477
プレイリストダッシュボードを取得する .....	1478
プレイリストを作成する .....	1478
プレイリストを更新する .....	1479
プレイリストを削除する .....	1481
プラグイン API .....	1481
プラグインをインストールする .....	1481
プラグインのアンインストール .....	1482
すべてのプラグインを取得する .....	1482
プラグインを取得する .....	1484
プラグインバージョンを取得する .....	1487
設定 API .....	1490
現在のユーザー設定を取得する .....	1490
現在のユーザー設定を更新する .....	1491
現在の組織設定を取得する .....	1491
現在の組織設定を更新する .....	1492
スナップショット API .....	1492
新しいスナップショットを作成する .....	1493
スナップショットのリストを取得する .....	1495
キーによるスナップショットの取得 .....	1495
キーによるスナップショットの削除 .....	1497

deleteKey によるスナップショットの削除 .....	1497
チーム API .....	1498
ページ分割によるチーム検索 .....	1498
ID でチームを取得する .....	1499
チームを追加する .....	1500
チームの更新 .....	1501
ID でチームを削除する .....	1502
チームメンバーを取得する .....	1502
チームメンバーを追加する .....	1503
チームからメンバーを削除する .....	1504
チーム設定を取得する .....	1505
チームの設定を更新する .....	1505
ユーザー API .....	1506
ユーザーがメンバーであるチームを取得する .....	1507
スナップショットのリストを取得する .....	1507
ダッシュボードのスターリングを解除する .....	1508
実際のユーザーの認証トークンを取得する .....	1508
実際のユーザーの認証トークンを取り消す .....	1509
オブザーバビリティソリューション: Amazon のモニタリング EKS .....	1511
このソリューションについて .....	1511
コスト .....	1515
前提条件 .....	1517
このソリューションの使用 .....	1519
を使用したインストール AWS CDK .....	1520
Terraform を使用したインストール .....	1522
追跡されるメトリクスのリスト .....	1524
作成されたアラートのリスト .....	1532
トラブルシューティング .....	1541
タグ付け .....	1545
ワークスペースのタグ付け .....	1546
セキュリティ .....	1552
データ保護 .....	1553
Amazon Managed Grafana でのデータ保護 .....	1554
Identity and Access Management .....	1554
対象者 .....	1554
アイデンティティを使用した認証 .....	1555

ポリシーを使用したアクセスの管理 .....	1559
Amazon Managed Grafana と の連携方法 IAM .....	1561
アイデンティティベースポリシーの例 .....	1568
AWS マネージドポリシー .....	1573
トラブルシューティング .....	1589
サービス間の混乱した代理の防止 .....	1591
サービスリンクロールの使用 .....	1592
他の サービスのアクセス許可とポリシー AWS .....	1597
1 つのアカウントのサービス管理アクセス許可 .....	1597
組織のサービス管理アクセス許可 .....	1600
カスタマー管理アクセス許可 .....	1606
IAM アクセス許可 .....	1607
Amazon Managed Grafana のアクセス許可 .....	1607
コンプライアンス検証 .....	1608
耐障害性 .....	1610
インフラストラクチャセキュリティ .....	1610
CloudTrail ログ .....	1611
での Amazon Managed Grafana 管理イベント CloudTrail .....	1613
Amazon Managed Grafana イベントの例 .....	1613
Grafana API イベントの例 .....	1617
セキュリティに関するベストプラクティス .....	1635
有効期間の短い API キーを使用する .....	1635
セルフマネージド Grafana からの移行 .....	1636
インターフェイス VPC エンドポイント .....	1636
インターフェイス VPC エンドポイントでの Amazon Managed Grafana の使用 .....	1637
Amazon Managed Grafana AWS PrivateLink に接続するための VPC エンドポイントの作成 .....	1638
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 .....	1638
エンドポイントポリシーを使用した Amazon Managed Grafana API VPC エンドポイントへのアクセスの制御 .....	1639
Service Quotas .....	1641
ドキュメント履歴 .....	1645
.....	mdcli

# 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よく使用される広くデプロイされているデータ可視化ツールである Grafana を簡単にデプロイ、運用、スケーリングできます。

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 ダッシュボードとビジュアライゼーションを作成して、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、またはデプロイすることなく、メトリクス、ログ、トレースを分析できます。

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、スケーリング、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。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、企業ガバナンス要件に準拠するための組み込みセキュリティ機能も用意されています。

Amazon Managed Grafana は、Amazon、Amazon OpenSearch Service CloudWatch、Amazon Timestream AWS X-Ray AWS IoT SiteWise、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。

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

Amazon Managed Grafana は、ワークスペース内のアクティブなユーザーごとに料金が設定されます。料金の詳細については、[「Amazon Managed Grafana の料金」](#)を参照してください。

## サポートされるリージョン

Amazon Managed Grafana は現在、次のリージョンをサポートしています。

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	HTTPS	
米国東部 (バージニア北部)	us-east-1	grafana.us-east-1.amazonaws.com	HTTPS	
米国西部 (オレゴン)	us-west-2	grafana.us-west-2.amazonaws.com	HTTPS	
アジアパシフィック (ソウル)	ap-northeast-2	grafana.ap-northeast-2.amazonaws.com	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	HTTPS	
アジアパシフィック (シドニー)	ap-southeast-2	grafana.ap-southeast-2.amazonaws.com	HTTPS	
アジアパシフィック (東京)	ap-northeast-1	grafana.ap-northeast-1.amazonaws.com	HTTPS	
欧州 (フランクフルト)	eu-central-1	grafana.eu-central-1.amazonaws.com	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	HTTPS	
欧州 (ロンドン)	eu-west-2	grafana.eu-west-2.amazonaws.com	HTTPS	

# Amazon Managed Grafana リソースを作成して使用する 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、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的な 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

をお持ちでない場合は 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 ようにセットアップする](#)

## ユーザー認証

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

- ID プロバイダー (IdPs) に保存され、Security Assertion Markup Language 2.0 (SAML 2.0) で認証されたユーザー認証情報
- AWS IAM Identity Center

## SAML

SAML を使用する場合、ユーザーは 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、まだ使用していない場合、アカウントで IAM Identity Center を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アクティブ化します。Amazon Managed Grafana が Organizations を有効にすると、組織も自動的に作成されます。Organizations の詳細については、[「とは AWS Organizations」](#)を参照してください。

### Note

既に組織のメンバー AWS 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。2019 年 11 月 25 日より前に管理アカウントで IAM Identity Center 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。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

承認に 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。

認証 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、少なくとも以下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、[「IAM Identity Center を使用して、単一の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します。」](#)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、次のステップに従います。

Amazon Managed Grafana でワークスペースを作成するには

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

必要に応じて、ワークスペースの説明を入力します。

4. [次へ] をクリックします。
5. 認証アクセス で、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、アカウントで IAM Identity Center をまだ有効にしていない場合は、最初の IAM Identity Center 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に関するユーザー管理を処理します。

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

- a. [ユーザーの作成] を選択します。
- b. ユーザーの E メールアドレス、名、姓を入力し、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 E メールアドレスを使用します。IAM Identity Center のこのアカウントのパスワードを作成するよう求める E メールメッセージが表示されます。

### Important

作成したユーザーは、Amazon Managed Grafana ワークスペースに自動的にアクセスできません。後のステップで、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をユーザーに許可します。

- SAML — SAML を選択した場合は、ワークスペースの作成後に SAML 設定を完了します。
6. [次へ] をクリックします。
  7. この最初のワークスペースでは、サービスマネージャがアクセス許可タイプに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を実行するデータソースを選択します。この入門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、リストされている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、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。これにより、これらのサービスが Grafana ワークスペースのデータソースとして完全に設定されるわけではありません。これは Grafana ワークスペースコンソールで実行できます。

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) 通知チャン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、 で始まる TopicName 値を使用して、アカウントの Amazon SNS トピックに発行する IAM ポリシーを作成できます grafana。これにより、ワークスペースの通知チャンネルとして Amazon SNS が完全に設定されるわけではありません。これは、ワークスペースの Grafana コンソールで実行できます。
11. [次へ] をクリックします。
12. ワークスペースの詳細を確認し、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。

初期状態では、ステータスは CREATING です。

#### Important

ステータスが ACTIVE になるまで待ってから、次のいずれかを実行します。

- SAML を使用している場合は、SAML セットアップを完了します。
- IAM Identity Center を使用している場合は、IAM Identity Center ユーザーにワークスペースへのアクセスを割り当てます。

現在のステータスを確認するには、ブラウザを更新する必要がある場合があります。

13. IAM Identity Center を使用している場合は、次の操作を行います。

- a. 認証 タブで、新しいユーザーまたはグループの割り当て を選択します。
- b. ワークスペースへのアクセス権を付与するユーザーの横にあるチェックボックスをオンにし、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オンにし、アクションドロップダウンリストから管理者アクションの作成を選択します。

**⚠ Important**

Grafana ワークスペース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Adminごとに少なくとも 1 人のユーザーを として割り当てます。

14. SAML を使用している場合は、次の操作を行います。

- a. 認証 タブの Security Assertion Markup Language (SAML) で、セットアップの完了 を選択します。
- b. インポートメソッド で、次のいずれかを実行します。
  - URL を選択し、IdP メタデータの URL を入力します。
  - を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、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している場合は、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 には、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 には、Amazon Managed Grafana ワークスペースでロールをすべて付与する必要がある IdP のユーザーAdminロールを入力するか、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かを選択します。

**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ことを選択した場合、データソース、ユーザー、ダッシュボードの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。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- e. (オプション) 追加の SAML 設定を入力するには、追加設定を選択し、以下を実行します。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 には、SAML ユーザーの完全な「わかりやすい」名前に使用する SAML アサーション内の属性名を指定します。
  - アサーション属性ログイン では、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 E メール では、SAML ユーザーのユーザー E メール名に使用する SAML アサーション内の属性の名前を指定します。
  - ログイン有効期間 (分単位) では、SAML ユーザーのサインインが有効である期間を指定してから、ユーザーが再度サインインする必要があります。
  - アサーション属性組織 では、ユーザー組織の「フレンドリ」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。
  - アサーション属性グループ では、ユーザーグループの「わかりやすい」名前に使用する SAML アサーション内の属性の名前を指定します。
  - 許可された組織 の場合、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。許可する組織をカンマで区切って 1 つ以上入力します。
  - エディタのロール値 には、IdP のユーザーロールを入力します。IdP はすべて、Amazon Managed Grafana ワークスペースで Editor ロールを付与される必要があります。1 つ以上のロールをカンマで区切って入力します。

**Note**

管理者ロールまたはエディタロールが特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 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が表示されます。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 を選択し AWS IAM Identity Center、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促す Amazon Managed Grafana からの E メールに応答した場合にのみ機能します。

これで、Grafana ワークスペース、または論理 Grafana サーバーにいます。データソースの追加を開始して、データのクエリ、視覚化、分析を行うことができます。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 ようにセットアップする

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 リソースを作成して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 AWSを含む 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、 次のステップ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 サービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して[ルートユーザーアクセスが必要なタスク](#)を実行してください。

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

### のセキュリティ保護 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 「[ルートユーザーの仮想 MFA デバイスを有効にする \(コンソール\)](#)」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的チュートリアルについては、「[ユーザーガイド](#)」の「[デフォルトでユーザーアクセス IAM アイデンティティセンターディレクトリを設定するAWS IAM Identity Center](#)」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「[グループの参加](#)」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、Grafana ダッシュボードとビジュアライゼーション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、ワークスペースを管理、編集、または表示するためのアクセス許可を管理します。

ワークスペースを新しいバージョンの Grafana にアップグレードするか、を更新して Enterprise プラグインのサポートを追加することで、ワークスペースがより多くのタイプのデータソースにアクセスできるようになります。ワークスペースへのネットワークアクセスを管理することもできます。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 バージョンの違い](#)
- [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を管理する](#)
- [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 ワークスペースへのユーザーおよびグループのアクセスを管理する](#)
- [データソースと通知チャネルのアクセス許可を管理する](#)
- [を使用した Amazon Managed Grafana リソースの作成 AWS CloudFormation](#)
- [Amazon Managed Grafana ワークスペースへのネットワークアクセスを設定する](#)
- [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 バージョンの違い

[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。これらの各には、以前のバージョンから追加された機能があります。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 ワークスペースの使用に関する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注意事項と Grafana Labs 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、次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をクエリする方法を定義し、Explore 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なるかを確認し、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、「[クラシック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。
- 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簡単な情報を収集して製品サポートと共有する方法を提供します。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、「[サポートのための情報を収集する](#)」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](#)、[Datagrid](#)、[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。バージョン 9 のワークスペースでは、XY チャートを使用することもできます。
- PagerDuty – Enterprise プラグインに 用のプラグインが含まれるようになりました PagerDuty。
- 変換の再設計 – 変換タブでは、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。変換は分類され、各変換タイプには適切な変換を選択するのに役立つ図があります。
- Prometheus メトリクスの百科事典 – Prometheus クエリビルダーの Prometheus メトリクスのメトリクスドロップダウンは、ページ分割された検索可能なメトリクスの百科事典に置き換えられました。
- API キー UI の非推奨 – [サービスアカウント](#)は、Grafana HTTP APIs への呼び出しを認証するための推奨方法です。Grafana Labs の一部は API キーの非推奨化に取り組んでいますが、ワークスペースユーザーインターフェイスから API キーを作成することはできません。API キーは、AWS APIsを介してのみ作成できます。

Grafana Labs による API キーの非推奨化の詳細については、[Grafana GitHub 問題リストのAPIKeys : Sunsetting of API keys](#)」を参照してください。

## 重大な変更

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いくつかの変更がありました。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、非本番稼働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を与え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、角度を使用するパネルには、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ことが通知されます。
- CloudWatch 削除された のエイリアス — 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ダッシュボードを開き、保存します。エイリアスは自動的にラベルに移行されます。

- 古いプラグインをアップグレードする必要がある – Athena および Amazon Redshift データソースのプラグインは、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。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ついては、「」を参照してください[プラグインカタログでプラグインを検索する](#)。

- DoiT BigQuery プラグインはサポートされなくなりました – 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使用してください。
- 変換の変更 — 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が加えられています。詳細については、Grafana Labs ドキュメントの「[変換の重大な変更](#)」を参照してください。
- データソース許可 APIs – データソース許可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 ドキュメントの「[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。

重大な変更、およびプラグインデベロッパーに影響する変更の詳細については、Grafana Labs ドキュメントの以下のトピックを参照してください。

- [Grafana v10.0 の重大な変更](#)
- [Grafana v10.3 での変更の中断](#)

## Grafana バージョン 9

Grafana v9 では、次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でグループ名がサポートされるようになりました。
- 探索：探索ビュー内からダッシュボードを作成します。
- Prometheus クエリ：Prometheus クエリ用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簡単になります。
- Loki クエリ: Loki クエリ用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簡単になります。
- 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して、コミュニティプラグインをワークスペースにインストール、削除、または更新できます。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を制御できます。
- **メトリクスへのトレース:** クエリとタグを含むメトリクスへのリンクを追加するようにトレースデータソースを設定します。
- **Canvas パネル:** 静的要素と動的要素を含む新しいパネルの視覚化。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が容易になるように 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:** バグ修正と全体的な軽微な改善。

## 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的な 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オンする必要があります。

IAM Identity Center を使用して承認を行う最初のワークスペースを作成するには、IAM プリンシパルに次の追加ポリシー (または同等のアクセス許可) もアタッチされている必要があります。

- AWSSSOMemberAccountAdministrator
- AWSSSODirectoryAdministrator

詳細については、「[IAM Identity Center を使用して、単一の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します。](#)」を参照してください。

## ワークスペースの作成

次の手順では、新しい Amazon Managed Grafana ワークスペースを作成するプロセスについて説明します。



## Amazon Managed Grafana でワークスペースを作成するには

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

必要に応じて、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ます。また、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。例えば、ワークスペースにタグを割り当てることができ、タグを使用してワークスペースにアクセスするアクセス許可を持つことができるのは、限定されたグループまたはロールのみです。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。

### Workspace details

**Workspace name**  
Give an unique name to your workspace.

Valid special characters include "-", "\_", "~". Cannot contain non-ASCII characters or spaces.

**Workspace description - optional**

---

**▼ Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value	
<input type="text" value="owner"/>	<input type="text" value="user@amazon.com"/>	<input type="button" value="Remove"/>
<input type="button" value="Add new tag"/>		

You can add up to 49 more tags.

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。バージョンの違いについては、「」を参照してください[Grafana バージョンの違い](#)。
5. [次へ] をクリックします。



6. 認証アクセスで、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 Identity Center — IAM Identity Center を選択し、アカウント AWS IAM Identity Center でまだ有効にしていない場合は、最初の IAM Identity Center 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に関するユーザー管理を処理します。

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

- a. [ユーザーの作成] を選択します。
- b. ユーザーの E メールアドレス、名、姓を入力し、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 E メールアドレスを使用します。IAM Identity Center のこのアカウントのパスワードを作成するよう求める E メールメッセージが届きます。

 Important

作成したユーザーは、Amazon Managed Grafana ワークスペースに自動的にアクセスできません。後のステップで、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をユーザーに許可します。

- SAML — SAML を選択した場合は、ワークスペースの作成後に SAML 設定を完了します。


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 を選択すると、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は、カスタマー管理の 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、メンバーアカウントが組織内の委任管理者アカウントである必要があります。委任された管理者アカウントの詳細については、「[委任された管理者の登録](#)」を参照してください。

8. (オプション) このページで Amazon Virtual Private Cloud (VPC) に接続するか、後で VPC に接続するかを選択できます。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ンネルに接続する](#)」を参照してください。
9. (オプション) 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せん。
  - [Grafana アラートを有効にします](#)。

 Note

Grafana アラートを有効にすると、Grafana (Prometheus ではない) で定義されたアラートルールは、複数の通知を連絡先に送信します。ネイティブの Grafana アラートを使用している場合は、従来のダッシュボードアラートを維持することをお勧めします (新しい Grafana アラート機能を有効にしないでください)。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 Alertmanager で作成されたアラートに 1 つの通知が送信されます。

- 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化パネルのタイプに制限される場合があります。


ワークスペースの作成後に、これらの設定を変更することもできます。ワークスペースの設定の詳細については、「」を参照してください[Amazon Managed Grafana ワークスペースを設定する](#)。

10. (オプション) 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こともできます。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。ワークスペースを作成した後、ネットワークアクセスコントロールを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、「」を参照してください[Amazon Managed Grafana ワークスペースへのネットワークアクセスを設定する](#)。

11. [次へ] をクリックします。
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、現在のアカウントでのみ 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 に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された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を許可する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。委任された管理者アカウントの詳細については、[「委任された管理者の登録」](#)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、 を有効にするように求められた場合は AWS CloudFormation StackSets、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 (OU) を追加します。OUs Amazon Managed Grafana は、選択した各 OU 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. Organization を選択した場合は、データソースと通知チャンネル - オプション 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 は、Amazon Managed Grafana がこれらのソースからデータを読み取ることができる IAM ロールとアクセス許可を作成できます。データソースは引き続き Grafana ワークスペースコンソールに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からの Grafana アラートを Amazon Simple Notification Service (Amazon SNS) 通知チャン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、 で始まる TopicName 値を使用して、アカウントの Amazon SNS トピックに発行する IAM ポリシーを作成できます grafana。これにより、ワークスペースの通知チャンネルとして Amazon SNS が完全に設定されるわけではありません。これは、ワークスペースの Grafana コンソールで実行できます。
  15. [次へ] をクリックします。
  16. ワークスペースの詳細を確認し、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。

初期状態では、ステータスは CREATING です。

**⚠ Important**

ステータスが ACTIVE になるまで待つから、次のいずれかを実行します。

- SAML を使用している場合は、SAML セットアップを完了します。
- IAM Identity Center を使用している場合は、IAM Identity Center ユーザーにワークスペースへのアクセスを割り当てます。

現在のステータスを確認するには、ブラウザを更新する必要がある場合があります。

17. IAM Identity Center を使用している場合は、次の操作を行います。

- a. 認証 タブで、新しいユーザーまたはグループの割り当て を選択します。
- b. ワークスペースへのアクセス権を付与するユーザーの横にあるチェックボックスをオンにし、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オンにし、管理者にする を選択します。


**⚠ Important**

Grafana ワークスペース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Adminごとに少なくとも 1 人のユーザーを として割り当てます。

18. SAML を使用している場合は、次の操作を行います。

- a. 認証タブの Security Assertion Markup Language (SAML) で、セットアップの完了 を選択します。
- b. インポートメソッド で、次のいずれかを実行します。
  - URL を選択し、IdP メタデータの URL を入力します。
  - を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、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している場合は、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 には、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. 管理者ロールの値には、Amazon Managed Grafana ワークスペースでロールをすべて付与する必要がある IdP のユーザーAdminロールを入力するか、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かを選択します。

 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ことを選択した場合、データソース、ユーザー、ダッシュボードの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。ワークスペースの管理上の変更は、Amazon Managed Grafana APIs を使用してのみ行うことができます。

- e. (オプション) 追加の SAML 設定を入力するには、追加設定を選択し、以下を実行します。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には、SAML ユーザーの完全な「わかりやすい」名前に使用する SAML アサーション内の属性名を指定します。
  - アサーション属性ログインでは、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 E メールでは、SAML ユーザーのユーザー E メール名に使用する SAML アサーション内の属性の名前を指定します。
  - ログイン有効期間 (分単位) では、SAML ユーザーのサインインが有効である期間を指定してから、ユーザーが再度サインインする必要があります。デフォルトは 1 日、最大は 30 日です。
  - アサーション属性組織では、ユーザー組織の「わかりやすい」名前に使用する SAML アサーション内の属性の名前を指定します。
  - アサーション属性グループでは、ユーザーグループの「わかりやすい」名前に使用する SAML アサーション内の属性の名前を指定します。
  - 許可された組織の場合、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。許可する組織をカンマで区切って 1 つ以上入力します。
  - エディタのロール値には、IdP のユーザーロールを入力します。このEditorロールはすべて Amazon Managed Grafana ワークスペースで付与される必要があります。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 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が表示されます。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。
  - でサインインを選択し AWS IAM Identity Center、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促す Amazon Managed Grafana からの E メールに応答した場合にのみ機能します。

これで、Grafana ワークスペースまたは論理 Grafana サーバーにいます。データのクエリ、視覚化、分析に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。

の詳細については、「」を参照してください。

#### 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 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て、ダッシュボードを編集して表示します。ワークスペースにユーザーを割り当て、[ユーザー、エディタ、または管理者に のアクセス許可を付与](#)できます。開始するには、ユーザーを認証するための ID プロバイダーを作成 (または既存の) します。

ユーザーは、IAM を使用するのではなく、組織の ID プロバイダーを使用してシングルサインオンすることで、Amazon Managed Grafana ワークスペースで Grafana コンソールを使用するように認証されます。各ワークスペースは、次の認証方法の一方または両方を使用できます。

- Security Assertion Markup Language 2.0 (SAML 2.0IdPs) をサポートする ID プロバイダー () に保存されているユーザー認証情報
- AWS IAM Identity Center. AWS Single-sign-on ( AWS SSO ) が IAM Identity Center にブランド変更されました。

ワークスペースごとに、SAML、IAM Identity Center、またはその両方を使用できます。1つの方法を使用して開始する場合は、他の方法の使用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許可を付与する必要があります。ユーザーにアクセス許可を付与する方法の詳細については、「」を参照してください[Amazon Managed Grafana ワークスペースへのユーザーおよびグループのアクセスを管理する](#)。

## 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 で使用する](#)

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

### Note

Amazon Managed Grafana は現在、ワークスペースの IdP 開始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、空白のリレーステートで設定する必要があります。

SAML 認証を使用して既存の ID プロバイダーを使用し、Amazon Managed Grafana ワークスペースの Grafana コンソールにログインするためのシングルサインオンを提供できます。Amazon Managed Grafana の SAML 認証では、IAM による認証ではなく、サードパーティーの ID プロバイダーを使用してログイン、アクセスコントロールの管理、データの検索、視覚化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、Okta、CyberArk、Ping Identity との統合アプリケーションを構築 OneLogin およびテストした ID プロバイダーをサポートしています。

ワークスペースの作成時に SAML 認証を設定する方法の詳細については、「」を参照してください[ワークスペースの作成](#)。

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 とやり取り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許可を設定できます。IdP で定義されている組織ルールを 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 でデベロッパー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できます。IdP

#### Note

認証に IdP と SAML を使用する Amazon Managed Grafana ワークスペースを作成するときは、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オンする必要があります。

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 設定を読み取り、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り、ユーザーがサインインしてワークスペースを使用できることを確認します。

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) へ：
  - HTTP-POST バインディング
  - HTTP リダイレクトバインディング
- ID プロバイダー (IdP) からサービスプロバイダー (SP) へ：
  - HTTP-POST バインディング

Amazon Managed Grafana は、署名付き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、署名付き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 開始リクエストをサポートしており、IdP 開始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。コールバックには、SAML レスポンスに埋め込まれた、認証対象の



ユーザーの関連情報がすべて含まれています。Amazon Managed Grafana は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と、アサーション内の個々の属性を調べます。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、それよりも多くの情報が含まれています。

Amazon Managed Grafana には、これらの値を確認するキーを変更するための設定オプションが用意されています。

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 では、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 には、SAML ユーザーの完全な「フレンドリ」名に使用する SAML アサーション内の属性名を指定します。
- アサーション属性ログイン では、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 E メール では、SAML ユーザーのユーザー E メール名に使用する SAML アサーション内の属性の名前を指定します。
- アサーション属性組織 では、ユーザー組織の「フレンドリ」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。
- アサーション属性グループ では、ユーザーグループの「フレンドリ」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。
- 許可された組織 の場合、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。
- エディタのロール値 には、IdP のユーザーロールを指定します。この Editor ロールはすべて Amazon Managed Grafana ワークスペースで付与されます。

## ID プロバイダーへの接続

以下の外部 ID プロバイダーは Amazon Managed Grafana でテストされており、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アプリケーションを提供し、Amazon Managed Grafana を SAML で設定するのに役立ちます。

## トピック

- [Azure AD を使用するように Amazon Managed Grafana を設定する](#)
- [を使用するように Amazon Managed Grafana を設定する CyberArk](#)
- [Okta を使用するように Amazon Managed Grafana を設定する](#)
- [を使用するように Amazon Managed Grafana を設定する OneLogin](#)
- [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

Azure Active Directory を ID プロバイダーとして使用するように Amazon Managed Grafana を設定するには、次のステップを実行します。これらのステップは、Amazon Managed Grafana ワークスペースを既に作成しており、ワークスペース ID、URLs、および を書き留めていることを前提としていますAWS リージョン。

### ステップ 1: Azure Active Directory で完了するステップ

Azure Active Directory で次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとして Azure Active Directory を設定するには

1. 管理者として Azure コンソールにサインインします。
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、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] (Single Sign-On) を選択します。
9. Next を選択して、SAML 設定ページに移動します。
10. SAML 設定を指定します。
  - 識別子 (エンティティ ID) については、Amazon Managed Grafana ワークスペースから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) の場合は、Amazon Managed Grafana ワークスペースからサービスプロバイダーの返信を貼り付けます。

- 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 で設定されます。
    - displayName は user.displayname で設定されます。
    - 一意のユーザー識別子は user.userprincipalname で設定されます。
    - 渡すその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、「」を参照してください[アサーションマッピング](#)。
  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 で完了するステップ

Amazon Managed Grafana コンソールで次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとしての Azure Active Directory の設定を完了するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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、前のセクションで SAML メタデータ URL からコピーした Azure Active Directory URL をアップロードまたはコピー/貼り付けを選択します。
7. アサーションマッピング で、次の操作を行います。
  - 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ことを選択した場合、データソース、ユーザー、ダッシュボードの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はできません。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- 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。
- ( オプション) 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は、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 displayName 属性は Name 属性として渡され、Ping Identity メール属性は 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。

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

ID プロバイダー CyberArk として使用するように Amazon Managed Grafana を設定するには、次のステップを実行します。これらのステップは、Amazon Managed Grafana ワークスペースを既に作成しており、ワークスペースの ID、URLs、およびリージョンを書き留めていることを前提としています。

#### ステップ 1: で完了するステップ CyberArk

の次の手順を実行します CyberArk。

Amazon Managed Grafana の ID プロバイダー CyberArk として を設定するには

1. CyberArk Identity Admin Portal にサインインします。
2. アプリ、ウェブアプリ を選択します。
3. ウェブアプリの追加 を選択します。
4. Amazon Managed Grafana for SAML2.0 を検索し、「追加」を選択します。
5. CyberArk アプリケーション設定で、信頼セクションに移動します。
6. ID プロバイダー設定 で、メタデータ を選択します。
7. URL をコピー を選択し、これらのステップで後で使用する URL を保存します。
8. 「サービスプロバイダー設定」で、「手動設定」を選択します。
9. SAML 設定を指定します。

- SP エンティティ ID の場合は、Amazon Managed Grafana ワークスペースからサービスプロバイダー識別子 URL を貼り付けます。
  - Assertion Consumer Service (ACS) URL の場合は、Amazon Managed Grafana ワークスペースからサービスプロバイダーの返信を貼り付けます。
  - Sign Response Assertion を Assertion に設定します。
  - NameID 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、次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。
    - displayName は で設定されますLoginUser。DisplayName
    - mail は LoginUser.Email で設定されます。
    - 渡すその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、「」を参照してください[アサーションマッピング](#)。
  12. [保存] を選択します。
  13. アクセス許可セクションで、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 で完了するステップ

Amazon Managed Grafana コンソールで次の手順を実行します。

Amazon Managed Grafana の ID プロバイダー CyberArk としての設定を完了するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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、前の手順でコピーした CyberArk URL を貼り付けます。
7. アサーションマッピング で、次の操作を行います。

- 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ことを選択した場合、データソース、ユーザー、ダッシュボードのアクセス許可の管理などのタスクを含むワークスペースの管理に Amazon Managed Grafana ワークスペースコンソールを使用することはできません。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- 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。
- (オプション) CyberArk アプリケーションのデフォルト属性を変更した場合は、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、CyberArk displayName 属性は name 属性に渡され、CyberArk mail 属性は email 属性と login 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

Okta を ID プロバイダーとして使用するように Amazon Managed Grafana を設定するには、次のステップに従います。これらのステップは、Amazon Managed Grafana ワークスペースを既に作成しており、ワークスペースの ID、URLs、およびリージョンを書き留めていることを前提としています。

### ステップ 1: Okta で完了するステップ

Okta で次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとして Okta を設定するには

1. 管理者として Okta コンソールにサインインします。
2. 左側のパネルで、アプリケーション、アプリケーション を選択します。
3. App Catalog を参照を選択し、Amazon Managed Grafana を検索します。
4. Amazon Managed Grafana を選択し、追加、完了 を選択します。

5. セットアップを開始するアプリケーションを選択します。
6. サインオンタブで、**編集** を選択します。
7. 「高度なサインオン設定」で、Amazon Managed Grafana ワークスペース ID とリージョンを名前空間とリージョンフィールドに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、**Workspace-*id.grafana-workspace.Region*. ### Amazon Managed Grafana ##### URL** amazonaws.comにあります。
8. [保存] を選択します。
9. SAML 2.0 で、ID プロバイダーメタデータの URL をコピーします。これは、Amazon Managed Grafana コンソールでこの手順の後半で使用します。
10. 「割り当て」タブで、Amazon Managed Grafana を使用できるようにする人およびグループを選択します。

## ステップ 2: Amazon Managed Grafana で完了するステップ

Amazon Managed Grafana コンソールで次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとしての Okta の設定を完了するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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、前の手順でコピーした Okta URL のアップロードまたはコピー/貼り付け を選択します。
7. アサーションマッピング で、次の操作を行います。
  - 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ことを選択した場合、データソース、ユーザー、ダッシュボードの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



を管理することはできません。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- 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。
- (オプション) 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は、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 displayName 属性は name 属性に渡され、Okta mail 属性は 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。

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

ID プロバイダー OneLogin として使用するように Amazon Managed Grafana を設定するには、次のステップを実行します。これらのステップは、Amazon Managed Grafana ワークスペースを既に作成しており、ワークスペースの ID、URLs、およびリージョンを書き留めていることを前提としています。

### ステップ 1: で完了するステップ OneLogin

の次の手順を実行します OneLogin。

Amazon Managed Grafana の ID プロバイダー OneLogin として を設定するには

1. 管理者として OneLogin ポータルにサインインします。
2. アプリケーション、アプリケーション、アプリケーションの追加 を選択します。
3. Amazon Managed Service for Grafana を検索します。
4. 選択した表示名を割り当て、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 タブに、Amazon Managed Grafana ワークスペース URL を入力します。
7. adminRole パラメータをデフォルトのデフォルトなしのままにして、ルールタブを使用して入力できます。この例では、Amazon Managed Grafana のアサーション属性ロールは adminRole に設定され、値は true になります。この値は、テナント内の任意の属性にポイントできます。+ をクリックして、組織の要件に合わせてパラメータを追加および設定します。



8. ルール タブを選択し、ルール を追加 を選択し、ルールに名前を付けます。条件フィールド (if ステートメント) に、[E メールアドレス] を含む E メールを追加します。Actions フィールド (次に ステートメント) で、Amazon Managed Service AdminRole で Set を選択し、ドロップダウンに adminRole を設定 でマクロを選択し、値を true にします。組織は、異なるルールを選択して、異なるユースケースを解決できます。
9. [保存] を選択します。その他のアクションに移動し、使用権限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するたびに、マッピングを再適用する必要があります。
10. Amazon Managed Grafana コンソールの設定で後で使用する発行者 URL を書き留めます。次に、[Save] (保存) を選択します。
11. アクセスタブを選択して、Amazon Managed Grafana にアクセスするロールを割り当て OneLogin、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 で完了するステップ

Amazon Managed Grafana コンソールで次の手順を実行します。

Amazon Managed Grafana の ID プロバイダー OneLogin としての設定を完了するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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、前の手順でコンソールからコピーした発行者 URL をアップロードまたはコピー/貼り付けを選択し、貼り付け OneLogin します。OneLogin
7. アサーションマッピング で、次の操作を行います。
  - 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ことを選択した場合、データソース、ユーザー、ダッシュボードの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

を管理することはできません。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- 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。のデフォルト値は adminRole OneLogin 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。
- (オプション) 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は、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 displayName 属性は name 属性に渡され、OneLogin mail 属性は email 属性と login 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

Ping Identity を ID プロバイダーとして使用するように Amazon Managed Grafana を設定するには、次のステップを実行します。これらのステップは、Amazon Managed Grafana ワークスペースを既に作成しており、ワークスペースの ID、URLs、およびリージョンを書き留めていることを前提としています。

### ステップ 1: Ping Identity で完了するステップ

Ping Identity で次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとして Ping Identity を設定するには

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」を選択します。
4. Amazon Managed Grafana for SAML アプリケーションを検索し、それを選択して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。次に、次の SAML 設定を行います。
  - Assertion Consumer Service の場合、Amazon Managed Grafana ワークスペースからサービスプロバイダーの返信 URL を貼り付けます。

- エンティティ ID の場合は、Amazon Managed Grafana ワークスペースから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 にあり、Ping Identity 属性が Identity Bridge 属性 にあることを確認します。次に、次の設定を行います。
    - 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 E メール (作業 ) である必要があります。次に、この属性で高度な 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。
    - 渡す他の属性に を追加します。
    - 渡すその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、「」を参照してください[アサーションマッピング](#)。
  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. で始まる SAML メタデータ URL をコピーします `https://admin-api.pingone.com/latest/metadata/`。これは、後で設定で使用します。
  12. [Finish] を選択します。

## ステップ 2: Amazon Managed Grafana で完了するステップ

Amazon Managed Grafana コンソールで次の手順を実行します。

Amazon Managed Grafana の ID プロバイダーとしての Ping Identity の設定を完了するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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、前の手順でコピーした Ping URL を貼り付けます。
7. アサーションマッピング で、次の操作を行います。
  - 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ことを選択した場合、データソース、ユーザー、ダッシュボードのアクセス許可の管理などのタスクを含むワークスペースの管理に Amazon Managed Grafana ワークスペースコンソールを使用することはできません。ワークスペースの管理変更は、Grafana APIs を使用してのみ行うことができます。

- 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。
- ( オプション) 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は、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 displayName 属性は name 属性に渡され、Ping Identity mail 属性は 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 を使用すると、ユーザーは既存の会社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します。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 を使用しても、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場合、IAM Identity Center は Amazon Managed Grafana のユーザーとグループの作成に使用できる統合ユーザーディレ

クトリを提供します。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許可を割り当てることはサポートされていません。

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 の開始方法の詳細については、[「の開始方法」](#) 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、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、Amazon Managed Grafana で Organizations をアクティブ化できます。

## IAM Identity Center を使用するシナリオに必要なアクセス許可

このセクションでは、IAM Identity Center で Amazon Managed Grafana 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 の管理に必要なポリシーは、AWS アカウントが組織の一部であるかどうかによって異なります。

### アカウントに Grafana AWS Organizations 管理者を作成する

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、次のポリシーをロールに割り当てます。

- Amazon Managed AWSGrafanaAccountAdministrator Grafana ワークスペースの管理を許可する IAM ポリシーを割り当てます。
- AWSSSODirectoryAdministrator は、Amazon Managed Grafana ワークスペースを設定するときに、ロールが IAM Identity Center を使用できるようにします。
- 組織全体で Amazon Managed Grafana ワークスペースの作成と管理を許可するには、ロールに IAM AWSSSOMasterAccountAdministrator ポリシーを付与します。または、組織の AWSSSOMemberAccountAdministrator 単一のメンバーアカウント内でワークスペースを作成および管理することを許可する IAM ポリシーをロールに付与します。
- Amazon Managed Grafana ワークスペースを AWSMarketplaceManageSubscriptions Grafana インタープライズにアップグレードすることをロールに許可する場合は、オプションで IAM ポリシー (または同等のアクセス許可) をロールに付与することもできます。

Amazon Managed Grafana ワークスペースの作成時にサービス管理アクセス許可を使用する場合は、ワークスペースを作成するロールに iam:CreateRole、iam:CreatePolicy、および アクセス iam:AttachRolePolicy 許可も必要です。これらは、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ができるポリシーをデプロイ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。

に付与されたアクセス許可を確認するには `AWSGrafanaAccountAdministrator`、「」を参照してください。[AWS マネージドポリシー：AWSGrafanaAccountAdministrator](#)

単一の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します。

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。の詳細については [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アクセス許可を付与するには、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。

に付与されたアクセス許可を確認するには `AWSGrafanaAccountAdministrator`、「」を参照してください。[AWS マネージドポリシー：AWSGrafanaAccountAdministrator](#)

## 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 ワークスペースを新しいバージョンの Grafana に更新するには、2 つの方法があります。

### Note

バージョンは、新しいバージョンの Grafana にのみ更新できます。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、「」を参照してください [プラグインカタログでプラグインを検索する](#)。各バージョンの変更のリストについては、「」を参照してください [Grafana バージョンの違い](#)。

### 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. で Amazon Managed Grafana コンソールを開きます <https://console.aws.amazon.com/grafana>。
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 WorkSpaces] を選択します。
4. 更新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。このオプションを含めるのは、更新の対象となるワークスペースのみです。

### Warning

更新プロセスは元に戻せず、一時停止またはキャンセルすることはできません。本番稼働用ワークスペースを更新する前に、非本番稼働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。更新中にワークスペースを変更す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定します。
6. Workspaces タブで更新のステータスを定期的に確認します。更新プロセスには最大 10 分かかる場合があります。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。バナーの更新が表示され、ワークスペースの更新が成功したか失敗したかが示されます。更新に失敗した場合は、バナーで説明されているアクション項目に従って、もう一度試してください。



## 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. で Amazon Managed Grafana コンソールを開きます <https://console.aws.amazon.com/grafana>。
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 WorkSpaces] を選択します。
4. 更新するワークスペースのハイパーリンクが設定された Workspace 名を選択します。このオプションを含めるのは、更新の対象となるワークスペースのみです。
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せず、一時停止またはキャンセルすることはできません。本番稼働用ワークスペースを更新する前に、非本番稼働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。更新中にワークスペースを変更す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定します。
7. Workspaces タブで更新のステータスを定期的に確認します。更新プロセスには最大 10 分かかる場合があります。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。バナーの更新が表示され、ワークスペースの更新が成功したか失敗したかが示されます。更新に失敗した場合は、バナーで説明されているアクション項目に従って、もう一度試してください。

### Note

Amazon Managed Grafana の [UpdateWorkspaceConfiguration](#) オペレーションを使用してバージョンを更新することもできますAPI。

更新されたワークスペースで問題が発生した場合は、「」を参照してください [更新されたワークスペースに関する問題のトラブルシューティング](#)。

## 更新されたワークスペースに関する問題のトラブルシューティング

更新されたワークスペースは、更新後も引き続き機能します。このセクションは、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間で変更されています。

- 機能に問題を引き起こす可能性のある変更など、バージョン間の主な変更のリストについては、「」を参照してください[Grafana バージョンの違い](#)。
- バージョン 9 固有の機能のドキュメントについては、「」を参照してください[Grafana バージョン 9 での作業](#)。バージョン 10 については、「」を参照してください[Grafana バージョン 10 での作業](#)。

#### • Postgre SQLTLSの問題

バージョン 8 require で TLS/SSL モードが に設定されていて、ルート証明書のみを使用している場合、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。PostgreSQL データソース TLS の設定を変更します (Grafana ワークスペースのサイドメニューで、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 に変更します verify-ca。
- TLS/SSL メソッドを に設定します Certificate content。
- 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を管理する

Amazon Managed Grafana コンソールを使用してワークスペースを管理し、エンタープライズ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、さまざまなサードパーティーの独立系ソフトウェアベンダー (ISVs) 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では、[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エンタープライズ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 インフラストラクチャモニタリング (旧 SignalFx )
- ウェーブフロント

アップグレード時に使用できるエンタープライズデータソースプラグインの詳細については、「」を参照してください[Enterprise データソースに接続する](#)。新しいプラグインはいつでも追加できます。完全かつ最新のリストについては、Amazon Managed Grafana ワークスペース内で[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、デフォルトではエンタープライズ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 ページからアクセスを追加または削除することを含め、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が変更されました。以前に使用したことがある場合は AWS Marketplace、[FAQ AWS Marketplace エンタープライズユーザー向け](#)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 プラグインへのアクセスの管理](#)
- [アカウントを Grafana Labs にリンクする](#)
- [FAQ AWS Marketplace エンタープライズユーザー向け](#)

## Amazon Managed Grafana Enterprise プラグインへのアクセスの管理

Enterprise プラグインへのアクセスを管理するには

1. <https://console.aws.amazon.com/grafana> で [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 WorkSpaces] を選択します。

ワークスペースのリストを表示できます。エンタープライズライセンスの列には、ワークスペースごとに、ワークスペースのライセンスのタイプ (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。これにより、そのワークスペースのワークスペースの詳細ページが開きます。
5. 要約すると、エンタープライズライセンスで、Amazon Managed Grafana Enterprise の管理または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ステータスに基づいて、1つのオプションのみを使用できます)。

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。アクティブなオプションは ( 現行 ) とマークされます。

- なし – これは、Amazon Managed Grafana Enterprise ライセンスを削除するか、所有していないかのオプションです。現在エンタープライズライセンスをお持ちの場合、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 – これにより、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だけでなく、[Grafana Labs](#) のコンサルティングおよびサポートサービスへのアクセスも許可できます。Enterprise プラグインをワークスペースに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. 選択したら、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にアクセスできます。この機能にアクセスするには、[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 [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 回だけ登録する必要があります。アカウントが以前にリンクされていた場合 (例えば、Enterprise プラグインにアクセスするためにリージョン内の別のワークスペースをアップグレードする場合)、再度リンクするように求められません。

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 で新しいアカウントを作成することも、既存のアカウントを使用することもできます。

Grafana Labs トークンをコピーして、将来の使用のために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には

- 「」の手順に従って[Amazon Managed Grafana Enterprise プラグインへのアクセスの管理](#)、アクセスエンタープライズプラグインでアカウントをアップグレードします。アップグレードプロセス中に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。
- トークンが既にある場合は、直接入力できます。トークンがない場合は、「トークンの取得」を選択します。これにより、[Grafana Labs のウェブサイト](#)が新しいブラウザタブで開きます。

Grafana Labs 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、トークンを取得できます。

- トークンをコピーしたら、Amazon Managed Grafana ブラウザのタブまたはウィンドウに戻ります。Grafana Labs トークンセクションにトークンを入力します。
- これで、保存を選択してアップグレードを完了できます。

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 トークンの入力を求められた場合 (例えば、別のリージョンでワークスペースをアップグレードする場合)、毎回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。

- トークンを取得するには、Grafana Labs アカウントで検索し、「」の<https://grafana.com/partners/amg/support>に移動し、「マイアカウント」を選択します。
- を使用してトークン [DescribeWorkspace](#) API を取得することで、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。
- トークンがこれらの方法のいずれかで利用できなくなった場合は、[Grafana Labs サポート に連絡](#)する必要があります。

## FAQ AWS Marketplace エンタープライズユーザー向け

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が AWS Marketplace。を通じて新しいライセンスを購入することはできなくなり AWS Marketplace、以前にを通じて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、以下が役立つ FAQ 場合があります。

から 30 日間の無料トライアルをサブスクライブしましたが AWS Marketplace、ワークスペースに関連付けていません。今すぐ適用できますか？

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。

から 30 日間の無料トライアルを購入しましたが AWS Marketplace、すでにワークスペースに関連付けています。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。Enterprise プラグインをアップグレードして使用する場合は、前のセクションで説明したように、Amazon Managed Grafana コンソールから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 有料ライセンスを持っていますが、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。更新方法

現在の AWS Marketplace ライセンスを持っている限り、そのライセンスをワークスペースに関連付けることしかできません。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 を通じてキャンセルします AWS Marketplace ) 。

以下の質問と回答で詳細を説明します。

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。それらの処理はどうなりますか？

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は、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](#) できます。

ライセンスの有効期限が切れるとダウンタイムが発生し、ワークスペースが Enterprise プラグインにアクセスできなくな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？

新しい Enterprise プラグインライセンスに切り替えると、ライセンスの有効期限が切れるという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、これらを注意深く読む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には。

1. 準備するには、まず [Grafana Labs のウェブサイト](#) に移動し、アカウントにサインインします (または新しいウェブサイトを作成します )。プロセスの後半で使用する Grafana Labs トークンを取得します。

プロセスのこの部分の詳細については、「」を参照してください [アカウントを Grafana Labs にリンクする](#)。

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、の管理を選択します。これにより、サブスクリプションの詳細が表示されます。

 Note

このページには、サービスの終了日が表示されます。キャンセルする前に現在のサブスクリプションを最大限に活用するために、その日が近づくまで待ってこれらのステップを続行できます。

4. Actions を選択し、Cancel subscription を選択します。

これにより、のサブスクリプションがキャンセルされます AWS Marketplace。ただし、Amazon Managed Grafana が 1 日の終わりに (ワークスペースの現地時間) 自動的にライセンスを削除するまで、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. で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、「の管理」を選択します。

5. なしを選択し、を保存します。これにより、Amazon Managed Grafana から AWS Marketplace ライセンスが削除されます。

Enterprise ライセンスを削除すると、ワークスペースの Enterprise 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 プラグインへのアクセスの管理](#)トピックの手順に従います。



**Note**

Amazon Managed Grafana でライセンスをキャンセルしてから Enterprise プラグインにアクセスするためにアップグレードするまで、ワークスペースは Enterprise データソースにアクセスできません。これは通常 10~15 分程度ですが、これらのステップをどれだけ早く実行できるかによっては時間がかかる場合があります。Grafana Labs トークンの準備が整っていることを確認すると、この時間は最小限に抑えられます。

自動更新の AWS Marketplace ライセンスを持っています。これも続行されますか？

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、自動更新をセットアップした場合、無効にするまで継続されます。これを行うと、前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られていない 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。これは、自動更新を有効にしない限り、30 日以内に行われます。詳細については、前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 1 つのワークスペースから別のワークスペースに移行する場合があります。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し、既存のコンテンツを新しいワークスペースに移行するとします。

Amazon Managed Grafana はワークスペース間のコンテンツの移行を直接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提供することで、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、の「[Amazon Managed Grafana Migrator](#)」を参照してください GitHub。



# Amazon Managed Grafana ワークスペースへのユーザーおよびグループのアクセスを管理する

ID プロバイダー (IdP) または で設定されたユーザーを使用して Amazon Managed Grafana ワークスペースにアクセスします AWS IAM Identity Center。これらのユーザー (またはそれらが属するグループ) にワークスペースへのアクセス許可を付与する必要があります。、User、EditorまたはのAdminアクセス許可を付与できます。

## ユーザーまたはグループにアクセス許可を付与する

### 前提条件

- Amazon Managed Grafana ワークスペースへのアクセス権をユーザーまたはユーザーグループに付与するには、まずユーザーまたはグループを ID プロバイダー (IdP) または でプロビジョニングする必要があります AWS IAM Identity Center。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は、AWSSSOMemberAccountAdministratorおよび IAM AWSSSODirectoryReadOnly ポリシー、または同等のアクセス許可も必要です。詳細については、「[Amazon Managed Grafana へのユーザーアクセスの割り当てと割り当て解除](#)」を参照してください。

Amazon Managed Grafana コンソールを使用して Grafana ワークスペースへのユーザーアクセスを管理するには

1. <https://console.aws.amazon.com/grafana/> で 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 WorkSpaces] 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、次のいずれかを実行します。

- Amazon Managed Grafana ワークスペースへのアクセス権をユーザーに付与するには、ユーザーの横にあるチェックボックスをオンにし、ユーザーの割り当て を選択します。
- ユーザーをワークスペースAdminの 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アクセスを削除するには、ユーザーの割り当て解除 を選択します。
- LDAP グループなどのユーザーのグループを追加するには、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。
  - Amazon Managed Grafana ワークスペースへのアクセス権をグループのメンバー全員に付与するには、グループの横にあるチェックボックスを選択し、グループの割り当て を選択します。
  - グループのすべてのメンバー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 のすべてのメンバーのワークスペースアクセスを削除するには、グループ の割り当て解除 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は、IAM Identity Center コンソールを使用して新しいユーザーとグループをプロビジョニングするだけです。Amazon Managed Grafana コンソールまたは APIs を使用して、Grafana ワークスペースへのアクセスを許可または削除します。

IAM Identity Center と Amazon Managed Grafana が同期しなくなった場合は、競合を解決するオプションが表示されます。詳細については、以下の[ユーザーとグループの設定時のアクセス許可の不一致エラー](#)「」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は、SAML 設定を選択し、次のいずれかを実行します。
  - インポートメソッドで、次のいずれかを実行します。
    - URL を選択し、IdP メタデータの URL を入力します。
    - を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、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している場合は、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 には、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。
- 管理者ロールの値 には、Amazon Managed Grafana ワークスペースでロールをすべて付与する必要がある IdP のユーザーAdminロールを入力するか、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かを選択します。

 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ことを選択した場合、データソース、ユーザー、ダッシュボードの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。ワークスペースの管理上の変更は、Amazon Managed Grafana APIs を使用してのみ行うことができます。

- (オプション) 追加の SAML 設定を入力するには、追加設定 を選択し、次の操作を 1 つ以上実行してから、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 には、SAML ユーザーの完全な「わかりやすい」名前に使用する SAML アサーション内の属性名を指定します。
- アサーション属性ログイン では、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 E メール では、SAML ユーザーのユーザー E メール名に使用する SAML アサーション内の属性の名前を指定します。
- ログイン有効期間 (分単位) では、SAML ユーザーのサインインが有効である期間を指定してから、ユーザーが再度サインインする必要があります。
- アサーション属性組織 では、ユーザー組織の「フレンドリ」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。
- アサーション属性グループ では、ユーザーグループの「わかりやすい」名前に使用する SAML アサーション内の属性の名前を指定します。
- 許可された組織 の場合、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。許可する組織をカンマで区切って 1 つ以上入力します。
- エディタのロール値 には、IdP のユーザーロールを入力します。このEditorロールはすべて Amazon Managed Grafana ワークスペースで付与される必要があります。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、LDAP グループなどのユーザーのグループを追加するには、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。
  - Amazon Managed Grafana ワークスペースへのアクセス権をグループのメンバー全員に付与するには、グループの横にあるチェックボックスを選択し、グループの割り当てを選択します。
  - グループのすべてのメンバー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。
  - グループのすべてのメンバーのワークスペースアクセスを削除するには、グループの割り当て解除を選択します。

## ユーザーとグループの設定時のアクセス許可の不一致エラー

Amazon Managed Grafana コンソールでユーザーとグループを設定するときに、不一致エラーが発生する可能性があります。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 は警告と不一致を解決する選択肢を表示します。解決を選択すると、Amazon Managed Grafana は同期されていないアクセス許可を持つユーザーのリストを含むダイアログを表示します。

IAM Identity Center から削除されたユーザーは、として表示されUnknown user、ダイアログに数値 ID が表示されます。これらのユーザーの場合、不一致を修正する唯一の方法は、解決を選択し、アクセス許可を削除することです。

IAM Identity Center にまだいるが、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、Resolve リストにユーザー名が表示されます。この問題を解決するには、2つの方法があります。Resolve ダイアログを使用してアクセスを削除または減らすか、前のセクションの指示に従ってアクセスを許可できます。

## 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、アクセス許可が一致しないというエラーが表示されるのはなぜですか？

このメッセージが表示されるのは、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 のアクセス許可が一致していないためです。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、Grafana ワークスペースにユーザーを追加または削除できます。ただし、Grafana ユーザーアクセス許可は、ビューワー、エディタ、または管理者ア

アクセス許可をユーザーまたはグループに割り当てることによって、Amazon Managed Grafana から (Amazon Managed Grafana コンソールまたは APIs を使用して) のみ定義できます。ユーザーは、さまざまなアクセス許可を持つ複数のグループに属することができます。その場合、そのアクセス許可は、すべてのグループで最高レベルのアクセスレベルと、ユーザーが属するアクセス許可に基づきます。

レコードの不一致は、次の原因によって発生する可能性があります。

- ユーザーまたはグループは IAM Identity Center から削除されますが、Amazon Managed Grafana では削除されません。これらのレコードは、Amazon Managed Grafana コンソールで不明ユーザーとして表示されます。
- ユーザーまたはグループの Grafana との関連付けは、IAM Identity Center (アプリケーション割り当ての下) では削除されますが、Amazon Managed Grafana では削除されません。
- ユーザーアクセス許可は、以前に Grafana ワークスペースから直接更新されました。Grafana ワークスペースからの更新は、Amazon Managed Grafana ではサポートされていません。

これらの不一致を回避するには、Amazon Managed Grafana コンソールまたは Amazon Managed Grafana APIs 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アクセス許可を管理します。

以前に 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。これで、アクセスレベルが古いアクセスレベルに戻ったことがわかります。これが表示される理由と解決方法

これは、ほとんどの場合、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 のアクセス許可レコードとの間に不一致があることが原因です。チームメンバーがさまざまなアクセスレベルを経験している場合、Amazon Managed Grafana の管理者または が Amazon Managed Grafana コンソールから不一致を解決し、一致しない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 APIs から必要なアクセスレベルを再割り当てして、必要なアクセス許可を復元できます。

#### Note

ユーザーアクセス管理は、Grafana ワークスペースではサポートされていません。Amazon Managed Grafana コンソールまたは APIs を使用して、ユーザーまたはグループのアクセス許可を割り当てます。

アクセスレベルの変化に気付くのはなぜですか？例えば、以前は管理者アクセス権を持っていましたが、現在はエディタ権限しかありませんでした。

ワークスペースの管理者がアクセス許可を変更し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 のアクセス許可が一致しない場合に、誤って発生する可能性があります。この場合、不一致を解決すると、より高いアクセス許可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割り当てするように管理者にリクエストできます。

## データソースと通知チャネルのアクセス許可を管理する

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自動的に作成できます。

データソースと通知チャネルのアクセス許可とポリシーを管理するには

1. <https://console.aws.amazon.com/grafana/> で 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 WorkSpaces] 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. サービス管理アクセス許可とカスタマー管理アクセス許可の使用を切り替えるには、IAM ロールの編集アイコンを選択してから選択します。詳細については、「[AWS データソースの Amazon Managed Grafana のアクセス許可とポリシー](#)」を参照してください。

サービス管理アクセス許可からカスタマー管理アクセス許可に変更した場合、Amazon Managed Grafana が作成したロールとポリシーは現在のアカウントでは削除されません。組織のサービス管理アクセス許可を使用している場合、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. サービス管理アクセス許可を使用している場合は、IAM アクセス許可のアクセス設定の横にある編集を選択して、サービス管理アクセス許可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できます。詳細については、「[AWS データソースの Amazon Managed Grafana の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と、Amazon Managed Grafana は IAM ロールとアクセス許可を作成し、Amazon Managed Grafana がこれらのソースからデータを読み取ることを許可します。データソースは引き続き Grafana ワークスペースコンソールに追加する必要があります。

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

このワークスペースで使用する AWS 通知チャンネルを選択します。通知チャンネルを選択すると、Amazon Managed Grafana は、Amazon Managed Grafana がこれらのサービスを使用できるようにする IAM ロールとアクセス許可を作成できます。Grafana ワークスペースコンソールに通知チャンネルを追加する必要があります。

#### Note

通知の使用の詳細については、「」を参照してください [アラート通知を管理する](#)。

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

Amazon Managed Grafana 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に役立つサービスであると統合されています。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し、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、複数の AWS アカウントおよびリージョンで同じリソースを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および設定するには、[AWS CloudFormation テンプレート](#) を理解する必要があります。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 スタックで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

トの使用を開始できません。詳細については、「[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。AWS CloudFormation

Amazon Managed Grafana は、ワークスペースの作成をサポートしています AWS CloudFormation。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

の詳細については 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 ワークスペースへのネットワークアクセスを設定する

ユーザーとホストが Grafana ワークスペースにアクセスする方法を制御できます。

Grafana では、すべてのユーザーが認証および承認される必要があります。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開かれています。ワークスペースのネットワークアクセスコントロールを設定して、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できます。

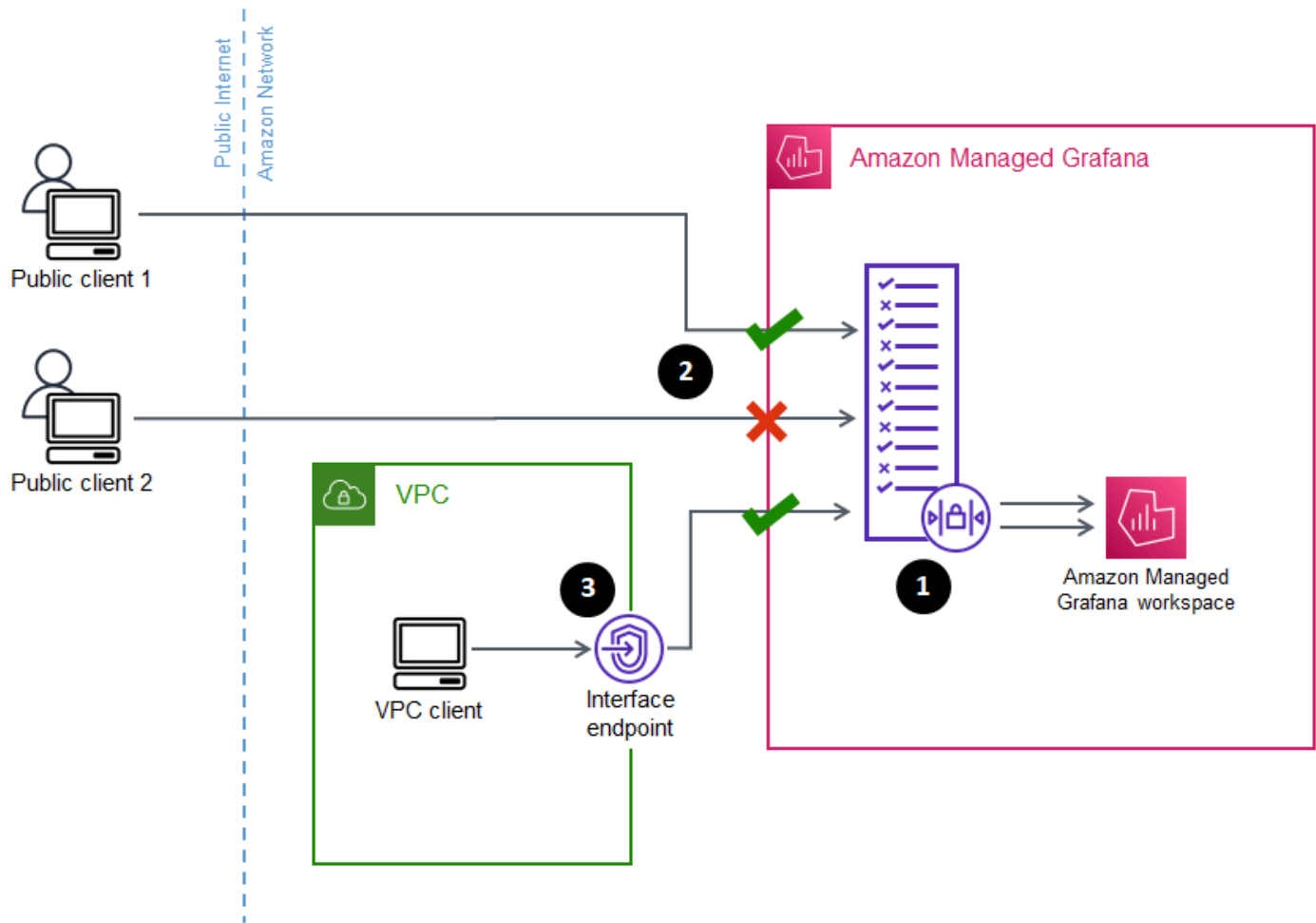
ワークスペースへのトラフィックは、2つの方法で制御できます。

- 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が許可されている IP 範囲を持つ [マネージドプレフィックスリスト](#) を作成できます。Amazon Managed Grafana は、ネットワークアクセスコントロール用のパブリック IPv4 アドレスのみをサポートします。
- VPC エンドポイント – 特定のワークスペースへのアクセスが許可されているワークスペースへの VPC エンドポイントのリストを作成できます。

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



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、Grafana ワークスペースへの接続を許可するリクエストを決定します。次の図は、このフィルタリングを示しています。



Amazon Managed Grafana ワークスペースのネットワークアクセスコントロール (1) を設定すると、ワークスペースへのアクセスを許可するリクエストを指定します。ネットワークアクセスコントロールは、IP アドレス (2) またはインターフェイスエンドポイントが使用されている (3) ごとに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コントロールの設定

ネットワークアクセスコントロールを既存のワークスペースに追加するか、ワークスペースの初回作成の一部として設定できます。

### 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ら、許可するエンドポイントごとに VPC エンドポイント ID が必要です。VPC エンドポイント IDs 形式は `vpce-1a2b3c4d`。

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、「」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](#)。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 エンドポイントの場合、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、そのアクセスをさらに制限できます。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とセキュリティグループのルールを関連付ける](#)」を参照してください。 [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html#SecurityGroupRules](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html#SecurityGroupRules)

- 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 に使用する場合、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は、最大 100 個の IP アドレス範囲を含めることができます。
  - プライベート IP アドレスの範囲 (例えば、`10.0.0.0/16`は無視されます。プライベート IP アドレス範囲をプレフィックスリストに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は無視します。これらのホストがワークスペースに到達できるようにするには、ワークスペースの VPC エンドポイントを作成し、アクセスを許可します。
  - Amazon Managed Grafana は、IPv6 ではなく、プレフィックスリスト内の IPv4 アドレスのみをサポートします。IPv6 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ら、Amazon Managed Grafana で許可する各リストのプレフィックスリスト ID が必要です。プレフィックスリスト IDs 形式は `p1-1a2b3c4d`。

プレフィックスリストの作成の詳細については、「Amazon Virtual Private Cloud 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ために必要なアクセス許可が必要です。例えば、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 IDs 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を作成する準備が整います。

#### Note

ネットワークアクセスコントロールを有効にし、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 アドレスを経由する場合を除き、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 エンドポイントを含める必要があります。含めない場合、どこからでもワークスペースにアクセスできなくなります。

ワークスペースのネットワークアクセスコントロールを設定するには

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

ワークスペースの作成時に、これらの同じオプションにアクセスできます。

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 を追加するかどうかを選択します。
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。

- 変更を保存を選択してセットアップを完了します。

**Warning**

ワークスペースの既存のユーザーがいる場合は、IP 範囲または VPC エンドポイントを設定に含めると、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ンネルに接続する

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ン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。これにより、Amazon Managed Grafana ワークスペースからパブリックにアクセス可能なサービスへの接続が制限されます。

**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。これには、CloudWatch、Amazon Managed Service for Prometheus、などのサービスが含まれます AWS X-Ray。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を経由しません。

VPC 内のプライベート向けデータソースに接続する場合、またはトラフィックを VPC にローカルに保つ場合は、Amazon Managed Grafana ワークスペースをこれらのデータソースをホストする Amazon Virtual Private Cloud (Amazon VPC) に接続できます。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 を経由します。

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 VPCs やパブリックインターネットを含む他の仮想ネットワークから論理的に分離されます。Amazon VPC を使用して、で VPCs を作成および管理します AWS クラウド。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 データソースおよびその他のリソースは、VPC で作成できます。Amazon VPC の詳細については、「Amazon Amazon Virtual Private Cloud [ユーザーガイド](#)」の「[Amazon VPC とは](#)」を参照してください。

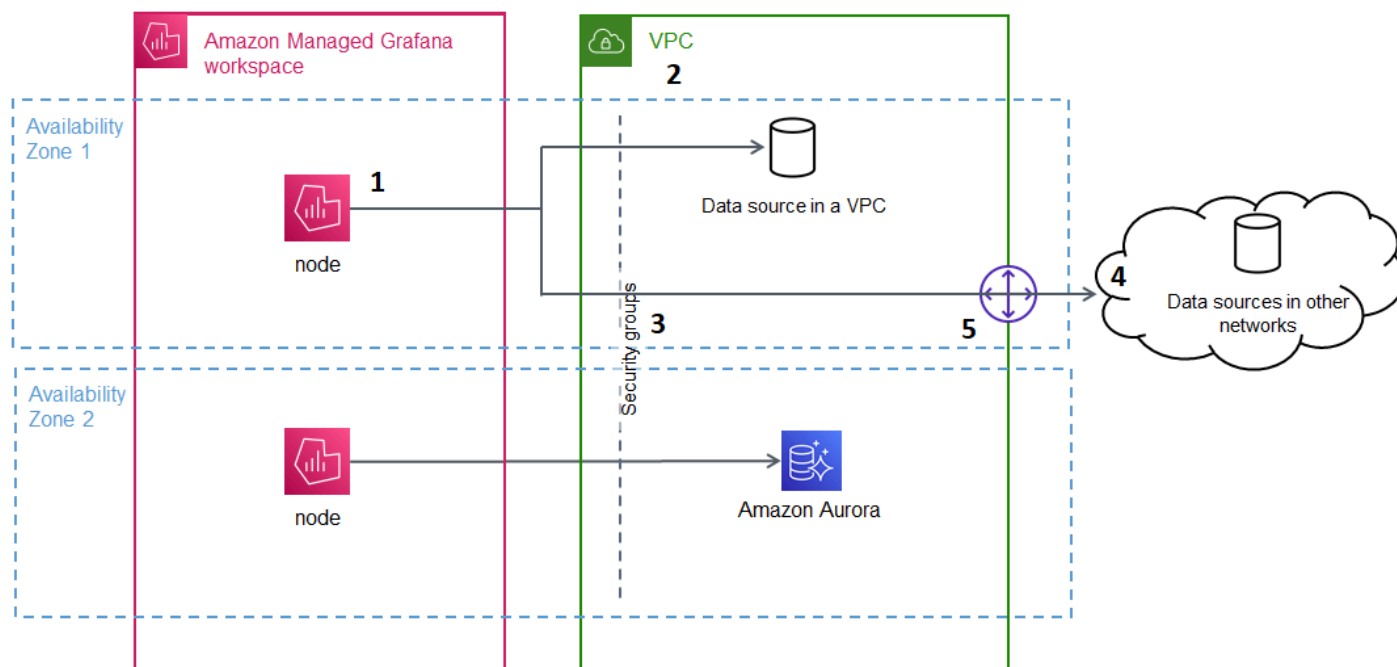
#### Note

Amazon Managed Grafana ワークスペースを VPC の外部にある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、他のネットワークに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 Virtual Private Cloud [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、接続するアプリケーション用のパブリックサブネットとプライベートサブネットの作成、サブネットにアクセスできるサービスまたはリソースを管理するセキュリティグループの作成など、仮想ネットワーク環境を完全に制御できます。

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 はその VPC 内の各アベイラビリティーゾーン内の指定された各サブネットにワークスペースを接続し、Amazon Managed Grafana ワークスペースとの間のすべてのトラフィックは VPC を経由します。次の図は、この接続が論理的にどのように見えるかを示しています。



Amazon Managed Grafana は、サブネットごとに接続 (1) を作成し ([Elastic Network Interface](#) または ENI を使用 )、VPC (2) に接続します。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。すべてのトラフィックは、アラートの送信先やデータソース接続など、設定された VPC を介してルーティングされます。他の VPCs またはパブリックインターネット (4) のデータソースとアラート送信先に接続するには、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。ワークスペースの作成時も、これらの同じ指示に従うことができます。ワークスペースの作成の詳細については、「」を参照してください [Amazon Managed Grafana ワークスペースを作成する](#)。

### 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の前提条件です。

- Amazon Managed Grafana ワークスペースを設定または作成するために必要なアクセス許可が必要です。例えば、AWS 管理ポリシー を使用できます `AWSGrafanaAccountAdministrator`。

- アカウントには、少なくとも 2 つの Availability Zones が設定され、それぞれに 1 つのプライベートサブネットが設定されている VPC を設定する必要があります。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。  
を Tenancy に設定して [設定された VPCs Dedicated](#) はサポートされていません。

- 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 CloudWatch されるなどのサービスが含まれます AWS PrivateLink。そうしないと、これらのデータソースへの接続が失われます。
- VPC に既に他のネットワークへの複数のゲートウェイがある場合は、複数のゲートウェイにまたがって DNS 解決を設定する必要がある場合があります。詳細については、[Route 53 Resolver](#) 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

次の手順では、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。このロールを使用すると、Amazon Managed Grafana は VPC への接続を作成できます。IAM ロールは、サービスにリンクされたロールポリシーを使用します AmazonGrafanaServiceLinkedRolePolicy。サービスにリンクされたロールの詳細については、「」を参照してください [Amazon Managed Grafana のサービスにリンクされたロールのアクセス許可](#)。

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

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」タブの「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 つを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 をサポートしている必要があります。
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. 変更を保存を選択してセットアップを完了します。

VPC 接続をセットアップしたので、その VPC から Amazon Managed Grafana ワークスペースにアクセス [データソースに接続する](#) 可能な を追加できます。

### アウトバウンド VPC 設定の変更

設定を変更するには、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 を使用できます。

#### Important

Amazon Managed Grafana が VPC 設定を管理します。Amazon EC2 コンソールまたは APIs 設定を編集しないでください。編集すると、設定が同期しなくなります。

## 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 (パブリックアクセス可能ではない) でのみ利用可能な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。

公開されているデータソース、またはパブリック向けエンドポイントを持つデータソースの場合、VPC を設定する必要はありません。

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、デフォルトでを介して AWS PrivateLink Amazon Managed Grafana に接続されます。

## 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に失敗するのはなぜですか？

既存のデータソースにはパブリックネットワーク経由でアクセスでき、Amazon VPC 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 を流れる必要があります。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 内のデータソース、VPC で利用できない AWS マネージドサービス、インターネット向けデータソースが含まれます。

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

- インターネット向けデータソースの場合は、VPC をインターネットに接続します。例えば、[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できます](#) (Amazon Virtual Private Cloud ユーザーガイド から)。
- 他の VPCs 間の VPCs 間にピアリングを作成します。詳細については、[VPCs ピアリングを使用して VPC を接続する](#) ( Amazon Virtual Private Cloud ユーザーガイド) 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る場合があります。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。

Managed AWS Services (Amazon Managed Service for Prometheus CloudWatch、または 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 ([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) を設定し、同じ VPC に接続するように Amazon Managed Grafana ワークスペースを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 **502 Bad Gateway Error**しようとする、を取得するのはなぜですか？

以下は、データソース接続が 502 エラーを返す最も一般的な 3 つの理由です。

- セキュリティグループエラー — Amazon Managed Grafana の VPC 設定中に選択されたセキュリティグループは、インバウンドルールとアウトバウンドルールを介したデータソースへの接続を許可する必要があります。

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は、データソースをクエリするための適切なアクセス許可がありません。

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 アクセス許可と、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するための正しいデータソースポリシーがあ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。

- 正しくない接続の詳細が提供されました — 正しくない接続の詳細が提供されているため、Amazon Managed Grafana ワークスペースは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、データソース認証やエンドポイント URL など、データソース接続の情報を確認し、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 VPCs に接続できますか？

Amazon Managed Grafana ワークスペースに対して設定できる VPC は 1 つだけです。別の 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。

別の 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 がある 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 [AWS Transit Gateway](#)、クロスリージョンまたはクロスアカウント VPCs 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 を接続できます。Amazon Managed Grafana は、VPC 内の他の接続として外部データソースに接続します。

#### Note

VPC ピアリングがオプションでない場合は、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](mailto: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？

はい。VPC とパブリックデータソースの両方のデータソースを単一の Amazon Managed Grafana ワークスペースに同時に接続できます。パブリックデータソースの場合、[NAT Gateway 経由で VPC 接続を設定するか](#)、他の [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、それらのリクエストをさらに可視化し、制御できます。

VPC 接続を設定する前に、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ためのルートがあることを確認します。例えば、サードパーティーがホストするアラート通知の送信先では、インターネットへの接続が必要になる場合があります。データソースと同様に、インターネットまたは AWS Transit Gateway、または外部宛先への他の VPC 接続を設定します。

VPC を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 VPC 接続は、セキュリティグループとサブネットを使用して、VPC と Amazon Managed Grafana ワークスペース間で許可されるトラフィックを制御します。セキュリティグループまたはサブネットが Amazon Managed Grafana コンソール外 (VPC コンソールなど) で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。この問題を解決するには、Amazon Managed Grafana コンソールで Amazon Managed Grafana ワークスペース用に設定されたセキュリティグループを更新します。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 設定は、Amazon Managed Grafana の認証とアクセス許可の設定、および Grafana ワークスペースの設定に分割できます。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 ワークスペースへのユーザーおよびグループのアクセスを管理する](#)
- [ユーザー、チーム、アクセス許可](#)

Amazon Managed Grafana 内の Grafana ワークスペースのプロパティを表示するときに、Workspace 設定オプションタブで Grafana ワークスペースの設定を変更できます。

Grafana インスタンスの設定を変更すると、インスタンスが再起動して新しい設定が再ロードされる可能性があります。設定が変更されると、ユーザーは Grafana ワークスペースを表示する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て Grafana ワークスペースの設定を変更するには

1. <https://console.aws.amazon.com/grafana/> で 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 WorkSpaces] 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すると、インスタンスのインスタンス設定オプションが表示されます。
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](#)できます。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。これにより、Grafana アラートに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、Grafana アラートをオンにすることはお勧めしません。

Prometheus アラートを表示せずに、代わりに Grafana マネージドアラートを使用する場合は、[Grafana アラートをオンにする](#)の横にあるチェックボックスをオフにします。これにより、[クラシックダッシュボードのアラート](#)が有効になります。Grafana アラートをオンにしない場合でも、既存の Grafana アラートが評価されます。

- プラグイン管理

プラグイン管理を有効にするには、チェックボックスをオンにして [プラグイン管理を有効に](#)します。プラグイン管理をオンにすると、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。これには、作成した新しいアラートルールが含まれます。

Grafana アラートの使用と、アラートをオンまたはオフにした場合の影響の詳細については、「」を参照してください[Grafana バージョン 10 のアラート](#)。

次のセクションでは、Amazon Managed Grafana API または を使用して Grafana インスタンス設定を変更する方法を示します AWS CLI。

## API または での設定 AWS CLI

Grafana ワークスペースの設定は、Amazon Managed Grafana API または を使用して設定できます AWS CLI。

### Note

configuration は、後で追加される将来の設定を可能にする JSON 文字列です。

## AWS CLI

を使用して Amazon Managed Grafana インスタンス設定を更新するには AWS CLI

次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>` 文字列をインスタンスに適した値に置き換えます。

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

設定は現在、次のオプションをサポートしています。これにより、Grafana アラートまたはプラグイン管理がオンまたはオフになります。

- Grafana アラートを有効にするには、こ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、こ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 を使用して Amazon Managed Grafana インスタンス設定を更新するには

次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。<code>workspace-id</code> 文字列をインスタンスに適した値に置き換えます。

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

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

設定は現在、次のオプションをサポートしています。これにより、Grafana アラートまたはプラグイン管理がオンまたはオフになります。

- Grafana アラートを有効にするには、こ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、こ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のすべての設定データも削除されます。これには、ダッシュボード、データソース設定、アラート、スナップショットが含まれます。

Amazon Managed Grafana ワークスペースを削除するには

1. <https://console.aws.amazon.com/grafana/> で 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 WorkSpaces] 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、ワークスペースの名前を入力し、「削除」を選択します。

### Note

この手順では、ワークスペースを削除します。他のリソースは削除できません。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 (使用されていない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

Grafana ワークスペースでは、プロジェクトに取り組み、視覚化を作成し、メトリクスを調べます。メトリクスのデータソースをセットアップしてクエリします。ダッシュボード内にパネルを作成して、メトリクスを表示します。データを調べます。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 ワークスペースを使用する方法について説明します。

## Note

一部のトピックは、ワークスペースにある Grafana のバージョンによって異なります。各バージョンに固有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「」を参照してください。[Grafana バージョン 8 での作業](#)。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、「」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](#)。

## トピック

- [Grafana とは](#)
- [ワークスペースに接続する](#)
- [ユーザー、チーム、アクセス許可](#)
- [最初のダッシュボードを作成する](#)
- [プラグインを使用してワークスペースを拡張する](#)
- [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](#)
- [サポートのための情報を収集する](#)
- [Classic 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化および分析ソフトウェアです。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、探索に使用できます。

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できます。お客様が企業の管理者であり、複数のチームの Grafana を管理している場合は、プロビジョニングと認証の設定が必要になる場合があります。

以下のセクションでは、Grafana データベースでできることの概要と、詳細を学習するためのリンクについて説明します。

## メトリクスとログを調べる

1 回限りのクエリまたはアドホッククエリと動的ドリルダウンを通じてデータを探索します。ビューを分割し、さまざまな時間範囲、クエリ、データソースを並べて比較できます。

詳細については、「[Grafana バージョン 10 でを試す](#)」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は、次のようなさまざまなアラート通知機能を使用してアラートを送信できます。

- 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リッチイベントでグラフに注釈を付けます。イベントを一時停止すると、完全なイベントメタデータと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を関連付けるのに役立ちます。Ctrl を押しながらグラフを選択し、テキストを入力すると、手動で注釈を作成できます。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、さまざまなユースケースで再利用できるダッシュボードを作成します。これらのテンプレートでは、値はハードコーディングされません。つまり、複数のサーバーにダッシュボードを使用できます。例えば、本番サーバーとテストサーバーがある場合、両方に同じダッシュボードを使用できます。

テンプレートを使用すると、データをドリルダウンできます。例えば、すべてのデータから北米データ、テキサスデータなどにドリルダウンできます。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用の優れたダッシュボードテンプレートを作成する場合は、コミュニティ全体に提供して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。

## ワークスペースに接続する

Amazon Managed Grafana ワークスペースを使用する前に、設定した ID プロバイダーに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](#)「」で詳細を確認してください。

### Note

プログラムでワークスペースに接続しようとする場合は、API トークンを使用する必要があります。詳細については、「[トークンによる認証](#)」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には

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開き、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 WorkSpaces] 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。

- 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。
- ワークスペース 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が表示されます。でサインインを選択し、E AWS IAM Identity Center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ている場合、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、アクセス許可

Amazon Managed Grafana のアクセス許可は、Amazon Managed Grafana コンソール全体および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 コンソールでを通じて設定した ID プロバイダー SAML 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 ロールでユーザーまたはグループに [アクセス権](#) を付与し、ワークスペースへのデフォルトのアクセス許可を付与できます。
- 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 つの方法でへのアクセスを許可できます。ID プロバイダー (または Identity Center) IAM でグループを作成できます。その後、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権を付与できます。または、Grafana ワークスペースで [Teams](#) を作成し、必要なロールを割り当てることもできます。
- 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によって付与された [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、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行われます。

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 ワークスペースにユーザーを追加しません。代わりに、IAM Identity Center または ID プロバイダーと 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をユーザーに許可します。詳細については、「[Amazon Managed Grafana ワークスペースへのユーザーおよびグループのアクセスを管理する](#)」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を有効にした各ユーザーは、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、次の操作を実行できます。
  - 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および編集できます。
  - ダッシュボードを含むフォルダを追加、編集、削除できます。
  - エディタロールで許可されていることをすべて実行できます。
- エディタロール — エディタロールを持つユーザーは、次の操作を実行できます。
  - ダッシュボード、パネル、アラートルールは、アクセスできるダッシュボードで表示、追加、編集できます。これは、特定のフォルダとダッシュボードで無効にできます。
  - 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。
  - 通知チャンネルを追加、編集、削除できます。
  - データソースを追加、編集、または削除することはできません。
  - ビューワーロールで許可されているすべての操作を実行できます。
- ビューワーロール — ビューワーロールを持つユーザーは、次の操作を実行できます。

- アクセス可能なダッシュボードを表示できます。これは、特定のフォルダとダッシュボードで無効にできます。
- プレイリストを作成、更新、または削除することはできません。
- Explore にアクセスできません。
- 通知チャンネルを追加、編集、削除することはできません。
- データソースを追加、編集、または削除することはできません。
- ダッシュボードやパネルを追加、編集、または削除することはできません。
- 他のユーザーやチームを管理することはできません。

Grafana ワークスペースからのユーザー割り当てとユーザーアクセス管理は、Amazon Managed Grafana ではサポートされていません。ユーザーおよびグループのアクセスを管理する方法は、IAM Identity Center を使用するか、SAML認証に使用するかによって異なります。

- ワークスペースで認証に IAM Identity Center を使用している場合は、Amazon Managed Grafana コンソールまたは を使用してロールAPIsを割り当てることができます。詳細については、[「Amazon Managed Grafana ワークスペースへのユーザーおよびグループのアクセスを管理する」](#)を参照してください。
- ワークスペースが認証SAMLに を使用する場合、ユーザーロールはアサーション属性によってのみ定義されます。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アクセス許可を同時に付与できます。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で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

グループ内のユーザーを管理するチームを作成します。

チームを作成するには

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、 の作成 を選択します。



## チームを削除するには

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. 確認するには、「削除」を選択します。

## チームへのユーザーの追加または削除

ユーザーをチームに追加したり、チームから削除したりするには、以下の手順に従います。

### ユーザーをチームに追加するには

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」ボックスで、チームに追加するユーザーを選択し、「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には

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. 確認するには、「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、認証プロバイダーのグループと Grafana の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 と SAML です。

Grafana チームと外部グループを同期するには。

1. Grafana コンソールで、設定 、チーム に移動します。
2. IAM Identity Center グループと同期するには、IAM Identity Center 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. [Add Group (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ているアクセス許可によって定義されます。

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に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るアクセス許可

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、およびデータソースのアクセス許可に基づいて、アクセス許可を付与できます。

### ダッシュボードとフォルダのアクセス許可の概要

ダッシュボードと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のロールベースのアクセス許可を削除できます。その後、特定のユーザーとチームに許可を割り当てることができます。詳細については、「[ダッシュボードとフォルダのアクセス許可](#)」を参照してください。

### データソースのアクセス許可の概要

デフォルトでは、データソースはどのユーザーでもクエリできます。例えば、Viewer ロールを持つユーザーは、アクセスできるダッシュボードに存在す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、クエリのアクセス許可を特定のユーザーおよびチームに制限できます。詳細については、「[データソースのアクセス許可](#)」を参照してください。

### ダッシュボードとフォルダのアクセス許可

ダッシュボードとダッシュボードフォルダの場合、アクセス許可ページを使用して、エディタとビューワーのデフォルトのロールベースのアクセス許可を削除できます。このページでは、特定のユーザーとチームにアクセス許可を追加して割り当てることができます。

Amazon Managed Grafana には、次のアクセス許可レベルがあります。アクセス許可は、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: ダッシュボードを編集および作成し、アクセス許可を編集できます。フォルダを追加、編集、削除することもできます。
- Edit: ダッシュボードを編集および作成できます。フォルダやダッシュボードのアクセス許可を編集したり、フォルダを追加、編集、削除したりすることはできません。
- View: 既存のダッシュボードとフォルダのみを表示できます。

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: ダッシュボードを作成、編集、または削除できます。フォルダを追加、編集、または削除し、フォルダにダッシュボードとサブフォルダを作成できます。管理者は、ダッシュボードとフォルダのアクセス許可を変更することもできます。
- Edit: ダッシュボードを作成、編集、または削除できます。フォルダを編集または削除し、フォルダにダッシュボードとサブフォルダを作成できます。エディタはフォルダまたはダッシュボードのアクセス許可を変更できません。
- View: 既存のダッシュボードとフォルダのみを表示できます。

## フォルダのアクセス許可の付与

フォルダのアクセス許可を付与するには

1. サイドバーで、ダッシュボード (二乗) アイコンにカーソルを合わせ、 の管理を選択します。
2. フォルダにカーソルを合わせ、フォルダ に移動します。
3. アクセス許可タブで、アクセス許可の追加 を選択します。
4. 「アクセス許可の追加」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、アクセス許可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オプションを選択した場合は、このステップをスキップします。
6. 3 番目のボックスで、追加するアクセス許可を選択します。
7. [保存] を選択します。

## ダッシュボードのアクセス許可の付与

ダッシュボードのアクセス許可を付与するには

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. アクセス許可タブで、アクセス許可の追加 を選択します。
3. 「アクセス許可の追加」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、アクセス許可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オプションを選択した場合は、このステップをスキップします。
5. 3 番目のボックスで、追加するアクセス許可を選択します。
6. [保存] を選択します。

## アクセスの制限

最高のアクセス許可が常に優先されます。

- Admin ロールを持つユーザーのアクセス許可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。
- アクセス許可レベルが低いより具体的なアクセス許可は、アクセス許可レベルが高いより一般的なルールが存在する場合、効果がありません。より一般的なルールのアクセス許可レベルを削除または下げる必要があります。

## Amazon Managed Grafana が複数のアクセス許可を解決する方法 – 例

次の例は、複数のアクセス許可がどのように解決されるかを示しています。

### 例 1: **user1** に **Editor** ロールがある

ダッシュボードのアクセス許可 :

- Editor ロールを持つすべてのユーザーが編集できます。
- **user1** を表示できます。

結果: 最も高いアクセス許可が常に優先されるため、には編集アクセス許可**user1**があります。

## 例 2: **user1** にビューワールールがあり、 のメンバーである **team1**

ダッシュボードのアクセス許可 :

- Viewer ルールを持つすべてのユーザーが表示できます。
- **user1** には Editorルールがあり、編集できます。
- **team1** には Adminルールがあります。

結果: は管理者権限**user1**を持ちます。これは、最も高い権限が常に優先されるためです。

## 例 3: 異なるレベルで複数のアクセス許可**user1**がある

ダッシュボードのアクセス許可 :

- **user1** には、Adminルール (親フォルダから継承) があります。
- **user1** には Editorルールがあり、編集できます。

結果: より低いアクセス許可に上書きすることはできません。 **user1**には管理者アクセス許可があります。これは、最も高いアクセス許可が常に優先されるためです。

### [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。
- アクセス許可レベルが低いより具体的なアクセス許可は、アクセス許可レベルが高いより一般的なルールが存在する場合、効果がありません。

## データソースのアクセス許可

デフォルトでは、データソースはどのユーザーでもクエリできます。例えば、Viewerルールを持つユーザーは、アクセス権のあるダッシュボードに存在す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して、ユーザーがデータソースをクエリするためのアクセスを制限できます。データソースごとに、特定のユーザーおよびチーム に対してクエリのアクセス許可を有効化または制限できるアクセス許可ページがあります。

## データソースのアクセス許可の有効化

データソースのアクセス許可が有効になっている場合は、デフォルトでそのデータソースの管理者およびクエリアクセスを管理者ユーザーに制限します。特定のユーザーやチームにアクセス権を選択的に追加できます。

データソースのアクセス許可を有効にするには

1. 設定、データソースに移動します。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、データソースに移動します。
2. アクセス許可を有効にするデータソースを選択します。
3. アクセス許可タブで、を有効にするを選択します。

### Warning

デフォルトのデータソースのアクセス許可を有効にすると、アクセス許可に記載されてい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、それらのユーザーの Access denied to data source エラーを返します。

ユーザーとチームがデータソースをクエリできるようにする

データソースのアクセス許可を有効にすると、管理者のみがデフォルトで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許可を割り当てることができます。クエリのアクセス許可により、データソースをクエリするためのアクセスが許可されます。

ユーザーとチームにクエリ許可を割り当てるには

1. 設定、データソースに移動します。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、データソースに移動します。
2. クエリ許可を割り当てるデータソースを選択します。
3. アクセス許可タブで、アクセス許可の追加を選択します。
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の場合は、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. クエリへのアクセスを許可するチーム、ユーザー、サービスアカウント、またはロールを選択し、の保存を選択します。

## データソースのアクセス許可の無効化

データソースのアクセス許可を有効にしている、データソースのアクセス許可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アクセス許可はすべて削除されます。

データソースのアクセス許可を無効にするには

1. 設定、データソース 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 に移動します。
2. アクセス許可を無効にするデータソースを選択します。
3. アクセス許可タブで、アクセス許可を無効にする を選択します。

## 最初のダッシュボードを作成する

### 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、次の手順に従います。

最初のダッシュボードを作成するには

1. 左パネルの + アイコンを選択し、ダッシュボードの作成 を選択し、新しいパネルの追加 を選択します。
2. 新しいダッシュボード/編集パネルビューで、クエリタブを選択します。
3. クエリを実行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 を追加した場合、Random Walk 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

外の温度が 1 日を通してどのように変化するか知りたいと想像してください。1 時間に 1 回、 の計測値をチェックし、現在の温度とともに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表示されます。



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す。時系列で順序付けられた一連の測定です。テーブルの各行は、特定の時間に 1 つの個別の測定値を表します。

テーブルは、個々の測定値を特定する場合に便利ですが、全体像を把握することが困難になる場合があります。時系列のより一般的な視覚化はグラフで、代わりに各測定値を時間軸に沿って配置します。グラフなどの視覚的な表現により、そうしないと見にくいデータのパターンや特徴を簡単に検出できます。

時系列のその他の例は次のとおりです。

- CPU とメモリの使用量
- センサーデータ
- 株式市場インデックス

これらの各例は時系列順に並べられた測定ですが、他の属性も共有します。

- 新しいデータは、最後に一定の間隔で追加されます。例えば、09:00、10:00、11:00 などです。
- 測定値は追加後に更新されることはほとんどありません。例えば、昨日の温度は変わりません。

時系列は強力です。システムの状態をいつでも分析できるようにすることで、過去の理解に役立ちます。時系列では、空きディスク容量がゼロ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。

時系列は、データの傾向を明らかにすることで、未来を予測するのにも役立ちます。例えば、登録済みユーザーの数が過去数か月間に毎月 4% 増加している場合、ユーザーベースが年末にどれだけ大きくなるかを予測できます。

一部の時系列には、既知の期間にわたって繰り返されるパターンがあります。例えば、温度は通常、夜間に低下する前に、日中に高くなります。これらの定期的または季節的な時系列を特定すること

で、次の期間について自信を持って予測を行うことができます。システムの負荷が毎日 18:00 頃にピークに達することがわかっている場合は、直前にマシンを追加できます。

## 時系列の集計

測定対象に応じて、データは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なりますか？ 温度を 1 時間に 1 回測定すると、1 日あたり 24 個のデータポイントになります。8 月の温度を長年比較するには、31 倍 24 のデータポイントを 1 つに結合する必要があります。

測定値のコレクションを組み合わせることを集約と呼びます。時系列データを集計する方法はいくつかあります。一般的なものをいくつか紹介します。

- Average は、すべての値の合計を値の合計で割った値を返します。
- Min と Max は、コレクション内の最小値と最大値を返します。
- Sum は、コレクション内のすべての値の合計を返します。
- Count はコレクション内の値の数を返します。

例えば、データを 1 か月で集計することで、2017 年 8 月が平均して前年よりも暑いと判断できます。どの月が最高温度であったかを確認したい場合は、各月の最高温度を比較します。

時系列データを集計する方法は重要な決定であり、データで伝えたいストーリーによって異なります。さまざまな集計を使用して、同じ時系列データをさまざまな方法で視覚化するのが一般的です。

## 時系列とモニタリング

IT 業界では、インフラストラクチャ、ハードウェア、アプリケーションイベントなどのモニタリングのために時系列データが収集されることがよくあります。マシンが生成する時系列データは、通常短い間隔で収集されるため、予期しない変更や変更が発生した直後に対応できます。データは急速に蓄積されるため、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不可欠です。その結果、時系列データ用に最適化されたデータベースは、ここ数年で人気が高まっています。

## 時系列データベース

時系列データベース (TSDB) は、時系列データ用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は可能ですが、TSDB にはいくつかの便利な最適化があります。

最新の TSDBs は、測定値が追加されることはなく、更新または削除されることはほとんどないという事実を利用しています。例えば、各測定のタイムスタンプは時間の経過とともに少しずつ変化するため、冗長データが保存されます。

次の例は、Unix タイムスタンプのシーケンスを示しています。

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

これらのタイムスタンプを見ると、それらはすべてで始まり1572524、ディスク容量の使用率が低下します。代わりに、次の例に示すように、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分または差分として保存できます。

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

次の例に示すように、これらの差分の差分を計算することで、さらに一步先に進むこともできます。

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

測定値が一定の間隔で行われる場合、そのほとんどは 0 delta-of-deltas になります。このような最適化により、TSDBs他のデータベースよりも大幅に少ないスペースを使用します。

TSDB のもう 1 つの機能は、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、測定が行われた場所などのコンテキスト情報を追加するタグが付けられます。

Grafana では、次の TSDBs 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

時系列を保存する場所ができたので、実際に測定値を収集する方法を教えてください。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、マシン、またはインスタンスにコレクターをイ

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、一部のコレクターは異なる出力先をサポートしています。

コレクターの例を次に示します。

- [収集済み](#)
- [統計](#)
- [Prometheus エクスポーター](#)
- [テレグラフ](#)

コレクターは、データをデータベースにプッシュするか、データベースがコレクターからデータをプルできるようにします。各アプローチには、独自の長所と短所のセットが付属しています。

長所	短所
複数の送信先にデータをレプリケートする方が簡単です。	TSDB は、送信されるデータの量を制御することはできません。
取り込まれたデータの量とデータの信頼性をより詳細に制御できます。	ファイアウォール、VPNs またはロードバランサーは、エージェントへのアクセスを困難にする可能性があります。

すべての測定値をデータベースに書き込むのは非効率であるため、コレクターはデータを事前に集約し、TSDB に定期的な書き込みます。

## 時系列ディメンション

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

一般的なケースは、1 つ以上の追加プロパティをディメンションとして持つ測定に対して 1 つのクエリを発行することです。例えば、場所プロパティとともに温度測定値をクエリできます。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、各シリーズにはディメンションとして一意の場所があります。

時系列のセット内の一意のシリーズを識別するため、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、オプションでラベルがあります。ラベルは、ディメンションを識別するためのキーと値のペアのセットです。ラベルの例は {location=us} または {country=us,state=ma,city=boston} です。時系列のセット内では、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} です。

時系列データのさまざまなソースには、ディメンションがネイティブに保存されているか、データをディメンションに抽出できる一般的なストレージパターンがあります。

通常、TSDBs次元をネイティブにサポートしています。Prometheus はラベル にディメンションを保存します。Graphite や TSDBs などの TSDB では、代わりにタグという用語が使用されます。OpenTSDB

SQL などのテーブルデータベースでは、これらのディメンションは通常、クエリのGROUP BYパラメータです。

## テーブル形式の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 のようなデータベースでは、追加のディメンションは通常、クエリレスポンステーブルの列です。

## 単一ディメンション

例えば、次の例のようなクエリを考えてみましょう。

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

クエリは、3つの列を持つテーブルを返す場合があります。

StartTime	Temp	ロケーション
09:00	24	LGA

StartTime	Temp	ロケーション
09:00	20	BOS
10:00	26	LGA
10:00	22	BOS

テーブル形式は長い形式の時系列で、Tallとも呼ばれます。Locationでは、タイムスタンプが繰り返し、値が繰り返されています。この場合、セット内の2つの時系列はTemp {Location=LGA}およびとして識別されますTemp {Location=BOS}。

セットの個々の時系列は、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として入力された時間型列
- シリーズ名Tempとしての数値型列
- Location=LGA など、ラベルを構築する文字列型Location列の名前と値

### 複数のディメンション

クエリが更新され、複数の文字列列 (などGROUP BY BUCKET(StartTime, 1h), Location, Sensor) で選択およびグループ化された場合、ディメンションが追加されます。

StartTime	Temp	ロケーション	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

StartTime	Temp	ロケーション	センサー
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 Location および に基づく 2つのキーがあります Sensor。データは 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 の複数のアラートにマッピングされる方法では、複数のディメンションはサポートされていません。代わりに、1つのアラートに対して複数の条件として扱われます。

## 複数の値

SQL のようなデータソースの場合、ディメンションとして使用する追加の文字列列の有無にかかわらず、複数の数値列を選択できます AVG(Temperature) AS AvgTemp, MAX(Temperature) AS MaxTemp。例えば、。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可能性があります。現在、複数の値を選択することは、視覚化でのみ使用されるように設計されています。

## ヒストグラムと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。値をバケット (ビンとも呼ばれます) にグループ化します。次に、各バケットに含まれる値の数をカウントします。

ヒストグラム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かかった値の頻度 (カウントなど) を表します。

ヒストグラムは、特定の時間範囲における値の分布のみを調べます。ヒストグラムの問題は、時間の経過とともに分布に傾向や変化が見られないことです。そこでヒートマップが役に立ちます。



## ヒートマップ

ヒートマップは時間の経過に伴うヒストグラムに似ており、各タイムスライスが独自のヒストグラムを表します。バーの高さを頻度の表現として使用する代わりに、セルを使用して、バケット内の値の数に比例したセルを色付けします。

### バケット化されたデータ

次のような多くのデータソースが、時間の経過とともにヒストグラムをサポートします。

- Amazon OpenSearch Service (ヒストグラムバケット集約を使用)
- Prometheus ([ヒストグラム](#)メトリクスタイプとフォーマットをオプションとしてヒートマップに設定)

通常、バケットバインドを表す名前のシリーズを返すか、バインドで昇順にソートされたシリーズを返す任意のデータソースを使用できます。

### 生データと集計データ

通常の時系列データ (事前にバケット化されていない) でヒートマップを使用する場合は、データが時系列バックエンドによって既に集計されていることが多いことに注意してください。ほとんどの時系列クエリは、未加工の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` 制限によるグループを集計関数 (通常は平均) と組み合わせて含めます。

クエリの時間範囲によって異なります。重要な点は、Grafana が実行するヒストグラムバケット化が、すでに集計および平均化されたデータに対して行われる可能性があることを知ることです。より正確なヒートマップを得るには、メトリクスの収集中にバケット化を行うか、データをに保存するか OpenSearch、生データに対してヒストグラムバケット化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でグループを時間単位で削除または下げる (または上げる `maxDataPoints`) と、ヒートマップの精度が向上します。ただし、これにより CPU とメモリに負荷がかかることもあります。データポイントの数が不合理に大きくなると、停止やクラッシュが発生する可能性があります。

## プラグインを使用してワークスペースを拡張する

Grafana プラグインは、新しいデータソースに接続したり、ワークスペースに視覚化やその他の機能を追加したりできます。一般的に、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で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、Grafana が使用できる形式でデータを返します。
- アプリケーションプラグイン – アプリケーション。アプリケーション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。

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にできます。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でプラグインを検索する

Amazon Managed Grafana ワークスペースには、インストールされている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表示する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てい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、大まかに次のカテゴリに分類されます。

- 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されます。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 Core 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 ワークスペースで使用できます。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 Enterprise としてタグ付けされます。これらは、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の詳細については、「」を参照してください [Amazon Managed Grafana Enterprise プラグインへのアクセスの管理](#)。
- コミュニティプラグイン – これらのプラグインは、Grafana Labs AWS など、さまざまなソースの Grafana ワークスペース向けに提供されています。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、これらはデフォルトでは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を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、お客様の判断で行ってください。お客様と 間の責任共有モデルの一環として AWS、これらのサードパーティープラグインのワークスペースにインストールする内容を理解する必要があります。また、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。

## プラグインのサポート

プラグインはさまざまなソースから提供され、サポートはそれらによって異なります。

- **AWS データソースプラグイン** — このプラグインは [AWS](#) によって提供され、[AWS](#) でサポートされています。
- **エンタープライズプラグイン** – Enterprise 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、[AWS](#) によって Amazon Managed Grafana でサポートされています。バグ修正や機能強化 [GitHub](#) については、[AWS](#) で問題を送信したり、AWS または Grafana Labs でチケットを作成したりできます。
- **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、問題やその他のフォーラムを通じて [GitHub](#) 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 <https://github.com/grafana/grafana/issues>。

## プラグインのバージョン

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ある古いバージョンがある場合、古いバージョンは可用性から削除されます。

既にインストールされているプラグインを[更新](#)することもできます。

### Note

インストールされているプラグインのセキュリティ問題を修正する新しいバージョンのプラグインが使用可能になることがあります。重大な問題の場合、Amazon Managed Grafana はワークスペースのプラグインをバージョンに自動的に更新し、修正する場合があります。

## プラグインカタログでプラグインを管理する

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。インストールできるのは、ワークスペース内のプラグインカタログにリストされているプラグインのみです。

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](#)と、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、[プラグイン管理が有効になっている必要があります](#)。
- 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。
- Enterprise プラグインをインストールして使用するには、まず [Enterprise 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を表示するには

- Amazon Managed Grafana ワークスペースにサインインします。
- 左側のメニューから、**管理** を選択し、次に **プラグイン** を選択します。これにより、プラグインカタログが開きます。
- デフォルトでは、プラグインカタログにはインストールされている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ています。

## プラグインのインストールまたは削除

### Note

前のセクションの前提条件を満たす必要があります。満たす場合、プラグインを変更するアクセス許可はありません。

Grafana プラグインをインストールまたは削除するには

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されている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ています。
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 50 個のプラグインをインストールできます (デフォルトの Core プラグイン以外)。

## プラグインを更新する

既存の Grafana プラグインを更新するには

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、管理 を選択し、次にプラグイン を選択します。これにより、プラグインカタログが開き、インストールされている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。その場合は、プラグインを更新するオプションを選択し、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アクセス許可がないというメモ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](#)である必要があります。

## 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 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を提供します。AWS データソースプラグインは、デフォルトで新しいワークスペースに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 を選択します。

AWS データソースプラグインインターフェイスには、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ものを選択します。例えば、Amazon Managed Service for Prometheus を選択します。これにより、データソースタブに移動し、サービスが選択されます。

- 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。

- 一部のサービスでは、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 (この場合は 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。例えば、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、ダッシュボードで使用するか、Amazon Managed Grafana でモニタリングする準備ができています。

- 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。

(オプション) 設定に移動を選択して、そのデータソースの設定を表示および編集できます。

**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は、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と共にインストールされるバージョンは、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 サービスがサポートされる場合があります。

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、[Changelog](#) を参照してください。



プラグインの新しいバージョンに更新するには、の標準手順に従ってください[プラグインを更新する](#)。

#### Note

データソース AWS 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、リソースにアクセス AWS するためのアクセス許可が必要です。そのための最も簡単な方法は、Amazon Managed Grafana がアクセス許可を管理できるようにすることです。データソースのサービス管理アクセス許可を設定する方法については、「」を参照してください[データソースと通知チャネルのアクセス許可を管理する](#)。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 によってアクセス許可が自動的に管理されない AWS リソースのサポートが追加される可能性があります。このような場合は、アクセス許可を自分で追加する必要があります。例えば、が最新バージョンのプラグイン (バージョン 1.9.0) AWS IoT TwinMaker 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のアクセス許可の詳細については、[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、[AWS IoT TwinMaker データソースに接続する](#)セクションには、へのアクセス許可を Amazon Managed Grafana に付与する方法の詳細が含まれています AWS IoT TwinMaker。

## 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。データソースは、Grafana でクエリしてダッシュボードの構築などの操作を実行できるストレージバックエンドです。各データソースには、特定のデータソースが公開する機能に合わせてカスタマイズされた特定のクエリエディタがあります。

各データソースのクエリ言語と機能は異なります。複数のデータソースのデータを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、多数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、そのデータソースに対応するプラグインを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をインストールする必要がある場合もあります。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 コミュニティはワークスペース内の [プラグインカタログ](#) で使用できる新しいデータソースを追加することがあります。

### 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 [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](#) を提供します。

## 特殊なデータソース

Amazon Managed Grafana には、次の3つの特別なデータソースが含まれています。

- Grafana (Grafana の以前のバージョンでは TestDB と呼ばれています – この組み込みデータソースを使用して、ランダムなウォークデータを生成したり、ファイルを一覧表示したりします。これは、視覚化のテストや実験の実行に役立ちます。
- 混合 – これを使用して、同じパネル内の複数のデータソースをクエリします。このデータソースを使用すると、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。最初のクエリでは、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。

混合データソースを使用するように既存のクエリを変更することはできません。

- ダッシュボード — これを使用して、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します。

### 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](#)
- [組み込み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、プライマリアカウントで AWS データソースを表示できるようにします。

例えば、組織内の 1 つのアカウントを Amazon Managed Grafana 管理アカウント として使用し、こ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を許可できます。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。これにより、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ます。このポリシーは、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、[「とは AWS Organizations」](#) 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 が組織全体のデータソースにアクセスするための IAM ポリシーを管理できるようにするには、組織の管理アカウント AWS CloudFormation StackSets で 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 のシナリオである可能性があります。

このシナリオでは、、、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AWSGrafanaAccountAdministrator、Amazon Managed Grafana によって AWSSSOMemberAccountAdministrator および Organizations が自動的に有効になります。詳細については、「[IAM Identity Center を使用して、単一の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します。](#)」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に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に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で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する場合は、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、「[カスタマー管理アクセス許可](#)」を参照してください。

サービス管理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の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。オプションで、管理アカウントで IAM Identity Center 統合アプリケーションをメンバーアカウントで有効にすることもできます。これらのアプリケーションを有効にするには、IAM Identity Center 統合アプリケーションセクションの IAM Identity Center 設定ページ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。

組織の管理者でない場合は、Organizations の管理者に連絡して、IAM Identity Center を有効にするようにリクエストする必要があります。IAM Identity Center を有効にしたら、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で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する場合は、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、「[カスタマー管理アクセス許可](#)」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するに

は、AWSGrafanaAccountAdministrator、AWSSSOMemberAccountAdministratorおよびAWSSSODirectoryAdministratorポリシー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、「[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。

## 組み込みデータソースに接続する

以下のデータソースは、すべての Amazon Managed Grafana ワークスペースで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、正しい Grafana 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追加のデータソース用のプラグインをインストールできます。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てワークスペースを拡張する](#)」を参照してください。

### トピック

- [Alertmanager データソースに接続する](#)
- [Amazon CloudWatch データソースに接続する](#)
- [Amazon OpenSearch Service データソースに接続する](#)
- [AWS IoT SiteWise データソースに接続する](#)
- [AWS IoT TwinMaker データソースに接続する](#)
- [Amazon Managed Service for Prometheus とオープンソースの Prometheus データソースに接続する](#)

- [Amazon Timestream データソースに接続する](#)
- [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 データソースに接続する](#)
- [Graphite データソースに接続する](#)
- [Google Cloud Monitoring データソースに接続する](#)
- [InfluxDB データソースに接続する](#)
- [Jaeger データソースに接続する](#)
- [Loki データソースに接続する](#)
- [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 データソースに接続する](#)
- [テスト用の TestData データソースを設定する](#)
- [Zipkin データソースに接続する](#)

## Alertmanager データソースに接続する

Grafana には Prometheus Alertmanager のサポートが組み込まれています。Grafana アラートを設定したら、Grafana アラート UI を使用して、無音、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オプションを使用すると、Grafana と設定された任意の Alertmanager データソース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

Alertmanager の [Prometheus](#)、[Cortex](#)、[Grafana Mimir](#) の実装がサポートされています。データソース設定ページで実装を指定できます。Prometheus の連絡先と通知ポリシーは、HTTP API による設定の更新をサポートしていないため、Grafana アラート UI では読み取り専用です。

### Alertmanager 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

Alertmanager を設定するには、次の前提条件を完了している必要があります。

- メトリクスが取り込まれ、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た Prometheus インスタンス。ワークスペースの URL が必要になります。
- Amazon Managed Grafana が Alertmanager 実装からアラート、アラートグループ、無音、およびコンタクトポイントへの読み取りアクセス権を持つように定義されたアクセス許可。

Alertmanager データソースを設定するには

1. Grafana コンソールの Grafana メニューで、設定 の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. 新しいデータソースに次の情報を入力します。
  - 名前 には、データソースに選択した名前を指定します。
  - 実装 で、アラートマネージャーの実装を選択します。Prometheus、Mimir、または Cortex のいずれかです。
  - HTTP で URL に Alertmanager URL を指定します。Prometheus の場合、これはワークスペース URL で、alertmanager が追加されています。例えば `https://myprometheus/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager` です。
  - 認証 で、Alertmanager の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、データソースの設定を完了します。

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

## Amazon CloudWatch データソースに接続する

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 を AWS データソース CloudWatch として追加できます。この機能により、既存の CloudWatch アカウントを検出してへのアクセスに必要な認証情報の設定を管理することで、データソース CloudWatch としてを簡単に追加できます CloudWatch。この方法を使用して認証を設定し、データソース CloudWatch としてを追加できます。または、セルフマネージド型の



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セットアップすることもできます。

## トピック

- [AWS データソース設定を使用してデータソース CloudWatch として を追加する](#)
- [データソース CloudWatch として を手動で追加する](#)
- [クエリエディタの使用](#)
- [厳選されたダッシュボード](#)
- [テンプレートクエリ](#)
- [ec2\\_instance\\_attribute の使用例](#)
- [JSON フォーマット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データソース CloudWatch 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 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、 をデータソース CloudWatch として追加します。

## AWS データソース設定を使用してデータソース CloudWatch として を追加するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、

「[AWS データソースの Amazon Managed Grafana の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オンにし、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. データソースタブを再度選択します。
8. Amazon CloudWatch 行の 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、 を選択しますCloudWatch。
11. CloudWatch 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ます。
12. 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。

データソース CloudWatch として を手動で追加する

CloudWatch 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) アイコンにカーソルを合わせ、データソース を選択します。
2. [データソースを追加する] を選択する。
3. CloudWatch データソースを選択します。必要に応じて、**CloudWatch**検索ボックスに入力して検索に役立てることができます。

CloudWatch 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、パネルとクエリでデータソースを表示する方法です。
Default	新しいパネル用に事前に選択するデータソースを指定します。

名前	説明
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 CloudWatch の名前空間を指定します。カンマで区切られた複数の名前空間を含めることができます。
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ときにログ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。 <a href="#">X-Ray データソース</a> が既に設定されている必要があります。

## 認証

Amazon Managed Grafana と 間の認証を有効にするには CloudWatch、Amazon Managed Grafana コンソールを使用して、必要なポリシーとアクセス許可をすばやく作成できます。または、自己管理型の Grafana サーバーと同じ方法の一部を使用して認証を手動でセットアップ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ポリシーをすばやくセットアップするには、「」のステップに従います [AWS データソース設定を使用してデータソース CloudWatch としてを追加する](#)。

アクセス許可を手動で設定するには、次のセクションのメソッドのいずれかを使用します。

## AWS 認証情報

3 つの異なる認証方法を使用できます。

- 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 — に対応します AWS SDK for Go StaticProvider。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。このメソッドにはフォールバックがなく、提供されたキーペアが機能しない場合に失敗します。

## IAM ロール

現在、へのアクセス CloudWatch はすべて、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 サーバーがで実行されている場合 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 では、CloudWatch メトリクスと EC2 タグ、インスタンス、リージョンを読み取るために、IAM を通じて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 ロールにアタッチし、組み込みの Grafana サポートを使用してロールを引き受けることができます。

次のコード例は、最小限のポリシーを示しています。

```
{
  "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

Assume Role ARN フィールドでは、引き受ける IAM ロールがあれば指定できます。これを空白のままにすると、提供された認証情報が直接使用され、関連するロールまたはユーザーには必要

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` コールを実行します。

## クエリエディタの使用

Amazon Managed Grafana の CloudWatch データソースは、 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 CloudWatch。クエリエディタは、メトリクス検索と CloudWatch ログの 2 つの異なるクエリモードをサポートしています。

メトリクスのクエリエディタモードでは、 CloudWatch APIを使用して にアップロードされたメトリクスを検索します CloudWatch。ログの モードは、 CloudWatch ログAPIsを使用してログレコードを検索します。各モードには独自の特殊なクエリエディタがあります。エディタの上部にあるAPIクエリモードスイッチを使用して、クエリを実行する 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。メトリクスクエリオプションは、Metrics Insights CloudWatch を使用してデータをクエリします。

## 一般的なクエリエディタフィールド

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する 3 つのフィールドがあります。

## 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。このフィールドを使用して、選択した ID を指定します。ID には数字、文字、アンダースコアを含めることができ、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 はパターン を使用して ID を生成します `query[refId of the current query row]`。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。

## [Period] (期間)

期間は、特定の CloudWatch 統計に関連する時間の長さです。期間は秒数で定義されます。有効な値には、1、5、10、30、または 60 の倍数が含まれます。期間フィールドを空白のままにするか、に設定すると auto、時間範囲と CloudWatch 保持ポリシーに基づいて自動的に計算されます。使用される式は `time range in seconds / 2000`、保持期間に基づいて期間を削除した後、事前定義された期間 [60、300、900、3600、21600、86400] の配列で次に大きい値に移行します。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
<code>{{region}}</code>	リージョンを返します。	us-east-1
<code>{{period}}</code>	ピリオドを返します。	3000
<code>{{metric}}</code>	メトリクスを返します。	CPUUtilization
<code>{{label}}</code>	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
<code>{{namespace}}</code>	名前空間を返します (メトリクス検索のみ)。	AWS/EC2
<code>{{stat}}</code>	統計を返します (メトリクス検索のみ)。	Average
<code>{{[dimension name]}}</code>	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。Match Exact が有効になっている場合は、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「検索式の構文」](#)を参照してください。

Match Exact がオフになっている場合は、フィルタリングするディメンションをいくつでも指定できます。フィルター条件に一致するメトリクスが最大 100 個返されます。

### 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。例えば、自動スケーリング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、新しいインスタンスを追跡しなくても自動的にグラフに表示されますIDs。この機能は現在、最大 100 個のメトリクスの取得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ように自動的に構築された検索式を表示できます。

デフォルトでは、検索式は、クエリされた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ます。つまり、この例では、名前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InstanceIdます。

他のディメンションが定義されているメトリクスを含めるには、完全一致をオフにします。Match Exact 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も作成されます。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。

### 複数值テンプレート変数

複数值のテンプレート変数に基づいて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、値の長いリストがある場合、クエリが失敗する可能性があります。特定のディメンション名に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。

複数值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せん。Region、Namespaceまたはの複数值テンプレート変数Metric Nameの使用はサポートされていません。

## Metric Math 式

新しい時系列メトリクスは、数学関数を使用して CloudWatch メトリクス上で操作することで作成できます。算術演算子、単項減算、およびその他の関数がサポートされており、CloudWatch メトリクスに適用できます。CloudWatch Metric Math 関数の詳細については、[「Metric Math 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、raw 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 フィールドでその ID に算術演算を適用できます。

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

### [Period] (期間)

期間は、特定の Amazon CloudWatch 統計に関連付けられている時間の長さです。期間は秒数で定義されます。有効な値には、1、5、10、30、または 60 の倍数が含まれます。

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、時間範囲に基づいて自動的に計算されます。使用される式は `time range in seconds / 2000`、定義済みのピリオドの配列で次に大きい値に移動します [60, 300, 900, 3600, 21600, 86400]。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 コンソールへのダイープリンク

パネルで時系列を選択すると、CloudWatch コンソール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。このリンクを選択すると、CloudWatch コンソールに移動する新しいタブが開き、そのクエリのすべてのメトリクスが表示されます。現在 CloudWatch コンソールにサインインしていない場合、リンクからサインインページに転送されます。提供されたリンクは有効です 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。

メトリクスクエリオプションを使用して Metrics Insights CloudWatch 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 の機能はすべてので利用できます AWS アカウント。機能は変更される可能性があります。

CloudWatch Metrics Insights のデータをクエリするには、Metric クエリエディタで `metric query` モードを選択します。

CloudWatch Metrics Insights は、メトリクスを大規模に SQL クエリするために使用できる強力な高性能クエリエンジンです。これは、すべての CloudWatch メトリクス内の傾向や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。のダイアレクトを使用します SQL。Metrics Insights クエリ構文の詳細については、「」を参照してください [クエリ構文とキーワード](#)。

## クエリ構文とキーワード

CloudWatch Metrics Insights は のダイアレクトを使用します SQL。次の例は、クエリ構文を示しています。

```
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が区別されます。識別子には、メトリクス、名前空間、ディメンションの名前が含まれます。

次の表に、クエリキーワードとその説明を示します。

キーワード	説明
FUNCTION	必須。使用する集計関数を指定し、クエリするメトリクスの名前も指定します。有効な値は、AVG、COUNT、MAX、MIN、SUM です。
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テーブル関数を指定できます。名前空間の例には、AWS/EC2とがありますAWS/Lambda。

キー ワード	説明
SCHEMA	(オプション) 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	(オプション) 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	(オプション) 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	(オプション) 返される時系列の順序を指定します。オプションは ASCと ですDESC。
LIMIT	(オプション) 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名前空間のCPUUtilizationメトリクスのみを一致させます。

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

に報告されたCPUUtilizationメトリクスのみを1つのディメンション CloudWatch と一致させますInstanceId。

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

CloudWatch からに報告されたRequestCountメトリクスのみを、正確に2つのディメンションであるLoadBalancerとAWS/ApplicationELBで一致させますAvailabilityZone。

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二重引用符自体 (など`Bytes"Input"`) が含まれている場合は、のように、その二重引用符をバックスラッシュでエスケープする必要があります`SELECT AVG("Bytes\"Input \")`。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

クエリはモードまたは Builder モードで作成できますCode。

### Builder モードでクエリを作成するには

1. 前の表の情報を使用して、メトリクス名前空間、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. これらのオプションごとに、使用可能なオプションのリストから選択します。

### Code モードでクエリを作成するには

1. コードエディタにクエリを書き込みます。
2. クエリを実行するには、コードエディタでクエリの実行を選択します。

モードでクエリを作成するには`builder :`

- 上記の表の情報を使用して、メトリクス名前空間、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。
- これらのオプションごとに、使用可能なオプションのリストから選択します。

Grafana は、選択した内容に基づいてSQLクエリを自動的に作成します。

モードでクエリを作成するにはcode :

- コードエディタにクエリを書き込みます。
- クエリを実行するには、コードエディタでクエリの実行を選択します。

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、ラベル値を提案するオートコンプリート機能が組み込まれています。候補は、スペース、カンマ、またはドル記号を入力すると表示されます。キーボードの組み合わせを使用することもできますCTRL+Space。

コードエディタはクエリを自動入力できます。ただし、コードエディタでテンプレート変数を使用すると、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、機能を追加または変更する場合があります。

このセクションでは、直接コピーして使用したり、クエリエディタでコピーして変更したりできる便利な CloudWatch Metrics Insights クエリの例を示します。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、[Metrics] (メトリクス) ビューで [Add query] (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 例

インスタンスメトリクスあたりのCPU使用率の表示

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

```
GROUP BY InstanceId
```

フリート全体の平均CPU使用率を表示する

```
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する

```
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最大数順に降順で表示します。

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

この場合、CloudWatch エージェントはアプリケーションごとにCPUUtilization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"
```

上位 10 個の Lambda 関数の平均実行時間を、最大値順に降順で表示します。



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

Lambda 実行時間の最大、平均、最小を表示する

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

Application Load Balancer の例

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

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

アクティブな同時TCP接続数を含むメトリクスを表示する

```
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 個を降順で表示する

```
SELECT AVG(VolumeWriteBytes)
FROM "AWS/EBS"
GROUP BY VolumeId
ORDER BY MAX() DESC
LIMIT 10
```

Amazon EBSボリュームの平均書き込み時間を表示する

```
SELECT AVG(VolumeTotalWriteTime)
```

```
FROM "AWS/EBS"
```

### Amazon EBSボリュームの平均アイドル時間を表示する

```
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ボリュームの平均書き込みバイト数を表示する

```
SELECT AVG(VolumeWriteBytes)
FROM "AWS/EBS"
```

### Amazon Simple Storage Service での例

#### バケット名ごとに平均レイテンシーグループを表示する

```
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する

```
SELECT AVG(NumberOfObjects)
FROM "AWS/S3"
GROUP BY BucketName
```

### Amazon Simple Notification Service での例

mazon-simple-notificaation-service-examples

```
SELECT AVG(NumberOfMessagesPublished)
FROM "AWS/SNS"
```

トピック名ごとに失敗したメッセージの平均数を表示する

```
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 件を表示する

```
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。

- クエリできるのは、最新の 3 時間分のデータのみです。
- 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、FROM、および WHERE 句が 10,000 を超えるメトリクスと一致する場合、見つかったこれらのメトリクスの最初の 10,000 個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。つまり、クエリが 500 を超えるメトリクスを処理している場合、すべてのメトリクスがクエリ結果で返されるわけではありません。ORDER BY 句を使用すると、処理されるすべてのメトリクスがソートされ、ORDER BY 句に従って最大値または最小値を持つ 500 が返されます。ORDER BY 句を含めない場合、返される 500 個の一致するメトリクスを制御することはできません。
- 各 GetMetricData オペレーションにはクエリを 1 つだけ含めることができますが、1 つのダッシュボードにそれぞれクエリを含む複数のウィジェットを含めることができます。

## Amazon CloudWatch Logs クエリエディタの使用

CloudWatch ログをクエリするには、クエリするリージョンと最大 20 のロググループを選択します。メイン入力領域を使用してクエリを書き込みます。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する必要があります。

クエリ入力フィールドの右側には、クエリで CloudWatch Logs Insights コンソールを開く CloudWatch Logs Insights リンクがあります。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

他の複数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、「[テンプレートと変数](#)」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場合は、クエリエディタの横にある CloudWatch Logs Insights ボタンを選択します。現在 CloudWatch コンソールにサインインしていない場合、リンクからサインインページに転送されます。提供されたリンクは、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、stats コマンドの使用などによって数値データを返す可能性があるため、アラートは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。

### 厳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

事前設定されたダッシュボードをインポートするには、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ます。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には、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。保存しないと、ダッシュボードの新しいバージョンがリリースされたときにダッシュボードが上書きされるためです。

### テンプレートクエリ

サーバー、アプリケーション、センサー名などの詳細をメトリクスクエリでハードコーディングする代わりに、変数を代わりに使用できます。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。

テンプレート機能とさまざまなタイプのテンプレート変数の概要[テンプレート](#)については、「」を参照してください。

### クエリ変数

CloudWatch データソースには、変数編集ビューの Query 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 region、namespaces、などのモノで埋めることができます metric namesdimension keys/values。

の代わりに region、クエリのデータソースで設定されたデフォルトのリージョンを使用する default ように を指定できます。

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()	が CloudWatch 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。(リージョンを指定するか、カスタムメトリクス"default" に使用します。)
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。

名前	説明
<code>dimension_values(region, namespace, metric, dimension_key, [filters])</code>	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、ディメンションを使用してより具体的な結果 filters を取得することもできます。
<code>ebs_volume_ids(region, instance_id)</code>	指定された region、instance_id に一致する EBS ボリュームのリストを返します。
<code>ec2_instance_attribute(region, attribute_name, filters)</code>	指定された region、attribute_name に一致する属性のリストを返します。
<code>resource_arns(region, resource_type, tags)</code>	指定された region、resource_type および tags に一致する ARNs のリストを返します。
<code>statistics()</code>	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、「[メトリクスを発行する AWS のサービス](#)」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。

Query	サービス
<code>dimension_values(us-east-1,AWS/ELB,RequestCount,LoadBalancerName)</code>	Elastic Load Balancing

Query	サービス
<code>dimension_values(us-east-1,AWS/ElastiCache,CPUUtilization,CacheClusterId)</code>	Amazon ElastiCache
<code>dimension_values(us-east-1,AWS/Redshift,CPUUtilization,ClusterIdentifier)</code>	Amazon Redshift
<code>dimension_values(us-east-1,AWS/RDS,CPUUtilization,DBInstanceIdentifier)</code>	Amazon RDS
<code>dimension_values(us-east-1,AWS/S3,BucketSizeBytes,BucketName)</code>	Amazon Simple Storage Service (Amazon S3)
<code>dimension_values(us-east-1,CWAgent,disk_used_percent,device,{"InstanceId":"\$instance_id"})</code>	CloudWatch エージェント
<code>resource_arns(eu-west-1,elasticloadbalancing:loadbalancer,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})</code>	Elastic Load Balancing
<code>resource_arns(eu-west-1,ec2:instance,{"elasticbeanstalk:environment-name":["myApp-dev","myApp-prod"]})</code>	Amazon EC2



## ec2\_instance\_attribute の使用例

### JSON フィルター

ec2\_instance\_attribute クエリは filtersJSON形式になります。の定義済みフィルターを指定できますec2:DescribeInstances。実際のフィルタリングは AWS Grafana ではなく で実行されることに注意してください。

次のコード例は、フィルター構文を示しています。

```
{ filter_name1: [ filter_value1 ], filter_name2: [ filter_value2 ] }
```

次の例は、 ec2\_instance\_attribute()クエリを示しています。

```
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すことができる属性は 1 つだけです。任意のフラット属性を選択できます (つまり、属性に単一の値があり、オブジェクトまたは配列でない場合)。次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

タグ名の前に を付けることで、タグを選択できますTags。

次の例は、 `ec2_instance_attribute()`クエリを示しています。

```
ec2_instance_attribute(us - east - 1, Tags.Name, { 'tag:Team': ['sysops'] });
```

## JSON フォーマット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入れ、Grafana はテンプレート変数の への変換をサポートしていますJSON。

の場合 `env = 'production', 'staging'`、次のクエリは、Environmentタグが `production`または ARNsであるEC2インスタンスを返します `staging`。

```
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呼び出し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 クエリを介して取り込まれ、アーカイブされ、分析されたデータの量に基づいています。詳細については、[「Amazon CloudWatch の料金」](#)を参照してください。

クエリエディタでディメンションを選択するたびに、Grafana はListMetricsリクエストを発行します。クエリエディタでクエリを変更するたびに、1つの新しいリクエストがに発行 GetMetricData されます。

API データサンプルを取得する リクエストは、GetMetricDataオペレーションを使用します。このオペレーションにより、CloudWatch メトリクス数学の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、Match Exact オプションをオフにしたりする場合の検索式の自動生成もサポートしています。GetMetricData オペレーションには料金が発生します。詳細については、[「Amazon CloudWatch の料金」](#)を参照してください。

### Service Quotas

AWS は、AWS アカウント内のリソース、オペレーション、およびアイテム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、アカウントごと、および制限に達した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にわたるアプリケーションをモニタリングおよびトラブルシューティングできます。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、まずで有効にしてから CloudWatch、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 ワークスペースが内で実行されている場合はVPC、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 [CloudWatch](#)」の「[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 ワークスペースが内で実行されている場合VPC、ワークスペースがインターネットAPIsで を呼び出すことを許可するNATゲートウェイも必要です。

#### Note

アクセスしようとしているアカウントの CloudWatch データを読み取るためのアクセスIAM 許可も必要です。

## Amazon OpenSearch Service 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、このデータソースで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てワークスペースを拡張する](#)」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 クラスターとレガシー Elasticsearch クラスターを実行する OpenSearch サービスドメインをサポートします。

AWS データソース設定オプションは、既存の OpenSearch サービスアカウントを検出することで、データソースとしての OpenSearch サービスの追加を簡素化し、へのアクセスに必要な認証情報の設定を管理します OpenSearch。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 サービスを追加するか、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 サービスデータソースは、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 サービスデータソースを使用して、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、さまざまなタイプのシンプルまたは複雑な 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でグラフに注釈を付けることもできます OpenSearch。

### トピック

- [AWS データソース設定を使用して OpenSearch サービス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 サービス設定](#)
- [Amazon OpenSearch Service データソースの使用](#)
- [Amazon OpenSearch Service Serverless](#)
- [トレースのサポート](#)

## AWS データソース設定を使用して OpenSearch サービス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サービス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 OpenSearch 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、OpenSearch Service をデータソースとして追加します。

AWS データソース設定を使用して OpenSearch サービスをデータソースとして追加するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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オンにし、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 行の 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 が検索して OpenSearch サービスリソースを検出するリージョンを選択し、追加するアカウントと OpenSearch サービスドメインを選択し、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には

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、**OpenSearch**検索ボックスに入力して検索に役立てることができます。
3. データを検索するリージョンを選択します。
4. [データソースを追加する] を選択する。

### 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ロールがないことを意味します。

## OpenSearch サービス設定

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、新しいパネルに対して事前に選択されることを意味します。
Url	OpenSearch サービスドメインのエンドポイント。エンドポイントの形式は <code>https://us-east-1.es.amazonaws.com search-my-domain</code> です。
Access Mode	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できる必要があります。

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。他に何も記載されていない場合は、サーバーが推奨される方法です。

## 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対して行われ、リクエストはデータソースに転送され、Cross-Origin Resource Sharing (CORS) の要件を回避します。このアクセスモードを選択した場合、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、のデフォルトを指定time field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/Elasticsearch バージョン

バージョンドロップダウンメニューで、OpenSearch またはレガシーの 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。Elasticsearch でサポートされているバージョンは 2.0+、5.0+、5.6+、6.0+、および 7.0+ です。値は、バージョン 5.6 以降で 6.0 より小さい 5.6+ を意味します。値は、バージョン 6.0 以降、7.0 未満 6.0+ を意味します。最後に、7.0+ はバージョン 7.0 以降で、8.0 より前のことを意味します。

## 最小時間間隔

時間間隔による自動グループの下限。書き込み頻度に設定することをお勧めします。例えば、データが 1 分ごとに書き込まれる 1m 場合などです。このオプションは、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、(1 分) や (30s30 秒) など、1m 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。次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

Message field name との 2 つのパラメータは Level field name、オプションでデータソース設定ページから設定できます。このページでは、でログを視覚化するときログメッセージとログレベルに使用するフィールドを決定します [探索](#)。

例えば、OpenSearch サービスへのログの配信に 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 : message
- 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 – リンクが外部の場合は、リンク URL 全体を入力します。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。いずれの場合も、`${__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。リンクが内部リンクの場合、データソースセクタを使用すると、ターゲットデータソースを選択できます。トレースデータソースのみがサポートされています。

## Amazon OpenSearch Service 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。右側のプラスとマイナスの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グループバイ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、メトリクスまたはグループをオプション別に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、のクエリと視覚化の機能がシンプルでありながら強力になります OpenSearch。PPL を使用すると、長い OpenSearch ドメイン固有言語 (DSL) ステートメントを作成したり、JSON オブジェクトを使用してクエリを記述したりすることなく、データを探索して検索できます。PPL を使用すると、UNIX パイプに似たパイプで区切られた一連のコマンドとしてクエリを記述できます。

例として、次のサンプル DSL クエリを使用します。

```
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}}}}}}}>
```

前述の 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。

## シリーズの命名パターン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
<code>{{term fieldname}}</code>	Group By という用語の値に置き換えました。
<code>{{metric}}</code>	メトリクス名 (例: 平均、最小、最大)。
<code>{{field}}</code>	をメトリクスフィールド名に置き換えました。

## パイプラインメトリクス

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。例えば、移動平均 や派生 などです。OpenSearch パイプラインメトリクスには、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

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## クエリ変数

OpenSearch サービスデータソースは、Query 変数の Query 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されます。

Query	説明
<code>{"find": "fields", "type": "keyword"}</code>	インデックスタイプのフィールド名のリストを返しますkeyword。

Query	説明
<pre>{"find": "terms", "field": "@hostname", "size": 1000}</pre>	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、現在のダッシュボードの時間範囲をクエリの時間範囲として使用します。
<pre>{"find": "terms", "field": "@hostname", "query": '&lt;.lucene query&gt;'}</pre>	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、現在のダッシュボードの時間範囲をクエリの時間範囲として使用します。

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、クエリで `size` プロパティを設定します。クエリ内で他の変数を使用できます。次のコード例は、という名前の変数のクエリ定義を示しています `$host`。

```
{"find": "terms", "field": "@hostname", "query": "@source:$source"}
```

前の例では、クエリ定義 `$source` 内で という名前の別の変数を使用します。 `$source` 変数の現在の値であるドロップダウンリストを使用して変更するたびに、 `$host` 変数の更新が開始されます。更新後、 `$host` 変数には によってフィルタリングされた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です。

これらのクエリは、デフォルトでは語順で返されます (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語句のリスト (上位 N 個の値リスト) を生成するには、 `orderBy` プロパティを追加します `doc_count`。これにより、降順のソートが自動的に選択されます。 `doc_count (ボトム N リスト) asc` を使用するには、 `order` を設定しますが `order: "asc"`、ドキュメント数のエラーが増加するため推奨されません。ドキュメント数順に語句を保持するには、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

次の 2 つの構文があります。

- `$<varname>` 例: `@hostname:$hostname`
- `[[varname]]` 例: `@hostname:[[hostname]]`

2つの方法があるのはなぜですか？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。複数值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、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 `$hostname`。また、フィールド入力による用語グループの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、グラフの上に豊富な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、任意の OpenSearch インデックスに注釈イベントをクエリできます。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ことも、Lucene クエリを指定することもできます。
Tim	時間フィールドの名前。は日付フィールドである必要があります。
Tim End	時間終了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 文字列を指定できません)。

## ログのクエリ

からのログデータのクエリと表示 OpenSearch については、Explore を参照してください。ログを表示するには、OpenSearch サービス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、「[探索](#)」を参照してください。

## ログクエリ

結果が返されると、ログパネルにはログ行のリストと棒グラフが表示されます。X 軸には時刻が表示され、Y 軸には頻度またはカウントが表示されます。

### ログメッセージのフィルタリング

オプションで、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します。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 サポートは、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できます。

OpenSearch サービスデータソースを使用して、Amazon Managed Grafana で Amazon OpenSearch Service Serverless データにアクセスできます。データへのアクセスは、データアクセスポリシーによって制御されます。次の例は、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を示しています。`collection_name`、および `index_name` ユースケースに適した `principal_arn` 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"
  }
]
```

]

## トレースのサポート

OpenSearch プラグインは、トレースのリストをテーブル形式で表示すること、およびトレーススパンのタイムラインを示すトレースビューで1つのトレースを表示することをサポートしています。

### Note

OpenSearch トレースのクエリは、Lucene クエリを使用するのみ使用できます。  
トレースサポートは、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Tracesで Lucene クエリタイプを使用します。必要に応じて、テーブルの視覚化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、クエリを使用し `traceid: {traceId}`、必要に応じてトレースの視覚化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、このデータソースで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てワークスペースを拡張する](#)」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 を追加できます。この機能により、既存の AWS IoT SiteWise アカウントを検出してへのアクセスに必要な認証情報の設定を管理することで、データソース AWS IoT SiteWise を簡単に追加できます AWS IoT SiteWise。この方法を使用して認証を設定し、データソース 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](#)
- [データソースの手動追加 AWS IoT SiteWise](#)
- [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、 をデータソースとして追加 AWS IoT SiteWise 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. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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の有効化 を選択します。
6. データソースタブを再度選択し、AWS IoT SiteWise 行の 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。



## データソースの手動追加 AWS IoT SiteWise

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) アイコンを一時停止し、データソース を選択します。
2. [データソースを追加する] を選択する。
3. AWS IoT SiteWise データソースを選択します。必要に応じて、**SiteWise**検索ボックスに入力して検索に役立てることができます。

### AWS IoT SiteWise 設定

名前	説明
[Name] (名前)	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表示する方法です。
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されます (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)。デフォルトは空白のままにします。
ロール Arn を引き受ける	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。

## 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。

## AWS IoT TwinMaker データソースに接続する

この機能は Amazon Managed Grafana のプレビューリリース AWS IoT TwinMaker であり、変更される可能性があります。

### 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、このデータソースで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てワークスペースを拡張する](#)」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 AWS IoT TwinMaker である Grafana ワークスペースのアプリケーションとデータソースとして追加できます。を使用すると AWS IoT TwinMaker、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あり、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るようにします。AWS IoT TwinMaker for Grafana には、デジタルツインデータに接続するためのカスタムパネル、ダッシュボードテンプレート、データソースが用意されています。

AWS IoT TwinMaker プラグインを有効にするには

をデータソース AWS IoT TwinMaker として有効にするには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
2. 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、作業しているワークスペースを選択します。これにより、ワークスペースの Grafana コンソールが開きます。
3. Grafana コンソールで、設定 (歯車) アイコンを選択します。
4. Config タブで Enable を選択します。

## 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 IoT TwinMakerから にアクセスできることを確認してください AWS アカウント。

にアクセスするためのアクセス許可をワークスペースの IAM ロールに追加する方法については AWS IoT TwinMaker、「」を参照してください[ワークスペースユーザーロール AWS IoT TwinMaker へのアクセス許可の追加](#)。

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エディタ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、データソース を選択します。
3. [データソースを追加する] を選択する。
4. AWS IoT TwinMaker データソースを選択します。必要に応じて、**TwinMaker**検索ボックスに入力して検索に役立てることができます。
5. 接続の詳細ページが開きます。の設定のステップに従います[AWS IoT TwinMaker 接続の詳細設定](#)。

### ワークスペースユーザーロール AWS IoT TwinMaker への のアクセス許可の追加

のアクセス許可 AWS IoT TwinMaker をワークスペースユーザーロールに追加するには、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の詳細については、「」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](#)。

### AWS IoT TwinMaker 接続の詳細設定

#### 接続の詳細設定を構成する

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 )。

- クエリするデフォルトリージョンを選択します。
- TwinMaker 設定 にワークスペース名を入力します AWS IoT TwinMaker。

## 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、「[のAWS IoT TwinMakerデータソース](#)」を参照してください GitHub。

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

- IAM コンソール (<https://console.aws.amazon.com/iam/>) にサインインします。
- 概要で 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
```

- 次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 >"
  }
}
```

- 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、リソース行にロール Amazon リソースネーム (ARNsのリストを追加します。
- IAM コンソールでダッシュボードロールを見つけます。SceneViewer ポリシーと、オプションでVideoPlayerポリシーが必要です。
- 信頼関係タブを選択します。
- [Edit trust relationship (信頼関係の編集)] を選択します。
- AMG ##### Arn WorkspaceRoleArn**に置き換えて、次のポリシーを入力します。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "AMGWorkspaceRoleARN"
    },
    "Action": "sts:AssumeRole"
  }
]
```

### 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最小限の AWS IoT TwinMaker ポリシーです。AWS IoT TwinMaker ワークスペース ARN と ID の値、および Amazon S3 バケット ARN の値は、独自のリソースに基づいて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 とオープンソースの Prometheus データソースに接続する

Amazon Managed Grafana では、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をデータソースとして使用すること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 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 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ことを簡素化し、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、統合 Grafana アラートインターフェイスで ごとに表示できます [Alertmanager データソースの設定](#)。

### 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 データソースを手動で追加する](#)
- [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](#)
- [例を設定する](#)

### 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、Amazon Managed Service for Prometheus をデータソースとして追加します。

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. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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オンにし、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 Managed Service for Prometheus 行の 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、Prometheus を選択します。
9. Amazon Managed Grafana が検索して Amazon Managed Service for Prometheus ワークスペースを検出する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、データソースの追加を選択します。

Prometheus データソースを手動で追加する

Prometheus 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、管理メニュー項目 (または Grafana v8 の設定 (歯車) アイコン) で一時停止し、データソース を選択します。
2. [データソースを追加する] を選択する。
3. Prometheus データソースを選択します。必要に応じて、**Prometheus**検索ボックスに入力して検索に役立てることができます。



## 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、新しいパネルに対して事前に選択されることを意味します。
Url	Prometheus サーバーの URL。例 : <code>https://prometheus.example.org:9090</code>
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。
Password	基本認証のパスワード。
Scrape interval	これを Prometheus で設定された一般的なスクレイプと評価間隔に設定します。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選択とメトリクス/ラベルのサポートが無効になります。これは、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 Parameters	Prometheus クエリ URL にカスタムパラメータを追加します。例えば、 <code>timeout</code> 、 <code>partial_response</code> 、 <code>dedup</code> 、または <code>max_source_resolution</code> などのパラメータを「&」と連結する必要があります。

## Prometheus クエリエディタ

以下のセクションでは、ダッシュボードと Explore で Prometheus クエリエディタの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
Q	Prometheus クエリ式の詳細については、 <a href="#">Prometheus ドキュメント</a> を参照してください。
L	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、 <code>{{hostname}}</code> はラベルのラベル値に置き換えられますhostname。
M	<a href="#">step Prometheus 範囲クエリのパラメータと、および</a> <code>\$__rate_interval</code> 変数の追加の下限。 <code>\$__interval</code> 制限は絶対値であり、解像度設定によって変更されません。
R	<code>1/1</code> 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 <code>\$__interval</code> 変数とパラメータの両方を設定します。 <a href="#">step</a> 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。 <code>1/2</code> は 1 ピクセルおきにデータポイントのみを取得し、10 ピクセルあたり 1 つのデータポイント <code>1/10</code> を取得します。最小時間間隔と最小ステップの両方が、 <code>\$__interval</code> との最終値を制限することに注意してくださいstep。
M	この入力フィールドでメトリクス名を検索します。
F	Table、Time series、またはの切り替えHeatmap。 はテーブルパネルでのみTable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、バケットバインドでシリーズをソートします。

## 名前

I Prometheus がリクエストされた時系列に対してスクレイピングした最新の値のみを返すには、「インスタント」クエリを実行します。インスタントクエリは、正常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。

M. この値に Resolution 設定の分母を乗算すると、Prometheus 範囲クエリの `$__interval` 変数とパラメータの両方に下限が設定されます。[step](#) 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になります。

### 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、メトリクスデータが一貫して表示されますが、グラフの右端にわずかなデータギャップが生じる可能性があります。

## ダッシュボード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。グラフで最新の値ポイントとして表示するには、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、シリーズオーバーライドの詳細については、[Transform > constant 「」](#)を参照してください [シリーズオーバーライド](#)。

