aws

ユーザーガイド

Amazon Managed Service for Prometheus

Amazon Managed Service for Prometheus: ユーザーガイド

Table of Contents

Amazon Managed Service for Prometheus とは

Amazon Managed Service for Prometheus は、コンテナのメトリクスをモニタリングする Prometheus 互換のサーバーレスサービスです。これにより、大規模なコンテナ環境を安全にモニタリングすることが容易になります。Amazon Managed Service for Prometheus では、現在使用されているものと同じオープンソースの Prometheus データモデルとクエリ言語を使用して、コンテナ化されたワークロードのパフォーマンスをモニタリングできます。また、基盤のインフラストラクチャを管理する必要なく、スケーラビリティ、可用性、セキュリティを強化できます。

Amazon Managed Service for Prometheus は、ワークロードのスケールアップとスケールダウンに応じて自動的に運用メトリクスの取り込み、保存、クエリをスケールします。セキュリティ AWS サービスと統合することで、データへの高速で安全なアクセスが可能になります。

Amazon Managed Service for Prometheus は、複数のアベイラビリティーゾーン (マルチ AZ) 配置を使用して高い可用性を実現するように設計されています。ワークスペースに取り込まれたデータは、同じリージョンの 3 つのアベイラビリティーゾーンにレプリケートされます。

Amazon Managed Service for Prometheus は、Amazon Elastic Kubernetes Service 環境および自己管理型 Kubernetes 環境で実行されるコンテナクラスターと連携します。

Amazon Managed Service for Prometheus では、Prometheus と同じオープンソースの Prometheus データモデルおよび PromQL クエリ言語が使用されます。エンジニアリングチームは、PromQL を使用してメトリクスのフィルタリングや集計を行ったり、メトリクスにアラームを設定したりして、コードを変更することなくすばやくパフォーマンスを可視化できます。Amazon Managed Service for Prometheus により、運用コストや複雑さを伴わずに柔軟なクエリ機能が提供されます。

ワークスペースに取り込まれたメトリクスは、デフォルトで 150 日間保存され、自動的に削除されます。この長さは調整可能なクォータ です。

サポートされるリージョン

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

リージョン名	リージョン	エンドポイント	プロトコル	
米国東部 (オハイオ)	us-east-2	aps.us-east-2.amazonaws.com	HTTPS	
		aps-workspaces.us-east-2.amazonaws.com	HTTPS	
		aps-workspaces.us-east-2.api.aws	HTTPS	
		aps.us-east-2.api.aws	HTTPS	
米国東部 (バージニア北部)	us-east-1	aps.us-east-1.amazonaws.com	HTTPS	
		aps-workspaces.us-east-1.amazonaws.com	HTTPS	
		aps-workspaces.us-east-1.api.aws	HTTPS	
		aps.us-east-1.api.aws	HTTPS	
米国西部 (オレゴン)	us-west-2	aps.us-west-2.amazonaws.com	HTTPS	
		aps-workspaces.us-west-2.amazonaws.com	HTTPS	
		aps-workspaces.us-west-2.api.aws	HTTPS	
		aps.us-west-2.api.aws	HTTPS	
アジアパシフィック (ムンバイ)	ap-south-1	aps.ap-south-1.amazonaws.com	HTTPS	
		aps-workspaces.ap-south-1.amazonaws.com	HTTPS	
		aps-workspaces.ap-south-1.api.aws	HTTPS	
		aps.ap-south-1.api.aws	HTTPS	
アジアパシフィック (ソウル)	ap-northeast-2	aps.ap-northeast-2.amazonaws.com	HTTPS	
		aps-workspaces.ap-northeast-2.amazonaws.com	HTTPS	
		aps-workspaces.ap-northeast-2.api.aws	HTTPS	
		aps.ap-northeast-2.api.aws	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル
		aps.ap-northeast-2.api.aws	
アジアパシフィック (シンガポール)	ap-southeast-1	aps.ap-southeast-1.amazonaws.com aps-workspaces.ap-southeast-1.amazonaws.com aps-workspaces.ap-southeast-1.api.aws aps.ap-southeast-1.api.aws	HTTPS HTTPS HTTPS HTTPS
アジアパシフィック (シドニー)	ap-southeast-2	aps.ap-southeast-2.amazonaws.com aps-workspaces.ap-southeast-2.amazonaws.com aps-workspaces.ap-southeast-2.api.aws aps.ap-southeast-2.api.aws	HTTPS HTTPS HTTPS HTTPS
アジアパシフィック (東京)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com aps-workspaces.ap-northeast-1.amazonaws.com aps-workspaces.ap-northeast-1.api.aws aps.ap-northeast-1.api.aws	HTTPS HTTPS HTTPS HTTPS
欧州 (フランクフルト)	eu-central-1	aps.eu-central-1.amazonaws.com aps-workspaces.eu-central-1.amazonaws.com aps-workspaces.eu-central-1.api.aws aps.eu-central-1.api.aws	HTTPS HTTPS HTTPS HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
欧州 (アイルランド)	eu-west-1	aps.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.amazonaws.com	HTTPS
		aps-workspaces.eu-west-1.api.aws	HTTPS
		aps.eu-west-1.api.aws	HTTPS
欧州 (ロンドン)	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.api.aws	HTTPS
		aps.eu-west-2.api.aws	HTTPS
欧州 (パリ)	eu-west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.api.aws	HTTPS
		aps.eu-west-3.api.aws	HTTPS
欧州 (ストックホルム)	eu-north-1	aps.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.amazonaws.com	HTTPS
		aps-workspaces.