## Explore のクエリエディタ

## 名前

Query Prometheus クエリ式の詳細については、[Prometheus ドキュメント](#) を参照してください。

名前	説明
Step	<a href="#">Step Prometheus 範囲クエリのパラメータ</a> 。ここで時間単位を使用できます。例: 5s、1m、3h、1d、1y。単位が指定されていない場合のデフォルト単位 s (秒)。
Query type	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、リクエスト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。この値はテーブルに表示されます。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、インスタント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構築できます。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、各メトリクスにはツールヒントとして HELP と TYPE が表示されます。

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、使用可能なラベル値が下部セクションのリストに表示される 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、クエリの範囲を絞り込みます。

### Note

開始するメトリクス名を覚えていない場合は、最初に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。その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、ジョブ、ジョブ名など、ラベルが選択された後にラベル間で値を見つけるのに役立ちます。探している値を持つ可能性があるのは 1 つだけです。

クエリに問題がなければ、「クエリを使用」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、クエリの周囲に `rate(...)[$__interval]` を追加して、カウンターメトリクスのクエリ

を書き込むのに役立ちます。「セレクトタの検証」ボタンは、そのセレクトタで利用できる時系列の数を Prometheus で確認します。

## 制限事項

メトリクスブラウザのハード制限は、10,000 ラベル (キー) と 50,000 ラベル値 (メトリクス名を含む) です。Prometheus インスタンスがより多くの結果を返す場合、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

サーバー、アプリケーション、センサー名などのモノをメトリクスクエリでハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレートと変数](#)。

## クエリ変数

Query 型の変数を使用すると、Prometheus にメトリクス、ラベル、またはラベル値のリストをクエリできます。Prometheus データソースプラグインは、クエリ入力フィールドで利用できる以下の関数を提供します。

名前	説明
<code>label_names()</code>	ラベル名のリストを返します。
<code>label_values(label)</code>	すべてのメトリクスlabelの のラベル値のリストを返します。
<code>label_values(metric, label)</code>	指定されたメトリクスlabelの のラベル値のリストを返します。
<code>metrics(metric)</code>	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。
<code>query_result(query)</code>	の Prometheus クエリ結果のリストを返しますquery。

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_ms` は  `$__range_s`、Grafana v5.3 からのみ利用できます。

クエリ変数には、 `$__interval`、 `$__interval_ms`、 `$__range_s`、 `$__range_ms`、および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。これらは、`query_result`関数がクエリをサポートしていないため、変数クエリをフィルタリングする必要がある場合に`label_values`、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 に設定してください `On Time Range Change`。

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 つのリクエストインスタンスを変数に入力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、 を使用して、ダッシュボードに表示される時間範囲で特定の状態を持つインスタンスを変数に入力する方法を示しています  `$__range_s`。

```
Query: query_result(max_over_time(<metric>[${__range_s}s]) != <state>)
Regex:
```

## `$__rate_interval` 変数の使用

`$__rate_interval` 変数は `rate` 関数で使用するためのものです。 `max($__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)、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、パネル内の最小間隔設定は無視されます。後者は解像度設定によって変更されるためです)。

### クエリでの変数の使用

次の 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])`

2つの方法があるのはなぜですか？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。複数值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 `[[varname]]` を使用する必要があります。

## 注釈

注釈を使用して、グラフの上に豊富な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声アラートの 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

Amazon Managed Service for Prometheus または Prometheus アラートを Amazon Managed Grafana で視覚化するには、既に接続している Prometheus データソースの Alertmanager データソースを設定します。

## 前提条件

Amazon Managed Service for Prometheus で使用するよう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す必要があります。

- 取り込まれたメトリクスと、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 [Amazon Managed Service for Prometheus](#) インスタンス。ワークスペースの URL が必要になります (Amazon Managed Service for Prometheus のワークスペースの詳細から、エンドポイント URL を確認できます。ワークスペース URI は、末尾 `api/v1/remote_write` に がないエンドポイント URL です)。

- [データソースとして設定された Prometheus](#) インスタンスで [作成された Amazon Managed Grafana](#) ワークスペース。
- 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 の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 Alertmanager データソースを設定するには

1. Grafana コンソールの Grafana メニューで、設定 の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. 新しいデータソースに次の情報を入力します。
  - 実装 で、Prometheus を選択します。
  - HTTP で、URL に Prometheus ワークスペース URL を指定し、`alertmanager` を追加します。例えば `https://aps-workspaces.us-east1.amazonaws.com/workspaces/ws-example-1234-5678-abcd-xyz00000001/alertmanager` です。
  - [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 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、データソースの設定を完了します。

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



新しい Alertmanager データソースを Prometheus データソースに接続するには

1. Grafana コンソールの Grafana メニューで、設定 の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 トグルスイッチを使用してアラートを管理するをオンにします。
3. Alertmanager データソースドロップダウンで、新しく作成した Alertmanager データソースを選択します。
4. 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。

例を設定する

#### Note

この機能には Prometheus バージョン 2.26 以降が必要です。  
Amazon Managed Service for Prometheus では、例外はサポートされていません。

Explore と Dashboards の両方で、メトリクスとともにサンプルデータを表示できます。例は、特定のイベントの高基数メタデータを従来の時系列データに関連付けます。

例へのリンクを追加することで、データソース設定で例を設定できます。URL でマクロを使用できます。例えば、などの URL を作成できます `https://example.com/${__value.raw}`。

Amazon Timestream 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、このデータソースで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てワークスペースを拡張する](#)」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 AWS、Amazon Timestream をデータソースとして追加できます。この機能により、既存の Timestream アカウントを検出してデータソースとして Timestream を簡単に追加し、Timestream へのアクセスに必要な認証情報の設定を管理できます。この方法を使用して認証をセットアップし、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セットアップできます。

## 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. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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 行の 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、Timestream を選択します。
10. Timestream 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。

## Timestream データソースを手動で追加する

Timestream 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) アイコンを一時停止し、データソース を選択します。
2. [データソースを追加する] を選択する。
3. Amazon Timestream データソースを選択します。必要に応じて、**Timestream**検索ボックスに入力して検索に役立てることができます。

### Timestream の設定

名前	説明
[Name] (名前)	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表示する方法です。
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されます (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)。デフォルトは空白のままにします。
ロール Arn を引き受ける	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。

## 認証

このセクションでは、Amazon Timestream データソースに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。

### 認証情報の例 AWS

Amazon Managed Grafana では、認証情報ファイルの認証方法は使用できません。

### 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、前述のマクロとダッシュボードテンプレート変数に加えて Timestream 構文を受け入れます。

Ctrl+Space を押して IntelliSense 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可能にするために、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
<code>\$_database</code>	選択した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。
<code>\$_table</code>	選択した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。
<code>\$_measure</code>	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。
<code>\$_timeFilter</code>	時間をダッシュボードの範囲に制限する式に置き換えられます。
<code>\$_interval_ms</code>	グラフ内の 1 つのピクセルがカバーする時間を表す数値に置き換えられます。

## 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、このデータソースで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てワークスペースを拡張する](#)」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 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設定オプションを使用して、Athena 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 Athena アカウントを検出して Athena にアクセスするために必要な認証情報の設定を管理することで、Athena をデータソースとして簡単に追加できます。この方法を使用して認証を設定し、Athena をデータソースとして追加できます。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。

Amazon Managed Grafana が Athena にアクセスできるようにするには、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、「」を参照してください [前提条件](#)。

### 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タスクを完了してください。

- Athena ワークグループに をタグ付けします `GrafanaDataSource: true`。
- で始まる名前の S3 バケットを作成します `grafana-athena-query-results-`。このポリシーは、その命名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アクセス許可を提供しません。

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 アクセス許可は、この管理ポリシーに含まれません。Amazon S3 バケットに必要なアクセス許可を手動で追加する必要があります。

あります case-by-case。詳細については、このガイドの「[Amazon Managed Grafana のアイデンティティ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および設定されている。
- アカウントから Athena にアクセスできます。

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エディタ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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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オンにし、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。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アタッチ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 データソースにクエリを実行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. の手順に従って Athena の詳細を設定します。 [Athena の詳細設定](#)

## Athena の詳細設定

### Athena の詳細設定を構成する

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 アカウントがあるターゲット 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、ドロップダウンにデフォルトのデータソースがあります。

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、ターゲット 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。デフォルトでは、プライマリは 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は、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、次のようになります: `s3://grafana-athena-plugin-test-data/query-result-output/`
6. 保存と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および設定されています。
- アカウントから Amazon Athena にアクセスできます。

#### Athena 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) アイコンを一時停止し、データソース を選択します。
2. [データソースを追加する] を選択する。
3. AWS Athena データソースを選択します。必要に応じて、**Athena**検索ボックスに入力して検索に役立てることができます。
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) 。

- ターゲットの 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](#)」の手順に従います。

- 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は、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/` です。
- 保存とテストを選択します。

Athena の詳細設定の例を次に示します。

Connection Details	
Authentication Provider	Workspace IAM Role
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Endpoint	https://{service}.{region}.amazonaws.com
Default Region	eu-west-1

Athena Details	
Data source	AwsDataCatalog
Database	athenacurcfn_o11y_costs
Workgroup	primary
Output Location	s3://grafana-athena-plugin-test-data/query-result-output/

Buttons: Back, Explore, Delete, Save & test

## Athena データソースの使用

### IAM ポリシー

Grafana は、Athena メトリクスを読み取るために IAM 経由で付与されたアクセス許可を必要とします。これらのアクセス許可を IAM ロールにアタッチし、Grafana の組み込みサポートを利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](#)する必要があります。データソースを追加するには、管理者ロールまたはエディタロールが必要です。組み込みの 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 データソースは、標準の SQL クエリエディタを提供します。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
<code>\$__dateFilter(column)</code>	<code>\$__dateFilter</code> は、パネルの日付範囲に基づいて ( を使用して <code>column</code> ) データを選択する条件付きフィルターを作成します。	<code>\$__date(my_date)</code>	<code>my_date BETWEEN date '2017-07-18' AND date '2017-07-18'</code>
<code>\$__parseTime(column, format)</code>	<code>\$__parseTime</code> は、指定された形式のタイムスタンプとして <code>varchar</code> をキャストします。	<code>\$__parseTime(event_time, 'yyyy-MM-dd' 'T' 'HH:mm:ss' 'Z')</code>	<code>parse_datetime(time, 'yyyy-MM-dd' 'T' 'HH:mm:ss' 'Z')</code>
<code>\$__timeFilter(column, format)</code>	<code>\$__timeFilter</code> は、パネルの時間範囲に基づいて ( <code>column</code> を使用して) データをフィルタリングする条件を作成します。2 番目の引数は、オプションで <code>varchar</code> から特定の形式のタイムスタンプに列を解析するために使用されます。	<code>\$__timeFilter(time, 'yyyy-MM-dd HH:mm:ss')</code>	<code>TIMESTAMP time BETWEEN TIMESTAMP '2017-07-18T11:15:52Z' AND TIMESTAMP '2017-07-18T11:15:52Z'</code>
<code>\$__timeFrom()</code>	<code>\$__timeFrom</code> は、パネルの範囲の現在の開始時刻を引用符で出力します。	<code>\$__timeFrom()</code>	<code>TIMESTAMP '2017-07-18 11:15:52'</code>

マクロ	説明	例	出力例
<code>\$__timeTo()</code>	<code>\$__timeTo</code> は、パネルの範囲の現在の終了時刻を引用符で出力します。	<code>\$__timeTo()</code>	TIMESTAMP '2017-07-18 11:15:52'
<code>\$__timeGroup(column, '1m', format)</code>	<code>\$__timeGroup</code> はタイムスタンプをグループ化し、グラフ上の期間ごとに1つのポイントしかしないようにします。3番目の引数は、オプションで列を <code>varchar</code> から特定の形式のタイムスタンプに解析するために使用されます。	<code>\$__timeGroup(time, '5m', 'yyy y-MM-dd' T'HH:mm: ss.SSSSSS' 'Z')</code>	FROM_UNIX TIME(FLOOR (TO_UNIX TIME(parse_dattim e(time, 'yyyy- MM-dd' T'HH:m m:ss.SSSS SS'Z'))/ 300)*300)
<code>\$__table</code>	<code>\$__table</code> は、テーブルセクタで選択されたテーブルを返します。	<code>\$__table</code>	my_table
<code>\$__column</code>	<code>\$__column</code> は、列セクタで選択された列を返します (テーブルが必要です)。	<code>\$__column</code>	col1

## 視覚化

Athena のほとんどのクエリは、テーブルの視覚化によって最もよく表されます。クエリは、返されたデータをテーブルに表示します。クエリできる場合は、テーブルとして表示できます。

この例では、テーブルの視覚化の結果を返します。

```
SELECT {column_1}, {column_2} FROM {table};
```

## 時系列/グラフの視覚化

時系列とグラフの視覚化では、以下を行う必要があります。

- date または datetime タイプの列を選択します。date 列は昇順である必要があります ( 使用 ORDER BY column ASC )。
- 数値列も選択します。

## クエリの検査

Amazon Managed Grafana は、Athena がサポートしていないマクロをサポートしています。つまり、Athena に直接コピーして貼り付けると、クエリが機能しない可能性があります。Athena で直接機能する補間されたクエリ全体を表示するには、Query Inspector ボタンをクリックします。完全なクエリは、クエリタブの下に表示されます。

## テンプレートと変数

Athena クエリ変数の追加の詳細については、「」を参照してください [クエリ変数の追加](#)。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。

変数を作成したら、を使用して Athena クエリでその変数を使用できます [変数構文](#)。変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## 注釈

[注釈](#) では、グラフの上に豊富な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
```

次の表は、注釈のレンダリングに使用できる列の説明を示しています。

名前	説明
Time	日付/時刻フィールドの名前。SQL のネイティブ日付/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。SQL のネイティブ日付/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	カンマ区切りの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、タイムアウトを回避するために非同期的に処理されます。非同期クエリでは、個別のリクエストを使用してクエリを開始し、その進行状況を確認し、最後に結果を取得します。これにより、長時間実行されるクエリのタイムアウトを回避できます。

## 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。クエリの再利用を有効にするには、クエリエディタのクエリ結果の再利用セクションで を有効にします。これは、クエリを再利用するクエリごとに実行する必要があります。

### Note

この機能を使用するには、Athena インスタンスがエンジンバージョン 3 に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 エンジンのバージョンの変更](#)」を参照してください。 Amazon Athena

## 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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

### Note

このガイドでは、ユーザーが Amazon Redshift データソースを使用する前に 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 Amazon Redshift にアクセスするために必要な認証情報の設定を管理することで、データソースとして Amazon Redshift を簡単に追加できます。この方法を使用して認証を設定し、Amazon Redshift をデータソースとして追加できます。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。

Amazon Redshift が Amazon Managed Grafana からアクセスできるようにするには、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、「」を参照してください。[前提条件](#)。

### 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。そうしないと、アクセスできなくなります。
- データベース認証情報は、次のいずれかの相互に排他的な方法で作成します。
  - デフォルトのメカニズム (一時的な認証情報オプション) を使用して Redshift データベースに対して認証する場合は、 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 `redshift_data_api_user`。

- Secrets Manager の認証情報を使用する場合は、シークレットに をタグ付けする必要があります。RedshiftQueryOwner: true。詳細については、このガイドの「[Amazon Managed Grafana のアイデンティティ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エディタ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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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、下の AWS アイコン (2 つあります) を選択し、Redshift を選択します。
8. Amazon Redshift 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、目的の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。
9. 「」の手順に従って接続の詳細を設定します [接続の詳細設定](#)。

Amazon Redshift データソースを手動で追加する

前提条件

- アカウントから Amazon Redshift にアクセスできます。

Amazon Redshift データソースを追加するには :

1. をワークスペースユーザーロール[AmazonRedshiftAccessPolicy](#)にアタッチします。
2. ユーザーロールが管理者またはエディタ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) アイコンを一時停止し、データソースを選択します。
5. [データソースを追加する] を選択する。
6. AWS Redshift データソースを選択します。必要に応じて、**Redshift**検索ボックスに入力して検索に役立てることができます。
7. 接続の詳細ページが開きます。の設定のステップに従います[接続の詳細設定](#)。

## 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た後、Amazon Redshift の設定を次のように設定します。

### 前提条件

- 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

#### 接続の詳細設定を構成する

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的な認証情報または AWS Secrets Manager 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. デフォルトで一時的な認証情報を選択した場合は、以下のステップに従います。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. ターゲットの Redshift データベースを選択します。
5. 上記のクラスター用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、一時的な認証情報設定の例です。

Connection Details	
Authentication Provider	Workspace IAM Role
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Endpoint	https://{service}.{region}.amazonaws.com
Default Region	us-east-1
<b>Authentication</b>	
Temporary credentials      AWS Secrets Manager	
Use the <code>GetClusterCredentials</code> IAM permission and your database user to generate temporary access credentials. <a href="#">Learn more</a>	
Cluster Identifier	amg-reinvent-demo
Database	dev
Database User	awsuser

AWS Secrets Manager メニューの例を次に示します。



The screenshot shows the 'Settings' page for an Amazon Redshift connection in Amazon Managed Grafana. The page is dark-themed and includes a navigation bar with 'Settings', 'Dashboards', 'Permissions', and 'Insights'. The main content area is titled 'Amazon Redshift' and has a 'Default' toggle switch. Below this is the 'Connection Details' section, which includes a table of configuration options: Authentication Provider (Workspace IAM Role), Assume Role ARN (arn:aws:iam:\*), External ID (External ID), Endpoint (https://{service}.{region}.amazonaws.com), and Default Region (Choose). The 'Authentication' section is active, showing 'AWS Secrets Manager' as the selected method. It includes a 'Managed Secret' dropdown (set to 'Choose') and input fields for 'Cluster Identifier', 'Database User', and 'Database'. At the bottom, there are four buttons: 'Back', 'Explore', 'Delete', and 'Save & test'.

## Amazon Redshift データソースの使用

### IAM ポリシー

Grafana が Redshift メトリクスを読み取るには、IAM を使用して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 ロールにアタッチし、Grafana の組み込みサポートを利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
<code>\$__timeEpoch(column)</code>	<code>\$__timeEpoch</code> は、UNIX タイムスタンプに変換し、列の名前を時刻に変更する式に置き換えられます。	<code>UNIX_TIMESTAMP(dateColumn) as "time"</code>
<code>\$__timeFilter(column)</code>	<code>\$__timeFilter</code> は、パネルの時間範囲に基づいて ( を使用して <code>column</code> ) データをフィルタリングする条件を作成します。	<code>time BETWEEN '2017-07-18T11:15:52Z' AND '2017-07-18T11:15:52Z'</code>
<code>\$__timeFrom()</code>	<code>\$__timeFrom</code> は、パネルの範囲の現在の開始時刻を引用符で出力します。	<code>'2017-07-18T11:15:52Z'</code>
<code>\$__timeTo()</code>	<code>\$__timeTo</code> は、パネルの範囲の現在の終了時刻を引用符で出力します。	<code>'2017-07-18T11:15:52Z'</code>
<code>\$__timeGroup(column, '1m')</code>	<code>\$__timeGroup</code> 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 つのポイントしかないようにします。	<code>floor(extract(epoch from time)/60)*60 AS "time"</code>
<code>\$__schema</code>	<code>\$__schema</code> は選択したスキーマを使用します	<code>public</code>
<code>\$__table</code>	<code>\$__table</code> は、指定されたからテーブルを出力します <code>\$__schema</code> (デフォルトではパブリックスキーマを使用します)	<code>sales</code>
<code>\$__column</code>	<code>\$__column</code> が現在のから列を出力する <code>\$__table</code>	<code>date</code>

マクロ	説明	出力例
<code>\$__unixEpochFilter (column)</code>	<code>\$__unixEpochFilter</code> 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に置き換えられます。	<code>column &gt;= 1624406400 AND column &lt;= 1624410000</code>
<code>\$__unixEpochGroup(column)</code>	<code>\$__unixEpochGroup</code> は同じ <code>\$__timeGroup</code> ですが、Unix タイムスタンプとして保存される時間の場合	<code>floor(time/60)*60 AS "time"</code>

## 視覚化

Redshift のほとんどのクエリは、テーブルの視覚化によって最もよく表されます。クエリでは、テーブルにデータが表示されます。クエリできる場合は、テーブルに格納できます。

この例では、テーブルの視覚化の結果を返します。

```
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、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` 列は昇順にする必要があります ( を使用 `ORDER BY column ASC` )。
- 数値列を選択する必要があります。

より合理的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;
```

## フィルモード

Grafana は、一部のデフォルトを持つ値なしでフレームをオートコンプリートします。この値を設定するには、クエリエディタで Fill Value を変更します。

## クエリの検査

Grafana は Redshift がサポートしていないマクロをサポートしているため、完全にレンダリングされたクエリは Redshift に直接コピーして貼り付けることができ、Query Inspector に表示されます。補間されたクエリ全体を表示するには、Query Inspector 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。

## テンプレートと変数

新しい Redshift クエリ可変を追加する方法の詳細については、「」を参照してください [クエリ変数の追加](#)。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可能性があるため、選択しすぎないようにしてください。

変数を作成したら、を使用して Redshift クエリでその変数を使用できます [変数構文](#)。変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## 注釈

[注釈](#) では、豊富なイベント情報をグラフの上に重ねることができます。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いた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、注釈をレンダリングするために考慮される列の値を示しています。

名前	説明
Time	日付または時刻フィールドの名前。SQL 日付または時刻のネイティブデータ型またはエポック値の列にすることができます。
Timeend	終了日または時刻フィールドのオプション名。ネイティブ SQL 日付または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	カンマ区切りの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、このデータソースで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てワークスペースを拡張する](#)」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 で探索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 AWS 設定オプションを使用して、データソースとして X-Ray を追加できます。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セットアップし、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 データソースを手動で追加する](#)
- [X-Ray の設定](#)
- [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の 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. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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管理アクセス許可を使用することを選択しなかった場合は、カスタマー管理アクセス許可の使用からサービス管理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ようにします。これを行うには、IAM ロールで編集アイコンを選択し、サービス管理、変更の保存 を選択します。詳細については、「[AWS データソースの Amazon Managed Grafana の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、 のチェックボックスを選択しAWS X-Ray、アクション、サービス管理ポリシーの有効化 を選択します。
6. データソースタブを再度選択し、AWS X-Ray行の 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。

## X-Ray データソースを手動で追加する

### 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) アイコンを一時停止し、データソース を選択します。
2. [データソースを追加する] を選択する。
3. X-Ray データソースを選択します。必要に応じて、**X-Ray**検索ボックスに入力して検索に役立てることができます。

### X-Ray の設定

名前	説明
[Name] (名前)	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表示する方法です。
デフォルト値	デフォルトのデータソースは、新しいパネルに対して事前に選択されることを意味します。
デフォルトリージョン	クエリエディタでリージョンを設定するために使用されます (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)。デフォルトは空白のままにします。
ロール Arn を引き受ける	引き受けるロールの ARN を指定します。
外部 ID	外部 ID で作成された別のアカウントの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 データソースに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 で実行されている場合 AWS、IAM ロールを使用して認証が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 には、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 ロールにアタッチし、組み込みの Grafana サポートを使用してロールを引き受けることができます。

次のコード例は、最小限のポリシーを示しています。

```
{
  "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": "*"
    }
  ]
}
```



## 認証情報の例 AWS

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。次の 4 つのクエリタイプがあります。

- 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 )
- インサイト

### トレースリスト

トレースリストタイプを使用すると、テーブルに表示されるトレースを検索できます。最初の列でトレース ID を選択すると、トレースが右側に開きます。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、「」の [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 1000 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表を確認できます。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

## インサイト

Insights では、Insights の概要テーブルを確認できます。を選択すると InsightId、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ことができるため、アラートは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 データソースに接続する

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の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) サービスであり、ライブウェブアプリケーションをモニタリングするために使用できます。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 によって収集され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て [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 AD で設定する方法であれば、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。

- [Azure Monitor 用 Azure Active Directory アプリケーションのセットアップガイド](#)
- [Azure Log Analytics 用 Azure Active Directory アプリケーションのセットアップガイド](#)。
- [Application Insights のクイックスタートガイド](#)。

1. Grafana のメインメニューからアクセスでき、新しくインストールされたデータソースはデータソースセクション内で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 データソースは、データソースのリストの クラウドセクションで選択できます。
2. 名前フィールドに、Grafana はデータソースの名前、Azure Monitor または などの名前を自動的に入力します。Azure Monitor - 3。複数のデータソースを設定する場合は、名前をより情報の多いものに変更します。
3. Azure Monitor を使用している場合は、Azure ポータルから 4 つの情報がが必要です (詳細な手順については、前述のリンクを参照してください)。
  - テナント ID (Azure Active Directory、プロパティ、ディレクトリ ID)
  - クライアント ID (Azure Active Directory、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 Active Directory、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. Azure Monitor API の詳細セクションのフィールドに、これら 4 つの項目を貼り付けます。

- 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、ページを更新して、指定されたクライアント ID で利用可能なサブスクリプションのリストを表示します。
5. Azure Log Analytics サービスも使用している場合は、これら 2 つの設定値を指定するか、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 Active Directory、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 Active Directory、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 を使用している場合は、Azure ポータルから 2 つの情報が必要です (詳細な手順については、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 の詳細セクションの適切なフィールドに貼り付けます。
  8. 保存と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 Active Directory 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 Azure Monitor データソースを選択した後、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

クエリエディタは、選択したオプションに応じて変わります。Azure Monitor がデフォルトです。

## Azure Monitor サービスのクエリ

Azure Monitor サービスは、実行しているすべての Azure サービスのメトリクスを提供します。Azure 上のアプリケーションがどのように動作しているかを理解し、アプリケーションに影響を与える問題をプロアクティブに検出するのに役立ちます。

Azure Monitor の認証情報で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の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しているメトリクスをクエリできます。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](#)にリストされ、メトリクスのディメンション列に 1 つ以上の値がリストされているメトリクスです。

Azure Monitor のエイリアスを使用した凡例キーのフォーマット

Azure Monitor API のデフォルトの凡例形式は次のとおりです。

```
metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}
```

これらは長い場合がありますが、エイリアスを使用してこの書式を変更できます。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られます
- {{ namespace }} = を名前空間の値 (Microsoft.Compute/virtualMachines など) に置き換えます。

- `{{ resourcename }}` = リソースネームの値に置き換えられます
- `{{ metric }}` = メトリクス名に置き換えられます (例: CPU の割合 )
- `{{ dimensionname }}` = 7.1 以降 (下位互換性のため) の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) (blobtype など) に置き換えられたレガシー
- `{{ dimensionvalue }}` = 7.1 以降 (下位互換性のため) の最初のディメンションの値 (キー/ラベルでソート) に置き換えられたレガシー (例 : BlockBlob )
- `{{ arbitraryDim }}` = 7.1 以降で使用可能、対応するディメンションの値に置き換えられます (例えば、`{{ blobtype }}` は になります BlockBlob )。

## 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、センサー名などのモノをハードコーディングする代わりに、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 サービスはまだ複数の値をサポートしていない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 Query 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埋めることができます。

名前	説明
<code>Subscriptions()</code>	サブスクリプションのリストを返します。
<code>ResourceGroups()</code>	リソースグループのリストを返します。
<code>ResourceGroups(12345678-aaaa-bbbb-cc cc-123456789aaa)</code>	指定された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。
<code>Namespaces(aResourceGroup)</code>	指定されたリソースグループの名前空間のリストを返します。

名前	説明
<code>Namespaces(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup)</code>	指定された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。
<code>ResourceNames(aResourceGroup, aNamespace)</code>	リソース名のリストを返します。
<code>ResourceNames(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace)</code>	指定されたサブスクリプションのリソース名のリストを返します。
<code>MetricNamespace(aResourceGroup, aNamespace, aResourceName)</code>	メトリクス名前空間のリストを返します。
<code>MetricNamespace(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace, aResourceName)</code>	指定された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。
<code>MetricNames(aResourceGroup, aNamespace, aResourceName)</code>	メトリクス名のリストを返します。
<code>MetricNames(12345678-aaaa-bbbb-cccc-123456789aaa, aResourceGroup, aNamespace, aResourceName)</code>	指定された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:

- Resource Groups クエリ: `ResourceGroups()`
- メトリクス名変数の受け渡し: `Namespaces(cosmo)`
- テンプレート変数の連鎖: `ResourceNames($rg, $ns)`
- パラメータを引用符で囲まないでください。 `MetricNames(hg, Microsoft.Network/publicIPAddresses, grafanaIP)`

テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレート](#)。

## サポートされている Azure Monitor メトリクスのリスト

Azure Monitor API によって返されるすべてのメトリクスに値があるわけではありません。クエリの構築を容易にするために、Grafana データソースにはサポートされている 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。このリストは、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 アラート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、「」を参照してください[Grafana アラート](#)。

### 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フォーマット

デフォルトの凡例の書式は次のとおりです。

```
metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}
```

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。

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+ 時点のレガシー (下位互換性用) を、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) に置き換えました
- `{{ groupbyname }}` = Grafana 7.1+ 時点のレガシー (下位互換性用) を最初のディメンションの値 (キー/ラベルでソート) に置き換えました (例 : BlockBlob )
- `{{ metric }}` = メトリクス名 (リクエスト/カウントなど) に置き換えられます
- `{{ 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られます。(例えば、`{{ client/city }}` はシカゴになります)



## Application Insights のフィルター式

フィルターフィールドは OData フィルター式を受け取ります。

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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレート](#)。

名前	説明
<code>AppInsightsMetricNames()</code>	メトリクス名のリストを返します。
<code>AppInsightsGroupBys(aMetricName)</code>	指定されたメトリクス名の <code>group by</code> 句のリストを返します。

例:

- メトリクス名クエリ: `AppInsightsMetricNames()`
- メトリクス名変数を渡す: `AppInsightsGroupBys(requests/count)`
- テンプレート変数の連鎖: `AppInsightsGroupBys($metricnames)`

## Application Insights アラート

Grafana アラートは Application Insights でサポートされています。これは Azure アラート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、「」を参照してください[Grafana アラート](#)。

## 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。ログ分析クエリは、時系列データまたはテーブルデータとしてフォーマットできます。

認証情報で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、グラフパネルとパネルなどの他の SingleStat 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、数値/日時以外の列を 1 つ以上含めることもできます。これらの列はディメンションと見なされ、レスポンスのラベルになります。例えば、時間、コンピュータ、でグループ化された集計カウントを返すクエリなどです CounterName。

```
Perf
| where $__timeFilter(TimeGenerated)
| summarize count() by bin(TimeGenerated, 1h), Computer, CounterName
| order by TimeGenerated asc
```

追加の数値列 (複数のディメンションの有無にかかわらず) を選択することもできます。例えば、時間、コンピュータ、でカウント CounterName と平均値を取得するとします InstanceName。

```
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、Kusto 構文 と を使用して列の名前を変更します。これは、スペースを許可する 2 番目の構文 `Samples=count()["Avg Value"]=...` です。これにより、Grafana が使用するメトリクスの名前が変更されます。その結果、シリーズ凡例やテーブル列などのモノは、指定した内容と一致します。この例では、 の代わりに `Samples` が表示されます `_count`。

## 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、列と行のリストが表示されます。このクエリ例では、6 つの列が指定された行が返されます。

```
AzureActivity
| where $__timeFilter()
| project TimeGenerated, ResourceGroup, Category, OperationName, ActivityStatus, Caller
| order by TimeGenerated desc
```

## ログ分析の表示名のフォーマット

デフォルトの表示名形式は次のとおりです。

```
metricName{dimensionName=dimensionValue,dimensionTwoName=DimensionTwoValue}
```

これは、表示名フィールド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。

- `$__timeFilter()` – Grafana タイムピッカーからの日時のと `TimeGenerated ≥ datetime(2018-06-05T18:09:58.907Z) and TimeGenerated ≤ datetime(2018-06-05T20:09:58.907Z)` どこにあるかに拡張されます。
- `$__timeFilter(datetimeColumn)` – Grafana タイムピッカーからの日時のと `datetimeColumn ≥ datetime(2018-06-05T18:09:58.907Z) and datetimeColumn ≤ datetime(2018-06-05T20:09:58.907Z)` どこにあるかに拡張されます。

- `$__timeFrom()` – Grafana ピッカーから From datetime を返します。例えば、`datetime(2018-06-05T18:09:58.907Z)` などです。
- `$__timeTo()` – Grafana ピッカーから From datetime を返します。例えば、`datetime(2018-06-05T20:09:58.907Z)` などです。
- `$__escapeMulti($myVar)` – は、無効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。`$myVar` に文字列として次の 2 つの値がある場合 `'\\grafana-vm\\Network(eth0)\\Total', '\\hello!'`、に拡張されます `@'\\grafana-vm\\Network(eth0)\\Total', '@'\\hello!'`。単一値変数を使用する場合、このマクロは必要ありません。代わりに変数をインラインでエスケープします: `@'\\$myVar'`。
- `$__contains(colName, $myVar)` – は複数値のテンプレート変数で使用されます。`$myVar` に値がある場合 `'value1', 'value2'`、に拡張されます `colName in ('value1', 'value2')`。

All オプションを使用する場合は、Include All Option チェックボックスをオンにし、Custom all value フィールドにと入力します `all`。に値 `$myVar` がある場合 `all`、マクロは代わりにに展開されます `1 == 1`。オプションが多数あるテンプレート変数の場合、大きな「where..in」句を作成しないため、クエリのパフォーマンスが向上します。

## Azure Log Analytics の組み込み変数

Azure Log Analytics クエリで使用できる Grafana 変数もあります。

- `$__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。5m やなどの時間粒度を返1hするので、など、bin 関数で使用できます `summarize count() by bin(TimeGenerated, $__interval)`。間隔変数の詳細については、「」を参照してください [間隔変数の追加](#)。

## Azure Log Analytics の変数を使用したテンプレート作成

値のリストを返すログ分析クエリは、変数編集ビューのクエリフィールドで使用できます。ログ分析には、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、「」を参照してください [テンプレートと変数](#)。

名前	説明
<code>workspaces()</code>	デフォルトの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。

名前	説明
<code>workspaces(12345678-aaaa-bbbb-cccc-123456789aaa)</code>	指定された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で囲むことも引用符で囲まないこともできます)。

次の表は、変数クエリの例を示しています。

Query	説明
<code>subscriptions()</code>	Azure サブスクリプションのリストを返します。
<code>workspaces()</code>	デフォルトの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。
<code>workspaces("12345678-aaaa-bbbb-cccc-123456789aaa")</code>	指定された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。
<code>workspaces("\$subscription")</code>	サブスクリプションパラメータのテンプレート変数。
<code>workspace("myWorkspace").Heartbeat \ distinct Computer</code>	仮想マシンのリストを返します。
<code>workspace("\$workspace").Heartbeat \ distinct Computer</code>	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。
<code>workspace("\$workspace").Perf \ distinct ObjectName</code>	Perf テーブルからオブジェクトのリストを返します。
<code>workspace("\$workspace").Perf \ where ObjectName == "\$object" \ distinct CounterName</code>	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 xample 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 のログ分析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。リンクを選択すると、Azure ポータルの Azure Log Analytics クエリエディタが開き、そこで Grafana パネルからクエリが実行されます。

現在 Azure ポータルにログインしていない場合、リンクをクリックするとログインページが開きます。提供されたリンクはどのアカウントでも有効ですが、アカウント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場合に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 でサポートされています。これは Azure アラートのサポートではありません。Grafana ワークスペースでのアラートの詳細については、「」を参照してください [Grafana アラート](#)。

## 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、ログ分析サービスと同様のエディタを使用できます。このサービスは Kusto 言語も使用するため、データのクエリ手順は同じですが [Azure Log Analytics サービスのクエリ](#)、代わりに Application Insights Analytics データをクエリする点が異なります。

## Graphite データソースに接続する

Grafana には、メトリクススペースの迅速な移動、関数の追加、関数パラメータの変更などを可能にする高度な Graphite クエリエディタがあります。エディタは、すべてのタイプのグラフアイトクエリを処理できます。クエリ参照を使用して、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 の設定

Graphite 設定にアクセスするには、設定 (歯車) アイコンで一時停止し、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、新しいパネルに対して事前に選択されることを意味します。
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。

名前	説明
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。他に何も記載されていない場合は、サーバーが推奨される方法です。

#### 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 に転送され、リクエストはデータソースに転送され、Cross-Origin Resource Sharing (CORS) の要件を回避します。このアクセスモードを選択した場合、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 (鉛筆) の切り替えアイコンを選択します。

#### クエリするメトリクスの選択

メトリクスの選択 を選択して、メトリクススペースをナビゲートします。起動後、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



にフォーカスが表示されます。パラメータを編集または変更するには、パラメータを選択すると、テキストボックスに変わります。- 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。

などの一部の関数は `aliasByNode`、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、それを選択し、空白のままにします。エディタによって削除されます。

## ソートラベル

一貫した順序付けが必要な場合は、 を使用します `sortByName`。これは、複数のグラフに同じラベルがあり、それぞれ異なるソートと異なる色を使用した場合に、迷惑になる可能性があります。これを修正するには、 を使用します `sortByName()`。

## ネストされたクエリ

クエリは、クエリが存在する行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、複合クエリを構築するための便利な方法が提供されます。

## ワイルドカード文字を使用した多くのクエリの回避

場合によっては、同じグラフに複数の時系列がプロットされることがあります。例えば、CPU がマシンでどのように使用されているかを確認することができます。最初は、`cpu.percent.user.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 `cpu.percent.system.g`。これにより、データソースに対してクエリが `n` 件行われ、非効率になります。

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、すべての時系列を 1 つのクエリで返すことができます。例えば `cpu.percent.*.g` です。

## 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()` です。

## ポイント統合

すべての Graphite メトリクスは統合されているため、Graphite はグラフにピクセルがあるよりも多くのデータポイントを返しません。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、凡例のサマリー値 (最大、最小、合計) がすべて同時に正しいとは限りません。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 だけが同時に正しい状態になります。

## 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。

データを調べる場合、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、タグ式 (、=、 、 =~!=~) を使用して時系列をフィルタリングする！ =seriesByTag関数を使用します。

Grafana クエリビルダーは、タグを選択すると自動的にこれを行います。

**Note**

ヒント：正規表現の検索は高カーディナリティ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スコープ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、結果を減らすのに役立ちます。

## テンプレート変数

サーバー、アプリケーション、センサー名などのモノをメトリクスクエリでハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレートと変数](#)。

タグ値を使用して変数を作成するには、Grafana 関数 tagsおよび tag\_values。

Query	説明
<code>tags()</code>	すべてのタグを返します。
<code>tags(server=~backend\*)</code>	フィルター式に一致する連続して発生するタグのみを返します。
<code>tag_values(server)</code>	指定されたタグのタグ値を返します。
<code>tag_values(server, server=~backend\*)</code>	指定されたタグに対して発生するフィルタリングされたタグ値を、それらの式と一致する連続して返します。
<code>tag_values(server, server=~backend\*, app=~\${apps:regex})</code>	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 の「Graphite docs」](#) 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.*` のクエリは、ワイルドカードの位置に存在するすべての可能な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、クエリ定義 \$app で 変数 apps.\$app.servers.\* を使用します。

を使用してクエリ変数の結果 `__searchFilter` をフィルタリングする

クエリフィールド `__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、のデフォルト値 `__searchFilter` は `*` であり、正規表現の一部として使用する場合は ``` です。

次の例は、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 `server` している間に、`__searchFilter` をクエリフィールド `__searchFilter` の一部として使用して検索を有効にする方法を示しています。

### Query

```
apps.$app.servers.$__searchFilter
```

### TagValues

```
tag_values(server, server=~${__searchFilter:regex})
```

### 可変使用量

変数は、メトリクスノードパスまたは関数のパラメータとして使用できます。

次の 2 つの構文があります。

- `<varname>` 例: apps.frontend.\$server.requests.count
- `${varname}` 例: apps.frontend.\${server}.requests.count

2 つの方法があるのはなぜですか？ 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。などの式で 2 番目の構文を使用します `my.server ${serverNumber}.count`。

### タグクエリでの可変使用量

タグクエリの複数值変数は、Grafana 5.0 で導入された高度なフォーマット構文を変数に使用します `{var:regex}`。タグ以外のクエリでは、複数值変数にデフォルトの 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、等号演算子を使用したタグ式を示しています=。

```
server=~${servers:regex}
```

詳細については、「[高度な変数形式のオプション](#)」を参照してください。

## 注釈

注釈を使用すると、グラフの上に豊富な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。そのためには、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。そのためには、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 (空のままにしておくとも機能します)。

## Google Cloud Monitoring データソースに接続する

### 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 データソースを追加して、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します。

### データソースの追加

1. 上部のヘッダーにある 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 Dashboards リンクの下に、データソースリンクがあります。
3. 上部のヘッダーで + データソースの追加ボタンを選択します。
4. Type 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キーファイルをアップロードまたは貼り付けます。サービスアカウント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、新しいパネルに対して事前に選択されることを意味します。
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、このドキュメントで後述する手順を参照してください。

## 認証

Google Cloud Monitoring プラグインを認証するには 2 つの方法があります。

- Google JWT ファイルをアップロードする
- Google メタデータサーバーから認証情報を自動的に取得する

後者のオプションは、GCE 仮想マシンで Grafana を実行する場合にのみ使用できます。

### Google サービスアカウント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。

### APIs有効化

最初に次の APIs を有効にする必要があります。

- [モニタリング API](#)

## • [クラウドリソースマネージャー API](#)

リストされているリンクを選択し、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. [APIs とサービス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、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"" >}}
```

3. サービスアカウントキーの作成ページで、キータイプを選択しますJSON。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" クラス "docs-image-no-shadow" キャプション "サービスアカウントキーの作成" >}}
```

4. 一部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、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" クラス "docs-image-no-shadow" 字幕 "Choose role" >}}
```

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、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、安全な場所に保存します。
6. データソース設定ページの Grafana にアップロードします。ファイルをアップロードするか、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" クラス "docs-image-no-shadow" キャプション "サービスキーファイルを Grafana にアップロード" >}}
```

7. ファイルの内容は暗号化され、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" クラス "docs-image-no-shadow" キャプション "Service キーファイルが Grafana" >}}
```

にアップロードされました

## クエリエディタの使用

Google Cloud Monitoring クエリエディタを使用すると、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を使用すると、メトリクスの選択、ラベルおよび時間によるグループ化/集計、フィルターを使用した結果の時系列の指定を行うことができます。

メトリクスクエリを作成するには、次の手順に従います。

1. クエリタイプのドロップダウンリストで、メトリクスオプションを選択します。
2. プロジェクト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 Google クラウドプラットフォームサービスを選択します。
4. メトリクスドロップダウンリストからメトリクスを選択します。
5. フィルターを追加または削除したり、句でグループ化したりするには、フィルターとセクションでプラスアイコン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 (リデューサーとアリーマー) がサポートされています。Grafana クエリエディタは、選択したメトリクスで使用可能な集計メソッドのリストを表示し、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リーマーを設定します。Y 軸の単位もクエリエディタによって自動的に選択されます。

### 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、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、`--remove filter--` を選択します。

### シンプルなワイルドカード文字

演算子が または に設定されている場合、`!=` フィルター値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 `*central-a` キャプチャし、「central-a」で終わるすべての値をキャプチャします。 `は` の部分文字列を持つすべての値を `*-central-*` キャプチャします `central-`。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

演算子が または に設定されている場合、`=~!=~` フィルター値フィールドに正規表現を追加できます。例えば、`us-central` で始まり、その後で 1 ~ 3 の範囲の数値、ダッシュ、その後で「a」または「f」のいずれかが続くすべての値 `us-central[1-3]-[af]` に一致します。正規表現を作成するときは、先頭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 フィールドを使用すると、同じグループの後に複数の時系列を時間間隔で調整できます。アライナーの詳細については、「[アラインメントメトリクスセレクタ](#)」を参照してください。

## 調整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、`cloud_monitoring_auto` はメトリクスを時間単位で Alignment Period 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が使用されます (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)。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 < 23 時間の場合は 1m
- 時間範囲 >= 23 時間 < 6 日で 5m
- 時間範囲 >= 6 日の場合は 1h

もう 1 つの自動オプションは `grafana auto` です。これにより、選択した時間範囲とグラフパネルの幅に応じて、時間ごとにグループが自動的に設定されます。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間隔を選択することもできます。

## グループ化の基準

リソースまたはメトリクスの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` でグループ化すると、コンピューティングインスタンスの集計メトリクスが表示されます。

## メタデータラベル

リソースメタデータラベルには、Google クラウド内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 Group By 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、選択したサービスおよびメトリクスで使用できるメタデータラベルを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付属しています。

ユーザーラベルは事前定義できませんが、グループ別フィールドに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 Group By 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

Alias By フィールドでは、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い場合があります。エイリアスフィールドで次のパターンを使用して、凡例キーを目的どおりに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
<code>{{metric.type}}</code>	完全なメトリクスタイプを返します。	<code>compute.googleapis.com/instance/cpu/utilization</code>
<code>{{metric.name}}</code>	メトリクス名の部分を返します。	<code>instance/cpu/utilization</code>
<code>{{metric.service}}</code>	サービスパーツを返します。	<code>compute</code>

## ラベルパターン

Group By ドロップダウンリストには、メトリクスとメトリクスのリソースラベルのリストが表示されます。これらは、エイリアスパターンを使用して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
<code>{{metric.label.xxx}}</code>	メトリクスラベル値を返します。	<code>{{metric.label.instance_name}}</code>	grafana-1-prod
<code>{{resource.label.xxx}}</code>	リソースラベルの値を返します。	<code>{{resource.label.zone}}</code>	us-east1-b
<code>{{metadata.system_labels.xxx}}</code>	メタデータシステムのラベル値を返します。	<code>{{metadata.system_labels.name}}</code>	grafana
<code>{{metadata.user_labels.xxx}}</code>	メタデータユーザーラベルの値を返します。	<code>{{metadata.user_labels.tag}}</code>	production

エイリアスの例: `{{metric.type}}` - `{{metric.label.instance_name}}`

結果の例: `compute.googleapis.com/instance/cpu/usage_time` - `server1-prod`

モニタリング対象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
<code>{{resource.type}}</code>	モニタリング対象のリソースタイプの名前を返します。	<code>gce_instance</code>

エイリアスの例: `{{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、次の手順に従います。

1. クエリタイプのドロップダウンリストで、サービスレベル目標 (SLO) オプションを選択します。
2. プロジェクト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. [セレクトアドロップダウンリストから時系列](#)セレクトアを選択します。

時系列セレクトアのフレンドリ名は Grafana に表示されます。次の表は、サービスモニタリングドキュメントで使用されているフレンドリ名からシステム名へのマッピングを示しています。

セレクトアドロップダウンリストの値	対応する時系列セレクトアが使用されている
SLI 値	select_slo_health
SLO コンプライアンス	select_slo_compliance
残りの SLO エラー予算	select_slo_Budget_fraction

### SLO クエリのエイリアスパターン

Alias By フィールドを使用して、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
{{selector}}	セレクターを返します。	select_slo_health

## SLO クエリの調整期間と時間別のグループ化

SLO クエリは、メトリクスクエリと同じ調整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

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレートと変数](#)。

## クエリ変数

Query 型の変数を使用すると、さまざまなタイプのデータを Google Cloud Monitoring にクエリできます。Google Cloud Monitoring データソースプラグインには、次の が用意されています Query Types。

名前	説明
Metric Types	指定された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	resource label 指定されたメトリクスの metric label および のキーのリストを返します。

名前	説明
Labels Values	指定されたメトリクス内のラベルの値のリストを返します。
Resource Types	指定されたメトリクスのリソースタイプのリストを返します。
Aggregations	指定されたメトリクスの集計 (シリーズ間のリデューサー) のリストを返します。
Aligners	指定されたメトリクスのアライナー (シリーズアライナーごと) のリストを返します。
Alignment periods	Grafana の Google Cloud Monitoring クエリエディタで使用できるすべての調整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サービスモニタリングサービスのリストを返します。
Service Level Objectives (SLO)	指定された 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

次の 2 つの構文があります。

- `$<varname>` 例: `metric.label.$metric_label`
- `[[varname]]` 例: `metric.label.[[metric_label]]`

2 つの方法があるのはなぜですか？ 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。複数值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 `[[varname]]` を使用する必要があります。

## 注釈

注釈を使用して、グラフの上に豊富な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にはコストがかかるため、返される行数を制限することが重要です。Google Cloud Monitoring の注釈とイベントはまだ表示できませんが、Google Cloud Monitoring の [カスタムメトリクス](#) でうまく機能します。

注釈の詳細については、「」を参照してください [注釈](#)。

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、クエリから返された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
<code>{{metric.value}}</code>	メトリクス/ポイントの値。	<code>{{metric.value}}</code>	555
<code>{{metric.type}}</code>	完全なメトリクスタイプを返します。	<code>{{metric.type}}</code>	<code>compute.googleapis.com/instance/cpu/utilization</code>
<code>{{metric.name}}</code>	メトリクス名の部分を返します。	<code>{{metric.name}}</code>	<code>instance/cpu/utilization</code>
<code>{{metric.service}}</code>	サービスパーツを返します。	<code>{{metric.service}}</code>	<code>compute</code>
<code>{{metric.label.xxx}}</code>	メトリクスラベル値を返します。	<code>{{metric.label.instance_name}}</code>	<code>grafana-1-prod</code>

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
<code>{{resource.label.xx}}</code>	リソースラベルの値を返します。	<code>{{resource.label.zone}}</code>	us-east1-b

Grafana パネルから Google クラウドコンソールの Metrics Explorer 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 クラウドコンソールの 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 クラウドコンソールで Metrics Explorer が開き、Grafana パネルからクエリが実行されます。リンクは、まずユーザーを Google アカウント選択に移動します。アカウントを正常に選択すると、ユーザーは Metrics Explorer にリダイレクトされます。提供されたリンクはどのアカウントでも有効ですが、アカウント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場合にのみクエリが表示されます。

## InfluxDB データソースに接続する

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. 上部のヘッダーにある Grafana アイコンを選択して、サイドメニューを開きます。
2. リnkの下側のサイドメニュー、ダッシュボードには、データソース という名前のリンクがあります。
3. 上部のヘッダーで + 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. 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ない場合は、現在のユーザーに Adminロールがないことを意味します。

## Jaeger データソースに接続する

Jaeger データソースは、オープンソースの end-to-end 分散トレースを提供します。

### データソースの追加

Jaeger の設定にアクセスするには、設定 (歯車) アイコンを選択し、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、および探索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、新しいパネルに対して事前に選択されることを意味します。
URL	Jaeger インスタンスの URL。例: <code>http://localhost:16686</code> 。
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。
Password	基本認証のパスワード。

### クエリトレース

Explore を使用して、Jaeger からトレースをクエリして表示できます。詳細については、「[探索](#)」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したり、トレースセレクトアからトレースを選択したりできます。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。

トレースセレクトアを使用して、Explore で選択した時間範囲に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。トレースセレクトアには 3 つのレベルのネストがあります: 1. 関心のあるサービス。1. 特定のオペレーションは、選択した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。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

内部リンクで派生フィールドを設定することで、Loki のログから Jaeger 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。

## Loki データソースに接続する

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. 上部にある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ない場合は、現在のユーザーに Admin ロールが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、および探索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、新しいパネルに対して事前に選択されることを意味します。

名前	説明
URL	Loki インスタンスの URL。例: <code>http://localhost:3100</code> 。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。
Maximum lines	Loki によって返されるログ行数の上限 (デフォルトは 1000)。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

派生フィールド設定を使用して、以下を実行できます。

- 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、ログからトレースバックエンドに直接リンクしたり、ログ行に `userId` がある場合はユーザープロフィー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。

派生した各フィールドは、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 — リンクが外部の場合は、リンク URL 全体を入力します。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。いずれの場合も、`${__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 — リンクが内部リンクか外部リンクかを選択します。内部リンクの場合、データソースセレクトを使用すると、ターゲットデータソースを選択できます。トレースデータソ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 の補間方法を確認できます。ログメッセージ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、Explore および視覚化のログパネルから利用できます。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、ログパネルにはログ行のリストと棒グラフが表示されます。X 軸には時刻、Y 軸には頻度/カウントが表示されます。

## 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、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られます。

```
{app="mysql",name="mysql-backup"}
```

現在、次のラベル一致演算子がサポートされています。

- = まったく同じです。
- != 等しくない。
- =~ regex-match。
- !~ regex-match は行いません。

例:

- {name=~"mysql.+"}
- {name!~"mysql.+"}

ラベルセレクトを追加するもう 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、各クエリにラベルセレクトを追加します。

## 検索式

ログストリームセレクトを記述した後、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、テキスト式または正規表現式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、式を順番にフィルタリングダウンします。結果のログ行はすべてのフィルターを満たします。

## 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ます。
- `!=` 行に文字列が含まれていません。
- `|~` 行は正規表現と一致します。
- `!~` 行が正規表現と一致しません。

### 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。

## ログコンテキスト

上記で説明した検索式を使用する場合、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送信されたログメッセージを調査できます。

## テンプレート

サーバー、アプリケーション、センサー名などのモノをメトリクスクエリでハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## 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 データベースを含む Microsoft SQL Server 2005 以降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。

ビジュアライゼーションを以前と同じように機能させるには、いくつかの手動移行が必要になる場合があります。解決策の 1 つは、[「Postgres/MySQL /MSSQL: 時系列クエリとデータ列の順序に関連する v8.0 の重大な変更」](#) で Github に記載されています。

## データソースの追加

1. 上部のヘッダーにある 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、Configuration にデータソースリンクがあります。
3. 上部のヘッダーで + 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、新しいパネルに対して事前に選択されることを意味します。
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 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 false (Grafana v5.4+)。
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 unlimited (Grafana v5.4+)。
Max idle	アイドル接続プール内の接続の最大数。デフォルト 2 (Grafana v5.4+)。
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。データを毎分書き込む1m場合など、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、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、(1分)や(30s30秒)など、1m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。次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

データソースの追加時に指定するデータベースユーザーには、クエリを実行する指定された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、`DELETE FROM user;`や `DROP TABLE user;`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取得し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、接続できるように暗号化を無効にする必要がある場合があります。可能な限り、最適な互換性を実現するために、利用可能な最新のサービスパックを使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、グラフ、Singlestat、またはテーブルパネルの編集モードのメトリクスタブにあります。編集モードに入るには、パネルタイトルを選択し、編集を選択します。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 パネルや 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、最初に使用できます。
5. 追加のクエリエディタが表示される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可能にするために、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
<code>\$_time(dateColumn)</code>	列の名前を時刻に変更する式に置き換えられます。例えば、dateColumn は時間です。

マクロの例	説明
<code>\$__timeEpoch(dateColumn)</code>	DATETIME 列タイプを Unix タイムスタンプに変換し、名前を時間に変更する式に置き換えられます。例えば、 <code>DATEDIFF(second, "1970-01-01", dateColumn)</code> AS time です。
<code>\$__timeFilter(dateColumn)</code>	指定された列名を使用する時間範囲フィルターに置き換えられます。例えば、 <code>dateColumn BETWEEN "2017-04-21T05:01:17Z" および "2017-04-21T05:06:17Z"</code> などです。
<code>\$__timeFrom()</code>	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、 <code>2017-04-21T05:01:17Z</code> などです。

マクロの例	説明
<code>\$__timeTo()</code>	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」などです。
<code>\$__timeGroup(dateColumn, '5m'[, fillvalue])</code>	GROUP BY 句で使用できる式に置き換えられます。fillValue を NULL または浮動小数点値に指定すると、時間範囲内の空のシリーズがその値で自動的に埋められます。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
<pre>\$__timeGroup(dateColumn, '5m', 0)</pre>	前と同じですが、fill パラメータが付いているため、そのシリーズの欠落ポイントは grafana によって追加され、0 が値として使用されます。
<pre>\$__timeGroup(dateColumn, '5m', NULL)</pre>	上記と同じですが、NULL が欠落ポイントの値として使用されます。
<pre>\$__timeGroup(dateColumn, '5m', previous)</pre>	上記と同じですが、まだ NULL が表示されていない場合は、そのシリーズの以前の値がフィル値として使用されます (Grafana 5.3+ 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された 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に設定されている場合、テーブルとしてフォーマットすると、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。

次のサンプルコードは、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')
```

次のサンプルコードは、クエリを示しています。

```
SELECT * FROM [mssql_types]
```

次のサンプルコードに示すように、通常の AS SQL 列選択構文を使用してテーブルパネル列の名前を制御できます。

```
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、グラフパネルで使用するために Format を時系列 に設定した場合、クエリには SQL 日時または Unix エポックを表す数値データ型を秒単位で返timeす 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 とを除くすべての列metric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ます。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 という名前の列を返す場合metric、この列はシリーズ名のプレフィックスとして使用されず。

時系列クエリの結果セットは、時間でソートする必要があります。

次のサンプルコードは、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1metric列を示しています。

```
SELECT
  time,
  valueOne,
  measurement as metric
FROM
  metric_values
WHERE
  $__timeFilter(time)
ORDER BY 1
```

上記のクエリをグラフパネルで使用するMetric Aと、とという名前Metric Bの2つのシリーズが生成され、にvalueOnevalueTwoプロットされますtime。

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
```

上記のクエリをグラフパネルで使用するMetric Aと、とという名前Metric Bの2つのシリーズが生成され、にvalueOnevalueTwoプロットされますtime。

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
```

前のクエリがグラフパネルで使用されると、Metric Aと という名前Metric Bの 2 つのシリーズが生成されvalueOne、 にvalueTwoプロットされますtime。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。

次のコード例は、fill パラメータをゼロに設定して \$\_\_timeGroup マクロを使用する方法を示しています。

```
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 という名前の 2 つのシリーズMetric Bになり、 にvalueTwoプロットされた の合計が表示されずtime。3 分間のウィンドウに値がないシリーズでは、値が 0 になり、グラフの右側にレンダリングされます。



## テンプレート

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレートと変数](#)。

### クエリ変数

タイプ のテンプレート変数を追加するとQuery、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のものを返すことができる MSSQL クエリを作成できます。

例えば、テンプレート変数 Query 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使用できます。

```
SELECT hostname FROM host
```

クエリは複数の列を返すことができ、Grafana は自動的にそれらからリストを作成します。例えば、次のクエリは、hostnameおよび の値を含むリストを返しますhostname2。

```
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および という名前の 2 つの列を返す必要があります\_\_value。\_\_text 列の値は一意である必要があります (一意でない場合、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。をテキストhostname、を値idとするクエリの例：

```
SELECT hostname __text, id __value FROM host
```

ネストされた変数を作成することもできます。例えば、 という名前の別の変数がある場合ですregion。次に、hosts 変数に、現在選択されているリージョンのホストのみを次のようなクエ

リで表示させることができます ( regionが複数値変数の場合は、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 の場合にのみ引用符で囲まれますmulti-value。

変数が複数値変数の場合は、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。

次の2つの構文があります。

`<varname>` という名前のテンプレート変数の例hostname :

```
SELECT
  atimestamp time,
  aint value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in($hostname)
ORDER BY atimestamp
```

`[[varname]]` という名前のテンプレート変数の例hostname :

```
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無効にする

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、 の形式になります 'server01', 'server02'。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}
```

可変フォーマットオプションの詳細については、「」を参照してください[テンプレートと変数](#)。

## 注釈

注釈を使用して、グラフの上に豊富な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。SQL のネイティブ日付/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。SQL のネイティブ日付/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	イベントタグに使用するオプションのフィールド名をカンマで区切った文字列として指定します。

次のサンプルコードは、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、で定義されているデータベーステーブルも使用します [時系列クエリ](#)。

次のサンプルコードは、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
```

次のサンプルコードは、エポック値を持つ時間および時間終了列を使用した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/時刻データ型の時間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は機能することが検証されています。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。列の同じ命名を使用し、それぞれのセクションで前述したのと同じ形式でデータを返す限り、ストアドプロシージャはテーブル、時系列、および注釈クエリでサポートされる必要があります。

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ています。列、valueOne、のすべての組み合わせなど、グラフパネルで4つのシリーズを視覚化するとvalueTwoしませmeasurement。右側のグラフパネルには、達成したいことが視覚化されています。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。

次のサンプルコードは、最初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
```

次のサンプルコードは、2番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プロシージャを定義できます。この場合、ストアプロシージャはint、データ型の@fromと@toの2つのパラメータを受け入れます。これは、ストア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(最初から最後まで)である必要があります。

これは、式によって選択およびグループ化\$\_\_timeGroup(time, '5m')されるを模倣しているため、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
```

その後、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プロシージャを定義できます。この場合、ストアプロシージャはdatetime、データ型の@fromと@toの2つのパラメータを受け入れます。これは、ストア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(最初から最後まで)である必要があります。

これは、式によって選択およびグループ化\$\_\_timeGroup(time, '5m')されるを模倣しているため、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
```

その後、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する必要があります。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 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を変更します。その結果、時系列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。

ビジュアライゼーションを以前と同じように機能させるには、いくつかの手動移行が必要になる場合があります。解決策の 1 つは、「[Postgres/MySQL/MSSQL: 時系列クエリとデータ列の順序に関連する v8.0 の重大な変更](#)」で [Github](#) に記載されています。

## データソースの追加

1. 上部のヘッダーにある 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. 上部のヘッダーで + データソースの追加ボタンを選択します。
4. タイプドリ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、新しいパネルに対して事前に選択されることを意味します。
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。

名前	説明
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 unlimited (Grafana v5.4+)。
Max idle	アイドル接続プール内の接続の最大数。デフォルト 2 (Grafana v5.4+)。
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+) で設定された <a href="#">wait_timeout</a> 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。データを毎分書き込む1m場合など、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、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、(1分)や(30s30秒)など、1m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。次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

識別子	説明
ms	ミリ秒

## データベースユーザーのアクセス許可

### Important

データソースの追加時に指定するデータベースユーザーには、クエリを実行する指定された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;`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を入力し、次に を編集します。

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された 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すると、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、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のテーブルまたはビューを選択するには、などの完全修飾名 (database.table) を手動で入力します otherDb.metrics。

Time 列フィールドは、時間値を保持する列の名前を参照します。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。

メトリクス列の候補には、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。メトリクス列とは異なるデータ型の列を使用する場合は、キャストを使用して列名を入力できます CAST(numericColumn as CHAR)。などのテキストデータ型に評価されるメトリクス列フィールドに任意の SQL 式を入力することもできます CONCAT(column1, " ", CAST(numericColumn as CHAR))。

## 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、などの列名の代わりに任意の式を記述できます column1 \* column2 / column3。

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 GROUP BY time と、エディタは自動的に 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、Columnメニューから 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、フィルターで を選択し、 を選択します Remove。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。

## グループ化の基準

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롭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。項目を選択して を選択すると、グループを削除できます Remove。

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が適用されている必要があります。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

時間別にグループ化すると、Grafana は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 番目の引数は Grafana で欠落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)

raw クエリエディタモードに切り替えるには、ハンバーガーアイコンを選択し、エディタモードを切り替えるを選択するか、クエリの下にある SQL の編集を選択します。

#### Note

raw 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可能にするために、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
<code>\$__time(dateColumn)</code>	UNIX タイムスタンプに変換し、列の名前をに変更する式に置き換えますtime_sec。 例えば、UNIX_TIME STAMP(dateColumn) をtime_secとして指定します。
<code>\$__timeEpoch(dateColumn)</code>	UNIX タイムスタンプに変換

マクロの例	説明
	<p>し、列の名前をに変更する式に置き換えますtime_sec。例え ば、UNIX_TIME STAMP(dateColumn) をtime_secとして指定します。</p>
<code>\$__timeFilter(dateColumn)</code>	<p>指定された列名を使用する時間範囲フィルターに置き換えられます。例え ば、dateColumn BETWEEN FROM_UNIX TIME(1494 410783) AND FROM_UNIX TIME(1494 410983) などで す。</p>
<code>\$__timeFrom()</code>	<p>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。</p>

マクロの例	説明
<code>\$__timeTo()</code>	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(1494410983) などです。
<code>\$__timeGroup(dateColumn, '5m')</code>	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)、*
<code>\$__timeGroup(dateColumn, '5m', 0)</code>	前の行と同じですが、fill パラメータを指定すると、そのシリーズの欠落したポイントは 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
<code>\$__timeGroup(dateColumn, '5m', NULL)</code>	上記と同じですが、NULL が欠落ポイントの値として使用されます。
<code>\$__timeGroup(dateColumn, '5m', previous)</code>	上記と同じですが、まだ NULL が表示されていない場合は、そのシリーズの以前の値がフィル値として使用されます (Grafana 5.3+ でのみ使用可能)。
<code>\$__timeGroupAlias(dateColumn, '5m')</code>	と同じ \$__timeGroup ですが、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
<code>\$__unixEpochFilter(dateColumn)</code>	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に置き換えられます。例えば dateColumn > 1494410783 AND dateColumn < 1494497183 です。
<code>\$__unixEpochFrom()</code>	現在アクティブな時間選択の開始を Unix タイムスタンプに置き換えます。例えば 1494410783 です。
<code>\$__unixEpochTo()</code>	現在アクティブな時間選択の最後に Unix タイムスタンプに置き換えられます。例えば 1494497183 です。

マクロの例	説明
<code>\$__unixEpochNanoFilter(dateColumn)</code>	指定された列名を使用し、時刻をナノ秒タイムスタンプとして表す時間範囲フィルターに置き換えられます。例えば dateColumn > 149441078315241521 4 AND dateColumn < 1494497183142514872 です。
<code>\$__unixEpochNanoFrom()</code>	現在アクティブな時間選択の開始をナノ秒のタイムスタンプに置き換えます。例えば 1494410783152415214 です。

マクロの例	説明
<code>\$__unixEpochNanoTo()</code>	現在アクティブな時間選択の最後にナノ秒タイムスタンプに置き換えられます。例えば 1494497183142514872 です。
<code>\$__unixEpochGroup(dateColumn,"5m", [fillmode])</code>	と同じ <code>\$__timeGroup</code> ですが、Unix タイムスタンプとして保存されている時間 (Grafana 5.3 以降でのみ使用可能) の場合も同様です。
<code>\$__unixEpochGroupAlias(dateColumn,"5m", [fillmode])`</code>	上記と同じですが、列エイリアスも追加します (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された SQL 文字列が展開されて表示されます。

## テーブルクエリ

クエリとしてフォーマット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)
```

通常の `as` SQL 列選択構文を使用して、テーブルパネル列の名前を制御できます。

## 時系列クエリ

`Format` を時系列に設定した場合、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 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` とを除くすべての列 `metric` 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 `time` という名前の列を返す場合 `metric`、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+ でのみ使用可能)。

時系列クエリの結果セットは、時間で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、代わりに 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についてはサポートされていません。

## テンプレート

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレート](#)。

## クエリ変数

タイプのテンプレート変数を追加すると Query、測定名、キー名、キー値などの項目を返す MySQL クエリを記述できます。このクエリは、ドロップダウン選択ボックスとして表示されます。

例えば、テンプレート変数 Query 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使用できます。

```
SELECT hostname FROM my_host
```

クエリは複数の列を返すことができ、Grafana は自動的にそれらからリストを作成します。例えば、次のクエリは、hostname および の値を含むリストを返します hostname2。

```
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 `$__timeFilter(column)` で などの時間範囲に依存する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` および という名前の 2 つの列を返す必要があります `__value`。 `__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、わかりやすい名前をテキストとして、ID を値として使用できます。

次のコード例は、 をテキスト hostname、 を値 id とするクエリを示しています。

```
SELECT hostname AS __text, id AS __value FROM my_host
```

ネストされた変数を作成することもできます。例えば、 という名前の別の変数がある場合です region。次に、hosts 変数に、現在選択されているリージョンのホストのみを、次のようなクエリで表示させることができます ( region が複数値変数の場合、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\_\_searchFilterで を使用すると、ドロップダウン選択ボックスでユーザーが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。ユーザーによって何も入力されていない場合、 のデフォルト値は \_\_searchFilterです%。

### Note

Grafana ではこれが行われなため、 \_\_searchFilter式を引用符で囲むことが重要です。

次の例は、 をクエリフィールド\_\_searchFilterの一部として使用して検索を有効にする方法を示しています。hostnameドロップダウン選択ボックスのユーザータイプ。

```
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、テンプレート変数を where 句の引用符で囲まないでください。

Grafana 4.7.0 から、テンプレート変数の値は、テンプレート変数が の場合にのみ引用符で囲まれますmulti-value。

変数が複数値変数の場合は、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。

次の 2 つの構文があります。

`<varname>` という名前のテンプレート変数の例hostname :

```
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]]` という名前のテンプレート変数の例hostname :

```
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無効にする

Grafana は、複数値変数の引用符で区切られたカンマ区切りの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、 の形式になります 'server01', 'server02'。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}
```

可変フォーマットオプションの詳細については、「」を参照してください [高度な変数形式のオプション](#)。

### 注釈

注釈を使用して、グラフの上に豊富な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。

次のサンプルコードは、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)
```

次のサンプルコードは、エポック値を持つ時間および時間終了列を使用した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/時刻データ型の時間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。SQL 日付/時刻のネイティブデータ型またはエポック値の列を指定できません。
timeend	終了日時フィールドのオプション名。SQL 日付/時刻のネイティブデータ型またはエポック値の列を指定できます。
text	イベントの説明フィールド。

名前	説明
tags	イベントタグにカンマ区切りの文字列として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する必要があります。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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

Amazon Managed Grafana では、データソースとしてオープンソース [OpenSearch](#) (またはレガシー Elasticsearch) を追加できます。に保存されているログやメトリクスを視覚化するには、さまざまなタイプの単純な OpenSearch クエリまたは複雑なクエリを実行できます OpenSearch。に保存されたログイベントで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 OpenSearch 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. 上部のヘッダーにある 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. 上部のヘッダーで + 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、新しいパネルに対して事前に選択されることを意味します。
Url	OpenSearch サーバーの HTTP プロトコル、IP、およびポート。
Access Mode	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できる必要があります。

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。他に何も記載されていない場合は、サーバーが推奨される方法です。

### 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対して行われ、リクエストはデータソースに転送され、Cross-Origin Resource Sharing (CORS) の要件を回避します。このアクセスモードを選択した場合、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、データソースへの OpenSearch ブラウザ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` のデフォルトを指定 `time field` 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/Elasticsearch バージョン

バージョンドロップダウンメニューで、OpenSearch またはレガシーの 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。Elasticsearch でサポートされているバージョンは 2.0+、5.0+、5.6+、6.0+、および 7.0+ です。値は、バージョン 5.6 以降で 6.0 より小さい 5.6+ を意味します。値は、バージョン 6.0 以降、7.0 未満 6.0+ を意味します。最後に、7.0+ はバージョン 7.0 以降で、8.0 より前のことを意味します。

### 最小時間間隔

時間間隔による自動グループの下限。書き込み頻度に設定することをお勧めします。例えば、データが 1 分ごとに書き込まれる 1m 場合などです。このオプションは、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、(1 分) や (30s30 秒) など、1m 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。次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

Message field name との 2 つのパラメータは Level field name、オプションでデータソース設定ページから設定できます。このページでは、で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するフィールドを決定します [探索](#)。

例えば、へのログの配信に Filebeat のデフォルト設定を使用する場合 OpenSearch、次の設定が機能します。

- メッセージフィールド名 : message
- 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 – リンクが外部の場合は、リンク URL 全体を入力します。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。いずれの場合も、`${__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。リンクが内部リンクの場合、データソースセレクトアを使用すると、ターゲットデータソースを選択できます。トレースデータソ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グループバイ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、メトリクスまたはグループをオプション別に表示および編集します。

### シリーズの命名パターン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
<code>{{term fieldname}}</code>	Group By という用語の値に置き換えました。

パターン	説明
<code>{{metric}}</code>	メトリクス名 (例 : 平均、最小、最大)。
<code>{{field}}</code>	をメトリクスフィールド名に置き換えました。

## パイプラインメトリクス

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。例えば、移動平均 や派生 などです。OpenSearch パイプラインメトリクスには、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

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## クエリ変数

OpenSearch データソースは、Query 変数の Query 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されます。

Query	説明
<pre>{"find": "fields", "type": "keyword"}</pre>	インデックスタイプ のフィールド名のリストを返します keyword。
<pre>{"find": "terms", "field": "@hostname", "size": 1000}</pre>	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、現在のダッシュボードの時間範囲をクエリの時間範囲として使用します。
<pre>{"find": "terms", "field": "@hostname"}</pre>	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、現在のダッシュボードの時間範囲をクエリの時間範囲として使用します。

Query	説明
<pre>e", "query": '&lt;.lucene query&gt;']}</pre>	

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、クエリで `size` プロパティを設定します。クエリ内で他の変数を使用できます。次のコード例は、 という名前の変数のクエリ定義を示しています `$host`。

```
{"find": "terms", "field": "@hostname", "query": "@source:$source"}
```

前の例では、クエリ定義 `$source` 内で という名前の別の変数を使用します。 `$source` 変数の現在の値を変更するたびに、ドロップダウンリストから `$host` 変数の更新が開始されます。更新後、 `$host` 変数には、 によってフィルタリングされた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です。

これらのクエリは、デフォルトでは語順で返されます (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語句のリスト (上位 N 個の値リスト) を生成するには、 `orderBy` プロパティを追加します `doc_count`。これにより、降順のソートが自動的に選択されます。 `doc_count` (最下位 N リスト) `asc` で を使用するには、 を設定しますが `order: "asc"`、ドキュメント数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語句を保持するには、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

次の 2 つの構文があります。

- `$<varname>` 例: `@hostname:$hostname`
- `[[varname]]` 例: `@hostname:[[hostname]]`

2 つの方法があるのはなぜですか？ 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。複数值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、 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\$hostname。また、フィールド入力による用語グループの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、グラフの上に豊富な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、任意の OpenSearch インデックスに注釈イベントをクエリできます。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ことも、Lucene クエリを指定することもできます。
Time	時間フィールドの名前。 は日付フィールドである必要があります。
TimeEnd	時間終了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、注釈は時間と終了の間のリージョンとしてマークされます。
Text	イベントの説明フィールド。
Tags	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 文字列を使用できます)。

## ログのクエリ

からのログデータのクエリと表示 OpenSearch については、Explore を参照してください。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、「[探索](#)」を参照してください。

## ログクエリ

結果が返されると、ログパネルにはログ行のリストと棒グラフが表示されます。X 軸には時刻、Y 軸には頻度またはカウントが表示されます。



## 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します。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 には、OpenTSDB の高度なサポートが付属しています。

### データソースの追加

1. 上部のヘッダーにある Grafana アイコンを選択して、サイドメニューを開きます。
2. Dashboards リンクのサイドメニューには、データソースリンクがあります。
3. 上部のヘッダーで + 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ない場合は、現在のユーザーに Admin ロールが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、パネルとクエリでデータソースを表示する方法です。
Defau	デフォルトのデータソースは、新しいパネルに対して事前に選択されることを意味します。
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 または 2.2。
Resol n	opentsdb のメトリクスには、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  $\leq 2.1$  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ときは、フィルターまたはタグのいずれかを使用してください。これらは相互に排他的です。一緒に使用すると、異常な結果が得られる可能性があります。

## オートコンプリート提案の使用

メトリクス名、タグ名、タグ値 の入力を開始するとすぐに、自動完了候補が強調表示されます。オートコンプリートは、OpenTSDB サジェスト 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化

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレートと変数](#)。

## 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、タグ値などです。

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。

Query	説明
<code>metrics(prefix)</code>	特定のプレフィックスを持つメトリクス名を返します (空でもかまいません)。
<code>tag_names(cpu)</code>	特定の cpu メトリクスのタグ名 (キー) を返します。
<code>tag_values(cpu, hostname)</code>	メトリクス cpu とタグキーホスト名のタグ値を返します。
<code>suggest_tagk(prefix)</code>	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 (空にすることもできます)。
<code>suggest_tagv(prefix)</code>	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 (空でもかまいません)。

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 `tsd.core.meta.enable_realtime_ts` で 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 `tsdb uid metasync` で を実行する必要があります。

## ネストされたテンプレート

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。最初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ように、いくつかの例を以下に示します。

Query	説明
<code>tag_values(cpu, hostname, env=\$env)</code>	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。
<code>tag_values(cpu, hostname, env=\$env, region=\$region)</code>	cpu メトリクスのタグ値、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 では、Postgres、MySQL、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ています。その結果、時系列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。ビジュアライゼーションをバージョン 8 以前と同じように機能させるには、いくつかの手動移行が必要になる場合があります。解決策の 1 つは、「[Postgres/MySQL /MSSQL: 時系列クエリとデータ列の順序に関連する v8.0 の重大な変更](#)」で [Github](#) 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、接続方法の変更が必要になる場合があります。詳細については、「[更新されたワークスペースに関する問題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. 上部のヘッダーにある Grafana アイコンを選択して、サイドメニューを開きます。
2. 設定アイコンの下のサイドメニューに、データソースリンクがあります。
3. 上部のヘッダーで + 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、パネルとクエリでデータソースを表示する方法です。

名前	説明
Default	デフォルトのデータソースは、新しいパネルに対して事前に選択されることを意味します。
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。データベース名は含めないでください。Postgres に接続するための接続文字列が正しくないため、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 unlimited (Grafana v5.4+)。
Max idle	アイドル接続プール内の接続の最大数。デフォルト 2 (Grafana v5.4+)。
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 (Grafana v5.4+)。
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)。

名前	説明
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time_bucket で 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。データを毎分書き込む1m場合など、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、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、(1分) や (30s30秒) など、1m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。次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

データソースの追加時に指定するデータベースユーザーには、クエリを実行する指定された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、`DELETE FROM user;`やなどのステートメント `DROP TABLE user;`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取得し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、グラフまたは 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、編集します。

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された 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すると、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、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。にないテーブルまたはビューを選択するには、`search_path` などの完全修飾名 (`schema.table`) を手動で入力できます `public.metrics`。

Time 列フィールドは、時間値を保持する列の名前を参照します。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。

メトリクス列の候補には、テキストデータ型 (char、varchar、text) 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の列を使用するには、キャストを使用して列名を入力しますip::text。などのテキストデータ型に評価されるメトリクス列フィールドに任意の SQL 式を入力することもできますhostname || ' ' || container\_name。

## 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、などの列名の代わりに任意の式を記述できますcolumn1 \* column2 / column3。

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 の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 を追加しますGROUP BY time。

エディタは、クエリのこの部分を簡素化して統合しようとしています。

さらに値列を追加するには、プラスボタンを選択し、メニューから列を選択します。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、フィルターを選択し、 の削除を選択します。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。

## グループ化の基準

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、 の削除 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が適用されている必要があります。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 番目の引数は Grafana で欠落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)

raw クエリエディタモードに切り替えるには、ハンバーガーアイコンを選択し、エディタモードを切り替えるか、クエリの下にある SQL の編集を選択します。

### Note

raw 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

マクロはクエリ内で使用して構文を簡素化し、動的部分を可能にします。

マクロの例	説明
<code>\$__time(dateColumn)</code>	UNIX タイムスタンプに変換し、列の名前を変更する式に置き換えられます <code>time_sec</code> 。 例えば、 <code>UNIX_TIMESTAMP(dateColumn)</code> を <code>time_sec</code> として指定します。
<code>\$__timeEpoch(dateColumn)</code>	UNIX タイムスタンプに変換し、列の名前を変更する式に置き換えられます <code>time_sec</code> 。 例えば、 <code>UNIX_TIMESTAMP</code>

マクロの例	説明
<code>\$__timeFilter(dateColumn)</code>	STAMP(dateColumn) を time_sec として指定します。  指定された列名を使用する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。
<code>\$__timeFrom()</code>	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
<code>\$__timeTo()</code>	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(1494410983) などです。
<code>\$__timeGroup(dateColumn, '5m')</code>	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)、*
<code>\$__timeGroup(dateColumn, '5m', 0)</code>	前の行と同じですが、fill パラメータを指定すると、そのシリーズの欠落したポイントは 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
<code>\$__timeGroup(dateColumn, '5m', NULL)</code>	上記と同じですが、NULL が欠落ポイントの値として使用されます。
<code>\$__timeGroup(dateColumn, '5m', previous)</code>	上記と同じですが、まだ NULL が表示されていない場合は、そのシリーズの以前の値がフィル値として使用されます (Grafana 5.3+ でのみ使用可能)。
<code>\$__timeGroupAlias(dateColumn, '5m')</code>	<code>\$__timeGroup</code> と同じですが、列エイリアスが追加されます
<code>\$__unixEpochFilter(dateColumn)</code>	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に置き換えられます。例えば、 <code>*dateColumn &gt; 1494410783 AND dateColumn &lt; 1494497183*</code>

マクロの例	説明
<code>\$__unixEpochFrom()</code>	現在アクティブな時間選択の開始を Unix タイムスタンプに置き換えます。例えば、*1494410783*
<code>\$__unixEpochTo()</code>	現在アクティブな時間選択の最後に Unix タイムスタンプに置き換えられます。例えば、*1494497183*
<code>\$__unixEpochNanoFilter(dateColumn)</code>	指定された列名を使用し、時刻をナノ秒タイムスタンプとして表す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*

マクロの例	説明
<code>\$__unixEpochNanoFrom()</code>	現在アクティブな時間選択の開始をナノ秒のタイムスタンプに置き換えます。例えば、*1494410783152415214*
<code>\$__unixEpochNanoTo()</code>	現在アクティブな時間選択の最後にナノ秒タイムスタンプに置き換えられます。例えば、*1494497183142514872*
<code>\$__unixEpochGroup(dateColumn,"5m", [fillmode])</code>	<code>\$__timeGroup</code> と同じですが、Unix タイムスタンプとして保存される時間。

## 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。

通常の `as` SQL 列選択構文を使用して、テーブルパネル列の名前を制御できます。

## 時系列クエリ

Format を に設定した場合 Time series、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 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` とを除くすべての列 `metric` 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 という名前の列を返す場合 `metric`、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、時間で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
```

次のコード例では、`$__timeGroup` マクロの `fill` パラメータを使用して、代わりに `null` 値をゼロに変換しています。

```
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

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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。

テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレート](#)。

## クエリ変数

タイプのテンプレート変数を追加するとQuery、測定名、キー名、キー値などの項目を返すPostgreSQL クエリを記述できます。このクエリはドロップダウン選択ボックスとして表示されません。

例えば、テンプレート変数 Query 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使用できます。

```
SELECT hostname FROM host
```

クエリは複数の列を返すことができ、Grafana は自動的にそれらからリストを作成します。例えば、次のクエリは、hostnameおよび の値を含むリストを返しますhostname2。

```
SELECT host.hostname, other_host.hostname2 FROM host JOIN other_host ON host.city =
other_host.city
```

クエリ\$\_\_timeFilter(column)で などの時間範囲に依存する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`および `__value` という名前の 2 つの列を返す必要があります。`__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。をテキスト `hostname`、を値 `id` とするクエリの例：

```
SELECT hostname AS __text, id AS __value FROM host
```

ネストされた変数を作成することもできます。という名前の変数を使用すると `region`、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、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 `__searchFilter` を使用すると、ドロップダウン選択ボックスでユーザーが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。ユーザーによって何も入力されていない場合、のデフォルト値は `__searchFilter` です。

#### Note

Grafana ではこれが行われないため、`__searchFilter` 式を引用符で囲むことが重要です。

次の例は、をクエリフィールド `__searchFilter` の一部として使用して検索を有効にする方法を示しています。 `hostname` ドロップダウン選択ボックスのユーザータイプ。

```
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 の場合にのみ引用符で囲まれます `multi-value`。

変数が複数値変数の場合は、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。

次の2つの構文があります。

`<varname>` という名前のテンプレート変数の例hostname :

```
SELECT
  atimestamp as time,
  aint as value
FROM table
WHERE $__timeFilter(atimestamp) and hostname in($hostname)
ORDER BY atimestamp ASC
```

`[[varname]]` という名前のテンプレート変数の例hostname :

```
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無効にする

Amazon Managed Grafana は、複数値変数の引用符で区切られたカンマ区切りの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、の形式になります'server01', 'server02'。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}
```

可変フォーマットオプションの詳細については、「」を参照してください[テンプレートと変数](#)。


注釈

注釈を使用して、グラフの上に豊富な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。

次のサンプルコードは、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)
```

次のサンプルコードは、エポック値を持つ時間および時間終了列を使用した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/時刻データ型の時間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。SQL 日付/時刻のネイティブデータ型またはエポック値の列を指定できます。
timeend	終了日時フィールドのオプション名。SQL のネイティブ日付/時刻データ型またはエポック値 (Grafana v6.6+) を持つ列を指定できます。
text	イベントの説明フィールド。
tags	イベントタグにカンマ区切りの文字列として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する必要があります。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 データソースに接続する

Tempo は、Grafana Labs の、大量の最小限の依存関係トレースストレージである 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、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、および探索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、新しいパネルに対して事前に選択されることを意味しません。
URL	Tempo インスタンスの URL。例: <code>http://tempo</code> 。

名前	説明
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。
Password	基本認証のパスワード。

## ログへのトレース

これは、トレースからログへの機能の設定です。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 クエリで使用されるタグ。デフォルトは `'cluster', 'hostname', 'namespace', 'pod'`
- スパン開始時刻シフト — スパン開始時刻に基づいて Loki クエリの開始時刻をシフトします。過去まで拡張するには、負の値を使用する必要があります。ここで時間単位を使用できます。例えば、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て Loki クエリの終了時刻をシフトします。ここで時間単位を使用できます。例えば、5s、1m、3h などです。デフォルトは 0 です。

## クエリトレース

Explore を使用して、Tempo からトレースをクエリして表示できます。データソース設定ページでトレースをログ設定に設定すると、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には、Loki クエリエディタを使用します。検索結果を取得するには、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、その ID をトレース 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ことで、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。

## テスト用の TestData データソースを設定する

Grafana には、任意のパネルのシミュレートされた時系列データを作成する TestData データソースが付属しています。これを使用して、独自のフェイクでランダムな時系列データを構築し、任意のパネルにレンダリングできるため、ダッシュボードの機能を検証し、データを安全かつ簡単に共有できます。

### データソースを設定する

のデータソース設定にアクセスするには TestData

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. を選択します TestData。

データソースは、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を提供するわけではありません。

名前	説明
Name	パネル、クエリ、Explore 内のデータソースの名前。
Default	このデータソースを新しいパネル用に事前に選択するかどうか。

### モックデータを作成する

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、任意のメトリクスパネルでデータソースとして使用でき、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を提供します。

### シナリオを選択する

データソースは、クエリエディタを提供する代わりに、パネルのシミュレートされたデータを生成するシナリオを選択する TestData のに役立ちます。

各シナリオにエイリアスを割り当てることができ、多くの には、選択したときに独自のオプションが表示されます。

## 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
- エクスポネンシャル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 データをロードする
- ログ
- データポイントなし
- ノードグラフ
- 予測可能な CSV Wave
- 予測可能な脈
- ランダムウォーク
- Random Walk (エラーあり )
- ランダムウォークテーブル
- Raw 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
- 米国で生成されたデータ

事前設定されたダッシュボードをインポートする

TestData は、ダッシュボードの例も提供します。

## サンプルダッシュボードをインポートするには

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. Simple Streaming サンプルダッシュボードのインポートを選択します。

インポートしたダッシュボードをカスタマイズするには：

インポートしたダッシュボードをカスタマイズするには、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 をアップグレードすると、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する

時系列データの使用またはレンダリング GrafanaLabs GitHub に関する問題を に報告する場合は、このデータソースを使用して問題をレプリケートすることを強くお勧めします。これにより、デベロッパーが問題をレプリケートして解決することが容易になります。

## Zipkin データソースに接続する

Zipkin はオープンソースの分散トレースシステムです。Amazon Managed Grafana の Explore でトレースをクエリできるようにする Zipkin データソースを追加する

### 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、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、および探索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、新しいパネルに対して事前に選択されることを意味します。
URL	Zipkin インスタンスの URL。例: <code>http://localhost:9411</code> 。
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。



名前	説明
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。
Password	基本認証のパスワード。

## クエリトレース

Zipkin からのトレースのクエリと表示は、Explore から入手できます。

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したり、トレースセレクトタからトレースを選択したりできます。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。

トレースセレクトタを使用して、Explore で選択した時間範囲に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。トレースセレクトタには 3 つのレベルのネストがあります: 1. 関心のあるサービス。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。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 注釈は、注釈キーの下に表示される注釈値を持つ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

内部リンクで派生フィールドを設定することで、Loki のログから Zipkin 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。

## 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を管理する](#)」を参照してください。

エンタープライズプラグインは定期的に更新されます。これには、既存のプラグインの更新と、場合によっては新しいデータソースの更新の両方が含まれます。以下のドキュメントには、利用可能なすべてのデータソースが含まれているとは限りません。Amazon Managed Grafana Enterprise プラグ

インでサポートされている現在のエンタープライズ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の場合、エンタープライズデータソースはデフォルトでインストールされなくなります。正しい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、すべてのエンタープライズデータソースにプラグインをインストールできます。インストール済みのプラグインのバージョンを更新することもできます。プラグインの管理の詳細については、「[」](#)を参照してください[プラグインを使用してワークスペースを拡張する](#)。

## 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 データソースに接続する](#)
- [Honeycomb データソースに接続する](#)
- [Jira データソースに接続する](#)
- [MongoDB データソースに接続する](#)
- [New Relic データソースに接続する](#)
- [Oracle Database データソースに接続する](#)
- [Salesforce データソースに接続する](#)
- [SAP HANA データソースに接続する](#)
- [データソースに接続する ServiceNow](#)
- [Snowflake データソースに接続する](#)
- [Splunk データソースに接続する](#)
- [Splunk インフラストラクチャモニタリングデータソースに接続する](#)
- [Wavefront データソースに接続する \(Wavefront による VMware Tanzu Observability\)](#)

## 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 AppDynamics を使用して 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

**データソース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しないようにするため) と基本認証を使用します。ユーザー名は「user@account」 (your.name@customer1 または my\_user@saas\_account\_name) であることに注意してください。

次のステップを使用して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は表示ボタンがあります。アクセスキーを表示するには、表示ボタンを選択します。
5. アクセスキーを Grafana の設定の基本的な認証の詳細ページのパスワードフィールドにコピーします。

次の手順を使用して、Amazon Managed Grafana のユーザーとロールを設定します。

1. で AppDynamics、設定、管理に移動します。
2. ロールタブを選択し、「+」ボタンを選択して新しいロールを作成します。例えば、grafana\_readonly。
3. 「ロールの作成」セクションの「アカウント」タブで、アクセス許可を追加します View Business Flow。
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 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、新しいユーザーを作成します。例えば、 などで grafana。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に割り当てます grafana\_readonly。例えば、

## テンプレート

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 BTs)
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 BTs)
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を指定できます)

## 凡例キー

凡例キーのデフォルトはかなり長くすることができますが、このフォーマットはカスタマイズできません。

凡例キーには、App on legend オプションを選択してアプリケーション名のプレフィックスを付けることができます。例: MyApp - Overall Application Performance | Average Response Time (ms)。

クエリが凡例キーを表示できない singlestat または他のパネルに対するものである場合は、メタデータを表示 オプションを選択して、クエリの凡例キー (エイリアスとも呼ばれます) を確認します。

凡例ドロップダウンリストには、Full Path、 、 Segments の 3 つのオプションがあります Custom。

## 凡例オプション - フルパス

凡例キーは、 などの完全なメトリクスパスです Overall Application Performance | Average Response Time (ms)。

## 凡例オプション – セグメント

メトリクス名はセグメントで構成されます。表示するセグメントを選択できます。

例えば、メトリクス名の場合：

```
Errors|mywebsite|Error|Errors per Minute
```

セグメントフィールドに以下を入力する2,4と、 が返されますmywebsite|Errors per minute。

インデックス作成は 1 で始まるため、 1を返しますErrors。

## 凡例オプション – カスタム

テキストと次のエイリアスパターンを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。 は最初のセグメント`{{1}}`を返し、 は 2 番目のセグメント`{{2}}`を返します。

生成される凡例キーパターンと凡例キーの例：

- `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。クエリをフォーマットして色分けする SQL エディタが含まれています。

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## Databricks データソースの追加

Grafana コンソールに Databricks データソースを追加するには、次の手順に従います。

Databricks データソースを追加するには

1. 上部のヘッダーにある 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 Dashboards リンクで、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ない場合は、Grafana のAdminロールがありません。

3. 上部のヘッダーで + 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 オプションが表示されず、それが必要な場合は、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

時系列の視覚化は、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。フィールドに特定のタイムゾーンが含まれていない場合、Grafana は時刻が UTC であると想定します。

### 複数行の時系列

複数行の時系列の視覚化を作成するには、クエリに次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 のdatetimeフィールドtime。
2. への値GROUP BY。
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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## 使用方法

### キャッシュ

大規模なダッシュボードでは、多くのクエリを Datadog API によってレート制限することができません (Datadog API が許可する 1 時間あたりの API 呼び出しの最大数に達します)。キャッシュ機能は、一意のクエリを 60 秒間キャッシュします。この間隔は、設定ページで長くまたは短く変更できます。

### クエリエディタ

簡単に - 集計とメトリクスを選択します。結果をフィルタリングする場合は、1 つ以上のタグを選択します。

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選択し、関数名を選択して配置します。

フィールドの使用可能性別のエイリアス：

- エイリアスを「別名」フィールドに入力します。
- スコープ付き変数を使用する：
  - `$__metric` = メトリクス名に置き換えられます
  - `$__display_name` = メトリクス名に置き換えられます
  - `$__expression` = 完全なメトリクス式に置き換えられます
  - `$__aggr` = メトリクス集計関数に置き換えられます (例: avg、max、min、sum)
  - `$__scope` = メトリクススコープ (リージョン、サイト、環境、ホストなど) に置き換えられます
- 正規表現を使用します。
  - 正規表現を「エイリアス RegExp」フィールドに `/you regexp here/flags` 形式で入力します。
  - 「別名」フィールドが空の場合、RegExp 結果は を使用して結合されます。メトリクス式 = `avg:system.load.5{*}:"Alias by"` フィールド入力: `""Alias RegExp"` フィールド入力: `avg:(.+)\.(\d)` 結果: `system.load, 5`
  - 「別名」フィールドで `$<group_number>` 変数を使用します。メトリクス式 = `avg:system.load.5{*}:"Alias by"` 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。メトリクス式 = `datadog.dogstatsd.packet.count{*}:"Alias by"` 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。

## メトリクス算術

メトリクス算術を使用するには、クエリタイプを算術 に設定します。# 署名を使用して、必要なメトリクスにリンクします。例えば、`#A * 2`はクエリ の結果を 2 倍にしますA。2つのメトリクス間



の算術は同じ方法で機能します。計算に使用する結果を含むクエリを追加し、3番目のクエリでなどのこれらのメトリクスにリンクします#A / #B。

## 注釈

アノテーションは、グラフの上にオーバーレイされるイベントです。イベントの例は、デプロイまたは停止です。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、Amazon Managed Grafana のグラフにオーバーレイできます。注釈イベントは、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

テンプレート変数の値を取得するには、メトリクスとタグといういくつかのオプションがあります。クエリフィールド\*で指定された使用可能なメトリクスのリストを取得するには。

すべてのタグを返すには、値 tagまたは を使用しますscope。

指定されたタググループのタグを返すには、次のいずれかのデフォルトのカテゴリ値を使用します。

- 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 の場合subscription\_name、クエリフィールドにそのタググループ名を入力します。

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、複数值変数がサポートされます。選択した複数のタグ値が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入使できます。Datadog のアドホック変数は、などのタグからすべてのキーと値のペアを取得し region: east, region: west、クエリタグとして使します。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/> のデータソース。このデータソースを使用するには、Dynatrace アカウントが必要です。

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## 既知の制限事項

テンプレート変数を複数選択することはできません。単一選択のみがサポートされています。

v2 メトリクス APIs 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 (複数選択なし)
  - 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
  - アラートのフルサポート

## Dynatrace 固有の機能

Dynatrace メトリクス v2 API を使用して、組み込み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 のドキュメント [「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している場合があります。

## Dynatrace アクセス許可

Dynatrace で次のアクセス許可が必要です - API V2 を使用したメトリクスの読み取り (metrics.read) アクセス許可 - API V2 を使用したエンティティの読み取り (entities.read) アクセス許可

## 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 および アクセス entities.read 許可を設定します。

## 構成

1. 論理 Grafana サーバー UI 内の設定/データソースを選択し、データソースの追加 を選択します。
2. データソースの追加ページで、Dynatrace をフィルタリングし、Dynatrace プラグインを選択します。
3. Dynatrace データソースを設定するには、次のパラメータが必要です。
  - 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 インスタンスのタイプ。これは SaaS または Managed Cluster のいずれかになります。

- Dynatrace API Token - これは前のステップで生成した API トークンです。

次の 2 つの設定は、Dynatrace SaaS が マネージドかによって異なります。

- SaaS の の例ではyfc55578.live.dynatrace.com、環境 ID は になりますyfc55578。
- の マネージド例ではyd8888.managed-sprint.dynalabs.io/  
e/abc99984-3af2-55tt-72k1-0672983gc45、環境 ID は にな  
りabc99984-3af2-55tt-72k1-0672983gc45、ドメインは になります。  
yd8888.managed-sprint.dynalabs.io

4. すべての設定値を設定したら、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。

## データソースをクエリする

クエリエディタを使用して、Dynatrace のメトリクスと問題をクエリします。クエリタイプは metricまたは ですproblem。

## 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できます。その後、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、[「マージ変換」](#)を参照してください。
- Filters— Dynatrace データソースは、各メトリクスの適切な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 クエリエディタでフィルターラベルの横にある + 記号を選択し、フィルタリングするフィールドを選択し、使用する演算子を選択し、フィルタリングする値を選択します。Dynatrace データソースを使用すると、複雑な論理比較を作成するために結合できるフィルターグループを作成できます。ほとんどのユースケースでは、フィルターグループは必須ではありません。タグを使用してフィルターを作成する場合、選択した組み合わせに関係なく、Dynatrace は常に AND を使用します。Dynatrace は、タグ付きの OR フィルターをサポートしていません。
- Alias— Dynatrace データソースの使用中に発生するエイリアスには 2 つの異なるタイプがあります。1 つ目は静的エイリアスです。このタイプのエイリアスは、構築するすべてのクエリで使用でき、エイリアスの名前は小文字で始まります。2 つ目は動的エイリアスで、クエリで使用してい

るメトリクスに基づいて変更され、エイリアスの名前は大文字で始まります。Dynatrace プラグインは、Metric Names、Aggregation、Display Name および Host の複数の異なるエイリアスをサポートしています。Description。

名前	値
\$name	builtin:apps.other.keyUserActionsreportedErrorCount.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。フィードの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— ステータスで結果の問題をフィルタリングします。
- Impact Filter- 影響レベルで結果の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は、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を追加する」](#)を参照してください。Dynatrace データソースを、次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。

クエリタイプ	説明
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領域のリストを返します。
Problem severity options	すべての問題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除外します。

#### 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。

変数を作成したら、メトリクスドロップダウンメニューで確認できます。

### 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、「」を参照してください[ダッシュボードのインポート](#)。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 データソースの選択 > ダッシュボードタブを選択すると、使用可能な事前作成済みダッシュボードが表示されます。

### GitLab データソースに接続する

GitLab データソースを使用すると、上位の寄稿者、1日あたりのコミット数、1日あたりのデプロイ数などの詳細な GitLab 統計情報を追跡できます。プロジェクトなどのテンプレート変数を使用し

て、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 からのデータと他のソースからのデータを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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

### 既知の制限事項

変換はアラートクエリではサポートされておらず、GitLab API raw 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あるため、このプラグインではアラートは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 コンソールを開き、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. [データソースを追加する] を選択する。

#### 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ロールがないことを意味します。

4. データソースのGitLabリストから選択します。
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 には、など、GitLab インスタンスのルート URL を入力します **https://gitlab.com/api/v4**。
  - アクセストークン には、個人用アクセストークンを入力します GitLab。

## GitLab データソースをクエリする

GitLab クエリエディタから、コミット、問題、リリースなど、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と表示

1. ドロップダウンメニューから、プロジェクト を選択します。
2. (オプション) 所有してい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、はい 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 には長い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と表示

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/タグリファレンスを追加します。

### 問題をフィルタリングして表示する

1. ドロップダウンメニューから、問題 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。status 属性は、created、running、successfailed または の値のいずれかになります canceled。



## 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。

## テンプレートと変数

新しい GitLab クエリ変数を追加するには、「」を参照してください[クエリ変数の追加](#)。GitLab データソースをデータソースとして使用します。リソースタイプを選択します: リリース、プロジェクト、またはラベル。

プロジェクト、ラベルなどの動的リストを取得して選択するには、クエリタイプ変数を作成します。クエリタイプ変数は、GitLab クエリエディタを使用してプロジェクト、ラベルなどのクエリと返します。次の例では、クエリをパラメータ化するためのプロジェクト変数を作成します。

### プロジェクト変数を作成してクエリをパラメータ化する

1. という名前の Query 型の変数を追加します **project**。
2. GitLab データソースを選択し、ダッシュボードロード で を更新します。
3. プロジェクトリソースタイプ、「はい」、「所有済み」、「表示フィールドの名前」、「値フィールドの ID」を選択します。
4. 更新 を選択して、変数をダッシュボードに追加します。
5. ダッシュボードに新しいパネルを追加し、 をプロジェクト ID **\$project**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が得られます。

## Grafana からの変換を使用して一般的な質問に答える

基本的な GitLab クエリを実行してコミット、問題などを検索できたので、変換を使用してデータセットを視覚化、集約、グループ化、結合したり、他の多く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/デプロイの数は 1 日あたりいくつですか？

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、プロジェクト ID を追加します。
2. 変換による新しいグループを追加する: によるグループで、created\_at\_date を選択し、(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を閉じるまでの平均時間はどれくらいですか？

1. クエリを追加します。リソースタイプの問題を選択し、プロジェクト ID を追加します。
2. 計算変換から新しい追加フィールドを追加します。モードではバイナリオペレーションを選択し、オペレーションでは `closed_at = created_at` を選択し、エイリアスでは `resolution_time` を選択します。
3. 計算変換から新しい追加フィールドを追加します。モードではバイナリオペレーションを選択し、オペレーションでは `resolution_time / 86400000` を選択し、エイリアスでは `resolution_time` を選択します。

すべてのフィールドを置き換える で、True を選択します。

### 4. 統計の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 データソースに接続する

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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

### 既知の制限事項

- このデータソース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した列の最初の 1000 個の一意的値のみを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 日間のデータのみをクエリできます。

## データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開き、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. [データソースを追加する] を選択する。
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ない場合は、現在のユーザーに Admin ロールがないことを意味します。

## Honeycomb の設定

名前	説明
Name	データソース名。これは、パネル、クエリ、および探索でデータソースを表示する方法です。
Honeycomb API key	Honeycomb から取得した API キー。
URL	Honeycomb API の URL。例えば <code>https://api.honeycomb.io</code> です。
Team	API キーに関連付けられた Honeycomb チーム。

## Honeycomb データソースをクエリする

メトリクスをクエリするには、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COUNTに対する です。
- クエリを絞り込むには、視覚化、視覚化、ここで、制約、によるグループ化、による順序、制限 など、残りのフィールドの値を選択します。

## テンプレートと変数

新しい Honeycomb クエリ変数を追加するには、「」を参照してください[クエリ変数の追加](#)。

YOu は、データセット、列、または列値を含む変数を作成できます。

- データセットが選択されていない場合、変数には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、変数には列名が含まれます。
- データセットと列の両方が選択されている場合、変数には列の値が含まれます。列の値は、エディタの「場所」フィールドを使用してさらに制約できます。

## Honeycomb UI でクエリを表示する

ダッシュボードパネルから Honeycomb UI で作成したクエリを表示するには、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、「」を参照してください[ダッシュボードのインポート](#)。

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された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、Dashboards タブを選択します。

## Jira データソースに接続する

Jira の問題データと他のソースからのアプリケーションパフォーマンスデータを組み合わせて、開発プロセス全体を把握します。

**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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

- 問題の作成または解決に基づいて注釈を作成し、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、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 プロジェクトにアクセスできるアトラシアン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てい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 コンソールを開き、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. [データソースを追加する] を選択する。

**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 には、 などのアトラシアンインスタンスのルート URL を入力します **https://bletchleypark.atlassian.net**。
  - ユーザー には、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 には、ユーザー用に生成された API トークンを入力します。

## Jira データソースをクエリする

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 に問題をクエリします。これは 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があります。

### 問題をフィルタリングして表示する

1. フィールド ドロップダウンを選択し、先行入力を使用して、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。試すいくつかのフィールド：
  - 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見積もり — チームが問題について推定したストーリーポイントの数
2. 問題をフィルタリングまたはソートします。これを行うには、有効な JQL 式を入力して、プロジェクト、担当者、またはアトラシアンクエリ言語 JQL を使用したスプリントなどのフィールドに基づいて問題をフィルタリングまたはソートします。

ここから、データをテーブルに表示するか、Grafana 変換を使用して、その発行データを操作したり、計算を実行したり、データを時系列グラフに変換したりできます。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には、日付フィールドと数値フィールドを選択し、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もり。

前述の例自体はあまり役に立ちません。数値フィールドは、変換から計算できます (おそらく計算されるでしょう)。Group By Transformation 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もりを要約して、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。

### テンプレートと変数

新しい Jira クエリ変数を追加するには、「」を参照してください [クエリ変数の追加](#)。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、JQL 式で参照できます。例えば、プロジェクトステータスダッシュボードを作成してプロジェクトを選択するか、エピック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するか、タスク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することができます。

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、または問題に関連するものを返します。以下に例を示します。

担当者変数を作成して、担当者別の問題のステータスを取得します。

1. という名前の Query 型の変数を追加します **assignee**。
2. フィールド : 担当者 を選択します。
3. ) オプション) JQL フィルタープロジェクト = '自分のプロジェクト' 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、変数をダッシュボードに追加します。
6. 新しいパネルをダッシュボードに追加し、JQL を編集して、新しい変数担当者 = \$assignee を使用してフィルタリングします。

ここで、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ません。

複数值変数を使用すると、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、担当者 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枠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。マクロは 2 つあります。

- \$\_\_timeFrom
- \$\_\_timeTo 。

次の JQL クエリの例では、ダッシュボードの時間枠内に作成された問題をフィルタリングします。

```
createdDate >= $__timeFrom AND createdDate <= $__timeTo
```

## データソースを最大限に活用する

Grafana の変換やその他の組み込み機能を使用すると、Jira データを表示するのに役立ちます。

### 変換を使用した JQL の拡張

Grafana には多くの変換から選択できますが、以下は、JQL に SQL の機能や能力の一部を与える強力な拡張を提供します。

この変換によるグループ化は、標準の Jira JQL 構文の一部ではない主要な機能であるグループ化を提供します。Group By 変換を使用すると、スプリントやその他の問題フィールドでグループ化し、グループ別に集計して、スプリントで実際に完了した速度やストーリーポイントの見積もりと実際の完了率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、共通のフィールドで 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結合し、他の変換を使用して複数のクエリ/データセットの値を計算することができます。

計算からフィールドを追加 SQL 式と同様に、この変換では、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。計算で使用されるフィールドは、単一のクエリからでも、結合したクエリからでもかまいません。また、計算を連結し、計算フィールドから計算を実行することもできます。

### Grafana からの変換を使用して一般的な質問に答える

変換を使用すると、データセットを視覚化、集計、グループ化、結合したり、他の多く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. フィールド: スプリント名、ストーリーポイントの見積り を選択します。
2. JQL フィルターを追加します。 `project = "Your Project" AND type != epic AND status = done order by created ASC`
3. Group By 変換を追加します。
  - スプリント名 | グループ化の基準
  - ストーリーポイントの見積もり | 計算 | 合計
4. バーゲージの視覚化を選択します。

### スプリントで完了と推定された内容を表示する方法



1. クエリを追加します。まず、フィールド: 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 を選択します。

次に、JQL フィルターを追加します。 `project = 'Your Project' AND type != epic`

2. 2 番目のクエリを追加します。まず、フィールド: 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 を選択します。

次に、JQL フィルターを追加します。 `project = 'Your Project' AND type != epic AND status = done`

3. Group By 変換を追加します。
  - スプリント名 | グループ化の基準
  - スプリント開始日 | グループ化の基準
  - ストーリーポイントの見積もり | 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を完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。まず、フィールド: 作成済み、ステータスカテゴリ変更済み を選択します。

次に、JQL フィルターを追加します。 `project = 'Your Project' AND type != epic AND status = done`

2. 変換を追加する: 計算からフィールドを追加する
  - モード = 行を減らす
  - 計算 = 差分
3. 変換を追加する: 計算からフィールドを追加する
  - モード = バイナリオペレーション
  - オペレーション = 差分 / 86000000
  - エイリアス = 日数
4. 変換を追加する: フィールドを整理する
  - 異なるフィールドを非表示にする

5. 変換の追加: 値でデータをフィルタリングする

- フィルタータイプ = 含める
  - 条件 = 任意の 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. 統計の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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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 シェルと同じ構文をサポートしますが、いくつかの制限があります。\* 実行できるコマンド/クエリは 1 つだけです。\* 読み取りコマンドのみがサポートされています。検索と集計 \* ほとんどのオブジェクトコンストラクタはサポートされていません (サポートされている ISODate を除く)

エディタは、次の方法で MongoDB シェル構文を拡張します。

- 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」は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 の無料階層ではソートが許可されません。無料利用枠を使用するユーザー向けに構文を拡張しました。

**Note**

MongoDB はこの構文でソートを実行しません。ソートは、結果がコレクションからクエリされた後に行われます。

```
sample_mflix.movies.aggregate({}).sort({"time": 1})
```

- 空のエディタを使用すると、Ctrl + Space に使用可能なすべてのデータベースの選択が表示されます。
- 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、ソート/制限という追加の関数が表示されます。

## クエリの実行

Cmd + S を押してクエリを実行します。

## 時系列

時系列データを視覚化する場合、プラグインはどのフィールドを時間として使用するかを知る必要があります。フィールドを「time」という名前のエイリアスで射影するだけです。フィールドデータ型は日付である必要があります。

日付以外のデータ型を現在まで強制できます。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。次の例は、MongoDB \$dateFromParts pipeline 演算子を使用して、int フィールド「year」を「time」として射影される日付に変換する方法を示しています。

```
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](#)

現在、次の診断コマンドがサポートされています。「stats」、「serverStatus」、「replSetGet「Status」、「getLog」、「connPoolStats「」、「connectionStatus」、「buildInfo」、「dbStats」、「hostInfo」、「lockInfo」

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

クエリでダッシュボードの時間範囲を参照できます。

- \$\_\_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 は「複合変数」という概念をサポートしています。これにより、1 つの変数を複数の変数として使用して、複雑なマルチキーフィルターを実行できます。

複合変数を作成するには、アンダースコアを使用して変数を分割する命名規則を使用します (アンダースコアで始まる必要があります): `_var1_var2` クエリを実行する場合、レスポンスは次の形式である必要があります。 `val1-val2`

例: 映画名と年の両方で結果をフィルタリングしたい。

1. Query 型の変数を作成します。 `_movie_year`
2. 次の例に示すように、変数クエリを 1 つの `movie-year` プロパティを持つ項目の配列を返すクエリに設定します。

```
// 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. クエリでは、「`$_variable`」という構文を使用して、「映画」と「年」を個別のテンプレート変数として参照できます。

## アドホックフィルターの使用

任意の名前の標準の「アドホックフィルター」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。「`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ある必要があります。クエリ結果は、選択可能なフィルターの入力に使用されます。この変数はそれ以上目的を果たさないため、表示から非表示にすることもできます。

```
sample_mflix.movies.aggregate([\n  {"$group": { "_id": "$year"}},\n  {"$project": { "year": "$_id", "_id":\n    0 }} ] )
```

## New Relic データソースに接続する

このセクションでは、New Relic [APM](#) と Grafana の [インサイト](#) 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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

## 構成

データソースを追加し、[管理者 API キー](#)、[個人用 API キー](#)、[アカウント ID](#) のフィールドに入力します。

## 使用方法

### サービスタイプ

- メトリクス: New Relic の [REST API を介して New Relic APM をクエリするための](#)。
- Insights : [NRQL](#) 経由で New Relic Insights をクエリする場合。

### エイリアス

プレーンテキストと次の変数を組み合わせて、カスタム出力を生成できます。

可変	説明	値の例
<code>\$_nr_metric</code>	メトリクス名	CPU/ユーザー時間
<code>\$_nr_metric_value</code>	メトリクス値	平均値

例:

```
<para>
  Server: $_nr_server Metric: $_nr_metric
</para>
<programlisting>
```

### テンプレートと変数

1. ダッシュボードのテンプレート変数を作成します。詳細については、「[テンプレートと変数](#)」を参照してください。
2. 「Query」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策定します。

利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) クエリを作成する際の書き込みエクスペリエンスを向上させるために、エディタは事前定義された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 `SINCE <from> UNTIL <to>` に基づいてに補間します。

例：

```
<para>
  SELECT average(value) FROM $event_template_variable
  $_timeFilter TIMESERIES
</para>
<programlisting>
```

マクロとテンプレート変数の使用方法に関するその他のヒントについては、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 データソースを選択し、追加のフィルターを設定します。フィルターが設定されていない場合、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ため、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、アプリケーションのリスト `$_app_` を取得し、エンティティ ID フィルターとして使用する変数を作成します。



## デプロイイベント

アプリケーション ID は必須フィールドです。

## Oracle Database データソースに接続する

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## データソースの追加

Grafana の左側のパネルにあるデータソースを選択します。

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とパスワードを入力します。

tnsnames オプショントグルを使用すると、tnsnames.ora 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 Kerberos を使用しています。kerberos を使用するように OS または docker コンテナを設定する方法の詳細については、kerberos 固有のセットアップガイドを参照してください。

オプションで、Oracle サーバーへの接続に使用するタイムゾーンを変更し、タイムゾーン対応マクロで使用します。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」という緑色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可能にするために、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
<code>*\$__time(dateColumn)*</code>   列の名前を「time」に変更する式に置き換えられます。例えば、`dateColumn as time` <code>*\$__timeEpoch(dateColumn)*</code>	列の名前を <code>time</code> 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。
<code>*\$__timeFilter (dateColumn)*</code>   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` <code>*\$__timeFrom ()*</code>	データ DATE 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 <code>TO_DATE('19700101','yyyymmdd') + (1/24/60/60/1000) * 1500376552001</code> です。
<code>*\$__timeTo ()*</code>   現在アクティブな時間選択が「DATE」データ型に変換された末尾に置き換えられます。 <code>*\$__timeGroup (dateColumn, "5m")*</code>	GROUP BY 句で使用できる式に置き換えられません。
<code>*\$__timeGroup (dateColumn, "5m"[, fillvalue ])*</code>	GROUP BY 句で使用できる式に置き換えられます。 <code>fillValue</code> を NULL または浮動値に指定すると、その値で時間範囲内の空のシリーズが自動的に埋められます。例えば、 <code>timeGroup createdAtt, '1m', 0.*\$__timeGroup (dateColumn, "5m", 0)*</code> です。
<code>*timeGroup (dateColumn, '5m', NULL)*</code>   Same as above but NULL will	上記と同じですが、まだ NULL が表示されていない場合は、そのシリーズの以前の値がフィル値として使用されます。

マクロの例	説明
<pre>lbeusdasvalueformissigoints .*__timeGroup (dateColumn ,"5m", previous)*</pre>	
<pre>*\$__unixEpochFilter(dateColumn)*   指定された列名を使用して、時刻を Unix タイムスタンプ (ミリ秒単位) で表した時間範囲フィルターに置き換えられます。例えば、`dateColumn &gt;= 1500376552001 AND dateColumn &lt;= 1500376552002` *\$__unixEpochFrom()*</pre>	<p>現在アクティブな時間選択の開始を Unix タイムスタンプに置き換えます。例えば 1500376552001 です。</p>
<pre>*\$__unixEpochTo()*</pre>	<p>現在アクティブな時間選択の最後に Unix タイムスタンプに置き換えられます。例えば 1500376552002 です。</p>

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は、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を実行し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、実行された未加工の補間された SQL 文字列が表示されます。

## テーブルクエリ

クエリとしてフォーマット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。テーブルパネル列の名前は、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` 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秒単位で返す `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のない `DATE` 列と `TIMESTAMP` 列を UTC として解釈します。`time` とを除くすべての列 `metric` 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 – の使用 `oracle-fake-data-gen`

```
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、フェイク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

メトリクスクエリでサーバー、アプリケーション、センサー名などのモノをハードコーディングする代わりに、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 のテンプレート変数を追加するとQuery、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のものを返す Oracle クエリを作成できます。

例えば、テンプレート変数 Query 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使用できます。

```
SELECT "hostname" FROM host
```

クエリは複数の列を返すことができ、Grafana は自動的にそれらからリストを作成します。例えば、次のクエリは、hostname および の値を含むリストを返しますhostname2。

```
SELECT "host.hostname", "other_host.hostname2" FROM host JOIN other_host ON host.city = other_host.city
```

クエリ\$\_\_timeFilter("time\_column")でなどの時間範囲に依存する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時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 および という名前の 2 つの列を返す必要があります\_\_value。\_\_text 列の値は一意である必要があります (一意でない場合、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、をテキストhostname、を値idとするクエリを示しています。

```
SELECT "hostname" AS __text, "id" AS __value FROM host
```

ネストされた変数を作成することもできます。例えば、 という名前の別の変数がある場合ですregion。次に、hosts 変数に、このようなクエリを使用して、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 ( が複数値変数である場合regionは、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 の場合にのみ引用符で囲まれますmulti-value。

変数が複数値変数の場合は、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。

次の 2 つの構文があります。

`${varname}` という名前のテンプレート変数の例hostname :

```
SELECT
  "atimestamp" as time,
  "aint" as value
FROM table
WHERE $__timeFilter("atimestamp") AND "hostname" IN('${hostname}')
ORDER BY "atimestamp" ASC
```

`[[varname]]` という名前のテンプレート変数の例hostname :

```
SELECT
  "atimestamp" as time,
  "aint" as value
FROM table
WHERE $__timeFilter("atimestamp") AND "hostname" IN('[[hostname]]')
ORDER BY atimestamp ASC
```

## Salesforce データソースに接続する

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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 から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 接続アプリケーション](#) が必要です。

### 既知の制限事項

- アドホックフィルターはまだサポートされていません。

- 現在、SOQL クエリと 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 クエリ形式はまだサポートされていません。

## 必須の設定

以下の設定が必要です。

### Note

プラグインは現在、OAuth 2.0 ユーザー名/パスワードフローを使用しています。接続されたアプリに必要なコールバック URL は使用されません。したがって、任意の有効な URL に設定できます。

名前	説明
Enable OAuth settings	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。
Callback URL	このプラグインでは使用されないため、有効な URL を指定できます。
Select OAuth Scopes (minimums)	データ (api) にアクセスして管理します。
Require Secret for Refresh Token Flow	これを有効または無効にすることができます。



## データソースの追加

1. Amazon Managed Grafana ワークスペースで Grafana コンソールを開き、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. [データソースを追加する] を選択する。

### 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 には、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 には、その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 には、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 には、Salesforce に接続するためのコンシューマーキーを入力します。これは、Salesforce 接続アプリケーションから取得できます。
  - コンシューマーシークレット には、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 接続アプリケーションから取得できます。
  - サンドボックスを使用する で、Salesforce サンドボックスを使用する場合は、これを選択します。

## Salesforce データソースをクエリする

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)

Query Builder は、SOQL クエリを構築するための使いやすいインターフェイスです。SOQL クエリの作成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 のエンティティを参照します。クエリビルダーの他のオペレーションの前に、FROM フィールドを選択する必要があります。

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 Builder 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/セールスフォース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、価値、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、「先月に作成されたオポチュニティをカウントする」などの結果を取得します。または「ステージ名でグループ化された機会の合計値は何ですか？」
- **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、「機会の数を でカウントする」などの結果を取得します CreatedDate。または「オポチュニティの終了日別にグループ化された合計値」。

クエリエディタで Entity/FROMと モードを選択したら、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	に適用可能	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は、値を集計する方法も選択します。
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、集計	(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、集計	(オプション) 返される結果の数を制限します。デフォルトは 100 です。
GROUP BY	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は、フィールドを選択します。

フィールド	に適用可能	説明
時間フィールド	トレンド	結果をグループ化する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、クエリエディタの下に生成された SOQL のプレビューも表示されます。クエリビルダーの制限でブロックされている場合は、SOQL エディタに安全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

raw SOQL エディタには、raw 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、オートコンプリート候補を提供します。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+Space 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 + SPACE を使用して、使用可能なコンテキストオプションを示すコード補完を表示します。

CMD + S はクエリを実行します。

## 時系列としてのクエリ

日付フィールドをエイリアスし、メトリクスフィールドをメトリクスにエイリアスして時系列クエリを行い、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でフィルタリングするには、SOQL クエリでマクロを使用できます。

- `$_timeFrom`— データtime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— データtime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せん。
- \$\_\_quarterStart— 会計四半期の開始 (SalesForce 事業年設定から派生)。
- \$\_\_quarterEnd— 会計四半期の終了 (SalesForce 事業年設定から派生)。

```
SELECT UserId, LoginTime from LoginHistory where LoginTime > $__timeFrom
```

## テンプレートと変数

新しい Salesforce クエリ変数を追加するには、「」を参照してください[クエリ変数の追加](#)。Salesforce データソースをデータソースとして使用します。任意の SOQL クエリをここで使用できます。

ユーザー ID やユーザー名など、名前と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。最初のフィールドが ID として使用されます。これは、クエリエディタ SOQL でキー (ID など) でフィルタリングする場合に行います。

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 データソースに接続する

[SAP HANA](#) は、データ駆動型のリアルタイムの意思決定と \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 によって開発および販売されています。SAP HANA データソースプラグインは、SAP HANA インスタンスを Grafana に接続するのに役立ちます。

SAP HANA Grafana Enterprise プラグインを使用すると、SAP HANA データを Grafana の他のすべてのデータソースとともに視覚化し、コンテキスト内のログ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、組み込みのクエリエディタが含まれており、注釈をサポートしており、アラートのしきい値の設定、アクセスの制御、アクセス許可の設定などを行うことができます。

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## 機能

- クエリエディター — 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 マクロを自動で完了できる構文強調表示付きの SQL クエリエディター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 — Grafana グラフに SAP HANA イベントまたはデータをオーバーレイして、イベントを他のグラフデータと関連付けます。
- 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 — SAP HANA データに基づく Grafana で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 コンソールを開き、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. [データソースを追加する] を選択する。

### 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、次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: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxx.hana.trial-us10.hanacloud.ondemand.com。
  - 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かどうかを 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 Client Auth を有効にします。

- (オプション) 自己署名 TLS 証明書の検証を有効にする場合は、CA 証明書で 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。これを省略する場合は、すべてのクエリでスキーマを指定する必要があります。

## アクセスとアクセス許可

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、制限付きユーザーを作成する例です。このクエリは、パスワードの強制変更も無効にします。

```
CREATE RESTRICTED USER <USER> PASSWORD <PASSWORD> NO FORCE_FIRST_PASSWORD_CHANGE;
```

次に、ユーザーが Grafana などのクライアントを介してシステムを次のように接続できるようにします。

```
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権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

SAP HANA 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クリックして、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、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 クエリエディタが付属しています。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。クエリエディタは、サポートされている Grafana マクロの自動入力と SQL クエリの構文強調表示を提供します。

## 注釈

SAP HANA クエリを 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、「」を参照してください [注釈](#)。

SAP HANA から注釈を作成するには

1. ダッシュボード設定の歯車アイコンを選択します。
2. 左側のメニューから、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. ドロップダウンメニューで、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。

## テンプレートと変数

新しい SAP HANA クエリ変数を追加するには、「」を参照してください [クエリ変数の追加](#)。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 username から 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、最初の列が表示値として使用され、2 番目の列が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、クエリは拒否されます。

## テンプレートと変数

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。次の例では、`${age}`は数値を受け入れ、テーブルの数値フィールドと比較するためのテキストボックス変数です。

```
select * from "users" where "age" > ${age}
--- will be translated into
select * from "users" where "age" > '36'
```

変数が複数の値を返す場合は、次のように SAP HANA クエリinの条件で使用できます。SAP HANA でwhere in条件を有効にするには、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

- `$__timeFilter(<time_column>)`— raw クエリで使用すると、Grafana の時間範囲を指定された列に適用します。date/timestamp/long time 列に適用されます。
- `$__timeFilter(<time_column>, <format>)`— 上記と同じです。ただし、では、データベースに保存されている `time_column` の形式を指定できます。
- `$__timeFilter(<time_column>, "epoch", <format>)`— 上記と同じですが、時間列がエポックの場合に使用できます。形式は「s」、「ms」、「ns」のいずれかになります。
- `$__fromTimeFilter(<time_column>)`— 上記と同じですが、時間列がエポックの場合に使用できます。形式は「s」、「ms」、「ns」のいずれかになります。
- `$__fromTimeFilter(<time_column>, <comparison_predicate>)`— 上記と同じですが、`comparison_predicate` を指定できます。
- `$__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 の時間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。date/timestamp/long time 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。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
```

マクロでは、形式はタイムスタンプ列に一致する有効な HANA 形式の 1 つにすることができます。例えば、YYYYMMDDHH24MISS は、データが形式で保存されている場合に有効な 20210421162012 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 DB 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、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つ `$_timeFilter (<time_column>, "epoch" <format>)` マクロについて説明します。マクロでは、形式は ms、s、ns のいずれかになります。指定しない場合、は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 タイムピッカーの時刻に基づいて、時刻フィールドで条件に拡張されます。

これにより、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名です。comparison\_predicate または time 列の形式を 2 番目の引数として渡すことができます。両方を

渡す場合、形式は 2 番目のパラメータであり、3 番目のパラメータとして `comparison_predicate` を使用します。

<format> 形式が指定されていない場合、プラグインは time 列が timestamp/date 型であると仮定します。time 列が timestamp/date 以外の形式で保存されている場合は、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 などのその他のカスタム形式のいずれかになります。

<comparison\_predicate> オプションのパラメータ。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` を最初から使用する代わりに、`from` から `to` 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(timestamp,1h)` は、クエリで `SERIES_ROUND("timestamp", 'INTERVAL 1 HOUR')` に展開されます。

次の例では、`$__timeGroup` について説明します。 `$__timeGroup(<time_column>, <interval>)` macro.

```
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')` に展開されます。

```
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**

によるグループ化を `$_timeGroup` マクロで使用する場合は、選択したフィールドでソートするフィールドの名前がグループでソートするフィールドと同じであることを確認します。そうしないと、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` を指定することで、その間隔を 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
```

そのクエリは、ダッシュボードの時間範囲に基づいて後続の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には

1. ダッシュボード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、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作成します。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。クエリの形式を以下に示します。

```
SELECT TO_TIMESTAMP('${__to:date}'), <METRIC> FROM <TABLE#> WHERE <YOUR CONDITIONS>
```

次の例では、テーブルにはユーザー名、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の数が3人未満になったときに通知します。

```
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'
```

## データソースに接続する ServiceNow

これは、インスタンスへの接続 ServiceNowに使用される 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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、およびその他のテーブル
- アラート
- 注釈 (ベータ機能)
- テンプレート変数

## 構成

Grafana の左側のパネルでデータソースを選択します。

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 「Connection 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はプラグインに含まれており、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には、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、の時点でv1.4.0、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。このオプションを使用すると、データソースの設定に使用されたユーザーが使用可能な任意のテーブルにあるServiceNow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する必要があります。

## TableAPI クエリ

TableAPIは、テーブルパネルに表示するのに適したデータを返します。これにより、表示するフィールドの順序付きの選択とフィルタリングオプションが可能になります。クエリエディタには、クエリによって返される行数を制限する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の例。

### 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。複数のフィールドを指定することもできます。フィールドは、指定された順序で返されます。

### 値の表示

Display Values フラグを使用すると、クエリは数値ではなく人間にわかりやすい値を返したり、values を表示したりします。

例えば、このフラグがないの重要度は、のみを表示します。フラグが有効になっている場合、表示される値はになります - High。

[ServiceNow API ドキュメント](#)によると、これはパフォーマンスに悪影響を及ぼす可能性があります。

#### Note

[...] 表示値を指定すると、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 (一般)

フィルター行を使用すると、複数のフィールドと値の条件に基づいて、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 オペレーションと結合されます。

カスタムテーブルを使用しない場合、次のフィールドを使用できます (このリストは今後展開される予定です)。

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 からフィールドが自動的に入力されます。

### 日付フィルター

時間 フィー ルド	演算子	値
で開 かれ まし た	今日以前、または今日以前、または以降	timestamp javascript:gs.daysAgo (30)
アク ティ ビ ティ 期限		



時間 フィー ルド	演算子	値
でク ロー ズ		
期日		
予想 され る開 始		
再開 時刻		
解決 日時		
作業 終了		
作業 開始		
時間 を無 視す る		

その他の日付値については、[https://developer.servicenow.com/app.do#!/api\\_doc?v=newyork&id=r\\_SGSYS-dateGenerate\\_S\\_S](https://developer.servicenow.com/app.do#!/api_doc?v=newyork&id=r_SGSYS-dateGenerate_S_S) を参照してください。

演算子 (一般、文字列ベース )

- 先頭が一致

- Ends With
- いいね
- 好きではない
- 等しい
- 等しくない
- Is Empty

### 演算子 (時間ベース)

- 本日
- 今日ではない
- 前
- 時または以前
- 後
- 以降

### [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 True/False 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のセットがあります

### ソート基準

ソート基準行では、複数のフィールドと値の条件に基づいて、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 オペレーションと結合されます。追加の演算子のサポートが追加されません。

### 制限

行制限を指定して、返されるデータが多すぎないようにできます。デフォルト値は 25 です。

## 時間フィールド

Time Field は、クエリされたデータを時系列に変換します。時系列として処理されるデータは、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含まれないことを意味します。

使用されるデフォルトの時間フィールドは「Opened At」ですが、時間値を保持する任意の使用可能なフィールドに変更できます。

特別な値「時間を無視」が提供され、「現在まで」の結果が可能になり、フィルターが表示するデータを制御できるようになります。

## AggregateAPI クエリ (統計)

AggregateAPI は常にメトリクスを返し、集計は avg、min、max、sum になります。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。複数のメトリクスを指定することもできます。

## フィルター (一般)

Aggregate Filters は、テーブルオプションと同様に、フィールドと値の条件に基づいて表示されるメトリクスを絞り込む機能を提供します。

すべてのフィルターは AND オペレーションと結合されます。追加の演算子のサポートが追加されません。

統計フィルターオプションは TableAPI と同じです。

## 集計

メトリクス集計には 4 つのタイプがあり、さらに「カウント」があります。

- [Average] (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の「数値」を返します。

## グループ化の基準

このセレクトは、メトリクスをより小さな集計に分割する機能を提供します。「priority」でグループ化すると、「tag」が優先され、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

クエリで名前をハードコーディングする代わりに、変数を代わりに使用できます。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。

クエリ変数を追加して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

タイプのテンプレート変数を追加するとQuery、ドロップダウン選択ボックスとして表示されるカテゴリ名、キー名、キー値などの項目を返すクエリを作成できます。

例えば、テンプレート変数 Query 設定でこのようなクエリを指定categoriesすることで、のすべての値を含む変数を作成できます。

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、タイプはインシデントと変更に限られています。タイプを選択すると、そのタイプに該当す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、Templating と一緒に使用してダッシュボードパネルでデータをフィルタリングできます。

例えば、カテゴリ という名前の変数を追加し、タイプ = インシデント、フィールド = カテゴリ を選択すると、カテゴリ のオプションの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、カテゴリ等号 `#{category}` を選択すると、パネルデータには、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

次の2つの構文があります。

`$(varname)` という名前のテンプレート変数の例hostname :

`[[varname]]` という名前のテンプレート変数の例hostname :

## アラート

標準の 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、アラートの生成に使用できます。

クエリとアラートの例を次に示します。このクエリは、すべての未解決の重大で優先度の高いインシデントのグラフを返します。

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 つ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、アラートが「OK」から「保留中」から「アラート中」に移行したことを示します。

グラフビューには縦線が表示され、アラートが保留されている間は上部のハートアイコンがオレンジ色に変わります。

アラートの基準を満たすと、ルールは赤に変わります。

グラフビューに赤い縦線が表示され、上部のハートアイコンが赤に変わります。

## アラートのインシデントの記述

### ベータ機能

- ServiceNow データソース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がインシデントを書き込むためのアクセス許可を持っている必要があります。

### HTTP プロキシを使用する

HTTP プロキシを使用する場合、Amazon Managed Grafana には、プロキシの場所に設定されている次の環境変数 (複数可) が必要です。

- HTTP\_PROXY+ または http\_proxy-

- フルパス - `http://host:port`
- または単に: `host:port`
- HTTPS\_PROXY (または `https_proxy`):
  - フルパス - `https://host:port`
  - または単に: `host:port`

## 注釈

Grafana Annotations は、このデータソースv1.4.0のベータ機能です。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。

Annotations クエリは、標準の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、いくつかの小さな違いがあります。

- 選択可能な列は 1 つだけです。これは今後の改善で修正される可能性が高いです。
- 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なりますか？

次のアクションを実行するには管理者アクセスが必要です

オプション 1: Grafana ユーザーにすべてのテーブルへのアクセスを許可するアクセス許可を付与します。

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 ACLs を適用します。

以下のアクションを実行するには、管理者アクセスが必要です。

1. ログインした管理者は、`security_admin` へのアクセスを昇格する必要があります。
  - a. 右上のナビゲーションペインで、プロフィールアイコンを選択します。プロフィー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. 表示されるモーダルから、`security_admin` チェックボックスをオンにします。
  - d. [OK] をクリックします。

2. 任意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. 上部の「新規」を選択します。
  - c. ロールの名前と関連する説明を入力します。
  - d. [送信] を選択します。
3. 新しいユーザーを作成するか、必要なロールを持つ既存のユーザーを変更します。
  - 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が許可されます)
4. 必要なテーブルとフィールドACLs を作成します。
  - sys\_db\_object テーブルの ACL を作成します。
    - i. 2 番目の検索ヘッダー列の名前で、 と入力し **sys\_db\_object**、Enter を押します。
    - ii. フィルタリングされた結果には、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、アクセスコントロールが選択したタブであることを確認します。
    - v. 新規 を選択して、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. 画面の下部にある「ロールが必要」セクションで、「新しい行を挿入」(ダブルクリック) を選択し、作成したロールを検索します。
    - viii. 作成した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. 画面の下部にある「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 フィールドの ACLs 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に対して、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を開いたまま、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. 下部のタブセクションで、アクセスコントロールタブで新規を選択します。
  - d. オペレーションを読み取りに変更する
  - e. 下部の「ルールが必要」テーブルに行テキストを挿入する (ダブルクリック) を選択します。
  - f. 作成したルールを検索し、緑色のチェックマークを選択します。
  - g. [送信] を選択します。
  - h. 名前、ラベル、表示名、拡張テーブルのすべての必須フィールドに対して、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 からクエリを実行するその他の非 CMDB テーブルについて、4.1 の手順を繰り返します。4.2 の手順を繰り返さないでください。その手順は `sys_db_object` にのみ必要です。

## Snowflake データソースに接続する

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ともに可視化し、コンテキスト内のログデータとメトリクスデータを可視化できます。このデータソースには強力な先行型クエリエディタが含まれており、複雑な注釈のサポート、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可能です。

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## 概要

### Snowflake とは

Snowflake は、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。一般的に「データ」と呼ばれ、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 warehouse-as-a-service 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



データを手頃な価格で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 に達しています。現在、多くの企業は、Snowflake をプライマリデータベースとして使用して、トランザクション数、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 と Amazon Managed Grafana を最大限に活用する

**Snowflake データを動かさずに視覚化する:** Grafana の独自のアーキテクチャは、データを動かして冗長ストレージと取り込みに支払うのではなく、データを直接クエリします。

**さまざまなソースからパネルを構成する:** 構築済みの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 つのペイン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することができるため、データの準備が少なく済みます。

**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 キーワードをオートコンプリートできるスマート SQL 自動コンプリートエディタです。

**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する

**注釈:** Grafana グラフに Snowflake イベントをオーバーレイして、イベントを他のグラフデータと関連付ける

**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する

**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インタラクティブなダッシュボードを作成します。

**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独自の列に記述する

**Snowflake プラグインの使用を開始する**

Grafana で Snowflake プラグインの使用を開始する 5 つの簡単な手順は次のとおりです。

## ステップ 1: Snowflake データソースを設定する

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、データソースをクエリする準備が整います。

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。
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)	クエリのロールを引き受けます。

## ステップ 2: Snowflake データのクエリを作成する

ダッシュボードにパネルを作成し、Snowflake データソースを選択してクエリエ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、クエリのどこにでも表示できます。
- 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は、オプションで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
```

## 間隔別の時系列グループの 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>
```

## テーブルの 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: テンプレート変数を作成して使用する

### テンプレート変数の使用

次の例に示すように、クエリに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}'
```

### Snowflake データソースを使用した変数の作成

ダッシュボード設定で、**変数** を選択し、**新規** を選択します。

「クエリ」変数タイプを使用して、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、必ず列を 1 つだけ選択してください。

例：

```
SELECT DISTINCT query_type from account_usage.query_history;
```

は、次の変数を提供します。

```
All DESCRIBE USE UNKNOWN GRANT SELECT CREATE DROP SHOW
```

#### ステップ 4: アラートを設定する

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。

クエリエディタでアラートタブボタンを選択し、アラートの作成 を選択します。

#### Step 5. 注釈を作成する

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、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には、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 への失敗したすべてのログイン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、Snowflake への失敗したすべてのログインの注釈がダッシュボードパネルに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の「名前の表示」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選択して、Snowflake へのアクセスを制限し、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、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 の請求および使用状況データを理解する

Snowflake データソース内で、請求情報と使用状況に関する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 ACCOUNTADMIN ロールが必要です。これを安全に行うには、ACCOUNTADMIN ロールを持つユーザーを持つ新しい Grafana データソースを作成します。次に、変数でそのデータソースを選択します。

### Splunk データソースに接続する

#### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」を参照してください。

## 構成

### 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 を使用し https、設定した 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 ではなく 8089 です (これは

デフォルトのウェブ UI ポートです )。Basic Auth を有効にし、Splunk のユーザー名とパスワードを指定します。

## ブラウザ (ダイレクト) アクセスモードと CORS

Amazon Managed Grafana は、Splunk 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。

## 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。これは実験的な機能です。実際に必要になるまで有効にしないでください。

### ポーリング結果

検索を実行してから、定期的に結果を確認します。このオプションは内部で search/jobs API コールを実行し、を exec\_mode に設定します normal。この場合、API リクエストはジョブ SID を返し、Grafana はジョブの結果を取得するためにジョブの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、このオプションは無効になっており、Grafana oneshot は を に設定 exec\_mode して、同じ API コールで検索結果を返すことを許可します。search/jobs API エンドポイントの詳細については、[Splunk ドキュメントを参照してください](#)。

### 検索ポーリング間隔

このオプションを使用すると、Amazon Managed Grafana が検索結果のスプラUNK をポーリングする頻度を調整できます。[min, max) 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のが理にかなっています。ヒント: 検索ジョブの実行に長い時間がかかる場合は Min、並列検索を多数実行する場合は Max を増やします (Grafana ダッシュボードのスプリングメトリクスが多い)。デフォルトは [500、3000) ミリ秒間隔です。

### 自動キャンセル

指定すると、この数秒間非アクティブ状態になった後にジョブが自動的にキャンセルされます (0 は自動キャンセルされないことを意味します)。デフォルトは 30 です。

### ステータスバケット

生成するステータスバケットが最も多い。0 はタイムライン情報を生成しないことを示します。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可能なフィールドのリストを取得しようとします。

- Quick - プレビューから最初に利用可能な結果を使用する
- full - ジョブが終了するのを待ち、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 (テンプレート変数クエリなど) を使用できません。このオプションは、Splunk の速度を低下させる可能性のある、常に検索されないようにするのに役立ちます。構文は整数で、時間単位は `です[+|-]<time_integer><time_unit>`。例えば「`-1w`」のようにです。[時間単位](#)は にすることができます `s, m, h, d, w, mon, q, y`。

## 変数検索モード

テンプレート変数クエリの検索モード。使用できる値:

- fast - イベント検索のフィールド検出がオフになります。統計検索のイベントまたはフィールドデータはありません。
- smart - イベント検索用の のフィールド検出。統計検索のイベントまたはフィールドデータはありません。
- verbose - すべてのイベントとフィールドデータ。

## 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、raw モードとビジュアルモードの 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、エディタモードの切り替えを選択します。

### Raw モード

次のコード例に示すように、時系列データには `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必要があることに注意してください。[timechart](#) コマンドについては、Splunk [公式検索リファレンス](#)を参照してください。

## Splunk メトリクスと `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。グラフをで正常に動作させるには `mstats`、`timeseries` コマンドと `prestats=t` オプションを組み合わせる必要があります。

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
```

コマンドの詳細については `mstats`、[「Splunk Search Reference」](#) を参照してください。

## 形式

サポートされている結果形式モードは、時系列 (デフォルト) と表の 2 つです。集計データを表示するときにテーブルパネルでを使用するのに適したテーブルモード。これは `raw` イベント (選択したすべてのフィールドを返す) と `stats` 検索関数で機能し、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
```

結果は Splunk UI の統計タブに似ています。

stats 関数の使用方法の詳細については、[「Splunk Search Reference」](#)を参照してください。

## ビジュアルモード

このモードでは、step-by-step検索が作成されます。このモードではtimechart、スプランク検索が作成されます。インデックス、ソースタイプ、メトリクスを選択し、必要に応じてフィールドごとに分割を設定します。

## 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して、複数のメトリクスを追加して検索できます。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。アグリゲートセグメント (avgデフォルトでは) を選択し、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 (または入力します)、必要に応じてエイリアスを設定します。

## と場所による分割

フィールドで分割を設定し、時系列モードを使用する場合、エディタはどこから使用できます。plusを選択し、演算子、集計、値を選択します。例えば、上位 10 の Where avg などです。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 docs」](#)を参照してください。

## オプション

デフォルトのタイムチャート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、[タイムチャートドキュメント](#)を参照してください。

## レンダリングされたスプリント検索

左側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スプリント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。アノテーションは、事前定義された Splunk アラートまたは通常のスプリント検索のいずれかです。

## Splunk アラート

アラート名を指定するか、フィールドを空白のままにして、発せられ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ています。

### Splunk 検索

次の例に示すように、必要なイベントを取得するには、スプランク検索を使用します。

```
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ています。

Event field as text オプションは、フィールド値を注釈テキストとして使用する場合に適しています。次の例は、ログから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\[d*\]: (.*)
```

正規表現では、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。その後、これらのデバイス名を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、値のリストを返す単純なクエリ (前述) です。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 および という名前の 2 つの列を返す必要があります。\_value。\_text 列の値は一意である必要があります (一意でない場合、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、わかりやすい名前をテキストとして、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"
```

## 複数値変数

クエリでは複数値変数を使用できます。補間検索は、変数の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するコンテキストは多数あります。選択した値 `foo` と `$container` を持つ変数があるとしませう `bar`。

- search コマンドの基本フィルター

```
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 インフラストラクチャモニタリングデータソースに接続する

Splunk インフラストラクチャモニタリング (旧) のサポートを提供します SignalFx。

### 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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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 コンソールを開き、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. [データソースを追加する] を選択する。

**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ロールがないことを意味します。

4. データソースのリストから Splunk インフラストラクチャモニタリングを選択します。
5. 次の情報を入力します。
  - アクセストークン には、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。
  - 領域 組織をホストする自己完結型デプロイ。SignalFx ユーザーインターフェイスにサインインすると、プロフィールページで領域名を確認できます。

## 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。

ラベルの場合、Signalflow 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"」

クエリタイプテンプレート変数の場合、クエリフィールドはありません。代わりに、次のいずれかのクエリタイプを選択します。

- デイメンション
- 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いるため、デイメンションを使用したグローバルフィルターが可能になります。

Grafana 注釈がサポートされています。注釈を作成するときは、アラートまたはイベントクエリを使用します SignalFlow。

ディテクターのアラートを取得する例 :

```
alerts(detector_name='Deployment').publish();
```

タイプ別にカスタムイベントを取得する例 :

```
events(eventType='simulated').publish();
```

## Wavefront データソースに接続する (Wavefront による VMware Tanzu Observability)

Wavefront (Wavefront による VMware Tanzu Observability) 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一緒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を管理する](#)」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、このデータソースで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てワークスペースを拡張する](#)」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。StatsD Wavefront を使用すると、ユーザーはこれらのシリーズに対して数学演算を実行し、異常を表示するためのグラフをレンダリングし、KPIs を追跡し、アラートを作成できます。

## Wavefront と Grafana による技術スタックの最大化

表面上では、Grafana と Wavefront は似ているように聞こえますが、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料金を支払うのではなく、データを直接クエリします。

さまざまなソースからパネルを構成する 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 つのペイン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することができるため、データの準備が少なく済みます。

パネル内での結合、コンピューティング、視覚化: Wavefront および Prometheus や InfluxDB など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
- Raw 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

### 構成

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 の 2 つだけですToken。

- API URL は、ウェーブフロント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、 の生成を選択します。
  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、新規アカウントの作成 を選択します。サービスアカウントの名前を入力します。これは何でもかまいません。「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。



- 最後のステップでは、アクセス許可でアカウント、グループ、ロールのチェックボックスが選択されていることを確認します。

トークンを取得したら、それをToken設定フィールドに追加します。

確定し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。これらを切り替えるには、クエリフォームの右上にあるセレクトターを使用します。

クエリビルダーモードでは、次の 4 つの選択肢が表示されます。

- クエリを実行するメトリクス
- 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？
- そのメトリクスクエリの結果をどのようにフィルタリングしますか？
- 結果に追加の関数を適用しますか？

メトリクスセレクトターは分類された階層です。カテゴリを選択し、もう一度を選択してサブカテゴリを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 というラベルが付けられた 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、各メトリクスの適切な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、Wavefront 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、および計算を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、フィールドごとのグループの設定やしきい値の適用など、さら

に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して、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タグを選択します。
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ラベルを入力します。Name 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、ダッシュボードの選択パネルに表示されるテンプレート変数の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. クエリオプション見出しで、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、「テンプレート可変クエリ構造」を参照してください。](#)
8. クエリから返された値のいずれかを除外する場合は、正規表現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、任意の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のダッシュボードを保存 を選択して変更を保存します。

## テンプレート変数のクエリ構造

メトリクスリスト: メトリクス: ts(...)

ソースリスト: ソース: ts(...)

ソースタグリスト: sourceTags : ts(...)

一致するソースタグリスト : matchingSourceTags: ts(...)

タグ名リスト: tagNames : ts(...)

タグ値リスト: tagValues (<tag>): ts(...)

Notes (メモ)

- 各クエリタイプの末尾の はオプションです。

- すべての小文字のサポート。tagnames または tagName 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オプションです

### 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

これで、[新しいテンプレート変数を追加する](#)ステップを完了したら、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な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. フィルターラベルの横にある + 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、ホストフィルター host= の場合は です。
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サイクルスルーし、パネルを動的に更新できるようになります。

## Ad-Hoc 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 つ目のヘルパー変数は、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、メトリクスの選択に使用されます。もう 1 つは実際のアドホックフィルター変数です。

### Important

必須のヘルパー変数には という名前を付ける必要があります metriclink。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するクエリ](#)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 変数を作成した後、[「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手順に従って、アドホックフィルターを追加できるようになりました。違いは、タイプとしてアドホックフィルターを選択し、クエリに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 アノテーションを作成するには、ダッシュボードの右上にある設定タグを選択します。
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、Limit 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はずですが、

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の「名前の表示」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、表示名関数は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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 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 のパネルとビジュアライゼーション](#)
- [Grafana バージョン 10 でを試す](#)
- [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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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構築し、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、設定された任意のからのデータを操作できます[データソースに接続する](#)。

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。その結果、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](#)
- [ダッシュボードの使用状況の評価](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このトピックでは、ダッシュボード機能とショートカットの概要と、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	ダッシュボードの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ド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. ダッシュボードを保存する	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には、 <a href="#"></a> を選択します。詳細については、「 <a href="#">ダッシュボードの使用状況の評価</a> 」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。ダッシュボード設定の詳細については、「 <a href="#"></a> 」を参照してください。 <a href="#">ダッシュボード設定の変更</a> 。
8. タイムピッカードロップダウン	<p>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。</p> <p>タイムゾーンと会計年の設定は、時間範囲コントロールから変更するには、時間設定の変更ボタンをクリックします。</p> <p>時間設定はダッシュボードごとに保存されません。</p>
9. ズームアウト時間範囲	を使用して時間範囲をズームアウトします。時間範囲コントロールの使用法の詳細については、「 <a href="#"></a> 」を参照してください。 <a href="#">ダッシュボードの時間範囲の設定</a> 。
10. ダッシュボードを更新する	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するには、 <a href="#"></a> を選択します。
11. ダッシュボードの時間間隔を更新する	ダッシュボードの自動更新の時間間隔を選択します。
12. 表示モード	テレビやセルフサービス機などの大きな画面にダッシュボードを表示するには、 <a href="#"></a> を選択します。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
13. ダッシュボードパネル	<p>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。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul>
14. グラフの凡例	<p>凡例から直接、シリーズの色、Y 軸、およびシリーズの可視性を変更します。</p>
15. ダッシュボード行	<p>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分割です。</p> <ul style="list-style-type: none"> <li>行を折りたたむか展開して、ダッシュボードの一部を非表示にすることができます。</li> <li>折りたたまれた行内のパネルはクエリを発行しません。</li> <li>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。行の繰り返しの詳細については、「」を参照してください <a href="#">ダッシュボードを作成する。</a></li> </ul>

## キーボードショートカット

Grafana には、多数のキーボードショートカットがあります。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 はセルフサービスモードを切り替えます (メニューを非表示)。

- `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` 全画面表示モードまたは編集モードの場合、 は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、ポインターを使用してパネルにカーソルを合わせます。

- `e` パネル編集ビューを切り替える
- `v` パネルの全画面表示を切り替える
- `ps` パネル共有機能を開く
- `pd` 重複パネル
- `pr` パネルを削除します
- `p1` パネルの凡例を切り替える

## ダッシュボードの時間範囲の設定

Grafana には、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、一般的な時間制御、ダッシュボード全体の時間設定、および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、`s` (seconds)、`m` (minutes)、`h` (hours)、`d` (days)、`M` (months)、`Q` (quarters) および `w` (weeks) の時間単位をサポートしています `y` (years)。

マイナス演算子を使用すると、現在の日付と時刻を基準にして、または をステップバックできます `now`。単位の全期間 (日、週、月) を表示する場合は、末尾 /`<time unit>` に を追加します。会計期間を表示するには、`fQ` (fiscal quarter) および `fy` (fiscal year) 時間単位を使用します。

`plus` 演算子を使用すると、現在を基準にして時間的に先に進むことができます。例えば、この機能を使用して、将来予測データを調べることができます。

次の表に、相対範囲の例を示します。

相対範囲の例	From	目的
過去 5 分	now-5m	now
これまでの日付	now/d	now
今週	now/w	now/w
今のところ、この週	now/w	now
今月	now/M	now/M
今月はここまで	now/M	now
前月	now-1M/M	now-1M/M
今のところ、この年	now/Y	now
本年	now/Y	now/Y
前事業年	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用。
- now-1n/n n の先頭から n の末尾まで。これは絶対タイムスタンプであるためです。

## 一般的な時間範囲の制御

ダッシュボードとパネルのタイムコントロールには、共通のユーザーインターフェイスがあります。以下に、一般的な時間範囲の制御について説明します。

- 現在の時間範囲は、タイムピッカーとも呼ばれ、表示している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。フィールドの上にカーソルを置くと、範囲とそのソースの正確なタイムスタンプ (ローカルブラウザ時間など) 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。現在の時刻は、過去 15 分などの相対時間範囲、または 2020-05-14 00:00:00からなどの絶対時間範囲を使用して変更できます2020-05-15 23:59:59。

- 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、過去 2 年、昨日、昨日の前日、先週のこの日、今日まで、今週まで、今月までなどがあります。
- 絶対時間範囲は、正確な時間値または相対時間値を From フィールドと To フィールドに入力し、時間範囲を適用をクリックするか、From フィールドまたは To 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、特定の日付からのアクティビティを表示するには、開始時刻の絶対時刻と終了時刻の相対時刻 ( などnow) 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徐々にズームアウトして、より多くの履歴とより少ない詳細を表示します。同じ速度で、高いデータ解像度が低下すると、期間全体の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 を使用します)。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により大きな時間範囲が表示されます。
- 視覚化のグラフに表示する時間範囲を選択してズームインします。

#### Note

ズームインはグラフの視覚化にのみ適用されます。

## ダッシュボードを更新する

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すぐに実行され、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。ただし、ダッシュボードを定期的に更新する場合は、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て、更新間隔を選択します。

## URL を使用して時間範囲を制御する

ダッシュボード URL に次のクエリパラメータを指定することで、ダッシュボードの時間範囲を制御できます。

- from は、ms エポック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、ms エポック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、から time-time.window/2 までの時間範囲 time.window を定義します。time+time.window/2。両方のパラメータは ms で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は から 1499999995000 までの 10 秒の時間範囲になります 1500000005000。

## ダッシュボードの構築

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

Grafana ワークスペースを作成してサインインしたら、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、[視覚化されたパネルで構成されています](#)。各パネルには、いずれかの 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 [データソースに接続する](#)。

[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。サーバー、アプリケーション、またはその他の名前をメトリクスクエリでハードコーディングする代わりに、代わりに変数を使用できます。

## トピック

- [ダッシュボードを作成する](#)。

- [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](#)

ダッシュボードを作成する。

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

## 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。視覚化を表示するには、各パネルに少なくとも 1 つのクエリが必要です。開始する前に、以下の前提条件を満たしていることを確認してください。

- 適切なアクセス許可があることを確認します。権限の詳細については、[ユーザー、チーム、アクセス許可](#) をご参照ください。
- 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 Dashboards を選択します。
2. 新規 を選択し、次に新規ダッシュボード を選択します。
3. 空のダッシュボードで、+ 視覚化を追加 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. データソースを選択します。既存のデータソース、Grafana の組み込みデータソースの 1 つをテスト用に選択するか、新しいデータソースを設定するを選択して新しいデータソースをセットアップできます (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)。

編集パネルビューが開き、データソースが選択されます。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構築します。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、必要な情報を正確に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、ページの右上隅にある保存をクリックします。

 Note

メモは、ダッシュボードを以前のバージョンに戻す必要がある場合に役立ちます。

- 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 **として保存** 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. [Save] 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で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、[「パネルの繰り返しの設定」](#)を参照してください。

また、Multi-valueまたは Include all values で設定された変数がある場合は、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、次のステップを完了する必要があります。

繰り返し行を設定するには


1. 左側のメニューからダッシュボードを選択し、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 + 行を追加 ボタンを選択することもできます。

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. 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。

5. [Update] (更新) を選択します。

### 行と Dashboard の特別なデータソースの繰り返し

行に特別な [Dashboard](#) データソース、つまりスマエ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を使用するパネルが含まれている場合、繰り返される行の対応するパネルは、繰り返される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、次のようになります。

- Row 1 には Panel 1Aと が含まれますPanel 1B。
- 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。
- 繰り返しには、Panel 2Aと Row 2が含まれますPanel 2B。
- Panel 2B は Panel 1Aではなく を参照しますPanel 2A。

### 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。ダッシュボードには、任意の場所にパネルを配置できます。

### 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のサイズは、ニーズに合わせて設定できます。

## 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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せん。

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の JSON 定義を選択する必要があります。JSON のインポート方法には 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領域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 URLs 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。
  - Import (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

[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表示されます。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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、必要なものがすべて含まれた 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できるよう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。

タイプのテンプレート変数Constantはダッシュボードに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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボード設定ページでは、次のことができます。

- 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します。
- 注釈クエリを追加します。
- ダッシュボード変数を追加します。
- リンクを追加します。
- 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、**一般** 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. 次の説明に従って時間設定を指定します。
4.
  - Timezone – 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モニタリング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します。エントリはカンマで区切られ、有効な時間単位を受け入れます。
  - Now delay – now遅延時間を入力して時間を上書きします。NULL 値を避けるために、データ集約の既知の遅延に対応するには、このオプションを使用します。
  - タイムピッカーを非表示にする – ダッシュボードにタイムピッカーを表示しない場合は、このオプションを選択します。

#### Note

時間制御を行うには、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、ダッシュボード全体のグラフで縦線として視覚化できます。また、カーソルを合わせるとイベント情報を表示できます。

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. 注釈クエリを追加 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記入して、クエリと注釈を作成します。

クエリエディタ 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、 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[プラグインカタログでプラグインを検索する](#)、[Grafana Labs ウェブサイトのドキュメント](#)を使用できます。

#### 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。サーバー、アプリケーション、センサー名などのものをメトリクスクエリでハードコーディングする代わりに、変数を代わりに使用できます。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 をクリックし、変数を追加 ボタンをクリックします。
2. 一般セクションで、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、ダッシュボードヘッダーのすぐ下に他のダッシュボードやウェブサイトへのリンクを配置できます。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、リンクを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オプションのいずれかを選択して、適用をクリックします。

### Note

タグを使用すると、ダッシュボードの動的なドロップダウンが作成され、すべてに特定のタグが付けられます。

4. ウェブリンクを追加するには、ユーザーがリンクにカーソルを合わせ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、リンクの横にあるアイコンを選択し、ダッシュボードリンクオプションのいずれかを選択します。


## 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 オブジェクトで表されます。ダッシュボードメタデータには、ダッシュボードのプロパティ、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana は、クエリパラメータとして渡された変数値をダッシュボード URLs。詳細については、[「ダッシュボードリンクの管理」](#) および [「テンプレートと変数」](#) 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、プレフィックスが付い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 var- 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
```

クエリパラメータは、ダッシュボード変数の例を の値で var-example=value 表します value。

### 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、 の 2 つの値を持つダッシュボード変数の例 var-example=value1&var-example=value2 として value1 を解釈し value2。

### 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。変数を追加する詳細と手順については、[「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の受け渡し

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の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
```



クエリパラメータは、`example_key`キー、`=`演算子、`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`var-adhoc=key|=value`を適用します。


#### Note

URLsアドホックフィルターと共有する場合は、URL をエンコードすることを忘れないでください。上記の例では、パイプ (`|`) を `%7C`、等価演算子 (`=`) を `%3D` に置き換えます。

## URL を使用した時間範囲の制御

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に設計されています。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 のすべてのインスタンスに反映されます。ライブラリパネルは、複数のダッシュボード間でのパネルの再利用を効率化します。

ライブラリパネルは、保存したダッシュボードと一緒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。

ライブラリパネルを作成するには

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 を選択し、ライブラリパネル を作成します。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. ライブラリの作成パネルを選択して、変更をライブラリに保存します。
7. ダッシュボードを保存します。

ライブラリパネルを作成したら、表示される任意のダッシュボードを使用してパネルを変更できます。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。

詳細... を選択してライブラリパネルを作成 を選択すると、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。

ライブラリパネルをダッシュボードに追加するには

1. 左側のメニューで ダッシュボード を選択します。
2. 新しい 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、+ ライブラリのインポートパネル 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. リストまたは検索をフィルタリングして、追加するパネルを検索します。
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

パネルを変更し、ライブラリパネルの他のインスタンスに影響を与えないようにする場合は、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 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. パネルタイトル (またはパネル上の任意の場所にポインターを置く) 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。パネルが編集モードで開きます。
7. ページの右上隅にあるリンク解除を選択します。
8. はい、リンク解除を選択します。

### 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たり、ライブラリパネルを検索したりできます。

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 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を表示できます。
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。フォルダまたはタイプでパネルを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。

ライブラリパネルを削除するには

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 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に設計されています。  
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、「」を参照してください。[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードのバージョンを保存するたびに、そのバージョンのコピーが保存され、ダッシュボードの以前のバージョンが失われないようにします。これらのバージョンのリストは、ダッシュボードの設定を入力し、左側のメニューでバージョンを選択することで利用できます。

#### Note

ダッシュボードの最新の 20 バージョンが保存されます。

ダッシュボードのバージョン履歴機能を使用すると、以前に保存したダッシュボードの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、リストから比較する 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。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 の差分を表示する場合は、下部の JSON 差分を表示 ボタンをクリックします。

#### 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、差分ビューのバージョン <x> ボタンを選択します。これらのいずれかを選択すると、復元を確認するプロンプトが表示されます。

以前のバージョンに復元すると、以前のバージョンと同じデータを含む新しいバージョンが、別のバージョン番号でのみ作成されます。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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

リンクを使用して、一般的に使用されるダッシュボード間を移動したり、他のユーザーを視覚化に接続したりできます。リンクを使用すると、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 は、ダッシュボードリンク、パネルリンク、および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には、パネルの左上隅にあるアイコンをクリックします。

### 使用するリンクの選択

まず、ダッシュボード間の現在の移動方法を調べます。多くの場合、ダッシュボードのセット間でジャンプし、それぞれで同じコンテキストを見つけるのに苦労している場合、リンクはワークフローの最適化に役立ちます。

次のステップでは、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、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すべてのパネルとは限らないものの、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。
- 外部サイトにリンクする場合は、ダッシュボードリンクまたはパネルリンクを使用できます。
- 特定のシリーズ、または単一の測定にドリルダウンする場合は、データリンクを使用します。

### URL を使用した時間範囲の制御

パネルまたはダッシュボードの時間範囲を制御するには、ダッシュボード URL にクエリパラメータを指定します。

- `from` は、ms エポックで指定された時間範囲の下限を定義します。
- `to` は、ms エポックで指定された時間範囲の上限を定義します。

- `time` および `from` は、`from` から `time-time.window/2` までの時間範囲 `time.window` を定義します。 `time+time.window/2`。両方のパラメータは `ms` で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は `from` から `1499999995000` までの 10 秒の時間範囲になります `1500000005000`。

## 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て、別のダッシュボードの同じ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 `GitHub` に関する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。

ダッシュボードへの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しい 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。
  - タグ付き - 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、デフォルトでは、Grafana はダッシュボードの上部 side-by-side にそれらをすべて表示します。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、リンクがドロップダウンに表示されます。
  - 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、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. [Add] (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 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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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しい 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ができます。
  - 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。
  - 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 – ms エポックで指定された時間範囲の下限を定義します。
    - to – ms エポック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- から time-time.window/2までの時間範囲を定義しますtime+time.window/2。両方のパラメータは ms で指定する必要があります。例えば、

?time=1500000000000&time.window=10000は から 149999995000 までの 10 秒の時間範囲になります1500000005000。

- 変数値 – 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-variable1}=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、次の手順に従います。

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、 の更新を選択します。

### 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ごみ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される独自のリンクセット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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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 を選択します。または、キーボードショートカット を使用できます。
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. リンクを追加 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人間が読めるラベルです。
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 フィールドを選択すると、使用可能な変数が表示されます。パネルリンクにテンプレート変数を追加することで、リンクはユーザーを適切なコンテキストに送信し、関連する変数は既に設定されています。

時間変数を使用することもできます。

- from は、ms エポックで指定された時間範囲の下限を定義します。
  - to は、ms エポック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、 から time-time.window/2までの時間範囲time.windowを定義しますtime+time.window/2。両方のパラメータは ms で指定する必要があります。例えば、?time=1500000000000&time.window=10000は から 1499999995000 までの 10 秒の時間範囲になります1500000005000。
- 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置く) 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 を使用することもできます。

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. 右上の「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置く) 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を使用することもできますe。
    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て、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

注釈は、視覚化上のポイントをリッチイベントでマークする方法を提供します。すべてのグラフパネルで縦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。

視覚化に注釈を付けるには、次の 3 つの方法があります。

- パネル内で直接、[組み込みの注釈を使用して をクエリします](#)。
- 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、「」を参照してください[注釈 API](#)。

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

### パネルに注釈を作成する

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。

パネルに注釈を直接追加するには：

- ダッシュボードは既に保存されている必要があります。
- 組み込み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. [Save] を選択します。

または、注釈を追加するには、Ctrl (または Cmd) を押してパネルを選択すると、注釈の追加ポップアップが表示されます。

### リージョンの注釈

また、視覚化の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。

## リージョンアノテーションを追加するには

1. ダッシュボードで Ctrl (または Cmd) を押しながらパネルのエリアを選択します。
2. 注釈の説明とタグを追加します (オプション)。
3. [Save] (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. [Save] 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、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する新しいクエリを追加できます。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. 注釈クエリの名前を入力します。

この名前はトグル (チェックボックス) に付けられ、このクエリからの注釈イベントの表示を有効にできます。

5. 注釈の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、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、有効化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. イベントマーカースの色を選択します。
9. 「表示」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 – 選択したすべてのパネルに注釈が表示されます。
  - 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。

注釈クエリのオプションは、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については、「」を参照してください[データソースに接続する](#)。

## 組み込みクエリ

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したすべての注釈イベントを取得し、作成されたパネルに表示します。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、組み込みのアノテーションクエリはGrafana特別なデータソースを使用し、手動アノテーションはこのデータソースを使用してのみサポートされます。組み込みのアノテーションクエリでは別のデータソースを使用できますが、自動アノテーションは、そのデータソースのクエリエディタを使用してのみ作成できます。

ダッシュボードに注釈を直接追加するには、このクエリを有効にする必要があります。

組み込み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 Disabled 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) クエリを選択します。
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。新しいアノテーションクエリを追加してタグでフィルタリングすると、それらを表示できます。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

新しいクエリを作成して、データソースを使用して組み込みのアノテーションクエリからアノテーションを取得するGrafanaには、 [タグによるフィルタリング](#) を に設定しますTags。

例えば、注釈クエリ名outagesを作成し、タグ を指定しますoutage。このクエリは、 outage タグを持つすべての注釈 (ダッシュボードから、または API 経由で) 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 [タグによるフィルタリング](#) に一致 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。

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、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に設計されています。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 オブジェクトで表されます。ダッシュボードメタデータには、ダッシュボードのプロパティ、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 を表示するには

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たデフォルト値です。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": []
}

```

以下に、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的な数値識別子 (db によって生成)
uid	誰でも生成できる一意的な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 などのダッシュボードのテーマ light
timezone	utc や などのダッシュボードのタイムゾーン browser
editable	ダッシュボードを編集可能かどうか
graphTooltip	共有クロスヘアまたはツールヒントなしの場合は 0 (デフォルト)、共有クロスヘアの場合は



名前	使用方法
	1、共有クロスヘアと共有ツールヒントの場合は 2
time	last 6 hours や などのダッシュボードの時間範囲 last 7 days
timepicker	timepicker メタデータ、詳細については「 <a href="#">timepicker セクション</a> 」を参照してください。
templating	メタデータのテンプレート化、詳細については「 <a href="#">テンプレート化セクション</a> 」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 <a href="#">注釈</a> を参照してください。
refresh	自動更新間隔
schemaVersion	JSON スキーマ (整数) のバージョン。Grafana の更新によってこのスキーマが変更されるたびに増分されます。
version	ダッシュボード (整数) のバージョン。ダッシュボードが更新されるたびに増加します。
panels	panels 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す。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、それぞれが 30 ピクセルを表します。
- x - 同じ単位の x の位置 w。
- y - 同じ単位の y の位置 h。

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上に移動します。

## タイムピッカー

```
"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

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"
  }
]
}
```

次の表に、テンプレートフィールドの使用法を示します。

名前	使用方法
enable	テンプレートが有効かどうか
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

名前	使用方法
allFormat	、、、 など、データソースからすべての値を取得する際に使用する wildcard glob形式 regex pipe。
current	ダッシュボードに現在選択されている変数のテキスト/値を表示する
datasource	変数のデータソースを表示する
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
options	ダッシュボードで選択できる可変テキスト/値ペアの配列
query	変数の値を取得するため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	は、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	変数のタイプ、custom、query、または interval

## 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このセクションでは、Grafana ダッシュボードを構築および維持する方法について、Grafana 管理者およびユーザー向けのベストプラクティスについて説明します。

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投稿「Grafana ダッシュボード: 構築できるさまざまなタイプの完全なガイド」](#)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。システムについて最もよく知っているので、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、モニタリングすべきことが多数ある場合は、モニタリングに十分な重要性を判断するための戦略が必要です。このページでは、モニタリング対象を選択するためのいくつかの一般的な方法について説明します。

論理戦略を使用すると、統一された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スケーリングできます。

## 戦略のガイドライン

- USE メソッドはマシンの満足度を示し、RED メソッドはユーザーの満足度を示します。
- 問題の原因に関する USE レポート。
- RED はユーザーエクスペリエンスを報告し、問題症状を報告する可能性が高くなります。
- 両方をモニタリングすることは、システムを理解する上で重要です。ベストプラクティスとして、原因ではなく症状に警告します。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になっている時間の割合。
- 飽和 — リソースが行う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 Method](#)」を参照してください。

## RED メソッド

RED は以下を表します。

- Rate – 1 秒あたりのリクエスト数
- エラー – 失敗したリクエストの数。
- 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。サービスごとに、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 SLAs 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 つのゴールドেন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、これら 4 つのメトリクスに集中してください。

この方法は RED メソッドと似ていますが、飽和度が含まれています。

- 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かかるか。
- エラー – 失敗したリクエストのレート。
- 飽和 – システムが「フル」であること、

## ダッシュボード管理の成熟度モデル

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がどの程度適切に設計され、効率的であるかを指します。ダッシュボードの設定を定期的に見直して、現在の状態と改善方法を確認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、高のいずれかに定義できます。

このトピックのコンテンツの多くは、KubeCon 2019 年の [「スリープ提供オンコール用のフルブールフ Kubernetes ダッシュボード」](#) のトーク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。ほぼすべてのユーザーがここで開始されます。

ここにいるとどのように伝えることができますか？

- 誰でもダッシュボードを変更できます。
- コピーされたダッシュボードが多数あり、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。
- 永遠にハングアラウンドする 1 回限りのダッシュボード。
- バージョン管理なし (バージョン管理ではダッシュボード JSON)。
- ダッシュボードの閲覧が多く、適切なダッシュボードが検索されます。つまり、必要なダッシュボードを見つけようとするのに多くの無駄な時間がかかっています。
- 適切なダッシュボードに誘導するアラートがない。

### 中 – 系統ダッシュボード

この段階では、ダッシュボードの使用を系統的なダッシュボードで管理し始めます。戦略を立てたかもしれないませんが、改善できる点がいくつかあります。

ここにいるとどのように伝えることができますか？

- 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、同じダッシュボードを異なるクラスター間で再利用したり、バックエンドをモニタリングしたりできます。

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](#) に従った方法論的なダッシュボード。
- 次のレベルへのドリルダウンを含む階層ダッシュボード。
- ダッシュボード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 行で RED メソッドを使用できます。ダッシュボードを下にスクロールすると、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。
- 「いいね」と比較します。大きさが異なる場合は、サービスダッシュボードを分割します。集計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。
- 色を意味のある方法で使用し、可能な軸を正規化する表現グラフ。
  - 意味のある色の例: 青は良いことを表し、赤は悪いことを表しています。[しきい値](#)が役立ちます。
  - 軸の正規化の例: CPU 使用率を比較するときは、マシンのコア数が異なる可能性があるため、raw 数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、ビューワーは CPU 数 CPUs。
- 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、ランダムまたは目的のない参照が困難になります。
  - ほとんどのダッシュボードは、アラートによって [リンク](#)されている必要があります。
  - 閲覧はリンクで誘導されます。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 – 最適化された使用

この段階では、一貫性のある考え抜かれた戦略でダッシュボード管理の使用を最適化しました。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す。
  - 既存のダッシュボードを定期的を確認し、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。
  - 追跡ダッシュボードの使用。Usage [Insights](#) を活用できます。
- 設計上の一貫性。
- 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)
- ブラウザでは編集されません。ダッシュボードビューワーは、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。
- 本番稼働用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の 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の概要を説明します。

### 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以下に示します。

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必要があります

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、一般的から特定へなど、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問してください）。

グラフをシンプルに保ち、質問する質問への回答に集中してください。例えば、「どのサーバーに問題がありますか？」という質問の場合、すべてのサーバーデータを表示する必要はありません。問題のあるデータを表示するだけです。

ダッシュボードは、認識負荷を軽減し、それに追加しないようにする必要があります

認識負荷とは、基本的に、何かを解き明かすために考えておく必要があるものです。ダッシュボードの解釈を容易にします。他のユーザーや将来 (2AM 時に何が壊れたのかを把握しようとする場合) はそれを評価します。

自問してください。

- 各グラフが正確に何を表すかわかりますか？ 明確ですか、それとも考えなければなりませんか？
- これを他のユーザーに表示した場合、その人物がそれを把握するのにどれくらいの時間がかかりますか？ それらは失われますか？

## モニタリング戦略がある


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て計画に従うことは難しいですが、それだけの価値があります。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必要があります。

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、それらを書き留めます。

## 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、わかりやすい名前になっていることを確認してください。
  - 再生または実験するダッシュボードを作成する場合は、名前に TEST または TMP 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の使用が完了したら、削除します。
- 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、このセクションの後半にある[「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](#) 全般と特定のデータソースに関する基本的な理解は重要です。
- ネットワークまたはバックエンドの負荷を軽減するために、不要なダッシュボードの更新は避けてください。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場合は、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指示事項などを記録します。
  - 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 i 左上隅にある小さな  にカーソルを合わせると表示されます。

- ダッシュボードを再利用し、[テンプレートと変数](#)を使用して一貫性を持たせます。
- グラフデータのスタックには注意してください。視覚化は誤解を招く可能性があり、重要なデータを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際に従うべきベストプラクティスの概要を説明します。

### 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以下に示します。

### 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。これらを調べて、そのうちの1つが自分に合っているか、それとも自分で考えてみるかを決定する必要があります。いずれにしても、計画を立てて書き留め、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はどのくらいですか？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。自分がどこにいるかを理解することは、自分がどこにいるかを判断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、ダッシュボードが制御されないほど増加します。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す。ダッシュボードの複製と「1つのモノ」の変更(悪い:元のタグを保持する)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作成する場合、何かをテストするには、名前の前に `TEST:` を付けます。完了したら、ダッシュボードを削除します。
- 大きな変更が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した。

- 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だけのコピーが作成されています。代わりに、マスターダッシュボードへのリンクを保持し、[URL パラメータ](#)を使用してビューをカスタマイズすることでこれを行う必要があります。
- 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。これは、いくつかの方法で行うことができます。
- 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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ワークスペースの Dashboards ページ (左側のメニューから Dashboards 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、「」を参照してください[ダッシュボードの構築](#)。

### ダッシュボードを参照する

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。

- フォルダ間でダッシュボードを移動します。
- 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、「[ダッシュボードとフォルダのアクセス許可](#)」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ダッシュボードや複数のチームが多数ある場合に役立ちます。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 管理者権限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、「」を参照してください[ダッシュボードとフォルダのアクセス許可](#)。

### 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「新規フォルダ」を選択します。
4. 一意の名前を入力し、 の作成 をクリックします。

#### Note

ダッシュボードを保存するときに、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. ヘッダーのタイトルの編集 (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、および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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、大画面に最適です。Dashboards 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使用して、既存のプレイリストにアクセスします。使用可能な 5 つの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および制御します。



## 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

プレイリストは5つの異なるビューモードで開始できます。表示モードは、メニューとナビゲーションバーをダッシュボードに表示する方法を決定します。

デフォルトでは、各ダッシュボードは、プレイリストを作成または編集するときに設定した間隔フィールドに入力された時間で表示されます。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で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見つけ、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 <プレイリスト名> をクリックします。

プレイリストには、プレイリストを作成または編集するときに設定された、Intervalフィールドで指定された時間の各ダッシュボードが表示されます。プレイリストが起動したら、画面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。

モード	説明
通常モード	<ul style="list-style-type: none"><li>• サイドメニューは引き続き表示されます。</li><li>• 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。</li></ul>
TVモード	<ul style="list-style-type: none"><li>• 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) は非表示または削除されません。</li></ul>



モード	説明
	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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 <code>?inactive</code> にパラメータを追加して手動で有効にします。</li> <li>ポインタの移動またはキーボードアクションで無効にします。</li> </ul>
TV モード (自動フィットパネル付き)	<ul style="list-style-type: none"> <li>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。</li> <li>ダッシュボードパネルは自動的に調整され、画面上のスペースが最適化されます。</li> </ul>
Kiosk モード	<ul style="list-style-type: none"> <li>サイドメニュー、ナビゲーションバー、行、およびパネルのコントロールは、完全に非表示/非表示になります。</li> <li>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。</li> <li>同じショートカットを使用して手動で無効にできません。</li> </ul>
Kiosk モード (自動フィットパネル付き)	<ul style="list-style-type: none"> <li>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は、完全に非表示/表示から削除されます。</li> <li>ダッシュボードパネルは自動的に調整され、画面上のスペースが最適化されます。</li> </ul>

## 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。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。

Button	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 (二乗)	プレイリストを終了し、現在のダッシュボードに終了します。
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表示モードでローテーションします。
[Time range] (時間範囲)	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、またはカスタム時間範囲を表示するように設定できます。
更新 (円矢印)	ダッシュボードを再ロードして、現在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 日ごとに自動的に再ロードするように設定できます。

## 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た順序でダッシュボードを表示できます。

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. 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. Dashboards で、タイトル別追加およびタグ別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 :

- ダッシュボードを名前、正規表現、またはタグで検索します。
  - スターステータスまたはタグで結果をフィルタリングします。
  - 上矢印と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. 保存 を選択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、プレイリストページに追加して開始できます。

### Important

プレイリストを保存する前に、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. プレイリストを選択すると、使用可能なプレイリストが表示されます。
3. 選択した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、および少なくとも 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 を選択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、再配置するか、プレイリストを削除します。

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。

- 名前と時間間隔を更新し、上記の「プレイリストの作成」の手順を使用して、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。
- 保存 を選択して変更を保存します。

プレイリストを削除するには

- プレイリスト を選択します。
- 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

- 移動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。
- 保存 を選択して変更を保存します。

ダッシュボードを削除するには

- 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。
- 保存 を選択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

希望するビューモードでリンクアドレスをコピーし、URL を送信先に貼り付けることで、プレイリストを共有できます。

プレイリストをビューモードで共有するには

- Dashboards の左側のメニューから、プレイリスト を選択します。
- 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。
- 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。
- リンクアドレスをコピーを選択して、URL をクリップボードにコピーします。
- URL を送信先に貼り付けます。

## 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana を使用すると、ダッシュボードやパネルを組織内の他のユーザーと共有できます。特定の状況では、ウェブ上で公開できます。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イメージを表示するには、許可されたビューワーの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。リンクがあるユーザーは誰でもアクセスできます。スナップショットには表示の認証を必要としないため、Grafana は、スナップショットから取得したアカウントに関連する情報と機密データを削除します。

### 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。

#### **i** 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。

## 直接リンクの共有

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. [コピー] を選択します。このアクションは、デフォルト URL 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。

## スナップショットの発行

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削除し、ダッシュボードに埋め込まれているメトリクスデータとシリーズ名のみを表示します。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発行するには

1. 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 を選択します。
3. Grafana は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パブリックに共有します。

## 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、「」の「インポートとエクスポート」セクションを参照してください[ダッシュボードの構築](#)。

## 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。オプションで、短縮 URL を有効にして共有できます。

### 直接リンクを使用するには

1. Copy 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。

## パネルの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 は機密データを削除し、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。

### パネルのスナップショットを発行するには

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. 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パブリックに共有します。

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

共有パネルダイアログボックスからライブラリパネルを作成するには。

ライブラリパネルを作成するには

1. ライブラリパネル 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 で、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. ライブラリの作成パネル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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、パネルのメトリクスクエリが新しい値を反映するように変更されます。

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、センサー名などのものをハードコーディングする代わりに、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、完全な編集権限を与えたくない管理者にとって特に役立ちます。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も作成できます。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、変数を含む任意のクエリです。例えば、ダッシュボードを管理して複数のサーバーをモニタリングする場合、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、次のような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、常に構文 `var-<varname>=value` 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 の変数リストにリスト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、最初に表示されます (ダッシュボードの左端)。
- 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、次のように変数設定を変更します。
  - 「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ます。
  - Custom all value 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に設計されています。  
 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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

次の表に、Grafana の変数のタイプを示します。

変数タイプ	説明
Query	メトリクス名、サーバー名、センサー IDs、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。
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と値のフィルター

変数タイプ	説明
	(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。
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の設定](#)
- [グローバル変数](#)
- [連鎖変数](#)
- [変数の管理](#)
- [正規表現で変数をフィルタリングする](#)

### 一般的なオプションの入力

作成する任意のタイプの変数の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 リストで、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選択します。
  - 選択なし (空白) – 変数ドロップダウンには変数名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ことができるデータソースクエリを作成できます。例えば、クエリ変数は、サーバー名、センサー IDs、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フェッチすると変化します。

クエリ変数は通常、文字列でのみサポートされます。クエリが数値や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る場合があります。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、[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。
  - ダッシュボードロード時 — ダッシュボードがロードされるたびにデータソースをクエリします。これにより、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、ダッシュボードのロードが遅くなります。
  - 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ると、データソースをクエリします。このオプションは、変数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`。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で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の線の上にカーソルを移動し、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式を入力して、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、「[正規表現で変数をフィルタリングする](#)」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を選択します。デフォルトのオプション Disabled 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ることを意味します。
7. (オプション) [選択オプションを入力します](#)。
8. 値のプレビューで、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 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、変更されないサーバー名やリージョン名がある場合は、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。スペースと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` です。
3. (オプション) [選択オプション](#) を入力します。
4. 値 のプレビューで、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## テキストボックス変数の追加

テキストボックス変数には、オプションのデフォルト値を含むフリーテキスト入力フィールドが表示されます。任意の値を入力できるため、これは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、または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 のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 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。これは、共有するダッシュボードのメトリクスパスプレフィックスに役立ちます。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。

定数変数には柔軟性がありません。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が、すべてのクエリで再入力したくない複雑な値がある場合に便利です。例えば、というサーバーパスがある場合 `i-0b6a61efe2ab843gg`、それを という変数に置き換えることができます `$path_gg`。

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。文字、数字、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 のプレビューで、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

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。これらは、データソースのインスタンスが複数ある場合に便利です。場合によっては、異なる環境にもあります。

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. Type リストで、変数のターゲットデータソースを選択します。

Open advanced data source picker を選択して、データソースの追加 (管理者のみ) 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 で、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変数値のドロップダウンリストから入力します。すべてのインスタンスを表示するには、このフィールドを空のままにします。
4. (オプション) [選択オプション](#) を入力します。
5. 値 のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 間隔変数の追加

間隔変数を使用して、1m、1h、などの時間範囲を表します1d。これらは、ダッシュボード全体のグループとしてタイムコマンドで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。自動オプションを使用して、時間範囲ごとに設定された数のデータポイントを返すこともできます。

間隔変数は、時間 (InfluxDB の場合)、日付ヒストグラム間隔 (Elasticsearch の場合)、または要約関数パラメータ (Graphite の場合) でグループ化するパラメータ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。次の時間単位がサポートされています: s (seconds)、m (minutes)、h (hours)、d (days)、w (weeks)、M (months)、および y (years)。デフォルト値を承諾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) オプションをリストに追加する場合は、自動auto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割して値を計算した回数を選択します。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30分単位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 — ステップカウント間隔が時間を割かない最小しきい値。30分の例を続けるには、最小間隔が2mに設定されている場合、Grafanaはデータを15の2分単位でグループ化します。
4. 値のプレビューでは、Grafanaは現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

### 間隔変数の例

次の例は、Graphite 関数myintervalのテンプレート変数を示しています。

```
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すると、指定し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と値の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記述します。

### 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。

Open advanced data source picker を選択して、データソースの追加 (管理者のみ) 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、変数をダッシュボードに追加します。

## アドホックフィルターを作成する

アドホックフィルターは、利用可能な最も複雑で柔軟な変数オプションの 1 つです。この変数では、変数オプションの通常のリストの代わりに、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。この方法で適用したフィルターは、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の設定

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オプションであり、デフォルトではオフになっています。

## 複数値変数

複数の値が選択された変数を補間することは、変数が使用され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、難しいです。Grafana は、各データソースプラグインが複数の値に使用する形式をテンプレート補間エンジンに通知できるようにすることで、この問題を解決しようとしています。

**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、変数のカスタムオール値オプションが空白である必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (結合) します。例えば `value1,value2,value3` です。カスタムall値を使用する場合は、代わりに `*または` になりますall。

### 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 {host1,host2,host3}` であるかのように補間されます。

### Prometheus または InfluxDB データソースを使用する複数値変数

InfluxDB と Prometheus は正規表現式を使用するため、同じ変数がとして補間されます (`host1|host2|host3`)。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、正規表現制御文字を持つ値は正規表現式を破棄します。

### 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、同じ変数がとしてフォーマットされます (`"host1" OR "host2" OR "host3"`)。この場合、値に Lucene コントロールワードと引用符のみが含まれるように、すべての値がエスケープされます。

### 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把握するのは難しい場合があります。特に InfluxDB および Prometheus では、正規表現構文を使用するには、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選択します。
- [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。

## すべてのカスタム値

このオプションは、Include All オプションが選択されている場合にのみ表示されます。

Custom all value フィールドに regex、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のように、カスタムのオール値を指定する方がよい場合があります。

カスタム正規表現、globs、または Lucene 構文を Custom all value オプションに含めるには、エスケープされなため、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リストし、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、および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つの時間範囲変数が組み込まれています。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ますが、日付の書式を制御できます。

構文	結果の例	説明
<code>\${__from}</code>	1594671549254	Unix ミリ秒エポック
<code>\${__from:date}</code>	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
<code>\${__from:date:iso}</code>	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
<code>\${__from:date:seconds}</code>	1594671549	Unix 秒エポック

構文	結果の例	説明
<code>\${__from:date:YYYY-MM}</code>	2020-07	: 文字を含まないカスタム日付形式

上記の構文は `でも機能します${__to}`。

## `$__interval`

`$__interval` 変数をパラメータとして使用して、時間 (InfluxDB、MySQL、Postgres、MSSQL の場合)、日付ヒストグラム間隔 (Elasticsearch の場合)、または要約関数パラメータ (Graphite の場合) でグループ化できます。

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くのデータポイントがある場合、間隔を長くしてグループ化することでクエリをより効率的に行うことができます。例えば、3 か月分のデータのグラフを見ると、分レベルで詳細を表示できない場合があります。時間または日でグループ化すると、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。

概算計算:  $(to - from) / resolution$

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 `1h` に計算され、ポイントは 2m 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、ポイントは日別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` を使用する必要があります。

InfluxDB および Elasticsearch データソースには、間隔をハードコードしたり、`$__interval` 変数の最小制限を設定したりするために使用する `Group by time interval` フィールドがあります (などの `>` 構文を使用 `>10m`)。

## `$__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です。

## `$__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`

Prometheus および Loki データソースでのみサポートされます。この変数は、現在のダッシュボードの範囲を表します。これは `to - from`。ミリ秒と および という 2 `$__range_ms` 番目の表現があります `$__range_s`。

## `$__rate_interval`

Prometheus データソースでのみサポートされます。 `$__rate_interval` 変数は `rate` 関数で使用するのためのものです。

## `$timeFilter` または `$__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、時間範囲間隔 `Last 7 days` 式は `time > now() - 7d`。

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。Grafana は、クエリエディタモードで 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手動で追加できます: `WHERE $timeFilter`。
- Azure Monitor データソースに分析クエリをログ記録します。
- MySQL、Postgres、MSSQL の SQL クエリ。
- `$__timeFilter` 変数は MySQL データソースで使用されます。

## `$__タイムゾーン`

`$__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: `)` を返します `America/New_York`。

現在選択されているタイムゾーンがブラウザの時刻 の場合、Grafana はブラウザのタイムゾーンを決定しようとしています。

## 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 と呼ばれ、変数クエリに 1 つ以上の他の変数を含むクエリ変数です。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべてのもので同じです。連鎖変数クエリは、それらを許可す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、深さが 5 レベルまたは 10 レベルまで可能です。技術的には、どれだけ深く、複雑に行け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。

### リンクされた新しい変数の作成

- 変数を連鎖すると、親/子の依存関係が作成されます。これらは、はしごまたは木として構想できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数のベース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重複変数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 を追加できます。
- この方法で作成された新しい変数は、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

各エントリの右側にある上矢印と下矢印をクリックして、ダッシュボード可変リスト内の可変の順序を変更できます。Grafana は、このリストに従って変数ドロップダウンを左から右に一覧表示し、変数は左端の上部にあります。

- 子変数の前に、依存関係のない変数をリストします。
- 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、依存関係を持つ変数が表示されないことに注意してください。他のユーザー (および自身) に対して簡単に使用できるように、変数を論理的な順序で一覧表示します。

### 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、リンクされた一連の変数 (国、リージョン、サーバー、メトリクス) があり、ルート変数値 (国) を変更する場合、Grafana はダッシュボードの視覚化を更新する前に、すべての依存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、変数を追加したり、既存の変数を管理したりできます。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特定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** — 変数のクローンを作成するには、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、というプレフィックスが付いた元の変数の名前を持つ変数のコピーが作成されます `copy_of_`。

**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。

## 正規表現で変数をフィルタリングする

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リスト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リスト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**02**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オプションをフィルタリングおよび変更し、テキストの一部を返します。

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の場合 :

```
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
```

この正規表現 :

```
/. *instance="  
(  
[^\"]*  
)  
.*/  

```



次の結果を返します。

```
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個別の「テキスト」部分と「値」部分を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、選択できる各値の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\_namesPrometheus 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いです。そのため、次の変数クエリ結果が得られます。

```
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
```

次の正規表現を通過します。

```
/chip_name="(?(<text>[ ^ " ] + ) |chip="(?(<value >[ ^ " ] + )/g
```

次のドロップダウンリストを生成します。

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

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表示されます。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、依存関係マップを表示できます。依存関係マップを移動できます。マウスホイールまたは同等のもの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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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` – この構文は読みやすいですが、単語の途中で変数を使用することはできません。

例: `apps.frontend.$server.requests.count`

- `${var_name}` – 式の途中で変数を使用する場合は、この構文を使用します。
- `${var_name:<format>}` – この形式により、Grafana が値を解釈する方法をより詳細に制御できます。詳細については、次のリストの「高度な変数形式オプション」を参照してください。
- `[[varname]]` – は使用しないでください。この構文は古く、廃止されました。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。補間中に、クエリ言語の構文とその使用場所に準拠するために、変数値をエスケープすることがあります。例えば、InfluxDB または Prometheus クエリの正規表現式で使用される変数は、正規表現をエスケープします。

## 高度な変数形式オプション

変数補間の形式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したい場合があります。

例えば、MySQL データソースのデフォルトは、複数の値を引用符で区切って結合することです `'server01', 'server02'`。場合によっては、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: `server01,server02`。これは、以下に示す高度な変数フォーマットオプションで実現できます。

### 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

複数の値を持つ変数をカンマ区切りの文字列としてフォーマットします。

```
servers = [ 'test1', 'test2' ]
String to interpolate: '${servers:csv}'
Interpolation result: 'test1,test2'
```

### 分散型 - OpenTSDB

OpenTSDB のカスタム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:distributed}'
Interpolation result: 'test1,servers=test2'
```

### 二重引用符

単一値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、`\"`によって各値をエスケープし、各値を `"` 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

複数の値を持つ変数をカンマ区切りの文字列として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")'
```

## パーセントエンコード

URL パラメータで使用する単一値変数と複数值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%20test2'
```

## パイプ

複数の値を持つ変数をパイプ区切りの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 クエリの一重引用符など、データソース固有の書式設定をオフにします。

```
servers = [ 'test.1', 'test2' ]  
String to interpolate: '${var_name:raw}'  
Interpolation result: 'test.1,test2'
```

## [Regex] (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)'
```

### シングルクオート

単一値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、 によって各値をエスケープ'、各値を で引用符で囲みます'。

```
servers = [ 'test1', 'test2' ]
String to interpolate: '${servers:singlequote}'
Interpolation result: "'test1','test2'"
```

### SQLstring

単一値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、 によって各値をエスケープ'、各値を で引用符で囲みます'。

```
servers = [ "test'1", "test2" ]
String to interpolate: '${servers:sqlstring}'
Interpolation result: "'test''1','test2'"
```

### [Text] (テキスト)

単一値および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、テキスト表現を と組み合わせた値を返します+。

```
servers = [ "test1", "test2" ]
String to interpolate: '${servers:text}'
Interpolation result: "test1 + test2"
```

### クエリパラメータ

単一値および複数値の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"
```

## ダッシュボードの使用状況の評価

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

すべてのダッシュボードとデータソースについて、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 Dashboard Insights を選択します。

ダッシュボードインサイトには、次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に対する最後のアクティビティ (上限は 20)。

### データソースインサイト

データソースインサイトは、データソースが過去 30 日間に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. Insights 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見ると、最近ダッシュボードを操作したユーザーのアバターを表示するプレゼンスインジケータを使用して、同じダッシュボードを誰が見ているかがわかります。デフォルトの期間は 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。これにより、ダッシュボードのインサイトが開き、最近のユーザーアクティビティの詳細が表示されます。

### 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、最も使用頻度の高いダッシュボード、壊れたダッシュボード、未使用のダッシュボードを検索できます。ダッシュボードは、次のようにソートできます。

- ビュー
- エラー
- ビュー
- 作成時刻
- 更新時刻

### 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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

ダッシュボードの一般的な問題のトラブルシューティングには、次の戦略を使用します。

## ダッシュボードが遅い

- グラフで数十 (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 highestMaxで) 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可能性があります。これにより、レスポンスサイズが大きくなります。を使用してalias、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？ これらの条件の両方により、Grafana またはデータソースが大量のデータを取り込む可能性があり、ダッシュボードが遅くな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い負荷がかかる可能性があります。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にクエリを実行します。ただし、ダッシュボードの更新レートを低く設定すると、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、これを頻繁にクエリする必要はありません。

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、意味のある長い期間に更新レートを設定します。
- 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 週間など長い場合は、自動更新は必要なく、無効にする必要があります。

## 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、欠落している値がある行を接続します。



これを修正する 1 つの方法は、線の代わりに棒を使用し、(標準オプションで) 値なしオプションを設定することです。この場合、欠損データはグラフの領域としてデータなしで表示されます。

## 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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

ダッシュボードは、ダッシュボード名とパネルタイトルで検索できます。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任意の部分を入力します。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
- 大文字と小文字を区別しない
- 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。

### **i** Tip

キーボードの矢印キーを使用して結果に移動し、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

タグは、特にダッシュボードの数が増えるにつれてダッシュボードを整理する優れた方法です。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。

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、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、タグによる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用して、次の操作を実行できます。

- 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。
- テーマを暗い色または明るい色に変更します。

コマンドパレットを開き、ctrl+k (cmd+kMacOS で) 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**

前のステップに進むには、コマンドパレットを空backspaceにして を押します。

## Grafana バージョン 10 のパネルとビジュアライゼーション

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

パネルは Grafana の基本的な視覚化構成要素です。各パネルには、パネルで選択されたデータソースに固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書式設定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。

ワークスペースに追加の[プラグイン](#)をインストールすることで、追加のパネルタイプが使用できるようになる場合があります。

特定のデータソースの使用の詳細については、「」を参照してください[データソースに接続する](#)。

### 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](#)
- [データリンクを設定する](#)

- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。

新しいダッシュボードにパネルを追加するには、ダッシュボードの中央で + 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で追加を選択し、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、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** — パネルを全画面表示で表示します。
- **編集** — パネルエディタを開いて、パネルと視覚化の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。
- **探索** — 探索 でパネルを開き、クエリに集中できます。
- **Inspect** — Inspect ドロワーを開きます。このドロワーでは、パネルデータ、統計、メタデータ、JSON、クエリを確認できます。

- データ – データタブで検査ドロワーを開きます。
- Query – Query タブで Inspect ドロワーを開きます。
- パネル JSON – JSON タブで Inspect ドロワーを開きます。
- 拡張機能 – インシデントの宣言など、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。
- 詳細 – 他のパネルアクションにアクセスします。
  - 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をクエリします。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 – 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ます。ここでは、パネルクエリに基づいて [Grafana 管理のアラート](#)を作成できます。
  - 凡例を非表示 - パネルの凡例を非表示にします。
  - ヘルプを受ける – 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以下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パネルに加えた変更を破棄します。
  - 保存 – パネルに加えた変更を保存します。
  - 適用 – 行った変更を適用し、パネルエディタを閉じてダッシュボードに戻ります。適用された変更を永続化するには、ダッシュボードを保存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、データを表示できます。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 raw 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 — 視覚化プレビューは、使用可能な領域を塗りつぶします。サイドペインの幅または下部ペインの高さを変更すると、使用可能なスペースを埋めるように視覚化が変更されます。
- 実際 — 視覚化プレビューのサイズは、ダッシュボードのサイズとまったく同じになります。十分なスペースが利用できない場合、視覚化はアスペクト比を保持しながらスケールダウンします。
- 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上位レベル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。最初にダッシュボードを作成すると、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 — アラートルールを書き込みます。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、データの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 raw データを表示したり、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たり、クエリリクエストを表示したり、パネルとデータ JSON をエクスポートしたりできます。

### Note

すべてのパネルタイプにすべてのタブが含まれてい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 raw 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 — クエリによって返された raw データに、変換が適用された状態を示します。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と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 をプロビジョニングまたは管理する場合に便利です。
- クエリタブ — Grafana がデータソースにクエリを実行したときに送信された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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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 raw データを検査したり、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たり、クエリリクエストを表示したり、パネルとデータ JSON をエクスポートしたりできます。

### **i** Note

すべてのパネルタイプにすべてのタブが含まれてい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 raw 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は、ペインの上部に Inspect: と表示します。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 – クエリによって返された raw データに、変換が適用された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と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 をプロビジョニングまたは管理する場合に便利です。
5. クエリタブ – Grafana がデータソースにクエリを実行したときにサーバーに送信された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。

### raw クエリ結果のダウンロード

Amazon Managed Grafana は、そのデータへの変換を含むデータを含む CSV ファイルを生成します。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または後にデータを表示するように選択できます。

#### raw 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. データ を選択します。

パネルに複数のクエリまたは複数のノードが含まれている場合は、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。
  - データ変換
  - 時間で結合 – 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 1 つの結果セットを表示します。列見出しを選択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 専用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、Download for Excel のトグルを選択します。
  6. [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、送信するクエリの数、および返された行数を示す統計が表示されます。この情報は、特に数値が予期せず高い場合や低い場合に、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. 統計 を選択します。

統計は読み取り専用形式で表示されます。

## クエリの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をトラブルシューティングする場合や、期待される結果を返さないクエリの場合は、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. の更新 を選択します。

パネルに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、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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](#)
- [クエリタブに移動する](#)
- [クエリの追加](#)
- [クエリの管理](#)
- [クエリオプション](#)
- [式クエリの書き込み](#)
- [クエリ結果を別のパネルと共有する](#)
- [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、視覚化用のデータを取得するクエリを介してデータソースと通信します。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 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は、独自の機能を利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を提供します。

クエリ言語が異なるため、各データソースクエリエディタは異なる検索と機能を行います。データソースによっては、クエリエディタによって自動補完機能、メトリクス名、変数の提案、または視覚的なクエリ構築インターフェイスが提供される場合があります。

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、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、「」を参照してください[Enterprise データソースに接続する](#)。
- 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[プラグインカタログでプラグインを検索する](#)、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ています。

## クエリ構文

データソースは、さまざまな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、その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>)
```

## 特殊なデータソース

Grafana には、Grafana、Mixed、Dashboard 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。

## クエリタブに移動する

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ー — クエリするデータソースを選択します。
- クエリオプション — 最大データ取り出し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、「[式クエリの書き込み](#)」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. [Queries] (クエリ) タブを開きます。
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 はデータソースをクエリし、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、クエリ为非表示、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ー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** – 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、または別の関数でそれらをまとめて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、視覚化された行を滑らかにしたりできます。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** — 自動的に計算される間隔の最小制限を設定します。通常は最小スクレイプ間隔です。データポイントが 15 秒ごとに保存される場合、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、これをスクレイプ間隔よりも高い最小値に設定して、より粗く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ごとに集計またはグループ化するときを使用できる時間範囲を設定します。

Grafana は、テンプレート化されたクエリの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、これは間隔変数を使用する Prometheus クエリです `rate(http_requests_total[$__interval])`。

この自動間隔は、グラフの幅に基づいて計算されます。ユーザーが視覚化を拡大すると、間隔が大きくなり、より粗粒度の集計になります。同様に、ユーザーがズームインすると間隔が短くなり、よりきめ細かな集計になります。

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のタイムピッカーで選択されているものとは異なります。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

パネルの時間オーバーライドは、ダッシュボードの時間範囲が絶対である場合は効果がありません。

例	相対時間フィールド
過去 5 分	now-5m
これまでの日付	now/d
過去 5 日間	now-5d/d
今のところ、この週	now/w
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

パネルの時間オーバーライドは、ダッシュボードの時間範囲が絶対である場合は効果がありません。

例	タイムシフトフィールド
過去 1 週間	1w/w
2 週間前	2w/w
過去 1 か月間	1M/M
この 1 年間	1d/y
過去 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にのみ表示) 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上書きします。この値を秒単位の数値として指定します。

## 式クエリの書き込み

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

サーバー側の式を使用すると、数学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、数学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のマイナーデータ再構築を除いて、データソースによって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](#) によって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、式はバックエンドのデータソースやビジュアライゼーションでも使用できます。

#### **i** 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、さまざまなデータソースからのクエリを組み合わせることで有効にするか、データソースで使用できないオペレーションを提供することで、データソースを強化することを目的としています。

#### **i** 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的であるため、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、複数のシリーズを返すクエリで、各シリーズがラベルまたはタグによって識別されます。

個々の式は、1つ以上のクエリまたは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それぞれ、RefID と呼ばれる名前付き識別子である変数 (デフォルト文字Aまたは などB) で表されます。

個々の式または別の式の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 種類のデータで機能します。

- 時系列のコレクション。
- 数値のコレクション。各数値は項目です。

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。データは一般的に、時系列データにラベル付けされていると見なされます。

データソースクエリは、式とともに使用すると、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 1 つの時系列またはデータフレームごとに 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シリーズを返すデータソースを使用する場合、式で実行すると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、時間なしのデータフレーム、文字列列、および 1 つの数値列を返すテーブルレスポンスのみです。

次の例の表は、式で動作する数値を生成します。文字列列はラベルになり、数値列は対応する値になります。例えば、値が `1 {"Loc": "MIA", "Host": "A"}` の場合です。



Loc	ホスト	平均 CPU
MIA	A	1
NYC	B	2

## オペレーション

式では、数学、削減、再サンプリングのオペレーションを使用できます。

### 数学

Math は、時系列または数値データの自由形式の数式用です。数学オペレーションでは、数値と時系列を入力として受け取り、それらを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、ドル記号のプレフィックスが付いた RefID で参照されます \$A。例えば、。変数の名前にスペースが含まれている場合は、のような中括弧構文を使用できます `{my variable}`。

数値定数は、10 進数 (2.24)、8 進数 (のように先頭に 0 を付ける 072)、または 16 進数 (のように先頭に 0x を付ける) にすることができます 0x2A。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 (+、バイナリおよび単項 -、\*、/、%、指数 \*\*)、リレーショナル (<、>、==、<=) !=>=、論理 (&&、||、および単項 !) 演算子がサポートされています。

データに対するオペレーションの動作は、数値データか時系列データかによって異なります。

`$A + $B` やなどのバイナリオペレーションでは `$A || $B`、データの種別に応じて以下の方法で演算子が適用されます。

- `$A` と `$B` の両方が数値の場合、オペレーションは 2 つの数値の間で実行されます。
- 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の間のオペレーションが実行されます。
- `$A` と `$B` の両方が時系列データである場合、2 つのシリーズ内の各値間のオペレーションは、`$A` と `$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 `$B`。Resample オペレーションを使用して、タイムスタンプをラインアップできます。

## 概要 :

- 数値 <Operation> number = number
- 数値 <Operation> シリーズ = シリーズ
- シリーズ <Operation> シリーズ = シリーズ

式は 1 つの変数で表される複数のシリーズまたは数値で動作するため、バイナリオペレーションは 2 つの変数間の結合 (結合) も実行します。これは、個々のシリーズまたは番号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、`{host=web01}`に のようなラベルを持つ数値\$Aと、同じラベル\$Bを持つ に別の数値がある場合、各変数内のこれら 2 つの項目間でオペレーションが実行され、結果は同じラベルを共有します。このユニオン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 の両方\$Bに 1 つの項目 (1 つのシリーズ、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と、結合されます。
- ラベルが他の項目のサブセットである場合、例えば、 の項目\$Aにはラベルが付けられ`{host=A, dc=MIA}`、 の項目にはラベル\$Bが付けられ`{host=A}`、結合されます。
- などの変数内に項目ごとに異なるタグキー\$Aがある場合、結合動作は未定義です。

リレーショナル演算子と論理演算子は、true の場合は false 1 に対して 0 を返します。

## 数学関数

ほとんどの関数は独自の式オペレーションに存在しますが、数学オペレーションには数学演算子や記号に似た関数がいくつかあります。関数が数値またはシリーズのいずれかを取ることができる場合、引数と同じ型が返されます。シリーズの場合、シリーズ内の各ポイントの値に対して実行される のオペレーション。

### abs

abs は、引数の絶対値を返します。これは、数値またはシリーズです。たとえば、`abs(-1)`、`abs($A)` などです。

### is\_inf

`is_inf` は数値またはシリーズを受け取り、Inf値 (負または正) 1 の場合は `1` を返し、他の値0の場合は `0` を返します。例えば「`is_inf($A)`」のようになります。

#### Note

例えば、負の無限大を具体的に確認する必要がある場合は、`is_inf($A) == 1` のような比較を行うことができます。`$A == infn()`。

#### `is_nan`

`is_nan` は数値またはシリーズを受け取り、NaN値1の場合は `1` を返し、他の値0の場合は `0` を返します。例えば「`is_nan($A)`」のようになります。NaNと等しくないためNaN、このチェックにはこの関数が必要ですNaN。

#### `is_null`

`is_null` は数値またはシリーズを受け取り、null値1の場合は `1` を返し、他の値0の場合は `0` を返します。例えば「`is_null($A)`」のようになります。

#### `is_number`

`is_number` は数値またはシリーズを受け取り、1すべての実数値と他の値 (null、`Inf+`、`Inf-`) 0に対して `0` を返しますNaN。例えば「`is_number($A)`」のようになります。

#### ログ

ログは、数値またはシリーズである引数の自然対数を返します。値が 0 未満の場合、NaN が返されます。たとえば、`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`

丸めは、丸められた整数値を返します。例えば、`round(3.123)`、`round($A)` などです。

#### `ceil`

Ceil は、数値を最も近い整数値に切り上げます。たとえば、`ceil(3.123)` は `4` を返します。

## floor

下限は、数値を最も近い整数値に切り下げます。例えば、`floor(3.123)` は を返します3。

## 削減

Reduce は、クエリまたは式から返された 1 つ以上の時系列を受け取り、各シリーズを 1 つの数値に変換します。時系列のラベルは、出力される各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 – 再サンプリングする変数 (`refID` ( などA))
- モード – シリーズに数値以外の値 (`null`、`NaN`、`+Inf`) が含まれている場合、リダクション関数の制御動作を許可します。

## 削減関数

### Count (カウント)

カウントは、各シリーズのポイントの数を返します。

### 平均値

平均は、各シリーズのすべての値の合計をそのシリーズのポイント数で割った値を返します。strict モードで、シリーズ内のいずれかの値が `null` または `nan` の場合、またはシリーズが空の場合、`NaN` が返されます。

### 最小と最大

Min と Max は、シリーズ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、シリーズ内のいずれかの値が `null` または `nan` の場合、またはシリーズが空の場合、`NaN` が返されます。

### 合計

Sum は、シリーズ内のすべての値の合計を返します。シリーズの長さがゼロの場合、合計は 0 になります。strict モードで、シリーズに `NaN` 値または `Null` 値がある場合、`NaN` が返されます。

### 最終

Last は、シリーズ内の最後の数値を返します。シリーズに値がない場合、 は `NaN` を返します。

## 削減モード

### 厳密

Strict モードでは、入力シリーズがそのまま処理されます。シリーズ内のいずれかの値が数値以外の場合 (null、NaN、または +-Inf) NaN が返されます。

### 非数値の削除

このモードでは、入力シリーズ内のすべての非数値 (null、NaN、または +-Inf) が除外されてから、リダクション関数が実行されます。

### 数値以外の を置き換える

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。

### 再サンプリング

再サンプリングでは、各時系列のタイムスタンプが一定の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再サンプリングして、それらの間で数学を実行できるようにするためです。これは、2つのシリーズをそれぞれ再サンプリングし、再サンプリングされた変数を参照する数学オペレーションで実行できます。

フィールド :

- 入力 – 再サンプリングする時系列データの変数 (refID ( A など))
- 再サンプリング先 — 再サンプリング先の期間。例: 10s。単位は、秒s、m分、h時間、d日、w週、年y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削減関数。動作の詳細については、「縮小オペレーション」を参照してください。
- Upsample – 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。
  - パッドが最後のナレッジ値で埋められる
  - 次の既知の値でバックフィルする
  - fillna で空のサンプルウィンドウを埋める NaNs

### 式を記述する

データソースがそれらを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. オペレーションフィールドで、書き込む式のタイプを選択します。
4. 式を記述します。
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シリーズまたは数値を返さない場合、式エンジンは `を返します` `NoData`。例えば、リクエストに式によってマージされた 2 つのデータソースクエリが含まれている場合、`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 `になります` `NoData`。Grafana アラートが `NoData` 結果を処理する方法の詳細については、「」を参照してください [Grafana マネージドアラートルールを設定する](#)。

複数のクエリで式を使用する場合、式エンジンでは、すべてのクエリが同じタイムスタンプを返す必要があります。例えば、数学を使用して、それぞれが `を使用する` 複数の SQL クエリの結果を結合する場合 `SELECT NOW() AS "time"`、式はすべてのクエリが同じタイムスタンプ `NOW()` に評価される場合にのみ機能しますが、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、`を` などの任意の時刻 `SELECT 1 AS "time"`、またはその他の有効な UNIX タイムスタンプ `NOW()` に置き換えます。

## クエリ結果を別のパネルと共有する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

Grafana では、ダッシュボード内の他のパネルに対して、1 つのパネルからのクエリ結果を使用できません。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。

この戦略では、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。Grafana 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 Dashboard 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 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すると、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に設計されています。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、「」を参照してください[Grafana バージョン 9 での作業](#)。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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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する強力な方法です。変換を使用すると、次のことができます。

- フィールドの名前を変更する

- 時系列データを結合する
- クエリ間で数学演算を実行する
- ある変換の出力を別の変換への入力として使用する

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

変換されたデータをシステムが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 はリストされている順序で変換を適用します。各変換は結果セットを作成し、処理パイプラインの次の変換に渡します。

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、Reduce 変換を使用して 1 つの列のすべての結果を 1 つの値に要約する場合、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手順を示しています。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。

### 変換をパネルに追加するには

1. 1 つ以上の変換を追加するパネルに移動します。
2. パネルの任意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から **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. 別の変換を適用するには、「**変換を追加**」を選択します。この変換は、前の変換によって返され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。

入力結果セットと出力結果セットは、変換のデバッグに役立ちます。

### 変換を無効にする

変換行の右上にある **表示 (目の)** 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、複数の変換を次々と変更するとき問題をも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する場合は、これらをフィルタリングして、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。

注釈 (例を含む) 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ヒントに反映されます。

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、機能しないことがあります。これは、1つの変換が前の変換の出力を受け取るためです。

### 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ユーザーに通知します。

## 変換を削除するには

1. 編集するパネルを開きます。
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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

データに対して次の変換を実行できます。

### 計算からフィールドを追加する

この変換を使用して、他の 2 つのフィールドから計算され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 - 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、選択した計算を個別に適用します。
  - バイナリオペレーション - 選択した 2 つのフィールドから 1 行の値に基本的なバイナリオペレーション (合計や乗算など) を適用します。
  - 単一オペレーション - 選択したフィールドの 1 行の値に基本的な単一オペレーションを適用します。使用可能なオペレーションは次のとおりです。
    - 絶対値 (abs) - 特定の式の絶対値を返します。0 からの距離を正の数で表します。
    - 自然指数 (exp) — e を特定の式の累乗に上げて返します。
    - 自然対数 (ln) — 特定の式の自然対数を返します。
    - 床 (床) - 特定の式以下の最大の整数を返します。
    - Ceiling (ceil) — 指定された式以上の最小整数を返します。
  - 累積関数 - 現在の行と前のすべての行に関数を適用します。

- 合計 — 現在の行までの累積合計を計算します。
- 平均 — 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、末尾または中央の 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。中央ウィンドウの場合、ウィンドウは現在の行の中央に配置されます。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央に配置されます。
  - 平均 — 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。
- 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 - 行モードを減らすを選択すると、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選択肢のリストが表示されます。使用可能な計算については、「[計算タイプ](#)」を参照してください。
- オペレーション - バイナリオペレーションまたは単一オペレーションモードを選択すると、オペレーション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、選択した2つのフィールドから1行の値に対して基本的な数学演算を実行できます。バイナリオペレーション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、As percentile スイッチが表示されます。この切り替えにより、行インデックスを行の合計数に対するパーセンテージで変換できます。
- 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。これを空白のままにすると、フィールドには計算に一致する名前が付けられます。
- すべてのフィールドを置き換える - (オプション) 他のすべてのフィールドを非表示にし、計算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最新です。Grafana Labs ではサポートが制限されており、この機能が一般公開される前に重大な変更が発生する可能性があります。

## フィールドを連結する

### フィールドを連結する

この変換を使用して、すべてのフレームのすべてのフィールドを1つの結果に結合します。

例えば、温度と稼働時間データ (クエリ A) と空気品質の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、すべての関連情報を含む統合データフレームが1つのビューに表示されます。

次の2つを考慮してください。

クエリ A:

Temp	稼働時間
15.4	1230233

クエリ B:

アザフト語	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。

Temp	稼働時間	アザフト語	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。

クエリ結果からの設定

クエリ結果からの設定

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 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シリーズを選択します。
- オプションに適用する – に適用する の選択内容に応じて、フィールドタイプを指定するか、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は、フィールドマッピングテーブルがあります。この表には、設定クエリによって返されたデータに含まれるすべてのフィールドと、としての使用 および 選択 オプションが一覧表示されます。フィールドと設定プロパティのマッピングを制御し、複数の行に対して選択する値を選択できます。

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

### Config クエリ

時間	値
1626178119127	100
1626178119129	100

出カクエリ (入力と同じですが、現在は値フィールドに設定があります)

時間	値 (設定: Max=100)
1626178119127	10
1626178119129	30

各フィールドに最大設定オプションが設定されました。最小、最大、単位、しきい値などのオプションは、`te` フィールド設定の一部です。設定されている場合、パネルエ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の代わりに、視覚化によって使用されます。

### 値マッピング

クエリ結果を値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つの値マッピング行を定義します。次の例を参照してください。

### Config クエリ結果

値	テキスト	色
L	低	青色
M	中程度	green
H	高い	赤色

フィールドマッピングで、以下を指定します。

フィールド	として 使用する	選択
値	値マッピング/値	すべての値
テキスト	値マッピング/テキスト	すべての値
色	値マッピング/色	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が、

## 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。

- フィールド – 使用可能なフィールドから選択します。
- as – FieldType 変換する を選択します。
  - Numeric – は値の数値を作成しようとします。
  - 文字列 – は値の文字列を作成します。
  - Time – は値を時間として解析しようとします。
    - yyyy-mm-dd や などの文字列で を入力 DateFormat として指定するオプションを表示します DD MM YYYY hh:mm:ss。
  - ブール値 – 値はブール値になります。
  - 列挙型 - 値は列挙型になります。
    - 列挙型を管理するテーブルを表示します。
  - その他 — は値を json として解析しようとします。

例えば、時間フィールド を時刻 として、日付形式 を として選択して変更できる次のクエリを考えてみましょうYYYYY。

時間	マーク	値
2017-07-01	上記の	25
2018-08-02	以下	22
2019-09-02	以下	29
2020-10-04	上記の	22

結果：

時間	マーク	値
2017-01-01 00:00:00	上記の	25

時間	マーク	値
2018-01-01 00:00:00	以下	22
2019-01-01 00:00:00	以下	29
2020-01-01 00:00:00	上記の	22

この変換により、データ型を柔軟に適応させ、視覚化の互換性と一貫性を確保できます。

## 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 – 形式 `a=b` または ソース `c:d` から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る – (オプション) このオプションを選択すると、他のすべてのフィールドが非表示になり、視覚化に計算フィールドのみが表示されます。
- 保持時間 - (オプション) すべてのフィールドの置換が `true` 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。

次のデータセットを検討してください。

## データセットの例

タイムスタンプ	json_data
1636678740000000000	{"value": 1}
1636678680000000000	{"value": 5}
1636678620000000000	{"value": 12}



この設定で[時系列パネル](#)で使用するデータを準備できます。

- ソース: json\_data
- 形式: JSON
  - フィールド: 値
  - エイリアス: my\_value
- すべてのフィールドを置き換える: true
- 保持時間: true

これにより、次の出力が生成されます。

変換されたデータ

タイムスタンプ	my_value
1636678740000000000	1
1636678680000000000	5
1636678620000000000	12

この変換により、さまざまな方法でデータを抽出して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する

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。

- 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。
- 検索 – 国、米国、および空港 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、このデータがある場合：

データセットの例

ロケーション	[値]
AL	0
AK	10
アリゾナ	5
アーカンソー州	1
どこか	5

この設定では、次のようになります。

- フィールド: location
- ルックアップ: 米国

次の出力が表示されます。

変換されたデータ

ロケーション	ID	名前	Lng	遅延	[値]
AL	AL	Alabama	-80.891064	12.448457	0
AK	AK	アーカンソー州	-100.891064	24.448457	10
アリゾナ					5
アーカンソー州					1
どこか					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強化し、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 でデータをフィルタリングする

この変換を使用して、複数のクエリがあるパネルで1つ以上のクエリを非表示にします。

Grafana は、クエリ識別文字をダークグレーテキストで表示します。フィルタリングを切り替えるクエリ識別子を選択します。クエリ文字が白の場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示になります。

#### Note

このデータソースは返されたデータとクエリの関連付けをサポートしていないため、この変換は Graphite では使用できません。

### 値でデータをフィルタリングする

この変換を使用して、視覚化内でデータポイントを直接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たり除外したりするためのオプションが提供されます。

この変換は、データソースが値でネイティブにフィルタリングしない場合に非常に役立ちます。共有クエリを使用している場合は、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。

- 正規表現 - 正規表現式を一致させます。
- Is Null - 値が null の場合に一致。
- Is Not Null - 値が null でない場合は一致します。
- Equal - 値が指定された値と等しい場合に一致します。
- Different - 値が指定された値と異なる場合に一致します。

数値フィールドで使用できるその他の条件は次のとおりです。

- Greater - 値が指定された値より大きい場合に一致します。
- 下限 - 値が指定された値よりも小さい場合は一致します。

- Greater or equal – 値がより大きいか等しい場合は一致します。
- lower or equal – 値が lower または equal の場合に一致。
- 範囲 – 指定された最小値と最大値、最小値と最大値の間の範囲を一致させます。

次のデータセットについて考えてみましょう。

時間	温度	高度
2020-07-07 11:34:23	32	101
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と、設定は次のようになります。

- フィルタータイプ: 'Include'
- 条件: 「温度」が「より低い」「30」と一致する行

また、30°C 未満の温度のみが含まれる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

時間	温度	高度
2020-07-07 10:31:22	20	85
2020-07-07 09:30:57	19	101

フィルターには複数の条件を追加できます。例えば、高度が 100 より大きい場合にのみデータを含めることができます。そのためには、次の設定にその条件を追加します。

- フィルタータイプ: 「すべて一致」条件を持つ「含める」行
- 条件 1: 「温度」が「30」より「低い」と一致する行
- 条件 2: 「高度」が「100 より大きい」と一致する行

複数の条件がある場合、すべての条件に一致する行にアクション (包含/除外) を適用するか、追加した条件のいずれかに一致する行に適用するかを選択できます。

上記の例では、温度が 30°C 未満で高度が 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 を一致を選択します。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 (最初の条件には一致しません) ですが、高度は 101 (2 番目の条件に一致) であるため含まれます。

無効または不完全な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。独自の分析ニーズに合わせてデータプレゼンテーションをカスタマイズする基準をカスタマイズします。

### 名前フィールドを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。フィールド名をフィルタリングするには、次の 3 つの方法があります。

- 正規表現を入力します。
- 含まれてい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。

### 正規表現を使用する

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、正規表現を使用すると、で始まるフィールドのみが返され 'prod.\*' ます。prod

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。

### 含まれているフィールドを手動で選択する

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、チェックされていない場合でも、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する

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で表示できます。

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るようにクエリ結果を柔軟に調整できます。

### 文字列のフォーマット

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でフォーマットします。
- 小文字 – 文字列全体を小文字でフォーマットします。
- センテンスケース – 文字列の最初の文字を大文字でフォーマットします。
- タイトルの大文字と小文字 — 文字列内の各単語の最初の文字を大文字でフォーマットします。
- 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でフォーマットし、単語間のスペースは含まれません。
- キャメルケース – 最初の単語を除く文字列内の各単語の最初の文字を大文字でフォーマットし、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でフォーマットし、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。
- Kebab ケース – 文字列内のすべての文字を小文字でフォーマットし、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。
- Substring – 指定された開始位置と終了位置を使用して、文字列の部分文字列を返します。

この変換により、文字列データの表示を標準化および調整して、視覚化と分析を改善するのに便利です。

#### Note

この変換は現在パブリックプレビュー中です。Grafana Labs ではサポートが制限されており、この機能が一般公開される前に重大な変更が発生する可能性があります。

## フォーマット時間

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 :

タイムスタンプ	イベント
1636678740000000000	システム開始
1636678680000000000	ユーザーログイン
1636678620000000000	データの更新

'YYYY-MM-DD HH:mm:ss' を適用した後 :

タイムスタンプ	イベント
2021-11-12 14:25:40	システム開始
2021-11-12 14:24:40	ユーザーログイン
2021-11-12 14:23:40	更新されたデータ

この変換により、ビジュアライゼーション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実現できます。

**Note**

この変換は、アルファ機能として Grafana v10 と互換性があるワークスペースで使用できません。

## グループ化

この変換は、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。選択すると、計算の選択肢のリストが表示されます。

オリジナルデータの例を次に示します。

時間	Server 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



時間	Server ID	CPU 温度	サーバーステータス
7/7/2020 午前 9 時 23 分 7 秒	サーバー 1	86	OK

この変換は 2 つのステップで行われます。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、ソートした場合と同じフィールドの値がすべて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、データは次のようにグループ化されます。

時間	Server 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

サーバー ID の値が同じ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したら、他のフィールドにさまざまな計算を追加し、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た平均計算を追加して、以下を取得できます。

Server 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:

- Time フィールドでは、Last 値を計算して、サーバーごとに最後のデータポイントがいつ受信されたかを確認できます。
- Server Status フィールドでは、Last 値を計算して、各サーバーの最終状態値を確認できます。
- フィールド温度については、最終値を計算して、各サーバーでモニタリングされる最新の温度を確認することもできます。

その後、以下を取得します。

Server ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最終)	サーバーステータス (最新)
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	オーバードロード
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、便利な方法で表示できます。

## 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 つのフィールドを結合し、行列を生成します。行列は次のように計算されます。

### 元のデータ

Server ID	CPU 温度	サーバスステータス
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

の値を列名 Server Status として使用し、の Server ID 値を行名として使用し、を各セルの内容 CPU Temperature として使用して、行列を生成できます。各セルの内容は、既存の列 (Server Status) と行の組み合わせ () に表示されます Server ID。残りのセルでは、Null、True、False、または空のの間で表示する値を選択できます。

### 出力

サーバー ID	Server 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行列を作成します。行列出力には、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。

### ネストされたテーブルにグループ化する

この変換を使用して、指定したフィールド (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。

次の表は、サンプルデータを示しています。

時間	Server 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すべての同じ値が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すると、Grafana は次のようにデータをグループ化します。

Server ID	[データ]		
サーバー 1	<b>時間</b>	<b>CPU 温度</b>	<b>サーバーステータス</b>
	7/7/2020 午前 11 時 34 分 20 秒	80	シャットダウン
	7/7/2020 午前 9 時 28 分 6 秒	80	OK
	7/7/2020 午前 9 時 23 分 7 秒	86	OK
サーバー 2	<b>時間</b>	<b>CPU 温度</b>	<b>サーバーステータス</b>
	7/7/2020 午前 10 時 32 分 20 秒	90	オーバーロード
	7/7/2020 午前 9 時 30 分 5 秒	88	OK
	7/7/2020 午前 9 時 25 分 5 秒	88	OK

Server 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したら、他のフィールドにさまざまな計算を追加し、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。

Server ID	CPU Temperature (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 時 28 分 6 秒	OK

Server ID	CPU Temperature (平均)		
		時間	サーバーステータス
サーバー 2	88.6	7/7/2020 午前 9 時 23 分 7 秒	OK
		時間	サーバーステータス
		7/7/2020 午前 10 時 32 分 20 秒	オーバーロード
		7/7/2020 午前 9 時 30 分 5 秒	OK
		7/7/2020 午前 9 時 25 分 5 秒	OK

Server ID	CPU Temperature (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

## ヒートマップの作成

この変換を使用して、時間の経過に伴う傾向を視覚化するためのヒストグラムデータを準備します。ヒートマップの視覚化と同様に、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、 の時間範囲は、X 軸に 1 時間幅のセル 1h を作成します。
- Count – non-time-related シリーズの場合、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。

- [Linear] (線形)
- 対数 – ログベース 2 またはログベース 10 のいずれかを選択します。



- Symlog – 対称対数スケールを使用します。負の値を許可して、ログベース 2 またはログベース 10 から選択します。

次のデータセットがあるとします。

タイムスタンプ	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 に設定Size: 15mし、Y バケットを に設定するとLinear、ヒストグラムは X 軸で 15 分の時間間隔に、Y 軸で直線的に値を整理します。
- X Bucket を Logarithmic (base 10)、Y Bucket Count: 2 をとして、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。

## Histogram

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し、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 (xMin から xMax )。
- バケットオフセット – non-zero-based バケットのオフセット。
- シリーズを組み合わせる – 使用可能なすべてのシリーズを使用して、統合されたヒストグラムを作成します。

## 元のデータ

### シリーズ 1:

A	B	C
1	3	5

A	B	C
2	4	6
3	5	7
4	6	8
5	7	9

シリーズ 2:

C
5
6
7
8
9

出力

xMin	xMax	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

xMin	xMax	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、データのスプレッドと密度に関するインサイトを提供します。

### 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し、異なるクエリからのデータを統合できます。

は、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。

### 内部結合

内部結合は、選択したフィールドのすべてのテーブルが同じ値を共有する複数のテーブルのデータをマージします。このタイプの結合は、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 つの結果にまとめ、成功した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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233

時間	ジョブ	稼働時間
7/7/2020 午前 11 時 14 分 20 秒	postgre	345001233

クエリ B:

時間	[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	稼働時間	[サーバー]	エラー
7/7/2020 午前 11 時 34 分 20 秒	node	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

## 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。内部結合は時間フィールドでクエリ A とクエリ B を結合しますが、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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	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	[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	稼働時間	[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		
7/7/2020 午前 11 時 34 分 20 秒	node	25260122	サーバー 1	15

時間	ジョブ	稼働時間	[サーバー]	エラー
7/7/2020 午前 11 時 24 分 20 秒	postgre	123001233	サーバー 2	5

## ラベルによる結合

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 – すべての時系列で使用可能なラベルまたは共通の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

設定 :

value: 'what'

出力:

クラスター	ジョブ	Temp	[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得ることができません。

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値として表示して、データの視覚化を強化します。

2つの時系列のクエリ結果を示します。

- シリーズ 1 – ラベル Server=Server A、Datacenter=EU
- シリーズ 2 – ラベル Server=Server B、Datacenter=EU

列モードでは、結果は次のようになります。

時間	[サーバー]	Datacenter	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

「行」モードでは、結果にはシリーズごとにテーブルがあり、次のような各ラベル値が表示されま  
す。

ラベル	value
[サーバー]	サーバー A
Datacenter	EU

ラベル	value
[サーバー]	サーバー B
Datacenter	EU

### 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、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	Datacenter	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、内部的に 2 つの別々の変換です。最初のは 1 つのシリーズで動作し、フィールドにラベルを抽出します。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、無効にすることはできません。

これを説明するために、ラベルが重複していない時系列を返すクエリが 2 つある例を次に示します。

- シリーズ 1 – ラベル Server=ServerA
- シリーズ 2 – ラベル Datacenter=EU

これにより、最初に次の 2 つのテーブルが作成されます。

時間	[サーバー]	値
7/7/2020 午前 11 時 34 分 20 秒	ServerA	10

時間	Datacenter	値
7/7/2020 午前 11 時 34 分 20 秒	EU	20

マージ後 :

時間	[サーバー]	値	Datacenter
7/7/2020 午前 11 時 34 分 20 秒	ServerA	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、データの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扱う場合に特に便利です。

以下は、制限変換がデータソースからのレスポンスに与える影響を示す例です。

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、「3」の値を持つ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

この変換は、最も関連性の高いデータに焦点を当てるようにデータの視覚的な表示を調整するのに役立ちます。

### シリーズ/テーブルをマージする

この変換を使用して、複数のクエリの結果を 1 つの結果にまとめます。これは、テーブルパネルの視覚化を使用するときに特に便利です。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。

以下は、Merge series/tables 変換がテーブルデータを返す 2 つのクエリに与える影響を示す例です。

#### クエリ A:

時間	ジョブ	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	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	node	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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	15	25260122

時間	ジョブ	エラー	稼働時間
7/7/2020 午前 11 時 24 分 20 秒	postgre	5	123001233

この変換は、クエリ A とクエリ B の値を統合テーブルに結合し、インサイトを向上させるためにプレゼンテーションを強化します。

### 名前でフィールドを整理する

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、1 つの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 は、クエリによって返されたフィールドのリストを表示します。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 — フィールド名の横にある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。
- フィールドの名前変更 — 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があるとします。

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の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調整できるため、Grafana でデータを明確かつインサイトのある形で表現できます。

### 値によるパーティション

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、複数のシリーズ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、次の例のように、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 午後 12 時 00 分	米国	1520
10/20/2022 午後 12 時 00 分	EU	2936
10/20/2022 午前 1 時 00 分 00 秒	米国	1327
10/20/2022 午前 1 時 00 分 00 秒	EU	912

Partition by values transformer 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) の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 午後 12 時 00 分	米国	1520
10/20/2022 午前 1 時 00 分 00 秒	米国	1327

時間	リージョン	値
10/20/2022 午後 12 時 00 分	EU	2936
10/20/2022 午前 1 時 00 分 00 秒	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シリーズを視覚化する柔軟性が向上します。

### 時系列を準備する

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場合の問題に対処します。この変換により、時系列データを幅広形式と長い形式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、視覚化との最適な互換性のために再形成する必要がある形式で、データソースが時系列情報を配信する場合に特に役立ちます。

### 例

この入力：

タイムスタンプ	Value1	Value2
2023-01-01 00:00:00	10	20

タイムスタンプ	Value1	Value2
2023-01-01 01:00:00	15	25

次のように変換できます。

タイムスタンプ	変数	値
2023-01-01 00:00:00	Value1	10
2023-01-01 00:00:00	Value2	20
2023-01-01 01:00:00	Value1	15
2023-01-01 01:00:00	Value2	25

### 幅広い時系列

このオプションを使用して、時系列データフレームを長い形式から広い形式に変換します。これは、データソースが時系列データを長い形式で配信し、視覚化に広い形式が必要な場合に特に役立ちます。

### 例

この入力：

タイムスタンプ	変数	値
2023-01-01 00:00:00	Value1	10
2023-01-01 00:00:00	Value2	20
2023-01-01 01:00:00	Value1	15
2023-01-01 01:00:00	Value2	25

次のように変換できます。

タイムスタンプ	Value1	Value2
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のに特に役立ちます。この変換を適用すると、時間フィールドは削除されます。

入力について考えてみましょう。

クエリ A:

時間	Temp	稼働時間
2020-07-07 11:34:20	12.3	256122
2020-07-07 11:24:20	15.4	1230233

クエリ B:

時間	アザフト語	エラー
2020-07-07 11:34:20	6.5	15
2020-07-07 11:24:20	3.2	5

リデュース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。
- フィールドを減らす – 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、シリーズで最初と最後の計算を使用して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	最初	最後
Temp	12.3	15.4
稼働時間	256122	1230233
アザフト語	6.5	3.2
エラー	15	5

最後の計算で Reduce フィールドは、それぞれ 1 行の 2 つのフレームになります。

クエリ A:

Temp	稼働時間
15.4	1230233

クエリ B:

アザフト語	エラー
3.2	5

## 正規表現で名前を変更する

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 `web-01.example.com` に設定した場合、`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になります。

## 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付けて設定できるため、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、結果のフィールドにより良い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。この表では、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 オプション)。返されたデータに複数の行があるかどうかを選択する値を選択することもできます。

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。代わりに使用するフィールドの「Use as 列」で「Field name」を選択すると、このデフォルト設定を上書きできます。

- 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、代わりに使用するフィールドの「Use as 列」で「Field value」を選択して、このデフォルト設定を上書きできます。

データを視覚化する場合に便利です。

- Gauge
- Stat
- 円グラフ

## 追加のフィールドをラベルにマッピングする

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 は自動的に出力フィールドのラベルのソースとして使用します。

例：

名前	DataCenter	値
ServerA	米国	100
ServerB	EU	200

出力:

ServerA (ラベル : DataCenter: 米国 )	ServerB (ラベル : DataCenter: EU)
100	200

フィールド表示名に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、クエリ結果変換から設定を使用する必要があります。

例

Input:

名前	値	最大
ServerA	10	100
ServerB	20	200
ServerC	30	300

出力:

ServerA (設定: max=100)	ServerB (設定: 最大 = 200)	ServerC (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るのがわかります。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、最大、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。

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。

### シリーズから行へ

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、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。ソースクエリでラベルを定義して、この値をカスタマイズします。

以下の例では、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

シリーズを行変換に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
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、合理化され統合されたデータセットが提供され、効率的な分析と視覚化が表形式で実現されます。

### 並べ替え

この変換を使用して、指定し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、データの理解と分析を容易にします。ソートする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。

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、変換によるソートを適用して、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で配置できます。この機能により、時系列データを簡単にナビゲートして解釈し、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ることができます。

### 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。

- アクション – アクションを選択します。
  - 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。

- 位置モード – 位置モードを選択します (これらのオプションは、計算値モードと変換モードによって共有されます)。
  - 自動 – 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。
  - 座標 – 緯度フィールドと経度フィールドを指定します。
  - Geohash – geohash フィールドを指定します。
  - ルックアップ – Gazetteer の場所フィールドを指定します。
- 値の計算 – ジオメトリを使用して新しいフィールド (見出し/距離/エリア) 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。
    - 見出し – 2 つのポイント間の見出し (方向) を計算します。
    - エリア – ジオメトリで定義されたポリゴンで囲まれたエリアを計算します。
    - 距離 – 2 つのポイント間の距離を計算します。
- 変換 – 空間オペレーションをジオメトリに適用します。
  - オペレーション – ジオメトリに適用するオペレーションを選択します。
    - 行として – 各行に頂点を持つ 1 行の特徴量を作成します。
    - Line Builder – 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。

### 時系列からテーブルへの変換

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。Trend 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。時系列クエリが複数ある場合、それぞれに個別のテーブルデータフレームが作成されます。これらは結合変換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。デフォルトは最終 NULL 以外の値です。この値はスパークラインの横に表示され、テーブル行のソートに使用されます。

### 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、混乱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回帰を使用して、数学関数をデータに適合させることで機能します。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。

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。
- 多項回帰 – 多項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 ではサポートが制限されており、この機能が一般公開される前に重大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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、さまざまな結果が得られます。

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。

## クエリの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 でバグやビジュアライゼーション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、パネルの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のデータの検査](#)」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はいくつですか？ 多数のデータポイントを返すクエリは遅くなります。以下を試してください。

- クエリオプション で、返される最大データポイントを制限します。
- クエリオプション で、最小間隔時間を長くします。
- クエリでは、group by関数を使用します。

## 計算タイプ

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

次の表に、Grafana で実行できる計算のリストを示します。これらの計算は、変換タブとバーゲージ、ゲージ、統計の視覚化で確認できます。

計算	説明
すべての null	すべての値が null の場合、true
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
すべてのゼロ	すべての値が 0 の場合、true
変更数	フィールドの値が変更される回数
カウント	フィールドの値の数
差分	値の累積変更、増分のみをカウント
違い	フィールドの最初と最後の値の違い



計算	説明
差分率	フィールドの最初と最後の値の間の変化率
個別カウント	フィールド内の一意的値の数
最初	フィールドの最初の値
First* (null ではない )	まず、フィールド内の null 値ではありません ( も除外されます NaNs )
最後	フィールドの最後の値
Last* (null ではない )	フィールド内の NULL 値ではない最後の値 ( も除外 NaNs )
最大	フィールドの最大値
平均値	フィールド内のすべての値の平均値
分散	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 (ゼロ以上 )	フィールドの最小値、正の値
[Range] (範囲)	フィールドの最大値と最小値の差
[ステップ]	フィールドの値間の最小間隔
合計	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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、次の方法について説明します。

- 編集用のパネルを開く
- パネル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したら、いつでもパネルを開いて、クエリの変更や更新、データ変換の追加、視覚化設定の変更を行うことができます。

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. パネルの任意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。

編集モードでパネルが開きます。

## 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ヒントに表示されます。

4. パネルと表示するデータの説明を書き込みます。

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ヒントに表示されます。

タイトルフィールド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、およびデータフレーム JSON モデルを調べ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. パネルの任意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、検査 > パネル JSON を選択します。
4. ソースの選択 フィールドで、次のいずれかのオプションを選択します。
  - パネル JSON – パネルを表す JSON オブジェクトを表示します。
  - パネルデータ – パネルに渡されたデータを表す JSON オブジェクトを表示します。
  - DataFrame structure – 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、可変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。パネルの繰り返しの詳細については、「」を参照してください[ダッシュボードを作成する。](#)。

### Note

パネルを繰り返すには、変数で 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

- クエリに複数值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

- 繰り返したいパネルを編集します。
- 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。
- 方向 を選択します。
  - パネル を配置するには、水平を選択します side-by-side。Grafana は繰り返しパネルの幅を調整します。行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。
  - 縦を選択して、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と同じです。
- 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ロードします。

## 標準オプションを設定する

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 9 での作業。](#)

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana で使用されるデータモデルは、時系列とテーブルの両方のクエリ結果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、視覚化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 による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、つまりすべてのシリーズ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、数値を持つすべてのフィールドがパーセンテージで表示されます。

フィールドフォーマット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。パネルの任意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 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、編集してい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、視覚化には影響しません。

#### Note

Grafana Labs は、すべての視覚化のオプションの追加と拡張に常に取り組んでいます。そのため、すべての視覚化ですべてのオプションを利用できるとは限りません。

## 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。

## カスタムユニット

単位ドロップダウンを使用して、カスタム単位、カスタムプレフィックス、またはサフィックスと日時形式を指定することもできます。

カスタム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。

- サフィックス:<suffix> は、値の後に続くカスタム単位です。
- prefix:<prefix> は、値の前にあるカスタム単位です。
- time:<format> カスタム日付時刻形式の場合は、 のように入力しますtime:YYYY-MM-DD。 [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。
- カスタム SI ユニットの si:<base scale><unit characters>。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、これは少し進んでいます。したがって、ソースデータがミリ秒(数千)で表される場合、単位の先頭にその SI スケール文字が付きます。
- カスタムカウント単位の count:<unit>。
- currency:<unit> カスタム通貨単位。

単位ピッカーにネイティブ絵文字を貼り付けて、カスタム単位として選択することもできます。

## 文字列単位

Grafana は、文字列を解析して数値として表示する際に、過度に攻撃的になることがあります。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、文字列単位を含む単位プロパティを追加します。

## スケール単位

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リングします。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。もう 1 つの例は、3000 kW が 3 MW として表示されることです。この動作を無効にする場合は、スケール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。

## 最小/最大フィールド

デフォルトでは、計算された最小値と最大値は、すべてのシリーズ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。

## 10 進数

Grafana がレンダリング値に含める小数点以下数を指定します。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を自動的に切り捨てます。例えば、1.1234 は 1.12 と表示され、100.456 は 100 と表示されます。

すべての小数を表示するには、単位を文字列 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で [変数](#) を使用できます。

複数の統計、フィールド、またはシリーズが表示される場合、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、タイトルにシリーズ名またはフィールド名のみを使用できます。

という名前のフィールドと Temp というラベルが与えられます `{"Loc"="PBI", "Sensor"="3"}`。

式の構文	例	へのレンダリング	説明
<code>\${__field.displayName}</code>	構文と同じ	Temp {Loc="PBI", Sensor="3"}	フィールド名が表示され、ラベル{}がある場合は に表示されます。レスポンスにラベルキーが 1 つしかない場合、ラベル部分の場合、Grafana は中括弧なしでラベルの値を表示します。
<code>\${__field.name}</code>	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)。
<code>\${__field.labels}</code>	構文と同じ	Loc="PBI", Sensor="3"	名前のないラベルを表示します。
<code>\${__field.labels.X}</code>	<code>\${__field.labels.Loc}</code>	PBI	指定されたラベルキーの値を表示します。
<code>\${__field.labels._values}</code>	Syntax と同じ	PBI, 3	ラベルの値をカンマで区切って表示します (ラベルキーなし)。

特定のフィールドの式をレンダリングした後に値が空の文字列である場合は、デフォルトの表示方法が使用されます。



## カラースキーム

色オプションとその視覚化への影響は、作業している視覚化によって異なります。一部の視覚化では、色オプションが異なります。

単一の色を指定するか、値に基づいて連続 (グラデーション) の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
単色	オーバーライドルールで便利な 1 つの色を指定する
色のシェード	オーバーライドルールで便利な単一色のシェードを選択します。
しきい値から	一致するしきい値から色を取得するように Grafana に通知します
クラシックパレット	Grafana は、パレット内の色をシリーズインデックスで検索して色を割り当てます。グラフ、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。
クラシックパレット (シリーズ名別)	Grafana はシリーズの名前に基づいて色を割り当てます。対比するシリーズ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 (値別)	連続したカラースキーム
赤-黄-緑 (値別)	連続したカラースキーム
Blue-Yellow-Red (値別)	連続したカラースキーム
黄赤 (値別)	連続したカラースキーム
Blue-Purple (値別)	連続したカラースキーム
黄-青 (値別)	連続したカラースキーム

カラーモード	説明
Blues (値別)	連続したカラースキーム (パネルの背景から青色)
赤 (値別)	連続色スキーム (パネルの背景色から赤)
緑 (値別)	連続色スキーム (パネルの背景色から緑色)
紫 (値別)	連続色スキーム (パネルの背景色から紫色)

## 値なし

フィールド値が空または null の場合に表示される Grafana 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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

パネルには、視覚化に表示されるデータを解釈するため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瞭さを追加します。

凡例は、次の視覚化でサポートされています。

- [棒グラフ](#)
- [キャンドルスティック](#)
- [Histogram](#)
- [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](#)には凡例もありますが、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。スイッチを使用して、凡例のオンとオフを切り替えます。

## Mode (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位置など、凡例に関するその他の情報も表示される場合があります。

## 配置

視覚化上の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)

## [Width] (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。Grafana が凡例の幅を自動的に設定できるようにするには、フィールドを空のままにするか、フィールドに値を入力します。

## [値]

凡例にシリーズ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、凡例をスクロールしてすべての値を表示できます。

## シリーズの色の変更

デフォルトでは、Grafana はシリーズデータの色を設定しますが、パネルの凡例を使用して変更できません。

シリーズ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、シリーズに関連付けられているカラーバーを選択します。
3. カラータブで事前設定された色を選択するか、ピッカーまたは RGB 値を使用して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。

## 視覚化でのシリーズデータの分離

視覚化は視覚的に複雑で、多くのデータシリーズが含まれることがよくあります。凡例を使用して視覚化からシリーズ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、表示するデータが分離されます。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。

視覚化でシリーズ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、分離するシリーズのラベルを選択します。

システムは、他のすべてのシリーズデータの表示から を削除します。

3. 分離されたシリーズにシリーズデータを増分的に追加するには、Ctrl キーまたは コマンドキーを押し、追加するシリーズ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ビューに戻すには、任意のシリーズラベルを 2 回クリックします。

## ソートシリーズ

凡例をテーブルとしてフォーマットし、そのテーブルに値を追加すると、テーブル内のシリーズをそれらの値でソートできます。

シリーズをソートするには

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. メニューから、**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. **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。

## データリンクを設定する

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

データリンクを使用すると、リンクに詳細なコンテキストを提供できます。シリーズ名、またはカーソルの下に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。時系列の視覚化では、例えば、データポイントや行を選択します。統計、ゲージ、バーゲージなどの広いエリアの視覚化では、視覚化上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。

ビジュアライゼーションにデータリンクが 1 つしかない場合は、ビジュアライゼーション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ビジュアライゼーション

- 棒グラフ
- バーゲージ
- ローソク足チャート
- キャンバス
- Gauge
- ジオマップ
- ヒートマップ
- Histogram
- 円グラフ
- Stat
- 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、シリーズ、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 \$ フィールドに入力します。

データリンク URLs 「」を参照してください [変数](#)。

### 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 (例: `?from=now-6h&to=now`)
- `$__from` – 詳細については、「[グローバル変数](#)」を参照してください。
- `$__to` – 詳細については、「[グローバル変数](#)」を参照してください。

### シリーズ変数

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

### 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

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セット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]` - フィールドの値 `i` (同じ行)。
- `__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します。

## 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、リンクをクリックするすべてのユーザー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 つの値を選択しました	<code>var-myvar=value1</code>
複数の値を選択しました	<code>var-myvar=value1&amp;var-myvar=value2</code>
選択済み All	<code>var-myvar=All</code>

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** を選択してパネルエディタを開きます。
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して展開します。
5. **リンクを追加** を選択します。



6. 開いているダイアログボックスに、タイトルを入力します。これは、UI に表示される、人間が読めるリンクのラベルです。
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、Ctrl+Space \$または Cmd+Space と入力または押すと、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. 変更をダッシュボードに保存します。

## 値マッピングを設定する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

フィールドオーバーライドに加えて、値マッピングは、視覚化でデータがどのように表示されるか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](#)で設定された単位フォーマットをバイパスします。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」セクションに値マッピングの概要を表示します。

### サポートされているビジュアルライゼーション

- 棒グラフ
- バーゲージ
- ローソク足チャート
- キャンバス
- Gauge
- ジオマップ

- Histogram
- 円グラフ
- Stat
- 状態のタイムライン
- ステータス履歴
- テーブル
- 時系列
- トレンド

## 値マッピングのタイプ

Grafana は、次の値マッピングタイプをサポートしています。

- 値 – 特定の値をテキストと色にマッピングします。例えば、値のすべてのインスタンスが数値Perfection!ではなくとして10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、値マッピングを使用します。
- 範囲 – 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、数値Highではなく Lowまたは を表示するように範囲値マッピングを設定できます。複数の連続値をフォートする場合は、範囲マッピングを使用します。
- 正規表現 – 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 の場合 `www.example.com`、Grafana がドメインを表示 `www` および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、正規表現マッピングを使用します。
- Special –、NaN (数値ではない) などの特別な値とNull、 true や などのブール値をテキストと色falseにマッピングします。例えば、値が として表示されるように特別なnull値マッピングを設定できますN/A。異常値、ブール値、または空の値をフォーマットする場合は、Special mappiong 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. メニューから、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. 新しいマッピングを追加 を選択し、次のいずれかを選択します。
  - 値 – 一致する値を 1 つ入力します。
  - 範囲 – 一致する範囲の開始値と終了値を入力します。
  - 正規表現 – 一致する正規表現パターンを入力します。
  - Special – 一致する特別な値を選択します。
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。

マッピングを追加すると、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードでは、しきい値はメトリクスに設定した値または制限であり、メトリクスが満たされたり超えたりすると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ビジュアルライゼーション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。

しきい値は、次の目的で使用できます。

- カラーグリッド線またはグリッドは、時系列の視覚化における領域です。
- 統計の視覚化で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リージョンとリージョンの色を定義します。
- 時系列の視覚化における色線。
- ゲージとしきい値マーカーをゲージで色付けします。
- ジオマップの色マーカー。
- テーブル内の色セルテキストまたは背景。

#### サポートされているビジュアライゼーション

- 棒グラフ
- バーゲージ
- ローソク足チャート
- キャンバス
- Gauge
- ジオマップ
- Histogram
- Stat
- 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 = 赤
- ベース = グリーン
- モード = 絶対
- 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

次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

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 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。これは通常、「良い」色です。

### しきい値モード

2つのしきい値モードがあります。

- 絶対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です。
- パーセントしきい値は、最小値または最大値を基準にして定義されます。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、キャンドルスティック、時系列、トレンドの視覚化でのみサポートされます。

以下のオプションを使用して、視覚化にしきい値を表示するかどうかと方法を設定します。

- Off - しきい値は表示されません。
- 線 - しきい値は視覚化に水平線として表示されます。
- 線 (破線) - しきい値は、破線の水平線として表示されます。
- 塗りつぶされた領域 - しきい値は水平領域として表示されます。
- 塗りつぶされた領域と線 - しきい値は、線で区切られた水平方向の領域として表示されます。
- 塗りつぶされた領域と線 (破線) - しきい値は、破線で区切られた水平方向の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 はしきい値を最高から最低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Thresholds検索バーに を入力します。
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矢印と下矢印を使用して、値を段階的に増減します。
7. しきい値の左側にある色付きの円をクリックすると、色ピッカーが開き、しきい値の色を更新できます。
8. 「しきい値モード」で、**絶対** または **パーセント** 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。

## フィールドオーバーライドを設定する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

オーバーライドを使用すると、特定のフィールドまたはシリーズの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、それぞれが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、正規表現一致の名前を持つフィールドを使用してオーバーライドを追加bytesし、オーバーライドルールに Unit オプションを追加して、テキストを含むすべてのフィールドに単位を設定します。

#### 例 1: 温度のフォーマット

結果セットは、time との 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しようtemperature。

time	temperature
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。

time	temperature
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。

time	temperature
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: 温度と湿度のフォーマット

結果セットが、time、high temp low temp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しょうhumidity。

time	高温度	低温度	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

摂氏の単位を追加して、小数点以下を削除しましょう。これにより、次の表が表示されます。

time	高温度	低温度	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好に見えますが、湿度を変更する必要があります。このフィールドオプションオーバーライドを湿度フィールドに適用し、単位を Misc > パーセント (0~100) に変更することで、これを修正できます。

time	高温度	低温度	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シリーズ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を追加するには

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。
  - 名前のフィールド – 使用可能なすべてのフィールドのリストからフィールドを選択します。このセレクタを使用してルールに追加するプロパティは、この単一のフィールドにのみ適用されます。
  - 正規表現に一致する名前のフィールド – 正規表現で上書きするフィールドを指定します。このセレクタを使用して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、上書き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。
  - タイプのフィールド – 文字列、数値などのフィールドをタイプ別に選択します。このセレクタを使用して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。このセレクタを使用して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を追加 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、フィールド内のホワイトテキストを削除します。
7. 上書きプロパティを追加 を選択して、このフィールドに上書きを追加し続けるか、上書きを追加 を選択し、上書き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、上書き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、関連す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表示

パネル表示オプションでフィールドオーバーライドを表示できます。

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、上書きが表示されるまで下にスクロールします。

### Note

All タブに表示されるオーバーライド設定は、Overrides 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。行った変更はすぐに有効になります。

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、上書き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を追加](#) 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana には、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が用意されています。ドキュメントのこのセクションでは、組み込みのビジュアライゼーション、それらのオプション、一般的な使用方法について説明します。

使用を開始し、パネルの使用の基本を学ぶための一般的なパネルは、[時系列](#)パネルです。

### **i** Note

どのビジュアライゼーション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ビジュアライゼーションの提案を提供できます。視覚化を選択すると、Grafana はその視覚化が適用されたプレビューを表示します。

- [グラフとグラフ](#)
  - [時系列](#)は、デフォルトおよびメインのグラフ視覚化です。
  - 時間の経過に伴う状態変化の状態[タイムライン](#)。
  - 経時的な定期的な状態の[ステータス履歴](#)。
  - [棒グラフ](#)には、カテゴリ別データがすべて表示されます。
  - [ヒストグラム](#)は、棒グラフの値分布を計算して表示します。
  - [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、通常は現象の大きさに使用されます。
  - [円グラフ](#)は通常、比例性が重要な場合に使用されます。
  - 通常、[キャンドルスティック](#)は、価格/データの移動が焦点となる財務データ用です。

- [ゲージ](#)は、1つ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められたビジュアルです。
- 時間ではない連続した数値 x を持つデータセットの[傾向](#)。
- [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ます。
- 統計と数値
  - 大きな[統計](#)の統計とオプションのスパークライン。
  - [バーゲージ](#)は、水平または垂直のバーゲージです。
- [Misc] (その他)
  - [テーブル](#)は、メインテーブルと唯一のテーブルの視覚化です。
  - [ログ](#)は、ログの主な視覚化です。
  - 有向[グラフ](#)または[ネットワークのノード](#)グラフ。
  - [トレース](#)は、トレースの主な視覚化です。
  - [フレームグラフ](#)は、プロファイリングの主な視覚化です。
  - [Geomap](#) は、地理空間データを視覚化するのに役立ちます。
  - [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。

詳細はこちら

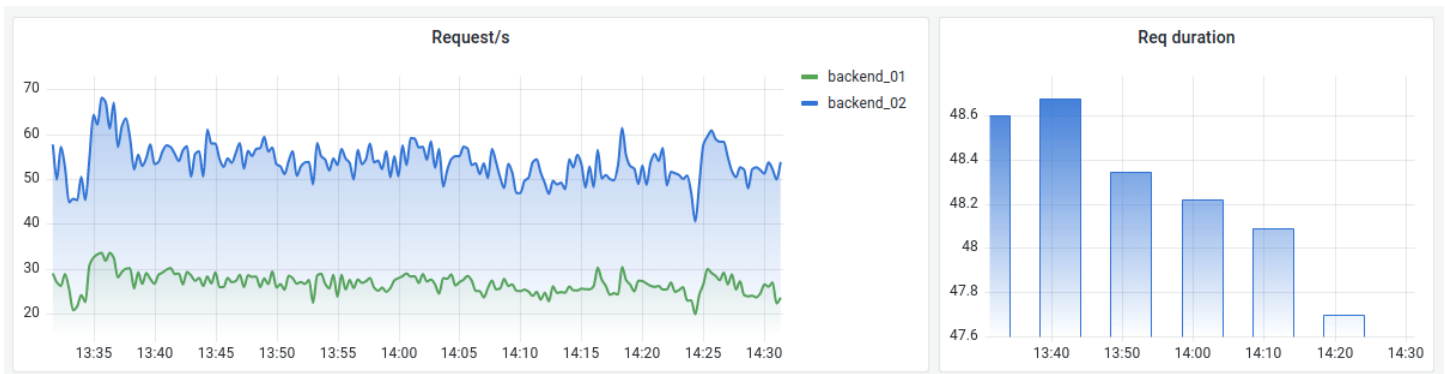
からパネルプラグインをインストールすることで、視覚化タイプを追加できます[プラグインカタログでプラグインを検索する](#)。

例

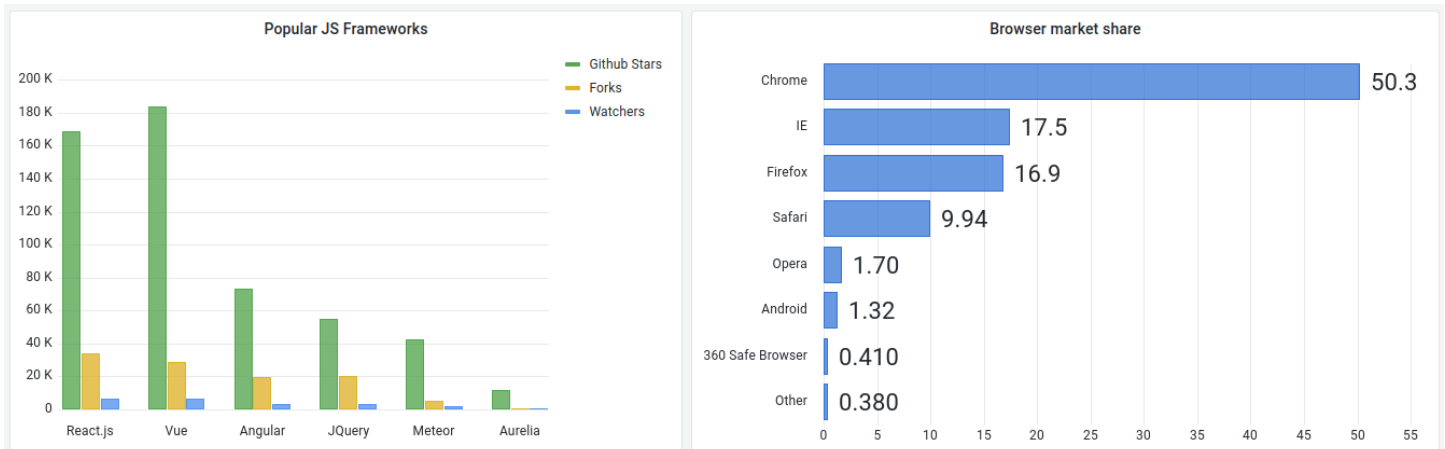
以下のセクションでは、視覚化の例を示します。

グラフ

時間ベースの折れ線、面、棒グラフの場合は、デフォルトの[時系列](#)視覚化をお勧めします。

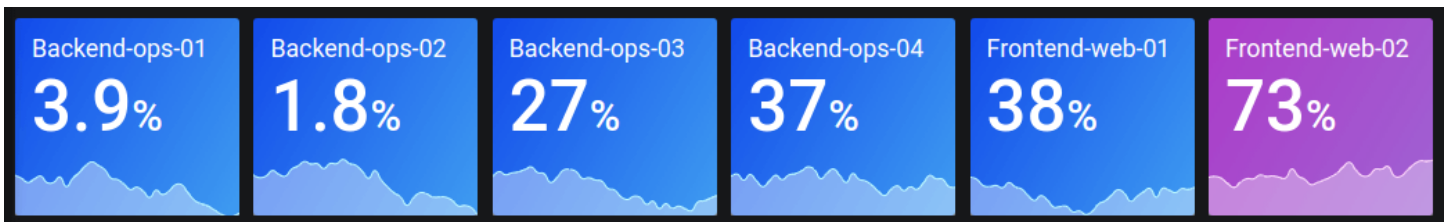


カテゴリ別データの場合は、[棒グラフ](#)を使用します。



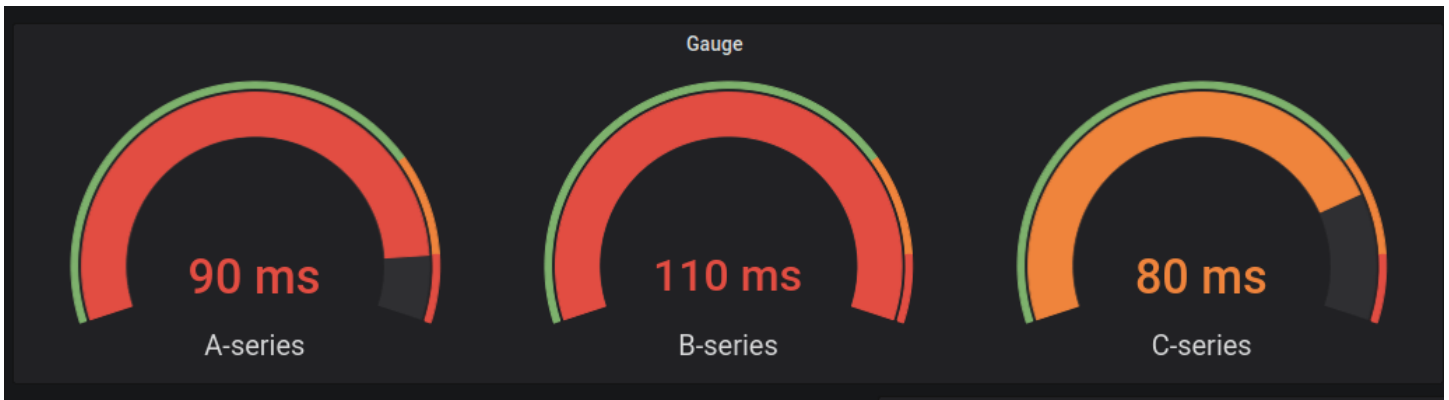
## 大きな数字と統計

[統計](#)の視覚化では、1つの大きな統計値とオプションのグラフスパークラインが表示されます。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。

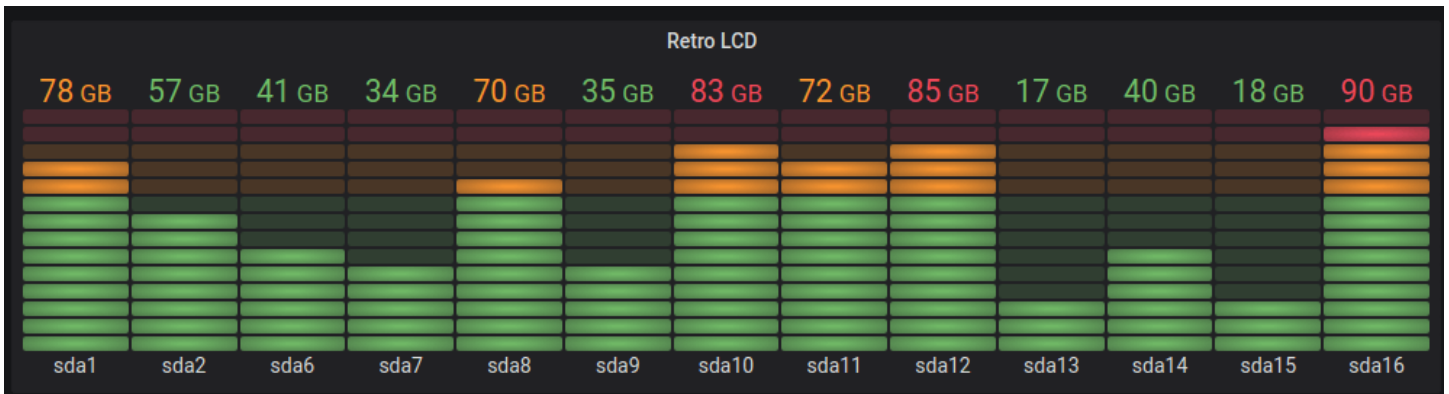


## Gauge

最小値と最大値に関連する値を表示する場合は、2つのオプションがあります。まず、標準の放射ゲージ：



次に、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**バーゲージ**もあります。



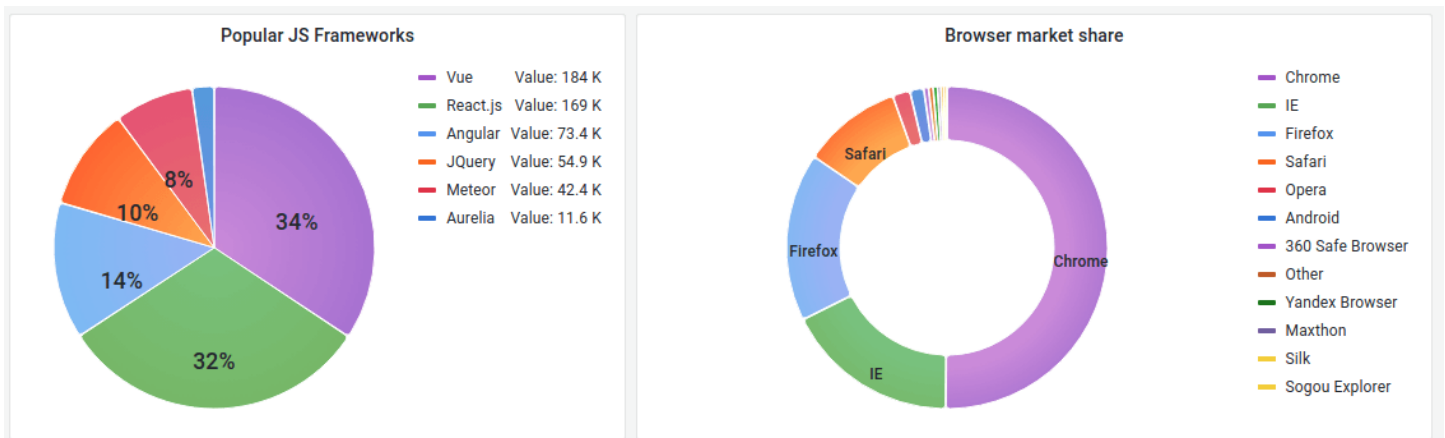
## テーブル

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。

Bar gauge cell display mode					
Time	Info	Min	Max ↑	Value	
2020-09-15 12:45:11	down	73.6 °	76.5 °	<div style="width: 74.0%;"></div>	74.0 °
2020-09-15 12:39:56	up	73.1 °	76.5 °	<div style="width: 75.1%;"></div>	75.1 °
2020-09-15 12:27:41	down	72.9 °	76.5 °	<div style="width: 74.2%;"></div>	74.2 °
2020-09-15 12:40:11	up	73.2 °	76.6 °	<div style="width: 75.2%;"></div>	75.2 °
2020-09-15 12:27:26	up	73.9 °	76.6 °	<div style="width: 74.2%;"></div>	74.2 °
2020-09-15 12:44:56	up	72.9 °	76.6 °	<div style="width: 74.2%;"></div>	74.2 °
2020-09-15 12:39:26	up	72.7 °	76.6 °	<div style="width: 74.7%;"></div>	74.7 °
2020-09-15 12:42:41	down	73.1 °	76.7 °	<div style="width: 74.4%;"></div>	74.4 °
2020-09-15 12:51:41	down	73.0 °	76.7 °	<div style="width: 75.4%;"></div>	75.4 °
2020-09-15 12:41:56	down fast	74.5 °	76.7 °	<div style="width: 74.8%;"></div>	74.8 °

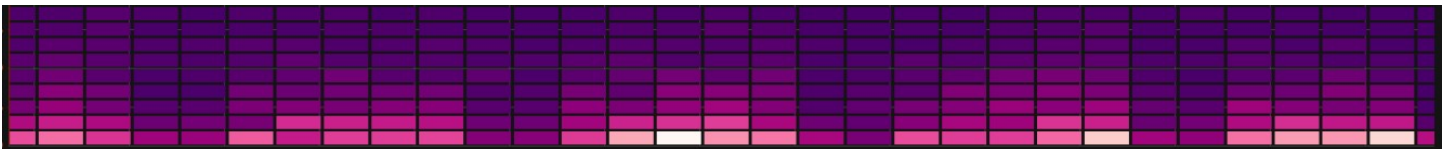
## 円グラフ

1 つ以上のクエリからの縮小されたシリーズまたは値を相互に関連して表示するには、[円グラフ](#)の視覚化を使用します。



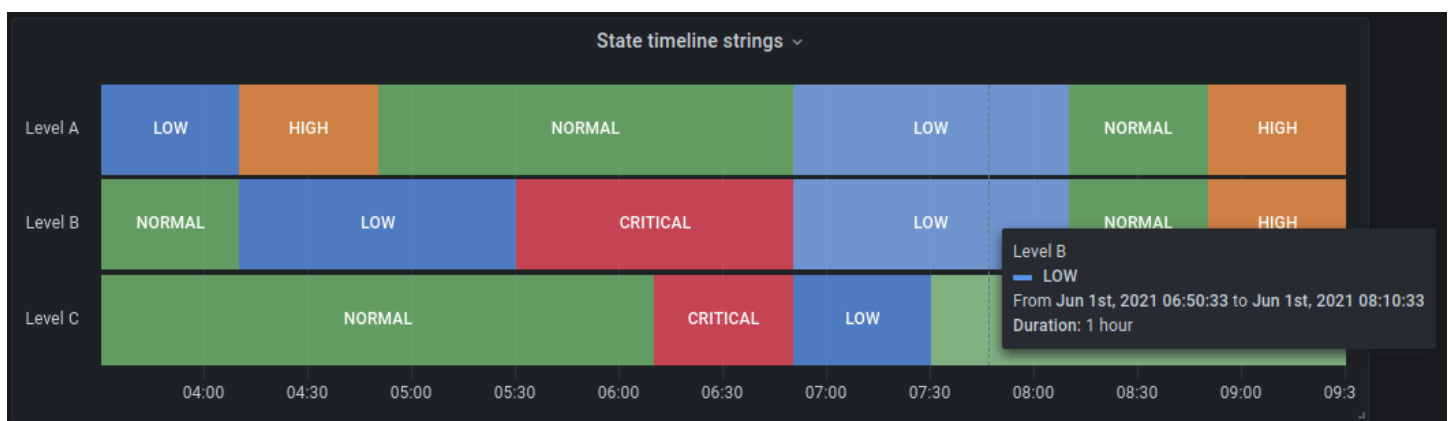
## ヒートマップ

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の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領域に変換します。



## アラートリスト

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「[アラート](#)」を参照してください。[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、「」を参照してください[Grafana バージョン 10 のアラート](#)。

これらの設定を使用して、視覚化を絞り込みます。

## オプション

- グループモード – 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するデフォルトのグループ化と、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するカスタムグループ化のどちらかを選択します。
- 最大アイテム数 — リストするアラートの最大数を設定します。
- ソート順 — 表示されるアラートの順序付け方法を選択します。
  - アルファベット順 (asc) – アルファベット順
  - アルファベット順 (desc) — アルファベット順の逆順
  - 重要度 — 次の値に従って重要度が異なり、1 が最も高くなります。
    - alerting または firing: 1
    - no\_data: 2
    - pending: 3
    - ok: 4
    - paused または inactive: 5
  - 時間 (asc) — 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。
  - Time (desc) – 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 – アラートリストがあるダッシュボードからのアラートのみを表示します。

## 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で制限できます。

- 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-.\*"} です。
- 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状態を選択します。

- アラート/ファイリング
- 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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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リストが表示されます。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するには、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、注釈のリストの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配置されてい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。以下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。以下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使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

limit 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、注釈パネルの表示に含まれる追加のメタデータを制御します。

### ユーザーを表示する

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。

### タグを表示する

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ライブフィルタリングできます。

### リンク動作

#### リンクターゲット

このオプションを使用して、注釈付きデータの表示方法を選択します。以下のオプションがあります。

- パネル - このオプションは、対応する注釈を含むパネルの全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、ダッシュボード全体のコンテキストで注釈に焦点を当てます。

### より前の時間

このオプションを使用して、注釈の前の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。

### 以降の時間

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。

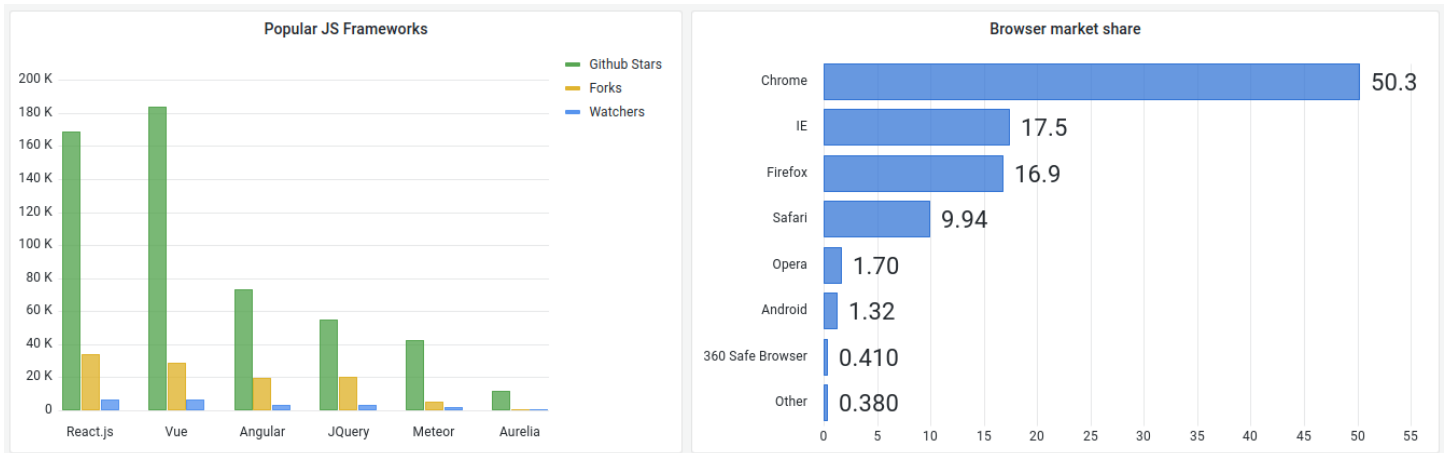
### 棒グラフ

**⚠** 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

棒グラフを使用すると、カテゴリ別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

サポートされているデータフレームは 1 つだけで、X 軸または Y 軸のカテゴリ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。データ形式の例を次に示します。

ブラウザ	マーケットシェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合、パネルにはグループ化された棒が表示されます。

## 時系列または複数の結果セットの視覚化

時系列またはテーブルが複数ある場合は、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、Reduce 変換を追加し、計算で Max と Last as オプションを指定します。

## 棒グラフオプション

これらのオプションを使用して、視覚化を絞り込みます。

### [Orientation] (向き)

- 自動 – Grafana は、パネルの寸法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。

- Vertical – Y 軸をカテゴリ軸にします。

### X 軸ティックラベルをローテーションする

グラフが垂直方向の場合、この設定ではバーの下のラベルがローテーションされます。この設定は、棒グラフのラベルが長く重複している場合に役立ちます。

### X 軸ティックラベルの最大長

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

棒グラフ間の最小間隔を設定します。

### 値を表示する

バーの左上に値を表示す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。
- Always – 常に値を表示します。
- Never – 値を表示しないでください。

### スタッキング

棒グラフの積み上げを制御します。

- オフ – バーはスタックされません。
- 正常 – 棒は互いに積み重なります。
- パーセント – 棒は互いに積み重なり、各棒の高さはスタックの合計の高さに対するパーセンテージです。

### グループ幅

グループの幅を制御します。

- 0 = Minimum width
- 1 = Maximum width

## 棒の幅

棒の幅を制御します。

- 0 = Minimum width
- 1 = Maximum width

## 棒の半径

バーの半径を制御します。

- 0 = Minimum radius
- 0.5 = Maximum radius

## ホバー時に全エリアを強調表示する

バーにカーソルを合わせると、バーの周囲エリア全体が強調表示されるかどうかを制御します。

## 線幅

棒の線幅を制御します。

## 不透明度を埋める

バー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

勾配フィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – 勾配フィルなし。これはデフォルトの設定です。
- 不透明度 – 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。
- Hue – グラデーション色は、線の色相に基づいて生成されます。

## ツールヒントモード

視覚化の上にカーソルを置くと、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、視覚化にカーソルを合わせているシリーズが 1 つだけ表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – 視覚化を操作するときにツールヒントを表示しないでください。

#### Note

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にできます。

## テキストサイズ

値を入力して、棒グラフのテキストのサイズを変更します。

## 凡例オプション

### 凡例モード

これらの設定を使用して、視覚化に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。

- List – 凡例をリストとして表示します。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 – グラフの右側。

### 凡例値

凡例に表示するシリーズデータ値または標準計算を選択します。複数のを持つことができます。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、編集してい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、視覚化には影響しません。

## 配置

Y 軸の配置を設定します。

- 自動 – Grafana は Y 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に、右軸を次の単位に割り当てます。
- 左 – 左側にすべての Y 軸を表示します。
- 右 – 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、オーバーライドを使用して異なるラベルを割り当てることができます。

## [Width] (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、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、軸が視覚的に互い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量を比較しやすくなります。

## 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マックス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マックスの設定により、データがほぼフラットなときにリップが山に変わるのを防ぐことができます。また、標準最小値と最大フィールドオプションから導出されるハードマムまたは最大値により、定義されたポイントを超えてスパイクをクリップすることで、断続的なスパイクが有用な詳細をフラット化することを防ぐことができます。

標準の最小/最大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。

## 複数の Y 軸を表示する

場合によっては、複数の y 軸を表示することもできます。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 2 つのシリーズで異なる単位を持つ 2 つの y 軸を表示するとします。

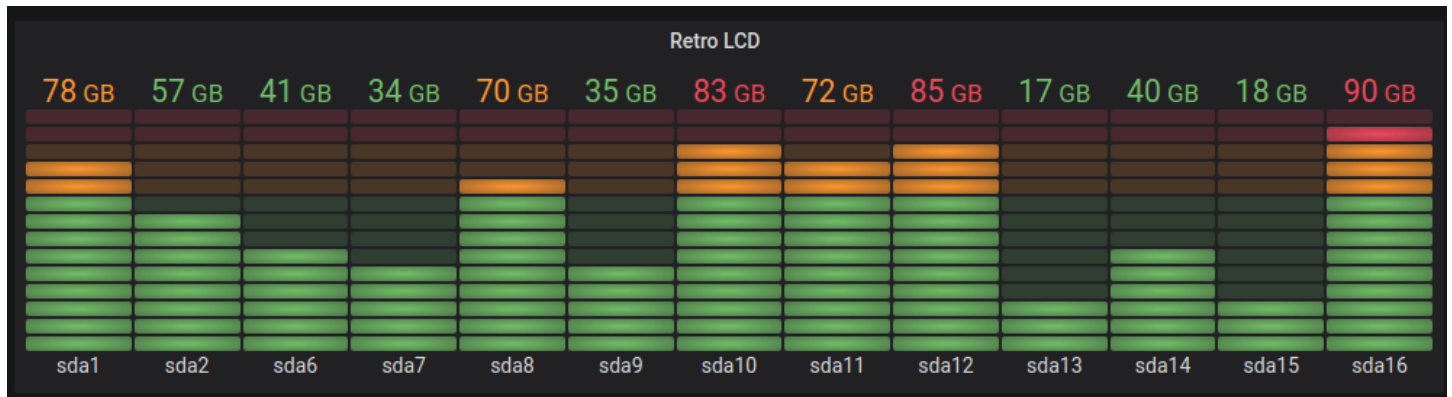
これを行うには、[フィールドオーバーライドを追加します](#)。必要な回数だけステップに従って、必要な数の y 軸を追加します。

## バーゲージ

- ⚠ このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

バーゲージは、すべてのフィールドを 1 つの値に減らすことで、データを簡素化します。Grafana による削減の計算方法を選択します。

このパネルには、クエリが返すシリーズ、行、または列の数に応じて、1 つ以上の棒グラフを表示できます。



## 値オプション

以下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

すべての行に基づいて計算された値を表示します。

- 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リデューサー関数を選択します。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。

## バーゲージオプション

バーゲージの表示方法を調整します。

### [Orientation] (向き)

スタッキング方向を選択します。

- 自動 – Grafana は、最適な向き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縮します。
- Vertical – バーは垂直方向に、下から上まで伸縮します。

## 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。
- リトロ LCD – ゲージは、ライトまたはライトのないスモールセルに分割されます。
- 基本 – 一致するしきい値に基づく単一色。

## 値の表示

値表示モードを選択します。

- 値の色 – 値の色は値によって決まります。
- テキスト色 – 値色はデフォルトのテキスト色です。
- Hidden – 値は非表示になります。

## 名前の配置

名前配置モードを選択します。

### Note

このオプションは、バーゲージの向きが水平の場合にのみ適用されます。バーゲージが垂直方向の場合、名前は常に各バーゲージの下部に配置されます。

- 自動 – Grafana が最適な配置を決定します。
- 上 – 名前は各バーゲージの上部に配置されます。
- 左 – 名前は各バーゲージの左側に配置されます。

## 未入力のエリアを表示する

バーの未入力領域を濃い灰色でレンダリングする場合は、このオプションを選択します。Retro LCD デ스플레이モードには適用されません。

## バーサイズ

バーサイズモードを選択します。

- 自動 – Grafana が最適なバーゲージサイズを決定します。
- 手動 – バーゲージのサイズを手動で設定します。

## 最小幅

ゲージが垂直方向に向いている場合は、棒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。

## 最小の高さ

ゲージを水平に向ける場合は、棒の行の最小の高さを制限します。

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

ゲージを水平に向ける場合は、棒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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

キャンドルスティック視覚化を使用すると、価格の動きに重点を置い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。キャンドルスティックパネルには、オープン/ハイ/ロークローズ (OHLC) 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。



キャンドルスティックビジュアライゼーションは、の基礎に基づいて構築[時系列](#)され、多くの一般的な構成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、視覚化に使用するディメンションを切り替えられます。

- キャンドル – パネルの寸法を、キャンドルスティックの視覚化で使用するオープン、ハイ、ロー、クローズの寸法に制限します。
- ボリューム – パネルのディメンションをボリュームのディメンションに制限します。
- 両方 — キャンドルスティックパネルのデフォルトの動作。これには、キャンドルスティックとボリュームの両方の視覚化が含まれます。

## キャンドルスタイル

- キャンドル – デフォルトの表示スタイルで、 はオープンディメンションとクローズディメンションの間にキャンドルスタイルの視覚化を作成します。
- OHLC バー – 4 つのコアディメンションのオープン、ハイ、ロー、クローズ値を表示します。

## カラー戦略

- Since Open – デフォルトの動作。このモードは、期間内価格の変動が正の場合、Up 色 (下) 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 色が使用されます。
- 以前のクローズ以降 — キャンドルの色が期間間の価格の変動または値の変化に基づく代替の表示方法。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 色が使用されます。オープン時の値がクローズ時の以前の値よりも小さい場合、ダウン色が使用されます。このオプションでは、燭台視覚化モードもトリガーされます。低めのキャンドルスティックは、ピリオド内の動きが正 (値はオープンに近いほど高い) であることを示します。塗り分けされたキャンドルスティックは、ピリオド内の変化が負 (値はオープンに近いほど低い) であることを示します。詳細については、[「のの違いの説明」](#)を参照してください。

## アップカラーとダウンカラー

アップカラーとダウンカラーオプションは、価格の変動がアップまたはダウンするとき使用する色を選択します。上記の色戦略は、期間内または期間間の料金移動を使用して、キャンドルまたは OHLC バーの色を選択するかどうかを決定します。

## オープン、高、低、クローズ

キャンドルスティックパネルは、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。

- Open は、指定された期間の開始値に対応します。
- High は、指定された期間の最大値に対応します。

- Low は、指定された期間の最小値に対応します。
- クローズは、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、指定された期間のサンプル数に対応します。(取引数など)

#### Note

ろうそくの凡例は、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「キャンドルスティック」オプションの下にある「オープン」、「高」、「低」、「クローズ」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

ろうそくパネルは時系列の視覚化に基づいています。オープン、ハイ、ロー、クローズ、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、で使用可能な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[時系列](#)。

#### キャンバス

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Canvases は、Grafana の能力とカスタム要素の柔軟性を組み合わせます。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、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を設計し、データをオーバーレイできます。これらはすべて Grafana の UI 内で実行できます。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、Canvas パネルの設計は非常に慣れていているように感じることができます。

### [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的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。

### [Text] (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。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のいずれかによってトリガーされる編集モードもサポートしています。

#### 楕円

楕円要素を使用すると、キャンバスに基本的な楕円を追加できます。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、背景色はデータのしきい値に基づいて変更できます。

#### 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、背景色はデータの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。

## [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、電球の点滅速度をサポートし、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、キャンバスに基本的なボタンを追加できます。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。

## 接続

キャンバスを構築するときに、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。要素にカーソルを合わせると、接続アンカーが表示され、インライン編集がオンになります。接続を削除するには、接続を選択し、Delete または を押し、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。そのためには、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。

### 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、一般提供前に重大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、背景画像を設定し、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、要素のレイヤーの配置を変更したりできます。

## Canvas 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され、意図しない変更が防止されます。

### データリンク

Canvases は [データリンク](#) をサポートします。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のデータリンクを表示できます。

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. 名前が の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を追加ボタンを選択します。
5. リストからデータリンク > データリンクを選択します。
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、データリンクツールヒントが表示されます。
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、データリンクを表示する要素を制御する場合は、[計算からフィールドを追加する](#)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、要素で使用できるフィールドとして表示されます。

## ダッシュボードリスト

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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するように設定できます。

ダッシュボードがロードされるたびに、このパネルはダッシュボードリストをクエリし、常に最も多く up-to-date の結果を提供します。

## オプション

これらのオプションを使用して、視覚化を絞り込みます。

- 現在の時間範囲を含める – 現在のダッシュボードの時間範囲をダッシュボードリンクに伝達するには、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、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がリンクに含まれます。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 – スタード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、変数補間がサポートされています。例えば、`$my_var`または `です${my_var}`。
- 見出しの表示 – 選択したリスト選択 (Starred、Recently viewed、Search) 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、Starred and Recently viewed 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最大 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、部分的な値が受け入れられます。
- 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する場所を次に示します。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は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

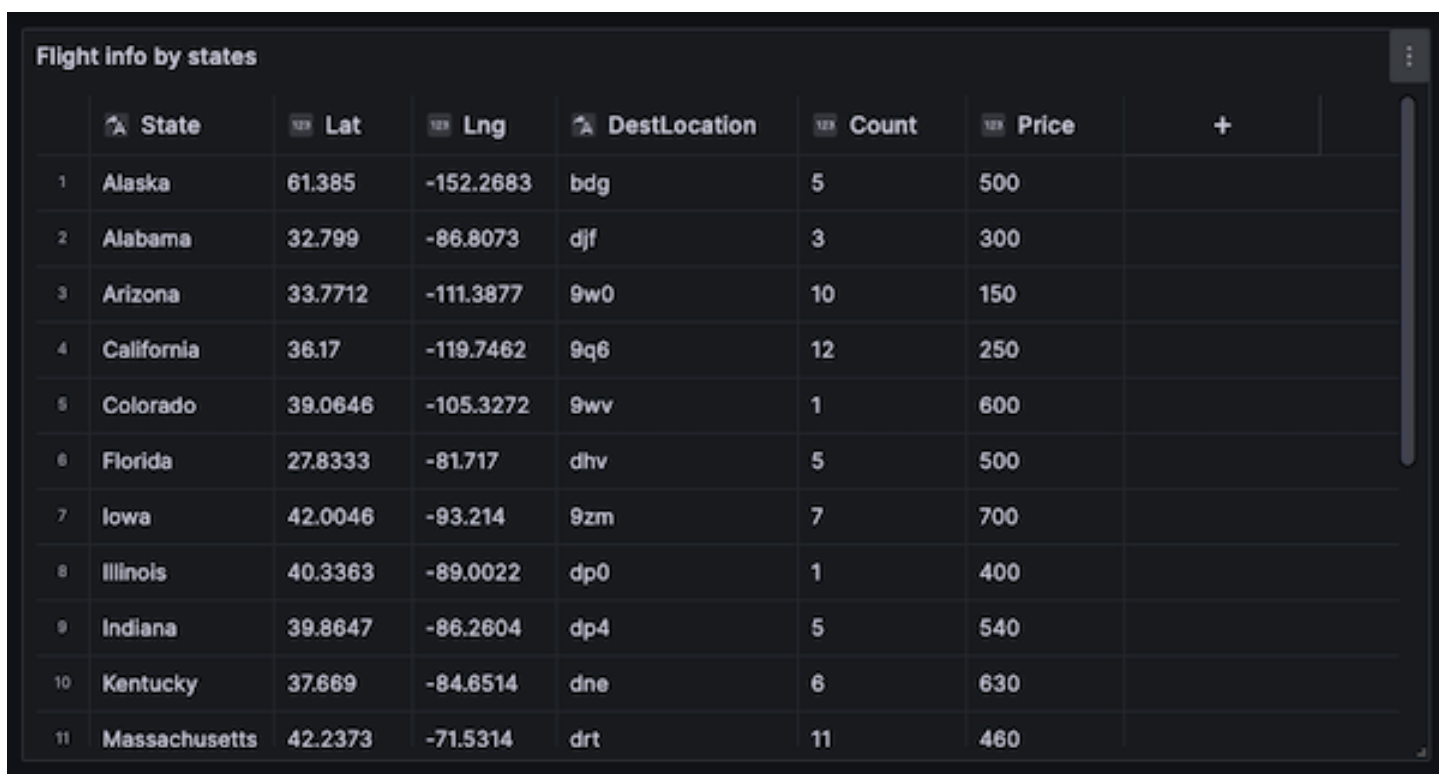
- ⚠ このドキュメントトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けに設計されています。  
Grafana バージョン 9.x をサポートする Grafana ワークスペースについては、「」を参照してください。[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

### Note

データグリッドの視覚化は現在、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、一般提供前に重大な変更が発生する可能性があります。

Datagrids では、Grafana 内でデータを作成、編集、微調整できます。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	-84.8514	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、空白のスレートから開始したり、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、完全に削除したりして、空白のスレート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、現在のデータセットがダッシュボードモデルにコピーされます。

その後、組み込みの Dashboard データソースを使用してデータグリッドデータをプルすることで、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 と Dashboard のデータソースの詳細については、「」を参照してください [特殊なデータソース](#)。

## コンテキストメニュー

データグリッドには、セル、列ヘッダー、または行セクター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あります。

- 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。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できるだけでなく、名前の変更、フリーズ、または列のフィールドタイプの変換もできます。

## シリーズの選択

シリーズが複数ある場合は、パネルオプションのシリーズの選択ドロップダウンを使用して、優先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

Datagrids は、データとやり取りするためのさまざまな方法を提供します。行と列を編集、移動、消去、削除できます。また、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、特定の列のフィールドタイプを変換したり、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の各セルに存在し、選択すると新しい行の作成がトリガーされ、選択したセルもアクティブになります。

## 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押したままにします。
2. 列を希望の場所にドラッグします。
3. 列を解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トを選択して保持し、目的の場所にドラッグします。行を離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、他のセルを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 キーを使用してデータを削除したりできます。

### 複数の行または列の削除または消去

複数の行を削除またはクリアするには、次の手順を実行します。

### 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の左側) にカーソルを合わせると、行のチェックボックスが表示されます。

2. 使用する行のチェックボックスをオンにします。複数の連続した行を選択するには、Shiftキーを押しながら最初の行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 (または同等) します。
4. 行の削除 または 行の消去 を選択します。

列ヘッダーをクリックして、列にも同じルールが適用されます。

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削除できます。

## フレームグラフ

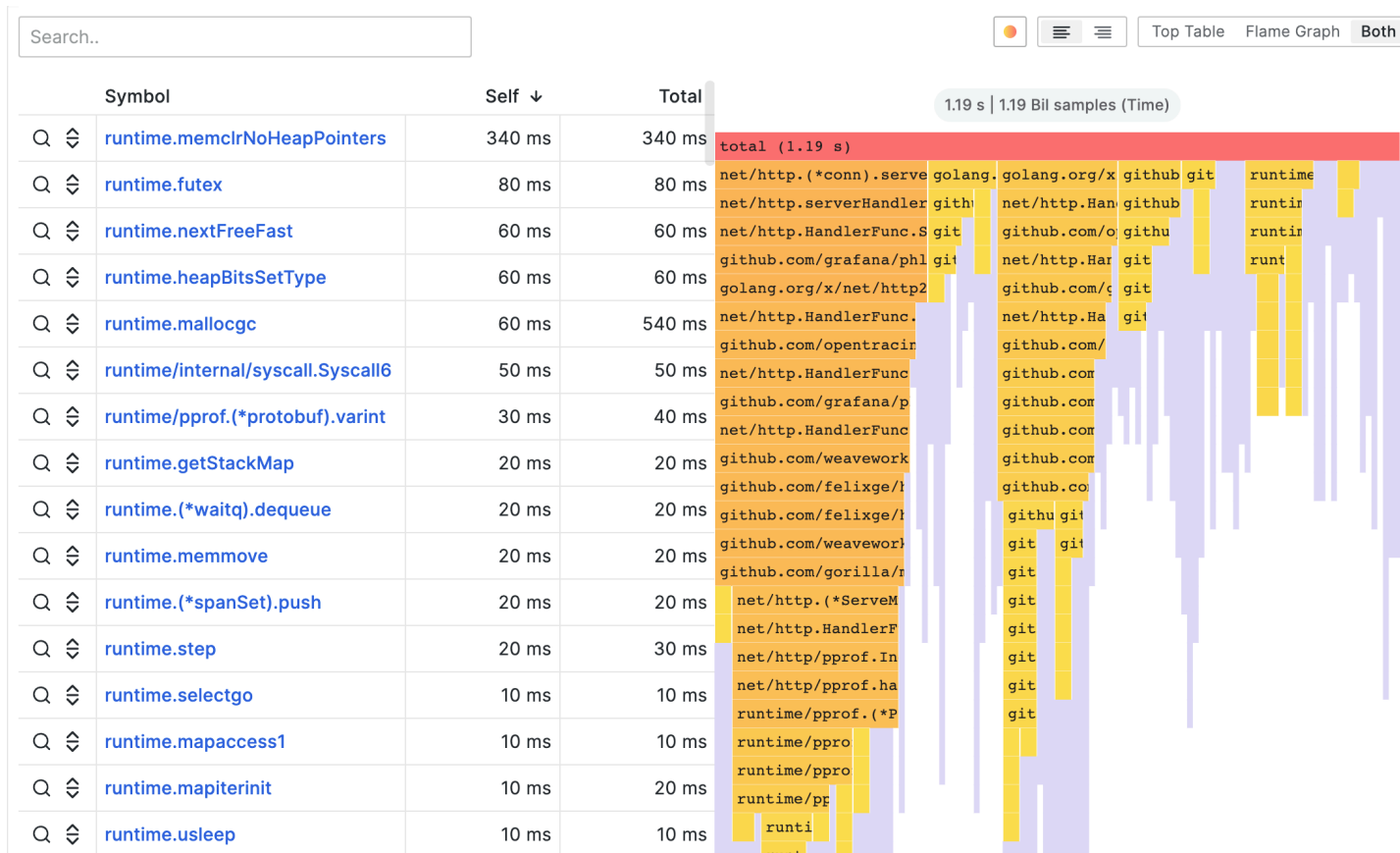
**⚠** このドキュメントトピックは、Grafana バージョン 10.x をサポートする Grafana ワークスペース向けに設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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、データのプロファイリングの階層的性質を活用します。これにより、データを形式に集約し、メモリの測定時に 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、幅はその値を表します。

グレースアウトされたセクションは、比較的小さな値を表す一連の関数であり、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、その関数の値、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ヒントが表示されます。

## ドロップダウンアクション

関数をクリックすると、追加のアクションを含むドロップダウンメニューが表示されます。

- 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、すべての子関数は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することが容易になります。

- 関数名のコピー – 関数名のコピー 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 – スカボライズビューでは、クリックされた関数のコンテキストを表示できません。関数のすべての発信者が上部に表示され、すべての発信者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表示するため、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表示され、冪形ビューに集約されます。

## ステータスバー

ステータスバーには、フレームグラフに関するメタデータと、グラフのどの部分に焦点が絞られているか、どの関数が墜落ビューに表示されるかなど、現在適用されている変更が表示されます。ステータスバーの 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内のすべての関数は色のままで、残りの関数はグレー表示されます。

### カラスキーマピッカー

色付け関数を値またはパッケージ名で切り替えて、同じパッケージの関数を視覚的に結び付けることができます。

### テキストの整列

テキストを左または右に揃えて、関数名がブロックに収まらない場合に、関数名のより重要な部分を表示します。

### 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

上の表は、プロファイルの関数をテーブル形式で示しています。テーブルには、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すると、合計時間または記号名で並べ替えることができます。各行は、関数がプロファイル内の複数の場所に表示される場合、指定された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冪形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、ネストレベルによってのみ整数値、メタデータ、およびデータフレーム内の順序で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ファーストトラバースルであり、子配列のようにデータフレーム内の可変長値なしで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最上位項目の間にある項目の数です。
value	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ます。
ラベル	string	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値です。

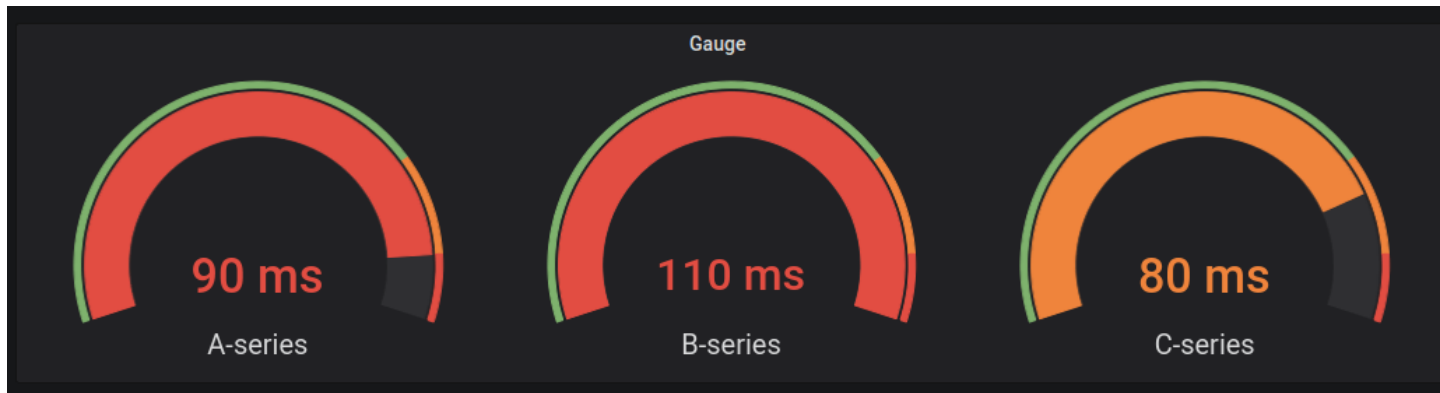
## Gauge

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ゲージは、各シリーズ、列、または行で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

以下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

すべての行に基づいて計算された値を表示します。

- 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リデューサー関数を選択します。使用可能な計算のリストについては、「」を参照してください[計算タイプ](#)。
- 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。

## Gauge

ゲージの表示方法を調整します。

### [Orientation] (向き)

スタッキング方向を選択します。

- 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。

### しきい値ラベルを表示する

しきい値を表示するかどうかを制御します。

### しきい値マーカを表示する

しきい値バンドが内部ゲージ値バンドの外部に表示され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する

- 自動 – Grafana が最適なゲージサイズを決定します。
- 手動 – ゲージサイズを手動で設定します。

## 最小幅

垂直方向の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に 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。

## 最小の高さ

水平方向のゲージの最小の高さをセクレートします。

最小の高さを設定すると、大量のデータがある場合に Y 軸スクロールバーが自動的に表示されません。

#### 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。

## 中立

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 – ゲージ値のサイズに数値を入力します。

## ジオマップ

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

ジオマップを使用すると、地理空間データを使用してワールドマップを表示および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ベースの特性に簡単に焦点を合わせることができます。

#### Note

によって提供され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、GeoMap パネルが最初にロードされたときに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。
- データへの適合は、マップレイヤーのデータ範囲とデータ変更時の更新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、単一のレイヤー、または選択したレイヤーの最後の値のデータに基づいて範囲を選択できます。
  - 1つ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トする場合は、レイヤーを選択できます。
  - パディングは、データの範囲を超えて相対的な割合でパディングを設定します (最終値のみを見る場合は利用できません)。
  - 最大ズームは、データをフィット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。
  - 緯度
  - 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
- ズームは初期ズームレベルを設定します。

## マップレイヤー

ジオマップは複数のレイヤーの表示をサポートしています。各レイヤーは、ベースマップの上に地理空間データを視覚化する方法を決定します。

## タイプ

Geomap 視覚化では、3 つのマップレイヤータイプから選択できます。

- [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。
- [ナイト/デイレイヤー \(アルファ\)](#) は、ナイトリージョンまたはデイ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。

実験的な (またはアルファ) レイヤータイプも 2 つあります。

- [最後のポイント \(アルファ\)](#) のアイコンは、最後のデータポイントのアイコンをレンダリングします。



- 動的 GeoJSON (アルファ ) は、クエリ結果に基づいて GeoJSON ファイルをスタイル設定します。

#### 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。Grafana Labs ではサポートが制限されており、この機能が一般公開される前に重大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、レイヤーの順序変更、削除を行うことができます。

- レイヤーを追加すると、ジオマップ用に設定可能な追加のデータレイヤーが作成されます。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、削除、順序を変更することができます。
  - レイヤー名を編集 (鉛筆アイコン) すると、レイヤーの名前が変更されます。
  - ごみ箱はレイヤーを削除します。
  - 順序を変更する (6 つのドット/ハンドル) と、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、ドラッグアンドドロップ時にレイヤーの順序が更新され、レイヤーの順序の選択が簡素化されます。

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

Geomaps には地理的データのソースが必要です。このデータはデータベースクエリから取得され、データには 4 つのマッピングオプションがあります。

- 自動は位置データを自動的に検索します。このオプションは、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に使用します。
  - geohash: geohash

- 緯度: latitude、lat
- 経度: longitude、lng、lon
- ルックアップ: lookup
- 座標は、クエリが座標データを保持す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に求められます。
- Geohash は、クエリが geohash データを保持するように指定します。データベースクエリから geohash の文字列データフィールドを選択するように求められます。
- Lookup は、クエリが値にマッピングする必要がある場所名データを保持するように指定します。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に求められます。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マーカー形状として表示できます。

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。

- サイズはマーカーのサイズを設定します。デフォルトは `fixed` です。これにより Fixed size、データに関係なくすべてのマーカーサイズが同じになります。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設定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マーカーサイズを設定する必要があります。
- シンボルを使用すると、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記号、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。イメージアセットを含む URL を指定することもできます。イメージはスケーラブルベクトルグラフィック (SVG) である必要があります。
- Symbol Vertical Align 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。シンボルのローテーション角度は、最初にデータポイントの周囲に適用され、次にシンボルのローテーションに対して垂直方向の配置が適用されることに注意してください。
- シンボル水平配置は、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。シンボルのローテーション角度は、最初にデータポイントの周囲に適用され、次にシンボルのローテーションに対して水平方向の配置が適用されることに注意してください。

- 色はマーカーの色を設定します。デフォルトでは、すべてのマーカーが特定の色にFixed color設定されます。選択したフィールドデータポイントの値と、標準オプションセクションで設定されたカラースキームに応じて、条件付き色を使用するオプションもあります。
- 塗りつぶし不透明度は、各マーカーの透明度を設定します。
- ローテーション角度は、各マーカーのローテーション角度を設定します。デフォルト値は固定値で、データに関係なくすべてのマーカーを同じ角度にローテーションします。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ローテーションを設定するオプションもあります。
- テキストラベルは、各マーカーのテキストラベルを設定します。
- 凡例を表示では、レイヤーの凡例を切り替えることができます。
- 表示ツールヒントを使用すると、レイヤーのツールヒント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 Layer のドロップダウンメニューで を選択し、 を選択しますHeatmap。

と同様にMarkers、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。

- 重み値は、ヒートマップクラスターの強度を設定します。固定値は、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 の範囲である必要があります。マーカーと同様に、ド

ポップダウンには、データ値に応じて重み値を自動的にスケーリングする代替オプションがありません。

- 半径は、ヒートマップクラスターのサイズを設定します。
- Blur は、各クラスターのぼかしの量を設定します。
- 不透明度は、各クラスターの不透明度を設定します。
- 表示ツールヒントを使用すると、レイヤーのツールヒントを切り替えることができます。

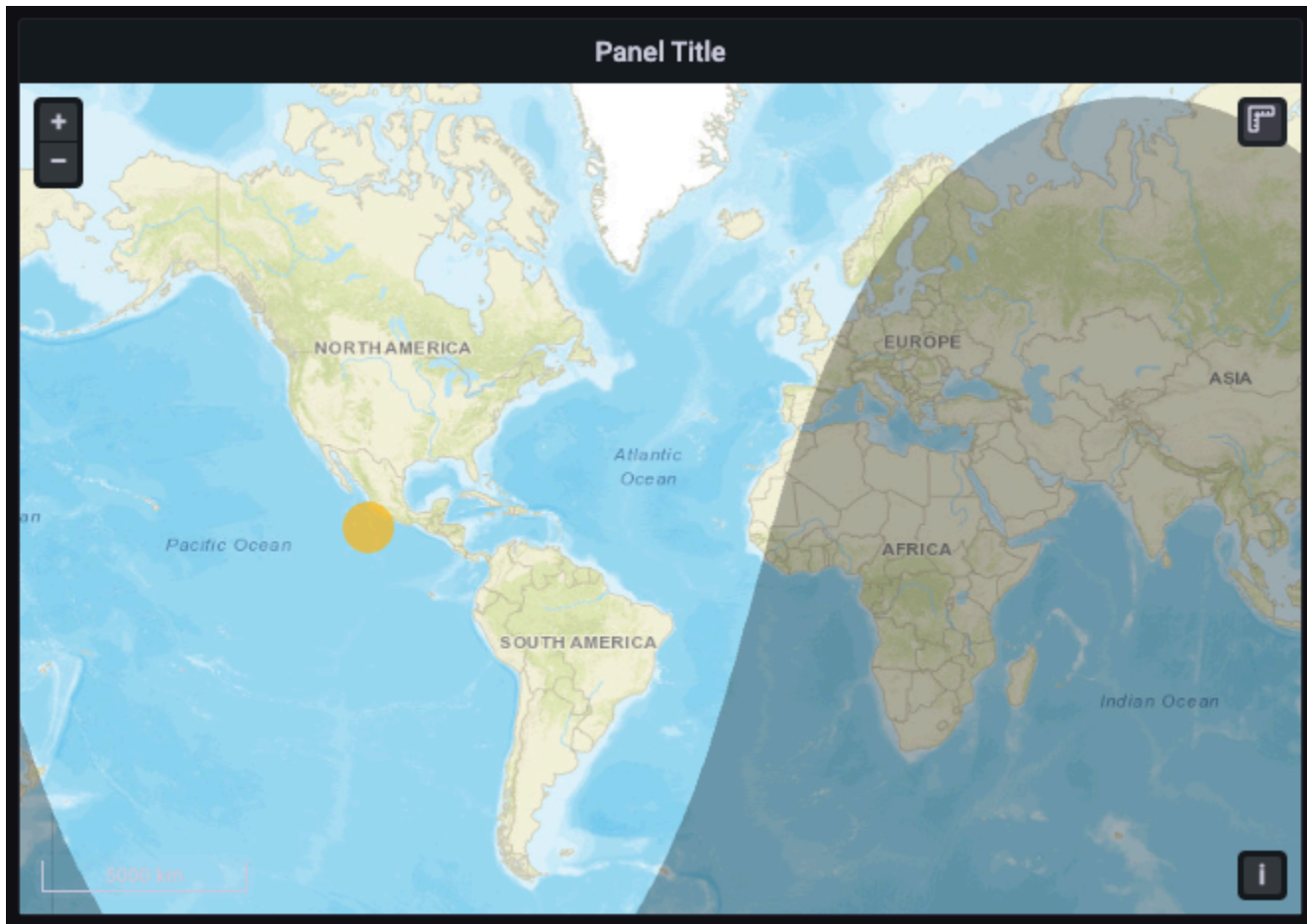
## GeoJSON レイヤー

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 は、Grafana に付属する GeoJSON ファイルの選択肢を提供します。
- デフォルトのスタイル 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。
  - 色はデフォルトのスタイルの色を設定します。
  - 不透明度はデフォルトの不透明度を設定します
- スタイルルールは、特徴プロパティに基づいてスタイルを適用します
  - ルールでは、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。
  - 色は、現在のルールのスタイルの色を設定します。
  - 不透明度は、現在のルールの透明度レベルを設定します。
- スタイルルールを追加すると、追加のスタイルルールが作成されます。
- ディスプレイツールヒントを使用すると、レイヤーのツールヒントを切り替えることができます。

## ナイト/デイレイヤー (アルファ)

ナイト/デイレイヤーには、現在の時間範囲に基づいてナイトリージョンとデイリージョンが表示されます。



## オプション

- Show は、時間ソースをパネルの時間範囲から切り替えます。
- ナイトリージョンの色は、ナイトリージョンの色を選択します。
- ディスプレイの太陽が太陽アイコンを切り替えます。
- 不透明度は 0 (透明) から 1 (不透明) です。
- ディスプレイツールヒントを使用すると、レイヤーのツールヒントを切り替えることができます。

### Note

詳細については、「- の [拡張機能 OpenLayers DayNight](#)」を参照してください。

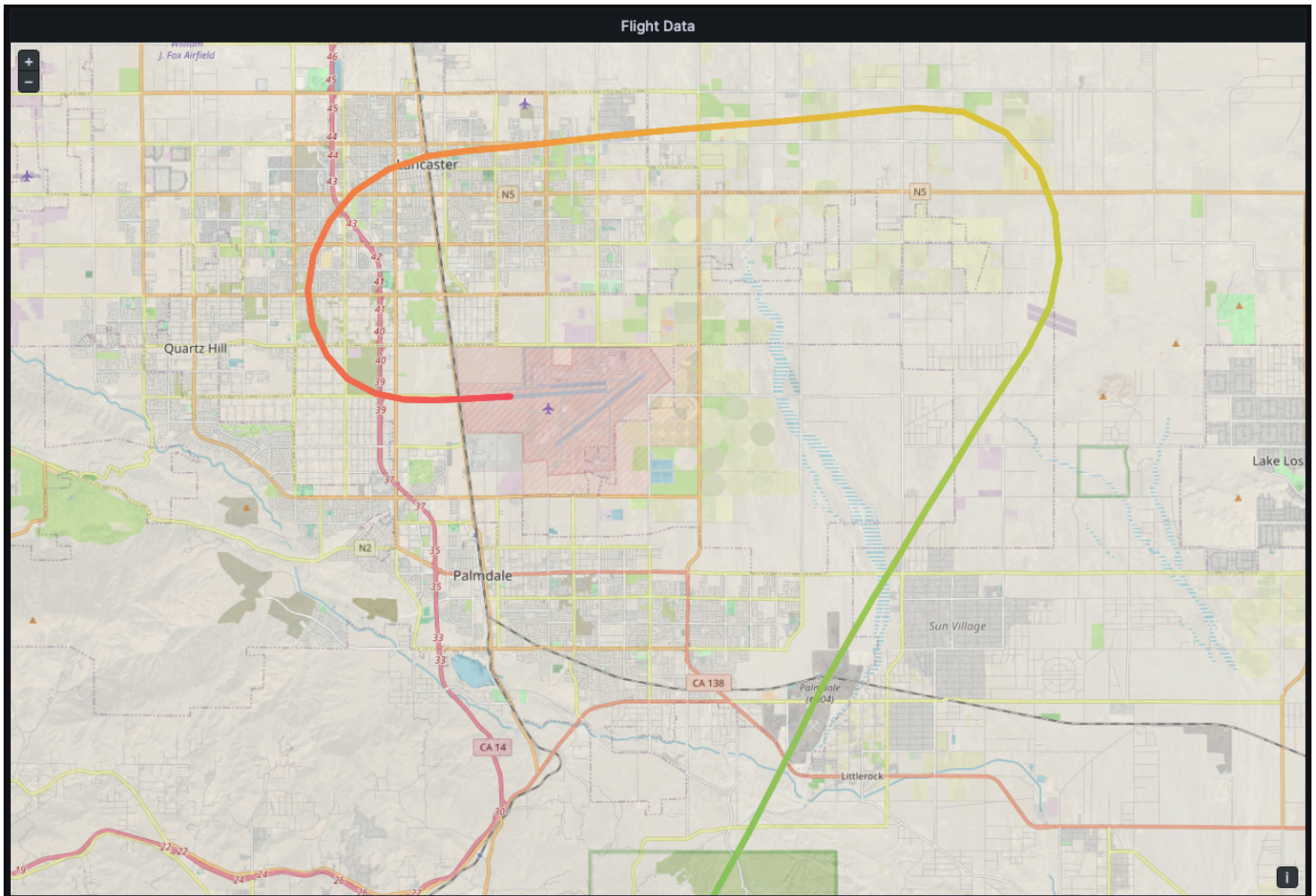
## ルートレイヤー (プレビュー)

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 ではサポートが制限されており、この機能が一般公開される前に重大な変更が発生する可能性があります。

**オプション**

- サイズはルートの太さを設定します。デフォルトで固定値。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リングできる最小範囲と最大範囲を設定できます。
- 色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し、色をフィールドデータに結び付けることもできます。
- 塗りつぶし不透明度は、ルートの不透明度を設定します。
- テキストラベルは、ルートごとにテキストラベルを設定します。
- 矢印は、データ順にルートに沿って表示されるように矢印のスタイルを設定します。

- なし
- 転送
- リバース
- 表示ツールヒントを使用すると、レイヤーのツールヒントを切り替えることができます。

#### Note

詳細については、[「の拡張 OpenLayers - フローラインスタイル」](#)を参照してください。

## 写真レイヤー (プレビュー)

Photos レイヤーは、各データポイントに写真を表示します。

#### 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 ではサポートが制限されており、この機能が一般公開される前に重大な変更が発生する可能性があります。

## オプション

- イメージソースフィールドでは、Base64 でエンコードされたイメージバイナリ () としてイメージデータを含む文字列フィールドを選択できます `data:image/png;base64,...`
- Kind はイメージの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)
  - アンカー
  - フォリオ
- トリミングは、画像が収まるようにトリミングするかどうかを切り替えます。
- シャドウは画像の背後にあるボックスシャドウを切り替えます。
- 境界線は、画像の周囲に境界サイズを設定します。
- 境界線の色は、画像の周囲に境界線の色を設定します。
- 半径は、イメージの全体的なサイズをピクセル単位で設定します。

- ツールヒントを表示すると、レイヤーのツールヒントを切り替えることができます。

### Note

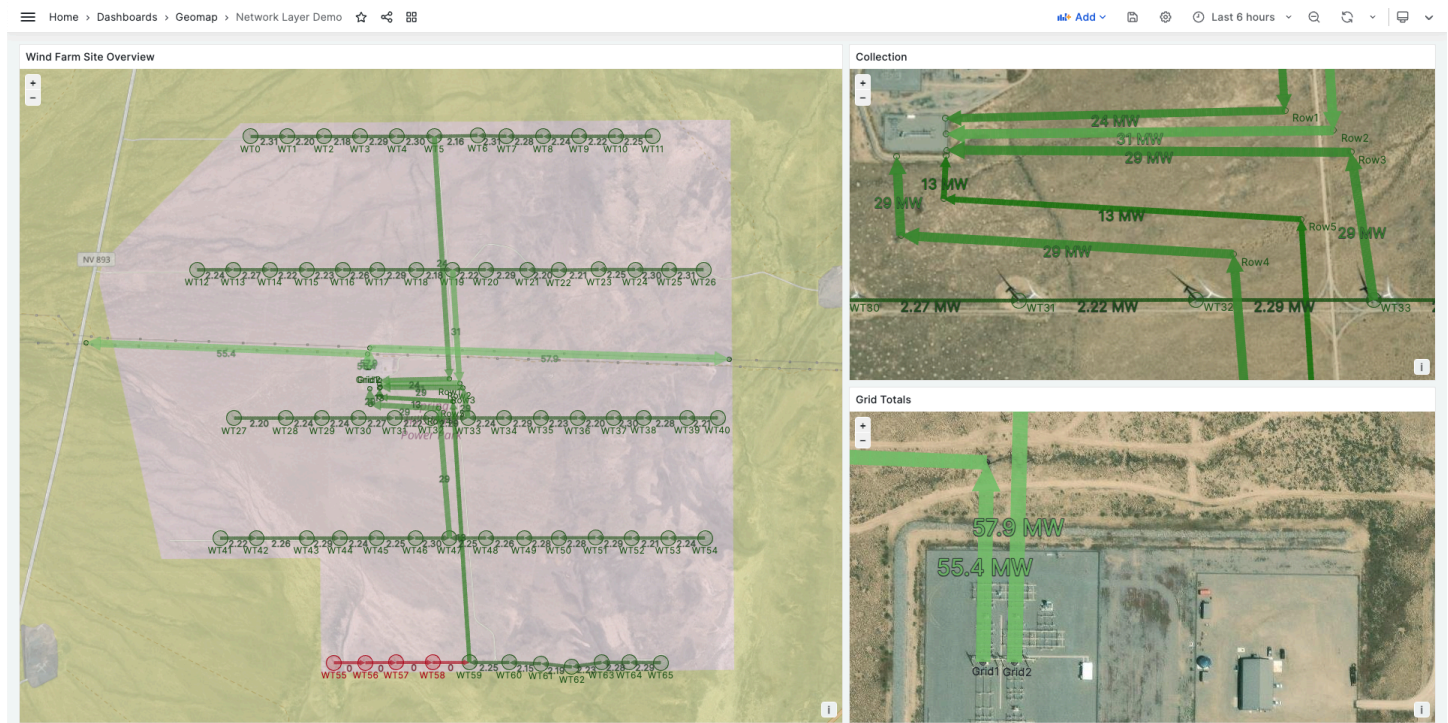
詳細については、[「Extensions for OpenLayers - Image Photo Style」](#)を参照してください。

## 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 ではサポートが制限されており、この機能が一般公開される前に重大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 ではサポートが制限されており、この機能が一般公開される前に重大な変更が発生する可能性があります。

## オプション

- 矢印は、各エッジに表示する矢印の方向を設定します。前方はソースからターゲットを意味します。以下から選択します。
  - なし
  - 転送
  - リバース
  - 両方
- 凡例を表示では、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。
- 表示ツールヒントを使用すると、レイヤーのツールヒントを切り替えることができます。

## ノードスタイル

- サイズはノードのサイズを設定します。デフォルトは固定サイズです。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設定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。
- 色はノードの色を設定します。デフォルトは固定色で、すべてのノードを特定の色に設定します。選択したフィールドデータポイントの値と、標準オプションセクションで設定されたカラースキームに応じて、条件付き色を使用するオプションもあります。
- シンボルを使用すると、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記号、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。イメージアセットを含む URL を指定することもできます。イメージはスケーラブルベクトルグラフィック (SVG) である必要があります。
- 塗りつぶし不透明度は、各ノードの透明度を設定します。
- ローテーション角度は、各ノードのローテーション角度を設定します。デフォルト値は固定値です。これにより、データに関係なくすべてのノードが同じ角度にローテーションされますが、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ローテーションを設定するオプションもあります。

- テキストラベルは、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ラインの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設定するオプションもあります。最小エッジサイズと最大エッジサイズは、エッジがこの範囲内でスケーリングできるように設定する必要があります。
- 色はエッジの色を設定します。デフォルトは固定色で、すべてのエッジを特定の色に設定します。選択したフィールドデータポイントの値と、標準オプションセクションで設定されたカラースキームに応じて、条件付き色を使用するオプションもあります。
- 塗りつぶし不透明度は、各エッジの透明度を設定します。
- テキストラベルは、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。

## オープンストリートマップレイヤー

Open Street Map レイヤー (デフォルトのベースマップ以外) 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の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。

ズームコントロールを表示する

左上隅にズームコントロールを表示します。

マウスホイールズーム


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をオンまたはオフにします。

属性を表示する

マップ上にベースマップレイヤーの属性を表示します。

スケールを表示する

左下隅にスケール情報を表示します。


 Note

[m]/[km] で単位を表示します。

メジャーツールを表示する

右上隅に測定ツールを表示します。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せん。

- クリックして測定を開始する
- クリックして測定を続行する
- ダブルクリックして測定を終了する

 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。

長さ

ジオメトリの球面長を取得します。この長さは、座標間の大きな円距離の合計です。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」にあると見なされます。

長さの測定には、次の単位を選択できます。

- メトリクス (m/km)
- フィート (ft)
- マイル (mi)
- ノーティカルマイル (nmi)

## エリア

ジオメトリの球面面積を取得します。この領域は、ポリゴンエッジが球上の大きな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」にあると見なされます。

エリア測定値には、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 (エーカー)
- ヘクタール (ha)

## デバッグを表示する

マップの右上隅にデバッグ情報を表示します。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。

- ズームは、マップの現在のズームレベルを表示します。
- センターには、現在の経度とマップセンターの緯度が表示されます。

## Tooltip

- None は、データポイントがクリックされた場合にのみツールヒントを表示します。
- ポインタがデータポイントにカーソルを合わせると、詳細にツールヒントが表示されます。

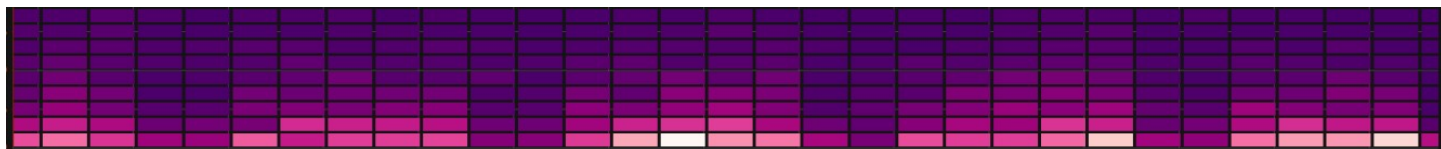
## ヒートマップ

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ヒートマップパネルの視覚化では、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、「」を参照してください[ヒストグラムとヒートマップの概要](#)。



### データから計算する

この設定により、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が決まります。

### 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、 の時間範囲1hは、X 軸でセルを 1 時間幅にします。

### 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。

### Y バケットスケール

次のいずれかの Y 軸値スケールを選択します。

- 線形 – 線形スケール。
- log (ベース 2) – ベース 2 の対数スケール。
- log (ベース 10) – ベース 10 の対数スケール。
- symlog – シンログスケール。

## Y 軸

Y 軸の表示方法を定義します。

### 配置

- 左 - 左
- 右 - 右
- 非表示 - 非表示

### 単位

#### 単位設定

#### 10 進数

この設定により、10 進数の設定が決まります。

#### 最小/最大値

この設定は、軸範囲を設定します。

#### リバーズ

選択すると、軸は逆の順序で表示されます。

#### 複数の Y 軸を表示する

場合によっては、複数の y 軸を表示することもできます。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 2 つのシリーズで異なる単位を持つ 2 つの y 軸を表示するとします。

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って、必要な数の y 軸を追加します。

#### カラー

色スペクトルは、値カウント (各バケット内) と各バケットに割り当てられた色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小数を表し、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自動的に反転されます。

色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よって変化します。

- Mode (モード)
  - スキーマ — セルの色で表されるバケット値。
    - スキーマ – モードがスキーマ の場合は、カラースキームを選択します。
  - 不透明度 – セル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。
    - 色 - セルのベース色。
  - スケール — バケット値を不透明度にマッピングするためのスケール。
    - 線形 – 線形スケール。バケット値は不透明度に直線的にマッピングされます。
    - sqrt – パワースケール。セルの不透明度は  $\text{value}^k$  として計算されます。ここで  $\text{value}$  はバケット値、 $k$  は設定された指数値です。指数が より小さい場合1、対数スケールが得られます。指数が より大きい場合1、指数スケールが得られます。の場合1、スケールは線形と同じになります。
  - Exponent – 指数の値、 より大きい0。

## 値からの開始/終了色

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。Min と Max を使用すると、これらの値を上書きできます。バケット値を Z 軸として、最小と最大をそれぞれ Z-Min と Z-Max として考えます。

- 開始 — セルの色計算に を使用する最小値。バケット値が Min より小さい場合は、「最小」色にマッピングされます。シリーズ最小値はデフォルト値です。
- 終了 — セルの色計算に を使用する最大値。バケット値が Max より大きい場合は、「最大」色にマッピングされます。シリーズの最大値はデフォルト値です。

## セル表示

これらの設定を使用して、視覚化を絞り込みます。

## 追加の表示オプション

### Tooltip

- ツールヒントを表示 - ヒートマップツールヒントを表示します。

- ヒストグラムを表示 — ツールチップに Y 軸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のバケット値の分布を表します。
- カラースケールを表示 — 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、バケット値と色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。

## 例

例データの表示に使用する色を設定します。

## Histogram

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ヒストグラムの視覚化は、値の分布を計算し、棒グラフとして表示します。Y 軸と各バーの高さは各角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は、時系列と、1つ以上の数値フィールドを持つテーブル結果をサポートします。

## 表示オプション

これらのオプションを使用して、視覚化を絞り込みます。

## バケットサイズ

バケットのサイズ。自動バケットサイズ設定 (全範囲の 10% まで) の場合は、これを空のままにします。

## バケットオフセット

最初のバケットがゼロから開始すべきでない場合。ゼロ以外のオフセットは、集計ウィンドウをシフトします。例えば、デフォルトの 0 オフセットを持つ 0~5、5~10、10~15 の 5 サイズのバケットは、2~7、7~12、12~17 になり、オフセットは 2 になります。この場合、オフセットは 0、5、または 10 で実質的に何もしません。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されます。この設定に影響を与えるには、オフセット量が 0 より大きく、バケットサイズより小さい必要があります。この範囲外の値は、この範囲内の値と同じ効果を持ちます。

## シリーズを組み合わせる

これにより、すべてのシリーズとフィールドが結合ヒストグラムにマージされます。

## 線幅

棒の線幅を制御します。

## 不透明度を埋める

バー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

勾配フィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – 勾配塗りつぶしなし、これがデフォルト設定です。
- 不透明度 – 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。
- Hue – 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、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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シングルカラーまたは Classic パレットに設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」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、視覚化に凡例がどのように表示されるかを定義します。

- List – 凡例をリストとして表示します。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。
- Hidden – 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 – 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。複数の を持つことができます。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。複数の を選択できます。

## ログ

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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からのログ行が表示されます。通常、グラフパネルの横にある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、クエリタブに入力されたクエリの結果が表示されます。複数のクエリの結果はマージされ、時間でソートされます。データソースが返す行数が表示可能な行数よりも多い場合は、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設定を使用できます。設定されていない場合、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式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、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 がログレベルを特定できない場合、不明なログレベルで視覚化されます。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、より堅牢な操作のために、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、視覚化を絞り込みます。

- Time – 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に関連付けられた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。これは、非共通ラベルのみを表示します。
- 共通ラベル – 共通ラベルの表示/非表示
- ラップライン – ラインラップを切り替えます。
- Prettify JSON – これを `true` に設定すると、すべての JSON ログがプリティーに出力されます。この設定は、JSON 以外の形式のログには影響しません。
- ログの詳細を有効にする – オプションを切り替えて、各ログ行のログの詳細ビューを表示します。デフォルトの設定は、`true` です。
- 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に設計されています。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、「[」](#)を参照してください。[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。


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 ブログの記事が表示され、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。このビジュアライゼーションタイプは他のクエリを受け付けないため、ユーザーはこのビジュアライゼーションを使用して RSS フィードデータを改ざんまたはクエリできることを期待できません。

#### 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ロードされます。その結果、適切な [CORS ヘッダー](#) で設定された RSS フィードのみがロードされます。表示しようとしている RSS フィードのロードに失敗する場合は、RSS フィードを再ホストするか、独自のプロキシを作成することを検討してください。

## ノードグラフ

-  このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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するため、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開発者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。

- ノードは円として表示されます。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性のあるその他のものを表す場合があります。
- 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つのノード間のその他の関係である可能性があります。

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、データソースごとに異なるタイプの値を表示したり、一部の値を表示したりすることはできません。

## ノード

通常、ノードにはノード内に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。ノードは、ノードの周囲に別の値のセットを色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、エラーの割合を円の赤い部分で表すことができます。

追加の詳細をコンテキストメニューで表示でき、ノードを選択すると表示されます。コンテキストメニューには、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もあります。

### 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

エッジにカーソルを合わせると、エッジに統計が表示されることもあります。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くことができます。

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。

## ノードグラフのナビゲーション

### パン

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、ポインタをドラッグします。

## ズームインまたはズームアウト

ノードグラフの左上隅にあるボタンを使用するか、マウスホイールまたはその他のスクロール入力を Ctrl (または Cmd) キーで使用してズームできます。

## 隠しノードを調べる

一定の時間に表示されるノードの数は、妥当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およその数を示す選択可能なマーカーの背後に非表示になります。マーカーを選択すると、そのノードの周囲に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、グラフ内の最も関心のある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の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、グラフィックアウトで表示オプションを選択して、選択したノードに焦点を当てたグラフィックアウトに切り替え、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。オプションで、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、ノードを説明する 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 `frame.meta.preferredVisualisationType = 'nodeGraph'` に を設定するか、edges それぞれ nodes と に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	string	エッジの一意の識別子。
ソース	string	ソースノードの ID。
target	string	ターゲットの ID。

#### オプションフィールド :

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るとオーバーレイに表示される最初の統計。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。
セカンダリ統計	文字列/数値	mainStat と同じですが、その下に表示されます。
detail__*	文字列/数値	エッジをクリックすると、というプレフィックスが付いたフィールドdetail__がコンテキストメニューのヘッダーに表示されます。より人間が読めるラベルconfig.displayName には 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	string	ノードの一意的識別子。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。

## オプションフィールド :

フィールド名	タイプ	説明
title	string	ノードのすぐ下に表示されるノードの名前。
subtitle	string	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	ノード自体内に表示される最初の統計。値はそのままの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。
セカンダリ統計	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せん。
arc_*	数値	プレフィックスが付いたフィールドarc_*はすべて、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。を使用して色

フィールド名	タイプ	説明
		を指定できません <code>config.color.fixedColor</code> 。
<code>detail__*</code>	文字列/数値	ノードをクリックすると、というプレフィックスが付いたフィールド <code>detail__</code> がコンテキストメニューのヘッダーに表示されます。より人間が読めるラベル <code>config.displayName</code> には を使用します。
<code>color</code>	文字列/数値	<code>arc__</code> 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 (許容される HTML カラー文字列) でも、数値でもかまいません。その場合、動作は <code>field.config.color.mode</code> 設定によって異なります。これは、例えば、フィールド値によって制御されるグラデーション色を作成するために使用できます。
アイコン	string	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 の組み込みアイコンのみが許可されず (使用可能なアイコンについては、 <a href="#">こちら</a> を参照してください)。

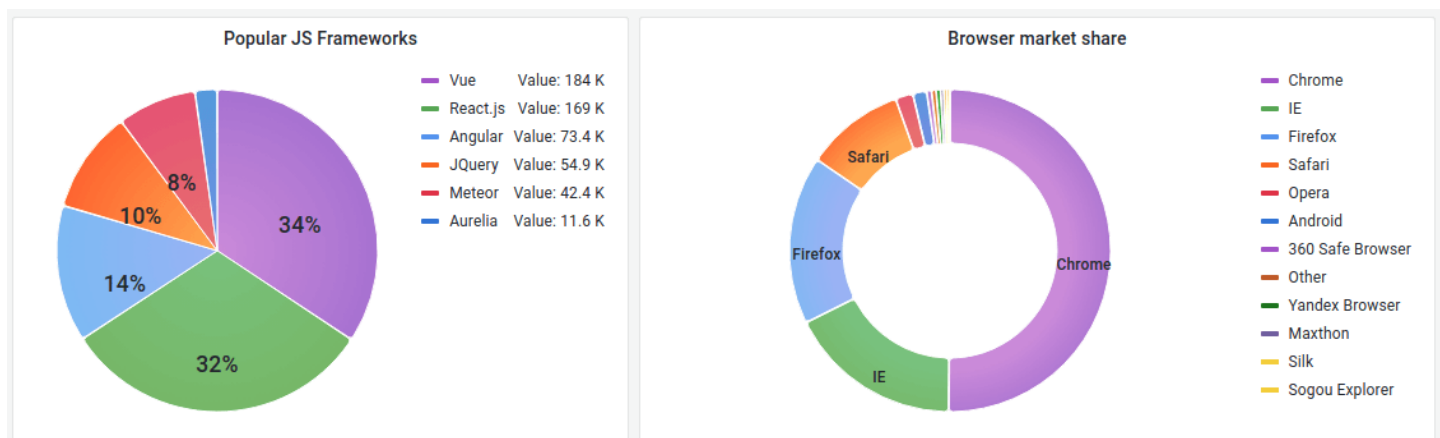
フィールド名	タイプ	説明
nodeRadius	数値	ピクセル単位の半径値。ノードサイズを管理するために使用されます。
強調表示	ブール値	ノードを強調表示するかどうかを設定します。例えば、複数のノードとエッジを強調表示してグラフ内の特定のパスを表すには、を使用します。デフォルトは false です。

## 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。



円グラフには、1つ以上のクエリからの縮小されたシリーズまたは値が、相互に関連して円のスライス形式で表示されます。スライスの円弧の長さ、面積、中心角度はすべて、すべての値の合計に関連

するため、スライス値に比例します。このタイプのグラフは、小さな値のセットを、気楽な形ですばやく比較したい場合に最適です。

## 値オプション

次のオプションを使用して、視覚化の値を調整します。

### 表示

表示する情報の量を選択します。

- Calculate – 各値をシリーズごとに 1 つの値に減らします。
- すべての値 – 1 つのシリーズのすべての値を表示します。

### 計算

計算を選択すると、計算が選択されたときに各シリーズが減ります。使用可能な計算については、[「計算タイプ」](#)を参照してください。

### 制限

単一のシリーズから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少なくとも 1 つ選択します。各フィールド名はリストで使用できません。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。
- Time – 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を使用して、視覚化の外観を絞り込みます。

### 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 – 標準の円グラフ

- 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。複数の を選択できます。

- 名前 – シリーズ名またはフィールド名。
- Percent – 全体のパーセンテージ。
- 値 – 未加工の数値。

デフォルトでは、ラベルはチャートの本文に白で表示されます。より暗いグラフの色を選択して、より見やすくすることができます。長い名前や数字はクリップされる場合があります。

## ツールヒントモード

視覚化の上にカーソルを置くと、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- Single – ホバーツールヒントには、視覚化にカーソルを合わせているシリーズが 1 つだけ表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – 視覚化を操作するときツールヒントを表示しないでください。

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

## 凡例オプション

これらの設定を使用して、視覚化に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

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 — 凡例をリストとして表示します。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 – グラフの右側。

## 凡例値

凡例に表示する値を選択します。複数の を選択できます。

- Percent – 全体のパーセンテージ。
- 値 – 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。

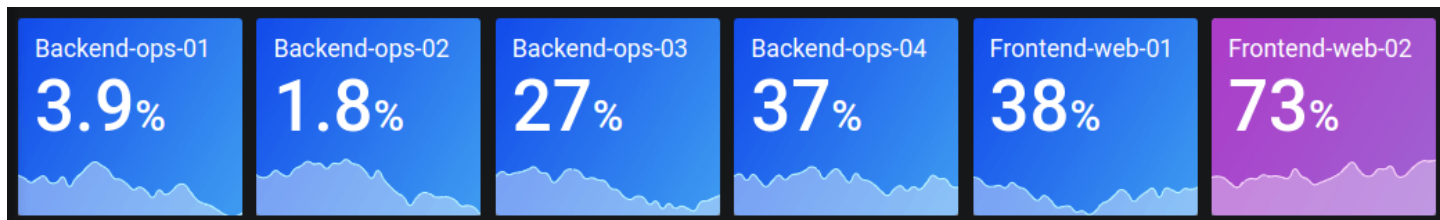
## Stat

**⚠** このドキュメントトピックは、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に設計されています。

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で 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、統計には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。
- 複数のシリーズ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、テキストを表示するかどうかを制御します。

### 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

以下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

すべての行に基づいて計算された値を表示します。

- 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リデューサー関数を選択します。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 — 視覚化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 — 視覚化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設定します。

#### [Orientation] (向き)

スタッキング方向を選択します。

- 自動 – Grafana は、最適な向き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縮します。
- Vertical – バーは上から下に垂直に伸縮します。

## テキストモード

テキストモードオプションを使用して、視覚化でレンダリングするテキストを制御できます。値が重要でない場合は、名前と色のみがである場合、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。代わりに、ホバーツールヒント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。値はホバーツールヒントに表示されます。
- なし – 何も表示しない (空)。名前と値は、ホバーツールヒントに表示されます。

## 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。

- オン – ワイドレイアウトが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に設定され、名前がに設定されている場合にのみ適用されます。ワイドレイアウトがオンになっている場合、パネルの幅が十分であれば、値と名前が右側に値 side-by-side とともに表示されます。ワイドレイアウトがオフになっている場合、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。
- 背景グラデーション – 値、グラフ領域、背景にわずかな背景グラデーションで色を適用します。
- 背景 実色 – 値、グラフ領域、背景に色を塗り、背景色を実色で塗ります。

## 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 – 値の下のエリア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。

## テキストの配置

アラインメントモードを選択します。

- 自動 – 値が 1 つだけ表示される場合 (繰り返しなし)、値は中央に配置されます。複数のシリーズまたは行が表示されている場合、値は左揃えになります。
- 中央 – 統計値は中央に配置されます。

## パーセント変更を表示する

パーセント変更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ません。

### Note

このオプションは、値オプションの Show 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。

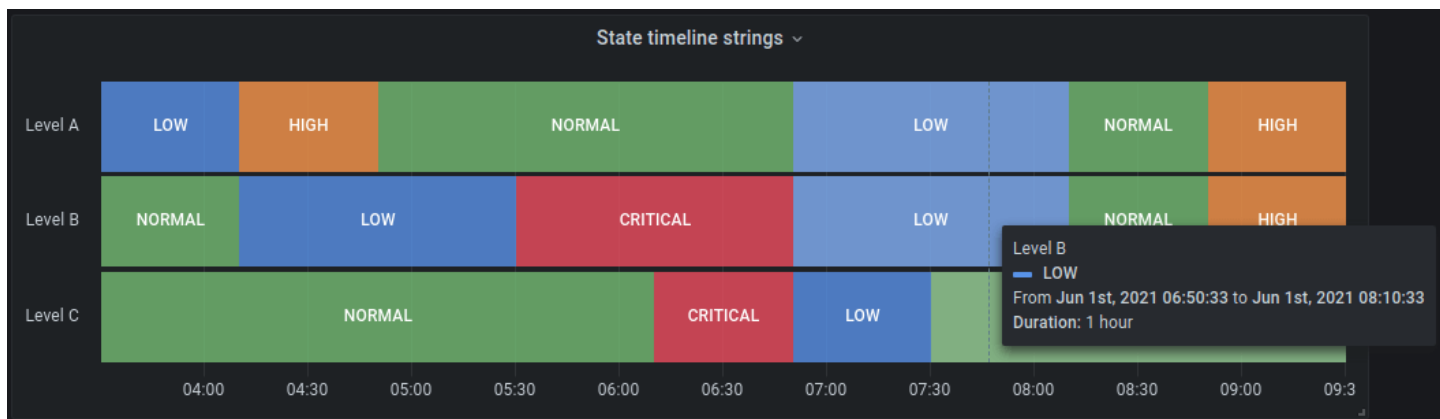
## 状態タイムライン

**⚠** 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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

状態タイムラインは、時間の経過に伴う個別の状態の変化を示します。各フィールドまたはシリーズが一意的な水平バンドとしてレンダリングされます。ステートリージョンは、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状態に適していますが、時系列でも使用できます。時系列とともに使用すると、しきい値を使用して数値を離散状態領域に変換します。



## 状態タイムラインオプション

視覚化を絞り込むには、以下のオプションを使用します。

### 等しい連続値をマージする

Grafana が同じ値を互いに隣り合っている場合にマージするかどうかを制御します。

### 値を表示する

値が状態リージョン内でレンダリングされるかどうかを制御します。十分なスペースがある場合、Auto は値をレンダリングします。

## 値の整列

状態リージョン内の値の配置を制御します。

## 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。

## 線幅

状態リージョンの線幅を制御します。

## 不透明度を埋める

状態リージョンの不透明度を制御します。

## NULL 値を接続する

データ内のギャップである NULL 値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、データのギャップが接続されなくなったしきい値に設定することもできます。

- Never – 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。
- Always – 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。

## 切断値

データ内のどの値を切断するかを超えるしきい値を設定するかどうかを選択します。

- Never – データ内の時系列データポイントが切断されることはありません。
- 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、この範囲外の値は接続されなくなった場合に便利です。

## 値マッピング

ブール値または文字列値に色を割り当てるには、[を使用します](#) [値マッピングを設定する](#)。

## 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リージョンに変換します。

## 凡例オプション

凡例オプションを有効にすると、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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「シングルカラー」または「クラシックパレット」に設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。

## 凡例モード

これらの設定を使用して、視覚化に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 — 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

## ステータス履歴

**⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ステータス履歴には、時間の経過とともに定期的な状態が表示されます。各フィールドまたはシリーズが水平行としてレンダリングされます。ボックスはレンダリングされ、各値を中心に配置されます。

### サポートされているデータ

ステータス履歴は、文字列、ブール値、および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を色付けしたり、テキスト値を数値範囲に割り当てることができます。

### 表示オプション

これらのオプションを使用して、視覚化を絞り込みます。

#### 値を表示する

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。十分なスペースがある場合、Auto は値をレンダリングします。

#### 列の幅

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。

#### 線幅

状態リージョンの線幅を制御します。

#### 不透明度を埋める

状態リージョンの塗りつぶしの不透明度を制御します。

#### 値マッピング

ブール値または文字列値に色を割り当てるには、を使用します[値マッピングを設定する](#)。

#### 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。グラデーションカラースキームを使用して色値を作成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、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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「シングルカラー」または「クラシックパレット」に設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。

## 凡例モード

これらの設定を使用して、視覚化に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。

- List – 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

## テーブル

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

テーブルは非常に柔軟で、時系列、テーブル、注釈、未加工の JSON データに対して複数のモードをサポートします。この視覚化では、日付の書式設定、値の書式設定、および色付けオプションも提供されます。

Bar gauge cell display mode					
Time	Info	Min	Max ↑	Value	
2020-09-15 12:45:11	down	73.6 °	76.5 °		74.0 °
2020-09-15 12:39:56	up	73.1 °	76.5 °		75.1 °
2020-09-15 12:27:41	down	72.9 °	76.5 °		74.2 °
2020-09-15 12:40:11	up	73.2 °	76.6 °		75.2 °
2020-09-15 12:27:26	up	73.9 °	76.6 °		74.2 °
2020-09-15 12:44:56	up	72.9 °	76.6 °		74.2 °
2020-09-15 12:39:26	up	72.7 °	76.6 °		74.7 °
2020-09-15 12:42:41	down	73.1 °	76.7 °		74.4 °
2020-09-15 12:51:41	down	73.0 °	76.7 °		75.4 °
2020-09-15 12:41:56	down fast	74.5 °	76.7 °		74.8 °

### Note

アノテーションとアラートはテーブルではサポートされていません。

## ソート列

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から昇順に変更します。列を選択するたびに、ソート順序はサイクルの次のオプションに変わります。追加の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。

### 列の幅

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、設定を上書きし、すべての列の幅をピクセル単位で定義できます。

例えば、と入力すると100、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は、その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 に設定すると75、すべての列の幅が 75 ピクセル以下にスケールされます。

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、デフォルトのピクセル値を 150 に減ら50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 がセルの内容をどのように整列させるかを選択します。

- Auto (デフォルト)
- 左
- 中央
- [Right] (右)

## セルタイプ

デフォルトでは、Grafana は自動的に表示設定を選択します。次のオプションのいずれかを選択して設定を上書きし、すべてのフィールドにデフォルトを設定できます。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対して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。

## カラーテキスト

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。

## 背景色 (グラデーションまたは単色 )

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。



## Gauge

セルはグラフィカルゲージとして表示でき、いくつかの異なるプレゼンテーションタイプがあります。

- 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 – ゲージは、ライトまたはライトのないスモールセルに分割されます。

さらに、ゲージと一緒に表示されるラベルは、値で色付けするか、テーマテキストの色と一致させるか、非表示にすることができます。

- 値の色
- テキストの色
- 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、JSON オブジェクトを参照できる JSON ビューがホバーに表示されます。

## 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。

## セル値検査

テーブルセルからの値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

セル値検査を使用できるのは、セル表示モードが Auto、Color text、Color background、または JSON View に設定されている場合のみです。

## 列フィルター

列データの表示方法は一時的に変更できます。例えば、値を最高から最低に順序付けしたり、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。

## ページ分割

このオプションを使用して、ページ分割を有効または無効にします。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。

## フィルターテーブル列

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で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。

フィルター (ファネル) アイコンが各列タイトルの横に表示されます。

## 列値をフィルタリングする

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 はその列のフィルターオプションを表示します。

表示する値の横にあるチェックボックスをオンにします。上部の検索フィールドにテキストを入力すると、それらの値がディスプレイに表示され、スクロールして検索するのではなく選択できるようになります。

列の値を表示するには、複数の演算子から選択します。

- 含む — 正規表現パターン (デフォルトではオペレータ) に一致します。
- 式 - ブール式を評価します。文字は、式の列値\$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「キャンセル」 ボタンの上にあるチェックボックスをオンにして、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。

### 列フィルターをクリアする

フィルターが適用された列には、タイトルの横に青いファネルが表示されます。

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、フィルターを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、次に計算するフィールドを選択できます。

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。

### 行をカウントする

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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

テキストの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、コンテキスト情報や説明を追加したり、複雑な HTML を埋め込んだりするために使用できます。

### Mode (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。以下のオプションがあります。

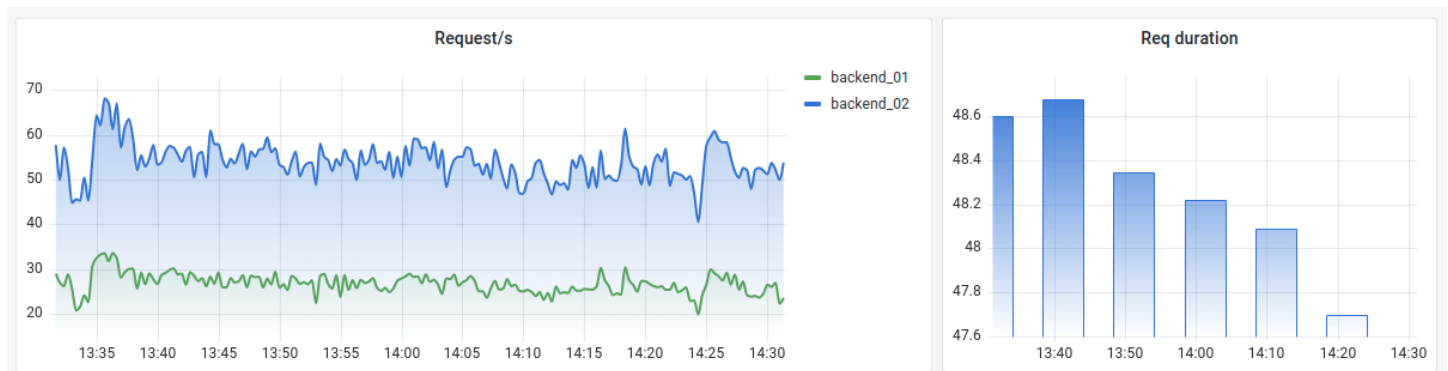
- Markdown – このオプションはコンテンツをマークダウンとしてフォーマットします。
- HTML – この設定では、コンテンツをサニタイズされた HTML としてレンダリングします。
- コード – この設定では、読み取り専用コードエディタ内でコンテンツをレンダリングします。適切な言語を選択して、埋め込みテキストに構文の強調表示を適用します。

## 変数

コンテンツ内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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。



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。線、ポイント、または棒としてシリーズをレンダリングできます。これらは、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を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

- [ツールヒントのオプション](#)
- [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](#)

## ツールヒントのオプション

- △** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ツールヒントオプションは、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。

## ツールヒントモード

- 単一 – ホバーツールヒントには、ホバーしているシリーズが 1 つのシリーズのみ表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – 視覚化を操作するときにツールヒントを表示しないでください。

## 凡例オプション

- △** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

凡例オプションは、グラフの下または右側に表示されるシリーズ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、視覚化に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、「」を参照してください[凡例を設定する](#)。

- List – 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

### グラフスタイル




- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。

- 線 - 時系列をグラフに線として表示します。
- 棒 - 時系列をグラフ上の一連の棒として、データポイントごとに 1 つずつ表示します。
- ポイント - 時系列をグラフにドットとして表示し、データポイントごとに 1 つずつ表示します。

## バーの配置

データポイントに対するバーの位置を設定します。ポイントがグラフに描画されます。棒の幅はあるため、ポイントの前、後、またはポイントの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。

-  Before - ポイントの前にバーが描画されます。ポイントはバーの後隅に配置されます。
-  中心 - 棒はポイントを中心に描画されます。ポイントはバー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。

## 線幅

線幅は、シリーズ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。

## 不透明度を埋める

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されます。

## グラデーションモード

グラデーションモードは、シリーズの色に基づいてグラデーションフィルを指定します。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。

勾配モードのオプションは次のとおりです。

- なし - 勾配フィルなし。これはデフォルトの設定です。
- 不透明度 - Y 軸の値が増加するとフィルの不透明度が増加する不透明度勾配。

- Hue – シリーズの色相に基づく微妙なグラデーション。
- スキーマ – カラースキームで定義される色グラデーション。この設定は、塗りつぶしエリアと行に使用されます。詳細については、「[色オプション](#)」を参照してください。

勾配の外観は、不透明度を塗りつぶす設定の影響を受けます。

### ポイントの表示

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。ポイントは、密度が低い場合に表示されます。
- Always – データ密度に関係なく、ポイントが表示されます。
- なし – ポイントを表示しないでください。

### ポイントサイズ

ポイントのサイズを、1~40 ピクセルの半径に設定します。

### 行補間

Grafana がシリーズ行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。
- スムーズ – ポイントは、ポイント間の移行をスムーズにする曲線で結合されます。
- Step before – 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。



ラインスタイルの選択肢は次のとおりです。

- 実線 — 実線を表示します。これはデフォルトの設定です。
- ダッシュ — 破線を表示します。このオプションを選択すると、ラインダッシュの長さギャップ (長さ、ギャップ) を選択するためのリストが表示されます。デフォルトでは、ダッシュ間隔は 10、10 に設定されています。
- ドット - 点線を表示します。このオプションを選択すると、ドット間隔のギャップの長さを選択するためのリストが表示されます。ドット間隔はデフォルトで 10 に設定されています。

## NULL 値を接続する

データ内のギャップである NULL 値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、データのギャップが接続されなくなるしきい値に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。

- Never – 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。
- Always – 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が接続されなくなった場合に便利です。

## 切断値

指定されたしきい値を超える時間があるデータの値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。

- Never – 時系列データポイントが切断されることはありません。
- 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、この範囲外の値は接続されなくなった場合に便利です。

## スタックシリーズ

積み上げにより、Grafana はシリーズを互いに重ねて表示できます。ビジュアライゼーションでスタッキングを使用する場合は、誤解を招くグラフを簡単に作成できるため、注意してください。ス

スタッキングが最適なアプローチではない理由の詳細については、[「スタッキングの問題」](#)を参照してください。

スタッキングの選択肢は次のとおりです。

- オフ – シリーズスタッキングをオフにします。
- Normal – スタックシリーズを互いに重ねて配置します。
- 100% – パーセンテージでスタックされ、すべてのシリーズの合計が最大 100% になります。

### グループ内のシリーズをスタックする

スタックの動作をグループ内のスタックシリーズに上書きできます。オーバーライドの作成の詳細については、「」を参照してください[フィールドオーバーライドを設定する](#)。

グループ内でシリーズをスタックするには

1. パネルを編集し、オーバーライド を選択します。
2. スタックシリーズ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. スタッキングモードでは、Normal を選択します。
4. シリーズを表示するスタッキンググループに名前を付けます。

スタッキング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。

以下を に入力します。

以下の Fill to オプションは、2 つのシリーズ間の領域を埋めます。このオプションは、シリーズまたはフィールド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、シリーズ行から 0 までではなく、2 つのシリーズ間の領域を埋めることができます。例えば、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 Fill 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 つのシリーズ線の間の領域のみが埋められます。

### 軸オプション

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

軸カテゴリのオプションは、X 軸と Y 軸のレンダリング方法を変更します。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しても有効になりません。を押すこともできますEnter。

### 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。

### Placement

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 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に割り当て、右軸を次の単位に割り当てます。
- 左 – 左側にすべての Y 軸を表示します。
- 右 – 右側にすべての Y 軸を表示します。
- 非表示 – すべての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ターゲット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 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。この設定により、軸が視覚的に互い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。

## グリッド線を表示する

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。
- オン – 常にグリッド線を表示します。
- オフ – グリッド線は表示されません。

## 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。
- シリーズ – シリーズの色に基づいて色を設定します。

## 境界線の表示

軸の境界線の可視性を設定します。

## スケール

Y 軸値のスケールリング方法を設定します。

- 線形 – スケールを等しい部分に分割します。
- 対数 – 対数スケールを使用します。このオプションを選択すると、バイナリ (ベース 2) または共通 (ベース 10) 対数スケールを選択するためのリストが表示されます。
- Symlog – 対称対数スケールを使用します。このオプションを選択すると、バイナリ (ベース 2) または共通 (ベース 10) 対数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。

## 中央ゼロ

Y 軸をゼロ中心に設定します。

## 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には、ソフト最小値またはソフトマックス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マックスの設定により、データの小さなバリエーションがほぼフラットな場合に拡大されないようにできます。対照的に、ハード最小値と最大値は、特定のポイントを超える断続的なスパイクをクリッピングすることで、データ内の有用な詳細が不明瞭になるのを防ぐのに役立ちます。

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、ツールヒント、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シリーズ値を変換します。次の 2 つの変換オプションがあります。

- 負の Y 変換 — 結果を Y 軸の負の値に反転させます。
- 定数 – 最初の値を定数行として表示します。

### 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。

## 複数の Y 軸を表示する

複数の y 軸を表示する場合があります。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 2 つのシリーズ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って、必要な数の y 軸を追加します。

## 色オプション

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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シリーズ色を割り当てます。凡例シリーズの色アイコンを選択して、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、特定のシリーズ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、シリーズカラーデフォルトを上書きするために使用できる追加オプションです。

### クラシックパレット

最も一般的な設定は、グラフに Classic パレットを使用することです。このスキームは、順序に基づいて各フィールドまたはシリーズ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。

### 単色

このモードを使用して色を指定します。凡例の各シリーズの横にある色付きの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。

### 値別のカラースキーム

From thresholds (値) や Green-Yellow-Red (値) などの値別カラースキームを選択すると、オプション別のカラーシリーズが表示されます。このオプションは、シリーズに色を割り当てるために使用する値 (最新、最小、最大) を制御します。

### スキーマ勾配モード

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、スキーマ という名前のモードがあります。スキーマ を有効にすると、線またはバー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。

### しきい値から

カラースキームが From しきい値 (値別) に設定され、グラデーションモードが スキーマ に設定されている場合、定義されたしきい値を超えると線または棒の色が変わります。スキーマ 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マ 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、シリーズ内の値が設定されたしきい値の間を移動するときに、選択した色が選択した色間にグラデーションを形成することを意味します。

## トレース

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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通過するリクエストに従うことができます。トレースの視覚化では、トレースデータを図に表示し、簡単に解釈できます。

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](#)
- [テンポデータソース](#)
- Grafana Labs [Tempo ドキュメント](#)の「Tempo の開始方法」。

### トレースビジュアライゼーション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ったら、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を使用するとtraceID、特定のトレース ID の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、「Variables [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](#)。

ダッシュボードで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トレース IDsを入力できます。トレース ID は視覚化できます。という変数を使用してテンプレートクエリとしてtraceId追加します。

トレースの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を追加を選択します。
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、「」を参照してください[パネルオプションの設定](#)。
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 `${traceId}` クエリフィールドに 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. ダッシュボードの設定に移動しtraceId、必要に応じて という新しい変数を 変数タイプカスタム に追加し、ラベルを付けます。Apply を選択して、変数をダッシュボードに追加します。
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。

## テーブルビジュアライゼーションでの TraceQL の追加

トレースの視覚化をダッシュボードに追加することはできますが、トレース IDs をダッシュボード変数として手動で追加する必要は煩雑です。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、前の手順を完了していることを前提としています。



テーブルの視覚化で TraceQL を追加するには

1. トレースの視覚化を追加したのと同じダッシュボードで、パネルを追加 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 の下に、トレースパネルのタイトルを入力します。
6. 適切な TraceQL クエリを追加して、ダッシュボードで視覚化するトレースを検索します。例えば、`my-server` というサーバーからのシンプルな静的クエリを次に示します。

```
{ .service.name = "my-server" && .http.status_code=500 }
```

TraceQL クエリをテンプレートクエリとして記述して、他のダッシュボード変数が存在する場合、それらを活用できます。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。

トレースの視覚化の結果には、トレースをレンダリングする探索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、トレースを同じダッシュボードで視覚化できます。

パネルで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. 右側のメニューのデータリンク で、リンクを追加 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. ダッシュボード全体を表示し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 の場合 :

```
https://g-example.grafana-workspace.us-east-1.amazonaws.com/d/1234abcd5/my-dashboar?orgId=1
```

次に、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
```

- 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。前のステップの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traceIdを使用する の値を入力します。トレース ID は、返された結果traceIDの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"]}
```

- データリンクを保存するには、保存を選択します。
- パネルエディタから適用 を選択して、ダッシュボードにパネルを適用します。
- 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 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探索ページを開くか、データリンクをたどるかを選択できます。他のフィールド (Start time、Name、などDuration) 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、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

トレンドビジュアライゼーションは、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、需要/供給関係、レースコースに沿った上昇プロットまたはハートレートプロット (x は開始からの距離または期間) などがあります。

トレンドビジュアライゼーションは、次の例外を除いて、[時系列ビジュアライゼーション](#)で使用できる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 (またはクロスヘア) なし
- マルチタイムゾーン x 軸なし
- ドラッグ選択によってダッシュボードの時間範囲を変更できない

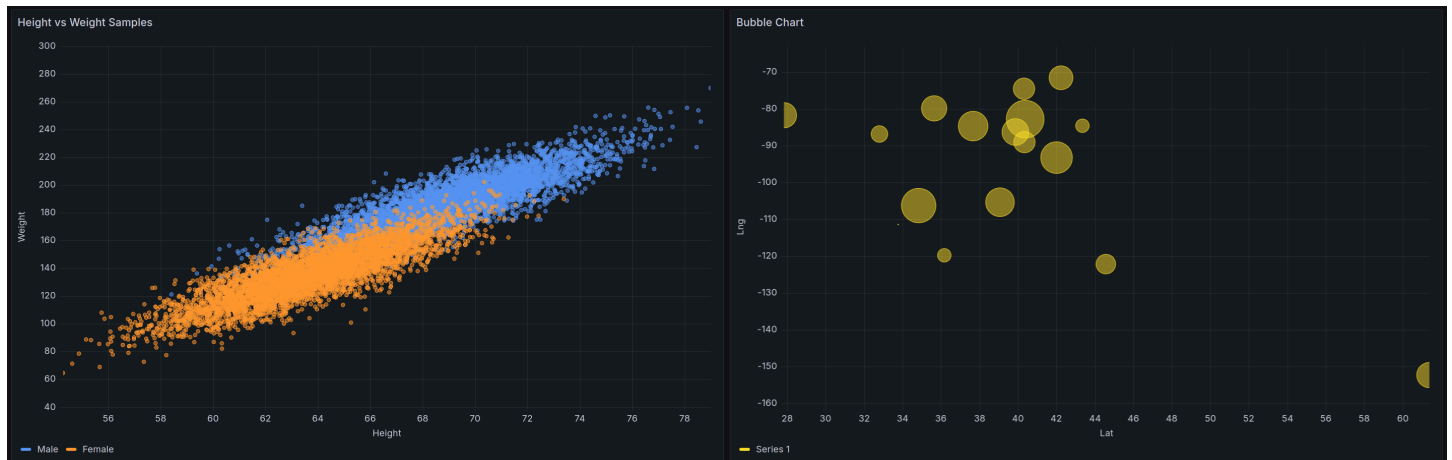
## X フィールドの選択

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、速度が x 軸にプロットされ、パワーとトルクが y 軸にプロットされるエンジンのパワーとトルクと速度を表すことができます。

## XY チャート

XY グラフでは、グラフ内の任意の x 値と y 値を視覚化できるため、2 つの変数間の関係を簡単に表示できます。XY グラフは通常、散布図の作成に使用されます。また、散布図を使用してバブルグラフを作成することもできます。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 チャートでは、少なくとも 2 つの数値フィールドを持つ任意のタイプの表形式データを使用できます。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を設定することもできます。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

#### シリーズマッピング

グラフでシリーズ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) からシリーズを自動的に生成します。フィルタリングしてフレームを 1 つだけ選択できます。
- 手動 – 使用可能なデータフレームから選択して、シリーズを明示的に定義します。

シリーズマッピングの選択に応じて、フレーム、X フィールド、Y フィールドのオプションが異なります。自動および手動シリーズ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。

#### 自動シリーズマッピングオプション

シリーズマッピングモードとして Auto 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。

- フレーム – デフォルトでは、XY チャートには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フレームを 1 つだけ選択できます。
- X フィールド – 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は his です。
- Y フィールド – 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。

チャートのシリーズが Y フィールドから生成されます。XY チャートのシリーズを変更するには、Y フィールドを上書きします。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、シリーズを生成しません。

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 Y フィールドごとに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: 名前が のフィールド
- オーバーライドプロパティ: シリーズ > エリアで非表示
- エリア: Viz

## 手動シリーズマッピングオプション

シリーズモードとして手動を選択すると、シリーズを追加、編集、削除できます。シリーズを管理するには、シリーズフィールドを選択します。シリーズの名前を変更するには、シリーズ名を選択します。

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。

- フレーム - 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フィールド - X が表すフィールドを選択します。
- Yフィールド - 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、グラフ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を基準にしています。

このオプションを選択すると、最小ポイントサイズオプションと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、グラフ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、グラフに1つのシリーズしか表示されていない場合に使用されます。

## 表示

視覚化での値の表現方法を設定します。

- 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。

- **Lines** – 値の間に行を追加します。このオプションを選択すると、線スタイルと線幅のオプションも表示されます。
- **両方** – ポイントとライン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、1~100 ピクセルの半径に設定します。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、ピクセルサイズをシリーズ (Y フィールド) で設定できます。

## 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します。これらのオプションは、特定のシリーズで上書きできます。

## 線のスタイル

行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。

- **実線** – 実線を表示します。これはデフォルトの設定です。
- **ダッシュ** – 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、ラインダッシュの長さギャップの設定を選択できます。デフォルトでは、長さギャップはに設定されます 10, 10。
- **ドット - 点線** – 点線を表示します。このオプションを選択すると、ドット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、ドット間隔はに設定されています 0, 10 (最初の数値はドットの長さを表し、常にゼロです)。

## 線幅

線の幅をピクセル単位で設定します。

## ツールヒントのオプション

ツールヒントオプションは、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。

## ツールヒントモード

- 単一 – ホバーツールヒントには、カーソルを合わせているシリーズが 1 つのシリーズのみ表示されます。
- 非表示 – 視覚化を操作するときツールヒントを表示しないでください。

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

### 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。default 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シリーズ名と統計を制御します。凡例の詳細については、「」を参照してください [凡例を設定する](#)。

### 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。

### Mode (モード)

これらの設定を使用して、視覚化に凡例がどのように表示されるかを定義します。

- リスト – 凡例をリストとして表示します。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 – グラフの右側。

### [値]

凡例に表示する [標準計算](#) を選択します。複数の を持つことができます。

### [Width] (幅)

視覚化の右側に配置する凡例の幅を制御します。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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

軸カテゴリのオプションは、X 軸と Y 軸のレンダリング方法を変更します。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しても有効になりません。を押すこともできますEnter。

## 配置 (y 軸)

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 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に割り当て、右軸を次の単位に割り当てます。
- 左 — 左側にすべての Y 軸を表示します。
- 右 — 右側にすべての Y 軸を表示します。
- 非表示 — すべての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ターゲット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、オーバーライドを使用して異なるラベルを割り当てることができます。

## [Width] (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、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。この設定により、軸が視覚的に互い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。

### グリッド線を表示する

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。
- オン – 常にグリッド線を表示します。
- オフ – グリッド線は表示されません。

### Color

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。
- シリーズ – シリーズの色に基づいて色を設定します。

### 境界線を表示する

軸の境界線の可視性を設定します。

### [Scale] (スケール)

Y 軸値のスケールリング方法を設定します。

- 線形 – スケールを等しい部分に分割します。
- 対数 – 対数スケールを使用します。このオプションを選択すると、バイナリ (ベース 2) または共通 (ベース 10) 対数スケールを選択するためのリストが表示されます。
- Symlog – 対称対数スケールを使用します。このオプションを選択すると、バイナリ (ベース 2) または共通 (ベース 10) 対数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。

### 中央ゼロ

Y 軸をゼロ中心に設定します。

## 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マックス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マックスの設定により、データの小さなバリエーションがほぼフラットな場合に拡大されないようにできます。対照的に、ハード最小値と最大値は、特定のポイントを超える断続的なスパイクをクリッピングすることで、データ内の有用な詳細が不明瞭になるのを防ぐのに役立ちます。

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、ツールヒント、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シリーズ値を変換します。次の 2 つの変換オプションがあります。

- 負の Y 変換 — 結果を Y 軸の負の値に反転させます。
- 定数 – 最初の値を定数行として表示します。

#### 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。

## 複数の Y 軸を表示する

複数の y 軸を表示する場合があります。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 2 つのシリーズ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って、必要な数の y 軸を追加します。

### 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、その変更はすべてのフィールドまたはシリーズに適用されます。フィールドの表示をより詳細に制御するには、「」を参照してください [フィールドオーバーライドを設定する](#)。

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/最大値を有効にして、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します。
- 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。シリーズ名、またはカーソルの下に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 を開く

詳細については、「[データリンクを設定する](#)」を参照してください。

## フィールドオーバーライド

オーバーライドを使用すると、特定のフィールドまたはシリーズの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。

次のいずれかのオーバーライドオプションから選択します。

- 名前のフィールド – 使用可能なすべてのフィールドのリストからフィールドを選択します。
- 正規表現に一致する名前のフィールド – 正規表現で上書きするフィールドを指定します。
- タイプのフィールド – 文字列、数値、時間などのフィールドをタイプ別に選択します。
- 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、最大、カウント、合計 など、定義したリデューサー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 でを試す

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana のダッシュボード UI には、視覚化用のダッシュボードを構築する機能があります。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詳しく調べます。これにより、クエリが機能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。

### **i** Note

データを探索するだけで、ダッシュボードを作成したくない場合は、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として表示します。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。

空の探索タブが開きます。

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探索オプションを選択します。これにより、パネルからクエリを含む探索タブが開き、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。

Open advanced data source picker を選択して、データソースの追加 (管理者のみ) 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が提供するクエリエディタを使用してクエリを記述します。

クエリの詳細については、「」を参照してください [データのクエリと変換](#)。

4. 右上隅にある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、ビジュアライゼーションを比較 side-by-side 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容易になります。

トップで分割ビューを開く

1. 探索ビューで、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 side-by-side つの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、タイムピッカーの1つにアタッチされたタイム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リンクできます (変更すると、もう1つも変更されます)。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間と終了時間が同期されます。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。

## コンテンツの概要

コンテンツの概要は、Explore で作成したクエリとビジュアライゼーションを追跡するサイドナビゲーションバーです。これにより、それら間をすばやく移動できます。

コンテンツの概要は分割ビューでも機能します。分割ビューでは、コンテンツアウトラインがペインごとに生成されます。

コンテンツの概要を開くには

1. 探索画面の左上隅にある概要ボタンを選択します。
2. コンテンツアウトライン内のパネルアイコンを選択して、そのパネルに移動します。

## Explore URLsを共有する

Explore を使用すると、クエリを変更すると、ブラウザのアドレスバーの URL が更新されます。この URL は共有またはブックマークできます。

**Note**

Explore は比較的長い URLs。URL が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。

## 外部ツールからの Explore URLsの生成

Explore URLs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 (最新バージョンの に設定する必要があります1。
- `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。

## 短縮リンクの共有

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 URLs を使用する代わりに、`/goto/:uid` 形式のより小さくてシンプルな URLs 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 (7) 日後に自動的に削除されます。リンクが少なくとも 1 回使用された場合、削除されません。

### 短縮リンクを絶対時間で共有する

ショートリンクには、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。短縮リンクをデフォルトで共有すると、選択した時間範囲が相対または絶対でコピーされます。短縮リンクの共有ボタンの横にあるドロップダウンボタンを選択し、Time-Sync URL Links のオプションの 1 つを選択すると、絶対時間で短い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いても、表示されている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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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、スタリングし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します。
- クエリを含む短縮リンクをクリップボードにコピーします。
- Star (お気に入り) クエリ。

## お気に入りのクエリを管理する

クエリ履歴タブに星が表示されているすべてのクエリは、星が表示されている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、ゼロから入力せずにこれらのクエリを再利用できます。

## クエリ履歴のソート

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でソートできます。

## クエリ履歴をソートするには

1. クエリをソートする 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。
  - 最新の 1 番目
  - 最も古いもの

## クエリ履歴のフィルタリング

クエリ履歴と星付きタブでクエリ履歴を特定のデータソースにフィルタリングできます。

## データソースに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する (複数可) 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、日付でクエリを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調整します。
- トップハンドルをドラッグして終了日を調整します。

## クエリ履歴での検索

クエリやコメントをまたいで履歴を検索できます。クエリ履歴タブと星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. クエリの検索 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。

## クエリ履歴の設定

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、次の表で説明します。

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**

クエリ履歴設定はグローバルで、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Explore では、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](#)
- [ロキ](#)

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。また、Explore では、ログを表示することで、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ができます side-by-side。

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

フルレンジログボリュームをサポートするデータソースを使用する場合、Explore 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を自動的に表示します。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、このフルレンジログボリュームはメトリクスクエリによってレンダリングされます。メトリクスクエリは、クエリされた時間範囲によっては高価になる可能性があります。このクエリは、小規模な Loki インストールでは特に処理が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、これらのクエリのカスタムタイムアウト (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、値を持つ 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 `Source=logvolhist`。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ログボリュームのロードをサポートしていない場合、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。時系列の終わりは、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サポートを提供していなくても、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方法について詳しく説明します。

### 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すると、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、行数の上限に達し、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、個別のページとしてナビゲーションに表示されます。すべてのページには、受信ログ行の `from` と `to` タイムスタンプが表示されます。各ページをクリックすると、以前の結果を表示できます。ログナビゲーションから実行された最後の 5 つのリクエストを探索するため、ページをクリックすると同じクエリを再実行せず、時間とリソースを節約できます。

### 視覚化オプション

ログの表示方法をカスタマイズし、表示する列を選択できます。

オプション	説明
Time (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に関連付けられた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。
ラップライン	ディスプレイtrueでラインラッピングを使用する場合は、これをに設定します。に設定するとfalse、水平スクロールになります。
JSON の指定	これをに設定するとtrue、すべてのJSONログがプリティーに出力されます。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上で行われます。署名は、すべての文字と数字を取り除き、残りの空白と句読点と一致するため、最も積極的な重複排除です。
結果の順序を表示する	受信したログの順序は、デフォルトの降順 (最新の順) 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または形式でダウンロードするにはjson、ダウンロード 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、便利でアクセス可能な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つい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。
- 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よって異なる場合があります、その結果、特定のデータソースについてこれらの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 (、) やタブ (\r) など \n、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 \t。このようなシーケンスを検出すると、Explore はエスケープ改行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. エスケープ改行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。その場合、オプションはエスケープ改行からエスケープの削除 変わります。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、置換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式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、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がログ行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を決定するために使用されるレベルラベルに抽出できます。これにより、ヒストグラムで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、ログレベルの略語と式のマッピング：

ログレベル	色	サポートされている式
非常事態	紫色	emerg、fatal、alert、crit、critical
エラー	赤色	エラー、エラー、エラー
warning	黄色	警告、警告
info	green	情報、情報、情報、通知
デバッグ	青色	デバッグ、デバッグ
トレース	水色	トレース
不明	グレー	*

**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特定し、焦点を絞ることが容易になります。

**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。

## ログの詳細ビュー

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、ログ行にアタッチされた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表示され、より堅牢な操作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、表示されるフィールドを2つの方法でフィルタリングできます。1つは正のフィルター (特定のフィールドに焦点を当てる) で、もう1つは負のフィルター (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、それに応じて等価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 との場合と同様に、ログの詳細では OpenSearch、フィールドが現在のクエリに既に存在しているかどうかチェックされ、アクティブ状態 (正のフィルターのみ) が表示され、クエリをオフにしたり、フィルター式を正から負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 にはデータリンクまたは関連機能があり、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。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、詳細をシームレスかつ便利に調べることができます。

### ログコンテキスト

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。これは、grep コマンドの `-C` パラメータに似ています。

テキストが長くなると、各ログエントリに関する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。ラップライン切り替えが便利な場所です。このトグルを有効にすると、Grafana は長いテキスト行を自動的にラップして、ビューワ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。

Open in split view 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、新しい探索ペインが開き、コンテキストクエリがログエントリとともに表示されるため、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クリップボードに簡単にコピーするには、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。ログ行の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する短い URL を生成してコピーできます。リンクを開くと、Grafana は自動的に対応するログ行までスクロールし、青色の背景で強調表示されるため、関連情報を簡単に特定して焦点を絞ることができます。

#### 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ません。

## ライブテール

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 のライブテール機能を活用できます。

ライブテールを使用してログをリアルタイムで表示するには

1. 探索ツールバーのライブボタンを選択して、ライブテールビューに切り替えます。
2. ライブテールビューでは、新しいログが画面の下部から表示され、背景がフェード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. ライブテールを一時停止し、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、すべてのログをディスプレイから削除するには、ログを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、ログ分析を続行するためのクリーンスレートが提供されます。
5. ライブテールを再開し、リアルタイムログを表示し続けるには、再開ボタンを選択します。

6. Live tailing を終了し、標準の Explore ビューに戻る場合は、停止ボタンを選択します。

ライブテール機能を使用すると、最新のログが入ってくる時に注意深く監視できるため、リアルタイムのイベントのモニタリングや問題を迅速に検出することが容易になります。

### ログのサンプル

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、メトリクスクエリでは、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への切り替え

を実装するメトリクスデータソース `DataSourceWithQueryExportSupport` (Prometheus など) から をサポートするログデータソース `DataSourceWithQueryImportSupport` (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をクエリします。

例えば、Loki データソースに切り替

え `grafana_alerting_active_alerts{job='grafana'}` 後の次の Prometheus クエリは、  
に変わります `{job='grafana'}`。これにより、選択した時間範囲内のログのチャンクが返され、削除/テキスト検索が可能になります。

### Explore でのトレース

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

Explore を使用して、データソースの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設定できます。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若干異なります。

各データソースのクエリについては、そのドキュメントを参照してください。

- [テンポ](#)
- [Jaeger](#)
- [AWS X-Ray](#)
- [Zipkin](#)

## トレースビューの説明

このセクションでは、トレースビュー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ハイライトします。
- メタデータ — トレースに関するさまざまなメタデータ。

### 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、より小さな時間範囲にズームインします。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ると、ズーム時に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、フィルタリングされたスパンがより具体的になります。

次のフィルターを1つ以上追加できます。

- リソースサービス名
- スパン名
- duration
- 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、「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、次のコンポーネントで構成されます。

- 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。
- スパン継続時間バー – トレース内のオペレーション継続時間の視覚的表現。

## スパンの詳細

スパン行を選択すると、次のような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すべてのタグ。
- プロセスメタデータ – このスパンを記録したプロセスに関するメタデータ。
- ログ – このスパンによってログに記録されたログのリストと、関連するキー値。Zipkin Logs 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をクエリします。

## メトリクスへのトレース

### Note

この機能は現在ベータ版です

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

トレースをプロファイルに使用すると、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、Grafana のさまざまなシグナルを関連付けることができます。

## 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を含む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

サービスグラフビューは、スパンメトリクス (レート、エラー率、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。要件を設定すると、この事前設定済みビューがすぐに利用可能になります。

詳細については、「[一時データソースページ](#)」を参照してください。Grafana Labs Tempo ドキュメントのサービス[グラフビューページ](#)も参照できます。

## Data API

この視覚化では、データを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、 を設定する必要があります  
`frame.meta.preferredVisualisationType = 'trace'`。

## 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	string	トレース全体の識別子。データフレームにはトレースが 1 つだけ必要です。
spanID	string	現在のスパンの識別子。SpanIDsはトレースごとに一意である必要があります。
parentSpanID	string	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ン <code>undefined</code> にすることができます。
serviceName	string	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	ミリ秒エポック時間単位のスパンの開始時刻。
duration	数値	ミリ秒単位のスパンの期間。

### オプションフィールド :

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ているログのリスト。
タグ	TraceKeyValuePair[]	現在のスパンに関連付けられているタグのリスト。
warnings	string[]	現在のスパンに関連する警告のリスト。
stackTraces	string[]	現在のスパンに関連付けられているスタックトレースのリスト。
errorIconColor	string	スパンが <code>error: true</code> でタグ付けされている場合のエラーアイコンの色 <code>error: true</code> 。

タイプの詳細については、「」の[TraceSpanRowTraceKeyValuePair](#)「」および[TraceLog](#)「」を参照してください [GitHub](#)。

## Explore の関連エディタ

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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、「」を参照してください [Grafana バージョン 10 の相関関係](#)。

### 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ビジュアライゼーションは、[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、相関を追加 を選択します ([コマンドパレット](#) から相関工データを選択することもできます)。

Explore は、青い境界線とトップバーで示される相関工データモードになりました。相関工データを終了するには、上部のバーで終了を選択します。

5. 新しいクエリの構築に使用できるデータにアタッチされたリンクを使用して、視覚化に対して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 – 各ログ行の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 – すべてのテーブルセルがリンクです。
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. ペインの上部にあるリストの変数を使用して、[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ています。
9. ラベルと説明を入力します (オプション)。ラベルは視覚化内のリンクの名前として使用され、変数を含めることができます。
10. 変換を指定します (オプション。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、相関工デー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存在す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に使用できます。相関関係の変換の詳細については、「」を参照してください[相関変換](#)。

エディタモードで生成されたリンクのいずれかを選択したら、変換ドロップダウンメニューで変換を追加を選択して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部分を選択します。例えば、ログ行などです。選択すると、このフィールドの値が変換の構築に役立ちます。
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、[相関変換](#)「」を参照してください。
3. 選択内容に基づいて、1つ以上の変数が入力されることもあれば、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もあります。
4. 「相関への変換の追加」を選択して、指定され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の正規表現の場合、他のドキュメントでmapValue参照されているはここで呼び出Variable Nameされます。Grafanaは、フィールド値の式に一致するテキストをすべて強調表示します。正規表現キャプチャグループを使用して、一致のどの部分を抽出するかを選択します。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で相関エディタを使用して相関を作成する方法を示します。

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、表形式の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、[テストデータソース](#)が設定されていることを確認してください。

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、相関を追加 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次のシナリオを選択します: CSV ファイル。
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をアタッチ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すべての変数が含まれています。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 フィールドは、各年の母集団値を含む変数を提供します: `${1980}`, `${2000}`, `${2020}`。これにより、変数値を使用してグラフが生成されます。
10. クエリエディタのエイリアス フィールドに、 と入力します `${State}`。

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るため、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。

どのデータソースにも同じ手順を適用できます。相関関係を使用すると、ビジュアライゼーションにリンクを作成して、選択したデータに基づいて動的クエリを実行できます。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

りを構築しました。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、[テストデータソース](#) が設定されていることを確認してください。

### 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、相関を追加 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. 分割ビューで開いた を探索します。左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すべての変数が含まれています。
8. 変換セクションを展開し、「変換を追加」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. もう一度変換を追加を選択し、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、 と入力します-([0-9]\\*)。これにより、ダッシュの右側にある任意の数字が選択されます。
15. 可変名 に、 と入力しますhostNumber。これにより、変数のリストが入力されます。
16. トランスフォーメーションを相関に追加 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、シナリオドロップダウンメニューを開き、CSV コンテンツ を選択します。
18. 以下の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すると、関連ボタンが表示されます。関連ボタンを選択すると、分割ビューが開きtime、(フィールド) msg、(ログラインから logfmt で抽出)、host number ( から正規表現で抽出hostname )、 status (ログラインから logfmt で抽出) が表示されます。

## Explore の Inspector

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。raw 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て、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と返される量を示します。
- クエリタブ – Grafana がデータソースにクエリを実行したときに送信された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た raw 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。

### raw クエリ結果の検査

raw 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、データ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：表示する結果セットデータを選択します。
- 時間で結合されたシリーズ：すべてのクエリの raw データを一度に表示し、列ごとに 1 つの結果セットを表示します。列見出しをクリックすると、データをソートできます。

### raw クエリ結果を CSV としてダウンロードする

Grafana は、デフォルトのブラウザのダウンロード場所に CSV ファイルを生成します。選択した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 raw データが表示されるまで、クエリ設定を絞り込みます。
3. [Download CSV] を選択します。

Excel 専用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、CSV のダウンロードオプションを選択する前に Excel 用ダウンロードのトグルをオンにします。

### TXT としてログ結果をダウンロードする

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のいずれかでトレース結果の JSON ファイルを生成します。Jaeger、Zipkin、または OTLP 形式です。

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 raw 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、特に数値が予期せず高い場合や低い場合に、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形式で表示されます。

## JSON モデルを表示する

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. Inspector 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、次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。
  - DataFrame structure – raw 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むと、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 raw リクエストとレスポンスを表示する

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の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 raw 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。クエリの特典部分をドリルダウンしたり、すべて展開または折りたたんだり、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るようにしたりできます。

## Grafana バージョン 10 の相関関係

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

相関関係を設定することで、Explore 視覚化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 はユーザー定義の相関関係を使用して、視覚化内にリンクを表示します。リンクをクリックして関連するクエリを実行し、「分割ビューを探索する」で結果を確認できます。

現在、相関関係に基づくリンクの表示をサポートしているビジュアライゼーションをご覧ください。

- [ログ](#)
- [テーブル](#)

相関関係は、Grafana の管理 > 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

- [相関設定](#)
- [新しい相関関係を作成する](#)

## 関連設定

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

各相関関係は、次のオプションで設定されます。

### 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

リンクが表示された結果のソース。

### 結果フィールド

リンクがビジュアライゼーションに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ソースデータへのオプション操作。

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の視覚化を探索するに表示されます。リンクは、相関設定 (結果フィールド) で提供された結果のフィールドの 1 つに割り当てられます。各視覚化では、リンクを含むフィールドが異なる方法で表示されます。



## ターゲットクエリ

ターゲットクエリは、ビジュアライゼーション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。リンクを選択すると、変数はソース結果の値で埋められます。使用できる変数には 2 つのタイプがあります。

- [フィールド変数](#) (フィールド値とラベルにアクセス)。
- 相関変数 (フィールド値と変換にアクセス)。

例: ソース結果に というフィールドが含まれている場合 `employee`、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 `${__data.fields.employee}`。
- 上記のフィールド値を にマッピングする相関変数 `${employee}`。

フィールド値を短い変数名にマッピングすることに加えて、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、選択したデータ行にすべての変数の値がある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新しい変数のセットです。

## 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、変換を結果フィールドとは異なるフィールドに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。の出力変数の例 `field = "host=srv001 endpoint=/test app=foo"` :

name	value
ホスト	srv001
エンドポイント	/テスト
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、最初的一致から変数が作成されます。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

名前付き一致グループのない単純な正規表現グループで使用します。デフォルトでは、最初的一致は、入力として使用されるフィールドの名前で変数を上書きします。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、ターゲット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便利です。

例: 選択したフィールド名が `employee`、フィールド値が `John Doe`。

式と `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
<code>/\w+ (\w+)/</code>	-	<code>employee=Doe</code>	<code>mapValue</code> が指定されていません。最初のマッチングは、既存のフィールド名変数 <code>()</code> にマッピングされません <code>employee</code> 。
<code>/(\w+) (\w+)/</code>	<code>name</code>	<code>name=John</code>	最初のマッチングは、という新しい変数にマッピングされます <code>name</code> 。
<code>/(?\w+) (?\w+)/</code>	-	<code>firstName=John , lastName=Doe</code>	名前付きグループを使用すると、出力変数の名前になり、 <code>mapValue</code> は無視されます。
<code>/(?\w+) (?\w+)/</code>	<code>name</code>	<code>firstName=John , lastName=Doe</code>	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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

相関関係は、Explore correlations 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、新しい相関を定義できます。

## Explore の相関エディタでの相関の作成

相関関係は、Explore 相関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、相関ページを開きます。
3. 右上隅にある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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana v10 では、Amazon Managed Grafana は更新されたアラートシステムである Grafana アラートへのアクセスを含み、アラート情報を 1 つの検索可能なビューに一元化します。Grafana アラートは Grafana v8 のオプション機能として導入 GrafanaLabs され、将来のバージョンでレガシーアラートの削除を発表しました。

### **i** 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、「」を参照してください[Classic ダッシュボードアラート](#)。

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について学習できます。

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、特定のイベントや状況を監視して、それらが見つかったときに通知を送信するようにアラートシステムを設定します。

これにより、手動による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のファーストラインが提供されます。

Grafana アラートを使用すると、データが保存されている場所に関係なく、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、データとアラートを組み合わせて、メトリクスとログを新しく独自の方法で柔軟に組み合わせることがができます。その後、1 つの統合ビューからアラートを作成、管理、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、データの近くで大規模なアラート式を実行できます。これらはすべて、すでに使い慣れている 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。

## 主な機能と利点

### すべてのアラートに 1 ページ

単一の Grafana アラートページは、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を 1 か所に統合します。

### 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ます。これにより、1 つのアラートルールだけでシステム全体の可視性を実現する能力と柔軟性が得られます。これを行うには、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、1 つ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、多次元アラートは各 CPU で警告し、標準アラートはサーバー全体で警告します。

### 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、および方法に関する一連のルールです。

### 無音アラート

サイレンスは、通知の作成を停止し、指定し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。特定の条件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、メインのアラートビューを整理することなく、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は、ポリシーの新しい通知が生成または送信されない定期的な時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間に発生し、繰り返し発生するのを防ぐために使用します。

無音と同様に、ミュートタイミングによってアラートルールの評価が妨げられることも、ユーザーインターフェイスにアラートインスタンスが表示されないようにすることもなくなります。通知の作成のみを防ぎます。

## アラートシステムの設計

複雑な IT システムをモニタリングし、すべてが正しく稼働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することは、ビジネス成果に影響を与え始める前に、問題が発生したときに通知するために不可欠です。

機能するアラート管理設定の設計と設定には時間がかかります。

ビジネス用に効果的なアラート管理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。

モニタリングと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なイベントや頻繁なイベントではなく、知ることが重要なイベントを見つけます。
- アラートは、即時の対応や介入が必要な大きなイベントに対してのみ作成する必要があります。
- 数量よりも品質を考慮します。

どのタイプのアラートを使用しますか？

- Grafana 管理のアラート、Grafana Mimir、Loki 管理のアラート、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したユーザーを選択します。通話中のユーザーや特定の Slack チャンネルに送信することを検討してください。
- アラート API またはアラートをコードとして使用して、可能な限り自動化します (Terraform)。

アラートの疲労を軽減するにはどうすればよいですか？

- 無音、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回避します。

- 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 アラートの制限事項

- 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、Alertmanager データソースからルールを取得できます。サポートされている他のデータソースから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センターに送信します。他のデータソースで定義され、Grafana で集計または表示されるアラートは定義されません。Prometheus 互換データソースで定義されたアラートを使用する場合は、Grafana アラートを有効にすることをお勧めします。

## トピック

- [概要](#)
- [アラートの設定](#)
- [アラートの設定](#)
- [アラートの管理](#)

## 概要

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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、対応に役立つ基礎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。



## 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。例えば、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、Alertmanager はアラートレシーバーであり、発音と解決された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 モデルに基づいて構築されています。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。Grafana アラートは、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ります。ただし、アラートのレシーバーとして Alertmanager を使用します。

アラートはアラートレシーバーに送信され、そこでルーティング、グループ化、禁止、無音、通知が行われます。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、アラートをいつ起動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、アラートルールが評価される頻度を決定する間隔、アラートが発動するために条件が満たされる必要がある期間で構成されます。

アラートルールは間隔にわたって評価され、各アラートルールには一度に発せられるアラートが 0、1、または任意の数のアラートを設定できます。アラートルールの状態は、最も severe アラートが多いことによって決まります。アラートルールは、通常、保留中、またはファイリングのいずれかです。例えば、アラートルールのアラートの少なくとも 1 つが発動している場合、アラートルールも発動します。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、エラー、および NoData です。

アラートルールの非常に重要な機能は、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含むアラートを計測し、アラートを特定の通知ポリシーに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、あるアラートにラベルが付き `foo=bar` であり、別のアラートにラベルが付いている場合などです `foo=baz`。アラートには、など多くのラベルを含めることができます `foo=bar, bar=baz` が、など同じラベルを 2 回持つことはできません `foo=bar, foo=baz`。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 などの同じラベルがある場合 `foo=bar, bar=baz, foo=bar, bar=baz`、アラートの 1 つが破棄されます。アラートは、アラートルールの条件が満たされなくなったとき、またはアラートルールが削除されたときに解決されます。

Grafana マネージドアラートでは、アラートの状態は、通常、保留中、アラート中、データなし、エラーのいずれかになります。Mimir や Loki などのデータソースの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 または エラーにすることはできません。

## 連絡先

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連絡先があるとします。

連絡先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、通知のタイトル、メッセージ、および構造を変更できます。通知テンプレートは、個々の統合や連絡先に固有ではありません。

## 通知ポリシー

通知ポリシーはアラートをグループ化し、それらを問い合わせ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一致します。これらは、アラートのラベルが正確に一致するか、完全に一致しないか、含まれていないか、または予想されるテキストの一部が含まれていないかをアサートする人間が読める式です。例えば、マッチャーはアラートをラベルと `foo=bar` 照合 `foo=bar` し、マッチャーは正規表現に一致する値を持つ `foo` という名前のラベルとアラートを `foo=~[a-zA-Z]+` 照合します `[a-zA-Z]+`。

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` 機能アラートを使用すると、任意の数の通知ポリシーを同時に一致させることができます。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のタイミング

無音とミュートのタイミングにより、特定のアラートの通知を一時停止したり、通知ポリシー全体を停止したりできます。サイレンスを使用して、メンテナンスウィンドウ中など、アドホックベースで通知を一時停止し、ミュートタイミングを使用して、夜間や週末など、一定の間隔で通知を一時停止します。

### 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](#)
- [連絡先](#)
- [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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。評価エンジンはバックエンドで実行されるため、Grafana アラートでは、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ことが知られています。

- [Amazon CloudWatch データソースに接続する](#)
- [Azure Monitor データソースに接続する](#)
- [Amazon OpenSearch Service データソースに接続する](#)
- [Google Cloud Monitoring データソースに接続する](#)
- [Graphite データソースに接続する](#)
- [InfluxDB データソースに接続する](#)
- [Loki データソースに接続する](#)
- [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 とオープンソースの Prometheus データソースに接続する](#)
- [Jaeger データソースに接続する](#)
- [Zipkin データソースに接続する](#)
- [Tempo データソースに接続する](#)
- [テスト用の TestData データソースを設定する](#)

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、「」を参照してください[データソースに接続する](#)。

## 数値データのアラート

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このトピックでは、時系列データではなく数値データに対するアラートを Grafana がどのように処理するか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) に直接アラートを送信したり、サーバーサイド式に渡したりできます。これにより、データソース内の処理と効率が向上し、アラートルールが簡素化されます。時系列データではなく数値データでアラートを実行する場合、ラベル付けされた各時系列を 1 つの数値に減らす必要はありません。代わりに、ラベル付き番号は 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 (ログ)、Azure Monitor (Azure リソースグラフ)、および Azure Data Explorer。

Grafana マネージドアラートまたは SSE を含むクエリは、次の場合、これらのデータソースでは数値と見なされます。

- データソースクエリでは、「Format AS」オプションが「Table」に設定されています。
- 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ます。

文字列列がある場合、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、各行はラベルを一意に識別する必要があります。

## 例

DiskSpace 「」 という MySQL テーブルの場合：

時間	ホスト	ディスク	PercentFree
2021-June-7	web1	/etc	3
2021-June-7	Web2	/var	4
2021-June-7	Web3	/var	8
...	...	...	...

データフィルタリングは時間どおりにクエリできますが、時系列を Grafana に返す必要はありません。例えば、空き領域が 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があるとします。

```
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)
```

このクエリは、Grafana に次のテーブルレスポンスを返します。

ホスト	ディスク	PercentFree
web1	/etc	3
Web2	/var	4
Web3	/var	0

このクエリをアラートルールの条件として使用すると、ゼロ以外の `PercentFree` が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
{ホスト=web1,ディスク=/etc}	アラート
{ホスト=web2,ディスク=/var}	アラート
{ホスト=web3、ディスク=/var}	[普通]

## ラベルと注釈

**△** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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す。名前付き値のセットですが、用途は異なります。ラベルの例、または同等の注釈は `alertname="test"`。

ラベルと注釈の主な違いは、ラベルは他のすべてのアラートとの区別に使用され、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ます。

例えば、2 つの高い CPU アラートを考えてみましょう。1 つは `server1`、もう 1 つは `server2`。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場合があります。ただし、`server` がサーバーの名前と CPU 使用率に置き換えられる場合など `"The CPU usage for server1 is above 75%."`、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 [ラベルと注釈のテンプレート化方法](#)については、「」のドキュメントを参照してください)。この種の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ています。ラベルの例は `server=server1`。各アラートには複数のラベルを付けることができ、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、アラートにラベルが設定されている `{alertname="High CPU usage",server="server1"}` のに対し、別のアラートにラベルが設定されている場合があります `{alertname="High CPU usage",server="server2"}`。ラベルは同じですが `alertname`、ラベルは異なるため、これらは 2 つの個別のアラート `server` 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、などの予約済みラベルの組み合わせです `alertname`。

## 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、アラートが発生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。カスタムラベルのテンプレート作成方法に関するドキュメントは、[で](#)入手できます。

テンプレートでカスタムラベルを使用する場合、ラベルの値がアラートルールの連続評価間で変更されないようにすることが重要です。これにより、多数の個別のアラートが生成されます。ただし、テンプレートがアラートごとに異なるラベル値を生成しても問題ありません。例えば、クエリの値をカスタムラベルに入れしないでください。値が変わるたびに新しいアラートセットが作成されます。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 つ以上含まれてい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、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`、`dashboardUIDpanelId` 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、アラートが発生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、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。注釈のテンプレート作成方法に関するドキュメントは、[こちら](#)で入手できます。

## トピック

- [ラベルマッチングの仕組み](#)
- [Grafana アラートのラベル](#)
- [ラベルと注釈のテンプレート化](#)

## ラベルマッチングの仕組み

**⚠** このドキュメントトピックは、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に設計されています。



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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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無音にリンクします。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無音に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。

- Label 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。
- Value 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。
- Operator フィールドは、ラベル値と照合する演算子です。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。
=~	値に正規表現が一致するラベルを選択します。
!~	値に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結合されます。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している必要があります。

## 例

アラートに次のラベルセットを定義した場合：

```
{ foo=bar, baz=qux, id=12 }
```

次に、次の操作を行います。

- 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foo=barと一致します。
- として定義されたラベルマッチャーfoo!=barがこのアラートルールと一致しません。
- 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id=~[0-9]+と一致します。
- 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baz!~[0-9]+と一致します。
- ととして定義された2つのラベルマッチャーはfoo=bar、このアラートルールid=~[0-9]+に一致します。

## ラベルを除外する

ラベルを除外するラベルマッチャーを記述することもできます。

ラベルを除外する方法の例を次に示しますteam。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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このトピックでは、ラベルがアラートの基本的な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 アラート内でアラートを一意に識別するものです。
- Alertmanager は、ラベルを使用して、通知ポリシー内の無音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はラベルにアクセスして、通知の結果であるアラートに固有の情報を含む通知を動的に生成できます。
- [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、ラベルが衝突した場合に優先されます (Grafana の予約ラベルの場合を除く。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 Alertmanager 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、アンダースコアのみを含める必要があります。正規表現と一致する必要があります `[a-zA-Z_][a-zA-Z0-9_]*`。無効な文字は、次のルールに従って外部アラートマネージャーに送信される前に、Grafana アラートエンジンによって削除または置き換えられます。

- 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、プレフィックスとして `_` が付けられます。

### 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、重複には元のラベルの短いハッシュがサフィックスとして追加されます。

## Grafana 予約ラベル

### Note

プレフィックスに `_grafana_` が付いたラベル `grafana_` は、Grafana が特別な使用のために予約しています。で始 `grafana_` まる手動で設定されたラベルを追加すると、衝突時に上書きされません。

Grafana の予約済みラベルは、手動で設定したラベルと同じ方法で使用できます。使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化

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。例えば、クエリの値に基づいてアラートの重要度ラベルを設定したり、サマリーアノテーションでクエリのインスタンスラベルを使用して、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。ラベルまたは注釈をテンプレート化しているかどうかにかかわらず、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に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サマリーアノテーションがあり、アラートルールに 10 個の発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。テンプレートで高額な計算を行わないようにする必要があります。

## 例

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書くのではなく、次の例では、テンプレートで見られた最も一般的なユースケースを表示しようとしています。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調整したりできます。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で区切って印刷する

すべてのラベルをカンマで区切って印刷するには、`$labels`変数を出力します。

```
{{ $labels }}
```

例えば、ラベル `alertname=High CPU usage`、`grafana_folder=CPU alerts` のアラートがあると `instance=server1`、次のよう出力されます。

```
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処理し、個別に印刷します。ここでは `$k`、`$v` の名前を参照し、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` のアラートがあると `instance=server1`、次のよう出力されます。

```
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

個々のラベルを出力するには、`$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、`instance=server1` というラベルのアラートがあると `instance=server1`、次のよう出力されます。

```
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。

### クエリの値を出力する

インスタントクエリの値を出力するには、`index`関数と `$values`変数を使用して Ref ID を出力します。

```
{{ 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、次のよう出力されます。

```
81.2345
```

範囲クエリの値を出力するには、まず、減少式を使用して時系列から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 `reduce` 式の結果を出力できます。例えば、Reduce 式が A の平均を取り、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出力するには、`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、次のよう出力されます。

```
81.234
```

範囲クエリのヒューマナイズされた値を出力するには、まず減少式を使用して時系列から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 `reduce` 式の結果を出力できます。例えば、Reduce 式が A の平均を取り、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出力する

インスタントクエリの値をパーセンテージで出力するには、`humanizePercentage`関数を使用します。

```
{{ humanizePercentage (index $values "A").Value }}
```

この関数は、値が 0~1 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 進数である場合は、クエリまたは数式を使用して 100 で除算できます。クエリが範囲クエリである場合は、まず、減少式を使用して時系列から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`ステートメントとより大きい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`$values`ため、と比較するときは必ず小数 (80.050.0、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、[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

クラシック条件を使用している場合、`$labels`を使用してクエリからラベルを印刷することはできません。`$values`代わりに `labels` を使用する必要があります。これは、従来の条件がこれらのラベルを破棄して一次元的な動作 (アラートルールごとに最大 1 つのアラート) を適用するためです。クラシック条件がこれらのラベルを破棄しなかった場合、多くの時系列を返したクエリでは、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、発砲と解決の間にアラートがフラッシングされます。

代わりに、`$values`変数には、発動しているすべての条件のすべての時系列の小さい値が含まれます。例えば、2 つの時系列を返すクエリ A と、2 つの条件を持つ従来の条件 B を持つアラートルールがある場合、`$values`には、`B0`、`B1 B2` が含まれます`B3`。クラシック条件 B に 1 つの条件しかない場合、`labels`には `B0`と `labels`のみ`$values`が含まれます`B1`。

すべての発射時系列のすべてのラベルを出力するには、次のテンプレートを使用します (正規表現 B の `labels` 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 つの時系列の従来の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、従来の条件を `reduce` 式と `Math` 式に置き換えます。



## クラシック条件のすべての値を出力する

クラシック条件のすべての値を出力するには、前の例に従って `$v.Labels` に置き換えます `$v.Value`。

```
{{ 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つの時系列の従来の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。

### labels 変数

`$labels` 変数には、クエリのすべてのラベルが含まれます。例えば、すべてのサーバーのCPU使用率を返すクエリがあり、いずれかのサーバーが過去5分間にCPU使用率の80%を超えたときに発生するアラートルールがあるとします。アラートに、CPU使用率が高いサーバーを示す概要アンテーションを追加します。`$labels` 変数を使用すると、次のような人間が読める文を出力するテンプレートを作成できます。

```
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「従来の条件のラベルをすべて印刷する」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、減少式と数式、および古典的な条件を含む文字列です。範囲クエリの結果は含まれません。範囲クエリは 10 から 10,000 行またはメトリクスまでの任意の場所を返す可能性があるためです。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 MBs のメモリが消費され、Grafana はメモリを非常に迅速に使い果たす可能性があります。

概要に `$value` 変数を出力するには、次のように記述します。

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes:
{{ $value }}
```

次のようになります。

```
CPU usage for instance1 has exceeded 80% for the last 5 minutes: [ var='A'
labels={instance=instance1} value=81.234 ]
```

ここでは `var='A'`、Ref ID A のインスタントクエリ `labels={instance=instance1}` を参照し、ラベルを参照し、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文字列の一部だけを出力する場合は、`$values` 変数を使用します。と同じ情報が含まれていますが `$value`、構造化テーブルには含まれており、目的のテキストだけに一致する正規表現を記述する方がはるかに簡単です。

## 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 IDs によってインデックス化されます。

参照 ID A を使用してインスタントクエリの値を出力するには :

```
CPU usage for {{ index $labels "instance" }} has exceeded 80% for the last 5 minutes:
{{ index $values "A" }}
```

例えば、ラベル付きのアラート `instance=server1` と値 のインスタントクエリがあると 81.2345、次のよう出力されます。

```
CPU usage for instance1 has exceeded 80% for the last 5 minutes: 81.2345
```

参照 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、参照 ID B を含む reduce 式を追加し、(index \$values "A")を に置き換えます(index \$values "B")。

```
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には、次の関数を使用できます。

## 引数

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[Grafana バージョン 10 でを試すの](#)のグラフィカル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}]
```

## ヒューマナイズ

`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 と似ています `humanize` 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、時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
```

一致

`match` 関数は、テキストを正規表現パターンと照合します。

```
{{ 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 [Grafana バージョン 10 でを試すの](#) 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!
```

`toLowerCase`

`toLowerCase` 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートルールは、アラートインスタンスが起動するかどうかを決定する一連の評価基準です。ルールは、1つ以上のクエリと式、条件、評価の頻度、および発音を開始するのに条件を満たす必要がある期間で構成されます。

クエリと式が評価するデータセットを選択する間、条件は、アラートを作成するためにデータが満たすか超える必要がある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。期間は、設定されている場合、条件を満たす必要がある期間を示します。アラートルールは、データがない場合のアラート動作を定義することもできます。

## 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](#)

## アラートルールタイプ

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana は、いくつか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の詳細、その仕組み、ユースケースに最適なアラートルールタイプを決定します。

## 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タイプです。これにより、既存のデータソースからのデータに作用でき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の場合：

- 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アラートを配信するように設定したり、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を追加する」](#)を参照してください。

## データソース管理ルール

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確認することで、データソースが Grafana による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：

- アラートルールはデータソース自体内に作成され、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間の次の比較を検討してください。

機能	Grafana が管理するアラートルール	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	はい	いいえ: 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



機能	Grafana が管理するアラートルール	Loki/Mimir マネージドアラートルール
		では、ルーラー API が有効になっている必要があります。
データソースを混在させて一致させる	はい	いいえ
記録ルールのサポートを含む	いいえ	はい
式を追加してデータを変換し、アラート条件を設定する	はい	いいえ
アラート通知でイメージを使用する	はい	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。	アラートルールをデータソース自体に保存し、「無限」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、外部アラートマネージャー、またはその両方を使用して Grafana 内から行われます。	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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

録画ルールは、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。

記録ルールを使用すると、頻繁に必要な式や計算コストの高い式を事前に計算し、その結果を新しい時系列の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、高速です。

Prometheus の記録[ルール](#)の詳細をご覧ください。

## クエリと条件

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana では、クエリは、MySQL や PostgreSQL などのデータベース、Prometheus、InfluxDB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。OpenSearch CloudWatch

サポートされているデータソースの詳細については、「」を参照してください[データソースと Grafana アラート](#)。

クエリを実行するプロセスには、データソースの定義、取得するデータの指定、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、これらのクエリの構築に使用されます。

アラートでは、測定するデータと、アラートルールが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、「」を参照してください[データのクエリと変換](#)。

## 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は、独自の機能を利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を提供します。

クエリ言語が異なるため、各データソースクエリエディタは異なる検索と機能を行います。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、または視覚的な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 使用率、ネットワークトラフィック、センサーの読み取りなど、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定義します。

フィルター – フィルターを適用して、特定のタグ、ホスト、アプリケーションによるデータのフィルタリングなど、特定の基準に基づいてデータを絞り込みます。

集計 – データに対して集計を実行して、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。

グループ化 – 特定のディメンションまたはタグでデータをグループ化して、集約された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をサポートしていません。詳細については、[Grafana Labs フォーラムを参照してください](#)。

## 式クエリ

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、カスタムメトリクスを作成したり、数学演算、関数、または論理式に基づいて既存のメトリクスを変更したりできます。

式クエリを活用することで、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、特定の時間範囲またはディメンションでのデータの集約、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 が管理するアラートルールにのみ式を使用できます。式ごとに、数式、縮小式、再サンプリング式から選択できます。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することもできます。これにより、条件が満たされたときに 1 つのアラートをトリガーするアラートルールが作成されます。その結果、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 は 1 つのアラートのみを送信します。

#### 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必要があります。

## 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。

## 数学

時系列および数値データに対して自由形式の数学関数/演算を実行します。時系列データの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。

## 再サンプリング

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さまざまなデータソースの時系列データを比較する場合に便利です。

## 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 false 0 の場合と、条件が true 1 の場合に が返されます。次のしきい値関数を使用できます。

- 以上 ( $x > y$ )
- 以下 ( $x < y$ )
- 範囲内にある ( $x > y1$  AND  $x < y2$ )
- 範囲外 ( $x < y1$  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

Classic 条件式クエリは、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必要があります。

## 集計

Grafana Alerting には、クエリをさらに絞り込むための以下の集計関数が用意されています。

これらの関数は、Reduce 条件式と Classic 条件式でのみ使用できます。

機能	式	その内容
avg	削減/クラシック	値の平均を表示する
min	削減/クラシック	最小値を表示する
max	削減/クラシック	最大値を表示する
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	最後の値を表示する
中央値	削減/クラシック	中央値を表示します。
差分	Classic	最新の値と最も古い値の差を表示します。
diff_abs	Classic	差分の絶対値を表示する
percent_diff	Classic	最も新しい値と最も古い値の差のパーセンテージ値を表示します。

機能	式	その内容
percent_diff_abs	Classic	percent_diff の絶対値を表示する
count_non_null	Classic	結果セットにない値の数を表示する null

## アラート条件

アラート条件は、アラートが生成される値に応じてアラートを起動するかどうかを決定するクエリまたは式です。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、アラートルール条件としてクエリや式のいずれかを選択します。

クエリされたデータが定義された条件を満たすと、Grafana は関連するアラートをトリガーします。アラートは、E メール、Slack、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 PagerDuty。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

フラッピングアラートのノイズを減らすために、アラートのしきい値とは異なる復旧しきい値を設定できます。

フラッピングアラートは、メトリクスがアラートのしきい値条件にカーソルを合わせると発生し、頻繁な状態変化を引き起こし、生成される通知が多すぎる可能性があります。

Grafana が管理するアラートルールは、特定の期間にわたって評価されます。各評価では、クエリの結果がアラートルールで設定されたしきい値と照合されます。メトリクスの値がしきい値を超えると、アラートルールが起動し、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、アラートは解決され、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、メトリクスの値がしきい値を継続的に上回ったり下回ったりする場合があります。これはフラッピングと呼ばれ、一連の発音 - 解決 - 発音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 ms のレイテンシーのアラートがあり、その数が約 1000 (980 ->1010 ->990 -> 1020 など) 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つのしきい値があることを意味します。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砲を停止します。

## アラートインスタンス

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana マネージドアラートは、多次元アラートをサポートしています。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、最初の評価後に観測する CPUs の量と同じ数のアラートインスタンスを作成し、1つのルールで各 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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートは、Grafana マネージドルールフォルダと、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コントロールを実行し、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 Mimir 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、グループの評価間隔を編集したりできます。

### Groups (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 ルール名前空間または Loki ルール名前空間とグループの名前を変更したり、グループの評価間隔を編集したりできます。


#### Tip

異なる間隔で同時にルールを評価する場合は、異なるグループにルールを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、順次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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。



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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートルール評価を使用して、アラートルールを評価する頻度と、その状態を変更する速度を決定します。

これを行うには、アラートルールが正しい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ています。

同じグループ内のデータソース管理の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が評価されるようにする場合に特に便利です。

Grafana が管理するアラートルールは、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、クロック上の最も近い 10 秒のウィンドウ、例えば 10:00:00、10:00:10、10:00:20 などに 10 秒ごとに評価されます。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

通知ポリシーでの評価グループとアラートのグループ化は、2 つの異なるものです。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間メッセージで送信できます。

## 保留中の期間

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、アラートルールが起動するまで条件に違反している可能性がある期間を選択します。

## 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ません。

[01:00] 2 回目の評価 - 条件違反。保留中のカウンターがスタートします。アラートは保留中にスタートします。

[01:30] 3 回目の評価 - 条件違反。保留中のカウンター = 30 秒。保留中の状態。

[02:00] 4 回目の評価 - 条件違反。保留中のカウンター = 60 秒保留中の状態。

[02:30] 第 5 評価 - 条件違反。保留中のカウンター = 90 秒。アラートが発射を開始する

アラートルールに、アクションを実行する前に一定期間違反する必要がある条件がある場合、その状態は次のように変化します。


- 条件が最初に超過すると、ルールは「保留中」状態になります。
- ルールは、必要な期間 - 保留期間にわたって条件が壊れるまで、「保留」状態のままになります。
- 必要な時間が経過すると、ルールは「ファイリング」状態になります。
- 保留期間中に条件が壊れなくなった場合、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 に設定するだけです。これにより、保留期間が実質的にスキップされ、条件に違反するとすぐにアラートルールが発動し始めます。

アラートルールが発生すると、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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、アラートルールの状態の3つの主要コンポーネントがあります。関連していますが、各コンポーネントは微妙に異なる情報を伝えます。

### 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	評価エンジンによって返される時系列は少なくとも 1 つ です Pending。
ファイリング	評価エンジンによって返される時系列は少なくとも 1 つ です Firing。

#### Note

アラートは最初に `pending` に移行し、次に `firing` に移行するため、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]	発音も保留もしないアラートの状態。すべてが正しく動作しています。
保留中	設定されたしきい値の期間未満にアクティブだったアラートの状態。

状態	説明
アラート	設定されたしきい値の期間よりも長くアクティブだ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ていません。
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または状態が発生したときに最後のError状態を維持するように設定できます。これにより、アラートの発音と、解決と再発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Firingすると Pending から に移行します。

### アラートルールの正常性

アラートルールには、次のいずれかのヘルスステータスがあります。

状態	説明
没頭	アラートルールを評価するときにエラーは発生しません。
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	ルール評価中に少なくとも 1 つの時系列にデータがないこと。

### NoDataおよび の特別なアラート **Error**

アラートルールを評価すると、状態 NoDataまたは が生成されます。Grafana アラートはError、次の追加ラベルを持つアラートインスタンスを生成します。

ラベル	説明
アラート名	状態 <code>DatasourceError</code> に応じて <code>DatasourceNoData</code> または <code>のいずれか</code> 。
<code>datasource_uid</code>	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、無音の追加、コンタクトポイントへのルーティングなどによって、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

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

連絡先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 として評価されます (エスケープに影響する可能性があります)。

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です。

ほとんどのコンタクトポイント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できます。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、「」を参照してください [通知をカスタマイズする](#)。

## ネストされたテンプレート

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" . }}
```

カスタムテンプレートを埋め込むには、次の組み込みテンプレートオプションのいずれかを使用できます。

名前	メモ
<code>default.title</code>	高レベルのステータス情報を表示します。
<code>default.message</code>	発声アラートと解決済みアラートのフォーマットされた概要を提供します。
<code>teams.default.message</code>	に似ており <code>default.message</code> 、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 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 `<grafana-install-dir>/public/emails/ng_alert_notification.html`。このファイルを編集して、すべてのアラート E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に設計されています。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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Alertmanager 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、選択したチャン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 Alertmanager または外部の Alertmanager 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が生成される場所によって異なります。

### Grafana アラートマネージャー

Grafana Alertmanager は、事前設定されており、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 Alertmanager は Grafana からアラートを受信できますが、例えば、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

抑制ルールは Grafana Alertmanager 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

単一のアラートマネージャーを使用して Grafana、Loki、Mimir、Prometheus のすべてのアラートを受信する場合は、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 自体内から設定および管理できます。

Grafana Alertmanager 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、設定した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があり、引き続き使用したいと考えています。
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と考えています。

アラートマネージャーは、アラートの連絡先ページと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  
ドhandleGrafanaManagedAlertsのフラグを に設定trueして、Grafana マネージドアラートを  
このアラートマネージャーに送信します。

## 連絡先

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

連絡先には、通知を送信するための設定が含まれています。連絡先は統合のリストで、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持つことができます。例えば、1つの連絡先に 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。その場合、通知は送信されません。

アクセスポイントは、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つの連絡先にのみアラートを送信できますが、1つの連絡先を複数の通知ポリシーに同時に追加できます。アラートが通知ポリシーに一致すると、アラートはその通知ポリシーの連絡先に送信され、その後、その設定の各統合に通知が送信されます。

Grafana Alertmanager 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タイプを示します。



名前	型
Amazon SNS	sns
OpsGenie	opsgenie
ページャーの義務	pagerduty
Slack	slack
VictorOps	victorops

連絡先の詳細については、[コンタクトポイントを設定する](#)「」および「」を参照してください。[通知をカスタマイズする](#)。

## 通知

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください。[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください。[Grafana バージョン 8 での作業](#)。

アラート通知を送信する方法、タイミング、場所の選択は、アラートシステムを設定する上で重要な部分です。これらの決定は、問題をすばやく解決する能力に直接影響し、重要なことを見逃さないようになります。

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[連絡先](#)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問い合わせポイントに追加して、通知に再利用と一貫性のある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、および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、アラート通知の送信先を定義します。

## 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、発音と解決されたアラート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 Alertmanager がありますが、[Prometheus Alertmanager などの他の Alertmanager](#) からの通知の送信もサポートしています。Grafana Alertmanager は、通知ポリシーと連絡先を使用して、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、通知が送信されるタイミングと場所を制御します。通知ポリシーは、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したり、1 日の特定時刻および特定の曜日に通知を禁止する 1 つ以上のミュートタイミングを設定したりできます。

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に整理されます。デフォルトポリシーは 1 つだけ存在でき、デフォルトポリシーは削除できません。

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、アラートに一致します。

ネストされたポリシーには、アラートの追加マッチングを可能にする独自のネストされたポリシーを含めることができます。ネストされたポリシーの例としては、インフラストラクチャアラートを Ops チームに送信することが挙げられます。一方、子ポリシーは優先度の高いアラートを Pagerduty に送信し、優先度の低いアラートを Slack に送信する場合があります。

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を調べ、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を送信します。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 つとアラートの照合を試みることができます。アラートに一致するネストされたポリシーがない場合、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か、アラートに一致するネストされたポリシーがない場合、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、「」を参照してください[通知ポリシー](#)。

## 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連絡先に限定されず、同じ連絡先内の多数の統合や、異なる連絡先間の統合でも使用できます。例えば、Grafana ユーザーは `というテンプレートを作成しcustom_subject_or_title`、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、[Go のテンプレート言語](#) で記述され、アラートページのコンタクトポイントタブにあります。

通知のカスタマイズの詳細については、「」を参照してください[通知をカスタマイズする](#)。

### 無音

無音を使用して、1 つ以上の発音ルールからの通知をミュートできます。サイレンスは、アラートの発音や解決を停止したり、ユーザーインターフェイスでアラートの発音を非表示にしたりしません。無音は、分、時間、日、月、または年で設定できる期間だけ続きます。

無音の使用の詳細については、「」を参照してください[Prometheus データソースのアラート通知のサイレンシング](#)。

### 通知ポリシー

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知ポリシーは、さまざまなレシーバーにアラートをルーティングする柔軟な方法を提供します。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成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、各ポリシーには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、処理に関心のあるラベルまたは関心の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。

ラベルマッチングの詳細については、「」を参照してください[ラベルマッチングの仕組み](#)。

### 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マッチングを継続を有効にしていない限り、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判断するには、デフォルトの通知ポリシーから始めて、既存の通知ポリシーのセットを見ることから始める必要があります。

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、デフォルトポリシーは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、それらの通知ポリシーは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、その子ポリシーに降順になり、存在する場合、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が引き続き検索され、その後、子ポリシーが見つからないようになります。

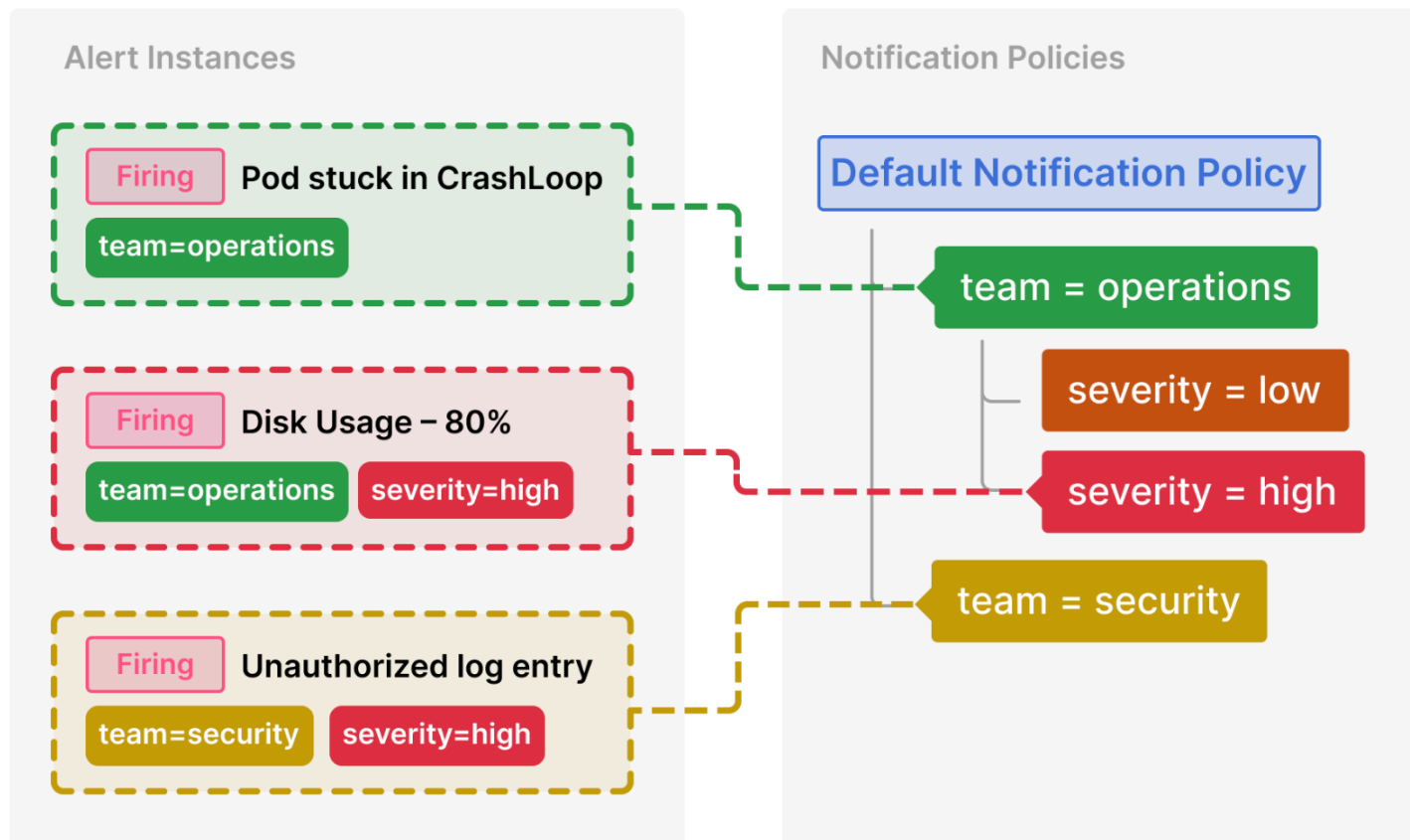
通知ポリシーで子ポリシーが定義されていない場合、またはアラートインスタンスのラベルに一致するラベルマッチャーが子ポリシーのいずれにもない場合、親通知ポリシーが使用されます。

一致するポリシーが見つかりとすぐに、システムは他の一致するポリシーを引き続き検索しません。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「兄弟のマッチングを続ける」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は次のとおりです。

でスタックしたポッド CrashLoop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、一致する兄弟の継続がそのteam=securityポリシーに対して設定されていないため、ポリシーは評価されません。

ディスク使用状況 - 80% には teamと の両方のseverity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、値が同じではないため、最初のポリシー (team=operations) と一致しないため、検索とteam=securityポリシーの一致が続行されます。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

アラートインスタンスをルーティングするための有用な概念である子ポリシーに加えて、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。

- 連絡先
- 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

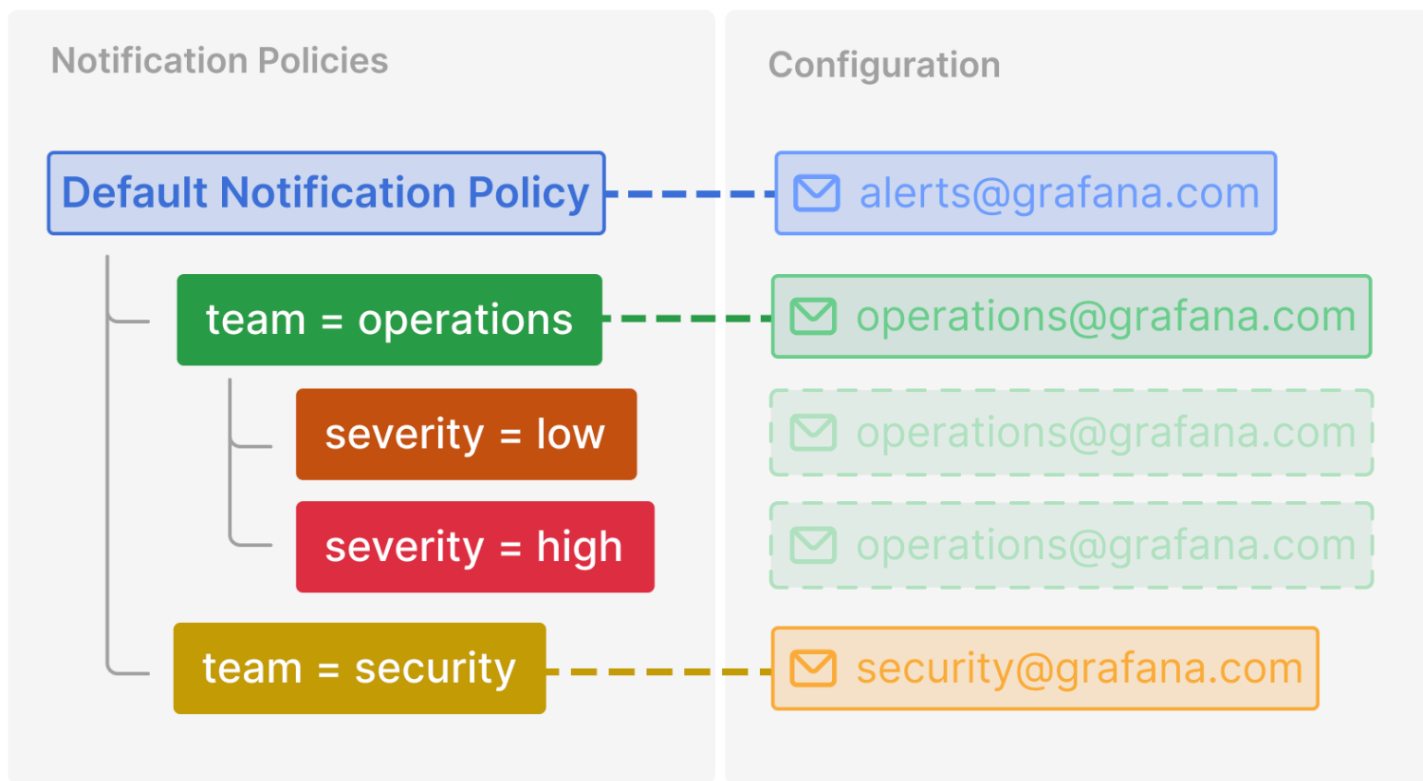
継承されたプロパティを上書きする場合は、これらのプロパティをそれぞれ個別のポリシーで上書きできます。

親ポリシーから連絡先を継承するには、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を上書きする を有効にします。

## 継承の例

以下の例は、前の例の通知ポリシーツリーで、の子ポリシーがコンタクトセンターteam=operationsを継承することを許可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がなくなります。



## 追加設定のオプション

### グループ分け

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少数の通知にまとめることができます。これは、エンジニアなどのファーストレスポnderに通知が配信され、短期間に多数の通知を受信すると圧倒され、場合によってはファーストレスポnderが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る場合に特に重要です。例えば、システムの多くが停止している大規模な停止を考えてみましょう。この場合、グループ化は、1 回の電話の受信と 100 回の電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化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アラートをまとめます (アラート名は複数のフォルダ間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化する場合は、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スではなく自動システムに配信されている場合に便利です。

## すべてのアラートに1つのグループ

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、Group by emptyのままにしておくことができます。

## 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を送信する頻度を決定します。グループ待機、グループ間隔、繰り返し間隔の3つのタイマーについて知っておく必要があります。

### 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前に Grafana が待機する時間です。グループの待機時間が長いほど、他のアラートが届くまでの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必要があります。

### 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、Grafana が変更に関する通知をグループに送信するまでに待機する時間です。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ている可能性があります。グループの待機が原因でアラートを最初の通知に含めるのが遅すぎた場合、グループ間隔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、グループが前回の通知以降に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動していることを思い出させるもの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なければならないということです。繰り返し間隔がグループ間隔の倍数でない場合、1つに強制されます。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5の最も近い倍数である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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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にまたがる複数のインスタンスの実行など、高可用性を実現するように設定されています。

Grafana Alerting は、アラートルールの評価と通知の配信を分離する Prometheus モデルを使用します。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られます。これは、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する方法です。この重複は状態履歴に表示されるため、高可用性を使用しているかどうかを確認するのに適しています。

アラートの設定

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、次の操作を行う必要があります。

- [データソース](#) を設定します。
- 選択したデータソースが互換性があり、[Grafana アラート](#) で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 マネージド型またはデータソースマネージド型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。
  - デフォルトの連絡先を確認し、システムの連絡先を更新します。
  - 必要に応じて、新しい連絡先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 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。
  - オプションで、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します。

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 管理のアラートを Grafana の外部にルーティングするなど、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

- [クラシック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](#)

## クラシック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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana アラートを使用しないワークスペースは、[Classic 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、この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、「」を参照してください[Grafana アラートとレガシーダッシュボードアラート](#)。

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 は Grafana アラートのプレビューを表示し、アップグレードが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。

### **i** Note

Grafana アラートを使用する場合、Prometheus ではなく Grafana で定義されたアラートルールは、複数の通知を連絡先に送信します。ネイティブの Grafana アラートを使用している場合は、従来のダッシュボードアラートを維持し、新しい Grafana アラート機能を有効にしない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、Prometheus Alertmanager で作成されたアラートの通知を 1 つだけ送信する Grafana アラートを有効にすることをお勧めします。

## Grafana アラートのプレビュー

移行する前に 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、アラートを変更して移行を変更できます。

Grafana アラート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、移行後にアラートがどのようになるかを確認できます。

#### 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。行った変更を元に戻すには、アップグレードページの右上にあるアップグレードの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なら、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 アラートのオンと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、「」を参照してください [Amazon Managed Grafana ワークスペースを設定する](#)。

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。Grafana インスタンスのアラートページで、移行されたアラートを新しいアラートとともに表示できます。Grafana アラートを使用すると、Grafana が管理するアラートルールは、一致すると 1 つのアラートではなく、複数の通知を送信します。

クラシックダッシュボードアラートと Grafana アラートへの読み取り/書き込みアクセスは、それらを保存するフォルダのアクセス許可によって管理されます。移行中、クラシックダッシュボードのアラート許可は、次のように新しいルール許可と一致します。

- 元のアラートのダッシュボードに アクセス許可がある場合、移行では、元のダッシュボードのアクセス許可 (フォルダから継承されたアクセス許可を含む) と一致する Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1} のように、この形式で という名前のフォルダが作成されます。
- ダッシュボードのアクセス許可がなく、ダッシュボードがフォルダにある場合、ルールはこのフォルダにリンクされ、そのアクセス許可を継承します。
- ダッシュボードのアクセス許可がなく、ダッシュボードが General フォルダにある場合、ルールは General Alerting フォルダにリンクされ、ルールはデフォルトのアクセス許可を継承します。

**Note**

Grafana アラートNoDataには Keep Last Stateのオプションがないため、このオプションは従来のルールの移行NoData中になくなります。Error 処理Keep Last Stateのオプションは、新しいオプションに移行されますError。の動作に一致させるためにKeep Last State、どちらの場合も、移行中に Amazon Managed Grafana は 1 年間の期間を持つアラートルールごとに自動的に無音を作成します。

通知チャンネルは、適切なルートとレシーバーを持つ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ンネルは、デフォルトのルートへの問い合わせポイントとして追加されます。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ンネルは、autogen-unlinked-channel-recvルートに送信されます。

**制限事項**

- Grafana アラートシステムは、使用可能なすべての Prometheus、Loki、Alertmanager データソースからルールを取得できます。サポートされている他の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行き来すると、1 つのシステムで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。もう 1 つのシステムでサポートされる機能は失われない可能性があります。

**Note**

クラシックダッシュボードアラートに戻ると、Grafana アラートを有効にしている間に行われたアラート設定に加えられたすべての変更が失われます。これには、作成された新しいアラートルールも含まれます。

**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 アラートには、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あります。

**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。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を生成します。例えば、1 つのホスト上

の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、各時系列はラベルセットで識別されます。

## Dashboards の外部でアラートを作成する

従来のダッシュボードアラートとは異なり、Grafana アラートを使用すると、複数のソースからのデータを一意の方法で組み合わせるクエリと式を作成できます。引き続き、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、監視対象のシステムを迅速にトラブルシューティングできます。

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 E メールにイメージやクエリ値は含まれません。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 と Grafana Mimir のアラートルールを作成する

Grafana アラートでは、Grafana マネージドアラートと同じ UI と API を使用して Loki および Grafana Mimir アラートルールを管理できます。

## 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一覧表示されるようになりました。複数のデータソースのラベルを検索して、関連するアラートをすばやく見つけることができます。

## アラート状態 NoData とエラーの特別なアラート

Grafana Alerting は、アラート状態の新しい概念を導入しました。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、次のラベルを持つ特別なアラートを生成します。

- `alertname` 状態 `DatasourceError` に応じて `DatasourceNoData` または の値を指定します。
- `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、無音の追加、コンタクトポイントへのルーティングなどによって、通常のアラートと同じ方法で処理できます。

**Note**

ルールが多数のデータソースを使用し、1つ以上の **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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 Grafana を設定します。この外部アラートマネージャーは、Grafana 自体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、無音、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、URL で外部アラートマネージャーを設定するときに以前に表示されていた HTTP 基本認証情報も暗号化できます。

**Note**

Grafana 9.2 以降、アラートページの管理者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。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. 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フィールドでhandleGrafanaManagedAlertsのフラグを に設定trueして、Grafana マネージドアラートをこのアラートマネージャーに送信します。

**Note**

Alertmanager 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 の問い合わせポイントと通知ポリシーは読み取り専用です。

5. 保存と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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。



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が異なる場所に常駐しています。これを複数のチームや組織にまたがってスケールアップすることは、特に困難な作業です。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、保守できるため、このプロセスが簡単になります。

次の 2 つのオプションから選択できます。

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、API でプロビジョニングされたアラートルールを Grafana UI から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、次のリクエストに `x-disable-provenance` ヘッダーを追加します。

```
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、テンプレートをサポートしています。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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、保守できます。

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](#) の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. を実行して terraform apply、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 以降を使用していることを確認します。
- 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 キーを作成する

[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 アラートサポートで自動的に機能する必要があります。Terraform で使用するキーの作成の詳細については、「[Amazon Managed Grafana オートメーションに Terraform を使用する](#)」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「アラートルールのプロビジョニング API へのアクセス」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確認するには、Terraform ドキュメントの「[Grafana 認証](#)」を参照してください。

## Terraform プロバイダーを設定する

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>
}
```

## 連絡先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。外部システムに接続する方法と通知を配信する場所を Grafana に伝えます。15 種類以上の[統合](#)から選択できます。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 ウェブフック URL (またはその他の連絡先の詳細) に置き換えます。

この例では、アラート通知を Slack に送信する連絡先を作成します。

```
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 スタイルのテンプレート](#) をサポートします。これにより、Grafana アラート通知テンプレートを Terraform で直接管理できます。

3. terraform apply コマンドを実行します。
4. Grafana UI に移動し、連絡先の詳細を確認します。

UI から Terraform を介して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動作することを確認します。

#### Note

同じテンプレートを多くのコンタクトポイントで再利用できます。上記の例では、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。{{ template "Alert Instance Template" . }}

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
}
```

## プロビジョニング通知ポリシーとルーティング

通知ポリシーは、アラートインスタンスをルーティングする方法を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、以前に定義したコンタクトポイントに発声アラートを接続します。

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、アラートはグループ化されています。つまりalertname、同じ名前のアラートから送信される通知はすべて、同じ Slack メッセージにグループ化されます。

特定の通知を異なる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さまざまな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 = ["..."]
}
}
```

- 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。
- terraform apply コマンドを実行します。
- Grafana UI に移動し、通知ポリシーの詳細を確認します。

#### Note

UI から Terraform 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。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。

- テスト をクリックして、通知ポイントが正しく動作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は、定義された期間のアラート通知をミュートする機能を提供します。

ミュートタイミングをプロビジョニングするには

- 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され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

UI から Terraform 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。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動作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、Grafana データソースに対してアラートを送信できます。これは、既に設定したデータソースでも、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. クエリを実行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、[テスト用の TestData データソースを設定する](#)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動しているかどうかを確認できます。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、Slack をコンタクトポイントとして選択した場合、Grafana の埋め込み [Alertmanager](#) は自動的にメッセージを Slack 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。

プロビジョニングされたリソースにはプロビジョニングされた というラベルが付けられるため、手動で作成されていないことがわかります。

### **i** 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り、ホットリロードが実行されたりした場合に上書きされるリソースに変更が加えられるのを防ぐことができます。

## アラートの設定

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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](#)
- [録画ルールを設定する](#)
- [コンタクトポイントを設定する](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana マネージドルールは、最も柔軟なアラートルールタイプです。これにより、サポートされている任意のデータソースからのデータに作用でき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。アラート通知でのイメージの使用もサポートされています。これは、1つ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タイプのルールです。

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。

次の手順では、Grafana が管理する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 が管理する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て、以下のステップに従います。

### アラートルール名の設定

1. アラートと IRM -> 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` ラベルでもあります。

次に、測定するデータと、アラートルールが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。

### クエリと条件を定義するには

1. データソースを選択します。
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、などの固定相対時間範囲のみをサポートします `now-24hr:`  
`now`。

絶対時間範囲 `2021-12-02 00:00:00 to 2021-12-05 23:59:59` または半相対時間範囲 はサポートされていません `now/d to: now`。

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 によって管理されます。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、データソース管理の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 Classic 条件を選択して単一のアラートルールを作成するか、数学、削減、再サンプリングの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生成します。

#### Note

Prometheus を使用する場合、インスタントベクトル関数と組み込み関数を使用できるため、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切り替えをオンにし、アラートルールが発動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. アラート条件として設定するクエリまたは式で、アラート条件として設定を選択します。

アラートルール評価を使用して、アラートルールを評価する頻度と、その状態を変更する速度を決定します。

これを行うには、アラートルールが正しい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を設定するには

1. フォルダを選択するか、+ 新規フォルダ 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」をクリックします。

新しい評価グループを作成する場合は、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、アラートルールが起動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である場合、アラートは Firing 状態に移行し、アクティブでない場合は Normal 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. 「データとエラー処理なしの設定」で、データがない場合のアラート動作を設定します。

このセクションで後述するガイドラインを使用してください。

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。

すべてのアラートルールとインスタンスは、ラベルに関係なく、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、またはアラートルールまたはアラートインスタンスのラベルに一致するネストされたポリシーがない場合、デフォルトの通知ポリシーは一致するポリシーになります。

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場合は、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る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。

アラート通知メッセージでアラートに関するコンテキストをさらに提供するための[注釈](#)を追加します。

アノテーションはメタデータを追加して、アラート通知メッセージ内のアラートに関する詳細情報を提供します。例えば、概要アノテーションを追加して、アラートが起動した値や発生したサーバーを示します。

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する

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. [Save rule] (ルールを保存) を選択します。

## 単一および多次元ルール

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

クラシック条件式を使用して、条件が満たされたときに単一のアラートをトリガー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。その結果、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 は 1 つのアラートのみを送信します。

### • 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)。
- ルールの条件を持つMath式を追加します。クエリまたは reduce 式がルールが発せられない場合は、発せられる場合は正の数値をすでに返す場合は必要ありません。例: B のクエリ/式が 70 を超える場合に  $B > 70$  を起動  $B < C * 100$  する場合。B の値が C の値に 100 を掛けた値より小さい場合に を起動する場合。比較される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、一方が他のクエリのサブセットである場合、それらが一致されます。

### Note

Grafana は、テンプレート変数を使用したアラートクエリを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。

## データおよびエラー処理を設定しない

アラートルールの評価でデータまたはエラーが返されない場合のアラート動作を設定します。

### Note

評価がデータが返されない場合、またはエラーが発生した場合に起動するように設定されたアラートルールは、評価期間全体が終了した場合にのみ起動されます。つまり、アラートルール条件に違反したときにすぐに発せられるのではなく、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経過するまで待機してから発せられるため、アラートノイズが減少し、一時的なデータ可用性の問題が発生します。

アラートルールの評価でデータが返されない場合は、アラートルールの状態を次のように設定できます。

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



データなし	説明
	DatasourceNoData を含む新しいアラートを作成します。
アラート	アラートルールの状態を に設定しますAlerting。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ってから発射します。
沖	アラートルールの状態を に設定しますNormal。

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 DatasourceError を持つアラートインスタンスを作成します。
アラート	アラートルールの状態を に設定しますAlerting。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ってから発射します。
沖	アラートルールの状態を に設定しますNormal。

### 古いアラートインスタンスの解決

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 状態の古いアラートインスタンスは自動的に とマークResolvedされ、理由 でgrafana\_state\_reason注釈がアラートインスタンスに追加されますMissingSeries。

## 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、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ルーラー 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。

#### **i** 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許可があることを確認します。それ以外の場合、Grafana Mimir マネージド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は、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 local である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 の両方をサポートしており、Grafana は [Query API](#) と [Ruler API](#) の両方が同じ URL にあることを想定しています。ルーラー 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たくない場合は、その設定に移動し、アラート UI によるアラートの管理チェックボックスをオフにします。

次の手順では、データソース管理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、以下のステップに従います。

アラートルール名を設定するには

1. アラートと IRM -> 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` ラベルでもあります。

クエリを定義して、測定するデータと、アラートルールが発生する前に満たす必要がある条件を取得します。

クエリと条件を定義するには

1. すべてのアラートルールは、デフォルトで Grafana によって管理されます。データソース管理のアラートルールに切り替えるには、データソース管理の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します。

Open advanced data source picker を選択して、データソースの追加 (管理者のみ) 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。

アラートルール評価を使用して、アラートルールを評価する頻度と、その状態を変更する速度を決定します。

アラート評価の動作を設定するには

### 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場合は、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. 保留中の期間を入力します。

保留期間は、アラートルールが起動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 Firing状態に移行し、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。

すべてのアラートルールとインスタンスは、ラベルに関係なく、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か、アラートルールまたはアラートインスタンスのラベルに一致するネストされたポリシーがない場合、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場合は、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、アラート通知のアラートに関するコンテキストを追加します。

アノテーションはメタデータを追加して、アラート通知のアラートに関する詳細情報を提供します。例えば、どの値でアラートが起動したか、どのサーバーで発生したかを示す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する

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. [Save rule] (ルールを保存) を選択します。

録画ルールを設定する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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できます。記録ルールは、頻繁に必要な式または計算コストの高い式を事前に計算し、その結果を新しい時系列の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、高速です。

**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許可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は、それぞれのサービスを設定してルーラー API を有効にします。
- Loki - Loki データソースのデフォルト local である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ます。ルールを編集するには、他の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 の両方をサポートしており、Grafana は Query API と Ruler API の両方が同じ URL にあることを想定しています。ルーラー 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たくない場合は、その設定に移動し、アラート UI によるアラートの管理チェックボックスをオフにします。

**録画ルールを作成するには**

1. Grafana コンソールの Grafana メニューで、アラートと IRM、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、空白が含まれていない必要があります。

**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、組織的な目的のみを持つことができます。
- 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、新しいグループを追加します。新しく作成されたルールが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する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を設定する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。

#### **i** Note

Grafana マネージドアラートのコンタクトポイントを作成および編集できます。データソースマネージドアラートの連絡先は読み取り専用です。

### コンタクトポイントの使用

次の手順は、連絡先を追加、編集、削除、テストする方法を示しています。

## コンタクトポイントを追加するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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. 連絡先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 チャンネルと、連絡が必要な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使用可能な場合は、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、アラートが解決されたときに通知されたくない場合は、オプションで解決済みメッセージの無効化を選択します。
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手順を繰り返します。
10. 変更を保存します。

## コンタクトポイントを編集するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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、既存の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。

通知ポリシーで使用されていないコンタクトポイントは削除できます。

## コンタクトポイントを削除するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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、既存の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、さらに、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**

通知ポリシーでコンタクトポイントが使用されている場合は、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。

連絡先が作成されたら、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できます。

テスト通知を送信するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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、既存の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、新しい連絡先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カスタム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でアラートをテストします。

### コンタクトポイント統合を設定する

アラートルールが発せられたときに通知を受信するための任意の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。

設定したら、統合を連絡先の一部として使用して、アラートの状態が変わ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、を PagerDuty 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、リアルタイムのアラートを受信し始め、モニタリングデータを把握できます。

### 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タイプを示します。

名前	型
Amazon SNS	sns
OpsGenie	opsgenie
ページャーの義務	pagerduty
Slack	slack
VictorOps	victorops

## アラート PagerDuty 用の の設定

を設定するには PagerDuty、統合キーを指定する必要があります。以下の詳細を入力します。

設定	説明
統合キー	の統合キー PagerDuty
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は、デフォルトで使用されているものとマージされます。

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によってデフォルトのキーが上書きされます。

## 通知ポリシーを設定する

**△** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知ポリシーは、アラートをコンタクトポイントにルーティングする方法を決定します。

ポリシーにはツリー構造があり、各ポリシーには 1 つ以上のネストされたポリシーを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することもできます。

各アラートは、デフォルトのポリシーによって評価され、その後に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 つ以上の一致後も評価が続行されます。親ポリシーの設定と連絡先情報は、子ポリシーのいずれにも一致しないアラートの動作を制御します。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。

通知ポリシーの詳細については、「」を参照してください[通知](#)。

次の手順は、通知ポリシーを作成および管理する方法を示しています。

デフォルトの通知ポリシーを編集するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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、「...」、「の編集」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を更新します。
6. Group by で、アラートをグループ化するラベルを選択します。このポリシーで複数のアラートが一致する場合、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、すべての通知が 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 (これにより、効果的にグループ化が無効になります)。
7. タイミングオプション で、次のオプションから選択します。
  - グループ待機 — 最初の通知を送信する前に、同じグループのアラートをバッファするのを待機する時間。デフォルト値は 30 秒です。
  - グループ間隔 — グループに対する 2 つの通知間の最小時間間隔。デフォルトは 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ていない場合、通知を再送信するまでの最小間隔。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、そのツリー構造に従う必要があります。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。各ブランチは、独自のネストされたポリシーを持つことができます。このため、デフォルトのポリシーまたは既にネストされている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。

ネストされた新しいポリシーを追加するには

1. 左側のメニューで、アラート管理ポータル of IRM を選択し、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ポリシーに eitehr) 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ルールを 1 つ以上追加します。
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に一致した後でも、後続の兄弟ノードのマッチングを継続を有効にして、兄弟ポリシーのマッチングを継続します。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受け取ることができます。
8. 必要に応じて、上書きグループ化を有効にして、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. 必要に応じて、Override general timings を有効にして、グループ通知ポリシーで設定されたタイミングオプションをオーバーライドします。
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、アラート管理ポータル of IRM を選択し、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、...、編集 を選択します。
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名前の一部または全体を入力します。その連絡先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、一致者による検索入力フィールドに有効なラベル一致者を入力します。複数のマッチャーをカンマで区切って入力できます。例えば、有効なマッチャー入力は `severity=high, region=~EMEA|NA`。

#### Note

ラベルで検索する場合、一致するポリシーはすべて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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートルール、連絡先、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

- [通知をカスタマイズする](#)
- [連絡先の管理](#)
- [Prometheus データソースの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](#)
- [アラートグループによる表示とフィルタリング](#)
- [通知エラーの表示](#)

### 通知をカスタマイズする

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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して、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
- 太字と斜体のテキストをフォーマットし、改行を追加または削除します。

通知テンプレートを次の目的で使用することはできません。

- 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の通知の設計を変更する

## トピック

- [Go のテンプレート言語の使用](#)
- [通知テンプレートを作成する](#)
- [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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知テンプレートは、Go のテンプレート言語である[テキスト/テンプレート](#) で記述します。

このセクションでは、Go のテンプレート言語の概要と、テキスト/テンプレートでのテンプレートの記述について説明します。

## ドット

テキスト/テンプレートにはドット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、テンプレートのどこで使用されるかによっ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`、`ExtendedData` オブジェクト、`StatusGroupLabelsCommonLabelsCommonAnnotations`、`ExternalURL`。ただし、ドットは、リスト `range` 上で使用する場合、内で使用する場合、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に `with`、別のものを参照することがあります。この例については、[通知テンプレートを作成する](#)「」、およびすべてのデータと関数については「」を参照してください [テンプレートリファレンス](#)。

## タグの開閉

テキスト/テンプレートでは、テンプレートが変数を出力するか、`if` ステートメントなどの制御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、変数の印刷で `and` を使用し、制御構造で `{%}` と `{%end%}` 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%}`。

## [Print]

何かの値を出力するには、`{%と 使用します%}`。ドットの値、ドットのフィールド、関数の結果、および [変数](#) の値を印刷できます。例えば、ドットが参照する `Alerts` フィールドを出力するには `ExtendedData`、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 `range` 反復処理できます。

```
{{ range .Alerts }}
  {{ .Labels }}
{{ end }}
```

範囲内のドットは、ではなく `ExtendedData`、を参照します `Alert`。を使用して `{% .Labels %}`、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ドット `{% range .Alerts %}` が変更されるため機能します。範



囲が終了すると、ドットは範囲の開始前に存在していた値にリセットされます。この例ではで  
ずExtendedData。

```
{{ 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形式で出力するテンプレートを記述しましょう。ここでThe name of the label is \$name, and the value is \$value、\$nameとには各ラベルの名前と値\$valueが含まれています。

前の例と同様に、範囲を使用してのアラートを反復処理し、ドットがアラートのリスト内の現在のアラートを参照.Alertnsするようにします。次に、ソートされたラベルで2番目の範囲を使用して、ドット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印刷するには、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

テンプレート内の ステートメントの場合は、`with` を使用できます。例えば、`withAlerts` にアラート `There are no alerts` がいない場合に出力するには `.Alerts`、次のように記述します。

```
{{ if .Alerts }}
There are alerts
{{ else }}
There are no alerts
{{ end }}
```

で

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` はドット `with` を更新しての値を参照します。

```
{{ with .Alerts }}
There are {{ len . }} alert(s)
{{ else }}
There are no alerts
{{ end }}
```

## 変数

テキスト/テンプレートの変数は、テンプレート内に作成する必要があります。例えば、`$variable` 現在の値がドットの `with` という変数を作成するには、次のように記述します。

```
{{ $variable := . }}
```

範囲内 `$variable` の `with` または `range` を使用できます。`with` また、変数が定義された時点でのドットの値を参照し、ドットの現在の値を参照しません。

例えば、2 `range` `Labels` 番目の範囲で `with` を使用するテンプレートを記述することはできません。ここでのドットは、現在のアラートではなく、現在のラベルを参照するためです。

```
{{ 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

範囲内の各アラートのインデックスを取得するには、範囲の開始時にインデックス変数と値変数を定義します。

```
{{ $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。、、、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定義しないでください `__subject__text_values_list__text_alert_listdefault.titledefault.message`。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のテンプレートが作成されている場合、Grafana はどちらのテンプレートも使用できます。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防止したり表示したりしません。

```
{{ define "print_labels" }}
{{ end }}
```

## テンプレートを実行する

定義済みテンプレートは `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す必要があるカーソルを使用してテンプレート内で実行できます。

```
{{ template "print_labels" . }}
```

## テンプレートにデータを渡す

テンプレートドット内では、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、ドットはその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、ドット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便利です。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、発砲アラートだけを印刷するには、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 */}}
```

コメントに改行が追加され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

インデント、タブとスペースの両方、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存在します。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

テキスト/テンプレートでは、`{{-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：

```
{{ 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"
```

各範囲の開始`-}}`時に`}}`に変更された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 }}
```

これで、テンプレートのインデントと改行がテキストに表示されなくなります。

```
alertname = "Test"  
grafana_folder = "Test alerts"
```

## 通知テンプレートを作成する

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

再利用可能な通知テンプレートを作成して、連絡先に送信します。

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 つ持つことはできません。、、、 \_\_subject \_\_text\_values\_list \_\_text\_alert\_list default.title 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default.message。

連絡先タブには、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. アラート管理ポータル IRM、連絡先 を選択します。
2. 通知テンプレート タブを選択し、+ 通知テンプレートを追加 を選択します。
3. などの通知テンプレートの名前を選択しますemail.subject。
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. アラート管理ポータル IRM、連絡先 を選択します。
2. 通知テンプレート タブを選択し、+ 通知テンプレートを追加 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 です。
4. コンテンツフィールドに各テンプレートを書き込み `{{ end }}` ます。これには、各テンプレートの最初 `{{ define "name-of-template" }}` と最後にと が含まれます。通知テンプレート内の各テンプレートには、email.subject や など、わかりやすい名前を使用できます email.message。この場合、上記で入力した通知テンプレートの名前は再利用しないでください。

後のセクションでは、作成する可能性のあるテンプレートの詳細な例を示します。

#### 5. [Save] ( 保存 ) をクリックします。

### プレビュー通知テンプレート

通知テンプレートをコンタクトポイントで使用する前に、テンプレートがどのように表示されるかをプレビューします。これにより、作成するテンプレートの結果を理解し、テンプレートを保存する前にエラーを修正できます。

#### Note

通知プレビューは Grafana Alertmanager 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. アラート管理ポータルでの IRM、連絡先 を選択します。
2. 通知テンプレート タブを選択し、+ 通知テンプレートを追加する 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、アラートインスタンスだけでなくアラートデータも追加または編集できます。ペイロードデータウィンドウ自体にアラートデータを直接追加するか、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. 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、カスタムデータを追加 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. 通知に発動アラートを追加するか、解決されたアラートを追加するかに応じて、発動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、テンプレートコンテンツがどのように表示されるか、および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、プレビューに表示され、保存する前に修正できます。

6. 変更を保存します。

## メッセージの件名のテンプレートの作成

次の例のように、発動アラートと解決された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. 次の内容`email.subject`で というテンプレートを作成します。

```
{{ define "email.subject" }}
{{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
{{ end }}
```

2. テンプレートは、`template`キーワード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を使用します。

```
{{ template "email.subject" . }}
```

## Eメールのメッセージのテンプレートの作成

次の例のように、すべての発声アラートと解決されたアラートの概要を含む Eメールのメッセージ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のテンプレートを作成するには

1. コンテンツに `email.message_alert`と の 2 つのテンプレート`email`を使用して、 という名前の通知テンプレートを作成します`email.message`。

`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、各発砲アラートと解決済みアラートのラベルと値を出力するために使用されません。

```
{{- 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、`template`キーワード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います。

```
{{ template "email.message" . }}
```

## Slack メッセージのタイトルのテンプレートの作成

次の例のように、発声アラートと解決された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. 次の内容`slack.title`でという名前のテンプレートを作成します。

```
{{ 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 メッセージの内容のテンプレートの作成

ラベル、注釈、ダッシュボード URL など、すべての発音アラートと解決されたアラートの説明を含む Slack メッセージの内容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 通知の設定の詳細については、[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 メッセージの内容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 `slack` 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.message`。

`slack.print_alert` テンプレートには通知の構造が含まれている間、`slack.message` テンプレートはラベル、注釈、DashboardURL を出力するために使用されます。

```
{{ 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 の両方の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、このタイトルの Slack メッセージを送信する場合は 1 firing alerts, 0 resolved alerts、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. 次の内容 `common.subject_title` で というテンプレートを作成します。

```
{{ 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メールの連絡先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で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すると、既存のコンタクトポイントのリストが表示されます。
2. [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連絡先を編集することもできます。

3. メッセージ や件名 などのフィールドで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、 の形式を使用します。 `template_name` は `{{ template "template_name" . }}`、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に設計されています。

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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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 (タイプ)

アラートタイプには、次のデータが含まれます。

名前	Kind	説明	例
ステータス	string	firing-または-resolved	<code>{{ .Status }}</code>
ラベル	KeyValue	アラートにアタッチされたラベルのセット。	<code>{{ .Labels }}</code>
注釈	KeyValue	アラートにアタッチされた注釈のセット。	<code>{{ .Annotations }}</code>
[値]	KeyValue	Classic 条件を含むすべての式の値	<code>{{ .Values }}</code>

名前	Kind	説明	例
StartsAt	時刻	アラートが発せられた時刻。	<code>{{ .StartsAt }}</code>
EndsAt	時刻	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。それ以外の場合は、最後のアラートを受信してから設定可能なタイムアウト期間に設定します。	<code>{{ .EndsAt }}</code>
GeneratorURL	string	Grafana または外部 Alertmanager へのバックリンク。	<code>{{ .GeneratorURL }}</code>
SilenceURL	string	アラートを無音にするリンク (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 マネージドアラートの場合のみ。	<code>{{ .SilenceURL }}</code>
DashboardURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードにリンクします。Grafana マネージドアラートの場合のみ。	<code>{{ .DashboardURL }}</code>

名前	Kind	説明	例
PanelURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードパネルにリンクします。Grafana マネージドアラートの場合のみ。	<code>{{ .PanelURL }}</code>
フィンガープリント	string	アラートの識別に使用できるフィンガープリント。	<code>{{ .Fingerprint }}</code>
ValueString	string	アラート内の各縮小式のラベルと値を含む文字列。	<code>{{ .ValueString }}</code>

## ExtendedData

ExtendedData オブジェクトには、次のプロパティが含まれています。

名前	Kind	説明	例
レシーバー	string	通知を送信する連絡先の名前。	<code>{{ .Receiver }}</code>
ステータス	string	少なくとも 1 つのアラートが発動 firing している場合はステータスになり、それ以外の場合は になります resolved。	<code>{{ .Status }}</code>
アラート	<code>[]Alert</code>	この通知のすべての発動アラートと解決	There are <code>{{ len .Alerts }}</code> alerts



名前	Kind	説明	例
		済みアラートのリスト。	
アラートの送信	[]Alert	この通知のすべての発動アラートのリスト。	There are <code>{{ len .Alerts.Firing }}</code> firing alerts
解決済みのアラート	[]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 <code>{{ len .Alerts.Resolved }}</code> resolved alerts
GroupLabels	KeyValue	これらのアラートをグループ化するラベルは、通知に含まれます。	<code>{{ .GroupLabels }}</code>
CommonLabels	KeyValue	この通知のすべてのアラートに共通するラベル。	<code>{{ .CommonLabels }}</code>
CommonAnnotations	KeyValue	この通知のすべてのアラートに共通する注釈。	<code>{{ .CommonAnnotations }}</code>
ExternalURL	string	この通知を送信した Grafana ワークスペースまたは Alertmanager へのリンク。	<code>{{ .ExternalURL }}</code>

## KeyValue タイプ

KeyValue タイプは、ラベルと注釈を表すキーと値の文字列ペアのセットです。

として保存されたデータの直接アクセスに加えて KeyValue、データのソート、削除、変換の方法もあります。

名前	引数	戻り値	メモ	例
SortedPairs		キーと値の文字列ペアのソート済みリスト		<code>{{ .Annotations.SortedPairs }}</code>
Remove	[]文字列	KeyValue	指定されたキーなしでキー/値マップのコピーを返します。	<code>{{ .Annotations.Remove "summary" }}</code>
名前		[]文字列	名前のリスト	<code>{{ .Names }}</code>
[値]		[]文字列	値のリスト	<code>{{ .Values }}</code>

## Time (時間)

時間は Go [time](#)パッケージからのものです。時刻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形式で出力するにはMonday, 1st January 2022 at 10:00AM、次のテンプレートを記述します。

```
{{ .StartsAt.Format "Monday, 2 January 2006 at 3:04PM" }}
```

Go の時刻形式のリファレンスは、[で確認できます](#)。

## 連絡先の管理

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

コンタクトポイントリストビューには、既存のコンタクトポイントと通知テンプレートがすべて一覧表示されます。

コンタクトポイントタブでは、次のことができます。

- 連絡先と統合の名前とタイプを検索します。
- 既存の連絡先と統合をすべて表示します。
- 各連絡先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します。
- 通知配信のステータスを表示します。
- 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します。
- 通知ポリシーで使用されていない連絡先を削除します。

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、または削除します。

## Prometheus データソースのアラート通知のサイレンシング

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

外部アラートマネージャーデータソース (Amazon Managed Service for Prometheus を含む) の場合、無音でアラート通知を抑制できます。無音は通知の作成のみを停止します。無音はアラートルールの評価を妨げず、インスタンスのアラートがユーザーインターフェイスに表示されないようにしません。アラートを無音にすると、アラートが抑制される時間ウィンドウを指定します。

外部 Alertmanager データソースの無音を設定できます。

**Note**

アラート通知を一定の時間間隔で抑制したり、他のデータソースに対して抑制したり (定期的なメンテナンス期間中など)、無音 [ミュートタイミング](#) ではなく [ミュート](#) を使用します。

**無音を追加するには**

1. Grafana コンソールの Grafana メニューで、アラートと IRM、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. 「無音の作成」を選択します。
5. 無音の開始と終了日を選択して、無音が有効になるタイミングと終了するタイミングを示します。
6. 終了時間を設定する代わりに、期間 で無音が強制される時間を指定します。これにより、無音の開始フィールド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。マッチャーは、無音が適用されるルールを決定します。一致するアラート (発音状態) は、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、無音を説明するコメントを追加します。
9. [送信] を選択します。

**無音を編集するには**

1. Grafana コンソールの Grafana メニューで、アラートと IRM、アラート を選択します。
2. 既存の無音のリストを表示するには、無音を選択します。
3. 編集する無音を見つけ、編集 (ペンアイコン) を選択します。
4. 必要な変更を加え、送信 を選択して変更を保存します。

編集アイコン (ペン) を選択すると、既存の無音を編集できます。

**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、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 `matcher` とコメントを指定します。 `matcher` パラメータは、次のいずれかの形

式[label][operator][value]operatorである必要があります。(regexではなく等しい)、!(regexではなく等しい)、=~(regals、regex)、(regals、regex)、!~(not equals、regex)。URLには、キーを持つ多くのクエリパラメータを含めることができますmatcher。例えば、一致するラベル severity=critical & cluster!~europe-.\*とコメントを持つ無音フォームにリンクするにはSilence critical EU alerts、URLを作成しますhttps://mygrafana/alerting/silence/new?matcher=severity%3Dcritical&matcher=cluster!~europe-\*&comment=Silence%20critical%20EU%20alert。

外部 Alertmanager の新しい無音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。


無音を削除するには

1. Grafana コンソールの Grafana メニューで、アラートと IRM、アラート を選択します。
2. 既存の無音のリストを表示するには、無音を選択します。
3. 終了する無音を選択し、無音 を選択します。これにより、アラートの抑制は終了します。

#### Note

サイレンシングを解除すると、終了時刻が現在の時刻に設定されているかのように、アラート抑制が終了します。(自動または手動で) 終了した無音は 5 日間保持され、一覧表示されます。リストから無音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に設計されています。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、「」を参照してください[Grafana バージョン 9 での作業](#)。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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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タイプ別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記載されていま

す。Prometheus 互換データソースのアラートルールを表示できますが、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記載されています。

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、および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、ヘルス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。

## アラートルールを表示する

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、アラートと IRM、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. として表示 では、任意のオプションを選択して、グループ化ビュー、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として生じる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

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。

- 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エクスポートルールグループ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。
- 「詳細」を選択し、グループ内の個々のアラートルールの横にあるエクスポートを変更して、プロビジョニングされたアラートルールを編集し、変更し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」リポジトリに侵入することなく、アラートルールクエリが正しいかどうかをすばやく確認します。

## グループ化されたビュー

グループ化ビューには、フォルダ別にグループ化された Grafana アラートルールと、namespace+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、ルールを管理するためのデフォルトのルールリストビューです。各グループを展開して、このグループのルールのリストを表示できます。ルールをさらに展開して詳細を表示します。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、状態別にグループ化されたアラートルールが表示されます。このビューを使用して、どのルールがどの状態にあるかの概要を取得します。各ルールを展開して詳細を表示できます。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、その詳細を表示できます。

## 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を表示できます。
2. 「ラベルによる検索」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` です。
3. 状態 で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を表示できます。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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ミュートタイミングは、ポリシーの新しい通知が生成または送信されない定期的な時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間に発生し、繰り返し発生するのを防ぐために使用します。

無音と同様に、ミュートタイミングによってアラートルールの評価が妨げられることも、ユーザーインターフェイスにアラートインスタンスが表示されないようにすることもなくなります。通知の作成のみを防ぎます。

外部の Alertmanager データソースの Grafana マネージド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無音

次の表は、ミュートのタイミングと無音の違いを示しています。

ミュートタイミング	無音
繰り返し発生する可能性のある時間間隔定義を使用します。	開始時刻と終了時刻が固定されています。
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、無音にするかどうかを決定します。

### ミュートタイミングの追加

Grafana ワークスペースでミュートタイミングを作成できます。

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、アラートと IRM、アラート を選択します。
2. 通知ポリシー を選択し、「タイミングをミュートする」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と照合する[時間間隔](#)を作成します。
6. ミュートの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を設定したら、定期的にミュートする通知ポリシーに追加して使用します。

通知ポリシーに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、アラートと IRM、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。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 – 開始時刻を含み、終了時刻を含まない時間 (UTC、ロケーションが選択されていない場合は UTC、それ以外の場合は現地時間)。
- Location – 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 – 曜日または曜日の範囲。例えば monday:thursday です。
- Days of the month – 1 か月内の日付。値の範囲は 1~ です31。負の値は、月の曜日を逆の順序で指定するため、-1は月の最終日を表します。
- 月 - 暦月の完全な月名のいずれかの数値で表された、その年の月。例えば 1, may:august です。
- 年 - 間隔の年。例えば 2023:2024 です。

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、を使用した範囲もサポートします:。例えば monday:thursday です。

フィールドを空白のままにすると、任意の時点でフィールドと一致します。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 UTC の 12:00 から 24:00 まで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：
  - 開始時刻: 12:00
  - 終了時刻: 24:00
- 曜日: monday
- 月: 3, 6, 9, 12
- 曜日: 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に設計されています。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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](#)

関連していますが、各コンポーネントは微妙に異なる情報を伝えます。

## 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、アラートと IRM、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、その状態とヘルスを表示します。

### 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]	評価エンジンによって返される時系列は少なくとも1つです pending。		
ファイリング	評価エンジンによって返される時系列は少なくとも1つあります firing。		

#### Note

アラートは最初に 移行 pending し firing、次に 移行するため、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 ももではないアラートの状態 firing。すべてが期待どおりに動作しています。		
[保留中]	設定されたしきい値の期間未満にアクティブだったアラートの状態。		
アラート	設定されたしきい値の期間よりも長くアクティブだ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ていません。		
アラート	アラートルールの評価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または状態が発生したときに最後のError状態を維持するように設定できます。これにより、アラートの発音と、解決と再発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 firingすると pending から に移行します。

### アラートルールの正常性

アラートルールには、次のいずれかのヘルスステータスがあります。

状態	説明		
没頭	アラートルールを評価するときにエラーは発生しません。		
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	ルール評価中に少なくとも1つの時系列にデータがないこと。		
{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維持するように設定されています。		

### NoData および エラーの特別なアラート

アラートルールを評価すると、状態NoDataまたは が生成されます。Grafana アラートはError、次の追加ラベルを持つアラートインスタンスを生成します。

ラベル	説明		
アラート名	状態に応じてDataSourceError、DataSourceNoData またはのいずれか。		

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](#)「」トピックで説明されているように、アラートルールErrorでデータまたはエラー処理なしをNoDataまたは に設定する必要があります。

これらのアラートは、無音の追加、コンタクトセンターへのルーティングなど、通常のアラートと同じ方法で処理できます。

### アラートグループによる表示とフィルタリング

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートグループは、Alertmanager 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と、重複するアラートルールが発せられなくなります。

アラートグループを表示したり、特定の条件に一致するアラートルールをフィルタリングしたりできます。

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、アラートと IRM、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のアラートマネージャーを選択します。
4. カスタムグループからドロップダウンでラベルの組み合わせを選択すると、デフォルト以外のグループが表示されます。これは、通知ポリシーのグループ化のデバッグと検証に役立ちます。

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のいずれかで指定されたラベルがアラートに含まれていない場合、アラートはのヘッダーを持つキャッチオールグループに追加されますNo grouping。

アラートは、ラベルまたはアラートの状態で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` です。

状態別にフィルタリングするには

- 状態で、アクティブ状態、抑制状態、または未処理状態のいずれかを選択して、選択した状態に一致するアラートを表示します。その他のアラートはすべて非表示になります。

通知エラーの表示

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知エラーを表示し、送信に失敗した、または受信されなかった理由を理解します。

**Note**

この機能は Grafana Alertmanager 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、アラートとコンタクトポイントを選択します。

いずれかの問い合わせポイントに障害が発生した場合、ワークスペースの右側隅にあるメッセージにエラーと数が表示されます。

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。

連絡先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. Health 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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 のパネルとビジュアルイゼーション](#)



- [Grafana バージョン 9 でを試す](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、設定された任意のからのデータを操作できます[データソースに接続する](#)。

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。その結果、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](#)
- [ダッシュボードの使用状況の評価](#)
- [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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このトピックでは、ダッシュボード機能とショートカットの概要と、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 ホームアイコンをクリックすると、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されます。
2. タイトル	ダッシュボードの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ジュー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

機能	説明
	<p>エリを管理します。ダッシュボード設定の詳細については、「」を参照してください<a href="#">ダッシュボード設定の変更</a>。</p>
6. タイムピッカードロップダウン	<p>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。</p> <ul style="list-style-type: none"><li>• タイムゾーンと会計年の設定は、時間範囲コントロールから変更するには、時間設定の変更ボタンをクリックします。</li><li>• 時間設定はダッシュボードごとに保存されます。</li></ul>
7. ズームアウト時間範囲	<p>クリックして時間範囲を拡大します。時間範囲コントロールの使用法の詳細については、「」を参照してください<a href="#">ダッシュボードの時間範囲の設定</a>。</p>
8. ダッシュボードの更新	<p>をクリックすると、すぐにクエリがトリガーされ、ダッシュボードデータが更新されます。</p>
9. ダッシュボードの時間間隔の更新	<p>ダッシュボードの自動更新の時間間隔をクリックして選択します。</p>
10. 表示モード	<p>クリックして、テレビやセルフサービス機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。</p>

機能	説明
11. ダッシュボードパネル	<p>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、パネルの追加をクリックします。</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> <li>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。</li> </ul>
12. グラフの凡例	<p>凡例から直接、シリーズの色、Y 軸、およびシリーズの可視性を変更します。</p>
13. ダッシュボード検索	<p>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。</p>
14. ダッシュボード行	<p>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分割です。</p> <ul style="list-style-type: none"> <li>行は折りたたんだり展開したりできるため、ダッシュボードの一部を非表示にできます。</li> <li>折りたたまれた行内のパネルはクエリを発行しません。</li> <li>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。</li> </ul>

## キーボードショートカット

Grafana には、多数のキーボードショートカットがあります。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 はセルフサービスモードを切り替えます (メニューを非表示)。
- 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 全画面表示モードまたは編集モードの場合、 は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、ポインターを使用してパネルにカーソルを合わせます。

- e パネル編集ビューを切り替える
- v パネルの全画面表示を切り替える
- ps パネル共有機能を開く
- pd 重複パネル
- pr パネルを削除します
- pl パネルの凡例を切り替える

## ダッシュボードの時間範囲の設定

Grafana には、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、一般的な時間制御、ダッシュボード全体の時間設定、および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、s (seconds)、m (minutes)、h (hours)、d (days)、M (months)、Q (quarters) および w (weeks) の時間単位をサポートしています y (years)。

マイナス演算子を使用すると、現在を基準にして、時間を遡ることができます。単位の全期間 (日、週、月) を表示する場合は、末尾 /<time unit> に を追加します。会計期間を表示するには、fQ (fiscal quarter) および fy (fiscal year) 時間単位を使用します。

plus 演算子を使用すると、現在を基準にして時間的に先に進むことができます。例えば、この機能を使用して、将来予測データを調べることができます。

次の表に、相対範囲の例を示します。

相対範囲の例	From	目的
過去 5 分	now-5m	now
これまでの日付	now/d	now
今週	now/w	now/w
今のところ、この週	now/w	now
今月	now/M	now/M
今月はここまで	now/M	now
前月	now-1M/M	now-1M/M
今のところ、この 1 年	now/Y	now
本年	now/Y	now/Y
前事業年	now-1y/fy	now-1y/fy

#### Note

Grafana アラートは `now+n`、将来のタイムスタンプと `n` の先頭 `now-1n/n` から `n` のタイムスタンプの終了まではサポートしていません。

### 一般的な時間範囲の制御

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。以下に、一般的な時間範囲の制御について説明します。

- 現在の時間範囲は、タイムピッカーとも呼ばれ、表示している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、範囲とそのソース

(ローカルブラウザなど)の正確なタイムスタンプが表示されます。現在の時間範囲をクリックして変更します。現在の時刻は、過去 15 分などの相対時間範囲、または 2020-05-14 00:00:00 からなどの絶対時間範囲を使用して変更できます。2020-05-15 23:59:59。

- 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、過去 2 年、昨日、昨日の前日、先週のこの日、今日から今日まで、この週から今日まで、今月から現在までなどがあります。
- 絶対時間範囲は 2 つの方法で設定できます。正確な時間値または相対時間値を From フィールドと To フィールドに入力し、時間範囲を適用 をクリックします。または、From フィールドまたは To 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します。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、クリックしてドラッグし、表示する視覚化の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。

### ダッシュボードを更新する

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すぐに実行され、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。ただし、ダッシュボードを定期的に更新する場合は、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て、更新間隔を選択します。

### URL を使用して時間範囲を制御する

ダッシュボード URL に次のクエリパラメータを指定することで、ダッシュボードの時間範囲を制御できます。

- from は、ms エポック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、ms エポック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 は、 から time-time.window/2までの時間範囲time.windowを定義しますtime+time.window/2。両方のパラメータは ms で指定する必要があります。例えば、 ? time=1500000000000000&time.window=10000は から 1499999995000 までの 10 秒の時間範囲になります1500000005000。

## ダッシュボードの構築

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana ワークスペースを作成してサインインしたら、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

- [ダッシュボードを作成する。](#)
- [パネルを追加または編集する](#)
- [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](#)



ダッシュボードを作成する。

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

## 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。視覚化を表示するには、各パネルに少なくとも 1 つのクエリが必要です。開始する前に、以下の前提条件を満たしていることを確認してください。

- 適切なアクセス許可があることを確認します。権限の詳細については、[ユーザー、チーム、アクセス許可](#) をご参照ください。
- 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：

- 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。
- 新しいパネルを追加 をクリックします。
- 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。
- データソースのクエリ言語でクエリを記述または構築します。
- 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、「[視覚化オプション](#)」を参照してください。
- 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、必要な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、ページの右上隅にある保存をクリックします。

**Note**

メモは、ダッシュボードを以前のバージョンに戻す必要がある場合に役立ちます。

## 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で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、[「パネルの繰り返しの設定」](#)を参照してください。

また、Multi-value または Include all values で設定された変数がある場合は、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、次のステップを完了する必要があります。

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。ダッシュボードには、任意の場所にパネルを配置できます。

## 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のサイズは、ニーズに合わせて設定できます。

## パネルを追加または編集する

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードを作成したら、いつでもパネルを追加、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。
- **編集パネル** ダッシュボード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、の削除を選択します。

## 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボード設定ページでは、次のことができます。

- 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します。
- 注釈クエリを追加します。
- ダッシュボード変数を追加します。
- リンクを追加します。
- 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、一般 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. 次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モニタリング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時間を使用します。
  - 表示ユーザーブラウザ用に設定されたタイムゾーン、ローカルブラウザの時刻 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で区切られ、有効な時間単位を受け入れます。
  - 遅延は、now遅延時間を入力して時間を上書きします。NULL 値を回避するために、データ集約の既知の遅延に対応するには、このオプションを使用します。
  - タイムピッカーを非表示にすると、Grafana タイムピッカーの表示が削除されます。

#### Note

時間制御を行うには、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、ダッシュボード全体のグラフで縦線として視覚化できます。また、カーソルを合わせるとイベント情報を表示できます。

### 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記入して、クエリと注釈を作成します。

クエリエディタ 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。サーバー、アプリケーション、センサー名などのものをメトリクスクエリでハードコーディングする代わりに、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** をクリックし、**変数を追加** ボタンをクリックします。
2. 一般セクションで、**変数の名前**。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、ダッシュボードヘッダーのすぐ下に他のダッシュボードやウェブサイトへのリンクを配置できます。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の**リンク** をクリックし、**リンクを追加** ボタンをクリックします。
2. 「**タイトル**」と「**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オプション**のいずれかを選択し、**適用** をクリックします。

#### Note

タグは、すべてのダッシュボードに特定のタグがある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、ユーザーがリンクにカーソルを合わせ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、リンクの横にあるアイコンを選択し、ダッシュボードリンクオプションのいずれかを選択します。

### 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 オブジェクトで表されます。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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana は、クエリパラメータとして渡された変数値をダッシュボード URLs。詳細については、[「ダッシュボードリンクの管理」](#)および[「テンプレートと変数」](#)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、プレフィックスが付い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var-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 ですvalue。

## 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、value1 との 2 つの値を持つダッシュボード変数の例 var-example=value1&var-example=value2 として を解釈します value2。

## 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。変数を追加する詳細と手順については、[「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の受け渡し

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の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ます。また、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
```

クエリパラメータは、example\_key キー、= 演算子、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 var-adhoc=key|=|value を適用します。

### Note

URLs アドホックフィルターと共有する場合は、URL をエンコードすることを忘れないでください。上記の例では、パイプを (|) に、等価演算子 %7C を に置き換え (=) ます %3D。

## URL を使用した時間範囲の制御

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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 のすべてのインスタンスに反映されます。ライブラリパネルは、複数のダッシュボード間でのパネルの再利用を効率化します。

ライブラリパネルは、保存したダッシュボードと一緒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、「ライブラリの作成」パネルをクリックします。
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. 変更を保存するには、ライブラリの作成パネル をクリックします。
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。

ライブラリパネルを作成したら、表示される任意のダッシュボードを使用してパネルを変更できます。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 Dashboards 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. リストまたは検索をフィルタリングして、追加するパネルを検索します。
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

パネルを変更し、ライブラリパネルの他のインスタンスに影響を与えないようにする場合は、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。パネルが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

パネルを変更し、ライブラリパネルの他のインスタンスに影響を与えないようにする場合は、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 Dashboard 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を表示できます。
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。フォルダまたはタイプでパネルを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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードのバージョンを保存するたびに、そのバージョンのコピーが保存され、ダッシュボードの以前のバージョンが失われることはありません。これらのバージョンのリストは、ダッシュボードの設定を入力し、左側のメニューでバージョンを選択して使用できます。

ダッシュボードのバージョン履歴機能を使用すると、以前に保存したダッシュボードの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、リストから比較する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。

ダッシュボードを表す raw JSON の差分を表示する場合は、下部にある JSON 差分を表示 ボタンをクリックして表示することもできます。

差分対象のバージョンに復元する場合は、右上のバージョンに復元 <x> ボタンをクリックします。

### 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クリックするか、差分ビューに表示されるバージョンに復元 <x> ボタンをクリックします。ボタンをクリックすると、復元を確認す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、以前のバージョンと同じデータを含む新しいバージョンが作成され、バージョン番号は異なります。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だけに行われます。

## 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

リンクを使用して、一般的に使用されるダッシュボード間を移動したり、他のユーザーを視覚化に接続したりできます。リンクを使用すると、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 は、ダッシュボードリンク、パネルリンク、および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には、パネルの左上隅にあるアイコンをクリックします。

### 使用するリンクの選択

まず、ダッシュボード間で現在どのように移動しているかを調べます。多くの場合、ダッシュボードのセット間でジャンプし、それぞれで同じコンテキストを見つけるのに苦労している場合、リンクはワークフローの最適化に役立ちます。

次のステップでは、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、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すべてのパネルに関連するわけではないが、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。
- 外部サイトにリンクする場合は、ダッシュボードリンクまたはパネルリンクを使用できます。
- 特定のシリーズ、または 1 つの測定にドリルダウンする場合は、データリンクを使用します。

### URL を使用した時間範囲の制御

パネルまたはダッシュボードの時間範囲を制御するには、ダッシュボード URL にクエリパラメータを指定します。

- `from` は、ms エポックで指定された時間範囲の下限を定義します。
- `to` は、ms エポック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` から `time-time.window/2` までの時間範囲 `time.window` を定義します `time+time.window/2`。両方のパラメータは ms で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は から 1499999995000 までの 10 秒の時間範囲になります `1500000005000`。

## 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て、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 GitHub に関する問題](#)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の追加または新しい 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。
  - タグ付き: 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: 多数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** : 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** : 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. [Add] (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 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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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の追加または新しい 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ができます。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** : 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、ms エポックで指定された時間範囲の下限を定義します。
    - **to** は、ms エポック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` までの時間範囲 `time.window` を定義します `time+time.window/2`。両方のパラメータは ms で指定する必要があります。例えば、?

`time=1500000000000&time.window=10000`は から `1499999995000` までの 10 秒の時間範囲になります `1500000005000`。

- 変数値: 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-variable1}=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. [Add] (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クリックします。
2. 設定を変更し、 の更新 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、削除する重複アイコンの横にあるごみ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される独自のリンクセット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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana の適切なデータにズームインされ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、 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. パネルタブで、リンクセクションまで下にスクロールします。
  3. リンクを展開し、リンクを追加 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人間が読めるラベルです。
  5. リンク先の URL を入力します。ダッシュボードで定義されている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、URL フィールドをクリックして、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、リンクはユーザーを適切なコンテキストに送信し、関連する変数は既に設定されています。

時間変数を使用することもできます。

- from は、ms エポックで指定された時間範囲の下限を定義します。
  - to は、ms エポックで指定された時間範囲の上限を定義します。
  - time および は、 から  $time - time.window/2$  までの時間範囲  $time.window$  を定義します。  $time + time.window/2$ 。両方のパラメータは ms で指定する必要があります。例えば、`?time=1500000000000&time.window=10000` は から 1499999995000 までの 10 秒の時間範囲になります `1500000005000`。
- 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、編集リンク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。

ダッシュボードの右上の保存 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 オブジェクトで表されます。ダッシュボードメタデータには、ダッシュボードのプロパティ、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 を表示するには。

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。

#### **i** 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たデフォルト値です。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": []
}
```

以下に、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的な数値識別子 (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
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
スタイル	暗いものや明るいものなど、ダッシュボードのテーマ

名前	使用方法
timezone	UTC やブラウザなどのダッシュボードのタイムゾーン
編集可能	ダッシュボードが編集可能かどうか
graphTooltip	共有クロスヘアまたはツールヒントなしの場合は 0、共有クロスヘアの場合は 1、共有クロスヘアと共有ツールヒントの場合は 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
タイムピッカー	timepicker メタデータ、詳細については「 <a href="#">timepicker セクション</a> 」を参照してください。
テンプレート	メタデータのテンプレート化、詳細については「 <a href="#">テンプレート化セクション</a> 」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 <a href="#">注釈</a> を参照してください。
更新	自動更新間隔
schemaVersion	JSON スキーマ (整数) のバージョン。Grafana 更新によってスキーマが変更されるたびに増分されます。
バージョン	ダッシュボード (整数) のバージョン。ダッシュボードが更新されるたびに増加します。
パネル	panels 配列 (詳細については、以下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す。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、それぞれが 30 ピクセルを表します。
- x: と同じ単位の x の位置w。
- y: と同じ単位の y の位置h。

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上に移動します。

## タイムピッカー

```
"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

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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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ダッシュボードや複数のチームが多数ある場合に役立ちます。

### 前提条件

Grafana 管理者権限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 Dashboards > New folder 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。

### **i** Note

ダッシュボードを保存するときに、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のできるフォルダページに移動します。

## 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で次のタスクを実行できます。

- フォルダ内のダッシュボードを移動または削除します。
- フォルダの名前を変更します (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継承される) にアクセス許可を割り当てます。

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、およびユーザーにアクセス許可を割り当てることのできる場合に使用されます。

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、必要なものがすべて含まれた 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できるよう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。

タイプのテンプレート変数Constantはダッシュボードに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。
  - ダッシュボードの JSON ファイルをアップロードします。
  - [Grafana.com](https://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たり、ダッシュボードで使用するデータソースを選択したり、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) を指定したりできます。

### 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、ダッシュボードの一般的な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可能性があります。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。
- 多くの時系列や長期間のクエリを行っていますか？ これらの条件の両方により、Grafana またはデータソースが大量のデータをプルし、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の負荷が高い可能性があります。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にクエリを実行します。ダッシュボードの更新レートを低く設定すると、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、変更が見られるようにデータがシステムに送信されないため、これを頻繁にクエリ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。

- 必要な場合を除き、ダッシュボード、パネル、または変数の自動更新を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ことも、妥当な期間 (10 分や 1 時間ごとなど) の更新レートを設定することもできます。
- 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。

### ダッシュボードとパネルの共有

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana を使用すると、ダッシュボードやパネルを組織内の他のユーザーと共有したり、特定の状況では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イメージを表示するには、承認されたビューワーの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。リンクがあるユーザーは誰でもアクセスできます。スナップショットには表示の認証を必要としないため、Grafana は、スナップショットから取得したアカウントに関連する情報と機密データを削除します。

### 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. 上部の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。

## 直接リンクの共有

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. [Copy] をクリックします。このアクションは、デフォルト URL 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。

## 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削除し、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を残します。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 は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パブリックに共有します。

## 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、[「ダッシュボードのエクスポートとインポート」](#)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。オプションで、短縮 URL を有効にして共有できます。

1. Copy 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、直接リンクレンダリングされたイメージをクリックして、パネルのイメージを共有することもできます。

サーバー側で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 UTC%2BHH%3AMMが UTC から時間と分でオフセットされる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、タイムアウトを増やすことができます。
- scale: 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 が高い) 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に共有します。Grafana は機密データを削除し、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」をクリックしてタブを開きます。
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パブリックに共有します。

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

共有パネルダイアログボックスからライブラリパネルを作成するには。

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 で、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. ライブラリの作成パネルをクリックして、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、大画面に最適です。Dashboards 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を使用して、既存のプレイリストにアクセスします。使用可能な 5 つの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。

### プレイリストへのアクセス

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

プレイリストは 5 つの異なるビューモードで開始できます。表示モードは、メニューとナビゲーションバーをダッシュボードに表示する方法を決定します。

デフォルトでは、各ダッシュボードは、プレイリストを作成または編集するときに設定した間隔フィールドに入力された時間で表示されます。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でプレイリストを制御できます。

プレイリストには、プレイリストを作成または編集するときに設定された、Intervalフィールドで指定された時間の各ダッシュボードが表示されます。プレイリストが起動したら、画面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。

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見つけ、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	<ul style="list-style-type: none"> <li>• サイドメニューは引き続き表示されます。</li> <li>• 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。</li> </ul>
TV モード	<ul style="list-style-type: none"> <li>• サイドメニューとダッシュボードサブメニュー (可変ドロップダウンとダッシュボードリンクを含む) は非表示または削除されます。</li> <li>• 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。</li> <li>• 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。</li> <li>• <code>dv</code> 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 <code>?inactive</code> にパラメータを追加して手動で有効にします。</li> <li>• ポインターの移動またはキーボードアクションで無効にします。</li> </ul>

モード	説明
TV モード (自動フィットパネル付き)	<ul style="list-style-type: none"> <li>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。</li> <li>ダッシュボードパネルは自動的に調整され、画面上のスペースが最適化されます。</li> </ul>
Kiosk モード	<ul style="list-style-type: none"> <li>サイドメニュー、ナビゲーションバー、ro、およびパネルコントロールは、完全に非表示/表示から削除されます。</li> <li>プレイリストの開始後、dv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。</li> <li>同じショートカットを使用して手動で無効にできます。</li> </ul>
Kiosk モード (自動フィットパネル付き)	<ul style="list-style-type: none"> <li>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は、完全に非表示/表示から削除されます。</li> <li>ダッシュボードパネルは自動的に調整され、画面上のスペースが最適化されます。</li> </ul>

## プレイリストの制御

プレイリストは、画面上部のナビゲーションバーを使用して、起動後に通常モードまたはテレビモードで制御できます。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。

Button	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 (二乗)	プレイリストを終了し、現在のダッシュボードに終了します。



Button	アクション
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表示モードでローテーションします。
[Time range] (時間範囲)	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、またはカスタム時間範囲を表示するように設定できます。
更新 (円矢印)	ダッシュボードを再ロードして、現在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た順序でダッシュボードを表示できます。

1. プレイリストページで新しい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. Dashboards で、タイトル別追加とタグ別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 :
  - ダッシュボードを名前、正規表現、またはタグで検索します。
  - スターステータスまたはタグで結果をフィルタリングします。
  - 上矢印と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. [Save] 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、プレイリストページに追加して開始できます。

### Important

プレイリストを保存する前に、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、使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、および少なくとも 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. [Save] 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. 名前と時間間隔を更新し、上記の「プレイリストの作成」の手順を使用して、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. [Save] 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再配置

1. 移動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. [Save] 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. [Save] をクリックして変更を保存します。

### ビューモードでのプレイリストの共有

希望するビューモードでリンクアドレスをコピーし、URL を送信先に貼り付けることで、プレイリストを共有できます。

1. Dashboards 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、目的の表示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して、URL をクリップボードにコピーします。
5. URL を送信先に貼り付けます。

### 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

変数は値のプレースホルダーです。メトリクス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、パネルのメトリクススクエリが新しい値を反映するように変更されます。

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。サーバー、アプリケーション、センサー名などのものをメトリクススクエリでハードコーディングする代わりに、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、完全な編集権限を与えたくない管理者にとって特に役立ちます。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も作成できます。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、変数を含む任意のクエリです。例えば、ダッシュボードを管理して複数のサーバーをモニタリングする場合、サーバーごとにダッシュボードを作成したり、ダッシュボードを1つ作成して次のようなテンプレートクエリでパネルを使用したりできます。

```
wmi_system_threads{instance=~"$server"}
```

変数値は、常に構文 `var-<varname>=value` 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 の変数リストにリスト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、最初に表示されます (ダッシュボードの左端)。

## 変数の追加と管理

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

次の表に、Grafana に同梱されている変数のタイプを示します。

変数タイプ	説明
Query	メトリクス名、サーバー名、センサー IDs、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。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で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。
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と値のフィルター

変数タイプ	説明
	(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。「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

作成する任意のタイプの変数の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、ページの上にある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、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選択します。
  - 選択なし (空白): 変数ドロップダウンには変数名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ことができるデータソースクエリを作成できます。例えば、クエリ変数は、サーバー名、センサー IDs、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フェッチ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。クエリが数値や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る場合があります。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、[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。
  - ダッシュボードロード時：ダッシュボードがロードされるたびにデータソースをクエリします。これにより、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、ダッシュボードのロードが遅くなります。
  - 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をクエリします。このオプションは、変数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で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の上にカーソルを移動し、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式を入力して、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、「[正規表現で変数をフィルタリングする](#)」を参照してください。

- 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を選択します。デフォルトのオプション Disabled 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ることを意味します。
- (オプション) [選択オプションを入力します](#)。
- 値のプレビューでは、Grafana は現在の変数値のリストを表示します。これらを確認して、期待どおりであることを確認します。
- 追加 を選択して、変数をダッシュボードに追加します。

## 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、変更されないサーバー名やリージョン名がある場合は、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

- 上記のように、一般的なオプションを入力します。
- 左

カンマリストで区切られた値で、この変数の値をカンマ区切りリストに入力します。スペースと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 です。

- (オプション) [選択オプションを入力します](#)。
- 値のプレビューでは、Grafana は現在の変数値のリストを表示します。これらを確認して、期待どおりであることを確認します。
- 追加 を選択して、変数をダッシュボードに追加します。

## テキストボックス変数の追加

テキストボックス変数には、オプションのデフォルト値を含むフリーテキスト入力フィールドが表示されます。任意の値を入力できるため、これは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、または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 のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 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。これは、共有するダッシュボードのメトリクスパスプレフィックスに役立ちます。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。

定数変数には柔軟性がありません。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が、すべてのクエリで再入力したくない複雑な値がある場合に便利です。例えば、というサーバーパスがある場合 `i-0b6a61efe2ab843gg`、それをという変数に置き換えることができます `$path_gg`。

## 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。文字、数字、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 のプレビューで、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 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。これらは、データソースのインスタンスが複数ある場合に便利です。場合によっては、異なる環境にもあります。

## 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. Type 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で、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変数値のドロップダウンリストから入力します。すべてのインスタンスを表示するには、このフィールドを空のままにします。
4. (オプション) [選択オプション](#) を入力します。
5. 値のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 間隔変数の追加

間隔変数を使用して、1m、1h、などの時間範囲を表します1d。これらは、ダッシュボード全体のグループとして時間コマンドで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。自動オプションを使用して、時間範囲ごとに設定された数のデータポイントを返すこともできます。

間隔変数は、時間 (InfluxDB の場合)、日付ヒストグラム間隔 (Elasticsearch の場合)、または要約関数パラメータ (Graphite の場合) でグループ化するパラメータ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。次の時間単位がサポートされています: s (seconds)、m (minutes)、h (hours)、d (days)、w (weeks)、M (months)、および y (years)。デフォルト値を承諾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) オプションをリストに追加する場合は、自動auto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割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30分単位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 - ステップカウンタ間隔が時間を割かない最小しきい値。30 分の例を続行するには、最小間隔が 2m に設定されている場合、Grafana はデータを 15 の 2 分単位でグループ化します。
4. 値のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

## 間隔変数の例

次の例は、Graphite 関数 `myinterval` のテンプレート変数を示しています。

```
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すると、指定し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と値の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記述します。

### 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、変数をダッシュボードに追加します。

## アドホックフィルターを作成する

アドホックフィルターは、利用可能な最も複雑で柔軟な変数オプションの 1 つです。この変数では、変数オプションの通常のリストの代わりに、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。この方法で適用したフィルターは、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の設定

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オプションであり、デフォルトではオフになっています。

### 複数值変数

複数の値が選択された変数を補間することは、変数が使用され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、難しいです。Grafana は、各データソースプラグインが複数の値に使用する形式をテンプレート補間エンジンに通知できるようにすることで、この問題を解決しようとしています。

#### 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、変数のカスタムオール値オプションが空白である必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (結合) します。例えば `value1,value2,value3` です。カスタム `all` 値を使用する場合は、代わりに `*または` になります `all`。

### Graphite データソースを使用した複数值変数

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` `{host1,host2,host3}` であるかのように補間されます。

### Prometheus または InfluxDB データソースを使用した複数值変数

InfluxDB と Prometheus は正規表現式を使用するため、同じ変数がとして補間されます (`host1|host2|host3`)。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、正規表現制御文字を持つ値は正規表現式を破棄します。

### Elastic データソースを使用した複数值変数

Elasticsearch は Lucene クエリ構文を使用するため、同じ変数がとしてフォーマットされます (`"host1" OR "host2" OR "host3"`)。この場合、値に明らかな制御語と引用符のみが含まれるように、すべての値をエスケープする必要があります。

### 複数值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把握するのは難しい場合があります。特に InfluxDB および Prometheus では、正規表現構文を使用するには、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、すべてを含めるオプションをオフにします。
- [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。

### すべてのカスタム値

このオプションは、Include All オプションが選択されている場合にのみ表示されます。

Custom all value フィールドに regex、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のように、カスタムのオール値を指定する方がよい場合があります。

カスタム正規表現、globs、または Lucene 構文を Custom all value オプションに含めるには、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リストし、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、および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 つの時間範囲変数が組み込まれています。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ますが、日付の書式を制御できます。

構文	結果の例	説明
<code>\${__from}</code>	1594671549254	Unix ミリ秒エポック
<code>\${__from:date}</code>	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
<code>\${__from:date:iso}</code>	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
<code>\${__from:date:seconds}</code>	1594671549	Unix 秒エポック
<code>\${__from:date:YYYY-MM}</code>	2020-07	: 文字を含まないカスタム日付形式

上記の構文は `$__to` でも機能します。

## `$__interval`

`$__interval` 変数をパラメータとして使用して、時間 (InfluxDB、MySQL、Postgres、MSSQL の場合)、日付ヒストグラム間隔 (Elasticsearch の場合)、または要約関数パラメータ (Graphite の場合) でグループ化できます。

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くのデータポイントがある場合、間隔を長くしてグループ化することでクエリをより効率的に行うことができます。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ことがあります。時間または日でグループ化すると、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。

概算計算:  $(to - from) / resolution$

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 に計算され、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、ポイントは日別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` を使用する必要があります。

InfluxDB および Elasticsearch データソースには、間隔をハードコードしたり、`$__interval` 変数の最小制限を設定したりするために使用される Group by time interval フィールドがあります (>構文 -> を使用>10m)。

### `$__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 の場合 `20m`、`$__interval_ms` は です `1200000`。

### `$__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です。

### `$__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。これは によって計算されます `to - from`。ミリ秒と および という 2 `$__range_ms` 番目の表現があります `$__range_s`。

### `$__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 `rate` 関数で使用するためのものです。

### `$timeFilter` または `$__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、時間範囲間隔 `Last 7 days` 式は です `time > now() - 7d`。

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。Grafana は、クエリエディタモードで 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手動で追加できます: `WHERE $timeFilter`。
- Azure Monitor データソースに分析クエリをログ記録します。
- MySQL、Postgres、MSSQL の SQL クエリ。



- `$_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 と呼ばれ、変数クエリに 1 つ以上の他の変数を含むクエリ変数です。このセクションでは、連鎖変数の仕組みについて説明し、連鎖変数を使用するダッシュボードの例へのリンクを提供します。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべてのもので同じです。チェーン変数クエリは、それらを許可す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、深さ 5~10 レベルまで可能です。技術的には、どれだけ深く、複雑に行け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。

### リンクされた新しい変数の作成

- 変数を連鎖すると、親/子の依存関係が作成されます。これらは、はしごまたは木として構想できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数のベース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重複変数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 を追加できます。
- この方法で作成された新しい変数は、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

各エントリの右側にある上矢印と下矢印をクリックして、ダッシュボード可変リスト内の可変の順序を変更できます。Grafana は、このリストに従って変数ドロップダウンを左から右に一覧表示し、変数は左端の上部にあります。

- 子変数の前に、依存関係のない変数をリストします。
- 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、依存関係を持つ変数が表示されないことに注意してください。他のユーザー (および自身) に対して簡単に使用できるように、変数を論理的な順序で一覧表示します。



## 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、リンクされた一連の 4 つの変数 (国、リージョン、サーバー、メトリクス) があり、ルート変数の値 (国) を変更する場合、Grafana はダッシュボードの視覚化を更新する前に、すべての依存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、変数を追加したり、既存の変数を管理したりできます。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特定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：変数のクローンを作成するには、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、というプレフィックスが付いた元の変数の名前を持つ変数のコピーが作成されます `copy_of_`。

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。

## 正規表現で変数をフィルタリングする

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リスト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リスト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**02**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オプションをフィルタリングおよび変更し、テキストの一部を返します。

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個別の「テキスト」部分と「値」部分を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、選択できる各値の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\_namesPrometheus 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いです。そのため、次の変数クエリ結果が得られます。

```
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
```

次の正規表現を通過します。

```
/chip_name="(?(<text>[ ^ " ] + ) |chip="(?(<value >[ ^ " ] + )/g
```

次のドロップダウンリストを生成します。

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
```

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、依存関係マップを表示できます。依存関係マップを移動できます。マウスホイールまたは同等のもので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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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` – この構文は読みやすいですが、単語の途中で変数を使用することはできません。

例: `apps.frontend.$server.requests.count`

- `${var_name}` – 式の途中で変数を使用する場合は、この構文を使用します。
- `${var_name:<format>}` – この形式により、Grafana が値を解釈する方法をより詳細に制御できます。詳細については、「高度な変数形式オプション」を参照してください。
- `[[varname]]` – は使用しないでください。この構文は古く、廃止されました。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。補間中に、クエリ言語の構文とその使用場所に準拠するために、変数値をエスケープす

ることがあります。例えば、InfluxDB または Prometheus クエリの正規表現式で使用される変数は、正規表現をエスケープします。

## 高度な変数形式オプション

変数補間の形式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したい場合があります。

例えば、MySQL データソースのデフォルトは、複数の値を引用符で区切って結合することです 'server01', 'server02'。場合によっては、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: server01,server02。これは、以下に示す高度な変数フォーマットオプションで実現できます。

### 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、がデフォルト/フォールバックオプション `nglob` です。

### CSV

複数の値を持つ変数をカンマ区切りの文字列としてフォーマットします。

```
servers = [ 'test1', 'test2' ]
String to interpolate: '${servers:csv}'
Interpolation result: 'test1,test2'
```

### 分散型 - OpenTSDB

OpenTSDB のカスタム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:distributed}'
Interpolation result: 'test1,servers=test2'
```

### 二重引用符

単一値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、 によって各値をエスケープ\"し、各値を 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

複数の値を持つ変数をカンマ区切りの文字列として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")'
```

## パーセントエンコード

URL パラメータで使用する単一値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の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 クエリの一重引用符など、データソース固有の書式設定をオフにします。

```
servers = [ 'test.1', 'test2' ]  
String to interpolate: '${var_name:raw}'  
Interpolation result: 'test.1,test2'
```

## [Regex] (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)'
```

## シングルクオート

単一値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、 によって各値をエスケープ'し、各値を で引用符で囲みます'。

```
servers = [ 'test1', 'test2' ]  
String to interpolate: '${servers:singlequote}'  
Interpolation result: "'test1','test2'"
```

## SQLstring

単一値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、 によって各値をエスケープ''し、各値を で引用符で囲みます'。

```
servers = [ "test'1", "test2" ]  
String to interpolate: '${servers:sqlstring}'  
Interpolation result: "'test''1','test2'"
```

## [Text] (テキスト)

単一値および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、テキスト表現を と組み合わせた値を返します+。

```
servers = [ "test1", "test2" ]
String to interpolate: '${servers:text}'
Interpolation result: "test1 + test2"
```

## クエリパラメータ

単一値および複数値の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"
```

## ダッシュボードの使用状況の評価

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

使用状況インサイトを使用すると、Grafana インスタンスの使用方法をよりよく理解できます。

使用状況インサイト機能は、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、ダッシュボードでの使用状況インサイトデータの視覚化など、複数の機能にアクセスできます。



この機能は、Loki にエクスポートできる詳細なログも生成します。

## ダッシュボードとデータソースに関するインサイト

すべてのダッシュボードとデータソースについて、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、上部のバーにある Dashboard Insights をクリックします。

ダッシュボードインサイトには、次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に対する最後のアクティビティ (上限は 20)。

### データソースインサイト

データソースインサイトは、データソースが過去 30 日間に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. Insights 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見ると、最近ダッシュボードを操作したユーザーのアバターを表示するプレゼンスインジケータを通じて、誰が同じダッシュボードを見ているかがわかります。デフォルトの期間は 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。これにより、ダッシュボードのインサイトが開き、最近のユーザーアクティビティの詳細が表示されます。

### 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、最も使用頻度の高いダッシュボード、壊れたダッシュボード、未使用のダッシュボードを検索できます。

- エラー合計
- 30 日間のエラー
- ビューの合計
- 30 日間の表示

## 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードは、ダッシュボード名とパネルタイトルで検索できます。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任意の部分を入力します。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
- 大文字と小文字を区別しない

- 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。

**i** Tip

キーボードの矢印キーを使用して結果に移動し、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

タグは、特にダッシュボードの数が増えるにつれてダッシュボードを整理する優れた方法です。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。

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、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。

**i** Tip

キーボードのみを使用する場合は、tabキーを押してタグによる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 のパネルとビジュアライゼーション

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

パネルは、Grafana の基本的な視覚化構成要素です。各パネルには、パネルで選択されたデータソースに固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書式設定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。

ワークスペースに追加の[プラグイン](#)をインストールすることで、追加のパネルタイプが使用できるようになる場合があります。

- 特定のデータソースの使用の詳細については、「」を参照してください[データソースに接続する](#)。

### 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](#)
- [データリンクを設定する](#)
- [フィールド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](#)

- [ビジュアライゼーションの注釈付け](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 (歯車) アイコン:** クリックしてダッシュボード設定にアクセスします。
  - **破棄:** ダッシュボードを最後に保存してからパネルに加えた変更を破棄します。
  - **保存:** パネルに加えた変更を保存します。
  - **適用:** 行った変更を適用し、パネルエディタを閉じてダッシュボードに戻ります。適用された変更を永続化するには、ダッシュボードを保存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、データを表示できます。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 raw 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:** 視覚化プレビューは使用可能な領域を塗りつぶします。サイドペインの幅または下部ペインの高さを変更すると、使用可能なスペースを埋めるように視覚化が変更されます。
  - **実際:** 視覚化プレビューのサイズは、ダッシュボードのサイズとまったく同じになります。十分なスペースが利用できない場合、視覚化はアスペクト比を保持しながらスケールダウンします。
  - **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上位レベル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: アラートルールを記述します。
- パネル表示オプション: 表示オプションセクションには、データの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 raw データを表示したり、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たり、クエリリクエストを表示したり、パネルとデータ JSON をエクスポートしたりできます。

注: すべてのパネルタイプにすべてのタブが含まれてい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 raw 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が右側にドロワーを開きます。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 - クエリによって返された raw データを変換を適用して表示します。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と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 をプロビジョニングまたは管理する場合に便利です。
- クエリタブ - Grafana がデータソースをクエリしたときに送信された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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、次の方法について説明します。

- 編集用のパネルを開く
- パネル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したら、いつでもパネルを開いて、クエリの変更や更新、データ変換の追加、視覚化設定の変更を行うことができます。

1. 編集するパネルを含むダッシュボードを開きます。
2. パネルの任意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。

編集モードでパネルが開きます。

## 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の上部とダッシュボードに表示されます。

4. パネルと表示するデータの説明を書き込みます。

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ヒントに表示されます。

タイトルフィールド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、およびデータフレーム JSON モデルを調べ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. パネルの任意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、検査 > パネル JSON を選択します。
4. ソースの選択 フィールドで、次のいずれかのオプションを選択します。
  - パネル JSON: パネルを表す JSON オブジェクトを表示します。
  - パネルデータ: パネルに渡されたデータを表す JSON オブジェクトを表示します。
  - DataFrame 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、可変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、変数で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:

- クエリに複数值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:

- 繰り返したいパネルを編集します。
- 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。
- 方向 を選択します。
  - パネルを配置するには、水平を選択します side-by-side。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させることはできません。
  - 縦を選択して、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と同じです。
- 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ロードします。

## 標準オプションを設定する

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。

Grafana で使用されるデータモデルは、時系列とテーブル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、視覚化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 による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、つまりすべてのシリーズ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、数値を持つすべてのフィールドがパーセンテージで表示されます。

フィールドフォーマットオプションの完全なリストについては、「[標準オプションの定義](#)」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、編集してい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、編集してい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、視覚化には影響しません。

#### Note

すべてのビジュアライゼーションのオプションの追加と拡張に常に取り組んでいるため、すべてのビジュアライゼーションですべてのオプションを利用できるとは限りません。

## 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。

## カスタムユニット

単位ドロップダウンを使用して、カスタム単位、カスタムプレフィックス、またはサフィックスと日時形式を指定することもできます。

カスタム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。

- 値より後になるカスタム単位の `suffix:<suffix>`。
- `prefix:<prefix>` は、値より前にする必要があるカスタム単位です。
- `time:<format>` カスタム日付時刻形式の場合、例:YYYY-MM-DD。 [形式の](#) 構文とオプションについては、「形式」を参照してください。
- カスタム SI 単位の `si:<base scale><unit characters>`。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、少し高度なものです。したがって、ソースデータがミリ秒 (数千) で表される場合、単位の先頭にその SI スケール文字が付きます。
- カスタムカウント単位の `count:<unit>`。
- `currency:<unit>` カスタム通貨単位。

単位ピッカーにネイティブ絵文字を貼り付けて、カスタム単位として選択することもできます。

## 文字列単位

Grafana は、文字列を解析して数値として表示する際に、過度に攻撃的になることがあります。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、文字列単位を含む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。すべてのシリーズ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。すべてのシリーズ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。

## 10 進数

Grafana がレンダリング値に含める小数点以下数を指定します。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数を自動的に切り捨てます。例えば、1.1234 は 1.12 と表示され、100.456 は 100 と表示されます。

すべての小数を表示するには、単位を文字列 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、またはシリーズが表示される場合、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、タイトルにシリーズ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"} のラベルが与えられる

式の構文	例	へのレンダリング	説明
<code>\${_field.display_name}</code>	構文と同じ	Temp	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、ラベル部分の場合、Grafana は中括弧なしでラベルの値を表示します。
<code>\${_field.name}</code>	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)。
<code>\${_field.labels}</code>	構文と同じ	Loc "PBI"、S sor"3"	名前のないラベルを表示します。

式の構文	例	へのレンダリング	説明
<code>\${_field.labels.&gt;}</code>	<code>\${_field.labels.&gt;}</code>	PBI	指定されたラベルキーの値を表示します。
<code>\${_field.labels._values:}</code>	Syntax と同じ	PBI、3	ラベルの値をカンマで区切って表示します (ラベルキーなし)。

特定のフィールドの式をレンダリングした後に値が空の文字列である場合は、デフォルトの表示方法が使用されます。

## カラースキーム

色オプションとその視覚化への影響は、作業している視覚化によって異なります。一部の視覚化では、色オプションが異なります。

単一の色を指定するか、値に基づいて連続 (グラデーション) の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
単色	オーバーライドルールで便利な 1 つの色を指定する
しきい値から	一致するしきい値から色を取得するように Grafana に通知します
クラシックパレット	Grafana は、パレット内の色をシリーズインデックスで検索して色を割り当てます。グラフ、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
Green-Yellow-Red (値別)	連続したカラースキーム
Blue-Yellow-Red (値別)	連続したカラースキーム
Blues (値別)	連続したカラースキーム (パネルの背景から青色)
赤 (値別)	連続色スキーム (パネルの背景色から青色)
緑 (値別)	連続色スキーム (パネルの背景色から青色)
紫 (値による)	連続色スキーム (パネルの背景色から青色)

## 値なし

フィールド値が空または null の場合に表示される Grafana 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、視覚化用のデータを取得するクエリを介してデータソースと通信します。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 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は、独自の機能を利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を提供します。

クエリ言語が異なるため、各データソースクエリエディタは異なる検索と機能を行います。データソースによっては、クエリエディタによって自動補完機能、メトリクス名、変数の提案、または視覚的なクエリ構築インターフェイスが提供される場合があります。

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、その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、「」を参照してください [Enterprise データソースに接続する](#)。

## クエリ構文

データソースは、さまざまな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、その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>)
```

## 特殊なデータソース

Grafana には、Grafana、Mixed、Dashboard 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。

### クエリタブに移動する

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 — クエリするデータソースを選択します。
- クエリオプション： — 最大データ取り出し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： — expression builder 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、「[式クエリの書き込み](#)」を参照してください。

### クエリを追加する

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. [Queries] (クエリ) タブを開きます。
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 はデータソースをクエリし、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、クエリ为非表示、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ー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** – 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、または別の関数でそれらをまとめて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、視覚化された行を滑らかにしたりできます。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 はライブテールモードでデータをストリーミングします。

- **最小間隔** — 自動的に計算される間隔の最小制限を設定します。通常は最小スクレイプ間隔です。データポイントが 15 秒ごとに保存される場合、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、これをスクレイプ間隔よりも高い最小値に設定して、より粗く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ごとに集計またはグループ化するときを使用できる時間範囲を設定します。

Grafana は、テンプレート化されたクエリの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、これは間隔変数を使用する Prometheus クエリです `rate(http_requests_total[$__interval])`。

この自動間隔は、グラフの幅に基づいて計算されます。ユーザーが視覚化を拡大すると、間隔が大きくなり、より粗粒度の集計になります。同様に、ユーザーがズームインすると間隔が短くなり、よりきめ細かな集計になります。

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のタイムピッカーで選択されているものとは異なります。これを使用して、異なる期間または日からのメトリクスを同じダッシュボードに表示できます。

#### Note

パネルの時間オーバーライドは、ダッシュボードの時間範囲が絶対である場合は効果がありません。

例	相対時間フィールド
過去 5 分	now-5m
これまでの日付	now/d
過去 5 日間	now-5d/d
今のところ、この週	now/w
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

パネルの時間オーバーライドは、ダッシュボードの時間範囲が絶対である場合は効果がありません。

例	タイムシフトフィールド
過去 1 週間	1w/w
2 週間前	2w/w
過去 1 か月間	1M/M
この 1 年間	1d/y
過去 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にのみ表示 ) 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上書きします。この値を秒単位の数値として指定します。

## 式クエリの書き込み

**⚠** このドキュメントトピックは、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に設計されています。  
 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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

サーバー側の式を使用すると、数学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、クエリから返されるデータを数学やその他のオペレーションで操作できます。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のいくつかのマイナーデータ再構築を除いて、データソースによって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せん。

#### 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、さまざまなデータソースからのクエリを結合できるようにするか、データソースで使用できないオペレーションを提供することで、データソースを強化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的であるため、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、複数のシリーズを返すクエリで、各シリーズ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それぞれ、RefID と呼ばれる名前付き識別子である変数 (デフォルトの文字 A や など B) で表されます。

個々の式または別の式の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 種類のデータで機能します。

- 時系列のコレクション。
- 数値のコレクション。各数値は項目です。

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。データは一般的に、時系列データにラベル付けされていると見なされます。将来的には、式がエラーをより適切に処理できるように、クエリリターンタイプ (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、式とともに使用すると、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 1 つの時系列またはデータフレームごとに 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シリーズを返すデータソースを使用する場合、式で実行すると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、時間なしのデータフレーム、文字列列、および 1 つの数値列を返すテーブルレスポンスのみです。

Loc	ホスト	平均 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、値が 1 {"Loc": "MIA", "Host": "A"} の場合です。

## オペレーション

式では、数学、削減、再サンプリングのオペレーションを使用できます。

### 数学

Math は、時系列または数値データの自由形式の数式用です。数学演算では、数値と時系列を入力として受け取り、それらを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、ドル記号のプレフィックスが付いた RefID で参照されます \$A。例えば、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 ( のように先頭に 0 を付ける 072 )、または 16 進数 ( のように先頭に 0x を付ける ) にすることができます 0x2A。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 (+、バイナリおよび単項 -、 、 \*/%、 、指数 \*\* )、リレーショナル (<、 >、 、 ==、 <= ) !=>=、論理 (&&、 ||、 および単項 !) 演算子がサポートされています。

データに対するオペレーションの動作は、数値データか時系列データかによって異なります。

$\$A + \$B$  やなどのバイナリオペレーションでは  $\$A || \$B$ 、データの種別に応じて以下の方法で演算子が適用されます。

- $\$A$  と の両方  $\$B$  が数値の場合、オペレーションは 2 つの数値の間で実行されます。
- 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の間のオペレーションが実行されます。
- $\$A$  と の両方  $\$B$  が時系列データである場合、2 つのシリーズ内の各値間のオペレーションは、 $\$A$  と の両方に存在するタイムスタンプごとに実行されます  $\$B$ 。Resample オペレーションを使用して、タイムスタンプをラインアップできます。

## 概要 :

- 数値 OP 数値 = 数値
- OP シリーズ数 = シリーズ
- シリーズ OP シリーズ = シリーズ

式は 1 つの変数で表される複数のシリーズまたは数値で動作するため、バイナリオペレーションは 2 つの変数間の結合 (結合) も実行します。これは、個々のシリーズまたは番号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、{host=web01} に のようなラベルを持つ数値  $\$A$  と、同じラベル  $\$B$  を持つ に別の数値がある場合、各変数内のこれら 2 つの項目間でオペレーションが実行され、結果は同じラベルを共有します。このユニオン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 の両方  $\$B$  に 1 つの項目 (1 つのシリーズ、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。

- ラベルが正確な数学の場合、結合されます。
- ラベルが他の項目のサブセットである場合、例えば、 の項目\$Aにはラベルが付けられ{host=A,dc=MIA}、 の項目にはラベル\$Bが付けられ{host=A}、結合されます。
- などの変数内に項目ごとに異なるタグキー\$Aがある場合、結合動作は未定義です。

リレーショナル演算子と論理演算子は、true の場合は false 1 に対して 0 を返します。

## 数学関数

ほとんどの関数は独自の式オペレーションに存在しますが、数学オペレーションには数学演算子や記号に似た関数がいくつかあります。関数が数値またはシリーズのいずれかを取ることができる場合、引数と同じ型が返されます。シリーズの場合、シリーズ内の各ポイントの値に対して実行される のオペレーション。

### abs

abs は、引数の絶対値を返します。これは、数値またはシリーズです。たとえば、abs(-1)、abs(\$A) などです。

### is\_inf

is\_inf は数値またはシリーズを受け取り、Inf値 (負または正) 1 の場合は を返し、他の値0の場合は を返します。例えば「is\_inf(\$A)」のようにです。

#### Note

例えば、負の無限大を具体的に確認する必要がある場合は、 のような比較を行うことができます\$A == infn()。

### is\_nan

is\_nan は数値またはシリーズを受け取り、NaN値1の場合は を返し、他の値0の場合は を返します。例えば「is\_nan(\$A)」のようにです。この関数は、NaN が と等しくないために存在しますNaN。

### is\_null

is\_null は数値またはシリーズを受け取り、null値1の場合は を返し、他の値0の場合は を返します。例えば「is\_null(\$A)」のようにです。

### is\_number

`is_number` は数値またはシリーズを受け取り、1すべての実数値と他の値 (`null`、`Inf+`、`Inf-`) 0に対して を返しますNaN。例えば「`is_number($A)`」のようになります。

## ログ

ログは、数値またはシリーズである引数の自然対数を返します。値が 0 未満の場合、NaN が返されます。たとえば、`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`

丸めは、丸められた整数値を返します。例えば、`round(3.123)`、`round($A)` などです。

`ceil`

Ceil は、数値を最も近い整数値に切り上げます。たとえば、`ceil(3.123)` は 4 を返します。

`floor`

下限は、数値を最も近い整数値に切り下げます。例えば、`floor(3.123)` は を返します3。

## 削減

Reduce は、クエリまたは式から返された 1 つ以上の時系列を受け取り、各シリーズを 1 つの数値に変換します。時系列のラベルは、出力される各数値のラベルとして保持されます。

フィールド :

- 関数 — 使用するリダクション関数
- Input – 再サンプリングする変数 (`refID` ( などA))
- モード — シリーズに数値以外の値 (`null`、`NaN`、`+Inf`) が含まれている場合、リダクション関数の制御動作を許可します。

## 削減関数

Count (カウント)

カウントは、各シリーズのポイントの数を返します。



## 平均値

平均は、各シリーズのすべての値の合計をそのシリーズのポイント数で割った値を返します。strict モードで、シリーズ内のいずれかの値が null または nan の場合、またはシリーズが空の場合、NaN が返されます。

## 最小と最大

Min と Max は、シリーズ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、シリーズ内のいずれかの値が null または nan の場合、またはシリーズが空の場合、NaN が返されます。

## 合計

Sum は、シリーズ内のすべての値の合計を返します。シリーズの長さがゼロの場合、合計は 0 になります。strict モードで、シリーズに NaN 値または Null 値がある場合、NaN が返されます。

## 最終

Last は、シリーズ内の最後の数値を返します。シリーズに値がない場合、は NaN を返します。

## 削減モード

### 厳密

Strict モードでは、入力シリーズがそのまま処理されます。シリーズ内のいずれかの値が数値以外の場合 (null、NaN、または +-Inf)NaN が返されます。

### 非数値の削除

このモードでは、入力シリーズ内のすべての非数値 (null、NaN、または +-Inf) が除外されてから、リダクション関数が実行されます。

### 数値以外の を置き換える

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。

## 再サンプリング

再サンプリングでは、各時系列のタイムスタンプが一定の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再サンプリングして、それらの間で数学を実行できるようにするためです。これは、2つのシリーズをそれぞれ再サンプリングし、再サンプリングされた変数を参照する数学オペレーションで実行できます。

## フィールド :

- 入力 – 再サンプリングする時系列データの変数 (refID ( Aなど))
- 再サンプリング先 – 再サンプリングする期間は、例えば、10s. 単位はs秒、m分、h時間、d日、w週、年y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削減関数。動作の詳細については、「縮小オペレーション」を参照してください。
- Upsample – 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。
  - パッドは最後の既知の値で埋められます
  - 次の既知の値でバックフィルする
  - fillna で空のサンプルウィンドウを埋める NaNs

## 式を記述する

データソースがそれらを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. オペレーション フィールドで、書き込む式のタイプを選択します。
4. 式を記述します。
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シリーズまたは数値を返さない場合、式エンジンは を返しますNoData。例えば、リクエストに式によってマージされた 2 つのデータソースクエリが含まれている場合、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 になりますNoData。Grafana アラートがNoData結果を処理する方法の詳細については、「」を参照してください[データまたはエラーケースなしの処理](#)。

## クエリ結果を別のパネルと共有する

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana では、ダッシュボード内の他のパネルの 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。

この戦略では、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 に変更しますSource panel。このパネルを他のパネルのソースとして使用します。
3. 共有するクエリを定義します。

使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。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 Dashboard 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 で、作成した最初の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すると、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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する強力な方法です。変換を使用すると、次のことができます。

- フィールドの名前を変更する
- 時系列データを結合する
- クエリ間で数学演算を実行する
- ある変換の出力を別の変換への入力として使用する

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

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 はリストされている順序で変換を適用します。各変換は結果セットを作成し、処理パイプラインの次の変換に渡します。

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、Reduce 変換を使用して 1 つの列のすべての結果を 1 つの値に要約する場合、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手順を示しています。

変換をパネルに追加するには

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. [Transform] (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. 別の変換を適用するには、「変換を追加」を選択します。この変換は、前の変換によって返された結果セットに作用します。

## 変換をデバッグする

変換の入力セットと出力結果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。

入力結果セットと出力結果セットは、変換のデバッグに役立ちます。

## 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ユーザーに通知します。

変換を削除するには

1. 編集するパネルを開きます。
2. [Transform] (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

データに対して次の変換を実行できます。

### 計算からフィールドを追加する

この変換を使用して、他の 2 つのフィールドから計算され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 - 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、選択した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 行の値に基本的な数学演算 (合計、乗算など) 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 - 行モードを減らすを選択すると、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のリストが表示されます。使用可能な計算については、[「計算タイプ」](#)を参照してください。
- オペレーション - バイナリオプションモードを選択すると、オペレーション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、選択した 2 つのフィールドから 1 行の値に対して基本的な数学演算を実行できます。バイナリオペレーション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。これを空白のままにすると、フィールドには計算に一致する名前が付けられます。
- すべてのフィールドを置き換える - (オプション) 他のすべてのフィールドを非表示にし、計算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。

### フィールドを連結する

この変換は、すべてのフレームのすべてのフィールドを 1 つの結果に結合します。これら 2 つのクエリを検討してください。

クエリ A:

Temp	稼働時間
15.4	1230233

クエリ B:

アザフト語	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。

Temp	稼働時間	アザフト語	エラー
15.4	1230233	3.2	5

## クエリ結果からの設定

この変換により、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て、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してください。

## オプション

- Config 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 – 設定を適用するフィールドまたはシリーズを選択します。
- オプションに適用する – 通常、 に適用する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です。

## 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。

- フィールド — 使用可能なフィールドから選択
- as – 変換 FieldType する を選択します。
  - 数値 – 値数値の作成を試みます
  - 文字列 - 値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
    - yyyy-mm-dd や DD MM YYYY hh:mm:ss などの文字列で を入力 DateFormat として指定するオプションを表示します。
  - ブール値 – 値はブール値になります

例えば、次のクエリは、時刻フィールドを時刻、日付形式を YYYY として選択することで変更できます。

時間	マーク	値
7/1/2017	上記の	25
8/2/2018	以下	22
9/2/2019	以下	29
10/4/2020	上記の	22

結果：

時間	マーク	値
1/1/2017	上記の	25
1/1/2018	以下	22
1/1/2019	以下	29
1/1/2020	上記の	22

### 名前でデータを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。

Grafana は識別子フィールドを表示し、その後にクエリによって返されるフィールドが表示されます。

フィルターは、次の 2 つの方法のいずれかで適用できます。

- 正規表現式を入力します。
- フィールドを選択して、その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は濃い灰色のテキストで表示され、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場合は、複数のクエリがあるパネルでこの変換を使用します。

Grafana は、クエリ識別文字をダークグレーテキストで表示します。クエリ識別子をクリックしてフィルタリングを切り替えます。クエリ文字が白の場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示になります。

### Note

このデータソースは返されたデータとクエリの関連付けをサポートしていないため、Graphite ではこの変換は使用できません。

## 値でデータをフィルタリングする

この変換により、Grafana でデータを直接フィルタリングし、クエリ結果から一部のデータポイントを削除できます。定義した 1 つ以上の条件に一致するデータを含めるか除外するかを選択できます。条件は、選択したフィールドに適用されます。

この変換は、データソースが値で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は、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。

- 正規表現 - 正規表現式を一致させる
- Is Null - 値が null の場合に一致
- Is Not Null - 値が null でない場合に一致
- Equal - 値が指定された値と等しい場合に一致
- 差分 - 値が指定された値と異なる場合に一致します

数値フィールドで使用可能な条件は次のとおりです。

- Greater - 値が指定された値より大きい場合に一致します
- 下限 - 値が指定された値よりも小さい場合は一致
- Greater or equal - 値がより大きいか等しい場合は一致
- 下限または等しい - 値が以下である場合に一致
- 範囲 - 指定された最小値と最大値、最小値と最大値の間の範囲に一致

次のデータセットを検討してください。

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
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と、設定は次のようになります。

- フィルタータイプ – Include
- 条件 — がTemperature一致する行 Lower Than 30

また、30°C 未満の温度のみが含まれる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

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 31 分 22 秒	20	85
7/7/2020 午前 9 時 30 分 57 秒	19	101

フィルターには複数の条件を追加できます。例えば、高度が 100 より大きい場合にのみデータを含めることができます。そのためには、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条件が設定されたInclude行
- 条件 1 – が LowerよりもTemperature一致する行 30
- 条件 2 – が GreaterよりもAltitude一致する行 100

複数の条件がある場合は、すべての条件に一致する行にアクション (含める/除外する) を適用するか、追加した条件のいずれかに一致する行に適用するかを選択できます。

上記の例では、温度が 30 より低く、高度が 100 より大きい行を含めるため、すべて一致を選択しました。温度が 30 より小さい行や高度が 100 より大きい行を含める場合は、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 (最初の条件には一致しない) ですが、高度は 101 (2 番目の条件に一致する) であるため、含まれます。

無効または不完全な条件は無視されます。

### [Group by] (グループ化の条件)

この変換は、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。クリックして、計算の選択肢のリストを表示します。

オリジナルデータの例を次に示します。

時間	Server ID	CPU 温度	サーバステータス
7/7/2020 午前 11 時 34 分 20 秒	サーバー 1	80	シャットダウン

時間	Server 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 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

この変換は 2 つのステップで行われます。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、ソートした場合と同じフィールドの値がすべて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、データは次のようにグループ化されます。

時間	Server 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

時間	Server ID	CPU 温度	サーバーステータス
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

サーバー ID の値が同じ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したら、他のフィールドにさまざまな計算を追加し、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た平均計算を追加して、以下を取得できます。

Server 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:

- Time フィールドでは、Last 値を計算して、サーバーごとに最後のデータポイントがいつ受信されたかを確認できます。
- Server Status フィールドでは、Last 値を計算して、各サーバーの最終状態値を確認できます。
- フィールド温度については、最終値を計算して、各サーバーでモニタリングされる最新の温度を確認することもできます。

その後、以下を取得します。

Server ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最終)	サーバーステータス (最新)
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	オーバードロード
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、便利な方法で表示できます。

### フィールドで結合する

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。

### 内部結合

内部結合は、選択したフィールドのすべてのテーブルが同じ値を共有する複数のテーブルのデータをマージします。このタイプの結合は、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 つの結果にまとめ、結合が成功し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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	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	[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	稼働時間	[サーバー]	エラー
7/7/2020 午前 11 時 34 分 20 秒	node	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

## 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。内部結合は時間フィールドでクエリ A とクエリ B を結合しますが、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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	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	[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	稼働時間	[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		
7/7/2020 午前 11 時 34 分 20 秒	node	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。

- シリーズ 1 – ラベル Server=Server A、Datacenter=EU
- シリーズ 2 – ラベル Server=Server B、Datacenter=EU

列モードでは、結果は次のようになります。

時間	[サーバー]	Datacenter	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

「行」モードでは、結果にはシリーズごとにテーブルがあり、次のような各ラベル値が表示されません。

ラベル	value
[サーバー]	サーバー A
Datacenter	EU

ラベル	value
[サーバー]	サーバー B
Datacenter	EU

### 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、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	Datacenter	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、内部的に 2 つの別々の変換です。最初のは 1 つのシリーズで動作し、フィールドにラベルを抽出します。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、無効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、以下で詳しく説明します。

これを説明するために、ラベルが重複していない時系列を返すクエリが 2 つある例を示します。

- シリーズ 1 – ラベル Server=ServerA
- シリーズ 2 – ラベル Datacenter=EU

これにより、最初に次の 2 つのテーブルが作成されます。

時間	[サーバー]	値
7/7/2020 午前 11 時 34 分 20 秒	ServerA	10

時間	Datacenter	値
7/7/2020 午前 11 時 34 分 20 秒	EU	20

マージ後 :

時間	[サーバー]	値	Datacenter
7/7/2020 午前 11 時 34 分 20 秒	ServerA	10	
7/7/2020 午前 11 時 34 分 20 秒		20	EU

## Merge

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、値はマージできます。

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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	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	node	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	稼働時間
7/7/2020 午前 11 時 34 分 20 秒	node	15	25260122
7/7/2020 午前 11 時 24 分 20 秒	postgre	5	123001233

フィールドを整理する

この変換を使用して、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。

**Note**

この変換は、1つの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 は、クエリによって返されたフィールドのリストを表示します。次のようにできます。

- フィールドの上にカーソルを置くことで、フィールドの順序を変更します。カーソルが手になり、フィールドを新しい場所にドラッグできます。
- フィールド名の横にある目の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。

**値によるパーティション**

この変換により、複数のシリーズをグラフ化するときに、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 テーブルを考えてみましょう。

時間	リージョン	値
10/20/2022 午後 12 時 00 分 00 秒	米国	1520
10/20/2022 午後 12 時 00 分 00 秒	EU	2936
10/20/2022 午前 1 時 00 分 00 秒	米国	1327
10/20/2022 午前 1 時 00 分 00 秒	EU	912

v9.3 より前のバージョンでは、同じ TimeSeries パネルで米国の赤いトレンドラインと EU の青いトレンドラインを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把握する必要があります。

Partition by values transformer 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) の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 を使用していません。

```
SELECT Time, Region, Value FROM metrics WHERE Time > '2022-10-20'
```

時間	リージョン	値
10/20/2022 午後 12 時 00 分 00 秒	米国	1520
10/20/2022 午前 1 時 00 分 00 秒	米国	1327

時間	リージョン	値
10/20/2022 午後 12 時 00 分 00 秒	EU	2936
10/20/2022 午前 1 時 00 分 00 秒	EU	912

## 削減

Reduce 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。

入力について考えてみましょう。

クエリ A:

時間	Temp	稼働時間
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	アザフト語	エラー
7/7/2020 午前 11 時 34 分 20 秒	6.5	15
7/7/2020 午前 11 時 24 分 20 秒	3.2	5

リデュース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。
- フィールドを減らす - 既存のフレーム構造を維持しますが、各フィールドを 1 つの値にまとめます。

例えば、シリーズから行への変換で最初と最後の計算を使用した場合、結果は次のようになります。

フィールド	最初	最後
Temp	12.3	15.4
稼働時間	256122	1230233
アザフト語	6.5	3.2
エラー	15	5

最終計算の Reduce フィールドでは、それぞれ 1 行の 2 つのフレームが生成されます。

クエリ A:

Temp	稼働時間
15.4	1230233

クエリ B:

アザフト語	エラー
3.2	5

## 正規表現で名前を変更する

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 `web-01.example.com` に設定すれば `$1`、になります `web-01`。

## 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付けして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、結果のフィールドにより良い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。この表では、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*` オプション)。返されたデータに複数の行があるかどうかを選択する値を選択することもできます。

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。代わりに使用するフィールドの「Use as 列」で「Field name」を選択すると、このデフォルト設定を上書きできます。

- 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、代わりに使用するフィールドの「Use as 列」で「Field value」を選択して、このデフォルト設定を上書きできます。

データを視覚化する場合に便利です。

- Gauge
- Stat
- 円グラフ

追加フィールドをラベルにマッピングする

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	DataCenter	値
ServerA	米国	100
ServerB	EU	200

出力:

ServerA (ラベル : DataCenter: 米国 )	ServerB (ラベル : DataCenter: EU)
10	20

追加のラベルをフィールド表示名で使えるようになり、より完全なフィールド名が提供されます。

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、クエリ結果変換から設定を使用する必要があります。

例

Input:

名前	値	最大
ServerA	10	100
ServerB	20	200
ServerC	30	300

出力:

ServerA (設定: max=100)	ServerB (設定: 最大 = 200)	ServerC (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るのがわかります。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。Min、Max、Unit、Thresholds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。

時系列を準備する

時系列変換の準備は、使用するパネルでサポートされていない形式でデータソースが時系列データを返す場合に便利です。

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。

Wide time series オプションを選択して、時系列データフレームを long 形式から wide 形式に変換します。

## シリーズから行へ

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、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。ソースクエリでラベルを定義して、この値をカスタマイズします。

以下の例では、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

シリーズを行変換に適用した後の結果を次に示します。

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

## でソート

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` をチェックすると、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。

以下の例では、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

時間	メトリクス	値
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

値が「3」の制限変換を追加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

## クエリのトラブルシューティング

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、さまざまな結果が得られます。

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。

## クエリの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 でバグやビジュアライゼーション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、パネルの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のデータの検査」](#)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はいくつですか？ 多数のデータポイントを返すクエリは遅くなります。以下を試してください。

- クエリオプション で、返される最大データポイントを制限します。
- クエリオプション で、最小間隔時間を長くします。
- クエリでは、`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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このセクションでは、視覚化でのしきい値の使用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説明します。

## しきい値について

しきい値は、しきい値が満たされ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指定する値です。

しきい値は、クエリ結果に基づいてビジュアライゼーションを条件付きでスタイル付けおよび色付けする方法を 1 つ提供します。しきい値は、すべてではないものの、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。

しきい値を使用して、次のことができます。

- カラーグリッドの線またはグリッドは、[時系列の視覚化](#)の領域です。
- [時系列視覚化](#)の色線
- [Stat ビジュアライゼーション](#)の背景または値のテキストを色分けする
- [ゲージ視覚化](#)のゲージマーカーとしきい値マーカーの色付け
- [Geomap ビジュアライゼーション](#)の色マーカー
- [テーブルの視覚化](#)における色セルのテキストまたは背景
- [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する

しきい値には 2 つのタイプがあります。

- 絶対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 です。
- パーセントしきい値は、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

これをサポートする視覚化では、Grafana はデフォルトのしきい値を次のように設定します。

- 80 = 赤
- ベース = グリーン
- モード = 絶対

Base 値は、マイナス無限大を表します。通常は「良い」色です。

### しきい値の追加または削除

パネルには必要な数のしきい値を追加できます。Grafana はしきい値を最高から最低に自動的にソートします。

しきい値が事業運営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. 時系列パネルでは、しきい値を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して、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. パネルタブで、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には両方の値が必要です。
  - lt または gt - しきい値が適用される対象を示すには、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 は、その値の Y 軸に沿ってしきい値線を描画します。
  - 色 - 色に対応する条件を選択するか、独自の色を定義します。
    - custom - 塗りつぶしの色と行の色を定義します。
    - critical - 塗りつぶしと行の色は赤です。
    - warning - 塗りつぶしと行の色は黄色です。
    - ok - 塗りつぶしと行の色は緑色です。
  - 塗りつぶし - しきい値塗りつぶしを表示するかどうかを制御します。
  - Line - しきい値の行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。



## データリンクを設定する

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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

データリンクの変数を使用して、シリーズフィールド、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、「」を参照してください[データリンク](#)。

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。

#### **i** Note

これらの変数は 6.4 で変更されたため、古いバージョンの Grafana がある場合は、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ます。

データリンク URLs 「」を参照してください[ダッシュボード変数の追加と管理](#)。

### 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、「」を参照してください[グローバル変数](#)。

### シリーズ変数

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

## 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: 時間=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、リンクをクリックするすべてのユーザーの変数値を選択します。

`${var-myvar:queryparam}` - `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 つの値を選択しました	<code>var-myvar=value1</code>
複数の値を選択しました	<code>var-myvar=value1&amp;var-myvar=value2</code>
すべてを選択	<code>var-myvar=すべて</code>

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、リンクに詳細なコンテキストを提供できます。シリーズ名、またはカーソルの下に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た場合、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、バーゲージなどのパネルでは、視覚化上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。

データリンクの変数を使用して、保持されたデータフィルターを含む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、時間範囲、シリーズ、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。

### 先行入力の提案

データリンクを作成または更新するときは、キーボードの Cmd+Space または Ctrl+Space を押して先行入力候補を開き、URL に変数を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します。
3. データリンクを展開し、リンクを追加 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人間が読めるラベルです。
5. リンク先の URL を入力します。

ダッシュボードで定義されている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 を押して、使用可能な変数のリストを表示します。パネルリンクにテンプレート変数を追加することで、リンクはユーザーを適切なコンテキストに送信し、関連する変数は既に設定されています。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 をクリックして、変更をダッシュボードに保存します。

## データリンクを更新する

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、編集リンク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。

## データリンクを削除する

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。

## フィールドオーバーライドを設定する

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

オーバーライドを使用すると、特定のフィールドまたはシリーズの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、それぞれが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、正規表現一致の名前を持つフィールドを使用してオーバーライドを追加し、オーバーライドルールに Unit オプションを追加して、テキスト「バイト」を含むすべてのフィールドに単位を設定します。

### 例 1: 温度のフォーマット

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としましょう。

time	temperature
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。

time	temperature
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。

time	temperature
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: 温度と湿度のフォーマット

結果セットは、時間、高温度、低温度、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としましょう。

time	高温度	低温度	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

摂氏の単位を追加して、小数点以下を削除しましょう。これにより、次の表が表示されます。

time	高温度	低温度	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好に見えますが、湿度を変更する必要があります。このフィールドオプションオーバーライドを湿度フィールドに適用し、単位を Misc > パーセント (0 ~ 100) に変更することで、これを修正できます。

time	高温度	低温度	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シリーズ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。
  - 名前のフィールド：使用可能なすべてのフィールドのリストからフィールドを選択します。このセレクタを使用してルールに追加するプロパティは、この単一のフィールドにのみ適用されます。
  - 正規表現に一致する名前のフィールド：正規表現で上書きするフィールドを指定します。このセレクタを使用して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。
  - タイプのフィールド：文字列、数値などのフィールドをタイプ別に選択します。このセレクタを使用して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。このセレクタを使用して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を追加 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、フィールド内のホワイトテキストを削除します。
7. 上書きプロパティを追加 をクリックして、このフィールドに上書きを追加し続けます。または、上書きを追加 をクリックして、上書き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を削除する

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、上書き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、関連する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

パネル表示オプションでフィールドオーバーライドを表示できます。

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、上書きが表示されるまで下にスクロールします。

All タブに表示されるオーバーライド設定は、Overrides 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を編集する

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。行った変更はすぐに有効になります。

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、上書き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 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

値マッピングは、フィールドオーバーライドに加えて、視覚化に表示されるデータの視覚的な処理を変更するために使用できる手法です。



値マッピングによってマッピングされた値は、単位の書式をバイパスします。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 はパネルエ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、グラフ (古い)、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。

## 値マッピングのタイプ

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。
- **範囲**： 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、数値ではなく低または高を表示するように範囲値マッピングを設定できます。
- **正規表現**： 正規表現を置換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` を表示して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。
- **Special Maps** は、Null、NaN (数値ではない) などの特殊な値、および true や false 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、特別な値マッピングを設定できます。

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 を選択します。
4. 一致させる Grafana の値を入力します。

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. Update 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で、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、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. Update 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、正規表現 を選択します。
4. 一致させる Grafana の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. Update をクリックして、値マッピングを保存します。

### 特別な値をマッピングする

一般的でない値、ブール値、または空の値をフォーマットする場合は、特別な値をマッピングします。

1. 特別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、特別な を選択します。
4. 一致させる Grafana の特別な値を選択します。
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. Update 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で、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. 変更を加え、 の更新 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

パネルには、視覚化に表示されるデータを解釈するため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瞭さを追加します。

## 視覚化でのシリーズデータの分離

視覚化は視覚的に複雑で、多くのデータシリーズが含まれることがよくあります。視覚化からシリーズ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、表示する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。

視覚化でシリーズデータを分離するには

1. パネルを開きます。
2. 凡例で、分離するシリーズのラベルを選択します。

システムは、他のすべてのシリーズデータの表示から を削除します。

3. 分離されたシリーズにシリーズデータを段階的に追加するには、Ctrl キーまたは コマンドキーを押し、追加するシリーズ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ビューに戻すには、任意のシリーズラベルを 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](#)
- [Histogram](#)
- [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、凡例にシリーズデータ値を追加できます。必要な数だけ値を追加できます。変更を適用したら、凡例を水平方向にスクロールして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。

## シリーズの色の変更

デフォルトでは、Grafana はシリーズデータの色を指定し、これを変更できます。

### シリーズの色を変更するには

1. パネルを編集します。
2. 凡例で、シリーズに関連付けられている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カスタム色を選択します。
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。

## ソートシリーズ

凡例モードをテーブルに変更し、凡例に表示する [計算タイプ](#) のように選択できます。凡例テーブルの計算名ヘッダーを選択して、テーブルの値を昇順または降順にソートします。

ソート順序は、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シリーズ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 チャートのパネルでのみサポートされています。

## 計算タイプ

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

次の表に、Grafana で実行できる計算のリストを示します。これらの計算は、変換タブとバーゲージ、ゲージ、統計の視覚化で確認できます。

計算	説明
すべての null	すべての値が null の場合、true
すべてのゼロ	すべての値が 0 の場合、true
変更数	フィールドの値が変更される回数
カウント	フィールドの値の数
差分	値の累積変更、増分のみをカウント
違い	フィールドの最初と最後の値の違い
差分率	フィールドの最初と最後の値の変化率
個別カウント	フィールド内の一意の値の数
最初 (null ではない)	まず、フィールドの null 値ではありません
最大	フィールドの最大値
平均値	フィールド内のすべての値の平均値
分散	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 (ゼロ以上)	フィールドの最小値、正の値

計算	説明
[Range] (範囲)	フィールドの最大値と最小値の差
[ステップ]	フィールドの値間の最小間隔
合計	フィールド内のすべての値の合計

## ビジュアライゼーションの注釈付け

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

注釈は、グラフ上のポイントをリッチイベントで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、ネイティブの注釈ストアと、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、注釈を追加 をクリックします。
3. 注釈の説明を追加し、オプションでタグを追加します。
4. [Save] ( 保存 ) をクリックします。

または、注釈を追加するには、Ctrl+Clickまたは時系列パネルCmd+Clickに注釈の追加ポップアップが表示されます。

## リージョンアノテーションの追加

1. ダッシュボードCtrl+ClickCmd+Clickまたは時系列パネル。
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. 注釈の説明を追加し、オプションでタグを追加します。
4. [Save] ( 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. 注釈ツールヒントの編集 (鉛筆) アイコンをクリックします。
3. 説明を変更し、オプションでタグを変更します。
4. [Save] ( 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. 注釈ツールヒントのごみ箱アイコンをクリックします。

## 組み込みクエリ

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アノテーションクエリが原因です。この注釈クエリは、現在のダッシュボードから発生したすべての注釈イベントを取得し、作成されたパネルに表示します。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 (Dashboard cogs メニューを使用) を開き、 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Annotations & Alerts (Built-in)。

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 が表示されるため、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てタグでフィルタリングすると、それらを表示することができます。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

でフィルタリングを に設定することで、 -- Grafana --データソースを介してネイティブアノテーションストアからアノテーションを取得する新しいクエリを作成できますTags。



Grafana v8.1 以降のバージョンでは、既存のタグの先行タイプ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、タグを指定しますoutage。このクエリは、停止タグを持つすべての注釈 (ダッシュボードから、または API を介して) 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、を有効にするとMatch any、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 以降では、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、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るようになりました。

## 他のデータソースのクエリ

注釈イベントは、注釈クエリを通じて取得されます。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」ボタンを押します。

注釈クエリの名前を指定します。この名前はトグル (チェックボックス) に付けられ、このクエリの注釈イベントの表示を有効または無効にできます。例えば、Deploysと という名前の 2 つの注釈クエリがあるとしますOutages。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

注釈クエリのオプションは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、クエリリクエストを表示し、パネルとデータ JSON をエクスポートできます。

#### Note

すべてのパネルタイプにすべてのタブが含まれてい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 raw 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は、ペインの上部に「検査: [NameOfPanelBeingInspected] 」と表示します。右上隅の矢印をクリックして、ペインを展開または縮小します。
2. データタブ - クエリによって返された raw データに、変換が適用された状態を示します。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と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、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 をプロビジョニングまたは管理する場合に便利です。
5. クエリタブ - Grafana がデータソースをクエリしたときに送信された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。

#### raw クエリ結果をダウンロードする

Grafana は、そのデータへの変換を含むデータを含む CSV ファイルを生成します。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または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 をクリックします。
3. データ をクリックします。

パネルに複数のクエリまたは複数のノードが含まれている場合は、追加のオプションがありません。

- 結果の選択: 表示する結果セットデータを選択します。
  - データの変換
  - 時間で結合: 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1つの結果セットを表示します。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、Download for Excel トグル をクリックします。
  6. CSV のダウンロード 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、送信するクエリの数、および返される行数を示す統計が表示されます。この情報は、特に数値が予期せず高い場合や低い場合に、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 をクリックします。
3. 統計 をクリックします。

統計は読み取り専用形式で表示されます。

### クエリの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をトラブルシューティングする場合、または期待される結果を返さない場合は、クエリのリクエストとレスポンスの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 をクリックします。
3. の更新 をクリックします。

パネルに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、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana には、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が用意されています。ドキュメントのこのセクションでは、組み込みパネル、それらのオプション、一般的な使用方法について説明します。

使用を開始し、パネルの使用の基本を学ぶための一般的なパネルは、[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](#)
- [バーゲージ](#)
- [キャンドルスティックパネル](#)
- [Canvas パネル](#)
- [クロックパネル](#)
- [ダッシュボードリスト](#)
- [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](#)
- [プロットパネル](#)
- [サンキーパネル](#)
- [散布図](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートリストパネルにダッシュボードアラートが表示されます。現在の状態または最近の状態の変化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、「」を参照してください[Grafana バージョン 9 のアラート](#)。

これらの設定を使用して、視覚化を絞り込みます。

### オプション

- グループモード – 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する場合はデフォルトのグループ化、カスタムラベルセット別にアラートインスタンスをグループ化する場合はカスタムグループ化を選択します。

- 最大アイテム数 — リストするアラートの最大数を設定します。
- ソート順 – 表示されるアラートの順序付け方法を選択します。
  - アルファベット順 (asc) – アルファベット順
  - アルファベット (desc) — アルファベットの逆順
- 重要度 — 次の値に従って重要度が異なり、1 が最も高くなります。
  - 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 – アラートリストがあるダッシュボードからのアラートのみを表示します。

## フィルター

次のオプションを使用して、選択したクエリ、フォルダ、またはタグと一致するようにアラートをフィルタリングします。

- 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-.\*"} です。
- 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状態を選択します。

- アラート/ファイリング
- 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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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リストが表示されます。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するには、さまざまなオプションを使用できません。

### 注釈クエリ

以下のオプションは、注釈のリストの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配置されてい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。以下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。以下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使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

limit 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、注釈パネルの表示に含まれる追加のメタデータを制御します。

### ユーザーを表示する

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。

### タグを表示する

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。注: タグを使用して、パネル自体の注釈リストをライブフィルタリングできます。

### リンク動作

#### リンクターゲット

このオプションを使用して、注釈付きデータの表示方法を選択します。以下のオプションがあります。

- パネル - このオプションは、対応する注釈を含むパネルの全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、ダッシュボード全体のコンテキストで注釈に焦点を当てます。

### より前の時間



このオプションを使用して、注釈の前の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。

## 以降の時間

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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このパネルの視覚化により、カテゴリ別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

サポートされているデータフレームは 1 つだけで、X 軸または Y 軸のカテゴリ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。データ形式の例を次に示します。

ブラウザ	マーケットシェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合、パネルにはグループ化された棒が表示されます。

## 時系列または複数の結果セットの視覚化

時系列またはテーブルが複数ある場合は、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、Reduce 変換を追加し、計算で Max と Last as オプションを指定します。

## 棒グラフオプション

視覚化を絞り込むには、以下のオプションを使用します。

### [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。
- Vertical – Y 軸をカテゴリ軸にします。

X 軸ティックラベルの最大長は、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切り捨てられます (省略記号付き)。

棒ラベルの最小間隔は、棒ラベル間の最小間隔を設定します。

### 値を表示する

棒の左上に値を表示す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。
- Always – 常に値を表示します。
- Never – 値を表示しないでください。

### スタッキング

棒グラフの積み上げを制御します。

- オフ – バーはスタックされません。
- 正常 – 棒は互いに積み重なります。
- パーセント – 棒は互いに積み重なり、各棒の高さはスタックの合計の高さに対する割合です (すべての棒スタックは同じ高さになり、最大 100% 増加します)。

グループ幅はグループの幅を制御します。0=最小、1=最大幅。

棒幅は棒の幅を制御します。0=最小、1=最大幅。

バー半径はバーの半径を制御します。0 = 最小、0.5 = 最大半径。

ポインターのあるバーにカーソルを合わせると、バーの周囲エリアが強調表示されている場合、ホバーコントロール上の全エリアが強調表示されます。

線幅は、棒の線幅を制御します。

塗りつぶし不透明度は、バー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は、グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – 勾配フィルなし、これがデフォルト設定です。
- 不透明度 — 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。
- Hue – グラデーション色は、線の色相に基づいて生成されます。

ツールヒントモード – 視覚化の上にカーソルを置くと、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、視覚化にカーソルを合わせているシリーズが 1 つだけ表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – 視覚化を操作するときツールヒントを表示しないでください。

#### Note

オーバーライドを使用して、ツールヒントから個々のシリーズ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、視覚化における凡例の表示方法を絞り込みます。詳細については、「[凡例を設定する](#)」を参照してください。

- List – 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

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、視覚化には影響しません。

- 配置 — Y 軸の配置を設定します。
- 自動 – Grafana は Y 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に、右軸を次の単位に割り当てます。
- 左 – 左側にすべての Y 軸を表示します。
- 右 – 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 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、軸が視覚的に互い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量を比較しやすくなります。

- 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、ソフト最小値とソフト最大値の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は、データがほぼフラットなときにリップが山に変わるのを防ぎ、標準の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派生したハード最小または最大フィールドオプションによって、定義されたポイントを超えてスパイクをクリップすることで、断続的なスパイクが有用な詳細を平坦化することを防ぐことができます。

標準の最小/最大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。

## バーゲージ

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

バーゲージは、すべてのフィールドを 1 つの値に減らすことで、データを簡素化します。Grafana による削減の計算方法を選択します。

このパネルには、クエリが返すシリーズ、行、または列の数に応じて、1 つ以上の棒グラフを表示できます。

### 値オプション

以下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 — すべての行に基づいて計算された値を表示します。

- 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リデューサー関数を選択します。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。

### バーゲージオプション

バーゲージの表示方法を調整します。

方向 — スタッキング方向を選択します。

- 自動 – Grafana は、最適な向き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縮します。
- Vertical – バーは垂直方向に、下から上まで伸縮します。

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。
- リトロ LCD – ゲージは、ライトまたはライトのないスモールセルに分割されます。
- 基本 – 一致するしきい値に基づく単一色。

塗りつぶされていない領域を表示する – バーの塗りつぶされていない領域を濃い灰色でレンダリングする場合は、このオプションを選択します。Retro LCD ディスプレイモードには適用されません。

### 最小幅

バー列の最小幅を垂直方向に制限します。

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。

### 最小の高さ

水平方向の棒行の最小の高さを制限します。

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。

### キャンドルスティックパネル

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 8 での作業](#)。

キャンドルスティックパネルを使用すると、価格の動きに重点を置い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。キャンドルスティックパネルには、オープン/ハイ/ロークローズ (OHLC) 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。

キャンドルスティックパネルは の基礎に基づいて構築 [時系列パネル](#) されており、多くの一般的な構成設定が含まれています。

## Mode (モード)

モードオプションを使用すると、視覚化に使用するディメンションを切り替えられます。

- キャンドルは、パネルの寸法を、キャンドルスティックの視覚化で使用されるオープン、ハイ、ロー、クローズの寸法に制限します。
- ボリュームは、パネルのディメンションをボリュームのディメンションに制限します。
- どちらも、キャンドルスティックパネルのデフォルトの動作です。これには、キャンドルスティックとボリュームの両方の視覚化が含まれます。

## キャンドルスタイル

- キャンドルはデフォルトの表示スタイルであり、オープンディメンションとクローズディメンションの間にキャンドルスタイルの視覚化を作成します。
- OHLC バーには、オープン、ハイ、ロー、クローズの 4 つのコアディメンションが表示されます。

## カラー戦略

- Open がデフォルトの動作であるため。このモードは、期間内価格の変動が正の場合、Up 色 (下) 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 色が使用されます。
- 「事前クローズ」は、期間間の価格の動きまたは値の変化に基づいてキャンドルの色が決ま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 色が使用されます。オープン時の値がクローズ時の以前の値よりも小さい場合、ダウン色が使用されます。このオプションでは、燭台視覚化モードもトリガーされます。低めのキャンドルスティックは、ピリオド内の動きが正 (値はオープンに近いほど高い) であることを示します。塗り分けされたキャンドルスティックは、ピリオド内の変化が負 (値はオープンに近いほど低い) であることを示します。詳細については、 [「 の違いの説明 」](#) を参照してください。

## アップカラーとダウンカラー

アップカラーおよびダウンカラーオプションは、価格の変動がアップまたはダウンするときを使用する色を選択します。上記の色戦略は、期間内または期間間の料金移動を使用して、キャンドルまたは OHLC バーの色を選択するかどうかを決定します。

オープン、高、低、クローズ

キャンドルスティックパネルは、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ています。Open、High、Low、Close オプションを使用すると、パネルが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にデータを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。

- Open は、指定された期間の開始値に対応します。
- High は、指定された期間の最大値に対応します。
- Low は、指定された期間の最小値に対応します。
- クローズは、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、指定された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

キャンドルスティックパネルは時系列パネルに基づいています。オープン、ハイ、ロー、クローズ、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。インクルードオプションと無視オプションを使用すると、パネルは、で使用可能な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、他のインクルードデータを視覚化できます [時系列パネル](#)。

Canvas パネル

-  このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。



Canvas は、Grafana の能力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 の視覚化は、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、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を設計し、データをオーバーレイできます。これらはすべて Grafana の UI 内で実行できます。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、Canvas パネルの設計は非常に慣れていているように見えます。

## [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。この要素には一意の「編集」モードがあり、コンテキストメニューの「編集」オプションまたはダブルクリックでトリガーできます。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。

## [Text] (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。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のいずれかによってトリガーされる編集モードもサポートしています。

### 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 に新しい編集エクスペリエンスが導入されました。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、オ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、背景画像を設定し、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、要素のレイヤーの配置を変更したりできます。

## Canvas 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。キャンバスパネルをオフにすると、ロックされ、要素が所定の位置にフリーズされ、意図しない変更が防止されます。

### クロックパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。

クロックパネルには、現在の時刻またはカウントダウンが表示されます。毎秒更新されます。

- モード – デフォルトは時間 です。カウントダウン を選択した場合は、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 – 時刻を表示するオプションは 12 時間形式と 24 時間形式です。
- Timezone – タイムゾーンは、Monce Timezone ライブラリによって提供されます。デフォルトはコンピュータのタイムゾーンです。
- カウントダウン期限 – モードをカウントダウン に設定している場合は、カウントダウンす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を表示す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、重み、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、秒数がティックダウンしない場合は、24 時間制hh:mmの場合は時刻形式を、12 時間制h:mm Aの場合は時刻形式を変更します。オプションの完全なリストについては、「[「」の表示](#)」を参照してください。
- Bg Color – クロックの背景色を選択します。

## ダッシュボードリスト

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ダッシュボードリストの視覚化では、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するように設定できます。

ダッシュボードがロードされるたびに、このパネルはダッシュボードリストをクエリし、常に最も多く up-to-date の結果を提供します。

## オプション

これらのオプションを使用して、視覚化を絞り込みます。

- Starred – スタード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、変数補間がサポートされています。例えば、`$my_var`または `です${my_var}`。
- 見出しの表示 – 選択したリスト選択 (Starred、Recently viewed、Search) 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、Starred および Recently で表示されたダッシュボードを表示すると、パネルには各セクションに 10 個ずつ、最大 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、部分的な値が受け入れられます。
- 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する場所を次に示します。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は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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ゲージは、各シリーズ、列、または行で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

以下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

すべての行に基づいて計算値を表示します。

- 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リデューサー関数を選択します。使用可能な計算のリストについては、「」を参照してください [計算タイプ](#)。
- 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。

## Gauge

ゲージの表示方法を調整します。

- 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 — しきい値バンドが内部ゲージ値バンドの外部に表示され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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Geomap パネルの視覚化では、地理空間データを使用してワールドマップを表示および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ベースの特性に簡単に焦点を合わせることができます。

## 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、GeoMap パネルが最初にロードされたときに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。
  - データに適合は、マップレイヤーのデータ範囲とデータ変更時の更新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」、単一の「レイヤー」、または選択したレイヤーの「最後の値」のデータに基づいて範囲を選択できます。
    - レイヤーの単一の「レイヤー」または「最後の値」からのデータを適合させる場合は、レイヤーを選択できます。
    - パディングは、データの範囲を超えて相対的な割合でパディングを設定します（「最後の値」のみを見ると使用できません）。
    - 最大ズームは、データをフィット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。
    - 緯度
    - 経度
  - デフォルトビューには、次のようなものもあります。
    - (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
- ズームは初期ズームレベルを設定します。

## マップレイヤー

Geomap の視覚化では、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、ベースマップの上に地理空間データを視覚化する方法を決定します。

## タイプ

Geomap 視覚化では、3 つのマップレイヤータイプから選択できます。

- [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。

現在アルファ状態にあるレイヤータイプは 5 つもあります。

- [ナイト/デイレイヤー \(アルファ\)](#) は、ナイトリージョンまたはデイリージョンをレンダリングします。
- 最後のポイント (アルファ) のアイコンは、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。
- 動的 GeoJSON (アルファ) は、クエリ結果に基づいて GeoJSON ファイルをスタイル設定します。
- ルート (アルファ) 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、レイヤーの順序変更、削除を行うことができます。

- レイヤーを追加すると、Geomap 視覚化用の設定可能なデータレイヤーが追加されます。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、削除、順序を変更することができます。
  - レイヤー名を編集 (鉛筆アイコン) すると、レイヤーの名前が変更されます。
  - ごみ箱はレイヤーを削除します。
  - 並べ替え (6 つのドット/ラバーハンドル) 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時にレイヤーの順序を更新し、レイヤーの順序の選択を簡素化します。

1 つの Geomap 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

Geomap パネルには、地理的データのソースが必要です。このデータはデータベースクエリから取得され、データには 4 つのマッピングオプションがあります。

- 自動は位置データを自動的に検索します。このオプションは、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に使用します。
  - geohash: 「geohash」
  - 緯度: 「緯度」、「緯度」
  - longitude: 「longitude」、「lng」、「lon」
  - ルックアップ: 「ルックアップ」
- 座標は、クエリが座標データを保持す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に求められます。
- Geohash は、クエリが geohash データを保持するように指定します。データベースクエリから geohash の文字列データフィールドを選択するように求められます。
- Lookup は、クエリが値にマッピングする必要がある場所名データを保持するように指定します。データベースクエリとガゼッターからルックアップフィールドを選択するように求められます。



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、四角、三角形、星などのさまざまなマーカー形状として表示できます。

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。

- マーカーカラーはマーカーの色を設定します。デフォルトでは、すべてのポイントが 1 つの色 Single color に保たれます。データポイントの値としきい値が Thresholds セクションに設定されているかに応じて、複数の色を持つ代替オプションがあります。
- マーカーサイズは、マーカーのサイズを設定します。デフォルトは `Fixed` です。これにより Fixed size、データポイントに関係なく、すべてのマーカーサイズが同じになります。ただし、円を対応するデータポイントにスケールリングするオプションもあります。Min と Max マーカーのサイズは、マーカーレイヤーがこの範囲内でスケールリングできるように設定する必要があります。
- マーカーシェイプを使用すると、シェープ、アイコン、またはグラフィックを選択して、データに追加のビジュアルコンテキストを提供できます。シンプルなシェイプや 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。イメージアセットを含む URL を指定することもできます。イメージはスケーラブルベクトルグラフィック (SVG) である必要があります。
- 塗りつぶし不透明度は、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 Layer 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。

と同様に Markers、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。

- 重み値は、ヒートマップクラスターの強度を設定します。は、すべてのデータポイントにわたって一定の重み値 Fixed value を維持します。この値は 0~1 の範囲である必要があります。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リングする代替オプションがあります。

- 半径は、ヒートマップクラスターのサイズを設定します。
- Blur 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 は、Grafana に付属する GeoJSON ファイルの選択肢を提供します。
- デフォルトのスタイル 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。
  - カラーはデフォルトスタイルの色を設定します。
  - 不透明度はデフォルトの不透明度を設定します
- スタイルルールは、特徴プロパティに基づいてスタイルを適用します
  - ルールでは、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。
  - 色は、現在のルールのスタイルの色を設定します。
  - 不透明度は、現在のルールの透明度レベルを設定します。
- スタイルルールを追加すると、追加のスタイルルールが作成されます。

## 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

- テーマ
  - テーマは、ライトテーマ、ダークテーマ、または自動テーマのいずれかを選択します。
- ラベルを表示すると、マップの上部に国の詳細が表示されます。
- 不透明度 0 (透明) から 1 (不透明)

## 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、[「Tiled Web Maps」](#) および [「Open Street Map Tile Servers のリスト」](#) を参照してください。

## オプション

- URL テンプレート

**Note**

有効なタイルサーバー URL を  $\{z\}/\{x\}/\{y\}$  で設定します。例: `https://tile.openstreetmap.org/{z}/{x}/{y}.png`。

- 属性は、[マップコントロール](#)に表示される場合、レイヤーの参照文字列を設定します。
- 不透明度 0 (透明) から 1 (不透明)

## オープンストリートマップレイヤー

Open [Street Map からのマップ](#)、共同で無料の地理的世界データベース。

## オプション

- 不透明度 0 (透明) から 1 (不透明)

## ArcGIS レイヤー

[ArcGIS](#) レイヤーは、[ESRI](#) ArcGIS MapServer のレイヤーです。

## オプション

- 次のマップタイプから選択するサーバーインスタンス。
  - ワールドストリートマップ
  - ワールドイメージ
  - ワールド物理
  - トポロジ
  - 米国トポロジ

- 世界海
- カスタム MapServer (フォーマットについては「[XYZ](#)」を参照)
  - URL テンプレート
  - 属性
- 不透明度 0 (透明) から 1 (不透明)

### ナイト/デイレイヤー (アルファ)

ナイト/デイレイヤーには、現在の時間範囲に基づいてナイトリージョンとデイリージョンが表示されます。

#### Note

詳細については、「[- の拡張機能](#)」を参照してください [OpenLayers DayNight](#)。

### オプション

- パネルの時間範囲から時間ソースを切り替えるを表示する
- ナイトリージョンの色がナイトリージョンの色を選択する
- ディスプレイの太陽が太陽アイコンを切り替える
- 不透明度 0 (透明) から 1 (不透明)

### 写真レイヤー (アルファ)

Photos レイヤーは、各データポイントに写真を表示します。

#### Note

詳細については、「[Extensions for OpenLayers - Image Photo Style](#)」を参照してください。

### オプション

- イメージソースフィールド

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。

- イメージ URLs
- Base64 でエンコードされたイメージバイナリ (data:image/png;base64,...)
- タイプ

イメージのフレームスタイルを選択する

- [Square] (平方)
- [Circle] (円形)
- アンカー
- フォリオ
- 画像が収まるようにトリミングされた場合のトリミングトグル
- シャドウが画像の背後にあるボックスシャドウを切り替える
- 境界線は、イメージの周囲に境界サイズを設定します。
- 境界線の色は、画像の周囲に境界線の色を設定します。
- 半径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。

#### ズームコントロールを表示する

左上隅にズームコントロールを表示します。

#### マウスホイールズーム

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をオンまたはオフにします。

#### 属性を表示する

マップ上にベースマップレイヤーの属性を表示します。

#### スケールを表示する

左下隅にスケール情報を表示します。

**Note**

[m]/[km] で単位を表示します。

### メジャーツールを表示する

右上隅に測定ツールを表示します。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する
- クリックして測定を続行する
- ダブルクリックして測定を終了する

**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。

### 長さ

ジオメトリの球面長を取得します。この長さは、座標間の円距離の合計です。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」にあると見なされません。

長さの測定には、次の単位を選択できます。

- メトリクス (m/km)
- フィート (ft)
- マイル (mi)
- ノーティカルマイル (nmi)

### エリア

ジオメトリの球面面積を取得します。この領域は、ポリゴンエッジが球上の大きな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」にあると見なされます。

エリア測定値には、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 (エーカー)
- ヘクタール (ha)

### デバッグを表示する

マップの右上隅にデバッグ情報を表示します。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。

- ズームはマップの現在のズームレベルを表示します。
- センターには、現在の経度とマップセンターの緯度が表示されます。

### Tooltip

- None は、データポイントがクリックされた場合にのみツールヒントを表示します。
- ポインタがデータポイントにカーソルを合わせると、詳細にツールヒントが表示されます。

### グラフパネル

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

グラフパネルは、線、ドットのパス、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のに十分な汎用性を備えています。

## データおよびフィールドオプション

グラフの視覚化を使用する場合は、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

視覚化を絞り込むには、次の設定を使用します。

- 棒グラフ - 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。
- 線幅 - シリーズの線の幅を指定します。デフォルトは 1 です。
- Staircase - 隣接するポイントを踏み台として描画します。
- エリアフィル - シリーズのカラーフィルの量を指定します。デフォルトは 1 で、0 はなしです。
- フィル勾配 — エリアフィルの勾配の度合いを指定します。デフォルトは 0 で、勾配はありません。10 は急勾配です。
- 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。
- アラートのしきい値 - パネルにアラートのしきい値とリージョンを表示します。

## スタックと null 値

- スタック — 各シリーズが別のシリーズの上に積み上げられます。
- パーセント — 各シリーズがすべてのシリーズの合計に対するパーセンテージで描画されます。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。以下の注意事項を参照してください。
  - connected - シリーズ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、その行はギャップをスキップし、次の NULL 以外の値に接続します。
  - null シリーズにギャップがある場合、つまり null 値の場合、グラフの線は折られ、ギャップが表示されます。これはデフォルトの設定です。



- null as zero – 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て、負荷が 100% に達すると、サーバーはロックアップされ、エージェント送信統計はロード統計を収集できません。これによりメトリクスにギャップが発生し、デフォルトの null 設定を使用すると、Amazon Managed Grafana はギャップを表示し、何か問題があることを示します。これを接続されたに設定すると、このシグナルを見逃しやすくなります。

## ホバーツールヒント

これらの設定を使用して、グラフの視覚化を一時停止し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 — ホバーツールヒントに表示されるシリーズの数を決定します。
  - すべてのシリーズ — ホバーツールヒントには、グラフ内のすべてのシリーズが表示されます。ツールヒントのシリーズリストで、Grafana ワークスペースは一時停止しているシリーズを太字で強調表示します。
  - 単一 — ホバーツールヒントには、グラフで一時停止しているシリーズが 1 つだけ表示されます。
- ソート順 — すべてのシリーズモードを選択した場合、ホバーツールヒントのシリーズの順序をソートします。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、ユーザーは最大値または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、目的のデータを見つけやすくなります。
  - なし — ツールヒントのシリーズの順序は、クエリのソート順序によって決まります。例えば、シリーズをシリーズ名でアルファベット順にソートできます。
  - 増加 — ホバーツールチップのシリーズが値と増加順にソートされ、リストの上部に最小値が表示されます。
  - Decreasing — ホバーツールチップのシリーズが、値と降順でソートされ、リストの上部に最大値が付きます。

## 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、グラフパネル内のシリーズを他のシリーズとは異なるレンダリングにすることができます。表示オプションは、シリーズごとにカスタマイズすることも、正規表現ルールを使用してカスタマイズすることもできます。例えば、1つのシリーズで線の幅を太くして目立たせたり、右のY軸に移動したりできます。

複数のシリーズオーバーライドを追加できます。

シリーズオーバーライドを追加するには

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 という名前の2つのシリーズに一致しません `Network in`。

3. `+` を選択し、シリーズに適用するスタイルを選択します。各エントリに複数のスタイルを追加できます。

- 棒グラフ - シリーズを棒グラフとして表示します。
- 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 — 面塗りつぶしを含む折れ線グラフを表示します。
- フィル勾配 — エリアフィル勾配量を指定します。
- 線幅 - 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 値を無視するか、ゼロに置き換えます。これは、データのギャップを無視する場合に重要です。
- 以下を に塗りつぶす — 2つのシリーズ間の領域を塗りつぶします。
- 踏み台線 - シリーズを踏み台線として表示します。
- Dashes - ダッシュを含む行を表示します。
- 隠しシリーズ - シリーズを非表示にします。
- ダッシュの長さ - 線のダッシュの長さを設定します。
- ダッシュスペース - 行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 - シリーズを個別のポイントとして表示します。
- ポイント半径 — ポイントレンダリングの半径を設定します。

- 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 - Y 軸シリーズを設定します。
- Z-index - シリーズ z-index (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、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 — 値を負に変換して Y 軸の下にレンダリングします。
- 凡例 - 凡例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、視覚化での軸の表示を制御します。

### 左 Y/右 Y

オプションは両方の y 軸で同じです。

- 表示 — 軸を表示または非表示にすることを選択します。
- 単位 - y 値のディスプレイ単位を選択します。
- スケール - y 値に使用するスケールを選択します。線形、対数です。デフォルトは線形です。
- Y-Min - y の最小値。デフォルトは自動です。
- Y-Max - Y の最大値。デフォルトは自動です。
- 10 進数 - y 値に対して表示される小数の数を定義します。デフォルトは自動です。
- Label - Y 軸ラベルを指定します。デフォルトは「」です。

### Y 軸

- 整列 — 左右の y 軸を値で整列させます。デフォルトはチェックなし/false です。
- レベル - Y=0 から始めて、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、Align が選択されているときに使用できます。

### X 軸

- 表示 — 軸を表示または非表示にすることを選択します。

- 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。1つの に 3つのパネルがあるようなものです。メインモードは、X軸に時間がある時系列モードです。他の2つのモードは、時間モードとヒストグラムモードではなく、X軸にシリーズを持つ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 (時間単位や分単位など) でグループ化されます。
- シリーズ – データは時間ではなく、シリーズごとに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、数値をバケットまたはビンと呼ばれる範囲にグループ化する棒グラフの一種です。タラーバーは、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、「」を参照してください[ヒストグラムとヒートマップの概要](#)。

- バケット – 値をグループ化するバケットの数を設定します。空のままにすると、Amazon Managed Grafana は適切な数のバケットを計算しようとします。
- X-Min – この最小制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-Max – 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、視覚化での凡例の表示方法を絞り込みます。

## オプション

- 表示 – クリアすると凡例が非表示になります。デフォルトが選択されています (true)。
- テーブル – 選択すると、テーブルに凡例が表示されます。デフォルトがチェックされます (true)。
- 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、右に を選択した場合に使用できます。

## [値]

凡例名と一緒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。

- Max – メトリクスクエリから返される最大値。
- 平均 — メトリクスクエリから返される平均値。
- Current – メトリクスクエリから返された最後の値。
- 合計 — メトリクスクエリから返されたすべての値の合計。
- 10 進数 – 凡例値とグラフホバーツールヒントに表示される小数点数。

Amazon Managed Grafana は、クライアント側で凡例値を計算します。これらの凡例は、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。上記の凡例の値はすべて、同時に正しい値にすることはできません。

例えば、平均をアグリゲータとして使用している可能性が高いリクエスト/秒のレートをプロットした場合、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。

### シリーズを非表示にする

メトリクスクエリからのシリーズのすべての値が特定の値である場合、シリーズを非表示にします。

- null のみを使用 – Value=null (デフォルトはチェックされていません)
- ゼロのみを使用 – 値 = ゼロ (デフォルトはチェックされていません)

### 時間リージョン

グラフで特定の時間帯を強調表示して、週末、営業時間、勤務時間外などを簡単に見ることができます。設定されているすべての時間リージョンは UTC 時間を参照します。

### ヒートマップパネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

ヒートマップパネルの視覚化では、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、「」を参照してください[ヒストグラムとヒートマップの概要](#)。

## データから計算する

この設定により、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が決まります。

## 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、 の時間範囲 1h は、X 軸でセルを 1 時間幅にします。

## 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。

## Y バケットスケール

次のいずれかの Y 軸値スケールを選択します。

- 線形 – 線形スケール。
- log (ベース 2) – ベース 2 の対数スケール。
- log (ベース 10) – ベース 10 の対数スケール。

## Y 軸

Y 軸の表示方法を定義します。

## 配置

- 左 – 左
- 右 – 右
- 非表示 – 非表示

## 単位

## 単位設定

## 10 進数

この設定により、10 進数の設定が決まります。

## 最小/最大値

この設定は、軸範囲を設定します。

## リバーズ

選択すると、軸は逆の順序で表示されます。

## カラー

色スペクトルは、値カウント (各バケット内) と各バケットに割り当てられた色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小数を表し、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。

色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よって変化します。

- Mode (モード)
  - スキーマ — セルの色で表されるバケット値。
    - スキーマ - モードがスキーマ の場合は、カラースキームを選択します。
  - 不透明度 - セル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。
    - 色 - セルのベース色。
  - スケール — バケット値を不透明度にマッピングするためのスケール。
    - 線形 - 線形スケール。バケット値は不透明度に直線的にマッピングされます。
    - sqrt - パワースケール。セルの不透明度は  $\text{value}^k$  として計算され、 $k$  は設定された指数値です。指数が より小さい場合1、対数スケールが得られます。指数が より大きい場合1、指数スケールが得られます。の場合1、スケールは線形と同じになります。
  - Exponent - 指数の値、 より大きい0。

## 値からの開始/終了色

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。Min と Max を使用すると、これらの値を上書きできます。バケット値を Z 軸として、最小と最大をそれぞれ Z-Min と Z-Max として考えます。

- 開始 — セルの色計算に を使用する最小値。バケット値が Min より小さい場合は、「最小」色にマッピングされます。シリーズ最小値はデフォルト値です。

- 終了 — セルの色計算に を使用する最大値。バケット値が Max より大きい場合は、「最大」色にマッピングされます。シリーズ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

- ツールヒントを表示 — ヒートマップツールヒントを表示します。
- ヒストグラムを表示 — ツールチップに Y 軸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の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。

## 例

例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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ヒストグラムの視覚化は、値の分布を計算し、棒グラフとして表示します。Y 軸と各バーの高さは各角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は、時系列と、1 つ以上の数値フィールドを含むテーブル結果をサポートしません。



## サポートされる形式

ヒストグラムの視覚化は、時系列と、1 つ以上の数値フィールドを含むテーブル結果をサポートします。

## 表示オプション

視覚化を絞り込むには、以下のオプションを使用します。

### バケットサイズ

バケットのサイズ。自動バケットサイズ設定 (全範囲の 10% まで) の場合は、これを空のままにします。

### バケットオフセット

最初のバケットがゼロから開始すべきでない場合。ゼロ以外のオフセットは、集計ウィンドウをシフトします。例えば、デフォルトの 0 オフセットを持つ 0~5、5~10、10~15 の 5 サイズのバケットは、2~7、7~12、12~17 になり、オフセットは 2 になります。この場合、オフセットは 0、5、または 10 で、実質的に何もしません。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されます。この設定に影響を与えるには、オフセット量が 0 より大きく、バケットサイズより小さい必要があります。この範囲外の値は、この範囲内の値と同じ効果を持ちます。

### シリーズを組み合わせる

これにより、すべてのシリーズとフィールドが結合ヒストグラムにマージされます。

線幅は、棒の線幅を制御します。

塗りつぶし不透明度は、バー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は、グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – グラデーションフィルなし、これがデフォルト設定です。
- 不透明度 — 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。
- Hue – グラデーション色は、線の色相に基づいて生成されます。

ツールヒントモード グラフにカーソルを合わせると、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、ホバーしているシリーズのみが表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – ツールヒントは表示しません。

#### Note

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

## 凡例オプション

凡例オプションを有効にすると、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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シングルカラーまたは Classic パレットに設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。

凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- リスト — 凡例をリストとして表示します。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 – 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。複数の を持つことができます。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。複数の を選択できます。

## ログパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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からのログ行が表示されます。通常、グラフパネルの横にある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、クエリタブに入力されたクエリの結果が表示されます。複数のクエリの結果はマージされ、時間でソートされます。データソースが返す行数が表示可能な行数よりも多い場合は、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設定を使用できます。設定されていない場合、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式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、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 がログレベルを特定できない場合、不明なログレベルで視覚化されます。詳細については、「[ログの視覚化](#)」を参照してください。

## ログの詳細

各ログ行には、より堅牢な操作のために、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、視覚化を絞り込みます。

- Time – 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に関連付けられた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。これは、非共通ラベルのみを表示します。
- 共通ラベル – 共通ラベルの表示/非表示
- ラップライン – ラインラップを切り替えます。
- Prettify JSON – これを に設定すると true、すべての JSON ログがプリティーに出力されます。この設定は、JSON 以外の形式のログには影響しません。
- ログの詳細を有効にする – オプションを切り替えて、各ログ行のログの詳細ビューを表示します。デフォルトの設定は、true です。
- 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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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 ブログの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、他のクエリを受け付けません。

**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ロードされます。その結果、適切な [CORS ヘッダー](#) で設定された RSS フィードのみがロードされます。表示しようとしている RSS フィードのロードに失敗する場合は、RSS フィードを再ホストするか、独自のプロキシを作成すること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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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または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、ノードとエッジで構成されます。

- ノードは円として表示されます。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。
- 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。接続には、リクエスト、オペレーション、または2つのノード間のその他の関係があります。

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、データソースごとに異なるタイプの値を表示したり、一部の値を表示したりすることはできません。

## ノード

通常、ノードにはノード内に 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。ノードは、ノードの周囲に別の値のセットを色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、合計が最大 1 である異なる値を表します。例えば、エラーの割合を円の赤い部分で表すことができます。

追加の詳細をコンテキストメニューで表示でき、ノードを選択すると表示されます。コンテキストメニューには、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もあります。

### 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

エッジにカーソルを合わせると、エッジに統計が表示されることもあります。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くことができます。

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。

## ノードグラフのナビゲーション

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、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してズームできます。

## 隠しノード

一定の時間に表示されるノードの数は、妥当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およその数を示すクリック可能なマーカーの背後に非表示になります。マーカーを選択すると、そのノードの周囲に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、グラフ内の最も関心のある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の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。統計名の横のマーカは、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、グラフィックで表示 オプションを選択して、選択したノードに焦点を絞ってグラフィックに戻し、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、データを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。オプションで、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、ノードを説明する 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 `frame.meta.preferredVisualisationType = 'nodeGraph'` に を設定するか、それぞれ `nodes` と `edges` に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	string	エッジの一意的識別子。
ソース	string	ソースノードの ID。
target	string	ターゲットの ID。

オプションフィールド：

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るとオーバーレイに表示される

フィールド名	タイプ	説明
		最初の統計。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。
セカンダリ統計	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	エッジをクリックするとdetail__、というプレフィックスが付いた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。より人間が読めるラベル config.displayName には 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	string	ノードの一意的識別子。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。

オプションフィールド：



フィールド名	タイプ	説明
title	string	ノードのすぐ下の に表示されるノードの名前。
subtitle	string	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	ノード自体内に表示される最初の統計。値はそのままの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。
セカンダリ統計	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せん。
arc__*	数値	プレフィックス が付いたフィールドarc__はすべて、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値はすべて、合計で 1 になります。を使用して色を指定できます config.color.fixed Color 。
detail__*	文字列/数値	ノードをクリックするとdetail__、コンテキストメニューのヘッダーに というプレフィックスが付いたフィールドが表示されます。より人間が読めるラベ

フィールド名	タイプ	説明
		ルconfig.displayName には を使用します。

## 円グラフパネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

円グラフには、1つ以上のクエリからの縮小されたシリーズまたは値が、相互に関連して円のスライス形式で表示されます。スライスの円弧の長さ、面積、中心角度はすべて、すべての値の合計に関連するため、スライス値に比例します。このタイプのグラフは、小さな値のセットを気楽な形式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、視覚化の値を調整します。

#### 表示

表示する情報の量を選択します。

- Calculate – 各値をシリーズごとに 1 つの値に減らします。
- すべての値 – 1 つのシリーズのすべての値を表示します。

#### 計算

計算を選択すると、計算が選択されたときに各シリーズが減ります。使用可能な計算については、「[計算タイプ](#)」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。各フィールド名はリストで使用できます。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。
- Time – 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を使用して、視覚化の外観を絞り込みます。

## 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 – 標準の円グラフ
- 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。複数の を選択できます。

- 名前 – シリーズ名またはフィールド名。
- Percent – 全体のパーセンテージ。
- 値 - 未加工の数値。

デフォルトでは、ラベルはグラフの本文に白で表示されます。より暗いグラフの色を選択して、より見やすくすることができます。長い名前や数字はクリップされる場合があります。

## ツールヒントモード

視覚化の上にカーソルを置くと、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、視覚化にカーソルを合わせているシリーズが 1 つだけ表示されます。

- **すべて** – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- **非表示** – 視覚化を操作するときツールヒントを表示しないでください。

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

## 凡例モード

これらの設定を使用して、視覚化に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。

- **List** – 凡例をリストとして表示します。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。
- **Hidden** – 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** – 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。複数の を持つことができます。

凡例に表示する値を選択します。複数の を選択できます。

- **Percent** – 全体のパーセンテージ。
- **値** – 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。

## プロットパネル

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Plotly パネルは、オープンソースの javascript グラフライブラリである [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 Config 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。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 チャートに挿入する前に、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 つの引数が含まれています。

- data – データソースによって返されるデータ。
- variables – 現在のダッシュボードに [Grafana 変数](#) (ユーザー変数と、以下のいくつかのグローバル変数: \_\_from、\_\_to、\_\_interval、および \_\_interval\_ms) を含むオブジェクト。

スクリプトは、data、layout config の 1 つ以上のプロパティを持つオブジェクトを返す必要があります frames。次に例を示します。

```
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
  }
}
```

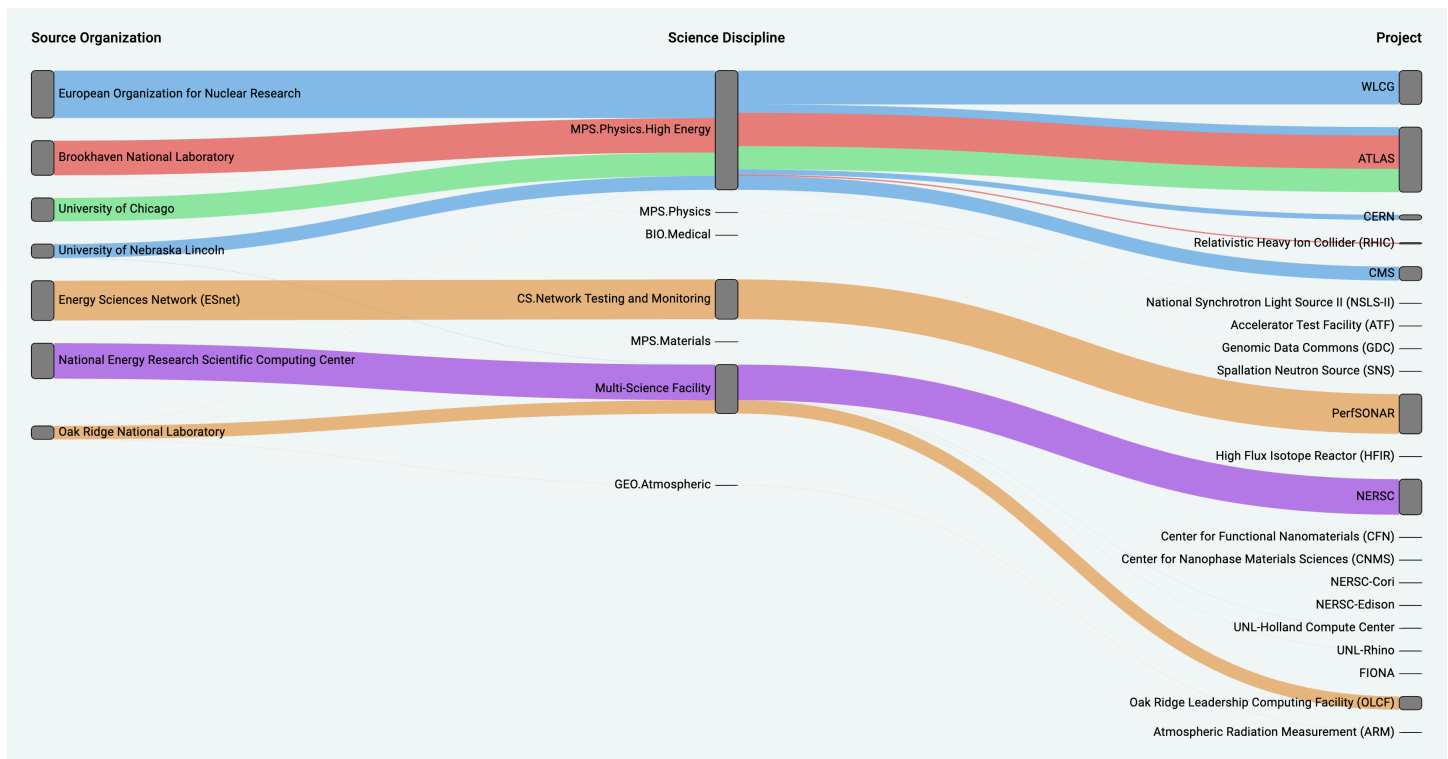
データ、レイアウト、および Config フィールドで指定されたスクリプトと JSON によって返されたオブジェクトはマージされます (ディープマージ)。

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 Config 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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

サンキーパネルにはサンキー図が表示されます。サンキー図は、フローデータを視覚化するのに適しており、フローの幅は選択したメトリクスに比例しています。次の画像は、送信元と送信先の 2 つのグループを含むサンキー図を示しています。



## 仕組み

サンキーパネルには、フローの送信元と送信先という 2 列以上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。リンクの太さは、クエリ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とシングルの2つのオプションがあります。デフォルトでは、複数色になっています。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を使用するか、入力ボックスに数値を入力することで調整できます。この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。この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少し調整できます。この数値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軌道反復の数です。

## 散布図

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

散布図パネルには、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 散布図が表示されます。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である必要はありません。散布図には、2 つ以上のデータ列を含むテーブル形式のデータセットが必要です。

そのうちの1つを X 軸に割り当てることができます。1つ以上の を一連の Y 軸値に割り当てることができ、結果のデータを一連のドットとしてプロットできます。各シリーズでは、オプションで、いくつかの統計のベストフィットの1つ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、、、AgeBoysおよびの3つの数値列HEIGHTで呼び出されGirls、年齢別の少年と少年の平均の高さを示すデータがあると想定します。

年齢	の高さ	女性の高さ
5	109.7	109.5
6	115.6	115.4
7	121.1	120.8
8	126.3	126
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記述します。この場合、 などのクエリを使用しますSELECT \* FROM HEIGHT。
3. 散布図を選択します。

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

散布図には、次の 4 つのカスタム設定オプションがあります。

- X 軸 – X 軸として使用するフィールド、および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。各フィールドの回帰線を表示することもできます。リグレッションライン設定の詳細については、次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、凡例内のテキストのサイズを選択したりできます。
- 表示 – グリッドの色や境界線スタイルなど、グラフの他のオプションを設定できます。

### 回帰行の設定

各 Y 軸データセットには、個々のドットに加えて線を表示できます。行タイプには 5 つのオプションがあります。

- なし – 回帰線は表示しません。
- Simple – 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、最適法を使用して直線を表示します。
- エクスポネンシャル – エクスポネンシャルのベストフィット回帰線を表示します。
- 電力 – 電力の最適な回帰線を表示します。

## 統計パネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

統計パネルの視覚化では、1つの大きな統計値とオプションのグラフスパークラインが表示されます。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。

- 単一のシリーズまたはフィールドの値のみ。
- 複数のシリーズ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、テキストを表示するかどうかを制御します。

### 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

以下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

すべての行に基づいて計算された値を表示します。

- 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リデューサー関数を選択します。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設定します。

### [Orientation] (向き)

スタッキング方向を選択します。

- 自動 – Grafana は、最適な向き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縮します。
- Vertical – バーは上から下に垂直に伸縮します。

## 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。値が重要でない場合は、名前と色のみがである場合、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。代わりに、ホバーツールヒント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。値はホバーツールヒントに表示されます。
- なし — 何も表示しない (空)。名前と値は、ホバー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。
- 背景 – 背景の色も表示されます。

## 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 – 値の下にエリア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。

## テキストの配置

アラインメントモードを選択します。

- 自動 – 単一の値のみが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。複数のシリーズまたは行が表示されている場合、値は左揃えになります。
- 中心 – 統計値は中央に配置されます。

## 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 – ゲージ値のサイズに数値を入力します。

## 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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シリーズが一意的な水平バンドとしてレンダリングされます。ステートリージョンは、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状態に適していますが、時系列でも使用できます。時系列とともに使用すると、しきい値を使用して数値を離散状態領域に変換します。

## 状態タイムラインオプション

視覚化を絞り込むには、以下のオプションを使用します。

### 等しい連続値をマージする

Grafana が同じ値を互いに隣り合っている場合にマージするかどうかを制御します。

### 値を表示する

値が状態リージョン内でレンダリングされるかどうかを制御します。十分なスペースがある場合、Auto は値をレンダリングします。

### 値の整列

状態リージョン内の値の配置を制御します。

### 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。

### 線幅

状態リージョンの線幅を制御します。

### 不透明度を埋める

状態リージョンの不透明度を制御します。

### 値マッピング

ブール値または文字列値に色を割り当てるには、[を使用します](#) [値マッピングを設定する](#)。

### 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リージョンに変換します。

## 凡例オプション

凡例オプションを有効にすると、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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「シングルカラー」または「クラシックパレット」に設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。

凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- List – 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

## ステータス履歴パネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ステータス履歴の視覚化では、定期的な状態が時間の経過とともに表示されます。各フィールドまたはシリーズが水平行としてレンダリングされます。ボックスはレンダリングされ、各値を中心に配置されます。

ステータス履歴の視覚化は、文字列、ブール値、および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を色付けしたり、テキスト値を数値範囲に割り当てることができます。

## 表示オプション

視覚化を絞り込むには、以下のオプションを使用します。

### 値を表示する

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。十分なスペースがある場合、Autoは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。

線幅は、状態リージョンの線幅を制御します。

塗りつぶし不透明度は、状態リージョンの塗りつぶし不透明度を制御します。

### 値マッピング

ブール値または文字列値に色を割り当てるには、[を使用します](#) [値マッピングを設定する](#)。

### 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。グラデーションカラースキームを使用して色値を作成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、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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「シングルカラー」または「クラシックパレット」に設定することが重要です。凡例のしきい値括弧を表示するには、カラースキームを「From thresholds」に設定します。

凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- List – 凡例をリストとして表示します。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 – 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。複数のを持つことができます。

## テーブルパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 データに対して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、値の書式、および色付けオプションも表示されます。

## ソート列

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から昇順に変更します。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。

### 列の幅

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、設定を上書きし、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フィールドにと入力すると、フィールドの外部をクリックすると、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は、その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フィールドにと入力すると、フィールドの外部をクリックすると、すべての列の幅が75ピクセル以下にスケールされます。

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は、デフォルトのピクセル値を150に減ら50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るようにします。

### 列の配置

Grafana がセルの内容をどのように整列させるかを選択します。

- Auto (デフォルト)
- 左
- 中央
- [Right] (右)

### セルタイプ

デフォルトでは、Grafana は自動的に表示設定を選択します。次のオプションのいずれかを選択して設定を上書きし、すべてのフィールドにデフォルトを設定できます。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対して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。

### カラーテキスト

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。

## 背景色 (グラデーションまたは単色)

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。

### Gauge

セルはグラフィカルゲージとして表示でき、いくつかの異なるプレゼンテーションタイプがあります。

#### Basic (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。

#### グラデーション

しきい値レベルは勾配を定義します。

### LCD

ゲージは、点灯または点灯していない小さなセルに分割されます。

### 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、JSON オブジェクトを参照できる JSON ビューがホバーに表示されます。

### セル値検査

テーブルセルからの値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

セル値検査を使用できるのは、セル表示モードが自動、カラーテキスト、カラー背景、または JSON ビューに設定されている場合のみです。

### 列フィルター

列データの表示方法は一時的に変更できます。例えば、値を最高から最低に順序付けしたり、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。

## ページ分割

このオプションを使用して、ページ分割を有効または無効にします。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。

## フィルターテーブル列

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で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。

各列タイトルの横にフィルターアイコンが表示されます。

## 列値をフィルタリングする

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 はその列のフィルターオプションを表示します。

表示する値の横にあるチェックボックスをオンにします。上部の検索フィールドにテキストを入力すると、それらの値がディスプレイに表示され、スクロールして検索するのではなく選択できるようになります。

## 列フィルターをクリアする

フィルターが適用された列には、タイトルの横に青いファネルが表示されます。

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、フィルターを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、次に計算するフィールドを選択できます。

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。

## 行をカウントする

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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、コンテキスト情報や説明を追加したり、複雑な HTML を埋め込んだりするために使用できます。

## Mode (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。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。
- コード – 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の強調表示を適用します。

## 変数

コンテンツ内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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

時系列パネルでは、時系列を線、ドットのパス、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を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

- [ツールヒントのオプション](#)
- [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](#)
- [カラーオプション](#)

## ツールヒントのオプション

-  このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

グラフにカーソルを合わせると、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 - ホバーツールヒントには、ホバーしているシリーズのみが表示されます。

- **すべて** – ホバーツールヒントには、グラフ内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- **非表示** – ツールヒントは表示しません。

## 凡例オプション

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

凡例モード — 凡例の表示方法を選択します。

- **List** – 凡例をリストとして表示します。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** – 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

## 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。スタイルには 3 つの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。

- 線 - 時系列をグラフに線として表示します。
- 棒 - 時系列をグラフ上の一連の棒として、データポイントごとに 1 つずつ表示します。
- ポイント - 時系列をグラフにドットとして表示し、データポイントごとに 1 つずつ表示します。

## バーの配置

棒グラフの場合、は、グラフ上の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。棒の幅はあるため、ポイントの前、後、または中心に配置することができます。このオプションの選択は、の前、中央、またはの後です。

## 線幅

折れ線グラフの線の太さ、または棒グラフの各棒グラフのアウトラインの太さを設定します。

## 不透明度を埋める

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されます。

## グラデーションモード

グラデーションモードは、シリーズの色に基づいてグラデーションフィルを指定します。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。勾配モードのオプションは次のとおりです。

- なし — 勾配フィルなし。
- 不透明度 - Y 軸の値が増加するとフィルの不透明度が増加する不透明度勾配。
- Hue - シリーズの色相に基づくグラデーション。

- スキーマ — カラースキームで定義される色グラデーション。この設定は、塗りつぶしと行で使用できます。詳細については、「[カラーオプション](#)」を参照してください。

勾配の外観は、不透明度を塗りつぶす設定によっても変更されます。

## ポイントの表示

折れ線グラフまたは棒グラフにポイントを追加するように視覚化を設定できます。Always、Never、または Auto を選択できます。Auto 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は、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、1~40 ピクセルの半径に設定します。

## 行補間

Grafana がシリーズ行を補間する方法を選択します。選択肢は、線形、スムーズ、より前のステップ、より後のステップです。

## 線のスタイル

行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。

線スタイルの外観は、線幅 と塗りつぶし不透明度 の設定の影響を受けます。

ラインスタイルの選択肢は、Solid、Dash、および Dots です。

## NULL 値を接続する

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、オプションでデータのギャップを接続すべきではないしきい値を設定することもできます。データポイントをギャップで接続しない、データポイントをギャップで常に接続する、またはデータ内のギャップを接続すべきでなくなるしきい値を設定することができます。

## スタックシリーズ

積み上げにより、Grafana はシリーズを互いに重ねて表示できます。ビジュアライゼーションでスタッキングを使用する場合は、誤解を招くグラフを簡単に作成できるため、注意してください。ス



スタッキングが最適なアプローチではない理由の詳細については、[「スタッキングの問題」](#)を参照してください。

スタックのオプションは次のとおりです。

- オフ — シリーズスタッキングをオフにします。
- 正常 — スタックシリーズを互いに重ね合わせます。
- 100% — パーセンテージでスタックします。すべてのシリーズの合計は最大 100% です。

### グループ内のスタックシリーズ

スタックの動作をグループ内のスタックシリーズにオーバーライドできます。オーバーライドの作成の詳細については、「」を参照してください[フィールドオーバーライドを設定する](#)。オーバーライドを作成するときは、シリーズに含めるスタッキンググループの名前を指定します。

以下を  に入力します。

以下の Fill to オプションは、2つのシリーズ間の領域を埋めます。このオプションは、シリーズまたはフィールド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、シリーズ行から 0 までではなく、2つのシリーズ間の領域を埋めることができます。例えば、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 Fill 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つのシリーズ線の間の領域のみが埋められます。

### 軸オプション

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

軸カテゴリのオプションは、X 軸と Y 軸のレンダリング方法を変更します。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しても有効になりません。または キーを押します Enter。

### 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 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に割り当て、右軸を次の単位に割り当てます。
- 左 – 左側にすべての Y 軸を表示します。
- 右 – 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シリーズに軸を割り当てるには、[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、オーバーライドを使用して異なるラベルを割り当てることができます。

## [Width] (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、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。この設定により、軸が視覚的に互い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には、ソフト最小値またはソフトマックス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、大きな変更が存在し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または最大は、特定のポイントを超えてスパイクをクリップすることで、断続的なスパイクが有用な詳細を平坦化することを防ぐことができます。

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 軸のスケールリング方法を設定します。選択肢は線形 または対数 です。対数を選択した場合は、さらに 2 から 10 までの対数スケールを選択できます。

## [Transform] (変換)

シリーズ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ヒント、コンテキストメニュー、凡例の値には影響しません)。次の 2 つの変換オプションがあります。

- 負の Y 変換 — 結果を Y 軸の負の値に反転させます。
- 定数 - 最初の値を定数行として表示します。

## カラーオプション

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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シリーズ色を割り当てます。凡例シリーズの色アイコンをクリックして、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、特定のシリーズ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、シリーズカラーデフォルトを上書きするために使用できる追加オプションです。

## クラシックパレット

最も一般的な設定は、グラフに Classic パレットを使用することです。このスキームは、順序に基づいて各フィールドまたはシリーズ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。凡例の各シリーズの横にある色付きの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。

## 値別のカラースキーム

From thresholds (値) や Green-Yellow-Red (値) などの値別カラースキームを選択すると、オプション別のカラーシリーズが表示されます。このオプションは、シリーズに色を割り当てるために使用する値 (最新、最小、最大) を制御します。

## スキーマ勾配モード

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、スキーマ という名前のモードがあります。スキーマ を有効にすると、線またはバー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。

## しきい値から

カラースキームが From しきい値 (値別) に設定され、グラデーションモードが スキーマ に設定されている場合、定義されたしきい値を超えると線または棒の色が変わります。スキーマ 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マ 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、シリーズ内の値が設定されたしきい値の間を移動するときに、選択した色が選択した色間にグラデーションを形成することを意味します。

## トレースパネル (ベータ)

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

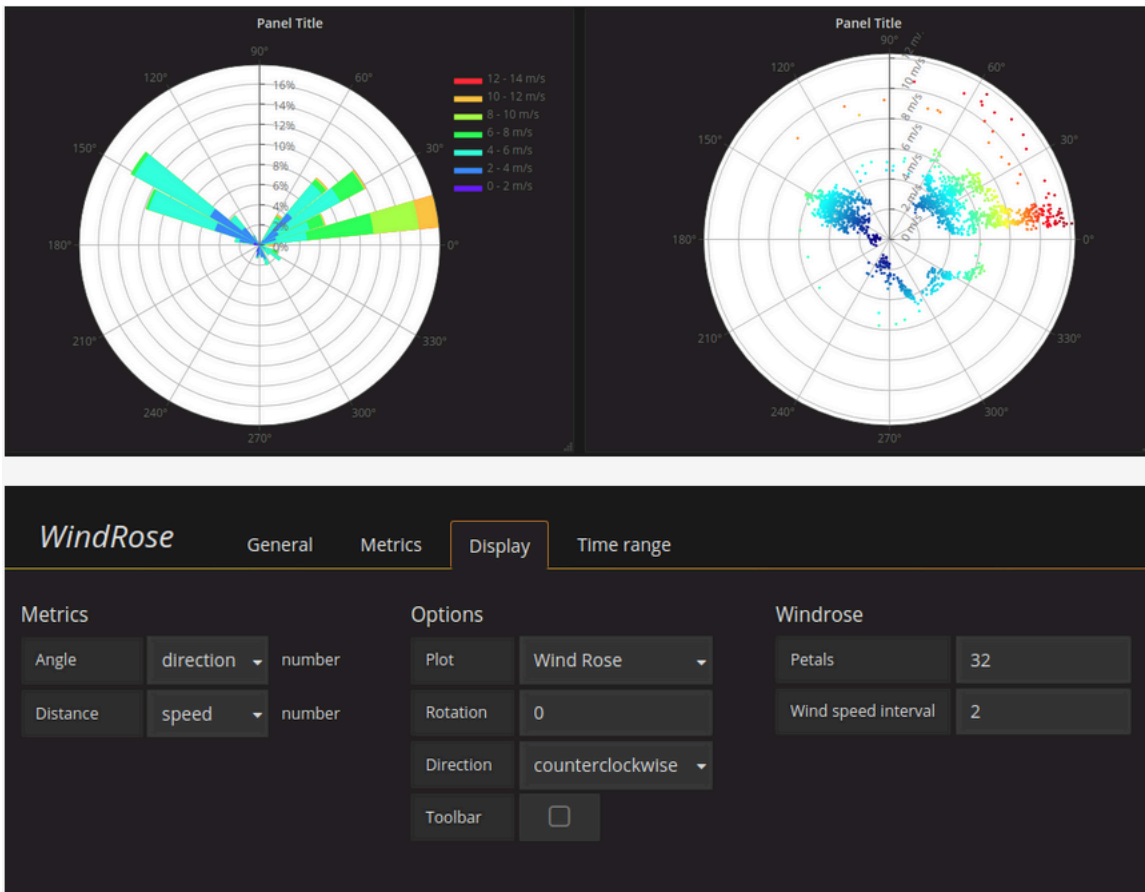
トレースは、インフラストラクチャ内のサービスを通過するリクエストを追跡およびログ記録できる視覚化です。

トレースの詳細については、「」を参照してください[Explore でのトレース](#)。

## WindRose

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

WindRose パネルは、生の時系列データを受け取り、データを変換して WindRose グラフにマッピングします。



## オプション

WindRose パネルでは、次のオプションがサポートされています。

- 軸周波数

- 軸スタイル (度またはコンパス)
- スケール (線形、正方形、ログ)

## Grafana バージョン 9 でを試す

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana のダッシュボード UI には、視覚化用のダッシュボードを構築する機能があります。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詳しく調べます。クエリが機能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。

### **i** Note

データを探索するだけで、ダッシュボードを作成したくない場合は、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として表示します。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

### **i** 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。  
空の探索タブが開きます。

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探索オプションを選択します。これにより、パネルからクエリを含む探索タブが開き、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](#)にはカスタム Explore 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記述します。クエリフィールドの横には、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 つのボタンがあります。通常のクエリエディタと同様に、複数のクエリを追加および削除できます。

クエリの詳細については、「」を参照してください[データのクエリと変換](#)。

## 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簡単に比較 side-by-side 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。

### トップで分割ビューを開く

1. 探索ビューで、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 side-by-side つの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、タイムピッカーの 1 つにアタッチされたタイム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リンクできます (変更すると、もう 1 つも変更されます)。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間と終了時間が同期されます。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。

## 短縮リンクの共有

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 URLs を使用する代わりに、/goto/:uid 形式のより小さくてシンプルな URLs 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 ツールバーの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 (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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

クエリのデバッグに役立つように、Explore では、クエリインスペクターを介してクエリのリクエストとレスポンス、クエリ統計を調査できます。この機能は、パネルインスペクタータスクの[クエリパフォーマンスを検査し、クエリリクエストとレスポンスデータを検査します](#)。

## 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します。
- クエリを含む短縮リンクをクリップボードにコピーします。
- Star (お気に入り) クエリ。

### お気に入りのクエリを管理する

クエリ履歴タブにスターが表示されているすべてのクエリが、スターが表示されている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、ゼロから入力せずにこれらのクエリを再利用できます。

### クエリ履歴のソート

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でソートできます。

### クエリ履歴をソートするには

1. クエリをソートする 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。
  - 最新の 1 番目
  - 最も古いもの

### クエリ履歴のフィルタリング

クエリ履歴と星付きタブでクエリ履歴を特定のデータソースにフィルタリングできます。

### データソースへの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する (複数可) 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、日付でクエリをフィルタリングします。
- トップハンドルをドラッグして開始日を調整します。

- トップハンドルをドラッグして終了日を調整します。

## クエリ履歴での検索

クエリやコメントをまたいで履歴を検索できます。クエリ履歴タブと星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. クエリの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。

## クエリ履歴の設定

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、以下の表で説明します。

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

クエリ履歴設定はグローバルで、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

Prometheus のカスタムクエリエクスペリエンスについて説明します。クエリが実行されると、実際には 2 つのクエリ、グラフの通常の Prometheus クエリとテーブルのインスタント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概要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、メトリクスを選択してメトリクスエクスプローラーを開きます。これは、メトリクスがプレフィックスでグループ化された階層メニューを示しています。例えば、すべての Alertmanager 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調べる場合の出発点として最適です。

## クエリフィールド

Query フィールドは、標準の Prometheus クエリエディタと同様に、メトリクス名と関数のオートコンプリートをサポートします。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。

オートコンプリートメニューは、Ctrl+Space を押すとトリガーできます。オートコンプリートメニューには、最近実行されたクエリのリストを含む新しい履歴セクションが含まれています。

提案はクエリフィールドの下に表示されることがあります。それらを選択して、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) の場合、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提案されます。
- 記録ルールの場合、ルールを拡張できます。

## テーブルフィルター

テーブルパネルのラベル列のフィルターボタンを選択して、クエリ式にフィルターを追加します。複数のクエリにフィルターを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。

## Explore のログ

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Explore では、メトリクスに加えて、次のデータソースのログを調査できます。

- [OpenSearch](#)
- [InfluxDB](#)
- [ロキ](#)

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。また、Explore では、メトリクスとログを表示して関連させることができます side-by-side。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. 再度ドリルダウンし、メトリクスと時間間隔 (および将来的には分散トレース) に関連するログを検索します。

## ログの視覚化

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、このフルレンジログボリュームのヒストグラムはメトリクスクエリによってレンダリングされます。メトリクスクエリは、クエリされた時間範囲によっては高価にな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、これらのクエリのカスタムタイムアウト (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、値を持つ 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 Source=logvolhist。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。時系列の終わりは、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決定し、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

ポートされている式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 がログレベルを特定できない場合、不明なログレベルで視覚化されます。

### 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、ログレベルの判別で使用されるレベルラベルにレベル情報を抽出します。これにより、ヒストグラムはさまざまなログレベルを別々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、ログレベルの略語と式のマッピング：

サポートされている式	ログレベル	色
Emerg	非常事態	紫色
致命的	非常事態	紫色
アラート	非常事態	紫色
クリティカル	非常事態	紫色
非常事態	非常事態	紫色
エラー	エラー	赤色
エラー	エラー	赤色
エラー	エラー	赤色
警告	warning	黄色
warning	warning	黄色
info	info	green
情報	info	green
通知	info	green

サポートされている式	ログレベル	色
デバッグ	デバッグ	青色
デバッグ	デバッグ	青色
トレース	トレース	水色
*	不明	グレー

## 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。行数の上限に達し、さらにログを表示したい場合は、これを使用してさらにログを取得できます。各リクエストは個別のページとしてナビゲーションに表示されます。すべてのページには、受信ログ行の から へ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。ログナビゲーションから実行された最後の 5 つのリクエストがキャッシュされるため、それらのページをクリックすると同じクエリが再実行されません。

## 視覚化オプション

ログの表示方法をカスタマイズし、表示する列を選択できます。

### Time (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に関連付けられた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。

### ラップライン

ディスプレイでラインラッピングを使用する場合は、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。

### JSON の指定

これを に設定すると true、すべての JSON ログがプリティイーに出力されます。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 – 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 – 期間、IP アドレスなどの数値を削除した後、行で一致します。
- 署名 – 最も積極的な重複排除。これにより、すべての文字と数字が除去され、残りの空白と句読点が一貫します。

### 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 (最新の順) 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、より堅牢な操作のために、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 (正のフィルター) し、フィルタリングアウト (負のフィルター) する機能が追加されました。各フィールドまたはラベルには、表示され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 (、) やタブ (\r) など \n、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 \t。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、検出され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. エスケープ改行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。その場合、オプションはエスケープ改行からエスケープの削除 に変わります。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、置換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の切り替え

ログ行を展開し、目のアイコンをクリックしてフィールドを表示または非表示にします。

### 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、ログの内容のインデックスを作成するのではなく、各ログストリームの一連のラベルを作成するため、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されます。ラベルを使用してログストリームをグループ化し、Prometheus ラベルと一致させることができます。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への切り替え

Prometheus クエリからログクエリに切り替えると (最初に分割してメトリクスとログを並べて表示することもできます)、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をクエリします。例えば、次の Prometheus クエリがある場合です。

```
grafana_alerting_active_alerts{job="grafana"}
```

Logs データソースに切り替えた後、次のように変更されます。

```
{job="grafana"}
```

これにより、検索可能な選択した時間範囲内のログのチャンクが返されます。

### ログのサンプル

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、メトリクスクエリでは、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。

### ライブテール



ライブテール機能を使用して、サポートされているデータソースのリアルタイムログを表示します。

探索ツールバーのライブボタンを選択して、ライブテールビューに切り替えます。

ライブテールビューでは、新しいログは画面の下部から取得され、背景にコントラストが消えるため、新しいものを追跡できます。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ルを一時停止し、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ルを再開するか、停止ボタンを選択してライブテールを終了し、標準の探索ビューに戻ります。

## Explore でのトレース

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Explore では、トレースデータソースからの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](#)
- [テンポ](#)
- [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、トレースビュー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。

## 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、より小さな時間範囲にズームインします。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ると、ズーム時に選択のリセットボタンが表示され、ズームがリセットされます。

## 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、次のコンポーネントで構成されます。

- 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。
- スパン継続時間バー: トレース内のオペレーション継続時間の視覚的表現。

## スパンの詳細

スパン行の任意の場所をクリックすると、次のような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タグ。
- プロセスメタデータ: このスパンを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を含む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をクエリします。

## サービスグラフビュー

サービスグラフビューは、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。要件を設定すると、この事前設定済みビューがすぐに利用可能になります。

詳細については、「[Tempo data source page](#)」を参照してください。Tempo ドキュメントの [サービスグラフビューページ](#) も参照できます。

## Explore の Inspector

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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。raw データを検査したり、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たり、ログ結果を TXT 形式でエクスポートしたり、クエリリクエストを表示したりできます。

## 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と返される量を示します。
- クエリタブ – Grafana がデータソースにクエリを実行したときに送信された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た raw 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。

### raw クエリ結果の検査

クエリによって返された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、データ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：表示する結果セットデータを選択します。
- 時間で結合されたシリーズ：すべてのクエリの raw データを一度に表示し、列ごとに 1 つの結果セットを表示します。列見出しをクリックすると、データをソートできます。

### raw クエリ結果を CSV としてダウンロードする

raw クエリ結果を表示したら、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 専用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、CSV のダウンロードオプションを選択する前に Excel 用ダウンロードの切り替えをオンにします。

### TXT としてログ結果をダウンロードする

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。Jaeger、Zipkin、OTLP 形式です。

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、送信するクエリの数、返される行数を示す統計が表示されます。この情報は、特に数値が予期せず高い場合や低い場合に、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 モデルを表示する

データフレーム JSON モデルだけでなく、データ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. Inspector 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、次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。
  - DataFrame structure – raw 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むと、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 raw リクエストとレスポンスを表示する

Explore と Inspector タブで作業しているときに、クエリで生成する未加工のリクエストとレスポンスの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 raw 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。クエリの特典部分をドリルダウンしたり、すべて展開または折りたたんだり、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することができます。

## Grafana バージョン 9 のアラート

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana アラートは、システムの問題が発生した直後にその問題を把握し、サービスの中断を最小限に抑えるのに役立つ、堅牢で実用的なアラートを提供します。

Amazon Managed Grafana には、アラート情報を 1 つの検索可能なビューに一元化する、更新されたアラートシステムである Grafana アラートへのアクセスが含まれています。これには、次の機能が含まれます。

- 集中ビューで Grafana アラートを作成および管理します。
- Cortex および Loki マネージドアラートを 1 つのインターフェイスで作成および管理します。
- Prometheus、Amazon Managed Service for Prometheus、およびその他の Alertmanager 互換データソースからのアラート情報を表示します。

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、を使用するかを選択できます[Classic ダッシュボードアラート](#)。このセクションでは、Grafana アラートについて説明します。

### **i** 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を切り替えることができます](#)。

## Grafana アラートの制限事項

- 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 Alertmanager データソースからルールを取得できます。サポートされている他のデータソースから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センターに送信します。ネイティブの Grafana アラートを使用している場合は、従来のダッシュボードアラートを維持し、新しい Grafana アラート機能を有効にしない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、Prometheus Alertmanager で作成されたアラートの通知を 1 つだけ送信する Grafana Alerting を有効にすることをお勧めします。

## トピック

- [概要](#)
- [アラートの探索](#)
- [アラートの設定](#)
- [クラシックダッシュボードアラートを Grafana アラートに移行する](#)
- [アラートルールを管理する](#)
- [アラート通知を管理する](#)

## 概要

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

以下に、Grafana アラートの仕組みの概要と、連携し、その柔軟で強力なアラートエンジンの中核を形成するいくつかの主要な概念を紹介します。

### 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よくあり、モニタリングおよび分析するシステムの詳細が表示されます。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

アラートインスタンスを起動するかどうかを決定する評価基準を設定します。アラートルールは、データソースからデータを取得するための 1 つ以上のクエリと式、アラートの必要性を構成する条件、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を記述する条件で構成されます。

Grafana マネージドアラートは多次元アラートをサポートしています。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無音に一致させます。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

アラートが発せられ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、ラベルマッチャーのセットを指定して、どのアラートに責任があるかを示します。通知ポリシーには、1 つ以上の通知機能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. 連絡先

アラートが発生したときに連絡先に通知する方法を定義します。アラートがチームに確実に届くように、さまざまな ChatOps ツールをサポートしています。

## 機能

### すべてのアラートに 1 ページ

単一の Grafana アラートページは、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を 1 か所に統合します。

### 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ます。これにより、1 つのアラートでシステム全体の可視性を実現できる能力と柔軟性が得られます。

### アラートのルーティング



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、および方法に関する一連のルールです。

## 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。特定の条件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、メインのアラートビューを整理することなく、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、新しい通知を生成または送信したくない時間間隔を指定できます。また、メンテナンス期間中など、繰り返し発生する期間のアラート通知をフリーズすることもできます。

## アラートの探索

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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と、問題を迅速に解決するチームの能力の向上に役立つ主要な概念と利用可能な機能の詳細をご覧ください。

まず、Grafana Alerting 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タイプです。これにより、サポートされている任意のデータソースからのデータに作用でき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、1つ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タイプのルールです。

## 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、ルーラー API がサポートされているかどうかを確認することで、データソースが Grafana による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

録画ルールは、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。記録ルールを使用すると、頻繁に必要な式や計算コスト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。

## 主要な概念と機能

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、定義のリストが含まれています。

主要な概念または機能	定義
アラート用のデータソース	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	ファイルプロビジョニングまたは Terraform 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。
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	アラートルールを起動するタイミングに関する一連の評価基準。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、および条件が満たされる期間で構成されます。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。

主要な概念または機能	定義
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。単次元のアラートルールには、アラートインスタンスが 1 つあります。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。10 VMs に対する CPU などの複数の結果に一致する 1 つのアラートルールは、複数の (この場合は 10 個) 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異なる場合があります。例えば、システム内のすべての VMs の CPU 使用率をモニタリングするアラートルールでは、VM が追加されるにつれて VMs。alert-instance クォータの詳細については、「」を参照してください <a href="#">クォータ到達エラー</a> 。
アラートグループ	Alertmanager 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。
連絡先	アラートルールが発生したときに連絡先に通知する方法を定義します。
メッセージテンプレート	再利用可能なカスタムテンプレートを作成し、問い合わせポイントで使用します。
通知ポリシー	アラートをグループ化してコンタクトポイントにルーティングする場所、タイミング、および方法に関するルールのセット。
ラベルとラベルマッチャー	ラベルはアラートルールを一意に識別します。アラートルールを通知ポリシーと無音にリンクし、どのポリシーがそれら进行处理するか、どのアラートルールを無音にするかを決定します。

主要な概念または機能	定義
無音	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。無音とミュートタイミングの違いは、無音が指定された期間のみ持続するのに対し、ミュートタイミングはスケジュールに従って繰り返されることです。ラベルマッチャーを使用してアラートインスタンスをサイレンスします。
ミュートタイミング	新しい通知を生成または送信しない時間間隔を指定します。また、メンテナンス期間中など、繰り返しアラート通知をフリーズすることもできます。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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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。

- [Alertmanager データソースに接続する](#)
- [Amazon CloudWatch データソースに接続する](#)
- [Amazon OpenSearch Service データソースに接続する](#)
- [AWS IoT SiteWise データソースに接続する](#)

- [AWS IoT TwinMaker データソースに接続する](#)
- [Amazon Managed Service for Prometheus とオープンソースの Prometheus データソースに接続する](#)
- [Amazon Timestream データソースに接続する](#)
- [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 データソースに接続する](#)
- [Google Cloud Monitoring データソースに接続する](#)
- [Graphite データソースに接続する](#)
- [InfluxDB データソースに接続する](#)
- [Loki データソースに接続する](#)
- [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 データソースに接続する](#)
- [Zipkin データソースに接続する](#)
- [Tempo データソースに接続する](#)
- [テスト用の TestData データソースを設定する](#)

## アラートルールについて

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートルールは、アラートインスタンスが起動するかどうかを決定する一連の評価基準です。ルールは、1つ以上のクエリと式、条件、評価の頻度、およびオプションで条件が満たされる期間で構成されます。

クエリと式が評価するデータセットを選択する間、条件はアラートを作成するためにアラートが満たす必要があるか超える必要がある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。期間を設定すると、条件を満たす必要がある期間を示します。アラートルールは、データがない場合のアラート動作を定義することもできます。

## 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](#)

## アラートルールタイプ

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana は、いくつか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、メリットとデメリットについて説明し、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タイプです。これにより、既存のデータソースからのデータに作用できるアラートを作成できます。

データソースのサポートに加えて、[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、および 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。データソースに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、ルーラー 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

録画ルールは、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、同じ式を繰り返しくエリするダッシュボードがある場合に便利です。

Prometheus の記録[ルール](#)の詳細をご覧ください。

## アラートインスタンス

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana マネージドアラートは、多次元アラートをサポートしています。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、評価中に観測された CPUs の量と同じ数のアラートインスタンスを作成し、1 つのルールで各 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートは、Mimir、Loki、または Prometheus のルールとグループ名の Grafana 管理のルールと名前空間のフォルダ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コントロールを実行し、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。

### Groups (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、ルールは同時に出現順に評価されません。

#### **i** Tip

異なる間隔で同時にルールを評価する場合は、異なるグループにルールを保存することを検討してください。

### 通知テンプレート

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。



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、「」を参照してください[Grafana バージョン 8 での作業](#)。

連絡先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 として評価されます (エスケープに影響する可能性があります)。

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です。

ほとんどのコンタクトポイント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できます。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、「」を参照してください[通知をカスタマイズする](#)。

### ネストされたテンプレート

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" . }}
```

カスタムテンプレートを埋め込むには、次の組み込みテンプレートオプションのいずれかを使用できます。

名前	メモ
default.title	高レベルのステータス情報を表示します。
default.message	発声アラートと解決済みアラートのフォーマットされた概要を提供します。

名前	メモ
teams.default.message	に似ておりdefault.message、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 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 <grafana-install-dir>/public/emails/ng\_alert\_notification.html。このファイルを編集して、すべてのアラート E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に設計されています。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、「」を参照してください [Grafana バージョン 10 での作業](#)。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、「」を参照してください [Grafana バージョン 8 での作業](#)。

このトピックでは、時系列データではなく数値データに対するアラートを Grafana がどのように処理するか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) に直接アラートを送信したり、サーバーサイド式に渡したりできます。これにより、データソース内の処理と効率が向上し、アラートルールが簡素化されます。時系列データではなく数値データでアラートを実行する場合、ラベル付けされた各時系列を 1 つの数値に減らす必要はありません。代わりに、ラベル付き番号は Grafana に返されます。

## 表形式データ

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 (ログ)、Azure Monitor (Azure リソースグラフ)、および Azure Data Explorer。

Grafana マネージドアラートまたは SSE を含むクエリは、次の場合、これらのデータソースでは数値と見なされます。

- データソースクエリでは、「Format AS」オプションが「Table」に設定されています。
- 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ます。

文字列列がある場合、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、各行はラベルを一意に識別する必要があります。

## 例

DiskSpace 「」という MySQL テーブルの場合：

時間	ホスト	ディスク	PercentFree
2021-June-7	web1	/etc	3
2021-June-7	Web2	/var	4
2021-June-7	Web3	/var	8
...	...	...	...

データフィルタリングは時間どおりにクエリできますが、時系列を Grafana に返す必要はありません。例えば、空き領域が 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があります。

```
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)
```

このクエリは、Grafana に次のテーブルレスポンスを返します。

ホスト	ディスク	PercentFree
web1	/etc	3
Web2	/var	4
Web3	/var	0

このクエリをアラートルールの条件として使用すると、ゼロ以外の `PercentFree` が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
{ホスト=web1,ディスク=/etc}	アラート
{ホスト=web2,ディスク=/var}	アラート
{ホスト=web3, ディスク=/var}	[普通]

## ラベルと注釈

- ⚠** このドキュメントトピックは、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に設計されています。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、「[アラートルール](#)」を参照してください。[Grafana バージョン 10 での作業](#)。
- 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、「[アラートルール](#)」を参照してください。[Grafana バージョン 8 での作業](#)。

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す。名前付き値のセットですが、用途は異なります。ラベルの例、または同等の注釈は `alertname="test"`。

ラベルと注釈の主な違いは、ラベルは他のすべてのアラートとの区別に使用され、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ます。

例えば、2つの高いCPUアラートを考えてみましょう。1つは `server1`、もう1つは `server2`。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場合があります。ただし、`server175%` 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化](#)のドキュメントを参照してください)。この種の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ています。ラベルの例は `server=server1`。各アラートには複数のラベルを付けることができ、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、アラートにラベルが設定されている `{alertname="High CPU usage",server="server1"}` のに対し、別のアラートにラベルが設定されている場合があります `{alertname="High CPU usage",server="server2"}`。ラベルは同じですが `alertname`、ラベルは異なるため、これらは2つの個別のアラート `server` 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、などの予約済みラベルの組み合わせです `alertname`。

## 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、アラートが発生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。カスタムラベルのテンプレート作成方法に関するドキュメントは、[こちら](#)で入手できます。

テンプレートでカスタムラベルを使用する場合、ラベルの値がアラートルールの連続評価間で変更されないようにすることが重要です。これにより、多数の個別のアラートが生成されます。ただし、テンプレートがアラートごとに異なるラベル値を生成しても問題ありません。例えば、クエリの値をカ

スタムラベルに入れなくてください。値が変わるたびに新しいアラートセットが作成されます。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 つ以上含まれてい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、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`、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、アラートが発生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、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。注釈のテンプレート作成方法に関するドキュメントは、[こちら](#)で入手できます。

## ラベルマッチングの仕組み

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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無音にリンクします。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無音に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。

- Label 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。
- Value 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。
- Operator フィールドは、ラベル値と照合する演算子です。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。
=~	値に正規表現が一致するラベルを選択します。
!~	値に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結合されます。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している必要があります。

### シナリオの例

アラートに次のラベルセットを定義した場合：

```
{ foo=bar, baz=qux, id=12 }
```

次に、次の操作を行います。

- 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foo=barと一致します。
- として定義されたラベルマッチャーfoo!=barがこのアラートルールと一致しません。
- 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id=~[0-9]+と一致します。
- 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baz!~[0-9]+と一致します。
- ととして定義された2つのラベルマッチャーはfoo=bar、このアラートルールid=~[0-9]+に一致します。

### Grafana アラートのラベル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください。[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

このトピックでは、ラベルがアラートの基本要素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 アラート内でアラートを一意に識別するものです。
- Alertmanager は、ラベルを使用して、通知ポリシー内の無音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は、ラベルにアクセスして、通知の結果であるアラートに固有の情報を含む通知を動的に生成できます。
- [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、ラベルが衝突した場合に優先されます (Grafana 予約ラベルの場合を除く。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 Alertmanager は、Unicode ラベルキーと値の両方をサポートしています。外部の 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、アンダースコアのみを含める必要があり、正規表現と一致する必要があります `[a-zA-Z_][a-zA-Z0-9_]*`。無効な文字は、次のルールに従って外部アラートマネージャーに送信される前に、Grafana アラートエンジンによって削除または置き換えられます。

- 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、プレフィックスとして `_` が付けられます。

### 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、重複には元のラベルの短いハッシュがサフィックスとして付加されます。



## Grafana 予約ラベル

### Note

プレフィックスに `grafana_` が付いたラベル `grafana_` は、Grafana が特別な使用のために予約しています。で始まるラベルを `grafana_` まる手動で設定されたラベルを追加すると、衝突時に上書きできません。

Grafana の予約済みラベルは、手動で設定したラベルと同じ方法で使用できます。使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
<code>grafana_folder</code>	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化

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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。以前に Prometheus を使用したことがある場合は、アラートのラベルと値を含む `$labels` および `$value` 変数に精通している必要があります。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、これらの各変数とそのテンプレート作成方法を心配しないでください。このページの残りの部分に従うと、`grafana_` が説明されます。

## 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である [テキスト/テンプレート](#) で記述されます。

## タグの開閉

テキスト/テンプレートでは、テンプレートが変数を出力するか、if }} ステートメントなどの制御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、変数の印刷で {{ と }} を使用し、制御構造で {{ }} 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%}}。

### [Print]

何かの値を出力するには、 {{ 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 変数を出力するには、次のように記述します。

```
{{ $labels }}
```

### ラベルを反復処理する

内の各ラベルを反復処理するには、 \$labels を使用できます range。ここで \$k、 は名前を指し、 \$v は現在のラベルの値 \$v を指します。例えば、クエリがラベルを返 instance=test した場合、 \$k は instance になり、 \$v は \$v になります test。

```
{{ 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。例えば、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、ダウン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出力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

印刷するラベルの名前に同じドット (完全な停止またはピリオド) が含まれている場合、テンプレート内の同じドットは機能しません。

```
Instance {{ $labels.instance.name }} has been down for more than 5 minutes
```

これは、テンプレートが `instance.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 `index` 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

`$value` 変数は Prometheus とは異なります。Prometheus `$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、Reduce 式と Math 式、および Classic 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まで返す可能性があるため、クエリの結果は含まれません。

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、RefID B を持つ式 `var='B'` 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}` はラベルを参照し、`value=6.789` を参照します。

RefID A がないことに気付いたかもしれません。これは、ほとんどのアラートルールでは RefID A がクエリを参照しているためです。また、クエリは多くの行や時系列を返すことができるため、`$value` に含まれていません。

## 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出力できます。`$values` とは異なり `$value`、`$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あり、RefID によってインデックス化されます。

アラートBの概要に RefID 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、は数字 6.789 {{ \$values.B }}を出力しますが、実際には、RefID B の浮動小数点値ではなく、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す。RefID B の浮動小数点値を使用するには、の Valueフィールドを使用する必要があります\$values.B。アラートの概要で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を返さない場合、またはデータソースのエラーやタイムアウトのために失敗した場合、そのクエリを使用するしきい値、削減、または数式もデータやエラーを返しません。この場合、これらの式には存在しません\$values。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。そうしないと、クエリがデータまたは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実行できます。

```
{{ 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、Reduce、および Math 式の代わりに Classic Conditions を使用する場合、\$values変数は Classic Condition の Ref ID と条件の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 Classic Condition がある場合、\$valuesには 2 つの条件 B0と が含まれますB1。

```
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。

### 引数

`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

この `externalURL` 関数は、ini ファイルで設定された Grafana サーバーの外部 URL を返します (複数可)。

例

```
{{ externalURL }}
```

```
https://example.com/grafana
```

## graphLink

`graphLink` 関数は、指定された式とデータソース [Grafana バージョン 9 でを試すの](#) のグラフィカル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}]
```

## ヒューマナイズ

`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 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、時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
```

一致

match 関数は、テキストを正規表現パターンと照合します。

例

```
{{ match "a.*" "abc" }}
```

```
true
```

pathPrefix

このpathPrefix関数は、ini ファイルで設定された Grafana サーバーのパスを返します (複数可)。

例

```
{{ pathPrefix }}
```

```
/grafana
```

tableLink

tableLink 関数は、指定された式とデータソース[Grafana バージョン 9 でを試すの](#)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!
```

`toLowerCase`

`toLowerCase` 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!
```

`toUpperCase`

`toUpperCase` 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!
```

`replaceAll`

`replaceAll` 関数は、正規表現に一致するテキストを置き換えます。

例

```
{{ replaceAll "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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、アラートルールの状態の3つの主要コンポーネントがあります。関連していますが、各コンポーネントは微妙に異なる情報を伝えます。

## 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	評価エンジンによって返される時系列は少なくとも 1 つあります Pending。
ファイリング	評価エンジンから返される時系列は少なくとも 1 つあります Firing。

**i** Note

アラートは最初に に移行 pending し firing、次に に移行するため、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]	発音も保留もしないアラートの状態。すべてが正しく動作しています。
保留中	設定されたしきい値の期間未満にアクティブだったアラートの状態。
アラート	設定されたしきい値の期間よりも長くアクティブだ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ていません。
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

アラートルールには、次のいずれかのヘルスステータスがあります。

状態	説明
大阪	アラートルールを評価するときにエラーは発生しません。
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	ルール評価中に少なくとも 1 つの時系列にデータがないこと。

## NoDataおよびの特別なアラート **Error**

アラートルールを評価すると、状態 NoData または が生成されます。Grafana アラートは Error、次の追加ラベルを持つアラートインスタンスを生成します。

ラベル	説明
アラート名	状態 DatasourceError に応じて DatasourceNoData または 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、無音の追加、コンタクトポイントへのルーティングなどによって、通常のアラートと同じ方法で処理できます。

## 連絡先

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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には、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含めることができます。アラートルールが発せられると、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 Alertmanager 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タイプを示します。

名前	型
Amazon SNS	sns
OpsGenie	opsgenie
ページャーの義務	pagerduty
Slack	slack
VictorOps	victorops

連絡先の詳細については、[コンタクトポイントの使用](#)「」および「」を参照してください[通知をカスタマイズする](#)。

## 通知

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana は、アラートマネージャーを使用して、発音と解決されたアラート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 Alertmanager がありますが、[Prometheus Alertmanager などの他の Alertmanager](#) からの通知の送信もサポートしています。Grafana Alertmanager は、通知ポリシーと連絡先を使用して、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、通知が送信されるタイミングと場所を制御します。通知ポリシーは、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

送信頻度を制御したり、1日の特定時刻および特定の曜日に通知を禁止する1つ以上のミュートタイミングを設定したりできます。

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に整理されます。ルートポリシーは1つだけ存在でき、ルートポリシーは削除できません。

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、アラートに一致します。

特定のルーティングポリシーには、アラートの追加マッチングを可能にする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ます。特定のルーティングポリシーの例としては、インフラストラクチャアラートを Ops チームに送信することが挙げられます。一方、子ポリシーは優先度の高いアラートを Pagerduty に送信し、優先度の低いアラートを Slack に送信する場合があります。

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、特定のルーティングポリシーごとに検索され、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つとアラートの照合を試みることができます。アラートに一致するネストされたポリシーがない場合、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場合、または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になります。

## 連絡先

連絡先には、通知を送信するための設定が含まれています。連絡先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持つことができます。例えば、1つの連絡先に Pager Duty 統合、Pager Duty と Slack 統合、または Pager Duty 統合、Slack 統合、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。その場合、通知は送信されません。

アクセスポイントは、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つの連絡先のみアラートを送信できますが、1つの連絡先を複数の通知ポリシーに同時に追加できません。アラートが通知ポリシーに一致すると、アラートはその通知ポリシーの連絡先に送信され、その後、その設定の各統合に通知が送信されます。

**Note**

コンタクトセンターでサポートされている統合の詳細については、「」を参照してください [連絡先](#)。

## テンプレティング通知

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連絡先に限定されず、同じ連絡先内の多数の統合や、異なる連絡先間の統合でも使用できます。例えば、Grafana ユーザーは というテンプレートを作成し custom\_subject\_or\_title、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 [Go のテンプレート言語](#) で記述され、アラートページのコンタクトポイントタブにあります。

## 無音

無音を使用して、1 つ以上の発音ルールからの通知をミュートできます。サイレンスは、アラートの発音や解決を停止したり、ユーザーインターフェイスでアラートの発音を非表示にしたりしません。無音は、分、時間、日、月、または年で設定できる期間だけ続きます。

## アラートの設定

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください [Grafana バージョン 10 での作業](#)。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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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を追加する](#)
- [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を追加する

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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 Grafana を設定します。この外部アラートマネージャーは、Grafana 自体内から設定および管理できます。

Alertmanager を追加したら、Grafana アラート UI を使用して、無音、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。

### **i** Note

Grafana 9.2 以降、アラートページの管理者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。今後のリリースで削除されます。

これで、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、外部アラートマネージャーを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、URL で外部アラートマネージャーを設定するときに以前に表示されていた HTTP 基本認証情報も暗号化できます。

外部 Alertmanager を追加するには、次のステップを実行します。

1. 設定、データソースの順にクリックします。
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handleGrafanaManagedAlertsのフラグをに設定trueして、Grafana マネージドアラートをこのアラートマネージャーに送信します。

#### Note

Alertmanager 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 の問い合わせポイントと通知ポリシーは読み取り専用です。

4. 保存と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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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が異なる場所に常駐しています。これを複数のチームや組織にまたがってスケーリングすることは、特に困難な作業です。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、保守できるため、このプロセスが簡単になります。

次の 2 つのオプションから選択できます。

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、API でプロビジョニングされたアラートルールを Grafana UI から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、次のリクエストに x-disable-provenance ヘッダーを追加します。



```
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、テンプレートをサポートしています。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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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、保守できます。

Terraform を使用したアラートリソースの管理の詳細については、Terraform ドキュメントの[Grafana プロバイダー](#)の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. を実行して terraform apply、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 以降を使用していることを確認します。
- 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 キーを作成する

[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 アラートサポートで自動的に機能します。Terraform で使用するキーの作成の詳細については、[「Amazon Managed Grafana オートメーションに Terraform を使用する」](#)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「アラートルールのプロビジョニング API へのアクセス」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確認するには、Terraform ドキュメントの [「Grafana 認証」](#)を参照してください。

#### Terraform プロバイダーを設定する

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>
}
```

## 連絡先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。外部システムに接続する方法と通知を配信する場所を Grafana に伝えます。15 種類以上の[統合](#)から選択できます。この例では、Slack コンタクトポイントを使用します。

### 連絡先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 ウェブフック URL (またはその他の連絡先

この例では、アラート通知を Slack に送信する連絡先を作成します。

```
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. テキストフィールドに通知のテキストを入力します。

text フィールドは、[Go スタイルのテンプレート](#) をサポートします。これにより、Grafana アラート通知テンプレートを Terraform で直接管理できます。

3. terraform apply コマンドを実行します。
4. Grafana UI に移動し、連絡先の詳細を確認します。

UI から Terraform を介して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動作することを確認します。

#### Note

同じテンプレートを多くのコンタクトポイントで再利用できます。上記の例では、ステートメントを使用して埋め込まれた共有テンプレート `{{ template "Alert Instance Template" . }}`

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
}
```

## プロビジョニング通知ポリシーとルーティング

通知ポリシーは、アラートインスタンスをルーティングする方法を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、以前に定義したコンタクトポイントに発声アラートを接続します。

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、アラートはグループ化されています。つまりalertname、同じ名前のアラートから送信される通知はすべて、同じ Slack メッセージにグループ化されます。

特定の通知を異なる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さまざまな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**

UI から Terraform 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。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動作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は、定義された期間のアラート通知をミュートする機能を提供します。

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され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**

UI から Terraform 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。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動作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、Grafana データソースに対してアラートを送信できます。これは、既に設定したデータソースでも、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. クエリを実行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、[テスト用の TestData データソースを設定する](#)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動しているかどうかを確認できます。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、Slack をコンタクトポイントとして選択した場合、Grafana の埋め込み [Alertmanager](#) は自動的にメッセージを Slack 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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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。

プロビジョニングされたリソースにはプロビジョニングされた というラベルが付けられるため、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り、ホットリロードが実行されたりした場合に上書きされるリソースに変更が加えられるのを防ぐことができます。

## クラシック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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana アラートを使用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、この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 アラートのオンと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、「」を参照してください[Amazon Managed Grafana ワークスペースを設定する](#)。

**Note**

Grafana アラートを使用する場合、Prometheus ではなく Grafana で定義されたアラートルールは、複数の通知を連絡先に送信します。ネイティブの Grafana アラートを使用している場合は、従来のダッシュボードアラートを維持し、新しい Grafana アラート機能を有効にしない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、Prometheus Alertmanager で作成されたアラートの通知を 1 つだけ送信する Grafana Alerting を有効にすることを勧めます。

## 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。Grafana インスタンスのアラートページで、移行されたアラートを新しいアラートとともに表示できます。Grafana アラートを使用すると、Grafana が管理するアラートルールは、一致すると 1 つのアラートではなく、複数の通知を送信します。

クラシック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は、それらを保存するフォルダのアクセス許可によって管理されます。移行中、クラシックダッシュボードのアラート許可は、次のように新しいルール許可と一致します。

- 元のアラートのダッシュボードに アクセス許可がある場合、移行は元のダッシュボードのアクセス許可 (フォルダから継承されたアクセス許可を含む) と一致する Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1} のように、この形式でという名前のフォルダを作成します。
- ダッシュボードのアクセス許可がなく、ダッシュボードがフォルダにある場合、ルールはこのフォルダにリンクされ、そのアクセス許可を継承します。
- ダッシュボードのアクセス許可がなく、ダッシュボードが General フォルダにある場合、ルールは General Alerting フォルダにリンクされ、ルールはデフォルトのアクセス許可を継承します。

**Note**

Grafana アラート NoData には Keep Last State のオプションがないため、このオプションは従来のルールの移行 NoData 中になくなります。Error 処理 Keep Last State のオプションは、新しいオプションに移行されます Error。の動作に一致させるために Keep

Last State、どちらの場合も、移行中に Amazon Managed Grafana は 1 年間の期間を持つアラートルールごとに自動的に無音を作成します。

通知チャンネルは、適切なルートとレシーバーを持つ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ンネルは、デフォルトのルートへの問い合わせポイントとして追加されます。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ンネルは、autogen-unlinked-channel-recvr ルートに移動します。


### 制限事項

- Grafana アラートシステムは、使用可能なすべての Prometheus、Loki、Alertmanager データソースからルールを取得できます。サポートされている他の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行き来すると、1 つのシステムでサポートされている機能のデータが失われる可能性があります。他のシステムでは失われない可能性があります。

#### Note

クラシックダッシュボードアラートに戻ると、Grafana アラートを有効にしている間に行われたアラート設定に加えられたすべての変更が失われます。これには、作成された新しいアラートルールも含まれます。

## アラートルールを管理する

-  このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

アラートルールは、アラートが発生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、およびオプションで条件が満たされる期間で構成されます。

クエリと式が評価するデータセットを選択する間、条件はアラートを作成するためにアラートが満たす必要があるか超える必要がある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。期間を設定すると、条件を満たす必要がある期間を示します。アラートルールは、データがない場合のアラート動作を定義することもできます。

#### 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 マネージドアラートルールの作成](#)
- [Grafana Mimir または Loki マネージドアラートルールの作成](#)
- [Grafana Mimir または Loki が管理する録音ルールの作成](#)
- [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 マネージドアラートルールの作成

- ⚠ このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana では、1 つ以上のデータソースをクエリし、結果を減少または変換して、相互に比較したり、固定しきい値と比較したりするアラートルールを作成できます。これらが実行されると、Grafana はコンタクトポイントに通知を送信します。

Grafana マネージド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、既存のアラートを一覧表示するアラートページを開きます。
2. 新しいアラートルール を選択します。
3. ステップ 1 で、ルール名、タイプ、および保存場所を次のように追加します。
  - 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` ラベルでもあります。
  - ルールタイプのドロップダウンから、Grafana マネージドアラート を選択します。
  - フォルダドロップダウンから、ルールを保存す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には、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。
  - デフォルト名のままにするか、カーソルを合わせ、編集アイコンを選択して名前を変更します。
  - クエリの場合は、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 `Classic` 条件を選択して 1 つのアラートルールを作成するか、数学、削減、再サンプリングオプションを選択して、シリーズごとに個別のアラートを生成します。これらのオプションの詳細については、「」を参照してください [単一および多次元ルール](#)。
  - クエリの実行 を選択して、クエリが成功したことを確認します。
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、クエリまたは式を選択してアラートルールを開始します。
  - すべての を評価する で、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。
  - の評価 では、アラートが開始される前に条件が `true` である必要がある期間を指定します。

**Note**

条件に違反すると、アラートは Pending状態になります。指定された期間、条件が超過したままの場合、アラートは Firing状態に移行します。満たされなくなった場合は、Normal状態に戻ります。

- 「データとエラー処理を設定しない」で、データがない場合のアラート動作を設定します。のガイドラインを使用します [データまたはエラーケースなしの処理](#)。
  - アラートのプレビューを選択して、現時点でクエリを実行した結果を確認します。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、ルールに関連付けられたメタデータを追加します。
- 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [ラベルと注釈](#)。
  - ランブック URL、パネル、ダッシュボード、アラート IDs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、「[アラート通知を管理する](#)」を参照してください。

## 単一および多次元ルール

Grafana マネージド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。

### 単次元ルール (クラシック条件)

クラシック条件式を使用して、条件が満たされたときに単一のアラートを開始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。その結果、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 は 1 つのアラートのみを送信します。

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。アラートのクォータに達した場合、ルールは評価されません。多次元ルールのクォータの詳細については、「」を参照してください[クォータ到達エラー](#)。

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。

- 選択した時間範囲の値を1つの値に集約するには、各クエリにReduce式を追加します。( [数値データを使用するルール](#) には必要ありません )。
- ルールの条件を持つMath式を追加します。これは、ルールがアラートを開始すべきでない場合、クエリまたは reduce 式が既に 0 を返す場合や、アラートを開始する必要がある場合は正の数値を返す場合は必要ありません。

例:

- $\$B > 70$  B クエリ/式の値が 70 を超える場合にアラートを開始するかどうか。
- $\$B < \$C * 100$  は、B の値が C の値に 100 を掛けた値よりも小さい場合にアラートを開始する必要があります。比較される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、一方が他のクエリのサブセットである場合、それらが照合されます。

#### Note

Grafana は、テンプレート変数を使用したアラートクエリを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「[テンプレート変数は、アラートの設定中にアラートクエリでサポートされていません](#)」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、アラートクォータにカウントされます。アラートクォータ内で対応できる数よりも多くの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 ワークスペースのパフォーマンスや、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際の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。以下の考慮事項は、モニタリングシステムのパフォーマンスを最適化する際に役立ちます。

- **ルール評価の頻度** – アラートルールのすべての評価プロパティは、ルール評価の頻度を制御します。許容される最も低い評価頻度を使用することをお勧めします。
- **結果セットの基数** – ルールで作成するアラートインスタンスの数は、そのパフォーマンスに影響します。フリート内のすべての VM で、すべての API パスの API レスポンスエラーをモニタリングしているとします。このセットには、パスの数に VMs。例えば、VM ごとのパスごとではなく、VM ごとの合計エラーをモニタリングすることで、結果セットの基数を減らすことができます。
- **クエリの複雑さ** – データソースが処理して応答できるクエリは、消費するリソースをすばやく減らすことができます。この考慮事項は上記の他の考慮事項ほど重要ではありませんが、それらをできるだけ減らすと、個々のクエリのパフォーマンスを見ると違いが生じる可能性があります。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してください。アラートクエリは、データベースのモニタリングによって処理されるクエリの大部分であることが多いため、Grafana インスタンスに影響するのと同じ負荷要因がクエリにも影響します。

## 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があります。この数に達すると、そのワークスペースに新しい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化する可能性があります。

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は次のとおりです。

- 単一ディメンションルールのみを作成する場合、各ルールは単一のアラートインスタンスです。1 つのワークスペースに、アラートインスタンスのクォータと同じ数のルールを作成できます。
- 多次元ルールは複数のアラートインスタンスを作成しますが、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すると、作成時に 50 個の EC2 インスタンス (つまり 50 個のアラートインスタンス) が存在する可能性があります。1 週間後にさらに 10 個の EC2 インスタンスを追加すると、次の評価には 60 個のアラートインスタンスが含まれます。

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ます。アラートインスタンスのクォータをすぐに超えるアラート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わる可能性があるため、クォータはルールが評価されるたびにチェックされます。

- ルールの評価時に、ルールによってアラートインスタンスのクォータを超えた場合、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アラートルールが更新されるまで、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使用します)。通知には、値の `Error` 注釈が含まれていません `QuotaReachedError`。
- が評価 `QuotaReachedError` を停止するルール。評価は、更新が行われ、更新後の評価によってそれ自体がが発生しない場合にのみ再開されます `QuotaReachedError`。評価されていないルールは、Grafana コンソールでクォータ到達エラーを表示します。
- アラートルールを削除するか、多次元アラートを編集してアラートインスタンスの数を減らすことができます (例えば、VM 内の API ごとに 1 つのアラートではなく、VM ごとに 1 つのエラーに対して 1 つのアラートを設定するなど)。
- 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、アラートインスタンスの数を減らすために他の変更を加えた場合は、変更なしで保存することもできます。再開できる場合は、です。別のが発生した場合は `QuotaReachedError`、保存できません。
- アラートが保存され、アラートクォータを超過せずに評価を再開すると、クォータ到達エラーがしばらくの間 (評価間隔まで) Grafana コンソールに表示され続ける可能性があります。ルールのしきい値が満たされると、アラートルールの評価が開始され、アラートが送信されます。
- アラートクォータおよびその他のクォータの詳細については、「」を参照してください [Amazon Managed Grafana サービスクォータ](#)。

## データまたはエラーケースなしの処理

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のオプションを選択します。

データを処理しないためのオプションを次の表に示します。

データなしオプション	Behavior
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

データなしオプション	Behavior
	DatasourceNoData を含むアラートを作成します。
アラート	アラートルールの状態を に設定し ずAlerting。
OK	アラートルールの状態を に設定し ずNormal。

エラーケースを処理するためのオプションを次の表に示します。

エラーまたはタイムアウトオプション	Behavior
アラート	アラートルールの状態を に設定する Alerting
OK	アラートルールの状態を に設定する Normal
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 DatasourceError を含むアラートを作成し ます。

## Grafana Mimir または Loki マネージドアラートルールの作成

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Grafana を使用すると、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許可があることを確認します。それ以外の場合、Cortex マネージド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は、それぞれのサービスを設定してルーラー API を有効にします。
- Loki - Loki データソースのデフォルト local である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ます。ルールを編集するには、他のストレージタイプのいずれかを設定します。
- Grafana Mimir - ではなくレガシー /api/prom プレフィックスを使用します / prometheus。Prometheus データソースは Grafana Mimir と Prometheus の両方をサポートしており、Grafana は Query API と Ruler API の両方が同じ URL にあることを想定しています。ルーラー 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たくない場合は、その設定に移動し、アラート UI によるアラートの管理チェックボックスをオフにします。

Grafana Mimir または Loki マネージド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、既存のアラートを一覧表示するアラートページを開きます。
2. アラートルールの作成 を選択します。
3. ステップ 1 では、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ラベルでもあります。
  - 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、名前を入力して作成します。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、組織的な目的のみを持つことができます。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、名前を入力して作成します。新しく作成されたルールが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 式または LogQL 式に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少なくとも 1 つある場合、ルールはアラートを開始します。各シリーズにアラートが作成されます。

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 テキストボックスで、アラートが開始される前に条件を true にする必要がある期間を指定します。を指定した場合 5m、アラートが開始されるまでの 5 分間は条件が true である必要があります。

 Note

条件が満たされると、アラートは Pending 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である場合、アラートは Firing 状態に移行します。満たされなくなった場合、Normal 状態に戻ります。

6. ステップ 4 で、ルールに関連付けられたメタデータを追加します。
- 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [ラベルと注釈](#)。
  - ランブック URL、パネル、ダッシュボード、アラート IDs 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、「[アラート通知を管理する](#)」を参照してください。

## Grafana Mimir または Loki が管理する録音ルールの作成

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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できます。記録ルールは、頻繁に必要な式または計算コストの高い式を事前に計算し、その結果を新しい時系列の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、高速です。

### 前提条件

Grafana Mimir および Loki データソースの場合は、それぞれのサービスを設定してルーラー API を有効にします。

- Loki – Loki データソースのデフォルト local である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ます。ルールを編集するには、他のストレージタイプのいずれかを設定します。
- Grafana Mimir – Grafana Mimir を指すようにデータソースを設定する場合は、ではなくレガシー /api/promプレフィックスを使用します/prometheus。Prometheus データソースは Grafana Mimir と Prometheus の両方をサポートしており、Grafana は Query API と Ruler API の両方が同じ URL にあることを想定しています。ルーラー 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 によるアラートの管理チェックボックスをオフにします。

Grafana Mimir または Loki が管理する録音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、既存のアラートを一覧表示するアラートページを開きます。
2. アラートルールの作成 を選択します。
3. ステップ 1 で、ルールタイプ、ルール名、およびストレージの場所を次のように追加します。
  - 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ラベルでもあります。
  - 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、名前を入力して作成します。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、組織的な目的のみを持つことができます。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、名前を入力して作成します。新しく作成されたルールが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 式または LogQL 式に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開始します。各シリーズにアラートが作成されます。
5. ステップ 3 で、ルールに関連付けられたメタデータを追加します。
  - 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [アラートルールの注釈とラベル](#)。



- ランブック URL、パネル、ダッシュボード、アラート IDs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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

ルールを整理できます。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編成されます。ルール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 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したりできます。

ルールグループまたは名前空間を編集するには

1. Grafana コンソールの Grafana メニューで、アラート (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を変更します。

### **i** 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は、グループ内のルールの名前または評価間隔を変更します。例えば、ルール1mを 1 分に 1 回評価するか、30 秒に 1 回評価30sするかを選択できます。

5. [変更を保存] を選択します。



## アラートルールの表示と編集

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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タイプ別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を表示できますが、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記載されています。

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、および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、ヘルス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。

### **i** Note

プロビジョニングされたアラートのクエリ定義を表示できますが、編集することはできません。そ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。

## アラートルールを表示する

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タイプ別にグループで表示されます。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、次のテキストで説明します)。
2. として表示では、任意のオプションを選択して、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

アラートの詳細を理解する方法の詳細については、「」を参照してください [アラートルールの状態と正常性](#)。

## 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+ 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、ルールを管理するためのデフォルトのルールリストビューです。各グループを展開して、このグループのルールを表示できます。ルールをさらに展開して、その詳細を表示します。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、状態別にグループ化されたアラートルールが表示されます。このビューを使用して、どのルールがどの状態にあるかの概要を取得します。各ルールを展開して詳細を表示できます。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できます。

## フィルターアラートルール

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。

- データソースの選択 を選択し、フィルタリングするデータソースを選択することで、特定のデータソースをクエリするルールをフィルタリングして表示できます。
- ラベルで検索 で検索条件を選択して、ラベルでフィルタリングできます。例えば、と入力 `environment=production,region=~US|EU,severity!=warning` 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。

- 状態でアラートをフィルタリングし、表示する状態を選択することで、特定の状態のルールを表示するようにフィルタリングできます。

## アラートルールの編集または削除

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 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。

### 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。ルールを作成するのと同じ方法で更新を行います。詳細については、[Grafana マネージドアラートルールの作成](#)「」または「」の手順を参照してください[Grafana Mimir または Loki マネージドアラートルールの作成](#)。
3. オプションで、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

エクスポート 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よう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。

## アラート通知を管理する

- ⚠ このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラート通知を送信する方法、タイミング、場所の選択は、アラートシステムを設定する上で重要な部分です。これらの決定は、問題を迅速に解決する能力に直接影響し、重要なことを見逃すことはありません。

最初のステップとして、アラート通知の送信先となる連絡先を定義します。コンタクトポイントは、通知を照合するための送信先のセットにすることができます。通知テンプレートを問い合わせポイントに追加して、通知に再利用と一貫性のあるメッセージングを行います。

次に、通知ポリシーを作成します。これは、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、および方法に関する一連のルールです。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、アラート通知の送信先を定義します。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、通知が送信されないようにする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。

Alertmanager 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、連絡先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の使用](#)
- [通知ポリシーの使用](#)
- [通知をカスタマイズする](#)
- [Prometheus データソースのアラート通知のサイレンシング](#)
- [ミュートタイミング](#)
- [アラートグループによる表示とフィルタリング](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

Alertmanager 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、選択したチャン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 Alertmanager または外部の Alertmanager 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が生成される場所によって異なります。

### Grafana アラートマネージャー

Grafana Alertmanager は、Grafana をオンプレミスまたはオープンソースで実行する場合にデフォルトで選択できる事前設定済みの内部アラートマネージャーです。

Grafana Alertmanager は Grafana からアラートを受信できますが、例えば、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。

#### **i** Note

抑制ルールは Grafana Alertmanager 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

単一のアラートマネージャーを使用して Grafana、Loki、Mimir、Prometheus のすべてのアラートを受信する場合は、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 自体内から設定および管理できます。

Grafana Alertmanager 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、設定した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アラートマネージャーがあり、引き続き使用したいと考えています。
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と考えています。

アラートマネージャーは、アラートの連絡先ページと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 `handleGrafanaManagedAlerts`のフラグを `true` に設定して、Grafana マネージド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の使用

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

コンタクトポイントを使用して、アラートが開始され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、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持つことができます。アラートが開始されると、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します。

#### **i** Note

Grafana マネージドアラートのコンタクトポイントを作成および編集できません。Alertmanager アラートの連絡先は読み取り専用です。

### コンタクトポイントの使用

次の手順では、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 Alertmanager が選択されています。
4. 連絡先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 チャンネルと、連絡が必要な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使用可能な場合は、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、アラートが解決されたときに通知されたくない場合は、オプションで解決済みメッセージの無効化を選択します。
8. コンタクトポイントにさらに多くのコンタクトポイントタイプが必要な場合は、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手順を繰り返すことができます。
9. アクセスポイントの保存 を選択して変更を保存します。

## コンタクトポイントを編集するには

1. コンタクトポイントを選択すると、既存の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、アクセスポイントの保存を選択して変更を保存します。

連絡先が作成されたら、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できます。

## テスト通知を送信するには

1. 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開きます。
2. テストするコンタクトポイントを選択し、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独自のカスタム注釈とラベルをテスト通知に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でアラートをテストします。

通知ポリシーで使用されていない連絡先は削除できます。

コンタクトポイントを削除するには

1. 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

通知ポリシーでコンタクトポイントが使用されている場合は、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子のリスト

名前	型
Amazon SNS	sns
OpsGenie	opsgenie
ペーজァーの義務	pagerduty
Slack	slack
VictorOps	victorops

通知ポリシーの使用

 このドキュメントトピックは、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に設計されています。



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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知ポリシーは、アラートを問い合わせポイントにルーティングする方法を決定します。ポリシーにはツリー構造があり、各ポリシーには 1 つ以上の子ポリシーを含めることができます。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と一致することもできます。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 つ以上の一致後も評価が続行されます。親ポリシーの設定と連絡先情報は、子ポリシーのいずれにも一致しないアラートの動作を制御します。ルートポリシーは、特定のポリシーに一致しないアラートを管理します。

#### Note

Grafana マネージドアラートの通知ポリシーを作成および編集できます。Alertmanager アラートの通知ポリシーは読み取り専用です。

## 通知のグループ化

グループ化は、類似する性質のアラート通知を 1 つのファネルに分類します。これにより、システムの多くの部分が一度に障害が発生し、多数のアラートが同時に開始されるときに、大規模な停止中のアラート通知を制御できます。

### グループ化の例

異なる環境のデータベースに 100 のサービスが接続されているとします。これらのサービスは、ラベルによって区別されます `env=environmentname`。サービスがデータベースに到達できるかどうかをモニタリングするためのアラートルールが設定されています。アラートルールは、という名前のアラートを作成します `alertname=DatabaseUnreachable`。

サービスの半分がデータベースに到達できなくなるネットワークパーティションが発生すると、50 個の異なるアラートが開始されます。この状況では、影響を受ける環境のリストを含む 1 ページの通知 (50 ではなく) を受け取る必要があります。

グループ化を (サービスごとに異なる env ラベルを使用group\_by: [alertname]せずに) に設定できます。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を含む単一のコンパクトな通知を送信します。

## 特別なグループ

Grafana には 2 つの特別なグループがあります。デフォルトのグループでは、group\_by: null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、グループを効果的に無効にして、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

次の手順は、通知ポリシーを作成および管理する方法を示しています。

### ルート通知ポリシーを編集するには

1. Grafana コンソールの Grafana メニューで、アラート (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 (pen) 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を更新します。
6. Group by で、アラートをグループ化するラベル (または特別なグループ) を選択します。
7. タイミングオプション で、次のオプションから選択します。
  - グループ待機 — 最初の通知を送信する前に、同じグループのアラートをバッファするのを待機する時間。デフォルト値は 30 秒です。
  - グループ間隔 — グループに対する 2 つの通知間の最小時間間隔。デフォルトは 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ていない場合、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。

### 新しい最上位ポリシーを追加するには

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. ラベルの一致セクションで、一致するアラートラベルを 1 つ以上追加します。ラベルマッチングの詳細については、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に一致した後でも、後続の兄弟ノードのマッチングを継続を有効にして、兄弟ポリシーのマッチングを継続します。このポリシーを有効にすると、同じアラートに対して複数の通知を受け取ることができます。
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. オプションで、グループ通知ポリシーのタイミングオプションをオーバーライドする一般的なタイミングをオーバーライドを選択します。
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追加するには

1. ネストされたポリシーを作成する特定のポリシーを展開します。
2. ネストされたポリシーを追加 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など)。
3. [Save policy(ポリシーの保存)] を選択して変更を保存します。

特定のポリシーを編集するには

1. アラートページから通知ポリシーを選択して、既存のポリシーを一覧表示するページを開きます。
2. 編集するポリシーを選択し、編集アイコン (ペン) を選択します。
3. 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名前の一部または全体を入力します。
- ラベルで検索するには、ラベルで検索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。例えば、有効なマッチャー入力は `severity=high, region=~EMEA|NA`。

#### Note


ラベルで検索する場合、一致するポリシーはすべて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。

## ラベルマッチングの仕組み

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一致する場合、ポリシーはアラートに一致します。

- Label – 一致するラベルの名前。アラートのラベル名と完全に一致する必要があります。
- Operator – 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。
  - = 指定された文字列と正確に一致する値を持つラベルを選択します。
  - != 値が指定された文字列と一致しないラベルを選択します。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルを選択します (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を選択します。
- 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一致させることができます。

## 通知をカスタマイズする

-  このドキュメントトピックは、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に設計されています。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、「」を参照してください [Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

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ません。ただし、コンタクトポイント統合ごと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して、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
- 太字と斜体のテキストをフォーマットし、改行を追加または削除します。

通知テンプレートを次の目的で使用することはできません。

- 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設計を変更する

## トピック

- [Go のテンプレート言語の使用](#)
- [通知テンプレートを作成する](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知テンプレートは、Go のテンプレート言語である[テキスト/テンプレート](#) で記述します。

このセクションでは、Go のテンプレート言語の概要と、テキスト/テンプレートでのテンプレートの記述について説明します。

## ドット

テキスト/テンプレートにはドット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、テンプレートのどこで使用されるかによっ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`、`および`などの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、ドットは、リスト `range` 上で使用する場合、内で使用する場合、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に `with`、別のものを参照することがあります。この例については、[通知テンプレートを作成する](#)「」、およびすべてのデータと関数については「」を参照してください [テンプレートリファレンス](#)。

## タグの開閉

テキスト/テンプレートでは、テンプレートが変数を出力するか、`if` ステートメントなどの制御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、変数の印刷で `と` を使用し、制御構造で `{ }` 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%}`。

## [Print]

何かの値を出力するには、`{{ と }}` を使用します。ドットの値、ドットのフィールド、関数の結果、および [変数](#) の値を印刷できます。例えば、ドットが参照する `Alerts` フィールドを出力するには `ExtendedData`、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` を反復処理できます。

```
{{ range .Alerts }}
{{ .Labels }}
{{ end }}
```

範囲内のドットは、ではなくExtendedData、を参照しますAlert。を使用し  
て{{ .Labels }}、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ドット{{ range .Alerts }}が変更されるため機能します。範囲  
が終了すると、ドットは範囲の開始前に存在していた値にリセットされます。この例ではで  
ずExtendedData。

```
{{ 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形式で出力するテンプレートを記述しましょう。ここでThe name of the  
label is \$name, and the value is \$value、\$nameとには各ラベルの名前と値\$valueが  
含まれます。

前の例と同様に、範囲を使用してのアラートを反復処理し、ドットがアラートのリスト内の現在の  
アラートを参照Alertsするようにします。次に、ソートされたラベルで2番目の範囲を使用  
して、ドット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

テンプレート内のステートメントの場合は、を使用できます。例えば、にアラートThere are  
no alertsがない場合に出力するには.Alert、次のように記述します。

```
{{ if .Alerts }}
There are alerts
```

```
{{ else }}
There are no alerts
{{ end }}
```

で

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` を更新しての値を参照します。

```
{{ with .Alerts }}
There are {{ len . }} alert(s)
{{ else }}
There are no alerts
{{ end }}
```

## 変数

テキスト/テンプレートの変数は、テンプレート内に作成する必要があります。例えば、`$variable` 現在の値がドットの という変数を作成するには、次のように記述します。

```
{{ $variable := . }}
```

範囲内 `$variable` の または `with` を使用できます。また、変数が定義された時点でのドットの値を参照し、ドットの現在の値を参照しません。

例えば、2 番目の範囲で `with` を使用するテンプレートを記述することはできません。ここでのドットは、現在のアラートではなく、現在のラベルを参照するためです。

```
{{ 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

範囲内の各アラートのインデックスを取得するには、範囲の開始時にインデックス変数と値変数を定義します。

```
{{ $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。、、、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定義しないでください\_\_subject\_\_text\_values\_list\_\_text\_alert\_listdefault.titledefault.message。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のテンプレートが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があります。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防止したり表示したりしません。

```
{{ define "print_labels" }}
{{ end }}
```

## テンプレートを埋め込む

定義済みのテンプレートをテンプレート内に埋め込むには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ます。

```
{{ template "print_labels" . }}
```

## テンプレートにデータを渡す

テンプレートドット内では、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、ドットはその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、ドット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便利です。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、発砲アラートだけを印刷するには、次のように記述します。

```
{{ template "print_alerts" .Alerts.Firing }}
```

これは、`.Alerts`と `の両方.Alert.Firing`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 */}}
```

コメントに改行が追加され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

インデント、タブとスペースの両方、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存在します。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

テキスト/テンプレートでは、`{{-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：

```
{{ 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。

```
alername = "Test"

grafana_folder = "Test alerts"
```

各範囲の開始`-}}`時に`}}`に変更された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 }}
```

これで、テンプレートのインデントと改行がテキストに表示されなくなります。

```
alertname = "Test"
grafana_folder = "Test alerts"
```

## 通知テンプレートを作成する

**△** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

再利用可能な通知テンプレートを作成して、連絡先に送信します。

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 つ持つことはできません。、、、 \_\_subject \_\_text\_values\_list \_\_text\_alert\_list default.title 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default.message。

連絡先タブには、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. などの通知テンプレートの名前を選択しますemail.subject。
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. [Save] (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 です。
3. コンテンツフィールドに各テンプレートを書き込みます。これには、各テンプレートの最初 `{{ define "name-of-template" }}` と最後に `{{ end }}` とが含まれます。通知テンプレート内の各テンプレートには、email.subjectやなど、わかりやすい名前を使用できますemail.message。この場合、上記で入力した通知テンプレートの名前は再利用しないでください。

以下のセクションでは、作成する可能性のあるテンプレートの詳細な例を示します。

4. [Save] (保存) をクリックします。

## Eメールの件名のテンプレートの作成

次の例のように、発声アラートと解決された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. 次の内容email.subjectで というテンプレートを作成します。

```
{{ define "email.subject" }}  
{{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts  
{{ end }}
```

2. テンプレートは、templateキーワード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のテンプレートの作成

次の例のように、すべての発声アラートと解決されたアラートの概要を含む E メールメッセージ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 メールメッセージのテンプレートを作成するには

1. コンテンツに `email.message_alert` との 2 つのテンプレート `email` を使用して、という名前の通知テンプレートを作成します `email.message`。

`email.message_alert` テンプレートは、`email.message` テンプレートに E メール構造が含まれている間、発動したアラートと解決されたアラートのラベルと値を出力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、`template`キーワードを使用して Text Body 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の作成

次の例のように、発声アラートと解決された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. 次の内容`slack.title`で というテンプレートを作成します。

```
{{ define "slack.title" }}
{{ len .Alerts.Firing }} firing alerts, {{ len .Alerts.Resolved }} resolved alerts
{{ end }}
```

2. テンプレートは、`template`キーワードを使用して Title 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 メッセージの内容のテンプレートの作成

ラベル、注釈、ダッシュボード URL など、すべての発音アラートと解決されたアラートの説明を含む Slack メッセージの内容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 メッセージの内容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 `slack` 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.message`。

`slack.print_alert` テンプレートには通知の構造が含まれている間、`slack.message` テンプレートはラベル、注釈、DashboardURL を出力するために使用されます。

```
{{ 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、`template`キーワードを使用して Text Body 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を使用します。

```
{{ template "slack.message" . }}
```

共有テンプレートを使用して E メールと Slack の両方の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、このタイトルの Slack メッセージを送信する場合は 1 firing alerts, 0 resolved alerts、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. 次の内容 `common.subject_title` で という名前のテンプレートを作成します。

```
{{ 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メールの連絡先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で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すると、既存のコンタクトポイントのリストが表示されます。
2. コンタクトポイントの追加 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. メッセージ や件名 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、 の形式を使用します。 *template\_name* は `{{ template "template_name" . }}`、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に設計されています。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、「」を参照してください [Grafana バージョン 10 での作業](#)。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、「」を参照してください [Grafana バージョン 8 での作業](#)。

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。

## テンプレートデータ

次のデータがメッセージテンプレートに渡されます。

名前	型	メモ
Receiver	string	通知の送信先の連絡先の名前。
Status	string	少なくとも 1 つのアラートが発せられている場合は発せられます。それ以外の場合は解決されます。
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するラベル。
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	string	通知を送信した Grafana へのバックリンク。外部 Alertmanager を使用している場合は、この Alertmanager へのリンクを戻します。

Alerts タイプは、返されたアラートをフィルタリングするための 2 つの関数を公開します。

- Alerts.Firing – 発声アラートのリストを返します。
- Alerts.Resolved – 解決されたアラートのリストを返します。

## アラート (タイプ)

アラートタイプには、次のデータが含まれます。

名前	型	メモ
ステータス	string	firing-または-resolved
ラベル	KeyValue	アラートにアタッチされたラベルのセット。
注釈	KeyValue	アラートにアタッチされた注釈のセット。
[値]	KeyValue	Classic 条件を含むすべての式の値
StartsAt	時刻	アラートが発せられた時刻。
EndsAt	時刻	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。それ以外の場合は、最後のアラートが受信されてから設定可能なタイムアウト期間に設定されます。
GeneratorURL	string	Grafana または外部 Alertmanager へのバックリンク。
SilenceURL	string	アラートを無音にするリンク (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 マネージドアラートの場合のみ。
DashboardURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードにリンクし

名前	型	メモ
		ます。Grafana マネージドアラートの場合のみ。
PanelURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードパネルにリンクします。Grafana マネージドアラートの場合のみ。
フィンガープリント	string	アラートの識別に使用できるフィンガープリント。
ValueString	string	アラート内の各縮小式のラベルと値を含む文字列。

## ExtendedData

ExtendedData オブジェクトには、次のプロパティが含まれています。

名前	Kind	説明	例
レシーバー	string	通知を送信する連絡先の名前。	<code>{{ .Receiver }}</code>
ステータス	string	ステータスはです。firing if at least one alert is firing, otherwise resolved.	<code>{{ .Status }}</code>
アラート	<code>[]Alert</code>	この通知のすべての発動アラートと解決済みアラートのリスト。	There are <code>{{ len .Alerts }}</code> alerts

名前	Kind	説明	例
アラートの送信	<code>[]Alert</code>	この通知のすべての発動アラートのリスト。	<code>There are {{ len .Alerts.Firing }} firing alerts</code>
解決済みのアラート	<code>[]Alert</code>	この通知で解決されたすべてのアラートのリスト。	<code>There are {{ len .Alerts.Resolved }} resolved alerts</code>
GroupLabels	KeyValue	これらのアラートをグループ化するラベルは、通知に含まれます。	<code>{{ .GroupLabels }}</code>
CommonLabels	KeyValue	この通知のすべてのアラートに共通するラベル。	<code>{{ .CommonLabels }}</code>
CommonAnnotations	KeyValue	この通知のすべてのアラートに共通する注釈。	<code>{{ .CommonAnnotations }}</code>
ExternalURL	<code>string</code>	この通知を送信した Grafana ワークスペースまたは Alertmanager へのリンク。	<code>{{ .ExternalURL }}</code>

## KeyValue タイプ

KeyValue タイプは、ラベルと注釈を表すキーと値の文字列ペアのセットです。

として保存されたデータの直接アクセスに加えて KeyValue、データのソート、削除、変換の方法もあります。

名前	引数	戻り値	メモ	例
SortedPairs		キーと値の文字列ペアのソート済みリスト		<code>{{ .Annotations.SortedPairs }}</code>
Remove	[]文字列	KeyValue	指定されたキーなしでキー/値マップのコピーを返します。	<code>{{ .Annotations.Remove "summary" }}</code>
名前		[]文字列	ラベル名のリスト	<code>{{ .Names }}</code>
[値]		[]文字列	ラベル値のリスト	<code>{{ .Values }}</code>

## Time (時間)

時間は Go [time](#) パッケージからのものです。時刻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形式で出力するには Monday, 1st January 2022 at 10:00AM、次のテンプレートを記述します。

```
{{ .StartsAt.Format "Monday, 2 January 2006 at 3:04PM" }}
```

Go の時刻形式のリファレンスは、[で](#)確認できます。

## 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な通知を生成できます。以下の関数を使用できます。

名前	引数の型	戻り型	説明
humanize	数値または文字列	string	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

名前	引数の型	戻り型	説明
			可能な形式に変換します。
humanize1024	数値または文字列	string	ヒューマナイズと同様に、は 1000 ではなく 1024 をベースとして使用します。
humanizeduration	数値または文字列	string	時間を秒単位でより読み取りやすい形式に変換します。
humanizePercentage	数値または文字列	string	比率値を 100 の分数に変換します。
humanizeTimestamp	数値または文字列	string	Unix タイムスタンプを秒単位でより読みやすい形式に変換します。
title	string	string	strings.Title、各単語の最初の文字を大文字にします。
toUpper	string	string	strings.ToUpper、すべての文字を大文字に変換します。
toLowerCase	string	string	strings.ToLower、すべての文字を小文字に変換します。
match	パターン、テキスト	ブール値	regexp. アンカーされていない regexp 一致の MatchString テスト。



名前	引数の型	戻り型	説明
reReplaceAll	パターン、置換、テキスト	string	Regex.ReplaceAllS tring Regex 置換、アンカーなし。
graphLink	string - exprおよび datasource フィールドを含む JSON オブジェクト	string	指定された式とデータソースについて、Explore でグラフィカルビューへのパスを返します。
tableLink	string - exprおよび datasource フィールドを含む JSON オブジェクト	string	指定された式とデータソースについて、Explore の表形式ビューへのパスを返します。
args	[]インターフェイス{}	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。この関数を使用して、複数の引数をテンプレートに渡します。
externalURL	なし	string	外部 URL を表す文字列を返します。
pathPrefix	なし	string	外部 URL のパスを返します。

次の表は、各関数の使用例を示しています。

機能	TemplateString	入力	予想
ヒューマナイズ	{ \$value をヒューマナイズ }	1234567.0	1.235 メートル
humanize1024	{ humanize1024 \$value }	1048576.0	1Mi
humanizeDuration	{ humanizeDuration \$value }	899.99	14 分 59 秒
humanizePercentage	{ humanizePercentage \$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
一致	{ 「a+」 \$labels.instance } に一致	aa	true
reReplaceAll	{{ reReplaceAll "localhost:(.*)" "my.domain:\$1" \$labels.instance }}	localhost:3000	my.domain:3000
graphLink	{{ graphLink "{ \"expr\": \"up\", \"datasource\": \"gdev-prometheus\" }" }}		/explore?left=["now-1h","now"]gdev-prometheus,{"datasource":"gdev-prometheus","expr":"up","instant":false,"range":true}]
tableLink	{{ tableLink "{ \"expr\": \"up\", \"datasource\"		/explore?left=["now-1h","now"]gdev-prom

機能	TemplateString	入力	予想
	<code>\":\"gdev-prometheus \"} } }</code>		<code>etheus\",{\"datasour ce\"gdev-prometheus \"expr\"up\"instant\": true,\"range\":false}]</code>
args	<code>{{define \"x\"}}{{.arg0}} {{.arg1}}{{end}}{ template \"x\" (args 1 \"2\")}}</code>		1 2
externalURL	<code>{ externalURL }</code>		<code>http://localhost/path/ prefix</code>
pathPrefix	<code>{ pathPrefix }</code>		<code>/path/prefix</code>

## Prometheus データソースのアラート通知のサイレンシング

**⚠** このドキュメントトピックは、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に設計されています。  
 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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
 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、「」を参照してください[Grafana バージョン 8 での作業](#)。

外部アラートマネージャーデータソース (Amazon Managed Service for Prometheus を含む) の場合、無音でアラート通知を抑制できます。無音は通知の作成のみを停止します。無音はアラートルールの評価を妨げず、インスタンスのアラートがユーザーインターフェイスに表示されないようにしません。アラートを無音にすると、アラートが抑制される時間ウィンドウを指定します。

外部 Alertmanager データソースの無音を設定できます。

**Note**

アラート通知を一定の時間間隔で抑制したり、他のデータソースに対して抑制したり (定期的なメンテナンス期間中など)、無音 [ミュートタイミング](#) ではなく [ミュート](#) を使用します。

無音を追加するには

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きます。
2. サイレンス を選択して、既存の を一覧表示するページを開きます [コンタクトポイントの使用](#)。
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. 無音の追加 を選択します。
5. 無音の開始と終了日を選択して、無音が有効になるタイミングと終了するタイミングを示します。

終了時間を設定する代わりに、期間 で無音が強制される時間を指定します。これにより、無音の開始フィールド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。マッチャーは、無音が適用されるルールを決定します。ラベルマッチングについては、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 を追加するか、作成者を変更して無音の所有者を設定します。
8. 作成 を選択して、無音を作成します。

編集アイコン (ペン) を選択すると、既存の無音を編集できます。

### アラート抑制のラベルマッチング

無音を作成するときは、無音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、アラートを抑制するために一致する必要があるラベルに関する一連のルールです。一致するラベルは 3 つの部分で構成されます。

- Label – 一致するラベルの名前。アラートのラベル名と完全に一致する必要があります。
- Operator – 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。
  - = 指定された文字列と正確に一致する値を持つラベルを選択します。
  - != 値が指定された文字列と一致しないラベルを選択します。

- `=~` 指定された文字列の正規表現解釈値と一致するラベルを選択します (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を選択します。
- 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一致させることができます。

無音は指定された終了日に終了しますが、抑制はいつでも手動で終了できます。

サイレンスを手動で終了するには

1. アラートページで、無音を選択して既存の無音のリストを表示します。
2. 終了する無音を選択し、無音 を選択します。これにより、アラートの抑制は終了します。

#### Note

サイレンシングを解除すると、終了時刻が現在の時刻に設定されているかのように、アラート抑制が終了します。(自動または手動で) 終了した無音は 5 日間保持され、一覧表示されます。リストから無音を手動で削除することはできません。

無音作成フォームへのリンクの作成

詳細がすでに入力されている無音作成フォームへの URL を作成できます。オペレーターはこれを使用して、運用イベント中にアラームをすばやく抑制できます。

無音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`[label][operator][value]`、カンマで区切られた形式の 1 つ以上の値が必要です。

URL の例

一致するラベルと を含む無音フォームにリンク`severity=critical`するには`cluster!~europe-.*`、というコメントを付けて`Silencing critical EU alerts`、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ます。

```
https://mygrafana/alerting/silence/new?matchers=severity%3Dcritical%2Ccluster!~europe-
*&comment=Silence%20critical%20EU%20alert
```

外部 Alertmanager の新しい無音ページにリンクするには、などの Alertmanager データソース名で alertmanager クエリパラメータを追加し、alertmanager=myAlertmanagerdatasource。

## ミュートタイミング

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 8 での作業](#)。

ミュートタイミングは、ポリシーの新しい通知が生成または送信されない定期的な間隔です。定期的なメンテナンス期間など、アラートが特定の期間に発生し、繰り返し発生するのを防ぐために使用します。

無音と同様に、ミュートタイミングはアラートルールの評価を妨げたり、ユーザーインターフェイスにアラートインスタンスが表示されないようにしたりしません。通知の作成のみを防ぎます。

外部 Alertmanager データソースの Grafana マネージド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。

## 無音と比較したミュートタイミング

次の表は、ミュートのタイミングと無音の違いを示しています。

ミュートタイミング	無音
繰り返し発生する可能性のある時間間隔定義を使用します。	開始時刻と終了時刻が固定されています。
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、無音にするかどうかを決定します。
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。	外部のアラートマネージャーでのみ動作します。

## ミュートタイミングを作成するには

1. Grafana コンソールの Grafana メニューで、アラート (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を追加」ボタンを選択します。
5. ミュートタイミングを適用する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。

## 通知ポリシーにミュートタイミングを追加するには

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。

ポリシーの保存ボタンを選択します。

## 時間間隔

時間間隔は、時間範囲の定義です。この間隔でアラートが開始されると、アラートは抑制されます。範囲は を使用してサポートされます：(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 (次のリストの詳細) で構成され、アラートを抑制するには、すべて一致する必要があります。例えば、曜日monday:fridayと時間範囲を 8:00~9:00 に指定すると、アラートは月曜日から金曜日まで 8~9 に抑制されますが、例えば土曜日の 8~9 は抑制されません。

- 時間範囲 – 通知を抑制する時刻。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。時刻の例は です14:30。時間は UTC で 24 時間表記されます。
- 曜日 – 曜日。などの 1 日monday、などの範囲monday:friday、または などのカンマ区切りの日数リストを使用できますmonday, tuesday, wednesday。
- 月 – 選択する月。月は、数値の指定で指定することも、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両方で 1 月を指定します。1 か月、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。

- Days of the month – 1 か月内の日付。値の範囲は 1 ~ です31。負の値は月の曜日を逆の順序で指定するため、-1は月の最後の日を表します。月の日は、1日、日の範囲、またはカンマ区切りの日のリストとして指定できます。
- 年 - 間隔の年。例えば 2023:2025 です。

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 に設定すると2023:2025, 2027、2023年、2024年、2025年、2027年(2026年ではない)に当てはまります。

フィールドを空白のままにすると、任意の時点がフィールドと一致します。完全な時間間隔と一致するには、特定の時点がすべてのフィールドと一致する必要があります。

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UTCの12:00から24:00まで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：
  - 開始時刻: 12:00
  - 終了時刻: 24:00
- 曜日: monday
- 月: 3, 6, 9, 12
- 曜日: 1:7

## アラートグループによる表示とフィルタリング

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 8 での作業](#)。

アラートグループには、Alertmanager インスタンスからのグループ化されたアラートが表示されます。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。一般的なアラートルールを1つのアラートグループにグループ化すると、重複するアラートルールが起動されなくなります。

アラートグループを表示したり、特定の条件に一致するアラートルールをフィルタリングしたりできます。

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、既存のアラートを一覧表示するアラートページを開きます。
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のアラートマネージャーを選択します。
4. カスタムグループからドロップダウンでラベルの組み合わせを選択すると、デフォルト以外のグループが表示されます。これは、通知ポリシーのグループ化のデバッグと検証に役立ちます。

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のいずれかで指定されたラベルがアラートに含まれていない場合、アラートは のヘッダーを持つすべてのグループをキャッチに追加されますNo grouping。

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` です。

状態別にフィルタリングするには

- 状態 で、アクティブ状態、抑制状態、または未処理状態のいずれかを選択して、選択した状態に一致するアラートを表示します。その他のアラートはすべて非表示になります。

通知エラーの表示

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 8 での作業](#)。

通知エラーを表示し、送信に失敗した、または受信されなかった理由を理解します。

#### Note

この機能は Grafana Alertmanager 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、既存のアラートを一覧表示する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すると、既存のコンタクトポイントのリストが表示されます。

いずれかの問い合わせポイントに障害が発生した場合、画面の右隅にあるメッセージにより、エラーがある旨と数をユーザーに警告します。

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。

連絡先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. Health 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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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 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](#)
- [探索](#)
- [Linking \(リンク中\)](#)
- [テンプレートと変数](#)
- [Grafana アラート](#)

## パネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

パネルは、Grafana サーバーの基本的な視覚化構成要素です。パネルは、いくつかの特殊用途パネルを除く 1 つ以上のクエリを視覚的に表現したものです。クエリは時間の経過とともにデータを表示します。これには、温度変動から現在のサーバーステータス、ログやアラートのリストまで、さまざまなものがあります。

各パネルには、パネルで選択されたデータソースに固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、視覚化を抽出してパネルに表示することができます。

データを表示するには、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、「」を参照してください[データソースに接続する](#)。

各パネルには、さまざまなスタイルと書式設定のオプションがあります。パネルは、ドラッグ、再配置、サイズ変更できます。

## トピック

- [パネルの追加](#)
- [パネルの削除](#)
- [クエリ](#)
- [記録されたクエリ](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、パネルを構築するための基本的な手順について説明します。

ダッシュボードにパネルを追加するには

1. パネルを追加するダッシュボードを選択します。
2. パネルの追加アイコンを選択します。

### 3. 新しいパネルを追加を選択します。

Grafana ワークスペースは、デフォルトのデータソースを選択した空のグラフパネルを作成します。

### 4. 必須ではありませんが、パネルに便利なタイトルと説明を追加することをお勧めします。オプションで、いずれかの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。詳細については、「[テンプレートと変数](#)」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されたテキストは、パネルエディタのパネルの上部とダッシュボードに表示されます。
- 説明 – このフィールドに入力されたテキストは、パネルの左上隅にあるツールヒントに表示されます。パネルの説明と表示するデータを入力します。

### 5. パネルのクエリを記述します。視覚化を表示するには、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で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含まれます。データソースの詳細については、「」を参照してください。[データソースに接続する](#)。
- b. データソースのクエリ言語でクエリを記述または構築します。オプションは異なります。特定のガイドラインについては、特定のデータソースドキュメントを参照してください。

### 6. パネルタブの視覚化セクションで、視覚化タイプを選択します。Grafana ワークスペースには、その視覚化が適用され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、変更を説明するメモを追加することをお勧めします。メモは、ダッシュボードを以前のバージョンに戻す必要がある場合に役立ちます。

### 8. ダッシュボードを保存するには、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に設計されています。

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、「」を参照してください。[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Grafana で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。

または

- ポインタをパネルにカーソルを合わせ、キーボードショートカットシーケンス「pr」を使用します。

## クエリ

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Grafana ワークスペースパネルは、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。クエリが適切に形成されている場合、データソースは応答します。パネル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の数を調整できます。

Grafana ワークスペースは、パネルごとに最大 26 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、クエリエディタによって自動補完、メトリクス名、または変数の提案が提供される場合があります。

クエリ言語が異なるため、データソースのクエリエディタは異なる場合があります。

## クエリ構文

データソースには、データを要求するクエリ言語と構文が異なります。2つのクエリの例を次に示します。

### 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記述の詳細については、そのデータソースのドキュメントを参照してください。データソースについては、「」を参照してください[データソースに接続する](#)。

## クエリタブ UI

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ト
- クエリオプション
- クエリインスペクターボタン
- クエリエディタのリスト

### データソースセレクト

データソースセレクトはドロップダウンリストです。追加した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 はデフォルトのデータソースを自動的に選択します。データソースの詳細については、「」を参照してください[データソースに接続する](#)。

Grafana ワークスペースで設定したデータソースに加えて、3つの特別なデータソースを使用できます。

- TestDataDB – ランダムなウォークデータを生成する組み込みデータソース。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。このデータソースを選択すると、追加する新しいクエリごと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したデータソースを使用します。
  - 混合データソースを使用するように既存のクエリを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の結果セットを使用するためのデータソース。

## 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クエリオプションを使用できます。

- 最大データポイント – データソースがサポートしている場合、は返される各シリーズ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、データソースはそれらを統合します (平均、最大、またはその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理由は、パフォーマンスとラインのスムーズ化の 2 つです。デフォルト値はグラフの幅 (またはピクセル数) です。これにより、グラフパネルが表示できる量よりも多くの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、最大データポイント値がローリングバッファに使用されます。(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 — 自動的に計算される間隔、通常は最小スクレイプ間隔の最小制限を設定します。データポイントが 15 秒ごとに保存される場合、それより短い間隔は必要ありません。もう 1 つのユースケースは、スクレイプ間隔よりも最小値を高く設定して、より粗く、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ごとに集計またはグループ化するときを使用できる時間範囲。

Amazon Managed Grafana は、テンプレートクエリの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` のいずれかです。



通常、sumやなどの集計関数で使用されますaverage。例えば、これは間隔変数を使用した Prometheus クエリですrate(http\_requests\_total[\$\_\_interval])。

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が多いと、間隔が大きくなり、より粗粒度の集計になります。ユーザーがズームインすると、間隔が短くなり、よりきめ細かな集計になります。

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、ダッシュボードの右上隅にあるダッシュボードのタイムピッカーで選択されているものとは異なっています。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を提供します。この関数は相対時間範囲でのみ機能し、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を上書きします。秒単位の数値として指定されます。

## 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して、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、パネルから送信されたクエリリクエストとレスポンスが表示されます。

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、「」を参照してください[パネルの検査](#)。

## クエリエディタのリスト

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。

## パネル間でのクエリ結果の共有

Amazon Managed Grafana では、ダッシュボード内の他のパネルに対して、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。

ダッシュボードデータソースでは、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。

この戦略では、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。

データソースクエリを別のパネルと共有するには

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。共有するクエリを定義します。現時点で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。Grafana 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 Dashboard 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、作成した最初の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。

クエリが定義されているパネルに移動するには、そのクエリを選択します。

## 記録されたクエリ

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください。[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

記録されたクエリを使用して、設定された間隔でデータポイントのスナップショットを取得することで、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、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、クエリを実行してリクエストの応答時間の履歴ServiceNowを表示したいが、現在の point-in-time メトリクスのみを返すことができます。

### 記録されたクエリの仕組み

記録されたクエリは、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 種類のクエリを記録できます。

- 単一行と列 - 単一の行と列を返すクエリ。
- 行数 - カウントする意味のある行を返すクエリ。
- expression - 任意の式。式の作成と使用の詳細については、「[式](#)」を参照してください。

記録されたクエリが作成または有効化されると、すぐにスナップショットが作成され、設定された間隔でスナップショットの作成が続行されます。記録されたクエリは、スナップショットが無効、削除されている場合、または Grafana が実行されていない場合にスナップショットの作成を停止します。データポイントは、記録されたクエリを実行し、各結果をリモート書き込み対応 Prometheus インスタンスに転送することによってバックエンドに収集されます。

#### Note

記録されたクエリを使用する前に、Prometheus データソースを設定し、リモート書き込みターゲットに関連付ける必要があります。

### 記録されたクエリを作成する

記録されたクエリを作成するには、次のステップを実行します。

1. 編集パネルのダッシュボードに記録したいクエリを検索または作成します。クエリは 1 つの行と列のみを返す必要があります。それ以上返された場合でも、count オプションを使用して返さ

れた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。式を使用して、時系列クエリからデータを集計できます。式の作成と使用の詳細については、[「式」](#)を参照してください。

- クエリエディタにあるレコードクエリメニューを選択します。
- 記録されたクエリ情報を入力します。特に指定がない限り、すべてのフィールドは必須です。
  - 名前 - 記録されたクエリの名前。
  - 説明 - (オプション) 記録されたクエリを、記録されたクエリリストに表示するとおりに記述します。
  - 間隔 - スナップショットが作成される間隔。間隔は、記録されたクエリの作成時に開始され、記録されたクエリを一時停止または削除すると停止します。記録されたクエリの一時停止と削除の詳細については、[「記録されたクエリの管理」](#)を参照してください。
  - 範囲 - クエリの相対時間範囲。30m の範囲と 1 時間の間隔を選択すると、クエリは過去 30 分ごとにスナップショットを作成します。
  - クエリ結果のカウント - クエリから返され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 行を返す必要があります。
- 記録されたクエリをテストするには、記録されたクエリのテストボタンを選択します。
- 「クエリの記録を開始する」を選択します。

## 記録されたクエリの追加

既存の記録されたクエリをダッシュボードのパネルに追加できます。

追加した記録されたクエリごとに、Prometheus クエリが作成されます:

```
generated_recorded_query_name{id="generated_id", name="recorded query name"}
```

Prometheus から作成されたクエリは、記録されたクエリで収集されたすべてのスナップショットを返します。

- 記録された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。
- + 記録されたクエリメニューを選択します。
- 記録されたクエリをデータソースでフィルタリングする場合は、データソースでフィルタリングするドロップダウンメニューからデータソースを選択します。
- 記録されたクエリのメニューを追加 を選択して、パネルに追加します。

記録された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。記録されたクエリは、記録された Prometheus クエリ名と一致する名前ラベルを持つ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。

記録されたクエリを追加した後、Prometheus `Mixed` データソースではなくデータソースを持つクエリが表示される場合は、記録されたクエリに対して Prometheus リモート書き込みターゲットが設定されていない可能性があります。[リモート書き込みポイントを設定するには、「リモート書き込みターゲット」](#)を参照してください。

## 記録されたクエリの使用

記録された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、記録済みクエリタブから `設定` で管理できます。

## 記録されたクエリの管理

記録されたクエリは、設定の「記録されたクエリ」タブで一時停止または削除できます。記録された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。記録された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。

## リモート書き込みターゲット

リモート書き込みターゲットは、記録された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になっている 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。

リモート書き込みターゲットを編集するには、設定の「記録されたクエリ」タブのコンソールメニューで「リモート書き込みターゲットの編集」を選択します。リモート書き込みが有効になっている Prometheus データソースを選択し、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

変換は、結果セットが視覚化に渡される前に処理します。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、別々の時系列の結合、クエリ間での数学などを行うことができます。大きなダッシュボードや重いクエリがある場合、あるパネルから別のパネルにクエリ結果を再利用できれば、パフォーマンスが大幅に向上する可能性があります。

#### Note

変換によって、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 はビジュアライゼーションに提案を表示します。テーブルの視覚化に切り替える提案を選択します。これは、多くの場合、変換がデータに対して何をしているのかをよりよく理解するのに役立ちます。

Amazon Managed Grafana は、画面に表示される順序で変換を適用します。すべての変換は、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を作成します。

この順序は、結果の外観に大きな違いをもたらす可能性があります。例えば、変換を減らすを使用して 1 つの列のすべての結果を 1 つの値に要約する場合、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、以下のすべてが満たされている必要があります。

- クエリを入力し、データソースからデータを返しました。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、クエリをサポートする視覚化を適用しました。
  - バーゲージ
  - Gauge
  - グラフ
  - ヒートマップ
  - ログ
  - Stat
  - 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 タブの横にある 変換 タブから使用できます。

変換を適用するには

1. 変換を追加するパネルで、パネルタイトルを選択し、編集 を選択します。
2. [Transform] (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、変換オプションを設定できます。

4. 別の変換を適用するには、「変換を追加」を選択します。次の変換は、前の変換によって返された結果セットに作用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](#)。

変換を削除するには、ごみ箱アイコンを選択します。

## 変換タイプとオプション

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](#)
- [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](#)
- [フィールドを整理する](#)
- [フィールドで結合 \(外部結合\)](#)
- [計算からフィールドを追加する](#)
- [フィールドへのラベル付け](#)
- [グループ化の基準](#)
- [グループ化の基準](#)
- [シリーズから行へ](#)
- [値でデータをフィルタリングする](#)

## • [デバッグ変換](#)

### 削減

結果を 1 つの値に簡素化する場合は、Reduce 変換を適用します。減らすと、基本的に時間コンポーネントが削除されます。テーブルとして視覚化すると、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。選択すると、計算の選択肢のリストが表示されます。使用可能な計算については、「」を参照してください[計算リスト](#)。

少なくとも 1 つの計算を選択すると、Amazon Managed Grafana は選択した計算を使用して 1 つの値を表示します。複数の計算を選択すると、複数の値が表示されます。

### 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。データは、変換を適用する前に 2 つの別々のテーブルとして視覚化されます。

#### クエリ A

時間	ジョブ	稼働時間
2020-07-07 11:34:20	node	25260122
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	node	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	稼働時間
2020-07-07 11:34:20	node	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 は識別子フィールドを表示し、その後にクエリによって返されるフィールドが表示されます。

フィルターは、次の 2 つの方法のいずれかで適用できます。

- 正規表現式を入力します。
- フィールドを選択して、その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は濃い灰色のテキストで表示され、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場合は、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 は、クエリ識別文字をダークグレーテキストで表示します。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。クエリ文字が白の場合、結果が表示されます。クエリ文字が暗い場合、結果は非表示になります。

フィールドを整理する

この変換を使用して、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。

#### Note

この変換は、単一の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、フィールドで結合 (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。

Amazon Managed Grafana は、クエリによって返されるフィールドのリストを表示します。次のいずれかの変更を行うことができます。

- フィールドを一時停止してフィールドの順序を変更します。カーソルが手になり、フィールドを新しい場所にドラッグできます。
- フィールド名の横にある目の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。

### フィールドで結合 (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、フィールドから結果を計算できるようにクエリを組み合わせる場合に便利です。

### 計算からフィールドを追加する

この変換を使用して、他の 2 つのフィールドから計算され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。

- Mode – 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、選択した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 sum 行の値に基本的な数学演算 (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 が新しいフィールドを作成するときに使用する計算を選択します。フィールドを選択すると、計算の選択肢のリストが表示されます。使用可能な計算の詳細については、「」を参照してください [計算リスト](#)。
- 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、フィールドには計算に一致する名前が付けられます。
- すべてのフィールドを置き換える – (オプション) 他のすべてのフィールドを非表示にし、計算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 はラベル付きデータをすべてフィールドに自動的に変換します。

例えば、2つの時系列のクエリ結果があるとします。

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	[サーバー]	Datacenter	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。値フィールド名 Serverとして を選択した場合、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	Datacenter	サーバー A	サーバー B
2020-07-07 11:34:20	EU	1	2

### グループ化の基準

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 をチェックすると、値は逆の順序で返されます。

### グループ化の基準

この変換は、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、Reduce 変換の場合と同じです。

元のデータの例を次に示します。

時間	Server 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	Server ID	CPU 温度	サーバーステータス
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 1 つ以上のフィールドを指定します。これにより、ソートした場合と同じフィールドの値がすべて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すると、次のようにデータがグループ化されます。

時間	Server 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 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

```

の値が同じすべての行 Server ID 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したら、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、CPU temperatureこれらの各サーバーの平均を計算できます。CPU Temperature フィールドに適用された平均計算を追加して、以下を取得できます。

Server 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、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。

- フィールドではTime、Last 値を計算して、サーバーごとに最後のデータポイントがいつ受信されたかを確認できます。
- フィールドではServer Status、各サーバーの最後の状態値を知るために、Last 値を計算できます。
- フィールドではTemperature、Last 値を計算して、各サーバーの最新のモニタリング温度を確認することもできます。

Group By 変換では、次の結果が生成されます。

Server ID	CPU 温度 (平均)	CPU 温度 (最新)	時間 (最終)	サーバーステータス (最新)
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、時系列からいくつかのキー情報を抽出し、便利な方法で表示できます。

## シリーズから行へ

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の3つの列が含まれますValue。Metric列が追加され、メトリクスのソースとなるクエリを確認できます。ソースクエリLabelでを定義して、この値をカスタマイズします。

以下の例では、2つのクエリが時系列データを返します。変換が適用される前に、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

時間	メトリクス	値
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

### 値でデータをフィルタリングする

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、クエリ結果から一部のデータポイントを削除できます。定義した 1 つ以上の条件に一致するデータを含めるか除外するかを選択できます。条件は、選択したフィールドに適用されます。

この変換は、データソースが値で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は、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。

- 正規表現 - 正規表現式を一致させます。
- Is Null - 値が null の場合に一致。
- Is Not Null - 値が null でない場合は一致します。
- Equal - 値が指定された値と等しい場合に一致します。
- Different - 値が指定された値と異なる場合に一致します。

数値フィールドで使用可能な条件は次のとおりです。

- Greater - 値が指定された値より大きい場合に一致します。
- 下限 — 値が指定された値よりも小さい場合に一致します。
- Greater or equal - 値が指定された値以上の場合に一致します。
- lower or equal - 値が指定された値以下の場合に一致します。
- 範囲 — 指定された最小値と最大値の間の範囲に一致します。最小と最大が範囲に含まれます。

フィルターには複数の条件を追加できます。複数の条件がある場合は、追加した条件のすべてまたはいずれかに一致する行にインクルードアクションまたは除外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、または設定が不完全である条件は無視されます。

## デバッグ変換

変換の入力セットと出力結果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 は、変換行の下に変換デバッグビューを表示します。

## 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

このセクションでは、Amazon Managed Grafana のフィールドオプションとフィールドオーバーライドとは何か、およびそれらを使用する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両方のクエリ結果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、視覚化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、Amazon Managed Grafana によるデータの表示方法を変更します。

## フィールドオプション

標準とカスタムの両方のフィールドオプションは、パネルエディタのフィールドタブにあります。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、数値を持つすべてのフィールドがパーセンテージで表示されます。でフィールドオプションを適用する方法について説明します [すべてのフィールドを設定する](#)。

## フィールドオーバーライド

フィールドオーバーライドは、パネルエディタのオーバーライドタブに追加できます。フィールドタブと同じオプションを追加できますが、特定のフィールドにのみ適用されます。でオーバーライドを適用する方法について説明します [特定のフィールドを設定する](#)。

## 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タイプは、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です。唯一の違いは、変更がすべてのフィールド (フィールドタブに適用) に適用されるか、フィールドのサブセット (オーバーライドタブに適用) に適用されるかです。

- [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視覚化に適用されます。
- [テーブルフィールドオプション](#) はテーブルパネルの視覚化にのみ適用されます。

## すべてのフィールドを設定する

すべてのフィールドのデータの表示方法を変更するには、フィールドタブのオプションを変更できます。オーバーライドタブで、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ことで、すべてのフィールドに表示される小数点以下の桁数を変更できます。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「」を参照してください [テーブルフィールドオプション](#)。

## 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を変更するには

1. 編集するパネルを選択し、パネルタイトルを選択し、編集を選択します。
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化に適用されます。
  - [テーブルフィールドオプション](#)。テーブルパネルの視覚化にのみ適用されます。

4. フィールドに値を追加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、フィールド内のホワイトテキストを削除します。
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としましょう。

time	temperature
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方法でフィールドオプションを適用できます。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。

time	temperature
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

この場合、小数点以下は意味しません。10 進数を から 0 (0) auto に変更すると、次の表になります。

time	temperature
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、特定の視覚化で利用できる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ことです。

例えば、数値に一致する型を持つフィールドの10進数オプションを変更することで、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できます。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化標準フィールドオプションに適用される」および「テーブルパネルの視覚化のみに適用されるテーブル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、必要な数だけ上書きできます。

フィールドオーバーライドを追加するには

1. 編集するパネルを選択し、パネルタイトルを選択し、編集を選択します。
2. オーバーライドタブを選択します。
3. のオーバーライドを追加を選択します。
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。
  - 名前のフィールド – これを使用して、使用可能なすべてのフィールドのリストからフィールドを選択します。このセレクタを使用してルールに追加するプロパティは、この単一のフィールドにのみ適用されます。
  - 正規表現に一致する名前のフィールド – これを使用して、正規表現で上書きするフィールドを指定します。このセレクタを使用して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。
  - タイプのフィールド – 文字列、数値などのフィールドをタイプ別に選択します。このセレクタを使用して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。
5. オーバーライドプロパティを追加を選択します。
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、フィールド内のホワイトテキストを削除します。
8. 上書きプロパティを追加を選択して、このフィールドに上書きを追加し続けるか、上書きを追加を選択して、上書き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、[Save] (保存) を選択します。

フィールドオーバーライドを削除する

1. 削除するオーバーライドを含む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

結果セットは、時間、高温度、低温度、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としましょう。

time	高温度	低温度	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

のフィールドオプションを適用 [フィールドオプションの例](#) して摂氏単位を適用し、小数点以下を削除しましょう。これにより、次の表が表示されます。

time	高温度	低温度	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好に見えますが、湿度は無意味です。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のオーバーライドを適用し、単位を Misc > パーセント (0 ~ 100) に変更します。これにより、より理にかなったテーブルが作成されます。

time	高温度	低温度	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一覧表示されます。

ほとんどの組み込み Grafana ワークスペースパネルに標準フィールド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、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](#)

## 10 進数

このオプションは、値をレンダリングするときに含める小数の数を設定します。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使用するには、空のままにします。

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

このオプションは、値または視覚化がリンクする 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)

このオプションは、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で変数を使用できます。テンプレート変数とテンプレート変数の詳細については、「」を参照してください [テンプレートと変数](#)。

複数の統計、フィールド、またはシリーズが表示される場合、このフィールドは各統計のタイトルを制御します。などの式を使用して `${__field.name}`、タイトルにシリーズ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"}` のラベルが与えられる

式の構文	例	へのレンダリング	説明
<code>\${__field.name}</code>	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、 <code>{}</code> 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、ラベル部分について、Amazon Managed Grafana は中括弧なしでラベルの値を表示します。
<code>\${__field.name}</code>	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)。
<code>\${__field.label}</code>	構文と同じ	Loc="PBI", Sensor="3"	名前のないラベルを表示します。
<code>\${__field.label} {__field.label}</code>	<code>\${__field.label} {__field.label}</code>	PBI	指定されたラベルキーの値を表示します。

式の構文	例	へのレンダリング	説明
<code>\${__fi</code> <code>.label</code> <code>_value</code>	Synta と 同	PBI, 3 じ	ラベルの値をカンマで区切って表示します (ラベルキーなし)。

特定のフィールドの式をレンダリングした後に値が空の文字列である場合は、デフォルトの表示方法が使用されます。

### 最大

このオプションは、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。すべてのシリーズとフィールドに基づく自動計算の場合は、この設定を空白のままにします。

### 最小

このオプションは、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。すべてのシリーズとフィールドに基づく自動計算の場合は、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 が表示する内容を入力します。

### 単位

このオプションは、フィールドが使用す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。

### カスタムユニット

単位ドロップダウンリストを使用して、カスタム単位、カスタムプレフィックスまたはサフィックス、および日付/時刻形式を指定することもできます。

カスタム単位を選択するには、単位を入力し、ドロップダウンリストで最後の Custom: xxx オプションを選択します。

- `suffix:<suffix>` 値より後になるカスタム単位の。

- `time:<format>` カスタム日付/時刻形式の場合は、など`time:YYYY-MM-DD`。形式の構文とオプションについては、「[の表示](#)」を参照してください。
- `si:<base scale><unit characters>` カスタム SI 単位の場合は。例: `si: mF`。このオプションは、単位とソースデータスケールの両方を指定できるため、少し進んでいます。例えば、ソースデータがミリ秒 (数千) 単位として表される場合、単位の前にその SI スケール文字を付けます。
- `count:<unit>` カスタムカウント単位の。
- `currency:<unit>` カスタム通貨単位の。

単位ピッカーにネイティブ絵文字を貼り付けて、カスタム単位として選択することもできます。

## 文字列単位

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、ユニットで `string` ユニット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 がシリーズまたはフィールドを色分けする方法を定義します。動作が異なるモードが複数あり、そのユーティリティは、現在選択されている視覚化に大きく依存します。

連続色モードは、最小と最大に対する値の割合を使用して色を補間します。

- 単一色 – カラーピッカーを使用して設定された特定の色。これは、オーバーライドルールから最も役立ちます。
- しきい値から – 一致するしきい値から派生した色。これは、ゲージ、統計、およびテーブルの視覚化に役立ちます。

## シリーズ別の色

Amazon Managed Grafana には、シリーズごとに色を定義するカラースキームが含まれています。これは、グラフや円グラフなどに役立ちます。

## 値別の色

Amazon Managed Grafana には、連続 (グラデーション) のカラースキームも含まれています。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化に役立ちます。



## しきい値

しきい値を使用して、値に基づいてフィールドの色を変更できます。詳細と手順については、「[しきい値](#)」を参照してください。

## 値マッピング

このオプションを使用して、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。複数の値マッピングを追加できます。

- マッピングタイプ — オプションを選択します。
  - 値 – 値を入力します。フィールド値が 値以上の場合、テキストが表示されます。
  - 開始と終了 – 範囲を入力します。フィールド値が範囲内の値以上である場合は、テキストが表示されます。
- テキスト – 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。このフィールドは変数を受け入れます。

## パネルエディタ

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

このトピックでは、Amazon Managed Grafana パネルエディタの部分について説明し、詳細情報が確認できる へのリンクが含まれています。

### パネルエディタを開く

パネルエディタにアクセスするには、パネル編集画面、編集モード、パネル編集モードとも呼ばれます。

- 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の詳細については、「」を参照してください[パネルの追加](#)。

- 既存のパネルのタイトルを選択し、編集を選択します。編集モードでパネルが開きます。
- 既存のパネル上の任意の場所を選択し、キーボードの e を押します。編集モードでパネルが開きます。

## パネルエディタセクションのサイズ変更

ドラッグして、パネルエディタのセクションのサイズを変更します。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。

## パネルエディタの一部

このセクションでは、パネルエディタ画面の各部分について説明し、各部分に関連付けられたフィールド、オプション、またはタスクに関する情報を示します。

### [Header] (ヘッダー)

ヘッダーセクションには、パネルがあるダッシュボードの名前と、いくつかのダッシュボードコマンドが一覧表示されます。Go 戻る矢印を選択してダッシュボードに戻ることができます。

ヘッダーの右側には、次のオプションがあります。

- ダッシュボード設定 (歯車) 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。
- 破棄 ダッシュボードを最後に保存してからパネルに加えたすべての変更を破棄するには、 を選択します。
- 保存 — パネルエディタで行ったすべての変更を含むダッシュボードを保存する場合に選択します。
- 適用 — 行った変更を適用することを選択し、パネルエディタを閉じてダッシュボードに戻ります。また、ダッシュボードを保存して、適用された変更を保持します。

## 視覚化プレビュー

視覚化プレビューセクションには、表示オプション、時間範囲コントロール、視覚化プレビュー、パネルタイトル、軸、凡例 (該当する場合) が含まれています。

- 塗りつぶし — 視覚化プレビューは、プレビューパートで使用可能な領域を塗りつぶします。サイドペインの幅または下部ペインの高さを変更すると、使用可能なスペースを埋めるように視覚化が調整されます。
- Fit – 視覚化プレビューは使用可能な領域を埋めますが、パネルのアスペクト比は保持されます。

- Exact – 視覚化プレビューのサイズは、ダッシュボードのサイズとまったく同じです。十分なスペースがない場合、視覚化はスケールダウンし、アスペクト比を維持します。
- 時間範囲の制御 – 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](#)。

## データセクション (最小ペイン)

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 – アラートルールを書き込みます。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)

このセクションには、データの視覚化方法のほぼすべての側面を制御するタブが含まれています。視覚化ごとにすべてのタブを使用できるわけではありません。

これらのタブの機能については、以下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](#)。視覚化を他のリソースに接続するのに役立ちます。

## ライブラリパネル

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

ライブラリパネルを使用すると、ユーザーは再利用可能なパネルを作成できます。再利用可能なパネルでは、ライブラリパネルの1つのインスタンスに加えられた変更が、パネルが使用されている他のすべてのインスタンスに影響を与えるすべてのダッシュボードに反映されます。これらのパネルは、Dashboards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の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、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。

## ライブラリパネルを作成するには

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、**編集**を選択します。
3. パネルの表示オプションサイドペインで、**下矢印オプション**を選択して視覚化に変更を加えます。
4. **ライブラリパネル**を選択し、**新しいライブラリパネルの作成**を選択します。
5. ライブラリパネルの名前を入力し、保存先のフォルダを選択します。
6. ライブラリの作成パネルを選択し、ダッシュボードを保存します。

任意のパネルに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、ライブラリパネルが表示されるダッシュボードを使用してライブラリパネルを変更できます。ライブラリパネルの変更を保存すると、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。

## ライブラリパネルを追加する

### ライブラリパネルをダッシュボードに追加するには

1. 左のメニューの **+** オプションで一時停止し、**の作成**を選択します。
2. **パネルライブラリ** から **パネルを追加**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、目的の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

ダッシュボードにライブラリパネルがあり、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変更する場合は、ライブラリパネルのリンクを解除できます。

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には

1. 左側のメニューのダッシュボードで一時停止し、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。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. リンクを解除して更新するパネルを選択します。
4. パネルのタイトルを選択し、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、どのダッシュボードでも使用されていないことを確認します。

ライブラリパネルを削除するには

1. 左側のメニューのダッシュボードで一時停止し、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。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。

## パネルエディタの一部

このセクションでは、パネルエディタ画面の各部分について説明し、各部分に関連付けられたフィールド、オプション、またはタスクに関する情報を示します。

### [Header] (ヘッダー)

ヘッダーセクションには、パネルがあるダッシュボードの名前と、いくつかのダッシュボードコマンドが一覧表示されます。Go back 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、次のオプションがあります。

- **ダッシュボード設定 (歯車) 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。
- **破棄** ダッシュボードを最後に保存してからパネルに加えたすべての変更を破棄するには、**破棄** を選択します。
- **保存** – パネルエディタで行ったすべての変更を含むダッシュボードを保存する場合に選択します。
- **適用** – 行った変更を適用することを選択し、パネルエディタを閉じてダッシュボードに戻ります。また、ダッシュボードを保存して、適用された変更を保持します。

## 視覚化プレビュー

視覚化プレビューセクションには、表示オプション、時間範囲コントロール、視覚化プレビュー、パネルタイトル、軸、凡例 (該当する場合) が含まれています。

- **塗りつぶし** – 視覚化プレビューは、プレビューパートの使用可能な領域を塗りつぶします。サイドペインの幅または下部ペインの高さを変更すると、使用可能なスペースを埋めるように視覚化が調整されます。
- **Fit** – 視覚化プレビューは使用可能な領域を埋めませんが、パネルのアスペクト比は保持されます。
- **Exact** – 視覚化プレビューのサイズは、ダッシュボードのサイズとまったく同じです。十分なスペースがない場合、視覚化はスケールダウンし、アスペクト比を維持します。
- **時間範囲の制御** – 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 (最小ペイン)

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** – アラートルールを書き込みます。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)

このセクションには、データの視覚化方法のほぼすべての側面を制御するタブが含まれています。視覚化ごとにすべてのタブを使用できるわけではありません。

これらのタブの機能については、以下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](#)。視覚化を他のリソースに接続するのに役立ちます。

## 視覚化

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Amazon Managed Grafana は、さまざまなユースケースに合わせてさまざまな視覚化を提供します。以下のセクションでは、Amazon Managed Grafana で使用できるビジュアライゼーションとその表示設定を一覧表示します。

### 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](#)
- [クロックパネル](#)
- [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](#)
- [プロットパネル](#)
- [サンキーパネル](#)
- [散布図](#)
- [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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

アラートリストパネルにダッシュボードアラートが表示されます。現在の状態または最近の状態の変化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、「」を参照してください[Grafana アラート](#)。

これらの設定を使用して、視覚化を絞り込みます。



## オプション

- 表示 — パネルに現在のアラート状態を表示するか、最近のアラート状態の変更を表示するかを選択します。
- 最大アイテム数 — リストするアラートの最大数を設定します。
- ソート順 — 表示されるアラートの順序付け方法を選択します。
  - アルファベット順 (asc) – アルファベット順
  - アルファベット順 (desc) — アルファベット順の逆順
- 重要度 — 次の値に従って重要度が異なり、1 が最上位になります。
  - アラート: 1
  - no\_data: 2
  - 保留中: 3
  - ok: 4
  - 一時停止: 5
- このダッシュボードからのアラート – アラートリストがあるダッシュボードからのアラートのみを表示します。

## フィルター

次のオプションを使用して、選択したクエリ、フォルダ、またはタグと一致するようにアラートをフィルタリングします。

- アラート名 – アラート名クエリを入力します。
- ダッシュボードタイトル — ダッシュボード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状態を選択します。

- 没頭
- 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

このパネルの視覚化により、カテゴリ別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 軸のカテゴリ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。データ形式の例を次に示します。

ブラウザ	マーケットシェア
Chrome	50
Internet Explorer	17.5

複数の数値フィールドがある場合、パネルにはグループ化された棒が表示されます。

### 時系列または複数の結果セットの視覚化

時系列またはテーブルが複数ある場合は、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、Reduce transform を追加し、Calculations で Max と Last as options を指定します。

## 棒グラフオプション

視覚化を絞り込むには、以下のオプションを使用します。

### [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。
- Vertical – Y 軸をカテゴリ軸にします。

### 値を表示する

バーの左上に値を表示す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。
- Always – 常に値を表示します。
- Never – 値を表示しないでください。

グループ幅はグループの幅を制御します。0=最小、1=最大幅。

棒幅は棒の幅を制御します。0=最小、1=最大幅。

線幅は、棒の線幅を制御します。

塗りつぶし不透明度は、バー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は、グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – 勾配フィルなし、これがデフォルト設定です。
- 不透明度 – 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は Y 軸の値とともに増加しています。
- Hue – グラデーション色は、線の色相に基づいて生成されます。

ツールヒントモード – 視覚化の上にカーソルを置くと、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- **Single** – ホバーツールヒントには、視覚化にカーソルを合わせているシリーズが 1 つだけ表示されます。
- **すべて** – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- **非表示** – 視覚化を操作するときツールヒントを表示しないでください。

#### Note

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

**凡例モード** – これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- **List** – 凡例をリストとして表示します。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。

- **下** – グラフの下。
- **右** – 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。複数の を持つことができます。

**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、視覚化には影響しません。

- **配置** — Y 軸の配置を設定します。
- **自動** – Grafana は Y 軸をシリーズに自動的に割り当てます。異なる単位を持つシリーズが 2 つ以上ある場合、Grafana は左軸を最初の単位に、右軸を次の単位に割り当てます。
- **左** – 左側にすべての Y 軸を表示します。
- **右** – 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 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、軸が視覚的に互い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量を比較しやすくなります。

- 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、ソフト最小値とソフト最大値の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マックスの設定により、データがほぼフラットなときにリップが山に変わるのが防ぐことができます。また、標準最小値と最大フィールドオプションから導出されるハードマムまたは最大値により、定義されたポイントを超えてスパイクをクリップすることで、断続的なスパイクが有用な詳細をフラット化することを防ぐことができます。

標準の最小/最大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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

バーゲージは、すべてのフィールドを 1 つの値に減らすことで、データを簡素化します。Amazon Managed Grafana による削減の計算方法を選択します。

このパネルには、クエリが返すシリーズ、行、または列の数に応じて、1 つ以上の棒グラフを表示できます。

### データおよびフィールドオプション

棒ゲージの視覚化では、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

以下のオプションを使用して、視覚化を絞り込みます。

- 表示 — Amazon Managed Grafana がデータを表示する方法を選択します。
  - 計算 — すべての行に基づいて計算された値を表示します。使用可能な計算のリストについては、「」を参照してください[計算リスト](#)。
  - 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した場合は、制限または表示する最大行数を選択することもできます。
- 値 — Amazon Managed 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リデューサー関数を選択します。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。
- 方向 — 積み上げ方向を選択します。
  - 自動 — Amazon Managed Grafana は、最適な向きを選択します。
  - 水平 — バーは左から右に水平方向に伸縮します。
  - Vertical — バーは上から下に垂直に伸縮します。
- 表示モード — 表示モードを選択します。
  - グラデーション — しきい値レベルを選択してグラデーションを定義します。
  - リトロ LCD — ゲージを小さなセルに分割して、照明の有無にかかわらず表示します。
  - 基本 — 一致するしきい値に基づいて単一の色を使用します。
- 塗りつぶされていない領域を表示する — バーの塗りつぶされていない領域を濃い灰色でレンダリングする場合は、このオプションを選択します。このオプションは、Retro LCD ディスプレイモードでは使用できません。

## クロックパネル

**⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

クロックパネルには、現在の時刻またはカウントダウンが表示されます。毎秒更新されます。

- モード – デフォルトは時間です。カウントダウン を選択した場合は、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 – 時間を表示するオプションは 12 時間形式と 24 時間形式です。
- Timezone – タイムゾーンは、Monce Timezone ライブラリによって提供されます。デフォルトはコンピュータのタイムゾーンです。
- カウントダウンの期限 – モードをカウントダウン に設定している場合は、カウントダウンす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を表示するテキストを指定します。
- 日付/時刻フォーマットオプション – フォントサイズ、重み、日付/時刻のフォーマットをカスタマイズします。カウントダウンを表示していて、秒数がティックダウンしない場合は、24 時間制 hh:mm の場合は時刻形式を、12 時間制 h:mm A の場合は時刻形式を変更します。オプションの完全なリストについては、「[の表示](#)」を参照してください。
- Bg Color – クロック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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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な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するように設定できます。

ダッシュボードがロードされるたびに、このパネルはダッシュボードリストをクエリし、常に最も多く up-to-date の結果を提供します。

## オプション

以下のオプションを使用して、視覚化を絞り込みます。

- Starred – スタード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。このオプションでは、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。
- 見出しを表示 — 選択したリスト選択 (星型、最近表示、検索) を見出しとして表示します。
- 最大項目数 – セクションごとにリスト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、星形の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 個が表示されます。

## 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。

- 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は区別されず、部分的な値が受け入れられます。
- 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。既存のタグは入力時に表示されず、タグでは大文字と小文字が区別されることに注意してください。

### 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に設計されています。



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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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パネルです。

## データおよびフィールドオプション

ゲージの視覚化では、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

視覚化を絞り込むには、次のオプションを使用します。

- 表示 — Amazon Managed Grafana がデータを表示する方法を選択します。
  - 計算 — すべての行に基づいて計算された値を表示します。使用可能な計算のリストについては、「」を参照してください[計算リスト](#)。
  - 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した場合は、制限または表示する最大行数を選択することもできます。
- 方向 — 積み上げ方向を選択します。
  - 自動 – Amazon Managed Grafana は、最適な向きであると思われ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縮します。
  - Vertical – バーは上から下に垂直に伸縮します。
- しきい値ラベルを表示する — しきい値を表示するかどうかを選択します。
- しきい値マーカーを表示する – 内部ゲージ値バンドの外部にしきい値バンド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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Geomap パネルの視覚化を使用すると、地理空間データを使用してワールドマップを表示およびカスタマイズできます。データの位置ベースの重要な特性に簡単に焦点を合わせるために、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は、ベースマップの上に地理空間データを視覚化する方法を決定します。

### クエリ

Geomap パネルには、地理的データのソースが必要です。このデータはデータベースクエリから取得され、データには 4 つのマッピングオプションがあります。

- 自動は位置データを自動的に検索します。このオプションは、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に使用します。
  - geohash: 「geohash」
  - 緯度: 「緯度」、「緯度」
  - longitude: 「longitude」、「lng」、「lon」
  - ルックアップ: 「ルックアップ」
- 座標は、クエリが座標データを保持す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に求められます。
- Geohash は、クエリが geohash データを保持するように指定します。データベースクエリから geohash の文字列データフィールドを選択するように求められます。
- Lookup は、クエリが値にマッピングする必要がある場所名データを保持するように指定します。データベースクエリと からルックアップフィールドを選択するように求められます

gazetteer。 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マーカー形状として表示できます。

- マーカーカラーはマーカーの色を設定します。デフォルトの固定サイズでは、すべてのポイントが 1 色に保たれます。データポイントの値としきい値セクションで設定されたしきい値に応じて、複数の色を持つ代替オプションがあります。
- マーカーサイズはマーカーのサイズを設定します。デフォルトは Fixed size で、データポイントに関係なくすべてのマーカーサイズが同じになります。ただし、円を対応するデータポイントにスケールリングするオプションもあります。 Minおよび Max マーカーのサイズは、マーカーレイヤーがこれらの範囲内でスケールリングできるように設定する必要があります。
- Marker Shape を使用すると、データポイントを異なる方法で柔軟に視覚化できます。
  - [Circle] (円形)
  - [Square] (平方)
  - 三角形
  - クロス
  - X
- 塗りつぶし不透明度は、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。

Markers と同様に、視覚化するデータポイントとその方法を決定するさまざまなオプションが表示されます。

- 重み値は、ヒートマップクラスターの強度を設定します。固定値は、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 の範囲である必要があります。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リングする代替オプションがあります。

- 半径は、ヒートマップクラスターのサイズを設定します。
- Blur 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

グラフパネルは、線、ドットのパス、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のに十分な汎用性を備えています。

### データおよびフィールドオプション

グラフの視覚化を使用する場合は、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

視覚化を絞り込むには、次の設定を使用します。

- 棒グラフ - 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。
- 線幅 - シリーズの線の幅を指定します。デフォルトは 1 です。
- Staircase - 隣接するポイントを踏み台として描画します。
- エリアフィル - シリーズのカラーフィルの量を指定します。デフォルトは 1 で、0 はなしです。
- フィル勾配 — エリアフィルの勾配の度合いを指定します。デフォルトは 0 で、勾配はありません。10 は急勾配です。

- 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。
- アラートのしきい値 – パネルにアラートのしきい値とリージョンを表示します。

## スタッキングと null 値

- スタック – 各シリーズが別のシリーズの上に積み上げられます。
- Percent – 各シリーズがすべてのシリーズの合計に対するパーセンテージで描画されます。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。以下の注意事項を参照してください。
  - connected – シリーズ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、その行はギャップをスキップし、次の NULL 以外の値に接続します。
  - null シリーズにギャップがある場合、つまり null 値の場合、グラフの線は折られ、ギャップが表示されます。これはデフォルトの設定です。
  - null as zero – 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て、負荷が 100% に達すると、サーバーはロックアップされ、エージェント送信統計は負荷統計を収集できません。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 がギャップを表示し、問題があることを示します。これを接続されたに設定すると、このシグナルを見逃しやすくなります。

## ホバーツールヒント

これらの設定を使用して、グラフの視覚化を一時停止し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 – ホバーツールヒントに表示されるシリーズの数を決定します。
  - すべてのシリーズ – ホバーツールヒントには、グラフ内のすべてのシリーズが表示されます。ツールヒントのシリーズリストで、Grafana ワークスペースは一時停止しているシリーズを太字で強調表示します。

- 単一 – ホバーツールヒントには、グラフで一時停止しているシリーズが 1 つだけ表示されま  
す。
- ソート順 – すべてのシリーズモードを選択した場合、ホバーツールヒントのシリーズの順序を  
ソートします。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。一般に、ユーザーは最大値または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、目的のデータを見つけやすくなります。
- なし – ツールヒントのシリーズの順序は、クエリのソート順序によって決まります。例えば、  
シリーズ名でシリーズをアルファベット順にソートできます。
- 増加 – ホバーツールチップのシリーズが値と増加順にソートされ、リストの上部に最小値が表  
示されます。
- Decreasing – ホバーツールチップのシリーズが値と降順でソートされ、リストの上部に最大値  
が付きます。

## 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、グラフパネル内のシリーズを他のシリーズとは異なるレンダ  
リングにすることができます。表示オプションは、シリーズごと、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 つのシリーズで線幅を太くして目立たせたり、右の Y 軸に移動  
したりできます。

複数のシリーズオーバーライドを追加できます。

シリーズオーバーライドを追加するには

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 つのシリーズに一致しま  
す。

3. `+` を選択し、シリーズに適用するスタイルを選択します。各エントリに複数のスタイルを追加  
できます。

- 棒グラフ – シリーズを棒グラフとして表示します。
- 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 – 面塗りつぶしを含む折れ線グラフを表示します。

- フィル勾配 — エリアフィル勾配量を指定します。
- 線幅 — 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 値を無視するか、ゼロに置き換えます。これは、データのギャップを無視する場合に重要です。
- 以下を に塗りつぶす — 2 つのシリーズ間の領域を塗りつぶします。
- 踏み台線 - シリーズを踏み台線として表示します。
- Dashes — ダッシュを含む行を表示します。
- 隠しシリーズ — シリーズを非表示にします。
- ダッシュの長さ — 線のダッシュの長さを設定します。
- ダッシュスペース — 行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 — シリーズを個別のポイントとして表示します。
- ポイント半径 — ポイントレンダリングの半径を設定します。
- 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 — Y 軸シリーズを設定します。
- Z-index — シリーズ z-index (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、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 — 値を負に変換して Y 軸の下にレンダリングします。
- 凡例 — 凡例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、視覚化での軸の表示を制御します。

### 左 Y/右 Y

オプションは両方の y 軸で同じです。

- 表示 — 軸を表示または非表示にすることを選択します。
- 単位 — y 値のディスプレイ単位を選択します。
- スケール — y 値に使用するスケールを選択します。線形、対数 です。デフォルトは線形 です。
- Y-Min — y の最小値。デフォルトは自動 です。

- Y-Max – Y の最大値。デフォルトは自動 です。
- 10 進数 – y 値に対して表示される小数点数を定義します。デフォルトは自動 です。
- Label – Y 軸ラベルを指定します。デフォルトは「」です。

## Y 軸

- 整列 — 左右の y 軸を値で整列させます。デフォルトはチェックなし/false です。
- レベル – Y=0 から始めて、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、Align が選択されているときに使用できます。

## X 軸

- 表示 — 軸を表示または非表示にすることを選択します。
- 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。1 つの に 3 つのパネルがあるようなものです。メインモードは、X 軸に時間がある時系列モードです。他の 2 つのモードは、時間モードとヒストグラムモードではなく、X 軸にシリーズを持つ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 (時間単位や分単位など) でグループ化されます。
  - シリーズ – データは時間ではなく、シリーズごとに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、数値をバケットまたはビンと呼ばれる範囲にグループ化する棒グラフの一種です。タラバーは、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、「」を参照してください[ヒストグラムとヒートマップの概要](#)。

- バケット – 値をグループ化するバケットの数を設定します。空のままにすると、Amazon Managed Grafana は適切な数のバケットを計算しようとします。
- X-Min – この最小制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-Max – 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、視覚化での凡例の表示方法を絞り込みます。



## オプション

- 表示 — クリアすると凡例が非表示になります。デフォルトが選択されています (true)。
- テーブル - 選択すると、テーブルに凡例が表示されます。デフォルトがチェックされます (true)。
- 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、右に を選択した場合に使用できます。

## [値]

凡例名と一緒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。
- Max - メトリクスクエリから返される最大値。
- 平均 — メトリクスクエリから返される平均値。
- Current - メトリクスクエリから返された最後の値。
- 合計 — メトリクスクエリから返されたすべての値の合計。
- 10 進数 - 凡例値とグラフホバーツールヒントに表示される小数点数。

Amazon Managed Grafana は、クライアント側で凡例値を計算します。これらの凡例は、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。上記の凡例の値はすべて、同時に正しい値にすることはできません。

例えば、平均をアグリゲータとして使用している可能性が高いリクエスト/秒のレートをプロットした場合、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。

## シリーズを非表示にする

メトリクスクエリからのシリーズのすべての値が特定の値である場合、シリーズを非表示にします。

- null のみを使用 - Value=null (デフォルトはチェックされていません )
- ゼロのみを使用 - 値 = ゼロ (デフォルトはチェックされていません )

## 時間リージョン

グラフで特定の時間帯を強調表示して、週末、営業時間、勤務時間外などを簡単に見ることができます。設定されているすべての時間リージョンは UTC 時間を参照します。

## ヒートマップ

- △** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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、「」を参照してください[ヒストグラムとヒートマップの概要](#)。

## 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調整します。

### Y 軸

- Unit – Y 軸の値の表示単位
- スケール – Y 軸の値に使用するスケール
  - 線形 – 線形スケール
  - log (ベース 2) – ベース 2 の対数スケール
  - log (ベース 10) – ベース 10 の対数スケール
  - log (ベース 32) – ベース 32 の対数スケール
  - log (ベース 1024) – ベース 1024 の対数スケール
- Y-Min – 最小 y 値 (デフォルトは自動)
- Y-Max – Y 最大値 (デフォルトは自動)
- 10 進数 – Y 軸値を でレンダリングする小数点以下の数 (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 (スケールが線形の場合にのみ表示されます)。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 – (スケールがログ (ベース 2) 以上の場合にのみ表示されます)。デフォルトでは、Amazon Managed Grafana はログベースで 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。数値または時間間隔 (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。

## バケットバインド

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バインドを表す名前を持つシリーズを返します。ただし、データソースによっては、バインドを上限または下限にすることができます。このオプションを使用して、バインドタイプを調整できます。Auto が設定されている場合、パネルのデータソースタイプに基づいてバインド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、カウント (最初の入力ボックス) またはサイズ間隔を指定して定義できます。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 に設定できます 1h。これにより、セルが X 軸で 1 h 幅になります。

## 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行います。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、すでに Y 軸バケットを表しています。時系列名 (エイリアス) は、バケットの上限または下限の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

これらの設定を使用して、視覚化を絞り込みます。

## カラー

色スペクトルは、値カウント (各バケット内) と各バケットに割り当てられた色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小数を表し、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。

カラーモードを Opacity 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。

- Mode (モード)
  - 不透明度 – セルの不透明度で表されるバケット値。不透明なセルは最大値を意味します。
    - 色 – セルのベース色。
    - スケール – バケット値を不透明度にマッピングするためのスケール。
      - 線形 – 線形スケール。バケット値は不透明度に直線的にマッピングされます。
      - sqrt – パワースケール。セルの不透明度は  $\text{value}^k$  として計算され、 $k$  は設定された指数値です。指数がより小さい場合 1、対数スケールが得られます。指数がより大きい場合 1、指数スケールが得られます。の場合 1、スケールは線形と同じになります。
    - Exponent – 指数の値。より大きい 0。
  - spectrum – セルの色で表されるバケット値。
    - スキーム – モードがスペクトルの場合は、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。最小と最大では、これらの値を上書きできます。バケットの値は z 軸、最小と最大はそれぞれ Z-Min と Z-Max 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 Min より小さい場合、最小色にマッピングされます。デフォルトは series min value です。
- Max – セルの色の計算に使用される最大値。バケット値が Max より大きい場合、最大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

- ゼロを非表示 - 値がゼロのセルは描画しないでください。
- スペース - 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。
- ラウンド - セルのラウンドネスをピクセル単位で設定します。デフォルトは 0 です。

## Tooltip

- ツールヒントを表示 - ヒートマップツールヒントを表示します。
- ヒストグラム - ツールチップに Y 軸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のバケット値の分布を表します。
- 10 進数 - バケット値をレンダリングする小数点数を設定します (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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

ヒストグラムの視覚化は、値の分布を計算し、棒グラフとして表示します。Y 軸と各バーの高さは各角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は、時系列と、1 つ以上の数値フィールドを含むテーブル結果をサポートします。

## 表示オプション

視覚化を絞り込むには、以下のオプションを使用します。

## バケットサイズ

バケットのサイズ。自動バケットサイズ設定 (全範囲の 10% まで) の場合は、これを空のままにします。

## バケットオフセット

最初のバケットがゼロから開始すべきでない場合。ゼロ以外のオフセットは、集計ウィンドウをシフトします。例えば、デフォルトの 0 オフセットを持つ 0~5、5~10、10~15 の 5 サイズのバケットは、2~7、7~12、12~17 になり、オフセットは 2 になります。この場合、オフセットは 0、5、または 10 で、実質的に何もしません。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されます。この設定に影響を与えるには、オフセット量が 0 より大きく、バケットサイズより小さい必要があります。この範囲外の値は、この範囲内の値と同じ効果を持ちます。

## シリーズを組み合わせる

これにより、すべてのシリーズとフィールドが結合ヒストグラムにマージされます。

線幅は、棒の線幅を制御します。

塗りつぶし不透明度は、バー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は、グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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、不透明度を塗りつぶす設定の影響を受けます。

- なし – グラデーションフィルなし、これがデフォルト設定です。
- 不透明度 – 勾配の透明度は、Y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。
- Hue – グラデーション色は、線の色相に基づいて生成されます。

ツールヒントモード グラフにカーソルを合わせると、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、ホバーしているシリーズのみが表示されます。
- すべて – ホバーツールヒントには、視覚化内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – ツールヒントは表示しません。

**Note**

オーバーライドを使用して、ツールヒントから個々のシリーズを非表示にします。

## 凡例オプション

凡例オプションを有効にすると、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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シングルカラーまたは Classic パレットに設定することが重要です。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。

凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- リスト — 凡例をリストとして表示します。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 – 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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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からのログ行が表示されます。通常、グラフパネルの横にある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、クエリタブに入力されたクエリの結果が表示されます。複数のクエリの結果はマージされ、時間でソートされます。データソースから返される行数が表示可能な行数よりも多い場合は、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設定を使用できます。設定されていない場合、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、視覚化を絞り込みます。

- Time – 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に関連付けられた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。これは、非共通ラベルのみを表示します。
- ラップライン – ラインラップを切り替えます。
- 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。最も古いログ行を最初に表示するには、昇順に設定します。

## ニュースパネル

**⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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 ブログの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、他のクエリを受け付けません。



## ノードグラフパネル (ベータ)

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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または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、ノードとエッジで構成されます。

- ノードは円として表示されます。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性のあるその他のものを表す場合があります。
- 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可能性があります。

### ノード

通常、ノードにはノード内に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。ノードは、ノードの周囲に別の値のセットを色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、エラーの割合を円の赤い部分で表すことができます。

追加の詳細をコンテキストメニューで表示でき、ノードを選択すると表示されます。コンテキストメニューには、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もあります。

## エッジ

エッジにカーソルを合わせると、エッジに統計が表示されることもあります。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くことができます。

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追加の詳細を表示できます。コンテキストメニューには、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もあります。

## ノードグラフのナビゲーション

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、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してズームできます。

## 円グラフパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

円グラフには、1 つ以上のクエリからの縮小されたシリーズまたは値が、円のスライスとして相互に関連するため表示されます。スライスの円弧の長さ、面積、中心角度はすべて、すべての値の合計に関連するため、スライス値に比例します。このタイプのグラフは、小さな値のセットを気楽な形式ですばやく比較したい場合に最適です。

円グラフの視覚化では、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

以下のオプションを使用して、視覚化を絞り込むことができます。

- 表示 — 表示する情報の量を選択します。Calculate は、各値をシリーズごとに 1 つの値に減らします。すべての値には、1 つのシリーズのすべての値が表示されます。
- 計算 — 計算が選択されている場合、計算を選択して各シリーズを減らします。使用可能な計算の詳細については、「」を参照してください[計算リスト](#)。
- 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。
  - Time — 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。複数の を選択できます。

- 名前 — シリーズ名またはフィールド名。
- Percent — 全体のパーセンテージ。
- 値 — 未加工の数値。

ラベルは、グラフの本文に白で表示されます。より見やすくするために、より暗いグラフの色を選択する必要がある場合があります。長い名前や数字はクリップされる場合があります。

## 凡例の配置と値

凡例を表示する場所を選択します。

- Bottom — チャートの下。
- 右 — チャートの右側。

凡例に表示する値は複数選択できます。パーセントは全体に対するパーセンテージで、値 は未加工の数値です。

## プロットパネル

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Plotly パネルは、オープンソースの javascript グラフライブラリである [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 Config 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。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 チャートに挿入する前に、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 つの引数が含まれています。

- data – データソースによって返されるデータ。
- variables – 現在のダッシュボードに [Grafana 変数](#) (ユーザー変数と、以下のいくつかのグローバル変数: `__from`、`__to`、`__interval`、および `__interval_ms`) を含むオブジェクト。

スクリプトは、`data`、`layout config` の 1 つ以上のプロパティを持つオブジェクトを返す必要があります `frames`。次に例を示します。

```
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,
  },
}
```

};

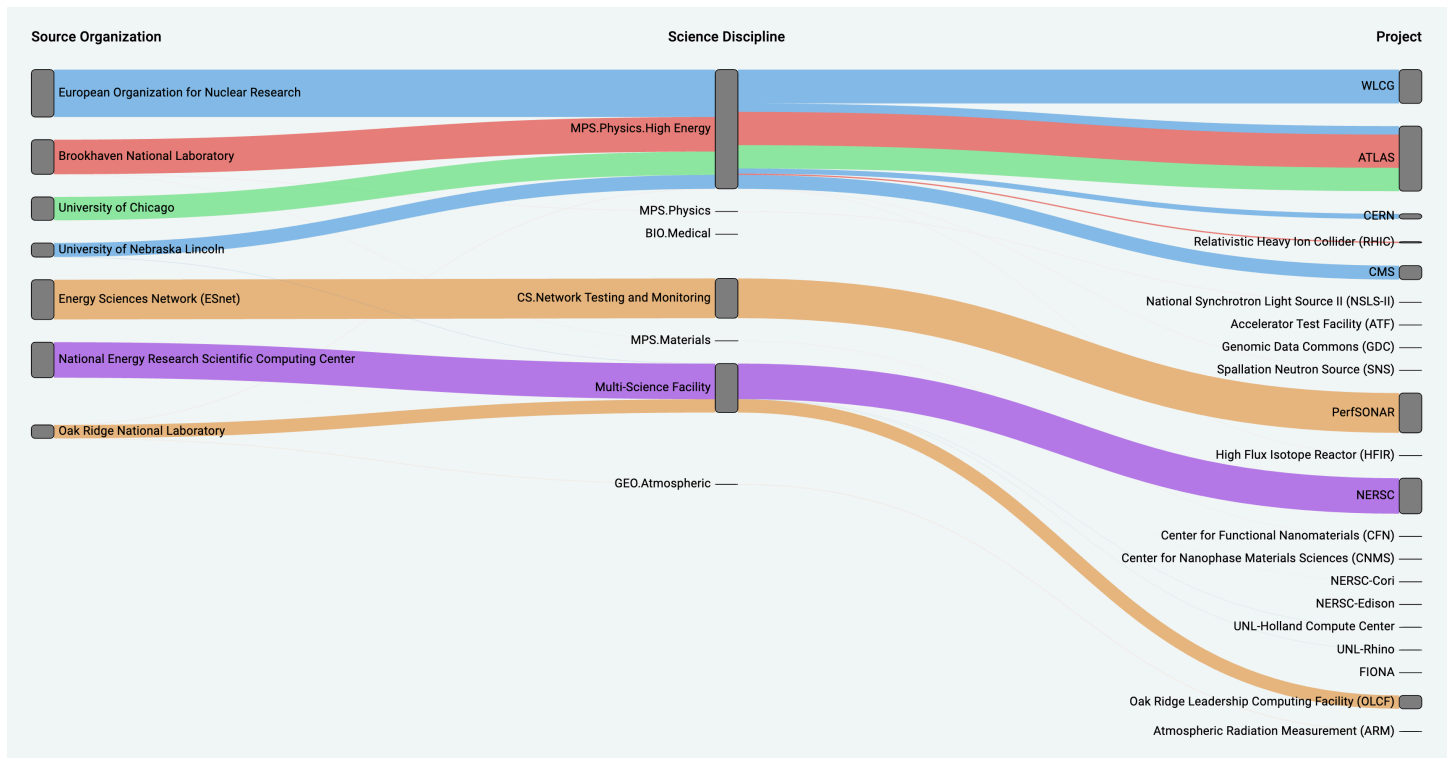
データ、レイアウト、および Config フィールドで指定されたスクリプトと JSON によって返されたオブジェクトはマージされます (ディープマージ)。

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 Config 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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

サンキーパネルにはサンキー図が表示されます。サンキー図は、フローデータを視覚化するのに適しており、フローの幅は選択したメトリクスに比例しています。次の図は、送信元と送信先の 2 つのグループを含むサンキー図を示しています。



## 仕組み

サンキーパネルには、フローの送信元と送信先という 2 列以上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。リンクの太さは、クエリ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とシングル の 2 つのオプションがあります。デフォルトでは、複数色になっています。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を使用するか、入力ボックスに数値を入力することで調整できます。この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。この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少し調整できます。この数値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裸裸反復の数です。

## 散布図

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

散布図パネルには、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 散布図が表示されます。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である必要はありません。散布図には、2 つ以上のデータ列を含むテーブル形式のデータセットが必要です。

そのうちの 1 つを X 軸に割り当てることができます。1 つ以上の を一連の Y 軸値に割り当てることができ、結果のデータを一連のドットとしてプロットできます。各シリーズでは、オプションで、いくつかの統計のベストフィットの 1 つ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、、、AgeBoysおよび の 3 つの数値列HEIGHTで呼び出されGirls、年齢別の少年と少年の平均の高さを示すデータがあると想定します。

年齢	の高度	女性の高度
5	109.7	109.5
6	115.6	115.4
7	121.1	120.8
8	126.3	126
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

年齢	の 高さ	女性の 高さ
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、「」を参照してください[パネルの追加](#)。
2. クエリには、必要なデータを返すクエリを記述します。この場合、 などのクエリを使用しますSELECT \* FROM HEIGHT。
3. 散布図を選択します。

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

散布図には、次の 4 つのカスタム設定オプションがあります。

- X 軸 – X 軸として使用するフィールド、および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。各フィールドの回帰線を表示することもできます。リグレッションライン設定の詳細については、次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、凡例内のテキストのサイズを選択したりできます。
- 表示 – グリッドの色や境界線スタイルなど、グラフの他のオプションを設定できます。

### 回帰行の設定

各 Y 軸データセットには、個々のドットに加えて線を表示できます。行タイプには 5 つのオプションがあります。

- なし – 回帰線は表示しません。
- Simple – 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、最適法を使用して直線を表示します。
- エクスポネンシャル — エクスポネンシャルのベストフィット回帰線を表示します。
- 電力 — 電力の最適な回帰線を表示します。

## 統計パネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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。

- 1 つのシリーズまたはフィールドの値のみ。
- 複数のシリーズ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して、テキストを表示するかどうかを制御できます。

### データおよびフィールドオプション

統計の視覚化では、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

以下のオプションを使用して視覚化を絞り込みます。

- 表示 — Amazon Managed Grafana がデータを表示する方法を選択します。
  - 計算 — すべての行に基づいて計算された値を表示します。
    - 計算 - 適用する計算を選択します。使用可能な計算の詳細については、「」を参照してください [計算リスト](#)。
  - 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 つの値に減らすために使用するリデューサー関数を選択します。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。
- 方向 — スタッキング方向を選択します。
  - 自動 – Amazon Managed Grafana は、最適な向きであると思われ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縮します。
  - Vertical – バーは上から下に垂直に伸縮します。
- テキストモード – テキストモードオプションを使用して、パネルに表示され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。値は色を決定するために引き続き使用され、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。名前はツールヒントに表示されます。
  - 値と名前 – 値と名前は常に表示されます。
  - 名前 – 値の代わりに名前を表示します。値はツールヒントに表示されます。
  - なし — 何も表示しない (空)。名前と値は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。
  - 背景 - 背景の色も付けます。
- 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 – 値の下にエリア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**。

- **アラインメントモード** — アラインメントモードを選択します。
  - **自動** — 値が 1 つだけ表示される場合 (繰り返しなし)、値は中央に配置されます。複数のシリーズまたは行が表示される場合、値は左揃えになります。
  - **中央** — 統計値は中央に配置されます。

## 状態タイムラインパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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シリーズが一意的な水平バンドとしてレンダリングされます。ステートリージョンは、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状態に適していますが、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領域に変換します。

### 状態タイムラインオプション

視覚化を絞り込むには、以下のオプションを使用します。

#### 等しい連続値をマージする

Grafana が同じ値を互いに隣り合っている場合にマージするかどうかを制御します。

#### 値を表示する

値が状態リージョン内でレンダリングされるかどうかを制御します。十分なスペースがある場合、Auto は値をレンダリングします。

#### 値の整列

状態リージョン内での値の配置を制御します。

#### 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。

## 線幅

状態リージョンの線幅を制御します。

## 不透明度を埋める

状態リージョンの不透明度を制御します。

## 値マッピング

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リージョンに変換します。

## 凡例オプション

凡例オプションを有効にすると、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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。凡例でしきい値の括弧を表示するには、カラースキームを「しきい値から」に設定します。


凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- List – 凡例をリストとして表示します。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 – グラフの右側。

## ステータス履歴パネル

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

ステータス履歴の視覚化では、定期的な状態が時間の経過とともに表示されます。各フィールドまたはシリーズが水平行としてレンダリングされます。ボックスはレンダリングされ、各値を中心に配置されます。

ステータス履歴の視覚化は、文字列、ブール値、および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を色付けしたり、テキスト値を数値範囲に割り当てることができます。

## 表示オプション

視覚化を絞り込むには、以下のオプションを使用します。

### 値を表示する

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。十分なスペースがある場合、Auto は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。

線幅は、状態リージョンの線幅を制御します。

塗りつぶし不透明度は、状態リージョンの塗りつぶし不透明度を制御します。

### 値マッピング

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。グラデーションカラースキームを使用して色値を作成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、値マッピングまたはしきい値括弧のいずれかを表示できます。凡例に値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。凡例のしきい値括弧を表示するには、カースキームを「From thresholds」に設定します。

凡例モード これらの設定を使用して、視覚化における凡例の表示方法を絞り込みます。

- リスト — 凡例をリストとして表示します。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 – グラフの右側。

## テーブルパネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

テーブルパネルは、時系列とテーブル、注釈、未加工の JSON データに対して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、値の書式、および色付けオプションも表示されます。

## データおよびフィールドオプション

テーブルの視覚化では、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

- ヘッダーを表示 — 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/降順で並べ替え — 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に変更します。選択するたびに、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を変更します。
- [フィルターテーブル列](#) — 列データの表示方法を一時的に変更します。例えば、最大値から最小値への値を順序付けしたり、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 と同じ順序で一覧表示されます。このトピックにリストされているオプションは、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「」を参照してください。[特定のフィールドを設定する](#)。

## 列の配置

Amazon Managed Grafana がセルの内容をどのように整列させるかを選択します。

- Auto (デフォルト)
- 左
- 中央
- [Right] (右)

## 列の幅

デフォルトでは、Amazon Managed Grafana はセルの内容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では、設定を上書きし、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フィールドにと入力すると、フィールドの外部を選択すると、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 は表示設定を自動的に選択します。次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ことで、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、表示モードは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。

## カラーテキスト

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。

## 背景色

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

しきい値レベルは勾配を定義します。

## LCD ゲージ

ゲージは、点灯または点灯していない小さなセルに分割されます。

## JSON ビュー

値はコード形式で表示されます。値がオブジェクトの場合、値を一時停止すると、JSON オブジェクトを参照できる JSON ビューが表示されます。



## 列フィルター

### フィルターテーブル列

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](#)。
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アイコンが表示されます。

### 列値をフィルタリングする

列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 ワークスペースには、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オンにします。上部の検索フィールドにテキストを入力すると、それらの値がディスプレイに表示され、スクロールして検索するのではなく選択できるようになります。

### 列フィルターをクリアする

フィルターが適用された列には、タイトルの横に青いファネルが表示されます。

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、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に設計されています。  
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、「」を参照してください。[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

テキストパネルを使用して、ダッシュボードの情報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。または、別のエディタからコンテンツを貼り付けることができます。

## 時系列パネル

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

時系列パネルでは、時系列を線、ドットのパス、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のに十分な汎用性を備えています。

### **i** Note

グラフパネルの視覚化を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。

時系列の視覚化では、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](#)

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したり、軸を調整したりすることもできます。

これらの設定を使用して、視覚化を絞り込みます。

### ツールヒントモード

グラフにカーソルを合わせると、Grafana はツールヒントを表示できます。ツールヒントの動作を選択します。

- 単一 – ホバーツールヒントには、ホバーしているシリーズのみが表示されます。
- すべて – ホバーツールヒントには、グラフ内のすべてのシリーズが表示されます。Grafana は、ツールヒントのシリーズリストでカーソルを合わせているシリーズを太字で強調表示します。
- 非表示 – ツールヒントは表示しません。

### 凡例モードと配置

凡例の表示方法を選択します。

- List – 凡例をリストとして表示します。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 – 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。

### 時系列を線でグラフ化する

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください [Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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行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。

## パネルを作成する

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、行を選択します。

## 線のスタイル設定

線のスタイル設定にはさまざまなオプションがあります。

- 行補間 — Grafana がシリーズ行を補間する方法を選択します。選択肢は、線形、スムーズ、より前のステップ、より後のステップです。
- 線幅 — 線の太さを 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 - グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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の影響を受けます。

勾配フィルの選択肢は、なし、不透明度、および Hue です。不透明度では、勾配の透明度は y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。Hue では、グラデーション色は線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。

線スタイルの外観は、線幅と塗りつぶし不透明度の設定の影響を受けます。

ラインスタイルの選択肢は、Solid、Dash、および Dots 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、オプションでデータのギャップを接続すべきではないしきい値を設定することもできます。データポイントをギャップで接続しない、データポイントをギャップで常に接続する、またはデータ内のギャップを接続すべきで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、Auto、Always、Never です。

以下をに入力します。

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には

1. 以下に入力するフィールドを選択します。
2. オーバーライドプロパティを追加で、へのフィルを選択します。
3. フィルを停止するシリーズを選択します。

時系列を棒としてグラフ化する

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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バー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。

パネルを作成する

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、バーを選択します。

## バーのスタイル設定

バーのスタイル設定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、の前、センター、およびの後です。
- 線の幅 - 棒のアウトラインの幅を 0~10 ピクセルに設定します。
- 塗りつぶしの不透明度 - バー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 - グラデーションフィルのモードを設定します。フィルグラデーションは線の色に基づいています。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 の設定の影響を受けます。

グラデーションフィルの選択は、なし、不透明度、および Hue です。不透明度 では、勾配の透明度は y 軸の値に基づいて計算されます。フィルの不透明度は、Y 軸の値とともに増加しています。Hue では、グラデーション色は線の色相に基づいて生成されます。

- ポイントを表示 — 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、Auto、Always、Never です。

## 時系列をポイントとしてグラフ化する

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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。

## パネルを作成する

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には、ポイントサイズを選択できます。

- ポイントサイズ – 1~40 ピクセルの半径の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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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の表示を制御する方法と、軸オプションの動作について説明します。

軸にはさまざまなオプションがあります。

- Y 軸配置 — Y 軸の配置を設定します。選択肢は、左、右、非表示です。
- Y 軸ラベル — Y 軸のテキストラベルを設定します。複数の Y 軸がある場合は、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、軸が視覚的に互い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量を比較しやすくなります。
- 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。

ソフト最小またはソフト最大設定は、データがほぼフラットなときに唇が山として表示されないようにし、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派生したハード最小また

は最大フィールドオプションによって、定義されたポイントを超えてスパイクをクリップすることで、断続的なスパイクが有用な詳細を平坦化しないようにします。

- 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。

## グラフ積み上げ時系列

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

このセクションでは、時系列パネルフィールドオプションを使用してシリーズのスタッキングを制御する方法と、スタッキング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 はシリーズを互いに重ねて表示できます。ビジュアライゼーションでスタッキングを使用する場合は、誤解を招くグラフを簡単に作成できるため、注意してください。スタックが最適なアプローチではない理由の詳細については、「[スタックの問題](#)」を参照してください。

## グループ内のスタックシリーズ

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

スタッキング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には

1. 「オーバーライド」セクションで、スタックシリーズ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. 通常のスタッキングモードを選択します。
3. シリーズを表示するスタッキンググループに名前を付けます。スタッキング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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

しきい値は、定義した条件に応じて、値のテキストまたは背景の色を設定します。

しきい値は、次の 2 つの方法のいずれかで定義できます。

- 絶対しきい値は数値に基づいて定義されます。例えば、1~150 のスケールで 80 です。
- パーセントしきい値は、最小値または最大値を基準にして定義されます。例えば、80% です。

次の視覚化に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は、Amazon Managed Grafana は次のデフォルトのしきい値を設定します。80 = red、Base = green、Mode = Absolute。

Base 値は、マイナス無限大を表します。通常は良い色です。

## しきい値の追加

パネルには必要な数のしきい値を追加できます。Grafana ワークスペースは、しきい値を最大値から最小値に自動的にソートします。

### Note

これらの手順は、統計、ゲージ、バーゲージ、およびテーブルの視覚化にのみ適用されません。

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の値を含むしきい値を追加します。

4. レコメンデーションを受け入れるか、新しいしきい値を編集します。
  - 色の編集 – 変更する色のドットを選択し、新しい色を選択します。
  - 番号の編集 – 変更する番号を選択し、新しい番号を入力します。
  - しきい値モード – このパネルのすべてのしきい値に合わせて変更するモードを選択します。
5. 保存 を選択して、ダッシュボードに変更を保存します。

## 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ことで、グラフが特定のしきい値を超えた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. 必要な数のフィールドに入力します。T1 フィールドのみは必須です。
  - T1 – しきい値を表示するには、両方の値が必要です。
  - lt または gt – しきい値が適用される対象を示すには、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 ワークスペースは、その値の y 軸に沿ってしきい値線を描画します。
  - 色 – 色に対応する条件を選択するか、独自の色を定義します。
    - custom – 塗りつぶしの色と行の色を定義します。
    - critical – 塗りつぶしと行の色は赤です。
    - 警告 – 塗りつぶしと線の色は黄色です。
    - ok – 塗りつぶしと線の色は緑色です。
  - 塗りつぶし – しきい値塗りつぶしを表示するかどうかを選択します。
  - Line – しきい値の行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 を選択して、ダッシュボードに変更を保存します。

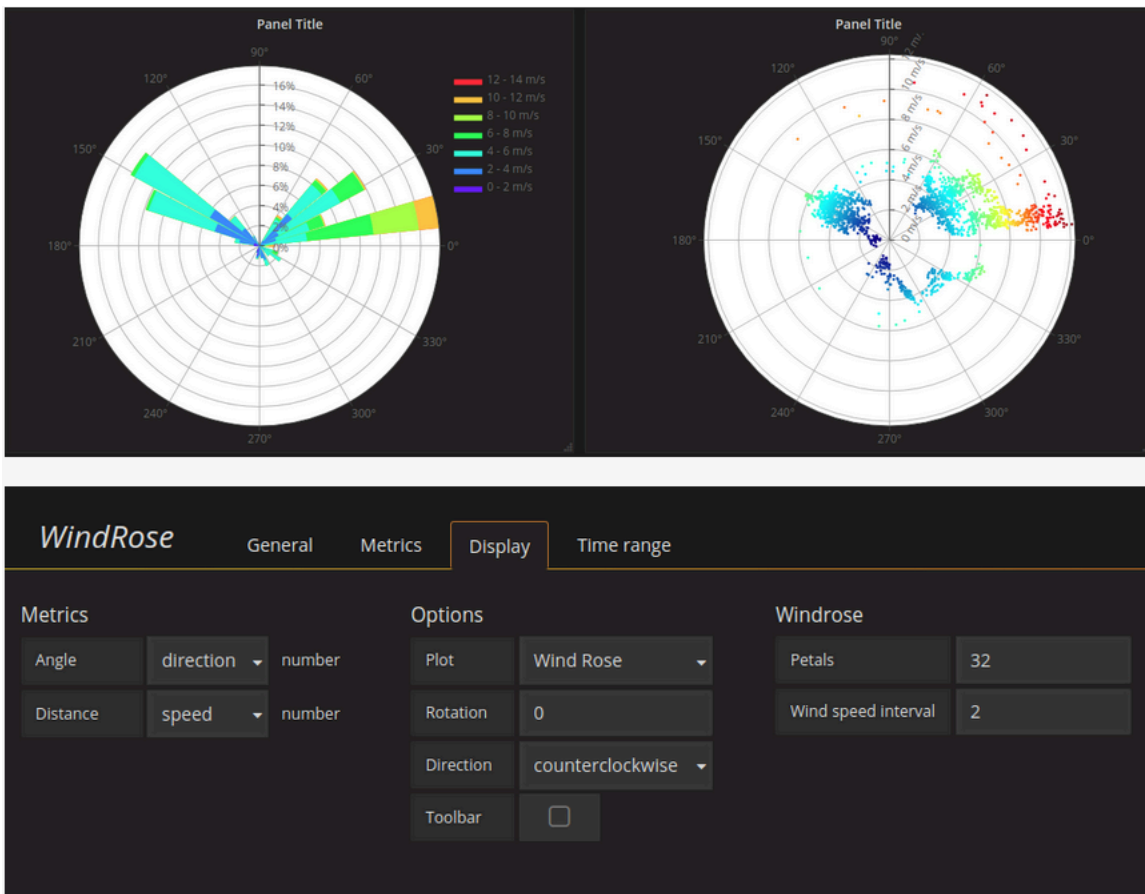
### しきい値の削除

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。(または、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 を選択して、ダッシュボードに変更を保存します。

### WindRose

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

WindRose パネルは、生の時系列データを受け取り、データを変換して WindRose グラフにマッピングします。



## オプション

この WindRose パネルでは、次のオプションがサポートされています。

- 軸周波数
- 軸スタイル (度またはコンパス)
- スケール (線形、正方形、ログ)

## パネルの検査

- ⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

パネルインスペクターは、パネルの理解とトラブルシューティングに役立ちます。Grafana ワークスペースパネルの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、クエリリクエストを表示し、パネルとデータ JSON をエクスポートできます。

## パネルインスペクター UI

パネルインスペクターは、ペインの上部に検査: <NameOfPanelBeingInspected > と表示します。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 4 つのタブで構成されています。

- データタブ – クエリによって返された raw データに、変換が適用された状態を示します。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と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 をプロビジョニングまたは管理する場合に便利です。
- クエリタブ – Amazon Managed Grafana がデータソースにクエリを実行したときに送信された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 つのタブがすべて含まれてい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 raw 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とダウンロード、クエリのパフォーマンスの検査、パネルの JSON モデルの表示、データソースへの未加工のリクエストとレスポンスの表示を行うことができます。

### パネルインスペクターを開く

表示できる任意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、検査するパネルを含む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、「の検査」を選択します。または、パネルタイトルにカーソルを合わせ、iを押します。

パネルインスペクターペインが画面の右側に表示されます。

### raw クエリ結果の検査

テーブルに未加工のクエリ結果を表示します。これらは、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に、変換が適用された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き、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または複数のノードが含まれている場合は、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。
- データの変換
  - 時間で結合 – すべてのクエリの未加工データを同時に表示し、列ごとに1つの結果セットを設定します。列見出しを選択してデータの順序を変更します。

フィールドオプションとオプションオーバーライドが適用されたテーブルで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。

### raw クエリ結果を CSV ファイルとしてダウンロードする

Amazon Managed Grafana は、デフォルトのブラウザのダウンロード場所に CSV ファイルを生成します。選択した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 raw データが表示されるまで、設定を調整します。
3. [Download CSV] 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、CSV のダウンロード を選択する前に Excel 用ダウンロード オプションをオンにします。

## 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、特に数値が予期せず高い場合や低い場合に、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. 統計タブを選択します。

統計は読み取り専用形式で表示されます。

## パネル JSON モデルを表示する

パネル、パネルデータ、およびデータフレーム JSON モデルを調べてエクスポートします。

1. パネルインスペクターを開き、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、次のいずれかのオプションを選択します。
  - パネル JSON – パネルを表す JSON オブジェクトを表示します。
  - パネルデータ – パネルに渡されたデータを表す JSON オブジェクトを表示します。
  - DataFrame 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たり、クリップボードにコピーして別のアプリケーションに貼り付けたりすることができます。

## データソースへの raw リクエストとレスポンスを表示する

1. パネルインスペクターを開き、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. [Refresh] (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、結果を表示します。クエリの特定部分をドリルダウンしたり、すべて展開または折りたたんだり、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るようにしたりできます。

## 計算リスト

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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バーゲージ、ゲージ、統計の視覚化で使用されます。

計算	説明
すべての null	すべての値が null の場合、true
すべてのゼロ	すべての値が 0 の場合、true
変更数	フィールドの値が変更される回数
カウント	フィールドの値の数
差分	値の累積変化
違い	フィールドの最初と最後の値の違い
個別カウント	フィールド内の一意的値の数
最初 (null ではない)	まず、フィールドの null 値ではありません
最大	フィールドの最大値
平均値	フィールド内のすべての値の平均値



計算	説明
最小	フィールドの最小値
最小 (ゼロ以上)	フィールドの最小値、正の値
[Range] (範囲)	フィールドの最大値と最小値の差
[ステップ]	フィールドの値間の最小間隔
合計	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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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付属しています。Amazon Managed Grafana を使用すると、適切なクエリを簡単に構築し、必要なダッシュボードを作成できるように表示プロパティ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を制御するには、ダッシュボードの右上[時間範囲コントロール](#)にある を使用します。

ダッシュボードは、テンプレートと変数を使用して、より動的でインタラクティブなものにすることができます。詳細については、「[テンプレートと変数](#)」を参照してください。

ダッシュボードでは、[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。

ダッシュボードにタグを付けることができ、ダッシュボードピッカーは特定の組織内のすべてのダッシュボードにすばやく検索可能なアクセスを提供します。

## [Rows] (行)

行はダッシュボード内の論理的な分割です。パネルをグループ化するために使用されます。

行の幅は常に 12 ユニットです。これらの単位は、ブラウザの水平解像度に基づいて自動的にスケールリングされます。特定の幅を設定することで、行内のパネルの相対幅を制御できます。

Amazon Managed Grafana は、ユニット抽象化を使用して、すべての画面サイズの外観を最適化します。


### Note

MaxDataPoint 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、行のタイトルを選択します。行を折りたたんだ状態でダッシュボードを保存すると、ダッシュボードはその状態に保存され、行を展開するまでこれらのグラフはロードされません。

繰り返し行機能を使用して、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します。

## 注釈

-  このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

注釈は、グラフ上のポイントをリッチイベントでマークする方法を提供します。注釈を一時停止すると、イベントの説明とイベントタグが表示されます。テキストフィールドには、詳細について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、ネイティブの注釈ストアと、グラフパネルから直接注釈イベントを追加する機能が含まれています。

### 注釈の追加

注釈を追加するには、Ctrl または Cmd を押し、注釈を追加する場所を選択します。注釈を他のダッシュボードから検索できるようにするには、それにタグを追加します。

### リージョンアノテーションの追加

リージョンの注釈を作成するには、Ctrl または Cmd を押しながらリージョンを選択します。

### 組み込みクエリ

注釈を追加すると、その注釈は引き続き表示されます。これは、組み込み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した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します。これには、アラート状態履歴の注釈が含まれます。ダッシュボード設定 (歯車) アイコンを選択し、注釈 を選択し、 という名前のクエリを変更することで、注釈の取得と表示を停止できます Annotations & Alerts (Built-in)。

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、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ません。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は、新しい注釈クエリを追加して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、 によるフィルタリングをタグ に設定することで、ネイティブアノテーションストアからアノテーションを取得する新しいアノテーション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、 という名前の注釈クエリを作成し outages、 という名前のタグを指定します outage。このクエリには、作成した (ダッシュボードから、または API 経由で) outage タグを持つすべての注釈が表示されます。

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。指定したタグの少なくとも 1 つを含む注釈を表示するには、任意の 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、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。

## 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。新しい注釈クエリをダッシュボードに追加するには、ダッシュボード設定 (歯車) アイコンを選択し、 を選択し Annotations、新しい を選択します。

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を表示または非表示にするためのチェックボックスに表示されます。例えば、Deploysと という名前の 2 つの注釈クエリがあるとしますOutages。チェックボックスをオンまたはオフにして、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](#)

## ダッシュボードフォルダ

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

フォルダは、ダッシュボードを整理およびグループ化する方法です。これは、ダッシュボードが多数ある場合や、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便利です。

## フォルダの作成

フォルダを作成するには、次のいずれかを実行します。

- サイドメニューの + アイコンで、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、 の作成を選択します。

ダッシュボードを保存するときは、既存のフォルダを選択するか、新しいフォルダを作成できます。

## ダッシュボードの管理

ダッシュボードの管理ページで、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許可を設定できます)。

## 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、検索結果のダッシュボードリストまたはダッシュボードの管理ページのフォルダで一時停止したときに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、次のタスクを実行できます。

- フォルダ内のダッシュボードを移動または削除します。
- フォルダの名前を変更します (設定タブ)。
- フォルダのアクセス許可を設定します (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て、そのフォルダに含まれるダッシュボードに継承できます。アクセスコントロールリスト (ACL) は、組織ルール、チーム、および個々のユーザーに許可を割り当てることができる場合に使用されます。詳細については、「[ダッシュボードとフォルダのアクセス許可](#)」を参照してください。

## プレイリスト

**△** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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。

Amazon Managed Grafana は、ダッシュボードを大画面を含む任意の解像度に自動的にスケールアップします。

Dashboards サブメニュー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。

### プレイリストの作成

プレイリストは、ダッシュボードの順序と時間間隔を設定した順序で表示します。

1. プレイリスト機能にアクセスするには、サイドメニューで一時停止します。
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. 名前テキストボックスに、プレイリストの名前を入力します。
5. 「Interval」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 が特定のダッシュボードに留まり、プレイリストの次のダッシュボードに進むまでの時間です。

6. プレイリストに追加する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。
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、作成時または保存後に編集できます。

1. プレイリスト機能にアクセスするには、サイドメニューで一時停止します。
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。

### プレイリストの名前の編集

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。

### プレイリストへのダッシュボードの追加

1. 追加する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、名前によるダッシュボードの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、星のステータスまたはタグで結果をフィルタリングします。デフォルトでは、スター付きダッシュボードがプレイリストに追加す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再配置

1. 移動す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. x アイコンを選択して、プレイリストからダッシュボードを削除します。
2. [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. 削除する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してプレイリストページに追加し、そこで開始できます。保存する前に、プレイリストに表示するすべてのダッシュボードを必ず追加してください。

1. プレイリスト機能にアクセスするには、サイドメニューで一時停止します。
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、および少なくとも 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

プレイリストは 5 つの異なるビューモードで開始できます。このモードでは、メニューとナビゲーションバーをダッシュボードに表示する方法が決まります。

デフォルトでは、各ダッシュボードは間隔フィールドに入力された時間で表示されます。これは、プレイリストの作成または編集中に設定できます。プレイリストを起動したら、画面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。



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で、次のいずれかの表示モードを選択します。
  - 通常モード
    - サイドメニューは引き続き表示されます。
    - 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。
  - TV モード
    - サイドメニューは非表示または削除されます。
    - 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。
    - ユーザーが 1 分間何も操作しないと、テレビ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、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 ?inactive に パラメータを追加します。
    - テレビモードは、マウスの動きやキーボードアクションで無効にできます。
  - 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または削除されます。
    - 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自動的に調整され、画面上のスペースが最適化されます。
  - Kiosk モード
    - サイドメニュー、ナビゲーションバー、行、およびパネルのコントロールは完全に非表示にされるか、ビューから削除されます。
    - 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 Kiosk モードを手動でオンにすることができます。
    - 同じショートカットを使用して Kiosk モードを手動でオフにできます。
  - Kiosk 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、およびパネルのコントロールは完全に非表示にされるか、ビューから削除されます。
    - ダッシュボードパネルは自動的に調整され、画面上のスペースが最適化されます。

## 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。

Button	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 (二乗)	プレイリストを終了し、現在のダッシュボードに終了します。
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表示モードに変更します。
[Time range] (時間範囲)	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、またはカスタム時間範囲を表示するように設定できます。
更新 (円矢印)	ダッシュボードを再ロードして、現在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秒ごとに自動的にリロードするように設定できます。

キーボードからプレイリストを停止するには、Esc を押します。

### ビューモードでのプレイリストの共有

必要なビューモードで URL をコピーし、URL を送信先に貼り付けることで、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、目的の表示モードを選択します。
3. URL をクリップボードにコピーするには、リンクアドレスのコピー を選択します。

例えば、Kiosk モードで Grafana Play サイトのプレイリストの URL は、<https://play.grafana.org/d/000000010/annotations?orgId=1&kiosk>

4. URL を送信先に貼り付けます。

## ダッシュボード検索

- △** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

ダッシュボードは、ダッシュボード名で検索したり、1 つ (または複数) のタグでフィルタリングしたり、星のステータスでフィルタリングしたりできます。ダッシュボード検索には、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アクセスできます。ダッシュボード検索は、ショートカット F を使用して開くこともできます。

キーボードのみを使用する場合は、キーボードの矢印キーを使用して結果を移動し、Enter キーを押して目的のダッシュボードを開きます。

### ダッシュボード名による検出結果

検索バーにダッシュボード名の任意の部分を入力します。入力すると、検索は文字列の部分一致の結果をリアルタイムで返します。

ダッシュボード検索は次のとおりです。

- リアルタイム
- 大文字と小文字を区別しない
- 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の機能

### タグでフィルタリング

タグは、特にダッシュボードの数が増えるにつれてダッシュボードを整理するのに役立ちます。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。

ダッシュボードリスト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。追加のタグを選択して、リストをさらにフィルタリングできます。

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選択します。タグを選択すると、ダッシュボード検索は即座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は、Tab キーを押してタグリンクに焦点を合わせ、下矢印キーを押してタグを検索し、Enter キー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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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。これにより、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が表示されます。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そ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パブリックに共有する即時の方法です。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削除し、表示されるメトリクスデータとシリーズ名のみをダッシュボードに埋めたままにします。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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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。リンクをコピーすると、現在の時間範囲と選択したテンプレート変数を含む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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Amazon Managed Grafana には、ダッシュボードレベルとパネルレベルの両方で、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、一般的な時間制御、ダッシュボード全体の時間設定、およびパネル固有の時間設定について説明します。

#### **i** Note

時間制御を行うには、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

次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)
- M (月)
- y (年)

マイナス演算子を使用して、現在を基準にして時間を遡ります。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。

plus 演算子を使用して、現在を基準にした時間を進めます。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	目的
過去 5 分	<code>now-5m</code>	<code>now</code>
これまでの日付	<code>now/d</code>	<code>now</code>
今週	<code>now/w</code>	<code>now/w</code>
週初来	<code>now/w</code>	<code>now</code>
前月	<code>now-1M/M</code>	<code>now-1M/M</code>

### 一般的な時間範囲の制御

ダッシュボードとパネルのタイムコントロールには、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

現在の時間範囲は、タイムピッカーとも呼ばれ、表示している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。

フィールドで一時停止すると、ローカルブラウザなど、範囲とそのソースの正確なタイムスタンプが表示されます。

時間範囲を変更するには、現在の時間範囲で `適用` を選択します。現在の時刻は、過去 15 分などの相対時間範囲、または `2020-05-14 00:00:00 to 2020-05-15 23:59:59` などの絶対時間範囲を使用して変更できます。

## 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
- 昨日の前日
- 先週のこの日
- 今日まで
- 今のところ、この週
- 今月はここまで

## 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。

- From フィールドと To 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、`適用` を選択します。
- From または To 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する日を選択し、`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化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 (グラフの視覚化のみ)

グラフの視覚化で、表示する時間範囲をドラッグして選択します。

## ダッシュボードを更新する

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すぐに実行され、視覚化が更新されます。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。ただし、ダッシュボードを定期的に更新する場合は、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、更新間隔を選択します。

## 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。

ダッシュボードの時間設定にアクセスするには、画面上部にあるダッシュボード設定 (歯車) アイコンを選択します。設定は、全般タブの時間オプションセクションにあります。

- **タイムゾーン** — モニタリングしてい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モニタリング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、または組織のデフォルトで選択されたタイムゾーン。ユーザープロファイル、ユーザーが所属しているチーム、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時間を使用します。
  - ブラウザのタイムゾーン — 使用中のブラウザ用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 ISO 8601 標準タイムゾーン。詳細については、[「タイムゾーンのリスト」](#)を参照してください。
- **自動更新** — 相対時間と自動更新設定をカスタマイズ可能なオプションで選択できます。エントリはカンマで区切られ、任意の有効な時間単位にすることができます。
- **Now delay now** -- 値を上書きする遅延時間nowの値。通常、この機能は、データ集約の既知の遅延に対応して 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にする – タイムピッカーを表示しないオプション。

## パネルの時間オーバーライドと時間シフト

では[クエリオプション](#)、個々のパネルの相対時間範囲をオーバーライドして、上部のナビゲーションバーのダッシュボードのタイムピッカーで選択されているものと異なるようにすることができます。異なる期間または日数のメトリクスを同時に表示できます。

## URL を使用した時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定することで、ダッシュボードの時間範囲を制御できます。

- `from-ms` エポックまたは相対時間で指定された時間範囲の下限を定義します。詳細については、「[相対時間範囲](#)」を参照してください。
- `to-ms` エポックまたは相対時間で指定された時間範囲の上限を定義します。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-` から `time-time.window/2` までの時間範囲を定義します `time+time.window/2`。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、`は ~1499999995000` の 10 秒の時間範囲 `?time=1500000000000&time.window=10000` になります。 `1500000005000`

## 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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Amazon Managed Grafana Dashboards 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 Amazon Managed Grafana JSON 形式でエクスポートされ、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、クエリなどがすべて含まれています。

エクスポート機能は共有ウィンドウでアクセスされ、ダッシュボードメニューの共有ボタンを選択して開きます。

### 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値のテンプレート変数を追加することをお勧めします。

タイプのテンプレート変数Constantは、ダッシュボードに自動的に非表示になります。また、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。テンプレート変数とテンプレート変数の詳細については、「」を参照してください[テンプレートと変数](#)。

### 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 + アイコンを選択し、インポート を選択します。

ダッシュボード JSON ファイルをアップロードしたり、ダッシュボード URL を貼り付けたり、ダッシュボード JSON テキストをテキストエリアに直接貼り付けたりできます。

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) を指定したりできます。

### Grafana.com でダッシュボードを検出する

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[Grafana.com/dashboards](https://grafana.com/dashboards)。

### ダッシュボードのバージョン履歴

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

ダッシュボードのバージョンを保存するたびに、そのバージョンのコピーが保存され、ダッシュボードの以前のバージョンが失われないようにします。これらのバージョンのリストは、Dashboard 設定を選択し、左側のメニューで Versions を選択することで利用できます。

ダッシュボードのバージョン履歴機能を使用すると、以前に保存したダッシュボードの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、リストから比較する 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したら、バージョンの比較を選択して差分ビューを開きます。デフォルトでは、次の図のように、変更のテキストサマリーが表示されます。

ダッシュボードを表す raw JSON の差分を表示するには、JSON 差分を表示 を選択します。

比較する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、差分ビューに表示される「バージョンに復元」<x> 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同じデータを含む新しいバージョンが作成されますが、バージョン番号は異なります。これは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いようにすることができます。

## キーボードショートカット

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Amazon Managed Grafana には、多数のキーボードショートカットがあります。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、キーボードの 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 は現在のダッシュボードを保存します。
- Ctrl+F で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 (非表示のコントロールはテレビディスプレイに適しています)。
- 全画面表示モードまたは編集モードでは、エスケープ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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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 オブジェクトで表されます。ダッシュボードメタデータには、ダッシュボードのプロパティ、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 を表示するには

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 `として表示されます。これはnull、ダッシュボードが保存されるまで割り当てられたデフォルト値です。ダッシュボードを保存すると、idフィールドに整数値が割り当てられます。`

```
{
  "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": []
}
```

次の表は、ダッシュボード JSON の各フィールドの使用の詳細を示しています。

名前	使用方法
id	ダッシュボードの一意的数値識別子 (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 ( や など darklight )。
timezone	ダッシュボードのタイムゾーン (utc または browser )。
editable	ダッシュボードを編集できるかどうか。
graphTool tip	ツールヒントスタイル。 <ul style="list-style-type: none"><li>• 0 共有クロスヘアまたはツールヒントがない場合 (デフォルト)</li><li>• 1 共有クロスヘア用の</li><li>• 2 共有クロスヘアと共有ツールヒント用</li></ul>
time	ダッシュボードの時間範囲 (過去 6 時間、過去 7 日間など)。
timepicke r	timepicker メタデータ。詳細については、「 <a href="#">タイムピッカー</a> 」を参照してください。
templat g	テンプレートメタデータ。詳細については、「 <a href="#">テンプレートと変数</a> 」を参照してください。
annotati ons	注釈メタデータ。詳細については、「 <a href="#">注釈</a> 」を参照してください。
refresh	自動更新間隔。
schemaVer sion	Amazon Managed Grafana の更新によってスキーマが変更されるたびに増分される JSON スキーマ (整数) のバージョン。

名前	使用方法
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分されます。
panels	パネル配列。詳細については、「 <a href="#">パネル</a> 」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す。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。次の例は、テキストパネルのパネル JSON を示しています。

```
"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。各グリッドの高さの単位は 30 ピクセルを表します。
- x - x の位置。x の位置は、と同じ列単位の を使用しますw。
- y - y の位置。y の位置は、と同じグリッドの高さの単位を使用しますh。

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移動します。

## タイムピッカー

次の例は、timepickerオプションを示しています。

```
"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"
}
```

次の表に、 の使用の詳細を示しますtimepicker。

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



名前	使用方法
status	TODO
type	TODO

## テンプレート

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含まれています。次の例はtemplatingメタデータを示しています。

```
"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"
  }
]
```

次の表に、テンプレートセクションの使用の詳細を示します。

名前	使用方法
を有効にする	テンプレートが有効かどうか。

名前	使用方法
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 (、 、 wildcardglobregex、 などpipe )。
現在の	現在選択されている変数テキストまたは値をダッシュボードに表示する
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
マルチ	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
options:	ダッシュボードで選択できる可変テキストと値のペアの配列
query	変数の値を取得するために使用されるデータソースクエリ
更新	TODO
regex	TODO
type	変数のタイプ (custom、 query、 または interval )

## スクリプト化されたダッシュボード

**⚠** このドキュメントトピックは、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に設計されています。  
 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、「」を参照してください [Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

**⚠ Warning**

この機能は廃止され、今後のリリースで削除されます。

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、新しいダッシュボードを常に作成するのに時間がかかることがあります。

スクリプト化されたダッシュボードを使用すると、を使用してダッシュボードを動的に作成できます JavaScript。Grafana インストールフォルダの の下に `public/dashboards/`、という名前のファイルがあります `scripted.js`。このファイルには、スクリプト化されたダッシュボードの例が含まれています。URL を使用してアクセスできます。 `http://grafana_url/dashboard/script/scripted.js?rows=3&name=myName`

を開く `ARGS` と `scripted.js`、変数から URL パラメータを読み取り、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;
```

## その他の例

その他の例は、Grafana インストールの `public/dashboards/` ディレクトリにあります。

## 探索

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 9 での作業](#)。

Grafana ワークスペースでは、ダッシュボード UI に視覚化用のダッシュボードを構築するためのツールが用意されています。クエリに集中できるように、すべてのダッシュボードとパネルオプションをストライプします。クエリが機能するまで繰り返し、ダッシュボードを計画して構築します。

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のために、問題が発生したことをデバッグするために他のツールに切り替える必要がなくなりました。Explore を使用すると、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 を使用すると、ダッシュボードを作成せずにデータを簡単に表示できます。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 は結果をグラフとテーブルの両方として表示します。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

デフォルトでは、ビューワールールを持つユーザーは を編集できず、Explore にアクセスできません。

左側のメニューの探索アイコンで、空の探索タブが開きます。

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探索オプションを選択します。これにより、パネルからクエリを含む探索タブが開きます。その後、ダッシュボードの外部でクエリを微調整または反復できます。

左上のドロップダウンリスト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ます。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。

クエリフィールドでは、クエリを記述してデータを探索できます。クエリフィールドの横には、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 つのボタンがあります。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

分割ビュー機能は、グラフとテーブルを比較したり side-by-side 、関連するデータを 1 ページでまとめて表示したりする方法です。分割 を選択して現在のクエリを複製し、ページを 2 つの side-by-side クエリに分割します。新しいクエリに別のデータソースを選択することもできます。これにより、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、タイムピッカーにアタッチされたタイム同期ボタンのいずれかを選択することで、両方のパネルのタイムピッカーをリンクできます (変更すると、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。

新しく作成された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

共有短縮リンク機能を使用すると、複雑なクエリパラメータを含む長い URLs を共有する / goto/:uid 代わりに、形式のより小さくてシンプルな URLs を作成できます。短縮リンクを作成す

るには、Explore ツールバーの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。

## 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。履歴はブラウザのローカルであり、共有されません。履歴を開いて操作するには、Explore でクエリ履歴を選択します。

### クエリ履歴の表示

Fluxクエリ履歴では、次の操作を実行できます。

- クエリを実行する
- コメントを作成または編集します。
- クエリをクリップボードにコピーします。
- クエリを含む短縮リンクをクリップボードにコピーします。
- クエリをスターします。

### お気に入りのクエリの管理

クエリ履歴タブに星が表示されているすべてのクエリは、星が表示されているタブに表示されます。任意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。

### ソートクエリ履歴

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でソートできます。

クエリ履歴の右側にあるドロップダウンリストで、次のオプションのいずれかを選択します: フィールド。

- 最新の 1 番目
- 最も古いもの
- データソース A~Z
- データソース Z-A

**Note**

分割ビューを使用している場合、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。

## フィルタークエリ履歴

クエリ履歴タブと星付きタブでは、データソース名でクエリ履歴をフィルタリングできます。

1. 特定のデータソースのクエリをフィルタリングする (複数可) 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できません。

クエリ履歴タブでは、垂直スライダーを使用して日付でクエリを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

クエリやコメントをまたいで履歴を検索できます。クエリ履歴と星付きタブのクエリを検索できません。

1. クエリの検索フィールドを選択します。
2. 検索フィールドに、検索語を入力します。

## クエリ履歴の設定

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、使用可能なオプションを示します。



設定	デフォルト値
Grafana がクエリ履歴を保存する期間を指定します。	1 週間
デフォルトのアクティブタブを変更します。	クエリ履歴タブ
Explore で現在アクティブなデータソースのクエリのみを表示します。	True
クエリ履歴をクリアします。	( クエリ履歴をクリア を選択して、保存されたすべてのクエリを完全に削除します。 )

### Note

クエリ履歴設定はグローバルで、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periエンスが含まれています。クエリを実行すると、Grafana は実際に 2 つのクエリを実行します。グラフの通常の Prometheus クエリとテーブルのインスタントクエリです。インスタントクエリは、各時系列の最後の値を返します。この値は、グラフに表示されるデータの適切な概要を示しています。

### 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、メトリクスを選択してメトリクスエクスプローラーを開きます。これは、メトリクスがプレフィックスでグループ化された階層メニューを示しています。例えば、すべての Alertmanager 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調べるための出発点として最適です。

### クエリフィールド

Query フィールドは、メトリクス名、関数の自動入力をサポートし、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 を押してクエリを実行します。

オートコンプリートメニューにアクセスするには、Ctrl+Space を押します。オートコンプリートメニューには、最近実行されたクエリのリストを含む新しい履歴セクションが含まれています。

提案はクエリフィールドの下に表示されます。提案を選択して、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、レート関数が推奨されます。
- バケットの場合、ヒストグラム関数が推奨されます。
- 記録ルールの場合、ルールを展開できます。

## テーブルフィルター

テーブルパネルのラベル列のフィルターボタンを選択して、クエリ式にフィルターを追加します。複数のクエリにフィルターを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

また、Explore を使用して、次のデータソースを使用してログを調査することもできます。

- InfluxDB
- Elasticsearch

## ログの視覚化

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、個々のログは以下に表示されます。データソースがリクエストされた時間範囲のヒストグラムデータを送信しない場合、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ます。ヒストグラムの開始は、結果の最初のログ行のタイムスタンプによって固定されます。時系列の終わりは、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決定し、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、そのコンテンツ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的一致によって決まります。Grafana がログレベルを特定できない場合、不明なログレベルで視覚化されます。次の表に、ログレベルとログレベルの略語と式のマッピングを示します。

サポートされている式	ログレベル	色
Emerg	非常事態	紫色

サポートされている式	ログレベル	色
致命的	非常事態	紫色
アラート	非常事態	紫色
クリティカル	非常事態	紫色
非常事態	非常事態	紫色
エラー	エラー	赤色
エラー	エラー	赤色
エラー	エラー	赤色
警告	warning	黄色
warning	warning	黄色
info	info	green
情報	info	green
通知	info	green
デバッグ	デバッグ	青色
デバッグ	デバッグ	青色
トレース	トレース	水色
*	不明	グレー

## 視覚化オプション

ログの表示方法をカスタマイズし、表示する列を選択できます。

### 時間

このオプションは、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

このオプションは、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。

## ラップライン

ディスプレイでラインラッピングを使用するには、これを True に設定します。このオプションを False 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反復的である可能性があります。Explore は、重複するログ行を非表示にすることで役立ちます。さまざまな重複排除アルゴリズムから選択できます。

- 完全一致 — 完全一致は、日付フィールドを除いて行全体で行われます。
- 数値 — 一致は、期間、IP アドレスなどの数値を除外した後に行上で行われます。
- 署名 — 最も積極的な重複排除。これにより、すべての文字と数字が削除されます。一致は残りの空白と句読点で行われます。

## 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 (最新の順) 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、より堅牢な操作のために、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 (正のフィルター) して除外 (負のフィルター) できます。各フィールドまたはラベルには、表示されるすべてのログに関連する 1 回限りの統計を表示する統計アイコンもあります。

## 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は、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を表示または非表示にします。

{{< docs-imagebox img"/img/docs/explore/parsed-fields-7-2.gif" 最大幅 "800px" 字幕 "Explore" >}} で  
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トレースを視覚化できます。現在サポートされているデータソース：

- [Jaeger データソースに接続する](#)
- [Tempo 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](#)
- [Zipkin データソースに接続する](#)

クエリエディタの使用については、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。

### [Header] (ヘッダー)

ヘッダーには、次の項目が含まれます。

- ルートスパンの名前とトレース ID を示すヘッダータイトル
- 検索。検索されたテキストを含むスパン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。ミニマップの上にマウスをドラッグすると、より小さな時間範囲にズームインできます。ズームするとメインタイムラインも更新されるため、短いスパンを簡単に確認できます。ミニマップで一時停止すると、ズーム時に選択のリセットボタンが表示され、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、トレース内のスパンのリストが表示されます。各スパン行は以下のコンポーネントで構成されます。

- 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む
- サービス名: スパンを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 Logs 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 からダッシュボードへの定期的な切り替え、またはその逆を含むワークフローを高速化するために、パネルのドロップダウンから Explore に移動した後にオリジンダッシュボードに戻る機能を追加しました。

Explore に移動したら、Explore 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オリジンダッシュボードに戻るか、Explore で行った変更をダッシュボードに戻す場合は、ボタンの横にある矢印を選択して「変更を含むパネルに戻る」メニュー項目を表示します。

## クエリインスペクター

クエリのデバッグに役立つように、Explore では、クエリインスペクターを介してクエリリクエストとレスポンス、およびクエリ統計を調査できます。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に似ています。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 raw リクエストとレスポンスを表示する](#)」を参照してください。

## Linking (リンク中)

**⚠** このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

リンクを使用して、一般的に使用されるダッシュボード間を移動したり、他のユーザーを視覚化に接続したりできます。リンクを使用すると、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 は、ダッシュボードリンク、パネルリンク、および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ればよいですか？

まず、現在ダッシュボード間をどのように移動しているかを調べます。多くの場合、ダッシュボードのセット間でジャンプし、それぞれで同じコンテンツを見つけるのに苦労している場合、リンクはワークフローの最適化に役立ちます。

次のステップでは、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、さまざまなコンテンツで動作します。

- ダッシュボードのパネルのほとんど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。
- 外部サイトにリンクするには、ダッシュボードリンクまたはパネルリンクを使用できます。
- 特定のシリーズ、または単一の測定にドリルダウンするには、[データリンク](#)を使用します。

## URL を使用した時間範囲の制御

ダッシュボード URL に次のクエリパラメータを指定することで、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、ms エポックで指定された時間範囲の下限を定義します。
- `to` は、ms エポックで指定された時間範囲の上限を定義します。

- `time` および `from` は、 `from-time-time.window/2` までの時間範囲 `time.window` を定義します `time+time.window/2`。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 `from` から `1499999995000` までの 10 秒の時間範囲になります。 `1500000005000`

## ダッシュボードリンク

- ⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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て、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見ていることを確認することができます。他のタイプのリンクについては、「」を参照してください [データリンク変数](#)。

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ながら、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。
  - タグ付き - 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。
  - 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、このオプションを選択し、ドロップダウンリストにオプションのタイトルを追加します。このオプションが選択されてい



い場合、Grafana ワークスペース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 — 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、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、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がリンクに含まれます。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内の一致するテンプレートはリンクの値に設定されます。
- 新しいタブで開く — 新しいタブまたはウィンドウでダッシュボードリンクを開くには、このオプションを選択します。

5. [追加] を選択します。

### ダッシュボードへの URL リンクの追加

現在のダッシュボードの上部に 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、使用可能な任意の URL にリンクできます。時間範囲を制御して、ユーザーが Grafana ワークスペースに特定のデータを表示できるようにすることもできます。

1. リンク先のダッシュボードを表示しながら、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ができます。
  - 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 — ユーザーが一時的に停止したときに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 — 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、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。
    - from は、ms エポックで指定された時間範囲の下限を定義します。
    - to は、ms エポックで指定された時間範囲の上限を定義します。

- `time` および `from` は、`from` から `time-time.window/2` までの時間範囲 `time.window` を定義します `time+time.window/2`。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、`from=1499999995000&time=1500000005000` の 10 秒の時間範囲? `time=1500000000000&time.window=10000` になります `1500000005000`。
- 変数値 – このオプションを選択すると、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がリンクに含まれます。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内の一致するテンプレートはリンクの値に設定されます。例えば、`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 – 新しいタブまたはウィンドウでダッシュボードリンクを開くには、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、次の手順を使用します。

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、 の更新 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、削除するリンクのごみ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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

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 URL URLs にリンクできます。時間範囲を制御して、ユーザーが Grafana ワークスペースに特定のデータを表示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、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、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します。
3. リンク を展開し、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。タイトルは UI に表示されます。
5. リンク先の URL を入力します。ダッシュボードで定義されている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+Space または Cmd+Space を押し、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、リンクはユーザーを適切なコンテキストに送信し、関連する変数は既に設定されています。時間変数を使用することもできます。
  - from は、ms エポックで指定された時間範囲の下限を定義します。
  - to は、ms エポックで指定された時間範囲の上限を定義します。
  - time および は、 から time-time.window/2までの時間範囲time.windowを定義しますtime+time.window/2。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、 は ~1499999995000 の 10 秒の時間範囲?  
time=1500000000000&time.window=10000になります1500000005000。
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」を選択して、ダッシュボードに変更を保存します。

### 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、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 アイコンを選択します。
3. 右上の「保存」を選択して、ダッシュボードに変更を保存します。

### データリンク

**⚠** このドキュメントトピックは、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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

データリンクは、リンクへのより詳細なコンテキストを提供します。シリーズ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、視覚化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、バーゲージなどのパネルでは、視覚化上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。

データリンクの変数を使用して、保持されたデータフィルターを含む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、時間範囲、シリーズ、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。

## 先行入力の提案

データリンクを作成または更新するときは、キーボードの Ctrl+Space または Cmd+Space Cmd+Space を押して先行入力候補を開き、URL に変数を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、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、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します。
3. データリンク を展開し、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。タイトルは UI に表示されます。
5. リンク先の URL を入力します。

ダッシュボードで定義されているテンプレート変数のいずれかを追加できます。URL フィールドを選択し、\$ と入力するか、Ctrl+Space または Cmd+Space を押して、使用可能な変数のリストを表示します。パネルリンクにテンプレート変数を追加すると、リンクはユーザーを適切なコンテキストに送信し、関連する変数は既に設定されています。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」を選択して、ダッシュボードに変更を保存します。

### 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、編集リンク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」を選択して、ダッシュボードに変更を保存します。

### 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 アイコンを選択します。

3. 右上の「保存」を選択して、ダッシュボードに変更を保存します。

## データリンク変数

データリンクで変数を使用して、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、「」を参照してください[データリンク](#)。

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 URLs でテンプレート変数を使用することもできます。詳細については、「[テンプレートと変数](#)」を参照してください。

## 時間範囲パネル変数

次の変数を使用して、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。

## シリーズ変数

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 にシリーズ名を追加します。
- `__series.labels.<LABEL>` — URL にラベルの値を追加します。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` - 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場合は、リンクを選択するすべてのユーザーの変数値を選択します。

を使用します。ここで `var-myvar=${myvar}`、`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。

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。

## テンプレートと変数

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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。サーバー、アプリケーション、センサー名などのものをメトリクスクエリでハードコーディングする代わりに、変数を代わりに使用できます。

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、完全な編集権限を与えたくない管理者にとって特に役立ちます。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、ダッシュボードを単一ソース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単になります。

サポートされている変数タイプのリスト、および各変数タイプを追加する手順については、「」を参照してください。 [可変タイプ](#)

## テンプレート

テンプレートは、変数を含む任意のクエリです。

例えば、複数のサーバーをモニタリングするダッシュボードを管理していた場合は、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、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"}
```

変数値は常に構文を使用して URL に同期されます `var-<varname>=value`。

## 変数のベストプラクティス

変数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リスト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き、ダッシュボードの左端に最初に表示されるようにします。

## 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表示できます。

- `$varname` この構文は、次の例のように読みやすいです: `apps.frontend.$server.requests.count`。ただし、単語の途中で変数を使用することはできません。
- `${var_name}` この構文は、式の途中で変数を補間する場合に使用します。
- `${var_name:<format>}` この形式により、Grafana が値を補間する方法をより詳細に制御できます。詳細については、「[高度な変数形式のオプション](#)」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その使用場所に準拠するようにエスケープされ



場合があります。例えば、Prometheus クエリの正規表現式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フォーマットを上書きする高度な構文については、「」を参照してください [高度な変数形式のオプション](#)。

## 可変タイプ

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

Grafana はいくつかのタイプの変数を使用します。

変数タイプ	説明
Query	メトリクス名、サーバー名、センサー IDs、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。詳細については、「 <a href="#">クエリ変数の追加</a> 」を参照してください。
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、「 <a href="#">カスタム変数の追加</a> 」を参照してください。
[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テキスト入力フィールドを表示します。詳細については、「 <a href="#">テキストボックス変数の追加</a> 」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、「 <a href="#">定数変数の追加</a> 」を参照してください。

変数タイプ	説明
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、「 <a href="#">データソース変数の追加</a> 」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、「 <a href="#">間隔変数の追加</a> 」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/値フィルター (InfluxDB Prometheus、および OpenSearch のみ)。詳細については、「 <a href="#">アドホックフィルターの追加</a> 」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。詳細については、「 <a href="#">グローバル変数</a> 」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、「 <a href="#">連鎖変数</a> 」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作成できます。例えば、クエリ変数は、サーバー名、センサー IDs、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フェッチすると変化します。

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、「」のデータソースのドキュメントを参照してください [データソースに接続する](#)。

## 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、ページの上部にある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、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名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、「」を参照してください[データソースに接続する](#)。
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。
  - Never - 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、大きく変更する場合は問題になります。
  - ダッシュボードロード時 - ダッシュボードがロードされるたびにデータソースをクエリします。これにより、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、ダッシュボードのロードが遅くなります。
  - 時間範囲変更時 - 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をクエリします。このオプションは、変数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。
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で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行を一時停止し、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式を入力して、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、「[正規表現を使用した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を選択します。デフォルトのオプション Disabled 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ることを意味します。
  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リストが表示されます。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、変数をダッシュボードに追加します。

## 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、変更されないサーバー名やリージョン名がある場合は、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、「」を参照してください[連鎖変数](#)。

## 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、ページの上にある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、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名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 - 変数リスト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。数字、文字列、その他の変数、またはコロンで区切られた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リストが表示されます。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、変数をダッシュボードに追加します。

## テキストボックス変数の追加

テキストボックス変数には、オプションのデフォルト値を含むテキスト入力フィールドが表示されます。任意の値を入力できるため、これは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、または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、ページの上部にある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、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名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と、テキストを入力できる空のテキストボックスが Grafana に表示されます。
2. 値 のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

### 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のメトリクスパスプレフィックスに役立ちます。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。

定数変数には柔軟性がありません。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が、すべてのクエリで再入力したくない複雑な値がある場合に便利です。例えば、というサーバーパスがある場合 `i-0b6a61efe2ab843gg`、それを という変数に置き換えることができます `$path_gg`。

## 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、ページの上にある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 リストで定数 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名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。文字、数字、記号を入力できます。高度な変数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、「[高度な変数形式のオプション](#)」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、変数をダッシュボードに追加します。

## 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。これらは、データソースのインスタンスが複数ある場合に便利です。場合によっては、異なる環境にもあります。

## 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、ページの上にある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、データソース 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名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. Type 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、「」を参照してください[データソースに接続する](#)。
2. ( オプション) インスタンス名フィルター で、データソースインスタンスが選択できる正規表現フィルターを変数値のドロップダウンリストから入力します。このフィールドを空のままにして、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 のプレビューでは、Grafana は現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。



## 間隔変数の追加

間隔変数を使用して、1m、1h、などの時間範囲を表します1d。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。Auto オプションを使用して、時間範囲ごとに設定された数のデータポイントを返すこともできます。

間隔変数は、時間 (InfluxDB の場合)、日付ヒストグラム間隔 ( の場合)、または要約関数パラメータ (Graphite の場合 OpenSearch) でグループ化するパラメータ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、ページの上にある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、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名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、変数の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。次の時間単位がサポートされています: s (seconds)、m (minutes)h (hours)、d (days)、w (weeks)、M (months)、および y (years)。デフォルト値 を承諾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) オプションをリストに追加する場合は、自動autoオプションをオンにします。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割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30の1分単位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 — ステップカウント間隔が時間を割かない最小しきい値。30分の例を続行するには、最小間隔がに設定されている場合2m、Grafanaはデータを15の2分単位でグループ化します。
3. 値のプレビューでは、Grafanaは現在の変数値のリストを表示します。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、変数をダッシュボードに追加します。

### 間隔変数の例

Graphite 関数myintervalで テンプレート変数を使用する例：

```
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、指定し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、それらを使用して既存のクエリのフィルターを記述します。

#### Note

注：1回限りまたはアドホックのフィルター変数は、InfluxDBPrometheus、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、ページの上にある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、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、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名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、「」を参照してください[データソースに接続する](#)。
2. 追加 を選択して、変数をダッシュボードに追加します。

## 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柔軟な変数オプションの 1 つです。この変数は、変数オプションの通常のリストの代わりに、ダッシュボード全体のアドホッククエリの構築を可能にします。この方法で適用したフィルターは、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 と呼ばれ、変数クエリに 1 つ以上の他の変数を含むクエリ変数です。このセクションでは、連鎖変数の仕組みについて説明し、連鎖変数を使用するダッシュボードの例へのリンクを提供します。

連鎖変数クエリはデータソースごとに異なりますが、前提はすべてのもので同じです。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。

5 レベルまたは 10 レベルの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を構築できます。技術的には、どれだけ深く、複雑に行け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

- 変数を連鎖すると、親子の依存関係が作成されます。これらは、はしごまたは木として構想できます。
- 新しい連鎖変数を作成する最も簡単な方法は、新しい変数のベース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重複変数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 を追加できます。
- この方法で作成した新しい連鎖変数がリストの下部に表示されます。リストに論理的な順序を付けるには、変数をリスト内の別の位置にドラッグします。

### 可変順序

ダッシュボード可変リストの可変の順序を変更するには、各エントリの右側にある上矢印と下矢印を選択します。Grafana ワークスペースには、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が一覧表示され、左端のリストの上部に変数が表示されます。

- 子変数の前に、依存関係のない変数をリストします。
- 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。エンドユーザー (および自分自身) に対してより明確にするために、変数を論理的な順序で一覧表示します。

### 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、リンクされた一連の 4 つの変数 (国、地域、サーバー、メトリクス) があり、ルート変数の値 (国) を変更する場合、Grafana ワークスペースはダッシュボードの視覚化を更新する前に、すべての依存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利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リストし、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、およびデータリンクに役立ちます。

### `$_ダッシュボード`

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 つの時間範囲変数が組み込まれています。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ますが、日付の書式を制御できます。

構文	結果の例	説明
<code>\${__from}</code>	1594671549254	Unix ミリ秒エポック
<code>\${__from:date}</code>	2020-07-13T20:19:09.254Z	引数なし、デフォルトは ISO 8601/RFC 3339
<code>\${__from:date:iso}</code>	2020-07-13T20:19:09.254Z	ISO 8601/RFC 3339
<code>\${__from:date:seconds}</code>	1594671549	Unix 秒エポック
<code>\${__from:date:YYYY-MM}</code>	2020-07	任意のカスタムデータ形式。詳細については、 <a href="#">「の表示」</a> を参照

構文	結果の例	説明
		してください。

上記の構文は `${__to}` でも機能します。

この変数は URLs 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表示するダッシュボードにエンドユーザーを送信するには、`https://play.grafana.org/d/000000012/grafana-play-home?viewPanel=2&orgId=1?from=now-6h&to=now` という URL を使用します。

## \$\_\_interval

`$__interval` 変数は、時間別にグループ化するパラメータ (InfluxDB、Myself、Postgres、MSSQL の場合)、日付ヒストグラム間隔 ( の場合 )、または要約関数パラメータ (Graphite の場合 OpenSearch) 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くのデータポイントがある場合、間隔を長くしてグループ化することで、クエリをより効率的に行うことができます。例えば、3 か月のデータを見ると、1 日でグループ化するのが 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、クエリは速くなります。`$__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。

概算計算:  $(\text{from} - \text{to}) / \text{resolution}$

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 に計算され、2m 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) になることがあります。ポイントは日別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 は同じ変数 `$interval` です。代わりに `$__interval` を使用します。

InfluxDB と OpenSearch データソースには、間隔をハードコードしたり、>構文-> 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 Group by time interval フィールドがあります >10m。

## \$\_\_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 の場合 20m、`$__interval_ms` は です 1200000。

## `$_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。変数はシリーズ名またはエイリアスに置き換えられます。

## `$_org`

この変数は、現在の組織の ID です。変数は現在の組織の名前 `$_org.name` です。

## `$_user`

変数は現在のユーザーの ID `$_user.id` です。変数は、現在のユーザーのログインハンドル `$_user.login` です。変数は、現在のユーザーの E メール `$_user.email` 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。これは `to - from` によって計算されます。ミリ秒と秒の表現は `$_range_ms` および `$_range_s` と呼ばれています。

## `$timeFilter` または `$_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、時間範囲間隔 Last 7 days 式は `time > now() - 7d`。

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。Grafana は、クエリエディタモードで 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手動で追加できません: `WHERE $timeFilter`。
- Azure Monitor データソースに分析クエリをログ記録します。
- MySQL、Postgres、MSSQL の SQL クエリ。
- `$_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、使用可能な他の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して、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オプションであり、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できます。詳細については、「[複数値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。

### すべてのカスタム値

このオプションは、Include All オプション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、Custom all value 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のように、カスタムのオール値を指定する方がよい場合があります。

Custom all value 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、その構文はエスケープされないため、データソースの有効な値を考慮する必要があります。

### 高度な変数形式のオプション

変数補間の形式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したい場合があります。

例えば、MySQL データソースのデフォルトは、複数の値を引用符で区切って結合することです 'server01', 'server02'。場合によっては、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: server01,server02。これを行うには、次の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、またはフォールバックオプション`glob`です。

#### CSV

複数の値を持つ変数をカンマ区切りの文字列としてフォーマットします。



```
servers = ['test1', 'test2']  
String to interpolate: '${servers:csv}'  
Interpolation result: 'test1,test2'
```

## 分散型 - OpenTSDB

OpenTSDB のカスタム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:distributed}'  
Interpolation result: 'test1,servers=test2'
```

## 二重引用符

単一値変数と複数値変数をカンマ区切り"の文字列にフォーマットし、 で各値をエスケープし\"、各値を 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

複数の値を持つ変数をカンマ区切りの文字列としてフォーマットします。

```
servers = ['test1', 'test2']  
String to interpolate: '${servers:json}'
```

```
Interpolation result: '["test1", "test2"]'
```

## Lucene - 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 OpenSearch。

```
servers = ['test1', 'test2']  
String to interpolate: '${servers:lucene}'  
Interpolation result: '("test1" OR "test2")'
```

## パーセントエンコード

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の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 クエリの一重引用符など、データソース固有の書式設定をオフにします。

```
servers = ['test1.', 'test2']  
String to interpolate: '${var_name:raw}'  
Interpolation result: '{test.1,test2}'
```

## [Regex] (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)'
```

## シングルクオート

単一値変数と複数値変数をカンマ区切り'の文字列にフォーマットし、 によって各値をエスケープ\ 'し、各値を で引用符で囲みます'。

```
servers = ['test1', 'test2']  
String to interpolate: '${servers:singlequote}'  
Interpolation result: "'test1','test2'"
```

## SQLstring

単一値変数と複数値変数をカンマ区切り'の文字列にフォーマットし、 によって各値をエスケープ' ' 'し、各値を で引用符で囲みます'。

```
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、テキスト表現を と組み合わせて返します+。

```
servers = ["test1", "test2"]  
String to interpolate: '${servers:text}'  
Interpolation result: "test1 + test2"
```

## 複数値変数のフォーマット

複数の値が選択された変数を補間することは、変数が使用され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、難しいです。Grafana は、各データソース

プラグインが複数の値に使用する形式をテンプレート補間エンジンに通知できるようにすることで、この問題を解決しようとしています。

#### 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、変数のカスタムオール値オプションが空白である必要があります。空白のままにすると、Grafana はクエリ内のすべての値を連結 (結合) します。のようなものです `value1,value2,value3`。カスタム `all` 値を使用する場合、代わりに `*または` のような値になります `all`。

### 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 {host1,host2,host3}` であるかのように補間されます。

### Prometheus または InfluxDB データソースを使用した複数値変数

InfluxDB と Prometheus は正規表現式を使用するため、同じ変数がとして補間されます (`host1|host2|host3`)。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、正規表現制御文字を持つ値は正規表現式を破棄します。

### 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、同じ変数がとしてフォーマットされます (`"host1" OR "host2" OR "host3"`)。この場合、値に Lucene の制御語と引用符のみが含まれるように、すべての値をエスケープする必要があります。

### 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可能性があります。特に InfluxDB と Prometheus では、正規表現構文の使用時に 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るため、問題の背後にあるロジックを把握するのは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。

- 複数値を含むすべてのオプションオプションをオフにします。
- `[raw 変数形式]({{< relref "advanced-variable-format-options.md#raw" >}})` を使用します。

## 正規表現を使用した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、変数クエリによって返されるオプションのリストをフィルタリングしたり、返されるオプションを変更したりできます。

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リスト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。

以下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 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02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ためのオプションのフィルタリングと変更

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、変数クエリによって返されるオプションから個別の「テキスト」部分と「値」部分をキャプチャできます。変数ドロップダウンリストには、選択できる各値の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` 値よりもフレンドリです。次の変数クエリ結果から始めます。

```
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して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、すべてのサービスに対して1つのダッシュボードを再利用できます。

テンプレート変数は、ダッシュボード全体でクエリを動的に変更するのに非常に役立ちます。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する場合は、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 オプションがオンになっている変数がある場合は、パネルを1つ選択し、選択した値ごとにそのパネルを繰り返すように Grafana に指示できます。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。

は、パネルの配置方法 direction を制御します。

を選択すると horizontal、パネルは に配置されます side-by-side。Grafana は、繰り返される各パネルの幅を自動的に調整して、行全体が塗りつぶされるようにします。現在、行上の他のパネルと繰り返しパネルを混在させることはできません。

を設定Max per rowして、行あたりのパネルの最大数を Grafana に指示します。デフォルトは 4 です。

を選択するとvertical、パネルは列の上から下に配置されます。繰り返されるパネルの幅は、繰り返される最初のパネル (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) にのみ変更を加えます。変更がすべてのパネルに適用されるには、動的なダッシュボードの再構築を開始する必要があります。これを行うには、変数値 (つまり、繰り返しの基準) を変更するか、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、変数で 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

上記のパネルで説明したように、Multi-valueまたは Include all value 選択オプションで変数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、行タイトルを一時停止し、歯車ボタンを選択してRow Options設定パネルにアクセスします。その後、行を繰り返す変数を選択できます。

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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Grafana アラートは、システムの問題が発生した直後にその問題を把握し、サービスの中断を最小限に抑えるのに役立つ、堅牢で実用的なアラートを提供します。



Amazon Managed Grafana には、アラート情報を 1 つの検索可能なビューに一元化する、更新されたアラートシステムである Grafana アラートへのアクセスが含まれています。これには、次の機能が含まれます。

- 集中ビューで Grafana アラートを作成および管理します。
- Cortex および Loki マネージドアラートを 1 つのインターフェイスで作成および管理します。詳細については、「[アラートルールを管理する](#)」を参照してください。
- Prometheus、Amazon Managed Service for Prometheus、およびその他の Alertmanager 互換データソースからのアラート情報を表示します。
- 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を作成します。詳細については、「[単一および多次元ルール](#)」を参照してください。
- terraform またはプロビジョニング APIs を使用してアラートリソースを管理します。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、デフォルトは [Classic ダッシュボードアラート](#)。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、「」を参照してください [Grafana アラートの新機能](#)。

Grafana アラートには 4 つの主要コンポーネントがあります。

- [アラートルール](#) - アラートが開始されるかどうかを決定する評価基準。1 つ以上のクエリと式、条件、評価の頻度、およびオプションで条件が満たされる期間で構成されます。
- [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を送信するためのチャンネル。
- [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致基準とグループ化基準のセット。
- [無音](#) - 通知の無音に使用される日付と一致基準。

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 マネージド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](#)
- [アラートの連絡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](#)

- [無音を追加または編集する](#)

## 制限事項

- 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 Alertmanager データソースからルールを取得できます。サポートされている他のデータソースから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センターに送信します。ネイティブの Grafana アラートを使用している場合は、従来のダッシュボードアラートを維持し、新しい Grafana アラート機能を有効にしない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、Prometheus Alertmanager で作成されたアラートの通知を 1 つだけ送信する Grafana Alerting を有効にすることを勧めます。

## トピック

- [Grafana アラートの新機能](#)
- [クラシックダッシュボードアラートを Grafana アラートに移行する](#)
- [アラートの基礎](#)
- [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シング](#)
- [コンタクトポイントの使用](#)
- [メッセージング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に設計されています。  
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、「」を参照してください[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください[Grafana バージョン 9 での作業](#)。

Grafana アラートには、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あります。

## 多次元アラートの作成

システム全体の可視性を提供する単一のアラートルールを作成し、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を生成できるようになりました。例えば、1つのホスト上の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、アラートクォータにカウントされます。アラートクォータ内で対応できる数よりも多くのインスタンスを作成する多次元ルールは評価されず、クォータエラーを返します。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。

## ダッシュボードの外部でアラートを作成する

従来のダッシュボードアラートとは異なり、Grafana アラートを使用すると、複数のソースからのデータを一意の方法で組み合わせるクエリと式を作成できます。引き続き、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、監視対象のシステムを迅速にトラブルシューティングできます。

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 E メールにイメージやクエリ値は含まれません。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 と Cortex のアラートルールを作成する

Grafana アラートでは、Grafana マネージドアラートと同じ UI と API を使用して Loki および Cortex アラートルールを管理できます。

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を表示および検索する

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。複数のデータソースのラベルを検索して、関連するアラートをすばやく見つけることができます。

## アラート状態 NoData とエラーの特別なアラート

Grafana アラートは、アラートルールの評価によって状態 NoData または が生成されると、次のラベルを持つ特別なアラートを生成します Error。


- `alertname` 状態 `DatasourceError` に応じて `DatasourceNoData` または  の値を指定します。
- `rulename` と、特別なアラートが属するアラートルールの名前。
- `datasource_uid` には、状態の原因となったデータソースの UID があります。
- 元のルールすべてのラベルと注釈。

これらのアラートは、例えば、無音を追加したり、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、1つ以上の  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。

## クラシック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に設計されています。  
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、「」を参照してください。[Grafana バージョン 10 での作業](#)。  
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、「」を参照してください。[Grafana バージョン 9 での作業](#)。

既存のワークスペース、または Grafana アラートを使用しないことを選択するワークスペースは、従来のダッシュボードアラートを使用します。新しい Grafana アラートに移行するには、この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 アラートのオンと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、「」を参照してください。[Amazon Managed Grafana ワークスペースを設定する](#)。

#### Note

Grafana アラートを使用する場合、Prometheus ではなく Grafana で定義されたアラートルールは、複数の通知を連絡先に送信します。ネイティブの Grafana アラートを使用してい

る場合は、従来のダッシュボードアラートを維持し、新しい Grafana アラート機能を有効にしない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、Prometheus Alertmanager で作成されたアラートの通知を 1 つだけ送信する Grafana Alerting を有効にすることを勧めます。

## 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。Grafana インスタンスのアラートページで、移行されたアラートを新しいアラートとともに表示できます。Grafana アラートを使用すると、Grafana が管理するアラートルールは、一致すると 1 つのアラートではなく、複数の通知を送信します。

クラシックダッシュボードアラートと Grafana アラートへの読み取り/書き込みアクセスは、それらを保存するフォルダのアクセス許可によって管理されます。移行中、クラシックダッシュボードの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一致します。

- 元のアラートのダッシュボードに アクセス許可がある場合、移行は元のダッシュボードのアクセス許可 (フォルダから継承されたアクセス許可を含む) と一致する Migrated {"dashboardUid": "UID", "panelId": 1, "alertId": 1} のように、この形式でという名前のフォルダを作成します。
- ダッシュボードのアクセス許可がなく、ダッシュボードがフォルダにある場合、ルールはこのフォルダにリンクされ、そのアクセス許可を継承します。
- ダッシュボードのアクセス許可がなく、ダッシュボードが General フォルダにある場合、ルールは General Alerting フォルダにリンクされ、ルールはデフォルトのアクセス許可を継承します。

### Note

Grafana アラート NoData には Keep Last State のオプションがないため、このオプションは従来のルールの移行 NoData 中になくなります。Error 処理 Keep Last State のオプションは、新しいオプションに移行されます。Error の動作に一致させるために Keep Last State、どちらの場合も、移行中に Amazon Managed Grafana は 1 年間の期間を持つアラートルールごとに自動的に無音を作成します。

通知チャンネルは、適切なルートとレシーバーを持つ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ンネルは、デフォルトのルートへの問い合わせポイントとして追加されます。

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ンネルは、`autogen-unlinked-channel-recv` ルートに移動します。

### 制限事項

- Grafana アラートシステムは、使用可能なすべての Prometheus、Loki、Alertmanager データソースからルールを取得できます。サポートされている他の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行き来すると、1つのシステムでサポートされている機能のデータが失われる可能性があります。他のシステムでは失われない可能性があります。

#### Note

クラシックダッシュボードアラートに戻ると、Grafana アラートを有効にしている間に行われたアラート設定に加えられたすべての変更が失われます。これには、作成された新しいアラートルールも含まれます。

## アラートの基礎

- ⚠️ このドキュメントトピックは、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に設計されています。  
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、「」を参照してください [Grafana バージョン 10 での作業](#)。  
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、「」を参照してください [Grafana バージョン 9 での作業](#)。

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

次の表は、Grafana アラートの主な概念を示しています。

主要な概念または機能	定義
アラート用のデータソース	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します。これは、データソースに対して定期的にクエリを実行するコンポーネントです。これは Grafana マネージド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。
アラートルール	アラートルールを起動するタイミングに関する一連の評価基準。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、および条件が満たされる期間で構成されます。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。単次元の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。10 VMs に対する CPU などの複数の結果に一致する 1つのアラートルールは、複数の (この場合は 10 個) 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異なる場合があります。例えば、システム内のすべての VMs の CPU 使用率をモニタリングするアラートルールでは、VM が追加されるにつれて VMs。alert-instance クォータの詳細については、「」を参照してください <a href="#">クォータ到達エラー</a> 。
アラートグループ	Alertmanager 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



主要な概念または機能	定義
	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。
連絡先	アラートルールが発生したときに連絡先に通知する方法を定義します。
メッセージテンプレート	再利用可能なカスタムテンプレートを作成し、問い合わせポイントで使用します。
通知ポリシー	アラートをグループ化してコンタクトポイントにルーティングする場所、タイミング、および方法に関するルールのセット。
ラベルとラベルマッチャー	ラベルはアラートルールを一意に識別します。アラートルールを通知ポリシーと無音にリンクし、それら进行处理するポリシーと無音にするアラートルールを決定します。
無音	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。無音とミュートタイミングの違いは、無音が、定期的なスケジュールでミュートタイミングが発生する特定の期間続くことです。ラベルマッチャーを使用してアラートインスタンスをサイレンスします。
ミュートタイミング	新しい通知を生成または送信したくない時間間隔を指定します。アラート通知は、メンテナンス期間中など、定期的な期間だけフリーズできます。既存の通知ポリシーにリンクする必要があります。

## アラートデータソース

Grafana マネージドアラートは、アラートが有効になっている次のバックエンドデータソースをクエリします。



- Grafana によって組み込みまたは開発および保守されるデータソース:  
Alertmanager、Graphite、、 Prometheus (Amazon Managed Service for Prometheus を含む ) Loki、、 InfluxDBAmazon OpenSearch Service、Google Cloud Monitoring、Amazon CloudWatch、Azure MonitorMySQL、、 PostgreSQL、MSSQL0penTSDB0racle、、 および 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、サーバー側の式に直接アラートを送信したり、サーバー側の式に渡したりできます。これにより、データソース内の処理と効率が向上し、アラートルールが簡素化されます。時系列データではなく数値データでアラートを実行する場合、ラベル付けされた各時系列を 1 つの数値に減らす必要はありません。代わりに、ラベル付き番号は 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 マネージドアラートまたはサーバー側の式を含むクエリは、次のデータソースでは数値と見なされます。

- データソースクエリTableで Format ASオプションが に設定されている場合。
- クエリから Grafana に返されるテーブルレスポンスに、数値 (int、double、float など) 列が 1 つしか含まれていない場合、オプションで追加の文字列列が含まれます。

文字列列がある場合、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 MySQL テーブルがある場合、次のようにします。

時間	ホスト	ディスク	PercentFree
2021-June-7	web1	/etc	3
2021-June-7	Web2	/var	4

時間	ホスト	ディスク	PercentFree
2021-June-7	Web3	/var	8
...	...	...	...

データフィルタリングは時間どおりにクエリできますが、時系列を Grafana に返す必要はありません。例えば、空き領域が 5% 未満の場合にホスト、ディスクごとに開始されるアラート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)
```

このクエリは、Grafana に次のテーブルレスポンスを返します。

ホスト	ディスク	PercentFree
web1	/etc	3
Web2	/var	4
Web3	/var	0

このクエリをアラートルールの条件として使用すると、値がゼロ以外のアラートの場合。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
{ホスト=web1,ディスク=/etc}	アラート

ラベル	ステータス
{ホスト=web2,ディスク=/var}	アラート
{ホスト=web3, ディスク=/var}	[普通]

## アラートマネージャー

Grafana には Prometheus Alertmanager のサポートが組み込まれています。アラートマネージャーは、アラートルールのグループ化と管理の両方を支援し、アラートエンジンの上にオーケストレーションレイヤーを追加します。デフォルトでは、Grafana マネージドアラートの通知は、コア Grafana の一部である埋め込み Alertmanager によって処理されます。Alertmanager ドロップダウンから Grafana オプションを選択すると、Grafana アラート UI から Alertmanager の連絡先、通知ポリシー、およびテンプレートを設定できます。

Grafana アラートは、外部 Alertmanager 設定をサポートしています (外部データソースとしての Alertmanager の詳細については、「」を参照してください[Alertmanager データソースに接続する](#))。外部 Alertmanager を追加すると、Alertmanager ドロップダウンに使用可能な外部 Alertmanager 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。アラート状態、アラートルール状態、アラートルールの状態の 3 つの主要コンポーネントがあります。関連していますが、各コンポーネントは少し異なる情報を伝えます。

### アラートルールの状態

- 正常 – 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 — 評価エンジンによって返される時系列の少なくとも 1 つは です Pending。
- ファイリング — 評価エンジンによって返される時系列の少なくとも 1 つは です Firing。

### アラート状態

- 正常 — アラートルールの条件は、評価エンジンによって返されるすべての時系列で false です。

- アラート — アラートルールの条件は、評価エンジンによって返される少なくとも 1 つの時系列で true です。アラートが開始される前に条件が true である必要がある期間。設定されている場合は、条件が満たされているか、超過しています。
- 保留中 — アラートルールの条件は、評価エンジンによって返される少なくとも 1 つの時系列で true です。アラートが開始される前に条件が true である必要がある期間。設定されている場合、条件は満たされません。
- 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 である、または時系列のすべての値がゼロである。
- エラー — アラートルールの評価を試みたときのエラー。

### アラートルールの正常性

- 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 — アラートルールを評価する際のエラー。
- 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ないこと。

## Grafana アラートルールの作成と管理

**⚠** このドキュメントトピックは、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に設計されています。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、「」を参照してください[Grafana バージョン 10 での作業](#)。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、「」を参照してください[Grafana バージョン 9 での作業](#)。

アラートルールは、アラートが開始されるかどうかを決定する一連の評価基準です。ルールは、1 つ以上のクエリと式、条件、評価の頻度、およびオプションで条件が満たされる期間で構成されます。

クエリと式が評価するデータセットを選択する間、条件はアラートを作成するためにアラートが満たす必要がある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。期間を設定すると、条件を満たす必要がある期間を示します。ルールでは、データがない場合のアラート動作を定義することもできます。

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 マネージドアラートルールの作成](#)
- [Cortex または Loki が管理する録画ルールの作成](#)
- [Grafana マネージド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 マネージドアラートルールの作成

Grafana を使用すると、外部 Cortex または Loki インスタンスのアラートルールを作成できます。

### 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許可があることを確認します。それ以外の場合、Cortex マネージド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 および Loki データソースの場合は、それぞれのサービスを設定してルーラー API を有効にします。
  - Loki – Loki データソースのデフォルト local である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ます。ルールを編集するには、他のストレージタイプのいずれかを設定します。
  - Cortex – ではなくレガシー /api/prom プレフィックスを使用します / prometheus。Prometheus データソースは Cortex と Prometheus の両方をサポートしており、Grafana は Query API と Ruler API の両方が同じ URL にあることを想定しています。ルーラー 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たくない場合は、その設定に移動し、アラート UI によるアラートの管理チェックボックスをオフにします。

## Cortex または Loki マネージド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、既存のアラートを一覧表示するアラートページを開きます。
2. 新しいアラートルールを選択します。
3. ステップ 1 で、次のようにルール名、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` ラベルでもあります。
  - ルールタイプのドロップダウンから、Cortex/Loki マネージド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、名前を入力して作成します。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、組織的な目的のみを持つことができます。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、名前を入力して作成します。新しく作成されたルールが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 式または LogQL 式に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少なくとも 1 つある場合、ルールはアラートを開始します。各シリーズにアラートが作成されます。

5. ステップ 3 で、条件を追加します。

条件の For テキストボックスで、アラートが開始される前に条件を true にする必要がある期間を指定します。を指定する場合 5m、アラートが開始される前に 5 分間条件が true である必要があります。

**Note**

条件が満たされると、アラートは Pending 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である場合、アラートは Firing 状態に移行します。満たされなくなった場合、Normal 状態に戻ります。

- ステップ 4 で、ルールに関連付けられたメタデータを追加します。
  - 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [アラートルールの注釈とラベル](#)。
  - ランブック URL、パネル、ダッシュボード、アラート IDs を追加します。
  - カスタムラベルを追加します。
- 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。
- 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 が管理する録画ルールの作成

外部 Cortex または Loki インスタンスの記録ルールを作成および管理できます。記録ルールは、頻繁に必要な式または計算コスト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、高速です。

### 前提条件

Cortex および Loki データソースの場合は、それぞれのサービスを設定してルーラー API を有効にします。

- Loki – Loki データソースのデフォルト local である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ます。ルールを編集するには、他のストレージタイプのいずれかを設定します。
- Cortex – Cortex を指すように Grafana Prometheus データソースを設定する場合は、ではなく `レガシー/api/promプレフィックス` を使用します `/prometheus`。Prometheus データソースは Cortex と Prometheus の両方をサポートしており、Grafana は Query API と Ruler API の両方が同じ URL にあることを想定しています。ルーラー 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 によるアラートの管理チェックボックスをオフにします。

Cortex または Loki マネージド録音ルールを追加するには

1. Grafana コンソールの Grafana メニューからアラート (ベル) アイコンを選択して、既存のアラートを一覧表示するアラートページを開きます。
2. 新しいアラートルールを選択します。
3. ステップ 1 で、次のようにルール名、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` ラベルでもあります。
  - ルールタイプのドロップダウンから、Cortex/Loki マネージド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、名前を入力して作成します。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、組織的な目的のみを持つことができます。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、名前を入力して作成します。新しく作成されたルールが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 式または LogQL 式に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少なくとも 1 つある場合、ルールはアラートを開始します。各シリーズにアラートが作成されます。
5. ステップ 3 で、ルールに関連付けられたメタデータを追加します。
  - 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [アラートルールの注釈とラベル](#)。



- ランブック URL、パネル、ダッシュボード、アラート IDs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 マネージド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、結果を減少または変換して、相互に比較したり、しきい値を修正したりするアラートルールを作成できます。これらが処理されると、Grafana は連絡先に通知を送信します。

### Note

Grafana アラートの使用中に Grafana マネージドアラートルールを作成すると、ルールが一致したときに複数の通知が送信されます。一部のコンタクトセンタープロバイダーには、通知の重複を解除す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 マネージドルールを追加するには

1. Grafana コンソールの Grafana メニューからアラート (ベル) アイコンを選択して、既存のアラートを一覧表示するアラートページを開きます。
2. 新しいアラートルールを選択します。
3. ステップ 1 で、次のようにルール名、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` ラベルでもあります。
  - ルールタイプのドロップダウンから、Grafana マネージドアラートを選択します。
  - フォルダドロップダウンから、ルールを保存す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には、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。
  - デフォルト名のままにするか、カーソルを合わせ、編集アイコンを選択して名前を変更します。

- クエリの場合は、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 Classic 条件を選択して 1 つのアラートルールを作成するか、数学、削減、再サンプリングオプションを選択して、シリーズごとに個別のアラートを生成します。これらのオプションの詳細については、「[」](#)を参照してください [単一および多次元ルール](#)。
  - クエリの実行 を選択して、クエリが成功したことを確認します。
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、クエリまたは式を選択してアラートルールを開始します。
  - すべての を評価する で、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。
  - の評価では、アラートが開始される前に条件が true である必要がある期間を指定します。
-  Note

条件に違反すると、アラートは Pending 状態になります。指定された期間、条件が超過したままの場合、アラートは Firing 状態に移行します。満たされなくなった場合は、Normal 状態に戻ります。
- 「データとエラー処理を設定しない」で、データがない場合のアラート動作を設定します。のガイドラインを使用します [データまたはエラーケースなしの処理](#)。
  - アラートのプレビューを選択して、現時点でクエリを実行した結果を確認します。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、ルールに関連付けられたメタデータを追加します。
- 説明と概要を追加して、アラートメッセージをカスタマイズします。のガイドラインを使用します [アラートルールの注釈とラベル](#)。
  - ランブック URL、パネル、ダッシュボード、アラート IDs 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。

## 単一および多次元ルール

Grafana マネージド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。

### 単次元ルール (クラシック条件)

クラシック条件式を使用して、条件が満たされたときに単一のアラートを開始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。その結果、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 は 1 つのアラートのみを送信します。

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。アラートのクォータに達した場合、ルールは評価されません。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。

- 選択した時間範囲の値を 1 つの値に集計するには、各クエリに Reduce 式を追加します。( [数値データを使用するルール](#) には必要ありません )。
- ルールの条件を持つ Math 式を追加します。これは、ルールがアラートを開始すべきでない場合、クエリまたは reduce 式が既に 0 を返す場合や、アラートを開始する必要がある場合は正の数値を返す場合は必要ありません。

例:

- $\$B > 70$  B クエリ/式の値が 70 を超える場合にアラートを開始するかどうか。
- $\$B < \$C * 100$  B の値が C の値に 100 を掛けた値よりも小さい場合にアラートを開始する必要がある場合に備えて、[クォータ到達エラー](#) を使用します。比較される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つか、一方が他のクエリのサブセットである場合、それらが照合されます。

### Note

Grafana は、テンプレート変数を使用したアラートクエリを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「[テンプレート変数は、アラートの設定中にアラートクエリでサポートされていません](#)」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、アラートクォータにカウントされます。アラートクォータ内で対応できる数よりも多くのインスタンスを作成する多次元ルールは評価されず、クォータエラーを返します。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 ワークスペースのパフォーマンスや、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際の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。以下の考慮事項は、モニタリングシステムのパフォーマンスを最適化する際に役立ちます。

- **ルール評価の頻度** – アラートルールのすべての評価プロパティは、ルール評価の頻度を制御します。許容される最も低い評価頻度を使用することをお勧めします。
- **結果セットの基数** – ルールで作成するアラートインスタンスの数は、そのパフォーマンスに影響します。フリート内のすべての VM で、すべての API パスの API レスポンスエラーをモニタリングしているとします。このセットには、パスの数に VMs。例えば、VM ごとのパスごとではなく VM ごとの合計エラーをモニタリングすることで、結果セットの基数を減らすことができます。
- **クエリの複雑さ** – データソースが処理して応答できるクエリは、消費するリソースをすばやく減らすことができます。この考慮事項は、上記の他の考慮事項ほど重要ではありませんが、それらができるだけ減らした場合、個々のクエリのパフォーマンスを見ると違いが生じる可能性があります。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、データベースのモニタリングによって処理されるクエリの大部分であることが多いため、Grafana インスタンスに影響するのと同じ負荷要因がクエリにも影響します。

## 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があります。この数に達すると、そのワークスペースに新しい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化する可能性があります。

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は次のとおりです。

- 単一ディメンションルールのみを作成する場合、各ルールは単一のアラートインスタンスです。1つのワークスペースに、アラートインスタンスのクォータと同じ数のルールを作成できます。
- 多次元ルールは複数のアラートインスタンスを作成しますが、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すると、作成時に 50 個の EC2 インスタンス (つまり 50 個のアラートインスタンス) が存在する可能性があります。1 週間後にさらに 10 個の EC2 インスタンスを追加すると、次の評価には 60 個のアラートインスタンスが含まれます。

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ます。アラートインスタンスのクォータをすぐに超えるアラート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更される可能性があるため、クォータはルールが評価されるたびにチェックされます。

- ルールの評価時に、ルールによってアラートインスタンスのクォータを超えた場合、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アラートルールが更新されるまで、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使用します)。通知には、値の `Error` 注釈が含まれています `QuotaReachedError`。
- が評価 `QuotaReachedError` を停止するルール。評価は、更新が行われ、更新後の評価によってそれ自体が が発生しない場合にのみ再開されます `QuotaReachedError`。評価されていないルールは、Grafana コンソールでクォータ到達エラーを表示します。
- アラートルールを削除するか、多次元アラートを編集してアラートインスタンスの数を減らすことができます (例えば、VM 内の API ごとに 1 つのアラートではなく、VM ごとに 1 つのエラーに対して 1 つのアラートを設定するなど)。
- 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、アラートインスタンスの数を減らすために他の変更を加えた場合は、変更なしで保存することもできます。再開できる場合は、です。別の が発生した場合は `QuotaReachedError`、保存できません。

- アラートが保存され、アラートクォータを超過せずに評価を再開すると、クォータ到達エラーがしばらくの間 (評価間隔まで) Grafana コンソールに引き続き表示されますが、ルールのしきい値が満たされると、アラートルールの評価が開始され、アラートが送信されます。
- アラートクォータおよびその他のクォータの詳細については、「」を参照してください [Amazon Managed Grafana サービスクォータ](#)。

### データまたはエラーケースなしの処理

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のオプションを選択します。

データを処理しないためのオプションを次の表に示します。

データなしオプション	Behavior
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 <code>DatasourceNoData</code> を含むアラートを作成します。
アラート	アラートルールの状態を に設定します <code>Alerting</code> 。
OK	アラートルールの状態を に設定します <code>Normal</code> 。

エラーケースを処理するためのオプションを次の表に示します。

エラーまたはタイムアウトオプション	Behavior
アラート	アラートルールの状態を に設定する <code>Alerting</code>
OK	アラートルールの状態を に設定する <code>Normal</code>
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かったデータソースの UID

エラーまたはタイムアウトオプション	Behavior
	DatasourceError を含むアラートを作成します。

## アラートルールの注釈とラベル

アノテーションとラベルは、アラートルール、データソースレスポンス、アラートルール評価の結果として発生したアラートに関連付けられたキーと値のペアです。アラート通知で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、通知連絡先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。次の注釈を使用できます: description、summary、runbook\_url、alertId、dashboardId、および panelId。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、アラートの評価と通知プロセスを通じて生成され、に追加されます。これらは次の方法で使用されます。

- アラートの完全なラベルセットは、Grafana アラート内でそのアラートを一意に識別します。
- Alertmanager は、ラベルを使用して、[通知ポリシー](#) 内の[無音](#)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は、ラベルにアクセスして、通知の結果であるアラートに固有の情報を含む通知を動的に生成できます。
- 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、他のラベルを参照したりできます。ここでアラートルールに追加されたラベルは、ラベルが衝突した場合に優先されます。

注釈とラベルを展開する場合、次のテンプレート変数を使用できます。

名前	説明
\$labels	クエリまたは条件のラベル。例えば、 <code>{{ \$labels.instance }}</code> と <code>{{ \$labels.job }}</code> です。これは、ルールがクラシック条件を使用している場合は使用できません。
\$values	すべての の値は、このアラートルールに対して評価された数式と数式を減らします。例えば、 <code>{{ \$values.A }}</code> <code>{{ \$values.A.Labels }}</code> <code>{{ \$values.A.Value }}</code> はrefID式の Aです。これは、ルールがクラシック条件を使用する場合は使用できません。
\$value	アラートインスタンスの値文字列。例えば <code>[ var='A' labels={instance=foo} value=10 ]</code> です。

## 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タイプ別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を表示できますが、編集することはできません。

### アラートルールを表示する

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タイプ別にグループで表示されます。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、次のテキストで説明します)。



2. として表示では、任意のオプションを選択して、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+ 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、ルールを管理するためのデフォルトのルールリストビューです。各グループを展開して、このグループのルールのリストを表示できます。ルールをさらに展開して、その詳細を表示します。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、状態別にグループ化されたアラートルールが表示されます。このビューを使用して、どのルールがどの状態にあるかの概要を取得します。各ルールを展開して詳細を表示できます。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できます。

## フィルターアラートルール

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。

- データソースの選択 を選択し、フィルタリングするデータソースを選択することで、特定のデータソースをクエリするルールをフィルタリングして表示できます。
- ラベルで検索 で検索条件を選択して、ラベルでフィルタリングできます。サンプル条件には、`environment=production`、`region=~US|EU`、などがあります `severity!=warning`。
- 状態 でアラートをフィルタリングし、表示する状態を選択することで、特定の状態のルールを表示するようにフィルタリングできます。

## アラートルールの編集または削除

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 Cortex 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。

## 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。ルールを作成するのと同じ方法で更新を行います。詳細については、[Grafana マネージドアラートルールの作成](#)「」または「」の手順を参照してください。[Cortex または Loki マネージドアラートルールの作成](#)。
3. オプションで、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

ルールを整理できます。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編成されます。ルール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。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したりできます。

### ルールグループまたは名前空間を編集するには


1. Grafana コンソールの Grafana メニューで、アラート (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を変更します。

#### 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は、グループ内のルールの名前または評価間隔を変更します。例えば、ルール1mを 1 分に 1 回評価するか、30 秒に 1 回評価30sするかを選択できます。

5. **[変更を保存]** を選択します。

## アラートグループ

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

アラートグループには、Alertmanager インスタンスからのグループ化されたアラートが表示されます。デフォルトでは、アラートは 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ず [通知ポリシーの使用](#)。一般的なアラートを 1 つのアラートグループにグループ化すると、重複するアラートが開始されるのを防ぐことができます。

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 (ベル) アイコンを選択し、アラートグループ化項目を選択して、既存のグループを一覧表示するページを開きます。
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のアラートマネージャーを選択します。デフォルトでは、GrafanaAlertmanager が選択されています。
3. カスタムグループからドロップダウンでラベルの組み合わせを選択すると、デフォルト以外のグループが表示されます。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または検証できます。


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ラベルが指定されていないアラートは、ヘッダーが のグループに追加されず No grouping。

アラートのフィルタリング

次のフィルターを使用して、特定の条件に一致するアラートを表示できます。

- ラベルで検索 – 検索 に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 などです。
- 状態によるアラートのフィルタリング – 状態 で、Active、Suppressed、または Unprocessed 状態から選択して、その状態のアラートを表示します。

Prometheus データソースのアラート通知のサイレンシング

 このドキュメントトピックは、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に設計されています。

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、「」を参照してください[Grafana バージョン 10 での作業](#)。

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、「」を参照してください[Grafana バージョン 9 での作業](#)。

外部の Alertmanager データソース (Amazon Managed Service for Prometheus を含む) では、無音でアラート通知を抑制できます。無音は通知の作成のみを停止します。無音はアラートルールの評価を妨げず、インスタンスのアラートがユーザーインターフェイスに表示されないようにしません。アラートを無音にすると、アラートが抑制される時間ウィンドウを指定します。

外部 Alertmanager データソースの無音を設定できます。

#### 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) で抑制するには、無音 [ミュートタイミング](#) ではなく [ミュート](#) を使用します。

無音を追加するには

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きます。
2. 無音を選択して、既存の連絡先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を選択します。
4. 新しい無音を選択します。
5. 無音の開始と終了日を選択して、無音がいつ有効になるか、いつ終了するかを示します。

終了時間を設定する代わりに、期間で無音期間を指定します。これにより、無音開始フィールド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。マッチャーは、無音が適用されるルールを決定します。ラベルマッチングについては、この手順の後に詳しく説明します。
7. 必要に応じて、コメントを追加するか、作成者を変更して無音の所有者を設定します。

アラート抑制のラベルマッチング

無音を作成するときは、無音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、アラートを抑制するために一致する必要があるラベルに関する一連のルールです。一致するラベルは 3 つの部分で構成されます。

- Label – 一致するラベルの名前。アラートのラベル名と完全に一致する必要があります。
- Operator – 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。
  - = 指定された文字列と正確に一致する値を持つラベルを選択します。
  - != 値が指定された文字列と一致しないラベルを選択します。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルを選択します (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を選択します。
- 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一致させることができます。

無音は指定された終了日に終了しますが、抑制はいつでも手動で終了できます。

サイレンスを手動で終了するには

1. アラートページで、無音を選択して既存の無音のリストを表示します。
2. 終了する無音を選択し、無音 を選択します。これにより、アラートの抑制は終了します。

#### Note

サイレンシングを解除すると、終了時刻が現在の時刻に設定されているかのように、アラート抑制が終了します。(自動または手動で) 終了した無音は 5 日間保持され、一覧表示されます。リストから無音を手動で削除することはできません。

無音作成フォームへのリンクの作成

詳細がすでに入力されている無音作成フォームへの URL を作成できます。オペレーターはこれを使用して、運用イベント中にアラームをすばやく抑制できます。

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 `[label][operator][value]`、カンマで区切られた形式の 1 つ以上の値が必要です。

## URL の例

一致するラベルを持つ無音フォームにリンクするには、`severity=critical`と`cluster!~europe-.*`、および というコメントを付けSilencing critical EU alerts、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ます。

```
https://mygrafana/alerting/silence/new?matchers=severity%3Dcritical%2Ccluster!~europe-
*&comment=Silence%20critical%20EU%20alert
```

外部 Alertmanager の新しい無音ページにリンクするには、などの Alertmanager データソース名を持つ alertmanager クエリパラメータを追加します alertmanager=myAlertmanagerdatasource。

## コンタクトポイントの使用

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

コンタクトポイントを使用して、アラートが開始され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、Amazon Simple Notification Service や Slack 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。アラートが開始されると、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。オプションで、[メッセージングテンプレートの使用](#)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します。

### **i** Note

Grafana マネージドアラートのコンタクトポイントを作成および編集できません。Alertmanager アラートの連絡先は読み取り専用です。

## コンタクトポイントの使用

次の手順では、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 Alertmanager が選択されています。
4. 連絡先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 チャンネルと、連絡が必要な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使用可能な場合は、オプションで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、アラートが解決されたときに通知されたくない場合は、オプションで解決済みメッセージの無効化を選択します。
8. コンタクトポイントでより多くのコンタクトポイントタイプが必要な場合は、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すことができます。
9. アクセスポイントの保存 を選択して変更を保存します。

### コンタクトポイントを編集するには

1. コンタクトポイントを選択すると、既存の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。

連絡先が作成されたら、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できます。

### テスト通知を送信するには

1. 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開きます。
2. テストするコンタクトポイントを選択し、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独自のカスタム注釈とラベルをテスト通知に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でアラートをテストします。

通知ポリシーで使用されていない連絡先は削除できます。

コンタクトポイントを削除するには

1. 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

通知ポリシーでコンタクトポイントが使用されている場合は、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子のリスト

名前	型
Amazon SNS	sns
OpsGenie	opsgenie
ページャーの義務	pagerduty
Slack	slack
VictorOps	victorops



## メッセージングテンプレートの使用

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください[Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください[Grafana バージョン 9 での作業](#)。

経由で送信される通知[コンタクトポイントの使用](#)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 として評価されます (エスケープに影響する可能性があります)。

ほとんどのコンタクトポイント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できます。この[テンプレートデータ](#)トピックでは、テンプレートに使用できる変数を一覧表示します。

### テンプレートの使用

テンプレートはメッセージの作成に使用されます。例えば、Slack アラートメッセージを使用すると、コンタクトポイントでタイトルと本文を設定できます。次の例は、デフォルトテンプレートを使用して、発せられて解決されたアラートの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示しています。

- タイトル :

```
{{ len .Alerts.Firing }} firing, {{ len .Alerts.Resolved }} resolved
```

- テキスト本文 :

```
{{ range .Alerts }}{{ .Status }}: {{ .Labels.alertname }}  
{{end }}
```

次の例のように、独自のカスタムテンプレートを作成できます。

- タイトル :

```
{{ template "slack.default.title" .}}
```

- テキスト本文 :

```
{{ 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 }}
```

次の手順は、カスタムメッセージテンプレートを作成、編集、削除する方法を示しています。

メッセージテンプレートを作成するには

1. Grafana コンソールの Grafana メニューで、アラート (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。

- Alertmanager 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 Alertmanager インスタンスを選択します。デフォルトは Grafana Alertmanager です。
- テンプレートの追加 を選択します。
- わかりやすい名前 を追加します。
- テンプレートのコンテンツを追加します。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あり、省略すると、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は、同じままにしておくことがベストプラクティスです。

- [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 はテキストとしてレンダリングされ、制御文字はエスケープされます。結果の通知での HTML のレンダリングは、Grafana で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

- 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。
- テンプレートテーブルで、編集するテンプレートを見つけ、編集アイコン (ペン) を選択します。
- 変更を行い、テンプレートの保存 を選択します。

メッセージテンプレートを削除するには

- 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。
- テンプレートテーブルで、削除するテンプレートを見つけ、削除アイコン (ゴミ箱) を選択します。

### 3. はい、削除を選択してテンプレートを削除します。

#### ネストされたテンプレート

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" . }}
```

次の組み込み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
<code>default.title</code>	高レベルのステータス情報を表示します。
<code>default.message</code>	発声アラートと解決されたアラートのフォーマットされた概要を提供します。

#### カスタムテンプレートの例

カスタムテンプレートの使用方法の例を次に示します。

1つのアラートをレンダリングするテンプレート :

```
{{ 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レンダリングするテンプレート :

```

{{ 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 }}

```

テンプレートデータ

次のデータがメッセージテンプレートに渡されます。

名前	型	メモ
Receiver	string	通知の送信先の連絡先の名前。
Status	string	少なくとも 1 つのアラートが発せられている場合は発せら

名前	型	メモ
		れます。それ以外の場合は解決されます。
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するラベル。
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	string	通知を送信した Grafana へのリンクを戻します。外部 Alertmanager を使用している場合は、この Alertmanager へのリンクを戻します。

Alerts タイプは、返されたアラートをフィルタリングするための 2 つの関数を公開します。

- Alerts.Firing – 発声アラートのリストを返します。
- Alerts.Resolved – 解決されたアラートのリストを返します。

## アラート (タイプ)

アラートタイプには、次のデータが含まれます。

名前	型	メモ
ステータス	string	firing-または-resolved

名前	型	メモ
ラベル	KeyValue	アラートにアタッチされたラベルのセット。
注釈	KeyValue	アラートにアタッチされた注釈のセット。
StartsAt	時刻	アラートが発せられた時刻。
EndsAt	時刻	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。それ以外の場合は、最後のアラートを受信してから設定可能なタイムアウト期間に設定します。
GeneratorURL	string	Grafana または外部 Alertmanager へのバックリンク。
SilenceURL	string	このアラートのラベルがあらかじめ入力されているの grafana サイレンスへのリンク。Grafana マネージドアラートの場合のみ。
DashboardURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードにリンクします。Grafana マネージドアラートの場合のみ。
PanelURL	string	アラートルールが 1 つに属している場合は、grafana ダッシュボードパネルにリンクします。Grafana マネージドアラートの場合のみ。

名前	型	メモ
フィンガープリント	string	アラートの識別に使用できるフィンガープリント。
ValueString	string	アラート内の各縮小式のラベルと値を含む文字列。

## KeyValue タイプ

KeyValue タイプは、ラベルと注釈を表すキーと値の文字列ペアのセットです。

として保存されたデータの直接アクセスに加えてKeyValue、データのソート、削除、変換の方法もあります。

名前	引数	戻り値	メモ
SortedPairs		キーと値の文字列ペアのソート済みリスト	
Remove	[]文字列	KeyValue	指定されたキーなしでキー/値マップのコピーを返します。
名前		[]文字列	ラベル名のリスト
[値]		[]文字列	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通知を生成できます。以下の関数を使用できます。



名前	引数の型	戻り型	説明
humanize	数値または文字列	string	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	string	ヒューマナイズと同様に、は 1000 ではなく 1024 をベースとして使用します。
humanizeduration	数値または文字列	string	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	string	比率値を 100 の分数に変換します。
humanizeTimestamp	数値または文字列	string	Unix タイムスタンプを秒単位でより読みやすい形式に変換します。
title	string	string	strings.Title、各単語の最初の文字を大文字にします。
toUpper	string	string	strings.ToUpper、すべての文字を大文字に変換します。
toLower	string	string	strings.ToLower、すべての文字を小文字に変換します。
match	パターン、テキスト	ブール値	regexp. アンカーされていない regexp マツ

名前	引数の型	戻り型	説明
			子のMatchString テスト。
reReplaceAll	パターン、置換、テキスト	string	Regex.ReplaceAllS string Regex 置換、アンカーなし。
graphLink	string - exprおよび datasource フィールドを持つ JSON オブジェクト	string	指定された式とデータソースについて、Explore でグラフィカルビューへのパスを返します。
tableLink	string - exprおよび datasource フィールドを持つ JSON オブジェクト	string	指定された式とデータソースについて、Explore の表形式ビューへのパスを返します。
args	[]インターフェイス{}	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。この関数を使用して、複数の引数をテンプレートに渡します。
externalURL	なし	string	外部 URL を表す文字列を返します。
pathPrefix	なし	string	外部 URL のパスを返します。

次の表は、各関数の使用例を示しています。

機能	TemplateString	入力	予想
ヒューマナイズ	{ \$value をヒューマナイズ }	1234567.0	1.235 メートル
humanize1024	{ humanize1024 \$value }	1048576.0	1Mi
humanizeDuration	{ humanizeDuration \$value }	899.99	14 分 59 秒
humanizePercentage	{ humanizePercentage \$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
一致	{ 「a+」 \$labels.instance } に一致	aa	true
reReplaceAll	{{ reReplaceAll "localhost:(.*)" "my.domain:\$1" \$labels.instance }}	localhost:3000	my.domain:3000
graphLink	{{ graphLink "{\`expr\`: \"up\", \"datasource\": \"gdev-prometheus\"}" }}		/explore?left=["now-1h","now"]gdev-prometheus,{"datasource": "gdev-prometheus", "expr": "up", "instant": false, "range": true}]
tableLink	{{ tableLink "{\`expr\`: \"up\", \"datasource\"": \"up\", \"datasource\"": \"up\", \"datasource\"": \"up\"}" }}		/explore?left=["now-1h","now"]gdev-prometheus

機能	TemplateString	入力	予想
	<code>\":\"gdev-prometheus \\\" } }</code>		<code>etheus\",{\"datasour ce\"gdev-prometheus \"expr\"up\"instant\": true,\"range\":false}]</code>
args	<code>{{define \"x\"}}{{.arg0}} {{.arg1}}{{end}}{ template \"x\" (args 1 \"2\")}}</code>		1 2
externalURL	<code>{ externalURL }</code>		<code>http://localhost/path/ prefix</code>
pathPrefix	<code>{ pathPrefix }</code>		<code>/path/prefix</code>

## 通知ポリシーの使用

- ⚠** このドキュメントトピックは、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に設計されています。
- 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、「」を参照してください [Grafana バージョン 10 での作業](#)。
- 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、「」を参照してください [Grafana バージョン 9 での作業](#)。

通知ポリシーは、アラートをコンタクトポイントにルーティングする方法を決定します。ポリシーにはツリー構造があり、各ポリシーには 1 つ以上の子ポリシーを含めることができます。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と一致することもできます。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 つ以上の一致後も評価が続行されます。親ポリシーの設定と連絡先情報は、子ポリシーのいずれにも一致しないアラートの動作を制御します。ルートポリシーは、特定のポリシーに一致しないアラートを管理します。

### Note

Grafana マネージドアラートの通知ポリシーを作成および編集できます。Alertmanager アラートの通知ポリシーは読み取り専用です。

## 通知のグループ化

グループ化は、類似する性質のアラート通知を 1 つのファネルに分類します。これにより、システムの多くの部分が一度に障害が発生し、多数のアラートが同時に開始されるときに、大規模な停止中のアラート通知を制御できます。

### グループ化の例

異なる環境のデータベースに 100 のサービスが接続されているとします。これらのサービスは、ラベルによって区別されます `env=environmentname`。サービスがデータベースに到達できるかどうかをモニタリングするためのアラートルールが設定されています。アラートルールは、という名前のアラートを作成します `alertname=DatabaseUnreachable`。

サービスの半分がデータベースに到達できなくなるネットワークパーティションが発生すると、50 の異なるアラートが開始されます。この状況では、影響を受ける環境のリストを含む 1 ページの通知 (50 ではなく) を受け取る必要があります。

グループ化を (サービスごとに異なる `env` ラベルを使用 `group_by: [alertname]` せずに) に設定できます。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を含む単一のコンパクトな通知を送信します。

### 特別なグループ

Grafana には 2 つの特別なグループがあります。デフォルトのグループでは、`group_by: null` 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、グループを効果的に無効にして、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

次の手順は、通知ポリシーを作成および管理する方法を示しています。

## ルート通知ポリシーを編集するには

1. Grafana コンソールの Grafana メニューで、アラート (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 (pen) 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を更新します。
6. Group by で、アラートをグループ化するラベル (または特別なグループ) を選択します。
7. タイミングオプション で、次のオプションから選択します。
  - グループ待機 — 最初の通知を送信する前に、同じグループのアラートをバッファするのを待機する時間。デフォルト値は 30 秒です。
  - グループ間隔 — グループに対する 2 つの通知間の最小時間間隔。デフォルトは 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ていない場合、通知を再送信するまでの最小間隔。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。

## 新しい最上位の特定のポリシーを追加するには

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. 「ラベルの一致」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。ラベルマッチングの詳細については、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. オプションで、一般的なタイミングをオーバーライドを選択して、グループ通知ポリシーのタイミングオプションをオーバーライドします。
9. [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追加するには

1. ネストされたポリシーを作成する特定のポリシーを展開します。
2. ネストされたポリシーを追加 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など)。
3. [Save policy(ポリシーの保存)] を選択して変更を保存します。

特定のポリシーを編集するには

1. アラートページから通知ポリシーを選択して、既存のポリシーを一覧表示するページを開きます。
2. 編集するポリシーを選択し、編集アイコン (ペン) を選択します。
3. 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。

ラベルマッチングの仕組み

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一致する場合、ポリシーはアラートに一致します。

- Label – 一致するラベルの名前。アラートのラベル名と完全に一致する必要があります。
- Operator – 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。
  - = 指定された文字列と正確に一致する値を持つラベルを選択します。
  - != 値が指定された文字列と一致しないラベルを選択します。
  - =~ 指定された文字列の正規表現解釈値と一致するラベルを選択します (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を選択します。
- 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一致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は、ポリシーの新しい通知が生成または送信されない定期的な時間間隔です。定期的なメンテナンス期間など、アラートが特定の期間に発生し、繰り返し発生するのを防ぐために使用します。

無音と同様に、ミュートタイミングによってアラートルールの評価が妨げられることも、アラートインスタンスがユーザーインターフェイスに表示されなくなることもありません。通知の作成のみを防ぎます。

外部の Alertmanager データソースの Grafana マネージド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。

### 無音と比較したミュートタイミング

次の表は、ミュートのタイミングと無音の違いを示しています。

ミュートタイミング	無音
繰り返し発生する可能性のある時間間隔定義を使用する	開始時刻と終了時刻が固定されている
作成され、通知ポリシーに追加される	ラベルを使用してアラートと照合し、無音にするかどうかを決定します

### ミュートタイミングを作成するには

1. Grafana コンソールの Grafana メニューで、アラート (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を追加」ボタンを選択します。
5. ミュートの推定を適用する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。

### 通知ポリシーにミュートタイミングを追加するには

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。

ポリシーの保存ボタンを選択します。

## 時間間隔

時間間隔は、時間範囲の定義です。この間隔でアラートが開始されると、アラートは抑制されます。範囲はを使用してサポートされます：(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(次のリストの詳細)で構成され、アラートを抑制するには、すべて一致する必要があります。例えば、曜日 `monday:friday` と時間範囲を 8:00~9:00 に指定すると、アラートは月曜日から金曜日まで 8~9 に抑制されますが、例えば土曜日の 8~9 には抑制されません。

- 時間範囲 – 通知を抑制する時刻。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。時刻の例は `14:30` です。時刻は UTC で 24 時間表記されます。
- 曜日 – 曜日。などの1日 `monday`、などの範囲 `monday:friday`、または `monday, tuesday, wednesday` などのカンマ区切りの日数リストを使用できます。
- 月 – 選択する月。月は、数値の指定で指定することも、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` 両方で1月を指定します。1か月、月の範囲、または月のリストをカンマで区切って指定できます。
- Days of the month – 1か月内の日付。値の範囲は `1~31` です。負の値は、月の曜日を逆の順序で指定するため、`-1` は月の最終日を表します。月の日は、`1` 日、日数の範囲、またはカンマ区切りの日数リストとして指定できます。

## 設定を変更する

設定タブでは、複数のタスクを実行できます。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を変更するには、「」を参照してください [Amazon Managed Grafana ワークスペースを設定する](#)。

## Amazon Managed Grafana プロファイルを編集する

プロフィールには、名前、ユーザー名、E メールアドレスが含まれます。

プロフィールを編集するには

1. 画面の左下隅にあるユーザーアイコンを一時停止し、設定 を選択します。
2. プロファイルの編集 セクションでは、次のいずれかを編集できます。
  - 名前 – このフィールドを編集して、プロフィールに関連付けられている表示名を変更します。
  - E メール – このフィールドを編集して、プロフィールに関連付けられている E メールアドレスを変更します。
  - Username – このフィールドを編集してユーザー名を変更します。
3. [保存] を選択します。

## 設定を編集する

設定には、 がダークテーマとライトテーマのどちらを使用するか、ホームダッシュボード、タイムゾーンが含まれます。

### 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を上書きします。

設定を変更するには

1. 画面の左下にあるユーザーアイコンを一時停止し、設定 を選択します。
2. 設定セクションでは、次のいずれかを編集できます。
  - UI テーマ – テーマを設定するには、ダークまたはライト を選択します。デフォルトは、ダークテーマまたは Grafana 管理者が選択したテーマのいずれかです。
  - ホームダッシュボード

- Timezone – Timezone リストのオプションを選択します。デフォルトは、ブラウザのローカルタイムゾーンまたは Grafana 管理者が選択したタイムゾーンです。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。

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。誰かが Amazon Managed Grafana 認証情報を悪用した疑いがある場合は、このセクションを確認できます。

セッション情報を表示するには

1. 画面の左下隅にあるユーザーアイコンを一時停止し、設定を選択します。
2. 「セッション」セクションまでスクロールします。Grafana には以下が表示されます。
  - 最終表示日 — ログオンした時間。
  - ログイン日 — 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。
  - ブラウザと OS – Grafana へのログオンに使用されるウェブブラウザとオペレーティングシステム。
  - インスタンスの Grafana 管理者の場合は、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。

## サポートのための情報を収集する

サポートバンドルは、ユーザーインターフェイスを介して Grafana ワークスペースに関する情報を収集する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した場合は、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
- インストール済みプラグイン
- Grafana の設定
- データベース情報と移行

**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。

- 使用状況統計 – 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情報 – データベース情報と移行ログ。
- プラグイン情報 – ワークスペースにインストールされている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、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されます)。
- 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されます)。
- 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されます)。

## サポートバンドルの作成

サポートバンドルを作成するには、次の手順に従います。

**Note**

この手順には、ワークスペースのadminアクセス許可が必要です。

サポートバンドルを作成するには

1. Grafana ワークスペースにサインインします。

- ヘルプアイコンを選択する
- ヘルプメニューから、サポートバンドルを選択します。
- 新しいサポートバンドルを選択します。
- 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。
- [作成]を選択します。
- サポートバンドルの準備ができたなら、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。

ファイルを開いて、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。バンドルにプライベート情報が含まれており、プライベートではないチャ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。

## Classic ダッシュボードアラート

**⚠** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください[Grafana バージョン 10 のアラート](#)。

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、「」を参照してください[Grafana バージョン 9 のアラート](#)。

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、「」を参照してください[Grafana アラート](#)。

### **i** Note

このセクションでは、Grafana のクラシック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 Alertmanager 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、「」を参照してください[Grafana アラート](#)。

Classic 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たとき。アラートルールは、Grafana によって定期的に評価される 1 つ以上の条件によって定義されます。
- 通知チャンネル — アラートの配信方法。アラートルールの条件が満たされると、Grafana はそのアラートに設定されたチャンネルに通知します。

現在、グラフパネルの視覚化のみがアラートをサポートしています。

## アラート設定

Amazon Managed Grafana ワークスペースでアラートを設定できます。

- アラート通知チャン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その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、トラブルシューティングを行います。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、アラートは限られた形式の高可用性をサポートしています。複数のワークスペースを実行すると、アラート通知は重複排除されます。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ために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の情報を含む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。

実際の通知は複数のアラート間で設定および共有されます。

## アラート実行

アラートルールは、スケジューラとクエリ実行エンジンの Amazon Managed Grafana で評価されます。

## アラート通知

**△** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください[Grafana バージョン 10 のアラート](#)。

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、「」を参照してください[Grafana バージョン 9 のアラート](#)。

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、「」を参照してください[Grafana アラート](#)。

アラートの状態が変わると、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 チャンネルを追加して設定する必要があります。

これは、通知チャンネルページから行われます。

### 通知チャンネルの追加

1. サイドバーで、アラート (ベル) アイコンを一時停止し、通知チャンネル を選択します。
2. [Add channel] (チャンネルの追加) を選択します。
3. 以下のセクションで説明するフィールドに入力するか、オプションを選択します。

### 新しい通知チャンネルフィールド

デフォルト (すべてのアラートで送信)

- 名前 – このチャン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表示されます。
- タイプ – チャンネル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子のリスト](#)」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) – 選択すると、このオプションはすべてのアラートルールについて、このチャンネルに通知を送信します。

- メッセージの解決を無効にする – 選択すると、アラート状態が `false` に戻ったときに送信される解決メッセージ [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、アラートに関する追加の通知 (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、秒数 (s)、分数 (m)、時間数 (h) を使用して指定できます。たとえば、30s、3m、5m などです1h。

### ⚠ Important

アラートのリマインダーは、ルールが評価された後に送信されます。したがって、設定されたアラートルールの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の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。

アラートルールの評価間隔	ごとにリマインダーを送信する	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
1m	5m	~ 5 分
5m	15m	~ 15 分
6m	20m	~ 24 分
1h	15m	~ 1 時間
1h	2h	最大 2 時間

## サポートされている通知子のリスト

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
<a href="#">Amazon Simple Notification Service</a>	sns	いいえ	はい



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	はい
<a href="#">PagerDuty</a>	pagerduty	いいえ	はい
<a href="#">Slack</a>	slack	いいえ	いいえ
VictorOps	victorops	いいえ	いいえ

## Amazon Simple Notification Service

サービス管理アクセス許可を有効にし、ワークスペースの通知チャンネルとして Amazon SNS を含めた場合は、通知チャンネルを作成するときのみ SNS トピック ARN を提供する必要があります。名前フィールドに、作成した SNS トピックの名前を指定します。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を作成した場合、通知をトピックに正常に発行するには、SNS トピック名 grafana に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管理アクセス許可を選択した場合、SNS トピック名の前に を付ける必要はありません grafana。

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。

ワークプレイスにカスタマー管理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 Publish アクセス許可を含める必要があります。

## Slack

Slack を設定するには、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、「[受信 Webhook を使用したメッセージの送信](#)」を参照してください。

Slack メッセージに発声アラートのスクリーンショットを含めるには、Grafana で外部イメージの送信先を設定するか、Slack Apps を介したボット統合を設定する必要があります。Slack ボット統合の設定の詳細については、「[Slack のガイドに従ってボット統合をセットアップする](#)」を参照してください。「xoxb」で始まる指定されたトークンを使用します。

設定	説明
Url	受信ウェブフック URL をスラックするか、最終的に <a href="#">chat.postMessage</a> Slack API エンドポイント をスラックします。
ユーザーネーム	ボットの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 受信者を上書きします。チャンネル 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)、またはチャンネルリファレンス (#<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。Slack API chat.postMessage エンドポイントを使用する場合は、これが必須です。
アイコン絵文字	ボットの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 URL	ボットのメッセージのアイコンとして使用するイメージへの URL を指定します。
ユーザーについて言及する	オプションで、Grafana から送信された Slack 通知で 1 人以上のユーザーについて言及します。対応する Slack IDs を介してカンマ区切りでユーザーを表示するには、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。
メンショングループ	オプションで、Grafana によって送信される Slack 通知で 1 つ以上のグループについて言及します。対応する Slack IDs (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。
言及チャンネル	オプションで、すべてのチャンネルメンバーまたはアクティブなチャンネルメンバーのみに言及します。

設定	説明
トークン	指定した場合、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。Slack API chat.post Message 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している場合は、通知を送信するチャンネルにボットを招待する必要があります。次に、受信者フィールドにチャンネルを追加します。

## PagerDuty

を設定するには PagerDuty、統合キーを指定します。

設定	説明
統合キー	の統合キー PagerDuty。
緊急度	動的通知のレベル。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 PagerDuty 後、でインシデントを解決します。
詳細のメッセージ	PD 概要フィールドからアラートメッセージを削除し、代わりにカスタムの詳細 (2) に配置します。

### Note

タグ Severity、Class、Group、Componentはdedup\_key、[PagerDuty共通イベント形式 – PD-CEF](#) で特別な意味を持ちます。アラートパネルでこれらのタグキーが定義されている場合、それらはに送信されるイベントのルートに変換されます PagerDuty。つまり、PagerDuty UI およびフィルタリングツール内で使用できます。アラートに設定された重要度

タグは、有効なレベルである場合、通知チャンネルに設定されたグローバル重要度を上書きします。

#### Note

Message In Details を使用すると、PagerDuty イベント内の `custom_details` フィールドの構造が変更されます。これにより、のフィールドに依存すると、ルールのカスタムイベント PagerDuty ルールが破損する可能性があります `payload.custom_details`。を使用する既存のルールを `custom_details.myMetric` に移動します `custom_details.queries.myMetric`。


#### Note

`dedup_key` タグを使用すると、カスタムキー `dedup_key` で生成された Grafana が上書きされます。

## アラート通知から Grafana へのリンクを設定する

すべてのアラート通知には、Grafana ワークスペースのトリガーされたアラートへのリンクが含まれています。

## アラートの作成

 このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください [Grafana バージョン 10 のアラート](#)。

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、「」を参照してください [Grafana バージョン 9 のアラート](#)。

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、「」を参照してください [Grafana アラート](#)。

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、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、アラートルール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. アラートルールを追加または編集するパネルに移動し、タイトルを選択し、編集を選択します。
2. アラート タブで、アラートの作成 を選択します。このパネルのアラートが既に存在する場合は、アラートタブの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. ルールの作成が完了したら、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返すようにするには、テストルール 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、アラートの下部までスクロールし、「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、アラートを作成するために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。アラートルールリストに名前が表示されます。
- すべての を評価する — スケジューラ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。
- - アラート通知がトリガーされるまでに、クエリが設定されたしきい値に違反する必要がある期間を指定します。

### ⚠ Warning

If no data or all values are null 設定を に設定Forして を使用しないでくださいNo Data。のトリガーNo Dataはすぐにトリガーされ、For考慮されません。これにより、アラートが から移行しても OK 通知が送信されない可能性がありますNo Data -> Pending -> OK。

アラートルールに が設定されForていて、クエリが設定されたしきい値に違反した場合、まず から OKに移動しますPending。Amazon Managed Grafana はPending、 から OKに移行しても通知を送信しません。アラートルールがFor期間を超えて発動している場合、 に変更Alertingされ、アラート通知が送信されます。

通常、この設定を使用することをお勧めします。アラート通知が開始されるまで数分待つよりも誤検出が発生する方が悪いことが多いためです。Alert list または を見るとAlert list panels、保留中の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、および集計関数を指定できるQuery条件のみです。

### クエリ条件の例

```
avg() OF query(A, 15m, now) IS BELOW 14
```

- avg() 各シリーズの値をしきい値と比較できる値に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。を使用して10m, now-2m、10分前から2分前の時間範囲を定義することもできます。これは、過去2分間のデータを無視する場合に便利です。
- IS BELOW 14 しきい値のタイプとしきい値を定義します。しきい値のタイプIS BELOWを変更することもできます。

アラートルールで使用されるクエリには、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 AND演算子と OR演算子のみがサポートされており、連続して実行されま

す。例えば、Condition:A(evaluates to: TRUE) OR condition:B(evaluates to: FALSE) AND condition:C(evaluates to: TRUE) の 3 つの条件があり、結果は ((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、集計関数としきい値チェックはシリーズごとに評価されます。現在、Amazon Managed Grafana はシリーズごとのアラートルールの状態を追跡しません。これによる影響については、次のシナリオで詳しく説明します。

- 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 シリーズにより、アラートルールが起動し、状態に切り替わりますAlerting。
- 通知は、ロードピーク (server1) というメッセージで送信されます。
- 同じアラートルールの後続の評価では、server2 シリーズによってアラートルールも起動されません。
- アラートルールが既に状態になっているため、新しい通知は送信されませんAlerting。

前のシナリオからわかるように、ルールが既に状態の場合Alerting、他のシリーズによってアラートが発せられたときに Grafana は通知を送信しません。

### Note

トリガーされたアラートに対して送信され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、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動する場合は、リマインダー通知に含まれます。使用している通知チャンネルによっては、この機能を利用して、アラートを発生させ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。

## データおよびエラー処理なし

次の表に、データまたは null 値のみを返すクエリをルール評価エンジンが処理する方法を制御するための条件を示します。

データオプションなし	説明
データなし	アラートルールの状態を に設定しますNoData。
アラート	アラートルールの状態を に設定しますAlerting。
最後の状態を保持する	現在のアラートルールの状態をそのまま維持します。
大阪	サポートされていますが、通常は有用ではありません。

## 実行エラーまたはタイムアウト

以下のオプションは、実行エラーまたはタイムアウトエラーの処理方法を Amazon Managed Grafana に指示します。

エラーまたはタイムアウトオプション	説明
アラート	アラートルールの状態を に設定しますAlerting。
最後の状態を保持する	現在のアラートルールの状態をそのまま維持します。

クエリがタイムアウトしたり、ランダムに失敗したりすることがある信頼できない時系列ストアがある場合は、このオプションを に設定Keep Last Stateして、基本的に無視できます。

## 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を指定することもできます。メッセージには、問題の解決方法に関する情報、ランブックへのリンクなど、あらゆる情報を含めることができます。

実際の通知は、複数のアラート間で設定および共有されます。通知を設定およびセットアップする方法については、「」を参照してください[アラート通知](#)。

- 送信先 — アラート通知チャンネルを設定している場合は、そのチャンネルを選択します。



- メッセージ – 通知チャネルで送信され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機能は、テキストを HTML またはその他のリッチ形式に変換することをサポートしています。
- タグ – 通知に含めるタグ (キーと値) のリストを指定します。これは一部の通知機能でのみサポートされています。

## アラート状態履歴と注釈

アラート状態の変更は、Amazon Managed Grafana データベースの内部注釈テーブルに記録されます。状態の変化は、アラートルールのグラフパネルに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

**⚠** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください [Grafana バージョン 10 のアラート](#)。

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、「」を参照してください [Grafana バージョン 9 のアラート](#)。

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、「」を参照してください [Grafana アラート](#)。

アラートルールの評価を一時停止すると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の発生を回避できます。

1. Grafana サイドバーで、アラート (ベル) アイコンで一時停止し、アラートルール を選択します。設定された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. リストでアラートを検索し、右側の一時停止アイコンを選択します。一時停止アイコンが再生アイコンに変わります。
3. 再生アイコンを選択して、アラートの評価を再開します。

## 既存のアラートルールの表示

**⚠** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください [Grafana バージョン 10 のアラート](#)。

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、「」を参照してください [Grafana バージョン 9 のアラート](#)。

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、「」を参照してください [Grafana アラート](#)。

Amazon Managed Grafana は、個々のアラートルールが定義されているパネルに保存しますが、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 サイドバーで、アラート (ベル) アイコンを一時停止し、アラートルールを選択します。設定された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、次の操作を実行できます。

- 名前でアラートをフィルタリングする – アラートの検索フィールドにアラート名を入力します。
- 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。その他はすべて非表示になります。
- アラートを一時停止または再開する – 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。
- アクセスアラートルール設定 — 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、アラートが発せられても、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

**⚠** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください[Grafana バージョン 10 のアラート](#)。

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、「」を参照してください[Grafana バージョン 9 のアラート](#)。

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、「」を参照してください[Grafana アラート](#)。

アラート通知テンプレート機能を使用すると、アラートクエリからラベル値を取得し、それをアラート通知に挿入できます。

アラートクエリの評価から存在するラベルは、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。アラートラベルデータは、アラートがアラート状態になると通知フィールドに挿入されます。同じラベルに一意の値が複数ある場合、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には

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、**編集** を選択します。
3. アラート タブで、アラートの作成 を選択します。このパネルのアラートが既に存在する場合は、直接編集できます。
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てください。アラートクエリラベルの詳細については、Grafana ドキュメントの「[メッセージテンプレート](#)」を参照してください。
5. 右上隅にある**保存** を選択します。

## アラートのトラブルシューティング

**⚠** このドキュメントトピックでは、Grafana でのレガシー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 アラートに移行して、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、「」を参照してください[Grafana バージョン 10 のアラート](#)。

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、「」を参照してください[Grafana バージョン 9 のアラート](#)。

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、「」を参照してください[Grafana アラート](#)。

アラートが期待どおりに動作していない場合は、次の手順でトラブルシューティングを行い、問題の原因を把握できます。

トラブルシューティングの最初のレベルは、テストルールを選択することです。クエリから返された raw データが表示されるポイントまで結果を展開できます。

# Grafana HTTP APIs

Grafana HTTP APIs Amazon Managed Grafana ワークスペースで使用できます。以下のセクションでは、APIsの使用方法を説明し、サポートされている Grafana APIs します。

## Note

Amazon Managed Grafana には AWS APIs も用意されています。AWS APIs [Grafana API リファレンス](#)」を参照してください。

## トピック

- [トークンによる認証](#)
- [アラート API](#)
- [アラート通知チャンネル 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](#)
- [Organization 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](#)

## トークンによる認証

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。トークンは、API の呼び出し元に認証と認可を提供します。トークンを作成するには 2 つの方法があります。

- 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、レポート生成などの自動ワークロードを Grafana で実行するために使用されます。サービスアカウントのサービスアカウントトークンを作成できます。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でき、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要な方法として APIs キーを置き換える予定です。
- API キー – API キー (API トークンとも呼ばれます) は、外部システムが Grafana HTTP APIs。API キーは Grafana ワークスペースのバージョン 8、9、および 10 で使用できますが、今後のリリースで GrafanaLabs 廃止される予定です。

### トピック

- [サービスアカウントを使用して Grafana HTTP APIs](#)
- [API キーを使用して Grafana HTTP APIs](#)

## サービスアカウントを使用して Grafana HTTP APIs

サービスアカウントを使用して、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ためのサービスアカウントとトークンを作成します](#)。

### Note

サービスアカウントは Grafana 9.x 以降で利用でき、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 キーを置き換えています。

サービスアカウントを作成するための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して、次のことができます。

- Grafana で使用するシステム内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 を操作する

#### Note

各サービスアカウントは、請求の目的で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 で認証するときにパスワードの代替として機能する、生成されたランダム文字列です。

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を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 API キーと同じ方法で使用できます。例えば、Grafana HTTP APIs にプログラムでアクセスできます。

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。これは、次の場合に実行できます。

- 複数のアプリケーションが同じアクセス許可を使用しますが、それらのアクションを個別に監査または管理したい場合。
- 侵害され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ります。

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からアクセス許可を継承します。

## サービスアカウントのメリット

API キーにサービスアカウントの利点には、次のようなものがあります。

- サービスアカウントは Grafana ユーザーと似ており、有効/無効にしたり、特定のアクセス許可を付与したり、削除または無効化されるまでアクティブのままにしたりできます。API キーは有効期限が切れるまで有効です。
- 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た場合で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](#) をご参照ください。

## 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、作成した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、更新、削除することもできます。

### 前提条件

サービスアカウントを作成および編集する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を作成および編集するには、組織の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](#) をご参照ください。

サービスアカウントを作成するには

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. 表示名は、サービスアカウントに関連付けられた ID を決定する際に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、将来のサービスアカウントのスケーリングと維持に役立ちます。
  - 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 Grafana 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。を使用して [CreateWorkspaceServiceAccount](#)、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にトークンを追加する

サービスアカウントトークンは、Grafana の HTTP API で認証するときパスワードの代替として機能する、生成されたランダム文字列です。

### 前提条件

サービスアカウントを作成および編集する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を作成および編集するには、組織の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](#) をご参照ください。

サービスアカウントにトークンを追加するには

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. トークンを追加するサービスアカウントを選択します。
4. サービスアカウントトークンの追加 を選択します。
5. トークンの名前を入力します。
6. 有効期限の設定を選択し、トークンの有効期限を入力します。
  - 有効期限は、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、数時間以下など、少し時間が経過すると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。
7. [Generate token] を選択します。

**Note**

Amazon Managed Grafana AWS APIs 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。を使用して [CreateWorkspaceServiceAccountToken](#)、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。

## サービスアカウントにロールを割り当てる

Grafana サービスアカウントにロールを割り当てて、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を作成および編集する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を作成および編集するには、組織の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](#) をご参照ください。

サービスアカウントにロールを割り当てるには

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。

## API キーを使用して Grafana HTTP APIs

Grafana APIs にアクセスする 1 つの方法は、API トークン とも呼ばれる API キー を使用することです。API キーを作成するには、次のいずれかの手順を使用します。API キーは、作成時に指定した期間限定で、最大 30 日間有効です。

### トピック

- [ワークスペースの Grafana API で使用する Grafana APIs キーの作成 \(コンソール\)](#)
- [を使用した Amazon Managed Grafana ワークスペース API キーの作成 AWS CLI](#)

**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を使用することをお勧めします。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主な方法として APIs。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は、キーのロールを指定します。ロールは、キーのユーザーが持つ管理能力のレベルを決定します。

次の表は、管理者、エディタ、およびビューワの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、組織の一般的なアクセス許可を示しています。この表では、フルとは、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、組織設定を管理するアクセス許可があるかどうかが表示されます。

ロール	ダッシュボード	プレイリスト	フォルダ	探索	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	なし	いいえ	いいえ	なし
[Editor] (エディタ)	フル	フル	フル	あり	いいえ	なし
Admin	フル	フル	フル	あり	フル	フル

次の表は、設定できる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、管理者、エディタ、ビューワ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	なし
[Edit] (編集)	作成、編集	ビュー	なし
Admin	作成、編集、削除	作成、編集、削除	あり

**Note**

アクセス許可レベルが低い、よりスコープの広いアクセス許可は、より多くのアクセス許可を持つより一般的なルールが存在する場合、効果がありません。例えば、ユーザーに組織のエディタロールを付与し、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、エディタロールのためにユーザーに完全な編集アクセス許可があるため、より制限の厳しい表示アクセス許可は効果がありません。

## ワークスペースの Grafana API で使用する Grafana APIs 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、ワークスペースに 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 ワークスペースの場合、API キーは AWS CLI または API からのみ作成できます。

API キーの削除は、今後のリリースで Grafana Labs によって発表されました。代わりにサービスアカウントを使用することをお勧めします。

ワークスペースコンソールで Grafana API で使用する Grafana APIs キーを作成するには

1. <https://console.aws.amazon.com/grafana/> で Amazon Managed Grafana コンソールを開きます。
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 WorkSpaces] 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、設定 (歯車) アイコンで一時停止し、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、キーを付与するアクセスレベルを選択します。Admin 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 APIs を使用できるようになります。Editor また

は Viewer を選択して、キーのユーザーをそれらのパワーレベルに制限します。詳細については、前の表を参照してください。

- 有効期限には、キーを有効にする期間を指定します。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、秒の場合は s、分の場合は m、時間の場合は h、日の場合は d、週の場合は w、月の場合は M です。例えば、12h は 12 時間、1M は 1 か月 (30 日) です。

キーの有効期間は、数時間以下など、より短く設定することを強くお勧めします。これにより、API キーを長期間有効にするよりもリスクがはるかに低くなります。

- 追加を選択します。

- (オプション) Terraform を使用して API キーの作成 API を使用して [API キー](#) の作成を自動化できます。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。

## を使用した Amazon Managed Grafana ワークスペース API キーの作成 AWS CLI

を使用して Amazon Managed Grafana ワークスペース API キーを作成するには AWS CLI

次の例では、*key\_name*、*key\_role*、*seconds\_to\_live*、*Workspace\_id* をユーザー自身の情報に置き換えます。key-name、key-role、およびの形式については seconds-to-live、API ガイド [https://docs.aws.amazon.com/grafana/latest/APIReference/API\\_CreateWorkspaceApiKey.html](https://docs.aws.amazon.com/grafana/latest/APIReference/API_CreateWorkspaceApiKey.html) の「」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 出力例

次のコマンドを実行すると、ワークスペースの *workspace\_id* を見つけることができます。

```
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。

Preferences 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 を使用してアラートを変更することはできません。新しいアラートを作成または変更するには、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。これは 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を作成する方法については、「[トークンによる認証](#)」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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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応答を制限します。複数のダッシュボードを指定できます。例えば、次のようになります: dashboardId=23&dashboardId=35

- `panelId` — ダッシュボード上の指定されたパネルのアラートに応答を制限します。
- `query` — レスポンスを、この値のような名前のアラートに制限します。
- `state` — 、 、 ALL、 、 または のアラート状態が 1 alerting okつ以上あるアラートを返しますno\_datapausedpending。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`
- `limit` — レスポンスを X 個のアラート数に制限します。
- `folderId` — 指定されたフォルダ内のダッシュボードのアラートに応答を制限します。複数のフォルダを指定できます。例えば、次のようになります: `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応答を制限します。
- `dashboardTag` — 指定したタグを持つダッシュボードのレスポンスアラートを制限します。複数のタグで「AND」フィルタリングを実行するには、`tags` パラメータを複数回指定します。例えば `dashboardTag=tag1&dashboardTag=tag2` です。これらは Grafana タグであり、AWS タグではない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 でアラートを受け取る

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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、アラートの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。あるサーバーのデータが最初にアラートをトリガーし、そのサーバーがアラート状態を離れるのを見る前に、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 によるアラートの一時停止

```
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 は、アラートtrueを一時停止するか、アラートfalseを一時停止解除することができます。

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "alertId": 1,
  "state": "Paused",
  "message": "alert paused"
}
```

## アラート通知チャンネル API

アラート通知チャンネル API を使用して、通知チャンネルを作成、更新、削除、取得します。

通知チャンネルの識別子 (id) は自動増分の数値であり、ワークスペースごとにのみ一意です。通知チャンネルの一意の識別子 (uid) を使用して、複数のワークスペース間のフォルダを一意に識別できます。通知チャンネルの作成時に指定しない場合、自動的に生成されます。uid を使用すると、通知チャンネルにアクセスしたり、複数の Amazon Managed Grafana ワークスペース間で通知チャンネルを同期したりするための一貫した URLs 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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## すべての通知チャンネルを取得する

認証されたユーザーが表示許可を持っているすべての通知チャンネルを返します。

```
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を取得する (ルックアップ)

すべての通知チャネルを返しますが、詳細情報は少なくなります。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されます。

```
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 ですべての通知チャネルを取得する

```
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 ですべての通知チャネルを取得する

```
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、「」でサポートされている通知機能のリストを参照してください[コンタクトポイントの使用](#)。

```
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 で通知チャネルを更新する

```
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 で通知チャンネルを更新する

```
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 で通知チャンネルを削除する

```
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 で通知チャンネルを削除する

```
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と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

Annotations API を使用して、Amazon Managed Grafana ワークスペースで注釈を作成、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で、どのダッシュボードにも表示できるグローバル注釈にすることができます。注釈はタグでフィルタリングされます。または、ダッシュボードのパネルに紐付けて、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。これは 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を作成する方法については、「[トークンによる認証](#)」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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### クエリパラメータ :

- from — (オプション) ミリ秒単位のエポック日時。
- ~ — (オプション) ミリ秒単位のエポック日時。
- limit — (オプション) 返される結果の最大数。デフォルトは 100 です。
- alertid — (オプション) 指定されたアラートの注釈Find。
- dashboardId — (オプション) 指定されたダッシュボードにスコープされた注釈を検索します。
- panelId — (オプション) 指定されたパネルにスコープされた注釈を検索します。

- `userId` — (オプション) 指定したユーザーによって作成された注釈を検索します。
- `type` — (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には、`alert` および `annotation` を指定します。値は `alert` および `annotation` です。
- `tags` — (オプション) グローバルアノテーション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、ダッシュボードまたはパネルに特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」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` です。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 データベースに注釈を作成します。dashboardId および panelId フィールドはオプションです。指定しない場合、グローバルアノテーションが作成され、Grafana アノテーション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。リージョンアノテーションを作成するときは、必ず timeEnd プロパティを含めてください。

time および timeEnd の形式は、ミリ秒単位のエポック番号timeEndである必要があります。

### リクエストの例

```
POST /api/annotations HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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,
}
```

## グラフィット形式で注釈を作成する

```
POST /api/annotations/graphite
```

Graphite 互換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 が指定されていない場合、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 形式 (スペースで区切られた複数のタグを含む文字列) より前の 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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
```

は、指定された ID に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には、パッチアノテーション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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
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"
}
```

## パッチアノテーション

```
PATCH /api/annotations/:id
```

指定された ID に一致する注釈の 1 つ以上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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
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 で注釈を削除する

```
DELETE /api/annotations/:id
```

指定された ID に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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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。これは 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を作成する方法については、「」を参照してください [トークンによる認証](#)。

## 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### クエリパラメータ :

- 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。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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "mykey",
  "role": "Admin",
  "secondsToLive": 86400
}
```

### JSON 本文スキーマ :

- name — キーの名前。
- 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時間を設定します。2592000 (30 日) 以下である必要があります。

### レスポンスの例

```
{"name":"mykey","key":"eyJrIjoiWHZiSWd3NzdCYUZnNUtibE9obUpESmE3bzJYNDRlcm0UjLCJuIjoibXlrZXkiLCJp
```

### エラーステータス :

- 400 - secondsToLiveが 2592000 より大きい
- 500 - キーをデータベースに保存できませんでした。

## API キーを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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

Dashboard API を使用して、Amazon Managed Grafana ワークスペースで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) は自動増分の数値であり、ワークスペースごとにのみ一意です。ダッシュボードの一意の識別子 (uid) を使用して、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できます。ダッシュボードの作成時に指定しない場合、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 URLs を持つことができます。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への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。これは 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を作成する方法については、「[」を参照してください](#)  
[トークンによる認証](#)。

## 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するか、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リクエストの例

```
POST /api/dashboards/db HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 本文スキーマ :

- ダッシュボード — 完全なダッシュボードモデル。null を使用して新しいダッシュボードを作成します。
- dashboard.id — null を使用して新しいダッシュボードを作成します。
- dashboard.uid — これを使用して新しいダッシュボードを作成する場合のオプション—意識別子。null の場合、新しい uid が生成されます。
- folderid — ダッシュボードを保存するフォルダの ID。
- folderUid — ダッシュボードを保存するフォルダの Uid。の値をオーバーライドします。  
folderid

- 上書き — 既存のダッシュボードを新しいバージョン、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 true するには、 を指定します。
- message — バージョン履歴のコミットメッセージを設定します。
- refresh — ダッシュボードの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、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の例

```
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、欠落しているフィールドなどのエラー
- 401 - 未承認
- 403 - アクセス拒否
- 412 - 前提条件が失敗しました

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を説明するために使用されます。

- ダッシュボードが他のユーザーによって変更されている `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` を持つダッシュボードが既に存在する `status=name-exists`
- ダッシュボードがプラグインに属している `plugin title status=plugin-dashboard`

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 でダッシュボードを取得する

```
GET /api/dashboards/uid/:uid
```

`uid` に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` に関する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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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 - 見つかりました
- 401 - 未承認
- 403 - アクセス拒否
- 404 - 見つかりません

## uid でダッシュボードを削除する

```
DELETE /api/dashboards/uid/:uid
```

uid に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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 未承認
- 403 - アクセス拒否
- 404 - 見つかりません

ホームダッシュボードを取得します。

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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
```

ダッシュボードのすべてのタグを返します。

### リクエストの例

```
GET /api/dashboards/tags HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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。

を使用したアクセス許可 `dashboardId=-1` は、ビューワールールとエディタール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 (ビューワールールまたはエディタールール) に設定できます。管理者にはアクセス許可を設定できません。管理者は常にすべてのアクセス許可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。これは 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を作成する方法については、「」を参照してください [トークンによる認証](#)。

## ダッシュボードのアクセス許可を取得する

```
GET /api/dashboards/id/:dashboardId/permissions
```

指定された を使用して、ダッシュボードに対する既存のアクセス許可をすべて取得します dashboardId。

### 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 未承認
- 403 - アクセス拒否
- 404 - ダッシュボードが見つかりません



## ダッシュボードのアクセス許可を更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "items": [
    {
      "role": "Viewer",
      "permission": 1
    },
    {
      "role": "Editor",
      "permission": 2
    },
    {
      "teamId": 1,
      "permission": 1
    },
    {
      "userId": 11,
      "permission": 4
    }
  ]
}
```

### JSON 本文スキーマ :

- items — 追加または更新するアクセス許可項目。リストから省略された既存の項目は削除されません。

### 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 - 未承認
- 403 - アクセス拒否
- 404 - ダッシュボードが見つかりません

## ダッシュボードバージョン API

Dashboard Versions API を使用して、ダッシュボードのバージョンを取得し、指定した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。これは 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を作成する方法については、「[トークンによる認証](#)」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を取得する

```
GET /api/dashboards/id/:dashboardId/versions
```

指定された `id` を使用して、ダッシュボードのすべての既存のダッシュボードバージョンを取得します `dashboardId`。

クエリパラメータ：

- `limit` — 返される結果の最大数。
- `start` — クエリを返すときに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 - 未承認

- 404 - ダッシュボードバージョンが見つかりません

## ダッシュボードのバージョンを取得する

```
GET /api/dashboards/id/:dashboardId/versions/:id
```

特定の 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 dashboardId。

### 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 - 未承認
- 404 - ダッシュボードバージョンが見つかりません

## ダッシュボードの復元

```
POST /api/dashboards/id/:dashboardId/restore
```

ダッシュボードを指定したダッシュボードバージョンに復元します。

リクエストの例

```
POST /api/dashboards/id/1/restore
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
```

```
"version": 1
}
```

JSON 本文スキーマ :

- version — 復元先のダッシュボードのバージョン。

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 レスポンス本文スキーマ :

- slug — ダッシュボードのタイトルの URL にわかりやすいスラグ。
- 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 - 未承認
- 404 - ダッシュボードまたはダッシュボードのバージョンが見つかりません
- 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を取得する際の問題を示します)

エラーレスポンスの例 :

```
HTTP/1.1 404 Not Found
Content-Type: application/json; charset=UTF-8
```

```
Content-Length: 46
```

```
{
  "message": "Dashboard version not found"
}
```

JSON レスポンス本文スキーマ :

- message — 失敗の理由を説明するメッセージ。

## ダッシュボードのバージョンを比較する

```
POST /api/dashboards/calculate-diff
```

JSON 差分を計算することで、2 つのダッシュボードバージョンを比較します。

リクエストの例

```
POST /api/dashboards/calculate-diff HTTP/1.1
Accept: text/html
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "base": {
    "dashboardId": 1,
    "version": 1
  },
  "new": {
    "dashboardId": 1,
    "version": 2
  },
  "diffType": "json"
}
```

JSON 本文スキーマ :

- base — ベースダッシュボードのバージョンを表すオブジェクト。
- new — 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 diff のタイプ。有効な値は、json および basic です。



## レスポンスの例 (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 diff のテキスト表現で、ダッシュボードの値は JSON で、GitHub やなどのサイトで見られる差分に似ています GitLab。

ステータスコード：

- 200 - OK
- 200 - 不正なリクエスト、無効な JSON が送信されました
- 401 - 未承認
- 404 - 見つかりません

## レスポンスの例 (基本的な差分)

```
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 が送信されました
- 401 - 未承認
- 404 - 見つかりません

# データソース API

データソース API を使用して、Amazon Managed Grafana ワークスペースで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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## すべてのデータソースを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 で単一のデータソースを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 で単一のデータソースを取得する

```
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
}
```

## 名前で単一のデータソースを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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
}
```

## データソース ID を名前で取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 リクエストの例

```
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 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": false,
    "basicAuthUser": "",
    "basicAuthPassword": "",
    "withCredentials": false,
    "isDefault": false,
    "jsonData": {},
    "secureJsonFields": {},
    "version": 1,
    "readOnly": false
  },
  "id": 1,
```

```
"message": "Datasource added",
"name": "test_datasource"
}
```

### Note

内で password および basicAuthPassword を定義すると secureJsonData、Amazon Managed Grafana はデータベース内の暗号化された BLOB としてそれらを安全に暗号化します。次に、レスポンスは の暗号化フィールドを一覧表示します secureJsonFields。

## 基本認証が有効になっている Graphite リクエストの例

```
POST /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "name": "test_datasource",
  "type": "graphite",
  "url": "http://mydatasource.com",
  "access": "proxy",
  "basicAuth": true,
  "basicAuthUser": "basicuser",
  "secureJsonData": {
    "basicAuthPassword": "basicpassword"
  }
}
```

## 基本認証を有効にしたレスポンスの例

```
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 added",
"name": "test_datasource"
}
```

## CloudWatch リクエストの例

```
POST /api/datasources HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUjY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "test_datasource",
  "type": "cloudwatch",
  "url": "http://monitoring.us-west-1.amazonaws.com",
  "access": "proxy",
  "jsonData": {
    "authType": "keys",
    "defaultRegion": "us-west-1"
  },
  "secureJsonData": {
    "accessKey": "014pIDpeKSA6Xikg0l4p",
    "secretKey": "dGVzdCBrZXkgYmxlYXNlIGRvbid0IHN0ZWFs"
  }
}
```

## 既存のデータソースを更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "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

データベースに暗号化された BLOB として安全に保存 `secureJsonData` されるように、`password` および `basicAuthPassword` を `basicAuthPassword` 内で定義することをお勧めします。次に、レスポンスは `secureJsonFields` の暗号化フィールドを一覧表示します。

## ID でデータソースを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Data source deleted"}
```

## UID でデータソースを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Data source deleted"}
```

## 名前でデータソースを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
```

```
Content-Type: application/json
```

```
{
  "message": "Data source deleted",
  "id": 1
}
```

## データソースプロキシ呼び出し

```
GET /api/datasources/proxy/:datasourceId/*
```

実際のデータソースへのすべての呼び出しをプロキシします。

## ID によるデータソースのクエリ

```
POST /api/tsdb/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。ほとんど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、および queries プロパティが必要です。

JSON 本文スキーマ :

- from/to — エポックタイムスタンプの絶対値をミリ秒単位で指定するか、Grafana 時間単位を使用して相対値にする必要があります。例えば now-1h です。
- queries.refId — (オプション) クエリの識別子を指定します。デフォルトは A です。
- queries.datasourceId — クエリ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。
- クエリ。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.intervalMs — (オプション) 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。

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 \"0opened\",\n sum(closed) AS\n \"Closed\"\nFROM\n issues_activity\nWHERE\n $__unixEpochFilter(time) AND\n period\n = 'm' AND\n repo IN('grafana/grafana') AND\n opened_by IN('Contributor','Grafana\n 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の例 :

```
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\n  \"Closed\"\nFROM\n  issues_activity\nWHERE\n  time <= 1420066800 AND time >=\n  1575845999 AND\n  period = 'm' AND\n  repo IN('grafana/grafana') AND\n  opened_by\n  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

データソースのアクセス許可 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のアクセス許可を設定できます。アクセス許可は、常にすべてにアクセスできるため、管理者には設定できません。

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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## データソースのアクセス許可を有効にする

```
POST /api/datasources/:id/enable-permissions
```

指定された 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がデータソースをクエリすることを許可する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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```



## 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することはできません。詳細については、レスポンス本文を参照してください。
- 401 - 未承認
- 403 - アクセス拒否
- 404 - データソースが見つかりません

## データソースのアクセス許可を無効にする

```
POST /api/datasources/:id/disable-permissions
```

指定された 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はすべて削除され、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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{}
```

## 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することはできません。詳細については、レスポンス本文を参照してください。
- 401 - 未承認
- 403 - アクセス拒否
- 404 - データソースが見つかりません

## データソースのアクセス許可を取得する

```
GET /api/datasources/:id/permissions
```

指定された `id` を使用して、データソースに対する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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

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 - 未承認
- 403 - アクセス拒否
- 404 - データソースが見つかりません

## データソースのアクセス許可を追加する

```
POST /api/datasources/:id/permissions
```

指定された `id` で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の例

```
POST /api/datasources/1/permissions
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "userId": 1,
  "permission": 1
}
```

### ユーザーアクセス許可を追加するためのレスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permission added"}
```

### チームアクセス許可を追加するリクエストの例

```
POST /api/datasources/1/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "teamId": 1,
  "permission": 1
}
```

### チームアクセス許可を追加するためのレスポンスの例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Datasource permission 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 - 未承認

- 403 - アクセス拒否
- 404 - データソースが見つかりません

## データソースのアクセス許可を削除する

```
DELETE /api/datasources/:id/permissions/:permissionId
```

指定された `id` を持つデータソースの指定された `permissionId` を持つ 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 未承認
- 403 - アクセス拒否
- 404 - データソースが見つからないか、アクセス許可が見つかりません

## 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。これは 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を作成する方法については、「」を参照してください [トークンによる認証](#)。

## 外部グループを取得する

```
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 - 未承認
- 403 - アクセス拒否

## 外部グループを追加する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk]

{
  "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 - グループは既にこのチームに追加されています
- 401 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つかりません

## 外部グループを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk]
```

## 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 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つからないか、グループが見つかりません

## フォルダ API

Folder 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) は自動増分の数値であり、ワークスペースごとにのみ一意です。フォルダの一意の識別子 (uid) を使用して、複数のワークスペース間のフォルダを一意に識別できます。フォルダの作成時に指定しない場合、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 ワークスペース間でフォルダを同期したりするための一貫した URLs を設定できます。uid を使用すると、フォルダのタイトルを変更しても、そのフォルダへの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 文字です。

フォルダはネストできません。

### 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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

が 0 id の General フォルダは、Folder API の一部ではありません。Folder API を使用して、一般フォルダに関する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 本文スキーマ :

- uid — オプションの一意の識別子。null の場合、新しい uid が生成されます。
- title — フォルダのタイトル。

### 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、欠落しているフィールドなどのエラー
- 401 - 未承認
- 403 - アクセスが拒否されました

## フォルダの更新

```
PUT /api/folders/:uid
```

uid に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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "title": "Department DEF",
  "version": 1
}
```

JSON 本文スキーマ：

- uid — 指定された場合は、一意の識別子を変更します。
- title — フォルダのタイトル。
- version — フォルダを上書きできるように、現在のバージョンを指定します。の場合は必要ありません `overwrite=true`。
- overwrite — 既存のフォルダを新しいバージョンで上書き `true` するには、 に設定します。

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、欠落しているフィールドなどのエラー
- 401 - 未承認
- 403 - アクセス拒否
- 404 - フォルダが見つかりません
- 412 - 前提条件が失敗しました

412 ステータスコードは、フォルダを更新できない理由を説明するために使用されます。

- フォルダが他のユーザーによって変更されている 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"
}
```

## すべてのフォルダを取得する

```
GET /api/folders
```

表示する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して、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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmN1MkZYbk
```

### 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": "k3S1ck1Gk",
    "title": "Department RND"
  }
]
```

## uid でフォルダを取得する

```
GET /api/folders/:uid
```

指定された uid に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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 見つかりました
- 401 - 未承認
- 403 - アクセス拒否
- 404 - 見つかりません

## ID でフォルダを取得する

```
GET /api/folders/id/:id
```

指定された ID に一致する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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 見つかりました
- 401 - 未承認
- 403 - アクセス拒否
- 404 - 見つかりません

## uid でフォルダを削除する

```
DELETE /api/folders/:uid
```

uid に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は元に戻すことができません。

### リクエストの例

```
DELETE /api/folders/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 未承認
- 403 - アクセスが拒否されました
- 404 - 見つかりません

## フォルダ/ダッシュボード検索 API

FolderDashboard-Search 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。これは 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を作成する方法については、「[トークンによる認証](#)」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。これらは Grafana タグであり、AWS タグではありません。
- `type` — 検索するタイプ。dash-folderまたは。dash-db
- `dashboardIds` — 検索するダッシュボード ID のリスト。
- `folderIds` — ダッシュボードで検索するダッシュボード ID のリスト。
- `starred` - 星型ダッシュボードのみが返されるように指定するフラグ。
- `limit` — 返される結果の数を制限します (最大数は 5000 です)。
- `page` — 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。番号付けは 1 から始まります。limit パラメータはページサイズとして機能します。

一般フォルダのフォルダとダッシュボードを取得するリクエストの例

```
GET /api/search?folderIds=0&query=&starred=false HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

一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の例

```
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を検索するリクエストの例

```
GET /api/search?query=Production%20overview&starred=true&tag=prod HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### スター付きダッシュボードを検索するためのレスポンスの例

```
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。

を使用したアクセス許可 `folderId=-1` は、ビューワールールとエディタール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 (ビューワールールまたはエディタールール) に設定できます。管理者にはアクセス許可を設定できません。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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## フォルダのアクセス許可を取得する

```
GET /api/folders/:uid/permissions
```

指定された `uid` を持つフォルダに対する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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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 - 未承認
- 403 - アクセス拒否
- 404 - フォルダが見つかりません

## フォルダのアクセス許可を更新する

```
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 本文スキーマ :

- items — 追加または更新するアクセス許可項目。リストから省略された既存の項目は削除されません。

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 - 未承認
- 403 - アクセス拒否
- 404 - ダッシュボードが見つかりません

## Organization API

Organization API を使用して、Amazon Managed Grafana ワークスペース内の組織と連携します。

### 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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## 現在の組織を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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を取得する

```
GET /api/org/users
```

必要なアクセス許可: スコープを持つ `org.users:read` アクション `users:*`

### リクエストの例

```
GET /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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を取得する (検索)

```
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、詳細情報は少なくなります。組織の管理者ロール、任意のフォルダ内の管理者、または任意のチームの管理者を持つユーザーがアクセスできます。主に Grafana UI によって使用され、チームメンバーを追加したり、フォルダ/ダッシュボードのアクセス許可を分割したりするときに、ユーザーのリストを提供します。

### 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"
  }
]
```

## 指定されたユーザーを更新します。

```
PATCH /api/org/users/:userId
```

必要なアクセス許可: スコープを持つ `org.users.role:update` アクション `users:*`

### リクエストの例

```
PATCH /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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を削除します

```
DELETE /api/org/users/:userId
```

必要なアクセス許可: スコープを持つ `org.users:remove` アクション `users:*`

### リクエストの例

```
DELETE /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
```



```
{"message":"User removed from organization"}
```

## 現在の組織を更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name":"Main Org."
}
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization updated"}
```

## 現在の組織にユーザーを追加する

```
POST /api/org/users
```

必要なアクセス許可: スコープを持つ `org.users:add` アクション `users:*`

### リクエストの例

```
POST /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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。これは 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を作成する方法については、「」を参照してください。[トークンによる認証](#)。

## プレイリストの検索

```
GET /api/playlists
```

ページ割りを使用して、現在の Amazon Managed Grafana ワークスペースのすべてのプレイリストを返します。

### リクエストの例

```
GET /api/playlists HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

クエリ文字列パラメータ :

- query — レスポンスを、この値のような名前のプレイリストに制限します。
- limit — レスポンスを X 個のプレイリストに制限します。

## 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 つ取得する

```
GET /api/playlists/:id
```

## リクエストの例

```
GET /api/playlists/1 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdeE9ZWmNrMkZYbk
```

## 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"  
  }  
]  
}
```

## プレイリスト項目を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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,
  }
]
```

## プレイリストを作成する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
{
  "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"  
}
```

## プレイリストを更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk  
{  
  "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"
    }
  ]
}
```

## プレイリストを削除する

```
DELETE /api/playlists/:id
```

### リクエストの例

```
DELETE /api/playlists/1 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{}
```

## プラグイン API

Plugin API を使用して、Amazon Managed Grafana ワークスペースの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、ワークスペースで [プラグイン管理が有効になっている](#) 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。これは 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を作成する方法については、「」を参照してください [トークンによる認証](#)。

## プラグインをインストールする

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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"
  }
]
}
```

## プラグインを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 ワークスペースのユーザー設定を操作します。

キー：

- テーマ — 有効な値はlight、デフォルトのテーマを使用する、dark、または空の文字列です。
- homeDashboardId— お気に入りダッシュボード:idの数値。デフォルトは 0 です。
- timezone — 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、現在の値が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。これは 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を作成する方法については、「」を参照してください[トークンによる認証](#)。

## 現在のユーザー設定を取得する

```
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を更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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"}
```

## 現在の組織設定を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{"theme":"","homeDashboardId":0,"timezone":""}
```

## 現在の組織設定を更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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。これは API リクエストの Authorization フィールドに含め

まず、API コールを認証するトークンを作成する方法については、「」を参照してください。[トークンによる認証](#)。

## 新しいスナップショットを作成する

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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 本文スキーマ :

- ダッシュボード — (必須) ダッシュボードモデル全体。
- name — (オプション) スナップショットの名前。
- 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表します。デフォルトでは、有効期限はありません。
- 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 false です。
- key — ( externalが の場合に必須true) 一意のキーを定義します。
- deletekey — ( externalが の場合に必須true) スナップショットの削除に使用する一意のキー。作成者だけがスナップショットを削除できるようにkey、とは異なります。

#### Note

API を使用してスナップショットを作成する場合は、スナップショットデータを含むダッシュボードのペイロード全体を提供する必要があります。このエンドポイントは Grafana UI 用に設計されています。

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json
{
  "deleteKey": "XXXXXXXX",
  "deleteUrl": "myurl/api/snapshots-delete/XXXXXXXX",
  "key": "YYYYYYYY",
  "url": "myurl/dashboard/snapshot/YYYYYYYY",
  "id": 1,
}
```

キー :

- deleteKey — スナップショットの削除に使用されるために生成されたキー。
- key — ダッシュボードを共有するために生成されたキー。

## スナップショットのリストを取得する

```
GET /api/dashboard/snapshots
```

クエリパラメータ：

- クエリ — クエリの検索
- limit — 返される結果の数を制限する

リクエストの例

```
GET /api/dashboard/snapshots HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":8,
    "name":"Home",
    "key":"YYYYYYYY",
    "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"
  }
]
```

## キー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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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
}
}
```

## キー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して認証なしで使用できません。

```
GET /api/snapshots-delete/:deleteKey
```

### リクエストの例

```
GET /api/snapshots-delete/XXXXXXXX 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

Team API を使用して、Amazon Managed Grafana ワークスペース内のチームを操作します。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。これは 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を作成する方法については、「[トークンによる認証](#)」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
```

または

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 パラメータが 10 に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結果を返します。スペースを含むクエリ値は URL エンコードする必要があります。例えば query=my%20team です。

## name パラメータの使用

name パラメータが name フィールドと一致する場合、パラメータは 1 つ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 - 未認可
- 403 - アクセス許可が拒否されました
- 404 - チームが見つからない (名前で検索した場合)

## ID でチームを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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"
}
```

## チームを追加する

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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 - 未承認
- 403 - アクセス許可が拒否されました
- 409 - チーム名は既に存在します

## チームの更新

```
PUT /api/teams/:id
```

更新できるのは name および email フィールドのみです。

リクエストの例

```
PUT /api/teams/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つかりません
- 409 - チーム名は既に存在します

## ID でチームを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つかりません

## チームメンバーを取得する

```
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 - 未承認
- 403 - アクセス許可が拒否されました

## 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "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 - ユーザーはすでにチームに入っています
- 401 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つかりません

## チームからメンバーを削除する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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 - 未承認
- 403 - アクセス許可が拒否されました
- 404 - チームが見つからない/チームメンバーが見つかりません

## チーム設定を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

レスポンスの例

```
HTTP/1.1 200
Content-Type: application/json

{
  "theme": "",
  "homeDashboardId": 0,
  "timezone": ""
}
```

## チームの設定を更新する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark",
  "homeDashboardId": 39,
  "timezone": "utc"
}
```

### JSON 本文スキーマ :

- theme — デフォルトのテーマを使用するにはlight、dark、または空の文字列を指定します。
- homeDashboardId— ダッシュボード:idの数値。デフォルトは 0 です。
- timezone — デフォルトを使用するにはutc、browser、または空の文字列を指定します。

パラメータを省略すると、現在の値が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 ワークスペースのユーザーを操作します。

### 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。これは 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を作成する方法については、「[トークンによる認証](#)」を参照してください。

## ユーザーがメンバーであるチームを取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### 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を取得する

実際のユーザーの指定されたダッシュボードをスターします。

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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リングを解除する

実際のユーザーの特定の Dashboard のスターリングを削除します。

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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を取得する

```
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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## 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を取り消す

```
POST /api/user/revoke-auth-token
```

実際のユーザーの指定された認証トークン (デバイス) を取り消します。発行された認証トークン (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する必要があります。

### 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 eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{  
  "authTokenId": 364  
}
```

## レスポンスの例

```
HTTP/1.1 200  
Content-Type: application/json  
  
{  
  "message": "User auth token revoked"  
}
```

# 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 が使用される最も一般的なシナリオの 1 つです。このページでは、このシナリオのテンプレートプロジェクトを提供するソリューションについて説明します。このソリューションは、[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では、以下を設定します。

- Amazon Managed Service for Prometheus ワークスペースは、Amazon EKS クラスターからメトリクスを保存し、メトリクスをスクレイプ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。ログはに保存され CloudWatch、Amazon Managed Grafana によってクエリされます。詳細については、[「Amazon のログ記録EKS」](#) を参照してください。
- これらのログと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とアラートを作成するための Amazon Managed Grafana ワークスペース。

このソリューションを適用すると、次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状態を評価します。
- Amazon EKS コントロールプレーンの状態とパフォーマンスを表示します。
- Amazon EKS データプレーンの状態とパフォーマンスを表示します。
- Kubernetes 名前空間全体の Amazon EKS ワークロードに関する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、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ことで、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 クラスターは、ノードレベルからポッド、Kubernetes レベルまで把握できます。これには、リソース使用状況の詳細なモニタリングが含まれます。

このソリューションは、予測機能と修正機能の両方を提供します。

- 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 SLAs の内部ユーザーにパフォーマンスと信頼性を提供するために、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 とメモリリソースを割り当てることができます。
  - 使用状況予測: ノード、Amazon [によってバックアップされる永続ボリューム](#)、[Application Load Balancer](#) などの Amazon EKS クラスターリソースの現在の使用率に基づいて、需要が似た新しい製品やプロジェクトなど、事前に計画できます。 <https://docs.aws.amazon.com/eks/latest/userguide/aws-load-balancer-controller.html>
  - 潜在的な問題を早期に検出: 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把握できます。
- 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) を短縮します。例えば、トラブルシューティングダッシュボードを見ることで、何が問題だったかについての仮説をすばやくテストし、それを排除できます。
  - 問題が発生しているスタック内の場所を特定します。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。

## Observability Accelerator Dashboards

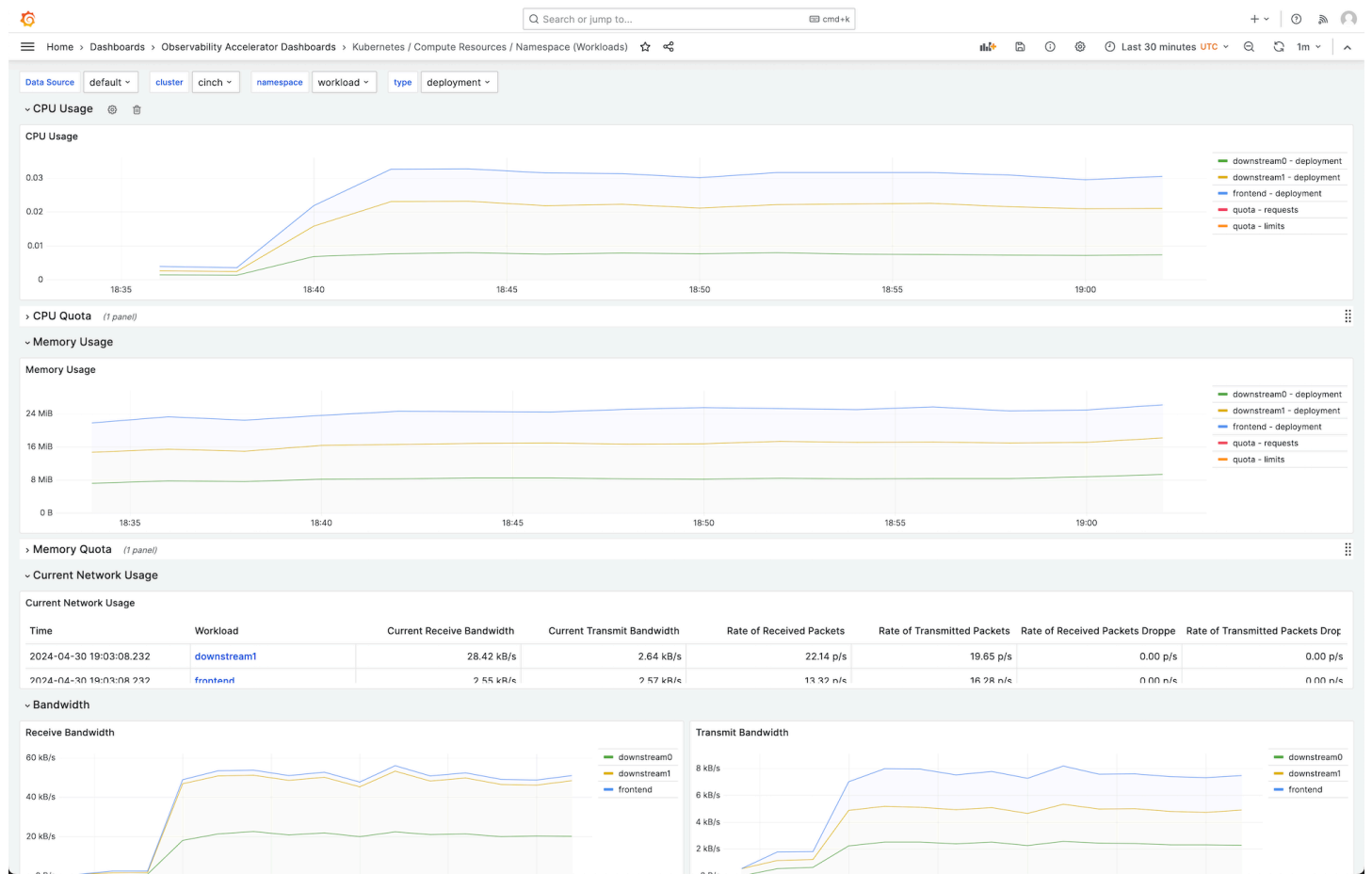
Manage folder dashboards and permissions

The screenshot shows the 'Dashboards' management page in Amazon Managed Grafana. At the top, there are navigation tabs for 'Dashboards', 'Panels', 'Permissions', and 'Settings'. Below the navigation is a search bar labeled 'Search for dashboards' and a 'New' button. Underneath the search bar are filter options: 'Filter by tag', 'Include panels', and 'Starred'. To the right of these filters are icons for folder and list views, and a 'Sort' dropdown menu. The main content area displays a list of dashboard entries, each with a checkbox, a folder path, a sub-name, and a tag. The entries are:

- Kubernetes / Compute Resources / Cluster  
Observability Accelerator Dashboards (tag: infrastructure)
- Kubernetes / Compute Resources / Namespace (Workloads)  
Observability Accelerator Dashboards (tag: workload)
- Kubernetes / Compute Resources / Node (Pods)  
Observability Accelerator Dashboards (tag: infrastructure)
- Kubernetes / Compute Resources / Workload  
Observability Accelerator Dashboards (tag: workload)
- Kubernetes / Kube-apiserver (advanced)  
Observability Accelerator Dashboards (tag: infrastructure)
- Kubernetes / Kube-apiserver (basic)  
Observability Accelerator Dashboards (tag: infrastructure)
- Kubernetes / Kube-apiserver (troubleshooting)  
Observability Accelerator Dashboards
- Kubernetes / Kubelet  
Observability Accelerator Dashboards (tag: infrastructure)
- Node Exporter / Nodes  
Observability Accelerator Dashboards (tag: infrastructure)

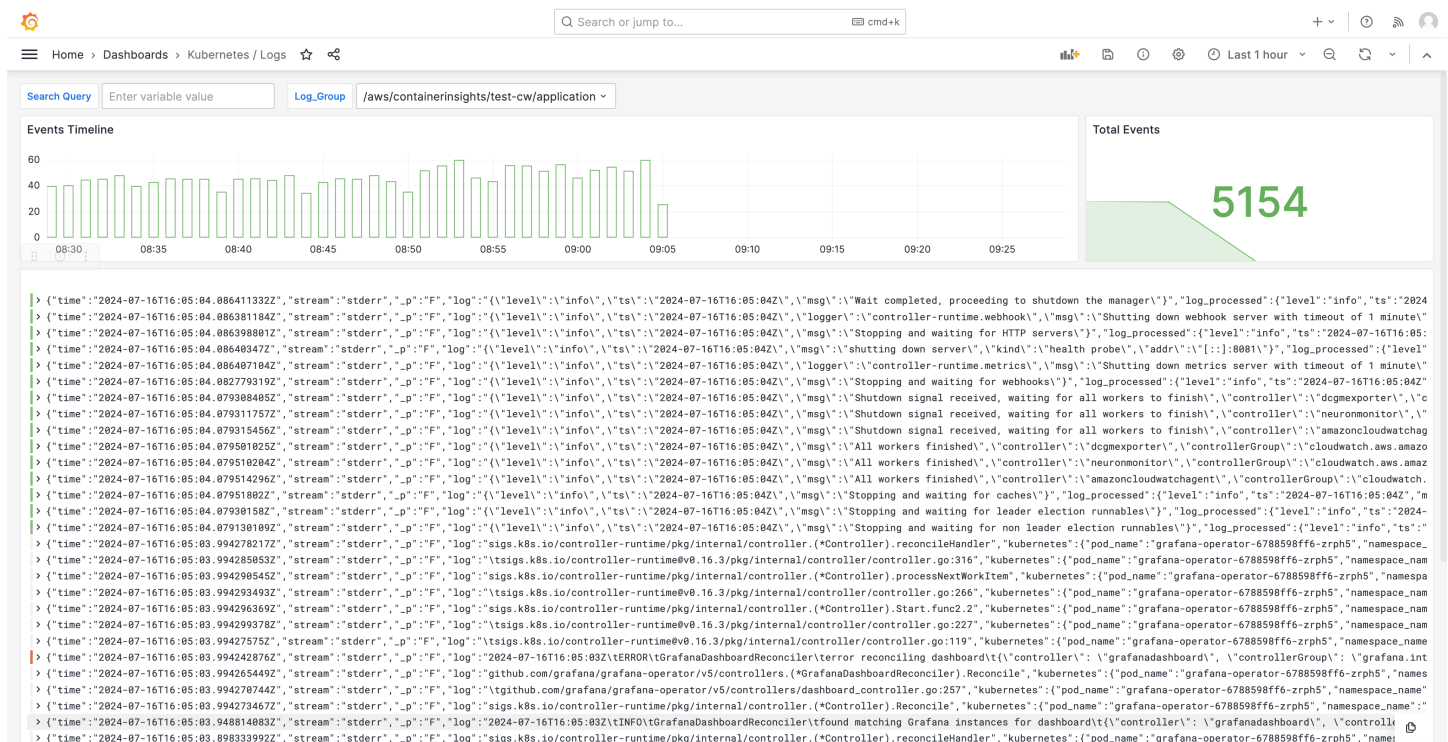
ダッシュボードを選択して詳細を表示できます。例えば、ワークロードのコンピューティングリソースを表示すると、次の図に示すようなダッシュボードが表示されます。





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計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。次の図は、ログダッシュボードを示しています。



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、「」を参照してください[追跡されるメトリクスのリスト](#)。

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、「」を参照してください[作成されたアラートのリスト](#)。

## コスト

このソリューションでは、ワークスペースにリソースを作成して使用します。作成されたリソースの標準使用量に対して、以下を含む料金が発生します。

- ユーザーによる Amazon Managed Grafana ワークスペースアクセス。料金の詳細については、「[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やメトリクス分析 (クエリサンプル処理) など、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の数は、Amazon EKS クラスターの設定と使用状況によって異なります。

Amazon Managed Service for Prometheus の取り込みとストレージのメトリクスは、CloudWatch を使用して表示できます。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](#)表ページにある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の数と、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 取り込み、ストレージ、分析を記録します。デフォルトでは、ログの保持は有効期限が切れないように設定されています。これはで調整できます CloudWatch。料金の詳細については、「[Amazon CloudWatch の料金](#)」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な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	計算ツールメトリクス	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)	コレクターの数	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く)
CloudWatch (ログ)	標準ログ: 取り込まれたデータ	24.5 GB (ベース)

製品	計算ツールメトリクス	値
		0.5 GB (ノードあたり)
	ログストレージ/アーカイブ (標準ログと提供されたログ)	ログの保存: 1 か月の保持を前提とする
	スキャンされた予想されるログデータ	Grafana からの各ログインサイトクエリは、指定された期間に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。

これらの番号は、追加のソフトウェアなしで実行されるソリューションの基本番号EKS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なくなります。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にあるかどうか AWS リージョン、および によって異なりますVPN。

#### Note

このテーブルの項目に、リソースあたりの(base)値と値 (など(per node)) が含まれている場合は、リソースごとの値に、そのリソースの数値を掛けた基本値を追加する必要があります。例えば、平均アクティブ時系列には、 の数値を入力します8000 + the number of nodes in your cluster \* 15,000。ノードが 2 つある場合は、 と入力します38,0008000 + ( 2 \* 15,000 )。

## 前提条件


このソリューションを使用するには、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 クラスターを持っているか作成する必要があります。また、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APIサーバーエンドポイントアクセスが設定されている必要があります (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**にはAPIアクセスを含める必要があります (APIまたは のいずれかに設定できますAPI\_AND\_CONFIG\_MAP )。これにより、ソリューションの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 (コンソールを使用してクラスターを作成する場合はデフォルトで true ですが、 または を使用して AWS API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 ) AddOns。AWS CNI、CoreDNS、Kube-proxy。

クラスター名を保存して、後で を指定します。これは、Amazon EKSコンソールのクラスターの詳細にあります。

 Note

Amazon EKSクラスターの作成方法の詳細については、「Amazon [の開始方法EKS](#)」を参照してください。

2. Amazon Managed Service for Prometheus ワークスペースは、Amazon クラスターと同じに作成する必要があります。AWS アカウント EKS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 ワークスペースを保存ARNして、後で指定します。

3. Amazon Managed Grafana ワークスペースは、Amazon クラスターと同じで Grafana バージョン 9 以降を使用して作成する必要があります。AWS リージョン EKS新しいワークスペースの作成の詳細については、「」を参照してください[Amazon Managed Grafana ワークスペースを作成する](#)。

ワークスペースロールには、Amazon Managed Service for Prometheus と Amazon CloudWatch にアクセスするためのアクセス許可が必要ですAPIs。これを行う最も簡単な方法は、[サービス管理アクセス許可](#)を使用し、Amazon Managed Service for Prometheus と を選択することで CloudWatch。[AmazonPrometheusQueryAccess](#) および [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 の形式ですg-123example。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URLの であり、ID が含まれています。例えば、https://g-123example.grafana-workspace.<region>.amazonaws.com/ と指定します。

4. Terraform でソリューションをデプロイする場合は、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、デプロイ用の 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 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 の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

ソリューションをセットアップするために必ずしも必要ではありませんが、ユーザーが作成したダッシュボードにアクセスする前に、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ように AWS インフラストラクチャを設定します。または [Terraform AWS Cloud Development Kit \(AWS CDK\)](#) を使用してインストールできます。

以下の 2 つのセクションでは、`awscli` をインストールする手順について説明します。

## を使用したインストール AWS CDK

このソリューションが提供される 1 つの方法は、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入力すると、ソリューションによってスクレイパー、ログ、ダッシュボードが作成されます。

### Note

ここでのステップでは、AWS CLI、AWS CDK、がインストールされている環境があることを前提と [NPM](#) しています。すべての前提条件を持つ環境を持つ 1 つの方法は、[AWS CloudShell](#) を使用することです。

このソリューションを使用して 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 [???](#) ステップをすべて完了していることを確認します。
2. Amazon S3 からソリューションのすべてのファイル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変更する必要はありません。

3. コマンドライン環境 (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) で、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。置換 `REGION`、`AMG_ENDPOINT`、`EKS_CLUSTER` および `AMP_ARN`、AWS リージョン 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 を使用します ARN。

```
export AWS_REGION=REGION
export AMG_ENDPOINT=AMG_ENDPOINT
export EKS_CLUSTER_NAME=EKS_CLUSTER
export AMP_WS_ARN=AMP_ARN
```



4. Grafana を呼び出すためのADMINアクセス権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HTTP APIs。詳細については、「[サービスアカウントを使用して Grafana HTTP APIs](#)」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。を置き換える必要があります `GRAFANA_ID` を Grafana ワークスペースの ID に指定します ( の形式になります `g-123example` )。このキーは 7,200 秒、つまり 2 時間後に期限切れになります。必要に応じて、時間 (`seconds-to-live`) を変更できます。デプロイには 1 時間未満かかります。

```
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 キーを に追加 AWS CDK することで、 API AWS Systems Manager キーを で使用できるようにします。置換 `AWS_REGION` ソリューションを ( の形式で) 実行する リージョンを使用します `us-east-1`。

```
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、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、同じリージョン内の他の Amazon EKS クラスター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、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。スクレイパーがメトリクスの収集を開始すると、メトリクスが表示されるまでに数分かかります。

## Terraform を使用したインストール

このソリューションが提供される方法の 1 つは、Terraform ソリューションとして提供される方法です。使用するリソースに関する情報を入力すると、ソリューションによってスクレイパー、ログ、ダッシュボードが作成されます。

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には

1. 前提条件の [???](#) ステップをすべて完了していることを確認します。
2. Amazon S3 からソリューションのすべてのファイル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/`、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、ディレクトリをデプロイ元のフォルダに変更します。

```
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変更する必要はありません。

3. コマンドライン環境 (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) で、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。置換 `REGION`、`AMG_ENDPOINT`、`EKS_CLUSTER`、`AMP_ARN` および `S3_ID`、新しいリソースをデプロイ AWS リージョンする (例: `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。

```
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 を呼び出すためのADMINアクセス権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HTTP APIs。詳細については、「[サービスアカウントを使用して Grafana HTTP APIs](#)」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。置き換える必要があります *GRAFANA\_ID* を Grafana ワークスペースの ID に指定します ( の形式になります g-123example )。このキーは 7,200 秒、つまり 2 時間後に期限切れになります。必要に応じて、時間 (seconds-to-live) を変更できます。デプロイには 1 時間未満かかります。

```
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` サービスアカウントの ID を入力します。

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、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。スクレイパーがメトリクスの収集を開始すると、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 に保存され、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によって公開されるすべての [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ソフトウェアをクラスターに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することで、メトリクスの数を減らすことができます](#)。

以下のメトリクスは、追加のソフトウェアがインストールされていない Amazon EKS クラスターの基本設定で、このソリューションで追跡されます。

メトリクス	説明/目的
aggregator_unavailable_apiservice	のゲージはAPIService、名前別に分類API Servicesされ、使用不可とマークされます。
apiserver_admission_webhook_admission_duration_seconds_bucket	アドミッションウェブフックのレイテンシーのヒストグラムを秒単位で表示し、名前で識別し、オペレーション、APIリソース、タイプ(検証または承認)ごとに分割します。
apiserver_current_inflight_requests	過去 1 秒のリクエストタイプあたりの、この API サーバーで現在使用されている処理中のリクエストの最大数。
apiserver_envelope_encryption_dek_cache_fill_percent	キャッシュされたによって現在占有されているキャッシュスロットの割合DEKs。
apiserver_flowcontrol_current_executing_requests	Priority and Fairness サブシステムの初期 ( の場合WATCH) または任意の ( 以外の場合WATCH) API 実行ステージのリクエスト数。
apiserver_flowcontrol_rejected_requests_total	Priority and Fairness サブシステムの初期 ( の場合WATCH) または任意の ( 以外の場合WATCH) API 実行ステージで拒否されたリクエストの数。
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最小数。
apiserver_flowcontrol_request_execution_seconds_bucket	Priority and Fairness サブシステムでのリクエスト実行の初期ステージ ( の場合WATCH) または任意の ( 以外の場合WATCH) API ステージの継続時間のバケット化されたヒストグラム。
apiserver_flowcontrol_request_queue_length_after_enqueue_count	Priority and Fairness サブシステムのリクエスト実行の初期ステージ ( の場合WATCH) または任意の ( 以外の場合WATCH) API ステージの数。

メトリクス	説明/目的
<code>apiserver_request</code>	API サーバーリクエストを示します。
<code>apiserver_requested_deprecated_apis</code>	API グループ、バージョン、リソース、サブリソース、および <code>removed_release</code> 別に分類された、リクエストAPIsされた非推奨のゲージ。
<code>apiserver_request_duration_seconds</code>	動詞、ドライラン値、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レスポンスレイテンシーの秒単位の分布。
<code>apiserver_request_duration_seconds_bucket</code>	動詞、ドライラン値、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レスポンスレイテンシー分布のバケット化されたヒストグラム。
<code>apiserver_request_slo_duration_seconds</code>	動詞、ドライラン値、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) レスポンスレイテンシーの秒単位の分布。
<code>apiserver_request_terminations_total</code>	API サーバーが自己防御で終了したリクエストの数。
<code>apiserver_request_total</code>	動詞、リハーサル値、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 API サーバーリクエストのカウンター。
<code>container_cpu_usage_seconds_total</code>	消費された累積 CPU 時間。
<code>container_fs_reads_bytes_total</code>	読み取りの累積バイト数。
<code>container_fs_reads_total</code>	完了した読み取りの累積数。
<code>container_fs_writes_bytes_total</code>	書き込まれたバイトの累積数。

メトリクス	説明/目的
container_fs_writes_total	完了した書き込みの累積数。
container_memory_cache	ページキャッシュメモリの合計。
container_memory_rss	のサイズRSS。
container_memory_swap	コンテナスワップの使用。
container_memory_working_set_bytes	現在のワーキングセット。
container_network_receive_bytes_total	受信した累積バイト数。
container_network_receive_packets_dropped_total	受信中にドロップされたパケットの累積数。
container_network_receive_packets_total	受信したパケットの累積数。
container_network_transmit_bytes_total	送信されたバイトの累積数。
container_network_transmit_packets_dropped_total	送信中にドロップされたパケットの累積数。
container_network_transmit_packets_total	送信されたパケットの累積数。
etcd_request_duration_seconds_bucket	各オペレーションとオブジェクトタイプの etcd リクエストのレイテンシーのバケット化されたヒストグラム。
go_goroutines	現在存在するゴルーティンの数。
go_threads	作成された OS スレッドの数。

メトリクス	説明/目的
kubelet_cgroup_manager_duration_seconds_bucket	cgroup manager オペレーションの継続時間のバケット化されたヒストグラム。メソッド別に分類されます。
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所要時間。メソッド別に分類されます。
kubelet_node_config_error	このメトリクスは、ノードで設定関連のエラーが発生した場合は true (1)、それ以外の場合は false (0) です。
kubelet_node_name	ノードの名前。カウントは常に 1 です。
kubelet_pleg_relist_duration_seconds_bucket	でポッドを再リストするための秒単位の期間を示すバケット化されたヒストグラムPLEG。
kubelet_pleg_relist_duration_seconds_count	でポッドを再一覧表示する時間を秒単位で表したものですPLEG。
kubelet_pleg_relist_interval_seconds_bucket	で再リストされるまでの間隔のバケット化されたヒストグラムPLEG。
kubelet_pod_start_duration_seconds_count	kubelet がポッドを初めて確認してからポッドの実行を開始するまでの時間の秒数。
kubelet_pod_worker_duration_seconds_bucket	1 つのポッドを同期するための秒単位のバケット化されたヒストグラム。オペレーションタイプ別に分類: 作成、更新、同期。
kubelet_pod_worker_duration_seconds_count	1 つのポッドを同期する秒単位の時間数。オペレーションタイプ別に分類: 作成、更新、同期。
kubelet_running_containers	現在実行中のコンテナの数。
kubelet_running_pods	実行中のポッドサンドボックスを持つポッドの数。

メトリクス	説明/目的
kubelet_runtime_operations_duration_seconds_bucket	ランタイムオペレーションの秒単位の期間を示すバケット化されたヒストグラム。オペレーションタイプ別に分類されます。
kubelet_runtime_operations_errors_total	オペレーションタイプ別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別のランタイムオペレーションの累積数。
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 を予約した後)。
kube_node_status_capacity	ノードで使用可能なリソースの合計量。
kube_pod_container_resource_limits (CPU)	コンテナによってリクエストされた制限リソースの数。
kube_pod_container_resource_limits (Memory)	コンテナによってリクエストされた制限リソースの数。
kube_pod_container_resource_requests (CPU)	コンテナによってリクエストされたリクエストリソースの数。
kube_pod_container_resource_requests (Memory)	コンテナによってリクエストされたリクエストリソースの数。
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はCPU、メモリ、および名前空間内のストレージなどのリソースの使用制限を適用します。
node_cpu	コアあたりのCPU使用量と合計使用量を含む、ノードの使用状況メトリクス。
node_cpu_seconds_total	各モードでCPUs費やされた の秒数。



メトリクス	説明/目的
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するのにかかった累積時間。
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	ノードによってディスクから読み取られた合計バイト数。
node_disk_written_bytes_total	ノードによってディスクに書き込まれた合計バイト数。
node_filesystem_avail_bytes	Kubernetes クラスター内のノードのファイルシステム上の使用可能なスペースのバイト数。
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CPUの使用状況の 1 分間の負荷平均。
node_load15	ノードCPUの使用状況の 15 分の負荷平均。
node_load5	ノードCPUの使用状況の 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によるバッファキャッシュに使用されるメモリの量。
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するメモリの量。
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の量。
node_memory_MemFree_bytes	ノードで使用可能な空きメモリの量。
node_memory_MemTotal_bytes	ノードで使用可能な物理メモリの合計量。
node_network_receive_bytes_total	ノードがネットワーク経由で受信した合計バイト数。

メトリクス	説明/目的
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た合計バイト数。
process_cpu_seconds_total	秒単位のユーザーとシステムの合計CPU所要時間。
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リクエストの数。
rest_client_request_duration_seconds_bucket	リクエストレイテンシーのバケット化されたヒストグラム。動詞とホストによって分類されます。
storage_operation_duration_seconds_bucket	ストレージオペレーションの継続時間のバケット化されたヒストグラム。
storage_operation_duration_seconds_count	ストレージオペレーションの継続時間の数。
storage_operation_errors_total	ストレージオペレーション中のエラーの累積数。
up	モニタリング対象ターゲット (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	ポリュームマネージャーによって管理されるポリュームの合計数。
workqueue_adds_total	ワークキューによって処理された追加の合計数。
workqueue_depth	現在のワークキューの深さ。
workqueue_queue_duration_seconds_bucket	項目がリクエストされるまでにワークキューに留まる秒単位の時間を示すバケット化されたヒストグラム。

メトリクス	説明/目的
workqueue_work_duration_seconds_bucket	ワークキューからの項目の処理にかかる秒単位のバケット化されたヒストグラム。

## 作成されたアラートのリスト

次の表に、このソリューションによって作成されたアラートを示します。アラートは Amazon Managed Service for Prometheus で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 ワークスペースのルール [設定ファイルを編集することで、ルールの追加や削除などのルールを変更](#) できます。

これら 2 つのアラートは、一般的なアラートとは少し異なる特殊なアラートです。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される情報が提供されます。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	説明と使用法
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ことを目的としたアラートです。このアラートは常に発せられるため、常に Alertmanager で発せられ、常にレシーバーに対して発せられる必要があります。これを通知メカニズムと統合して、このアラートが発せられていないときに通知を送信できます。例えば、DeadMansSnitchで統合を使用できます PagerDuty。
InfoInhibitor	これは、情報アラートを禁止するために使用されるアラートです。それ自体では、情報レベルのアラートは非常にノイズが多い可能性があります。他のアラートと組み合わせると関連性があります。このアラートは、severity=info アラートが発生するたびに発動し、重要度が別のアラート、warningまたは同じ名

アラート	説明と使用法
	前空間で発音criticalが開始されると発音を停止します。このアラートは null レシーバーにルーティングされ、でアラートを禁止するように設定する必要がありますseverity=info。

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のステータスが頻繁に変更される
NodeFilesystemSpaceFillingUp	warning	ファイルシステムは、今後 24 時間以内に領域が不足することが予測されます。
NodeFilesystemSpaceFillingUp	critical	ファイルシステムは、今後 4 時間以内に領域が不足することが予測されます。
NodeFilesystemAlmostOutOfSpace	warning	ファイルシステムの残りのスペースは 5% 未満です。
NodeFilesystemAlmostOutOfSpace	critical	ファイルシステムの残りのスペースは 3% 未満です。
NodeFilesystemFilesFillingUp	warning	ファイルシステムは、今後 24 時間以内に inode が不足すると予測されます。
NodeFilesystemFilesFillingUp	critical	ファイルシステムは、今後 4 時間以内に inode が不足すると予測されます。

アラート	緊急度	説明
NodeFilesystemAlmostOutOfFiles	warning	ファイルシステムの inode の残りは 5% 未満です。
NodeFilesystemAlmostOutOfFiles	critical	ファイルシステムの inode の残りは 3% 未満です。
NodeNetworkReceiveErrs	warning	ネットワークインターフェイスは、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は、多くの送信エラーを報告しています。
NodeHighNumberConntrackEntriesUsed	warning	conntrack エントリの数が制限に近づいています。
NodeTextFileCollectorScrapeError	warning	Node Exporter テキストファイルコレクターのスクレイプに失敗しました。
NodeClockSkewDetected	warning	クロックスキューが検出されました。
NodeClockNotSynchronizing	warning	クロックが同期しません。
NodeRAIDDegraded	critical	RAID 配列のパフォーマンスが低下している
NodeRAIDDiskFailure	warning	RAID 配列内の失敗したデバイス
NodeFileDescriptorLimit	warning	カーネルは間もなくファイル記述子の制限を使い果たすと予測されます。

アラート	緊急度	説明
NodeFileDescriptorLimit	critical	カーネルは間もなくファイル記述子の制限を使い果たすと予測されます。
KubeNodeNotReady	warning	ノードは準備できません。
KubeNodeUnreachable	warning	ノードに到達できません。
KubeletTooManyPods	info	Kubelet は容量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。
KubeletPlegDurationHigh	warning	Kubelet Pod Lifecycle Event Generator の再リストに時間がかかりすぎています。
KubeletPodStartupLatencyHigh	warning	Kubelet Pod の起動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。

アラート	緊急度	説明
KubeletDown	critical	ターゲットが Prometheus ターゲット検出から消えました。
KubeVersionMismatch	warning	実行中の Kubernetes コンポーネントの異なるセマンティックバージョン。
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。
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	集約された Kubernetes API がエラーを報告しました。
KubeAggregatedAPIDown	warning	集約された Kubernetes API がダウンしています。
KubeAPIDown	critical	ターゲットが Prometheus ターゲット検出から消えました。
KubeAPITerminatedRequests	warning	kubernetes apiserver は、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。

アラート	緊急度	説明
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 Inode がいっぱいです。
KubePersistentVolumeInodesFillingUp	warning	永続ボリューム 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。
KubeCPUOvercommit	warning	クラスターはCPUリソースリクエストを過剰にコミットしました。
KubeMemoryOvercommit	warning	クラスターはメモリリソースリクエストを過剰にコミットしました。
KubeCPUQuotaOvercommit	warning	クラスターはCPUリソースリクエストを過剰にコミットしました。
KubeMemoryQuotaOvercommit	warning	クラスターはメモリリソースリクエストを過剰にコミットしました。
KubeQuotaAlmostFull	info	名前空間のクォータは満杯になります。
KubeQuotaFullyUsed	info	名前空間クォータは完全に使用されます。
KubeQuotaExceeded	warning	名前空間のクォータが制限を超えました。



アラート	緊急度	説明
CPUThrottlingHigh	info	プロセスではCPUスロットリングが高くなります。
KubePodCrashLooping	warning	ポッドがクラッシュループしています。
KubePodNotReady	warning	ポッドは 15 分以上準備状態ではありませんでした。
KubeDeploymentGenerationMismatch	warning	ロールバックの可能性によるデプロイ生成の不一致
KubeDeploymentReplicasMismatch	warning	デプロイが予想されるレプリカ数と一致していません。
KubeStatefulSetReplicasMismatch	warning	StatefulSet は、予想されるレプリカ数と一致しません。
KubeStatefulSetGenerationMismatch	warning	StatefulSet ロールバックの可能性による生成の不一致
KubeStatefulSetUpdateNotRolledOut	warning	StatefulSet 更新はロールアウトされていません。
KubeDaemonSetRolloutStuck	warning	DaemonSet ロールアウトがスタックしています。
KubeContainerWaiting	warning	1 時間以上待機しているポッドコンテナ
KubeDaemonSetNotScheduled	warning	DaemonSet ポッドは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なかった

アラート	緊急度	説明
KubeJobFailed	warning	ジョブを完了できませんでした。
KubeHpaReplicasMismatch	warning	HPA は、必要な数のレプリカと一致しません。
KubeHpaMaxedOut	warning	HPA は最大レプリカで実行されています
KubeStateMetricsListErrors	critical	kube-state-metrics はリストオペレーションでエラーが発生しています。
KubeStateMetricsWatchErrors	critical	kube-state-metrics は監視オペレーションでエラーが発生しています。
KubeStateMetricsShardingMismatch	critical	kube-state-metrics シャーディングの設定が間違っています。
KubeStateMetricsShardsMissing	critical	kube-state-metrics シャードがありません。
KubeAPIErrorBudgetBurn	critical	API サーバーが書き込み中のエラー予算が多すぎます。
KubeAPIErrorBudgetBurn	critical	API サーバーが書き込み中のエラー予算が多すぎます。
KubeAPIErrorBudgetBurn	warning	API サーバーが書き込み中のエラー予算が多すぎます。
KubeAPIErrorBudgetBurn	warning	API サーバーが書き込み中のエラー予算が多すぎます。

アラート	緊急度	説明
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。
etcdHighNumberOfLeaderChanges	warning	etcd クラスターのリーダーの変更数が多い。
etcdNoLeader	critical	etcd クラスターにはリーダーがありません。
etcdHighNumberOfFailedGRPCRequests	warning	etcd クラスターの多数の失敗した gRPC リクエスト。
etcdGRPCRequestsSlow	critical	etcd クラスター gRPC リクエストは遅くなります。
etcdMemberCommunicationSlow	warning	etcd クラスターメンバーの通信が遅い。
etcdHighNumberOfFailedProposals	warning	etcd クラスターは、失敗した提案の数が多いです。
etcdHighFsyncDurations	warning	etcd クラスターの高同期時間。
etcdHighCommitDurations	warning	etcd クラスターのコミット時間が予想よりも長い。
etcdHighNumberOfFailedHTTPRequest	warning	etcd クラスターがHTTPリクエストに失敗しました。
etcdHighNumberOfFailedHTTPRequests	critical	etcd クラスターは、失敗したHTTPリクエストの数が多いです。

アラート	緊急度	説明
etcdHTTPRequestsSlow	warning	etcd クラスターHTTPリクエストは遅くなります。
HostClockNotSynchronizing	warning	ホストクロックが同期しません。
HostOomKillDetected	warning	ホストOOMの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となるいくつかのことがあります。必ず以下を確認してください。

- ソリューションをインストールする前に、[すべての前提条件](#)を完了する必要があります。
- ソリューションを作成したり、メトリクスにアクセスしたりする前に、クラスターに少なくとも1つのノードが必要です。
- Amazon EKSクラスターには、AWS CNI、CoreDNSおよび kube-proxyアドオンがインストールされている必要があります。インストールされていない場合、ソリューションが正しく動作しません。これらは、コンソールを使用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` を通じ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 AWS SDK。
- 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場合に発生する可能性があります。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クラスターは、Amazon ではなく Fargate で初期化されましたEC2。このプロジェクトには Amazon が必要ですEC2。
  - ノードは[テイント](#)されているため、使用できません。

```
kubectl describe node NODENAME | grep Taints
```

を使用してテイントを確認できます。次に `kubectl taint node NODENAME TAINT_NAME-`、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、新しいノードを作成するか、容量を増やすことができます。

- 間違った Grafana ワークスペース ID を使用しても、Grafana にダッシュボードは表示されません。

次のコマンドを実行して、Grafana に関する情報を取得します。

```
kubectl describe grafanas external-grafana -n grafana-operator
```

正しいワークスペースの結果を確認できますURL。想定どおりのものでない場合は、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 には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。

この場合を検索するには、grafana 演算子を取得し、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
```

これにより、次のような演算子名が返されます。

NAME	READY	STATUS	RESTARTS	AGE
<i>grafana-operator-1234abcd5678ef90</i>	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ている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、新しいAPIキーを作成し、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、再デプロイする前に次のコマンドを使用して同期を強制できます。

```
kubectl delete externalsecret/external-secrets-sm -n grafana-operator
```

- CDK installs – 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た場合は、 を実行して `cdk bootstrap`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、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のインストールの場合)。

```
| 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、IAMポータルに移動してOIDCプロバイダーを削除し、もう一度試してください。

- Terraform のインストール — `cluster-secretstore-sm failed to create kubernetes rest client for update of resource`とを含むエラーメッセージが表示されます `failed to create kubernetes rest client for update of resource`.

このエラーは通常、Kubernetes クラスターに External Secrets Operator がインストールされていないか、有効になっ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の一部として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ない場合があります。

次のコマンドを使用して、インストールされている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、オペレーターが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して、必要なカスタムリソース定義 (CRDs) 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 など、外部シークレット演算子CRDsに関連するを一覧表示する必要があります `externalsecrets.external-secrets.io`。リストにない場合は、数分待つてからもう一度確認してください。

CRDs 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 リソースにタグを追加する

タグは、AWS 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 (、 、 111122223333Production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 のサービスではタグ付けがサポートされているため、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 [AWS 「リソースのタグ付け」](#) 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、AWS CLI、APIs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。SDKs タグを使用してワークスペースを識別、整理、追跡するだけでなく、IAM ポリシーのタグを使用して、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こともできます。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。

## 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 サービスや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。
- aws:、AWS:、またはキーワードの大文字または小文字の組み合わせをキーまたは値のプレフィックスAWSとして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が付いたタグは、制限には tags-per-resource 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 [AWS 「リソースのタグ付け」](#) を参照してください。

## Amazon Managed Grafana ワークスペースにタグを付ける

Amazon Managed Grafana ワークスペースのタグを操作するには、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 リソースの特定と整理に役立ちます。まず、ワークスペースに 1 つ以上のタグを追加してから、これらのタグに基づいてワークスペースへのアクセスを管理する IAM ポリシーを作成できます。コンソールまたは AWS CLI 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。

作成時に Amazon Managed Grafana ワークスペースにタグを追加する方法の詳細については、[Amazon Managed Grafana ワークスペースを作成する](#)「Amazon Managed Grafana ユーザーガイド」の「」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 1 つ以上のタグを追加できます。

1. <https://console.aws.amazon.com/grafana/> で 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. 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、タグの名前を入力します。値 フィールドにタグのオプションの値を追加できます。
8. ( オプション) 別のタグを追加するには、タグの追加 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

AWS CLI 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、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と値を指定します。ワークスペースには複数のタグを追加できます。例えば、Grafana ワークスペースの *My-Workspace* に *Name* という名前のタグキーに *my\_key\_value* のタグ値でタグ付けするには、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/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. <https://console.aws.amazon.com/grafana/> で 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 WorkSpaces] 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

AWS CLI 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リストを表示するには、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 を使用して、ワークスペースに関連付けられたタグの値を 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には、UntagResource と TagResource APIs を組み合わせる必要があります。

### 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に影響を与える可能性があります。ワークスペースのタグを編集する前に、タグのキーまたは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た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 WorkSpaces] 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、タグの名前を入力します。値フィールドにタグのオプションの値を追加できます。
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するか、別のキーを追加できます。

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、タグを更新してタグキーとタグ値を指定する Amazon Managed Grafana ワークスペースの Amazon リソースネーム (ARN) を指定します。

例えば、タグの値を新しい値 `Key_value_new` で変更するには、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"
```

新しい名前 `Name_new` でタグのキーを変更するには、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する可能性があります。ワークスペースからタグを削除する前に、タグのキーまたは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 WorkSpaces] 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、 の削除を選択します。
8. タグの削除が完了したら、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、他のリソースからタグは削除されませんが、タグとワークスペース間の関連付けは削除されます。

### 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すべてのタグの関連付けが削除されます。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、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、`space-IDstring` という名前のワークスペースでタグキー を持つ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
```

成功すると、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た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、AWS とユーザーの間で共有される責任です。[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 サービスを実行するインフラストラクチャを保護する責任 AWS を担います AWS。AWS また、は、お客様が安全に使用できるサービスも提供します。コンプライアンス [AWS プログラム](#) コンプライアンスプログラムの一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。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、お客様は、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 を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するように 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や保護に役立つ他の AWS の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 の Identity and Access Management](#)
- [AWS データソースの Amazon Managed Grafana のアクセス許可とポリシー](#)
- [IAM アクセス許可](#)
- [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](#)
- [を使用した Amazon Managed Grafana API呼び出しのログ記録 AWS CloudTrail](#)

- [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

AWS [責任共有モデル](#)、Amazon Managed Grafana でのデータ保護に適用されます。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。お客様は、このインフラストラクチャでホストされてい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 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FAQ](#)」を参照してください。欧州におけるデータ保護の詳細については、AWS 「セキュリティブログ」の[AWS 「責任共有モデル」とGDPR](#)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 ( ) 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IAM。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。1TLS.2 が必要で、1.3 TLS をお勧めします。
- を使用して 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。
- AWS 暗号化ソリューションと、内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 サービス。
- Amazon Macie などの高度なマネージド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 AWS を介して にアクセスするときに FIPS 140-3 検証済みの暗号化モジュールが必要な場合はAPI、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は、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配置しないことを強くお勧めします。これは、コンソール、または を使用して Amazon Managed Grafana または他の AWS サービス を使用する場合API AWS CLIも同様です AWS SDKs。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。URL を外部サーバーに提供する場合は、そのサーバーへのリクエストを検証URLするために認証情報を 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 は、次のタイプのデータを収集して保存します。

- Grafana ワークスペース用にお客様が提供するダッシュボードとアラート設定。
- ワークスペースに保存した 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、 で暗号化されます AWS Key Management Service。転送中のデータは、Secure Sockets Layer () で自動的に暗号化されますSSL。

## Amazon Managed Grafana 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 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する (アクセス許可を付与する) かを制御します。IAM は追加料金なしで AWS サービス 使用できる 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 の連携方法 IAM](#)
- [Amazon Managed Grafana のアイデンティティベースのポリシーの例](#)
- [AWS Amazon Managed Grafana の マネージドポリシー](#)
- [Amazon Managed Grafana の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 のサービスにリンクされたロールの使用](#)

### 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 – Amazon Managed Grafana サービスを使用してジョブを実行する場合、管理者から必要な認証情報とアクセス許可が与えられます。さらに多くの Amazon Managed Grafana 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解しておく、管理者に適切な許可をリクエストするうえで役立ちます。Amazon Managed Grafana の機能にアクセスできない場合は、「」を参照してください[Amazon Managed Grafana の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。次に、サービスユーザーのアクセス許可を変更するリクエストをIAM管理者に送信する必要があります。このページの情報を確認して、の基本概念を理解してくださいIAM。会社で Amazon Managed Grafana IAM を使用する方法の詳細については、「」を参照してください[Amazon Managed Grafana との連携方法 IAM](#)。

IAM 管理者 – IAM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。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IAM、「」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の例](#)。

## アイデンティティを使用した認証

認証とは、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 の認証情報は、フェデレーテッド ID の例です。フェデレーテッド ID としてサインインすると、管理者は以前に IAMロールを使用して ID フェデレーションをセットアップしていました。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となります。

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、「ユーザーガイド」の「[へのサインイン AWS アカウント](#)方法AWS サインイン」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に暗号で署名します。AWS

ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、「IAMユーザーガイド」の[AWS API「リクエストの署名」](#)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。例えば、AWS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[「ユーザーガイド」の「での多要素認証 \(MFA\) AWS IAM の使用」](#)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 AWS サービス およびリソースへの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[「ルートユーザーの認証情報を必要とするタスク」](#)を参照してください。

## フェデレーテ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、一時的な認証情報を使用してにアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS サービスします。

フェデレーテッド 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のユーザー、または ID ソースを通じて提供された認証情報 AWS 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ことも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用できるようにすることもできます。IAM Identity Center の詳細については、「ユーザーガイド」の[IAM「Identity Center」とはAWS IAM Identity Center](#)」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アイデンティティです。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持つ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ユーザーとの長期的な認証情報を必要とする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、「[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でアクセスキーを定期的にローテーションするIAM](#)」を参照してください。

[IAM グループ](#)は、IAMユーザーのコレクション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、という名前のグループを作成しIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

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、「[ユーザーガイド](#)」のIAM「[\(ロールではなく\)ユーザーを作成する場合IAM](#)」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アイデンティティです。ユーザーと似ていますがIAM、特定のユーザーに関連付けられていません。IAM ロール を切り替える AWS Management Console ことで、[でロール](#)を一時的に引き受けることができます。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 を使用しますURL。ロールの使用の詳細については、「[ユーザーガイド](#)」のIAM「[ロールの使用IAM](#)」を参照してください。

IAM 一時的な認証情報を持つ ロールは、以下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の詳細については、「[ユーザーガイド](#)」の「[サードパーティー ID プロバイダーのロールの作成IAM](#)」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証後にアクセスできる内容を制御するために、IAM Identity Center はアクセス

許可セットをのロールに関連付けますIAM。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「[アクセス許可セット](#)」を参照してください。

- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、IAMロールを引き受けて、特定のタスクに対して異なるアクセス許可を一時的に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、別のアカウントのユーザー (信頼されたプリンシパル) が自分のアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の AWS サービス、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、「[ユーザーガイド](#)」の「[でのクロスアカウントリソースアクセスIAMIAM](#)」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 サービス を使用します AWS サービス。例えば、サービスで呼び出しを行うと、そのサービスが Amazon でアプリケーションを実行 EC2したり、Amazon S3 にオブジェクトを保存したりするのが一般的です。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 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を AWS サービス、ダウンストリームサービス AWS サービスへのリクエストのリクエスト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 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権限が必要です。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 – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける [IAMロール](#) です。IAM 管理者は、内からサービスロールを作成、変更、削除できますIAM。詳細については、「[ユーザーガイド](#)」の「[にアクセス許可を委任するロールの作成 AWS サービスIAM](#)」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 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、サービスにリンクされた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 で実行されているアプリケーション EC2 – 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インスタンス内にアクセスキーを保存するよりも望ましいです。AWS ロールをEC2インスタンスに割り当て、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、ロールが含まれており、EC2インスタンスで実行されているプログラムが一時的な認証情報を取得できるようにします。詳細については、「[ユーザーガイド](#)」の「[IAMロールを使用して Amazon EC2インスタンスで実行されているアプリケーションにアクセス許可を付与するIAM](#)」を参照してください。

IAM ロールとIAMユーザーのどちらを使用するかについては、「[ユーザーガイド](#)」の「[\(ユーザーではなく\) IAMロールを作成する場合IAM](#)」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、AWS ID 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、「[ユーザーガイド](#)」の[JSON「ポリシーの概要IAM](#)」を参照してください。

管理者はポリシーを使用して AWS JSON、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。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するために、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、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に使用するメソッドに関係なく、アクションのアクセス許可を定義します。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 からロール情報を取得できますAPI。

## アイデンティティベースのポリシー

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできる JSONアクセス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のポリシーを作成する方法については、「ユーザーガイド」の[IAM「ポリシーの作成IAM」](#)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。管理ポリシーとインラインポリシーのどちら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選択」](#)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、リソースにアタッチする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の例としては、IAMロールの信頼ポリシーや Amazon S3 バケットポリシーなどがあります。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コントロールできます。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、フェデレーテッドユーザー、またはを含めることができます AWS 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IAMでは、の AWS 管理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 (ACLs )

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs 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を使用しません。

Amazon S3、AWS WAF、および Amazon VPCは、をサポートするサービスの例ですACLs。の詳細についてはACLs、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、一般的ではない追加のポリシータイプをサポートします。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のポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できるアクセス許可の上限を設定する高度な機能です。工

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) に対する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、AWS アカウント 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 (SCPs) をアカウントの一部またはすべてに適用できます。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SCPを制限します AWS アカウントのルートユーザー。Organizations との詳細についてはSCPs、「AWS Organizations ユーザーガイド」の「[サービスコントロールポリシー](#)」を参照してください。
- 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があります。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、「ユーザーガイド」の「[セッションポリシーIAM](#)」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。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、ユーザーガイドの「[ポリシー評価ロジックIAM](#)」を参照してください。

## Amazon Managed Grafana と の連携方法 IAM

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機能を確認してください。



## IAM Amazon Managed Grafana で使用できる の機能

IAM 機能	Amazon Managed Grafana のサポート
<a href="#">アイデンティティベースのポリシー</a>	あり
<a href="#">リソースベースのポリシー</a>	なし
<a href="#">ポリシーアクション</a>	あり
<a href="#">ポリシーリソース</a>	Yes
<a href="#">ポリシー条件キー</a>	なし
<a href="#">ACLs</a>	なし
<a href="#">ABAC (ポリシー内のタグ )</a>	あり
<a href="#">一時的な認証情報</a>	あり
<a href="#">転送アクセスセッション (FAS )</a>	あり
<a href="#">サービスロール</a>	あり
<a href="#">サービスリンクロール</a>	あり

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 「と連携する のサービスIAM」](#)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

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできる JSONアクセス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のポリシーを作成する方法については、「ユーザーガイド」の[IAM 「ポリシーの作成IAM」](#)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、許可または拒否されたアクションとリソース、およびアクションが許可または拒否され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は、「ユーザーガイド」の「[IAMJSONポリシー要素のリファレンスIAM](#)」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の例

Amazon Managed Grafana の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、「」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の例](#)。

## 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、リソースにアタッチする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の例としては、IAMロールの信頼ポリシー や Amazon S3 バケットポリシー などがあります。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コントロールできます。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、フェデレーテッドユーザー、またはを含めることができます AWS サービス。

クロスアカウントアクセスを有効にするには、リソースベースのポリシーのプリンシパルとして、アカウント全体または別のアカウントのIAMエンティティを指定できます。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 がある場合 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、「ユーザーガイド」の「[でのクロスアカウントリソースアクセスIAMIAM](#)」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

管理者はポリシーを使用して AWS JSON、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 要素は、ポリシーで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を記述します。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 オペレーションがないアクセス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権限を付与するポリシーで使用されます。

Amazon Managed Grafana アクションのリストを確認するには、「サービス認証リファレンス」の「[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 のポリシーアクションは、アクションの前に次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 の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、「」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の例](#)。

## 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

管理者はポリシーを使用して AWS JSON、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。

Policy ResourceJSON 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 リソースタイプとその のリストを確認するにはARNs、「サービス認証リファレンス」の [「Amazon Managed Grafana で定義されるリソース」](#) を参照してください。各リソースARNの を指定できるアクションについては、 [「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 の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、「」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の例](#)。

## 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をサポート： いいえ

管理者はポリシーを使用して AWS JSON、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することで、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、リソースにIAMユーザー名でタグ付けされている場合にのみ、リソースへのアクセス許可をIAMユーザーに付与できます。詳細については、「ユーザーガイド」の [IAM 「ポリシー要素: 変数とタグIAM」](#) 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ます。すべての AWS グローバル条件キーを確認するには、「ユーザーガイド」の [AWS 「グローバル条件コンテキストキーIAM」](#) 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 (ACLs )

をサポートACLs : いいえ

アクセスコントロールリスト (ACLs) は、リソースへのアクセス許可を持つプリンシパル (アカウントメンバー、ユーザー、またはロール) を制御します。ACLs はリソーススペ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を使用しません。

## Amazon Managed Grafana による属性ベースのアクセスコントロール (ABAC )

サポート ABAC (ポリシー内のタグ): はい

Amazon Managed Grafana は、リソースと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、「」を参照してください [Amazon Managed Grafana ワークスペースにタグを付ける](#)。

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コントロール (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、の最初のステップですABAC。次に、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する場合に、オペレーションを許可するABACポリシーを設計します。

ABAC は、急速に成長している環境や、ポリシー管理が煩雑になる状況に役立ちます。

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。

の詳細についてはABAC、「ユーザーガイド」の「[とはABACIAM](#)」を参照してください。のセットアップ手順を含むチュートリアルを表示するにはABAC、「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IAM」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 は、一時的な認証情報を使用してサインインすると機能 AWS サービス しません。一時的な認証情報 AWS サービス を使用する などの詳細については、「[ユーザーガイドAWS サービス](#)」の「[と連携IAMする IAM](#)」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。ロールの切り替えの詳細については、「IAMユーザーガイド」の「[ロールへの切り替え\(コンソール\)](#)」を参照してください。

一時的な認証情報は、AWS CLI または を使用して手動で作成できます AWS API。その後、これらの一時的な認証情報を使用して . AWS recommends にアクセスできます AWS。これは、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することを推奨しています。詳細については、「[」の「一時的なセキュリティ認証情報IAM](#)」を参照してください。

## Amazon Managed Grafana の転送アクセスセッション

転送アクセスセッションをサポート (FAS): はい

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を AWS サービス、ダウンストリームサービス AWS サービス へのリクエストのリクエスト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 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権限が必要です。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

サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受ける [IAM ロール](#)です。IAM 管理者は、内からサービスロールを作成、変更、削除できますIAM。詳細については、「ユーザーガイド」の「[にアクセス許可を委任するロールの作成 AWS サービスIAM](#)」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 の機能が破損する可能性があります。Amazon Managed Grafana が指示する場合以外は、サービスロールを編集しないでください。

## Amazon Managed Grafana のサービスにリンクされたロール

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 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、サービスにリンクされた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 サービスにリンクされたロールの作成または管理の詳細については、「」を参照してください [Amazon Managed Grafana のサービスにリンクされたロールの使用](#)。

## Amazon Managed Grafana のアイデンティティベースのポリシーの例

デフォルトでは、ユーザーとロールには Amazon Managed Grafana リソースを作成または変更するアクセス許可はありません。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受けることができます。

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 ポリシーの作成](#)」を参照してください。

各リソースタイプの ARNs」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

- [ポリシーの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](#)

## ポリシーの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、または削除できるか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を付与するには、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して、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「[AWS ジョブ機能の管理ポリシー](#)」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。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、最小特権権限とも呼ばれています。IAM を使用して権限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 でのポリシーと権限](#)」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、などの特定の を介してサービスアクションが使用される場合に AWS 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 policy elements: Condition](#)」(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 ポリシーの検証](#)」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、セキュリティを強化するために MFA を有効にします。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 保護 API アクセスの設定](#)」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、のリソースの詳細を一覧表示および表示できます AWS アカウント。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アイデンティティ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ための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のAWSMarketplaceManageSubscriptionsポリシーまたは同等のアクセス許可。

### IAM Identity Center を使用した管理アカウントの 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、AWSGrafanaAccountAdministratorAWSSSOMasterAccountAdministratorと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 IAM AWSMarketplaceManageSubscriptions 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アクセス許可を使用する場合は、ワークスペースを作成するユーザーにも iam:CreateRole、iam:CreatePolicy、およびアクセスiam:AttachRolePolicy許可が必要です。これらは、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ができるポリシーをデプロイ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。

に付与されたアクセス許可を確認するには `AWSGrafanaAccountAdministrator`、「」を参照してください。 [AWS マネージドポリシー：AWSGrafanaAccountAdministrator](#)

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` と `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 IAM `AWSMarketplaceManageSubscriptions` 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アクセス許可を使用する場合は、ワークスペースを作成するユーザーにも `iam:CreateRole`、`iam:CreatePolicy`、および `iam:AttachRolePolicy` アクセス許可が必要です。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れるように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。

に付与されたアクセス許可を確認するには `AWSGrafanaAccountAdministrator`、「」を参照してください。 [AWS マネージドポリシー：AWSGrafanaAccountAdministrator](#)

IAM Identity Center を使用して、単一の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します。

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。組織の詳細については、[「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アクセス許可を付与するに

は、`AWSGrafanaAccountAdministrator`、`AWSSSOMasterAccountAdministratorAWSOrganizationsFullAccess` 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 `IAM AWSMarketplaceManageSubscriptions` 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。

に付与されたアクセス許可を確認するには `AWSGrafanaAccountAdministrator`、[「」を参照してください。AWS マネージドポリシー：AWSGrafanaAccountAdministrator](#)

Amazon Managed Grafana へのユーザーアクセスの割り当てと割り当て解除

ワークスペースのそれらのユーザーに 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、そのユーザーに `AWSGrafanaWorkspacePermissionManagementV2` のポリシーを割り当てます。IAM Identity Center を使用してこのワークスペースのユーザーを管理する場合、ユーザーには `AWSSSOReadOnly` および `AWSSSODirectoryReadOnly` ポリシーも必要です。

`AWSGrafanaWorkspacePermissionManagementV2` に付与されたアクセス許可を確認するには、[「」を参照してください。AWS 管理ポリシー：AWSGrafanaWorkspacePermissionManagementV2](#)

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表示と表示、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。

に付与されたアクセス許可を確認するには、`AWSGrafanaConsoleReadOnlyAccess` 「」を参照してください [AWS 管理ポリシー：AWSGrafanaConsoleReadOnlyAccess](#)。

## 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。ユースケース別に [カスタマー マネージドポリシー](#) を定義して、マネージドポリシー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 管理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) が更新されます。は、新しい AWS サービスが起動されたとき、または既存のサービスで新しい API AWS オペレーションが使用可能になったときに、AWS 管理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、Amazon Managed Grafana 内でアクセスを提供し、組織全体のアカウントとワークスペースを作成および管理します。

IAM エンティティ `AWSGrafanaAccountAdministrator` に をアタッチできます。

#### 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。

- `iam` — 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。これにより、管理者はロールをワークスペースに関連付けたり、ロールを Amazon Managed Grafana サービスに渡すことができます。
- Amazon Managed Grafana – プリンシパルにすべての Amazon Managed Grafana APIs。

```
{
  "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 管理ポリシー: AWSGrafanaWorkspacePermissionManagement ( 廃止 )

このポリシーは廃止されています。このポリシーは、新しいユーザー、グループ、またはロールにアタッチしないでください。

Amazon Managed Grafana は、このポリシーを置き換える新しいポリシー

[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することで、ワークスペースのセキュリティを向上させます。

## AWS 管理ポリシー : AWSGrafanaWorkspacePermissionManagementV2

AWSGrafanaWorkspacePermissionManagementV2 ポリシーは、Amazon Managed Grafana ワークスペースのユーザーおよび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。

IAM エンティティに AWSGrafanaWorkspacePermissionManagementV2 をアタッチできます。

### 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。

- 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およびグループのアクセス許可を読み取って更新できるようにします。
- IAM Identity Center — プリンシパルが IAM Identity Center エンティティを読み取ることを許可します。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必要な部分ですが、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": "*"
}
]
```

## 追加のポリシーが必要

ユーザーに許可の割り当てを完全に許可するに

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。

このポリシーを作成するには、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. 左側のメニューからアプリケーションを選択します。
3. AWS マネージドタブで、Amazon Grafana-workspace-name というアプリケーションを見つけます。ここで、workspace-nameはワークスペースの名前です。アプリケーション名を選択します。
4. Amazon Managed Grafana が管理するワークスペース用の 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 の形式になります `arn:aws:sso::owner-account-id:application/ssoins-unique-id/apl-unique-id`。

作成するポリシーは次のようになります。を、前のステップで確認した ARN `grafana-application-arn`に置き換えます。

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "sso:CreateApplicationAssignment",  
      "sso>DeleteApplicationAssignment"  
    ],  
    "Resource": [  
      "grafana-application-arn"  
    ]  
  }  
]
```

ロールまたはユーザーにポリシーを作成して適用する方法については、[ユーザーガイドの「IAM ID アクセス許可の追加と削除」](#)を参照してください。

## AWS 管理ポリシー : 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許可します。

IAM エンティティ AWSGrafanaConsoleReadOnlyAccess に をアタッチできます。

### 許可の詳細

このポリシーには、次のアクセス許可が含まれます。

- Amazon Managed Grafana — Amazon Managed Grafana APIs

```
{  
  "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 へのスコープ付きアクセスを許可し、Amazon Managed Grafana. AmazonGrafanaRedshiftAccess policy で Amazon Redshift プラグインを使用するために必要な依存関係により、ユーザーまたは IAM ロールが Grafana で 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的な認証情報はデータベースユーザーに限定redshift\_data\_api\_userされ、シークレットがキーで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RedshiftQueryOwner。このポリシーは、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GrafanaDataSource。カスタマー管理ポリシーを作成する場合、タグベースの認証はオプションです。

IAM エンティティ AmazonGrafanaRedshiftAccess に をアタッチできます。Amazon Managed Grafana は、Amazon Managed Grafana がユーザーに代わってアクションを実行できるようにするサービスロールにもこのポリシーをアタッチします。

### 許可の詳細

このポリシーには、次のアクセス許可が含まれます。

- Amazon Redshift — プリンシパルがクラスターを記述し、 という名前のデータベースユーザーの一時的な認証情報を取得できるようにしますredshift\_data\_api\_user。
- Amazon Redshift-data — プリンシパルが 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GrafanaDataSource。
- Secrets Manager – プリンシパルがシークレットを一覧表示し、 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RedshiftQueryOwner。

```
{
  "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、Amazon Managed Grafana の Athena プラグインから Amazon S3 へのクエリと結果の書き込みを有効にするために必要な依存関係と Athena へのアクセスを許可します。AmazonGrafanaAthenaAccess ポリシーは、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することを許可します。Athena ワークグループ GrafanaDataSource に

アクセスするには、タグ付けする必要があります。このポリシーには、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。grafana-athena-query-results-。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 アクセス許可は、このポリシーに含まれません。

IAM エンティティに AWSGrafanaAthenaAccess ポリシーをアタッチできます。Amazon Managed Grafana は、Amazon Managed Grafana がユーザーに代わってアクションを実行できるようにするサービスロールにもこのポリシーをアタッチします。

## 許可の詳細

このポリシーには、次のアクセス許可が含まれます。

- Athena — プリンシパルがとして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 GrafanaDataSource。
- Amazon S3 — プリンシパルが、プレフィックスが のバケットに対してクエリ結果の読み取りと書き込みを行うことができます grafana-athena-query-results-。
- 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 Glue データカタログを使用 AWS できるようにするために必要です。

```
{
  "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 へのアクセス CloudWatch と、Amazon Managed Grafana 内のデータソース CloudWatch として を使用するために必要な依存関係を付与します。

IAM エンティティに AWSGrafanaCloudWatchAccess ポリシーをアタッチできます。Amazon Managed Grafana は、Amazon Managed Grafana がユーザーに代わってアクションを実行できるようにするサービスロールにもこのポリシーをアタッチします。

### 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 からメトリクスデータとログを一覧表示して取得できるようにします CloudWatch。また、ソースアカウントから共有されたデータを CloudWatch クロスアカウントオブザーバビリティで表示することもできます。
- Amazon EC2 — プリンシパルがモニタリング対象のリソースに関する詳細を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": "*"
}
]
```

## マネージド AWS ポリシーに対する Amazon Managed Grafana の更新

Amazon Managed Grafana の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページの変更に関する自動アラートを受け取るには、[Amazon Managed Grafana ドキュメント履歴ページの 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
<a href="#">AWSGrafanaWorkspacePermissionManagement</a> – 廃止	<p>このポリシーは AWSGrafanaWorkspacePermissionManagementV2 に置き換えられました。</p> <p>このポリシーは古いと見なされ、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を提供することで、ワークスペースのセキュリティが向上します。</p>	2024 年 1 月 5 日
<a href="#">AWSGrafanaWorkspacePermissionManagementV2</a> – 新しいポリシー	Amazon Managed Grafana は、古いポリシーを置き換えAWSGrafanaWorkspacePermissionManagementV2る新しいAWSGrafanaWorkspacePermissionManagementポリシーを追加しました。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することで、ワークスペースのセキュリティを向上させます。	2024 年 1 月 5 日
<a href="#">AmazonGrafanaCloudWatchAccess</a> - 新しいポリシー	Amazon Managed Grafana に新しいポリシー が追加され	2023 年 3 月 24 日

変更	説明	日付
	ましたAmazonGrafanaCloud WatchAccess。	
<a href="#">AWSGrafanaWorkspac ePermissionManagement</a> – 既存ポリシーへの更新	<p>Amazon Managed Grafana は、Active Directory の IAM Identity Center ユーザーとグループAWSGrafanaWorkspac ePermissionManagementを Grafana ワークスペースに関連付けることができるように、新しいアクセス許可を に追加しました。</p> <p>次のアクセス許可が追加されました: sso-directory:DescribeUser 、および sso-directory:DescribeGroup</p>	2023 年 3 月 14 日



変更	説明	日付
<p><a href="#">AWSGrafanaWorkspacePermissionManagement</a> – 既存ポリシーへの更新</p>	<p>Amazon Managed Grafana は、IAM Identity Center のユーザーとグループでAWSGrafanaWorkspacePermissionManagementをGrafana ワークスペースに関連付けることができるように、新しいアクセス許可を追加しました。</p> <p>次のアクセス許可が追加されました: <code>sso:DescribeRegisteredRegions</code>、<code>sso:GetSharedSsoConfiguration</code>、<code>sso:ListDirectoryAssociations</code>、<code>sso:GetManagedApplicationInstance</code>、<code>sso:ListProfiles</code>、<code>sso:AssociateProfile</code>、<code>sso:DisassociateProfile</code>、<code>sso:GetProfile</code>、<code>sso:ListProfileAssociations</code>。</p>	2022 年 12 月 20 日
<p><a href="#">AmazonGrafanaServiceLinkedRolePolicy</a> – 新しい SLR ポリシー</p>	<p>Amazon Managed Grafana は、Grafana サービスにリンクされたロールの新しいポリシーを追加しましたAmazonGrafanaServiceLinkedRolePolicy。</p>	2022 年 11 月 18 日

変更	説明	日付
<a href="#">AWSGrafanaAccountAdministrator</a> , <a href="#">AWSGrafanaConsoleReadOnlyAccess</a>	すべての Amazon Managed Grafana リソースへのアクセスを許可する	2022 年 2 月 17 日
<a href="#">AmazonGrafanaRedshiftAccess</a> - 新しいポリシー	Amazon Managed Grafana に新しいポリシーが追加されましたAmazonGrafanaRedshiftAccess。	2021 年 11 月 26 日
<a href="#">AmazonGrafanaAthenaAccess</a> - 新しいポリシー	Amazon Managed Grafana に新しいポリシーが追加されましたAmazonGrafanaAthenaAccess。	2021 年 11 月 22 日
<a href="#">AWSGrafanaAccountAdministrator</a> - 既存ポリシーへの更新。	<p>Amazon Managed Grafana からアクセス許可を削除しましたAWSGrafanaAccountAdministrator。</p> <p>sso.amazonaws.com サービスの対象となるiam:CreateServiceLinkedRole            アクセス許可が削除されたため、代わりにこのアクセス許可をユーザーに付与するAWSSSOMasterAccountAdministratorポリシーをアタッチすることをお勧めします。</p>	2021 年 10 月 13 日

変更	説明	日付
<p><a href="#">AWSGrafanaWorkspacePermissionManagement</a> – 既存ポリシーへの更新</p>	<p>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AWSGrafanaWorkspacePermissionManagementできるように、新しいアクセス許可を に追加しました。</p> <p>アクセスgrafana:DescribeWorkspaceAuthentication 許可が追加されました。</p>	2021 年 9 月 21 日
<p><a href="#">AWSGrafanaConsoleReadOnlyAccess</a> – 既存ポリシーへの更新</p>	<p>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AWSGrafanaConsoleReadOnlyAccessできるように、新しいアクセス許可を に追加しました。</p> <p>grafana:Describe* および grafana:List* のアクセス許可がポリシーに追加され、以前のより狭いアクセス許可 grafana:DescribeWorkspace、grafana:ListPermissions、および が置き換えられましたgrafana:ListWorkspaces。</p>	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 が AWS マネージド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 の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

以下の情報は、Amazon Managed Grafana と IAM の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちます。

### トピック

- [Amazon Managed Grafana でアクションを実行する権限がない](#)
- [iam を実行する権限がありません。PassRole](#)
- [AWS アカウント以外のユーザーに 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

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、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 を実行する権限がありません。PassRole

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 にロールを渡すことができるようにする必要があります。

一部の AWS 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、そのサービスに既存のロールを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。

次の例のエラーは、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 アカウント以外のユーザーに 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、「」を参照してください [Amazon Managed Grafana との連携方法 IAM](#)。
- 所有 AWS アカウントしている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 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でのロールとリソースベースのポリシーの使用の違いについては、IAM ユーザーガイドの「[IAM でのクロスアカウントリソースアクセス](#)」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。これを防ぐために、AWSには、アカウント内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が用意されています。

リソースポリシーで [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、Amazon Managed Grafana が別のサービスに付与するアクセス許可をリソースに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。

の値は、Amazon Managed Grafana ワークスペースの ARN aws:SourceArn 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**:\* のように指定します。

次の例は、Amazon Managed Grafana Workspace IAM ロールの信頼ポリシーで aws:SourceArn および グローバルaws:SourceAccount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ています。

```
{
  "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 のサービスにリンクされたロールの使用

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一意のタイプの 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、ユーザーに代わってサービスから他の AWS の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。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、Amazon Managed Grafana の設定が簡単になります。Amazon Managed Grafana は、サービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、Amazon Managed Grafana のみがそのロールを引き受けることができます。定義したアクセス許可には、信

頼ポリシーと許可ポリシーが含まれます。こ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は、まずその関連リソースを削除しなければ削除できません。これにより、Amazon Managed Grafana リソースへのアクセス許可が誤って削除されないため、Amazon Managed Grafana リソースが保護されます。

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については、「[IAM と連動するAWS のサービス](#)」を参照し、Service-linked roles (サービスにリンクされたロール) の列内で Yes (はい) と表記されたサービスを確認してください。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、[はい] リンクを選択します。

## Amazon Managed Grafana のサービスにリンクされたロールのアクセス許可

Amazon Managed Grafana は、という名前のサービスにリンクされたロールを使用します AmazonManagedGrafana。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 サービスにリンクされたロールは、次のサービスを信頼してロールを引き受けます。

- grafana.amazonaws.com

### AmazonManagedGrafana サービスにリンクされたロール

は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、「[マネージド AWS 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により、Amazon Managed Grafana は指定されたリソースに対して以下のアクションを実行できます。

```
{
  "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"
      }
    }
  }
]
}
```

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) 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 ユーザーガイド」の「[サービスにリンクされたロールのアクセス許可](#)」を参照してください。

## Amazon Managed Grafana のサービスにリンクされたロールの作成

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 VpcConfiguration で CreateWorkspace を使用して 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ロールは、このロール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、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、サービスにリンクされたロールのサポートが開始されると、Amazon Managed Grafana はアカウントに AmazonManagedGrafana ロールを作成しました。詳細については、「[AWS 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ある場合は、同じ手順でアカウントにロールを再作成できます。CreateWorkspace で を呼び出すと VpcConfiguration、Amazon Managed Grafana によってサービスにリンクされたロールが再度作成されます。

IAM コンソールを使用して、Grafana ユースケースでサービスにリンクされたロールを作成することもできます。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「[サービスにリンクされたロールの作成](#)」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 のサービスにリンクされたロールの編集

Amazon Managed Grafana では、AmazonManagedGrafana サービスにリンクされた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を作成した後は、多くのエンティティによってロールが参照され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 を使用したロールの説明の編集はできます。詳細については、「IAM ユーザーガイド」の「[サービスにリンクされたロールの編集](#)」を参照してください。

## Amazon Managed Grafana のサービスにリンクされたロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、その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たときに Amazon Managed Grafana サービスがロールを使用している場合、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。

で使用される Amazon Managed Grafana リソースを削除するには AmazonManagedGrafana

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. 内のすべてのワークスペースを削除しますRegion。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 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Region。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」](#) トピックを参照してください。

### Note

ワークスペースがある各 AWS リージョン に対してこの手順を繰り返します。サービスにリンクされた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。

サービスにリンクされたロールを IAM で手動削除するには

IAM コンソール、または AWS API を使用して AWS CLI、サービスにリンクされたロールを削除します AmazonManagedGrafana。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 サービスにリンクされたロールでサポートされているリージョン

Amazon Managed Grafana は、サービスが利用可能な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ています。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 の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アクセス許可
- 組織のサービス管理アクセス許可
- カスタマー管理アクセス許可

ワークスペースを作成するときは、使用するアクセス許可モードを選択します。必要に応じて、後でこれを変更することもできます。

どちらのサービス管理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。その後、必要に応じて IAM コンソールでこれらのポリシーを編集できます。

### 1 つのアカウントのサービス管理アクセス許可

このモードでは、Amazon Managed Grafana は AmazonGrafanaServiceRole-**random-id** というロールを作成します。次に、Amazon Managed Grafana は、Amazon Managed Grafana ワークスペースからアクセスするように選択した 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 マネージドポリシー をアタッチします AmazonGrafanaCloudWatchAccess。

**Note**

AmazonGrafanaCloudWatchAccess 管理ポリシーの作成 CloudWatch 前に が使用したワークスペースの場合、Amazon Managed Grafana は AmazonGrafanaCloudWatchPolicy-**random-id** という名前のカスタマー管理ポリシーを作成しました。

## Amazon OpenSearch サービス

Amazon Managed Grafana は、-AmazonGrafanaOpenSearchPolicy**random-id** という名前のカスタマー管理ポリシーを作成します。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が機能するためには必要ありません。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 マネージドポリシー をアタッチします `AWSIoTSiteWiseReadOnlyAccess`。

## Amazon Redshift

Amazon Managed Grafana は、AWS 管理ポリシー をアタッチします `AmazonGrafanaRedshiftAccess`。

## Amazon Athena

Amazon Managed Grafana は、AWS 管理ポリシー をアタッチします `AmazonGrafanaAthenaAccess`。

## Amazon Managed Service for Prometheus

Amazon Managed Grafana は、`-AmazonGrafanaPrometheusPolicy`***random-id*** という名前のカスタマー管理ポリシーを作成します。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。プラグインが機能するためには必要ありません。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 は、`AmazonGrafanaSNSPolicy` ***random-id*** という名前のカスタマー管理ポリシーを作成します。このポリシーでは、文字列で始まるアカウントの SNS トピック

クのみで使用に制限されていますgrafana。これは、独自のポリシーを作成する場合には必要ありません。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 マネージドポリシー をアタッチしますAmazonTimestreamReadOnlyAccess。

## X-Ray

Amazon Managed Grafana は、AWS マネージドポリシー をアタッチしますAWSXrayReadOnlyAccess。

## 組織のサービス管理アクセス許可

このモードは、組織内の管理アカウントまたは委任された管理者アカウントで作成されたワークスペースでのみサポートされます。委任された管理者アカウントは、組織のスタックセットを作成および管理できます。委任された管理者アカウントの詳細については、[「委任された管理者の登録」](#)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。選択した組織単位の各アカウントで、Amazon Managed Grafana は AmazonGrafanaOrgMemberRole-**random-id** というロールを作成します。このロールの作成は、との統合を通じて実行されます AWS CloudFormation StackSets。

このロールには、ワークスペースで使用する 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、「」を参照してください [1つのアカウントのサービス管理アクセス許可](#)。

Amazon Managed Grafana は、組織の管理アカウントに AmazonGrafanaOrgAdminRole-**random-id** というロールも作成します。このロールは、Amazon Managed Grafana ワークスペースに、組織内の他のアカウントへのアクセス許可を付与します。AWS サービス通知チャン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して、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する

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。

-AmazonGrafanaStackSet**random-id** スタックセットの内容は次のとおりです。

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:GetInsightRuleReportResource: "\*"
- Sid: AllowReadingLogsFromCloudWatch  
Effect: Allow  
Action:
  - logs:DescribeLogGroups
  - logs:GetLogGroupFields
  - logs:StartQuery
  - logs:StopQuery
  - logs:GetQueryResults
  - logs:GetLogEventsResource: "\*"
- Sid: AllowReadingTagsInstancesRegionsFromEC2  
Effect: Allow  
Action:
  - ec2:DescribeTags
  - ec2:DescribeInstances
  - ec2:DescribeRegionsResource: "\*"
- 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]

```

-AmazonGrafanaOrgAdminPolicy*random-id* の内容は次のとおりです。

```

{
  "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"
  ]
}
```

## カスタマー管理アクセス許可

カスタマー管理アクセス許可を使用する場合は、Amazon Managed Grafana ワークスペースを作成するときに、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。

このようなポリシーの例を次に示します。

```
{
  "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ン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、[にリストされている CloudWatch ポリシーのアクセス許可が必要です](#)1つのアカウントのサービス管理アクセス許可。

に示されている Amazon OpenSearch Service および Amazon Managed Service for Prometheus のポリシーの List および アクセスDescribe 許可 [1つのアカウントのサービス管理アクセス許可](#) 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では、これらのデータソースは必要ありません。

## 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 111111111111 のロールを指定する必要があります。この例では、このロールを呼び出しますWorkspaceRole。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 でロールを作成する必要があります。その を呼び出しますDataSourceRole。次に、 WorkspaceRoleと の間に信頼関係を確立する必要がありますDataSourceRole。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 は、使用する各データソースについて、このセクションの前半でリストし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 を引き受ける 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、 で定義されているアクセス許可を持つアカウント 999999999999 にアクセスしますDataSourceRole。

## IAM アクセス許可

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、アクションを実行し、クラウド AWS リソースに関する Amazon Managed Grafana データを取得するなど、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

次の表に、Amazon Managed Grafana のアクションと、その必要なアクセス許可を示します。

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します。ワークスペースは、メトリクス、ログ、トレースの作成と視	grafana:CreateWorkspace

アクション	必要なアクセス許可
覚化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します。	<code>grafana:DeleteWorkspace</code>
Amazon Managed Grafana ワークスペースに関する詳細情報を取得します。	<code>grafana:DescribeWorkspace</code>
ワークスペースに関連付けられた認証設定を取得します。	<code>grafana:DescribeWorkspaceAuthenticat ion</code>
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します。	<code>grafana:ListPermissions</code>
アカウントに存在する Amazon Managed Grafana ワークスペースのリストを取得します。	<code>grafana:ListWorkspaces</code>
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します。	<code>grafana:UpdatePermissions</code>
Amazon Managed Grafana ワークスペースを更新します。	<code>grafana:UpdateWorkspace</code>
ワークスペースに関連付けられた認証設定を更新します。	<code>grafana:UpdateWorkspaceAuthenticatio n</code>
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ます。	<code>grafana:AssociateLicense</code>

## Amazon Managed Grafana のコンプライアンス検証

AWS 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、コンプライアンスプログラム [AWS サービスによる対象範囲内のコンプライアンスプログラム](#) を参照し、

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「コンプライアンスプログラム」](#)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、「[でのレポートのダウンロード AWS Artifact](#)」の」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 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。では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- [セキュリティとコンプライアンスのクイックスタートガイド](#) – これらのデプロイガイドでは、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に重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- [アマゾン ウェブ サービスHIPAAのセキュリティとコンプライアンスのためのアーキテクチャ](#) – このホワイトペーパーでは、企業が AWS を使用して HIPAA対象アプリケーションを作成する方法について説明します。

**Note**

すべての AWS サービス がHIPAA対象となるわけではありません。詳細については、[HIPAA「対象サービスリファレンス」](#)を参照してください。

- [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。このガイドでは、ガイダンスを保護し AWS サービス、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) など) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- [「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、リソース設定が社内プラクティス、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 サービス、内のセキュリティ状態を包括的に確認できます 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リストについては、[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](#) — これにより AWS アカウント、疑わしい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監視することで、ワークロード、コンテナ、データに対する潜在的な脅威 AWS サービス を検出します。GuardDuty は、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DSS、PCI などのさまざまなコンプライアンス要件への対応に役立ちます。
- [AWS Audit Manager](#) – これにより AWS 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ゾーンを提供します。アベイラビリティゾーンでは、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 [AWS 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。

が AWS 公開したAPI呼び出しを使用して、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 )。1TLS.2 が必要で、1.3 TLS をお勧めします。

- (Ephemeral Diffie-HellmanPFS) や DHE (Elliptic Curve Ephemeral Diffie-Hellman) などの完全前方秘匿性 ECDHE () を備えた暗号スイート。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。

さらに、リクエストは、IAMプリンシパルに関連付けられたアクセスキー ID と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 Service](#) (AWS STS) を使用して、一時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。

## を使用した Amazon Managed Grafana API呼び出しのログ記録 AWS CloudTrail

Amazon Managed Grafana は、ユーザー[AWS CloudTrail](#)、ロール、または [IAM](#) によって実行されたアクションを記録するサービスであると統合されています AWS サービス。CloudTrail は、Amazon Managed Grafana のすべてのAPI呼び出し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 を使用する一部の呼び出しもキャプチャします APIs。キャプチャされる呼び出しは、リソースを作成、更新、削除する呼び出しなど、データを変更する呼び出しです。Amazon Managed Grafana でAPIsサポートされている Grafana の詳細については、「[Grafana HTTP APIs](#)」を参照してください

で収集された情報を使用して CloudTrail、Amazon Managed Grafana に対するリクエスト、リクエスト元の IP アドレス、リクエスト日時など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
- IAM Identity Center ユーザーに代わってリクエストが行われたかどうか。
- 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 サービスによって送信されたかどうか。

CloudTrail アカウント AWS アカウント を作成すると、は [CloudTrail](#) 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 イベント履歴は、[CloudTrail](#) に記録された過去 90 日間の

管理イベントの表示可能、検索可能、ダウンロード可能、およびイミュータブルなレコードを提供します AWS リージョン。詳細については、[「ユーザーガイド」の CloudTrail 「イベント履歴」の使用AWS CloudTrail](#)」を参照してください。イベント履歴を表示するための CloudTrail 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

証跡により 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 CloudTrail することで、から Amazon S3 バケットに継続的な管理イベントのコピーを 1 つ無料で配信できますが、Amazon S3 ストレージ料金が発生します。CloudTrail 料金の詳細については、「[の料金AWS CloudTrail](#)」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 では、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、既存のイベントを行ベースのJSON形式で [Apache ORC](#) 形式に変換します。ORC は、データを迅速に取得できるように最適化された列指向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、「ユーザーガイド」の[AWS CloudTrail 「Lake」の使用AWS CloudTrail](#)」を参照してください。

CloudTrail Lake イベントデータストアとクエリにはコストが発生します。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 料金の詳細については、「[の料金AWS CloudTrail](#)」を参照してください。

## での Amazon Managed Grafana 管理イベント CloudTrail

[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。デフォルトでは、は管理イベントを CloudTrail ログに記録します。

Amazon Managed Grafana 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を管理イベントとしてログに記録します。Amazon Managed Grafana がに記録する 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 CloudTrail、[「Amazon Managed Grafana API Reference」](#)を参照してください。

### Amazon Managed Grafana イベントの例

イベントは任意のソース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が含まれます。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 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": "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、アクションの CloudTrail UpdateWorkspaceAuthentication ログエントリを示しています。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId":
      "AR0AU2UJBF3NR035YZ3GV: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": "AR0AU2UJBF3NR035YZ3GV",
        "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/
NjMwMDg2NDc40TA3X2lucy1jY2E2ZGU3ZDlmYjdiM2Vh"
    }
  },
  "workspaceId": "g-84ea23c1b4"
},
"responseElements": {
  "authentication": {
    "awsSso": {
      "ssoClientId": "gAR0cWGs9-LoqCMIQ56XyEXAMPLE"
    },
    "providers": [
      "AWS_SSO",
      "SAML"
    ],
    "saml": {
      "configuration": {
        "idpMetadata": {
          "url": "https://portal.sso.us-east-1.amazonaws.com/saml/
metadata/NjMwMDg2NDc40TA3X2lucy1jY2E2ZGU3ZDlmYjdiM2Vh"
        }
      },

```

```
        "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 レコードの内容の詳細については、「ユーザーガイド」の[CloudTrail 「レコードの内容」](#)を参照してください。

## Grafana API イベントの例

Amazon Managed Grafana は、一部の Grafana API 呼び出しもログに記録します CloudTrail。キャプチャされる呼び出しは、リソースを作成、更新、削除する呼び出しなど、データを変更する呼び出しです。Amazon Managed Grafana で APIs サポートされている Grafana の詳細については、「」を参照してください [Grafana HTTP APIs](#)。

ユーザーがを使用して Amazon Managed Grafana ワークスペースにサインインする AWS IAM Identity Center

```
{
  "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.eyJwbGFpbnRleHQiOiJZUzEwYWtaWHpBZUowTD1Qc
            ],
          "state": [
            "QUFBQUVtdGx1UzB4TlRZNE9UVTF0ekkyM2RUWUFUaHZHYXcyOU9ULUVaWHhNUXAwX184N25RVGVWMD0enFpVE1iWlRVP
            ]
          }
        },
        "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": "{\"AuthToken\":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/注釈

```
{  
    "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\":\n\"test_amp_datasource_NEW_name\",\"type\":\"Amazon Managed Prometheus\",\"url\":\n\"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/チーム

```
{
  "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適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 APIs 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。キーを作成するときは、キーの有効期限を指定します。これは、キーが有効である期間を最大 30 日間まで定義します。キーの有効期間は、数時間以下など、より短く設定することを強くお勧めします。これにより、API キーを長期間有効にするよりもリスクがはるかに低くなります。

また、API キーは、セキュリティ保護の観点からパスワードとして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。

## セルフマネージド Grafana からの移行

このセクションは、既存のセルフマネージド型の 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ています。これは、オンプレミスの Grafana と AWS、での Grafana デプロイの両方に独自のアカウントで適用されます。

オンプレミスまたは自分の 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、および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 の外部で管理され、IAM Identity Center を使用するか、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、ダッシュボードの表示など、タスクを実行するために必要に応じて特定のアクセス許可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、「」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](#)。

さらに、オンプレミスの Grafana を実行するときは、存続期間の長いキーまたはシークレット認証情報を使用してデータソースにアクセスします。Amazon Managed Grafana に移行するときは、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、「[データソース CloudWatch としてを手動で追加する](#)」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アタッチすることで、Virtual Private Cloud (VPC) 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 種類の VPC エンドポイントをサポートしています。Amazon Managed Grafana サービスに接続して、Amazon Managed Grafana APIs 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への VPC エンドポイントを作成することもできます。

## インターフェイス VPC エンドポイントでの 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には、2 つの方法があります。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したり、Amazon Managed Grafana ワークスペースへのネットワークアクセスを制限する一環として VPC エンドポイントを使用したりできます。

- 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](#) 間の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の間にプライベート接続を確立できます。

Amazon VPC は AWS サービス、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ます。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 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。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 接続を必要とせずに、Amazon Managed Grafana への信頼性が高くスケーラブルな接続を提供します。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、Elastic Network Interface とプライベート IP アドレス AWS 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 [新規 - AWS サービス用 AWS PrivateLink](#) を利用しています。詳細については、[「新規 - AWS サービス用 AWS PrivateLink」](#) を参照してください。

Amazon VPC の使用を開始する方法については、「Amazon VPC [ユーザーガイド](#)」の「[使用開始](#)」を参照してください。

## Amazon Managed Grafana AWS PrivateLink に接続するための 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。

- ワークスペースを管理するために Amazon Managed Grafana API に接続するには、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の詳細については、「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 APIs「[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、フォーム内の URLs ローカル解決が可能になります。`*.grafana-workspace.region.amazonaws.com`

## 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 エンドポイントの場合、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、そのアクセスをさらに制限できます。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とセキュリティグループのルールを関連付ける](#)」を参照してください。[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html#SecurityGroupRules](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html#SecurityGroupRules)

## エンドポイントポリシーを使用した Amazon Managed Grafana API VPC エンドポイントへのアクセスの制御

Amazon Managed Grafana API ( を使用 `com.amazonaws.region.grafana`) 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

ワークスペースに接続されている VPC エンドポイント ( を使用 `com.amazonaws.region.grafana-workspace`) 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 評価項目ポリシーは、評価項目の作成時または変更時に評価項目に加える国際機械技術者協会 ( IAM ) の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。

評価項目のポリシーは、JSON形式で記載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制御](#)」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いようにします。

```
{
  "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 コンソール 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、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、ポリシータブを選択します。
5. [ポリシーの編集] を選択し、変更を行います。

# Amazon Managed Grafana 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	<a href="#">はい</a>	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	いいえ	ワークスペースあたり 100。
クラシックアラートのワークスペースあたりのルール数、または Grafana アラートのワー		

リソース	調整可能	デフォルトのクォータ
ワークスペースあたりのルールインスタンス数。		
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	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。
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降のドキュメントの重要な変更点を示します。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
<a href="#">Amazon Managed Grafana が Amazon EKS クラスターをモニタリングするためのソリューションを更新</a>	Amazon Managed Grafana はソリューションを更新して、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し AWS CDK、デプロイを簡素化します。	2024 年 6 月 10 日
<a href="#">Amazon Managed Grafana が Grafana バージョン 10 のサポートを追加</a>	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 および 9 も引き続き使用できます。	2024 年 5 月 15 日
<a href="#">Amazon Managed Grafana が Amazon EKS クラスターをモニタリングするためのソリューションを追加</a>	Amazon Managed Grafana は、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するソリューションのサポートを追加します AWS CloudFormation。	2024 年 4 月 30 日
<a href="#">Amazon Managed Grafana が古い管理ポリシーを置き換える</a>	Amazon Managed Grafana は、新しい管理ポリシー AWSGrafanaWorkspacePermissionManagementV2 を追加して、古	2024 年 1 月 5 日

いAWSGrafanaWorkspac  
ePermissionManagement管理  
ポリシーを置き換えます。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  
セットを提供することで、ワ  
ークスペースのセキュリティ  
を向上させます。

[Amazon Managed Grafana が  
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  
は、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を追  
加します。

2023 年 11 月 15 日

[Amazon Managed Grafana が  
既存のワークスペースのバー  
ジョンの更新のサポートを追  
加](#)

Amazon Managed Grafana  
は、既存のワークスペースを  
より新しいバージョンに更新  
するためのサポートを追加し  
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 が  
OpenSearch データソースの  
トレース分析のサポートを追  
加](#)

Amazon Managed Grafana  
は、バージョン 9 以降をサ  
ポートするワークスペースを  
使用する場合に、OpenSearc  
h データソースのトレースの  
リストを表示するためのサ  
ポートを追加します。

2023 年 6 月 22 日

### [Amazon Managed Grafana が 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 が新しい管理ポリシーを追加](#)

Amazon Managed Grafana に、Amazon Managed Grafana AmazonGrafanaCloudWatchAccess がのメトリクスにアクセスできるようにする新しい管理ポリシーが追加されました CloudWatch。

2023 年 3 月 24 日

### [Amazon Managed Grafana が新しいアクセス許可を追加](#)

Amazon Managed Grafana は、Microsoft Active Directory または Active Directory Connector で管理されている IAM Identity Center ユーザーおよびグループを Amazon Managed Grafana ワークスペースに関連付けるか、関連付けを解除AWSGrafanaWorkspacePermissionManagementできるように、新しいアクセス許可を追加しました。

2023 年 3 月 22 日

### [アラートインスタンスの明確化](#)

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ます。これにより、クォータに達する速度に影響しません。ドキュメントが更新され、追加の詳細が提供されません。

2023 年 3 月 20 日

[Amazon Managed Grafana が新しいデータソースを追加](#)

Amazon Managed Grafana は、Databricks と Google 用の新しいデータソースプラグインを追加します BigQuery。

2023 年 3 月 14 日

[Amazon Managed Grafana が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](#)

Amazon Managed Grafana は、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を追加します。

2023 年 2 月 16 日

[Amazon Managed Grafana が新しいアクセス許可を追加](#)

Amazon Managed Grafana は、IAM Identity Center のユーザーとグループで AWS Grafana Workspace Permission Management を Grafana ワークスペースに関連付けることができるように、新しいアクセス許可を追加しました。

2022 年 12 月 20 日

[Grafana サービスの新しい SLR ポリシー](#)

Amazon Grafana Service Linked Role Policy AWS マネージド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 Amazon Managed Grafana 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を追加しました。

2022 年 11 月 23 日

<a href="#">Amazon Managed Grafana がワークスペース設定を追加</a>	Amazon Managed Grafana が Grafana ワークスペースインスタンスごとに設定変更を行うサポートを追加	2022 年 11 月 23 日
<a href="#">Amazon Managed Grafana が Grafana アラートのサポートを追加</a>	Amazon Managed Grafana は、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、更新された Grafana アラート機能を使用するためのサポートを追加しました。	2022 年 11 月 23 日
<a href="#">Amazon Managed Grafana が 3 つの新しいビジュアライゼーションを追加</a>	Amazon Managed Grafana は、Plotly、Sankey、および Scatter パネルの視覚化を追加します。	2022 年 11 月 17 日
<a href="#">AWS SSO の IAM Identity Center へのブランド変更</a>	AWS SSO は IAM Identity Center にブランド変更されました。	2022 年 7 月 26 日
<a href="#">Amazon Managed Grafana の機能強化</a>	Amazon Managed Grafana は、Grafana バージョン 8.4、Pixie、GitHub および Moogsoft データソース、記録されたクエリ、パネルの WindRose 視覚化のサポートを追加します。	2022 年 5 月 13 日
<a href="#">Amazon Managed Grafana の機能強化</a>	Amazon Managed Grafana でタグ付けのサポートが追加されました。	2022 年 3 月 31 日
<a href="#">Amazon Managed Grafana の機能強化</a>	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した。Amazon Managed Grafana は、Geomap パネルの視覚化と 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 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 をサポートするよう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。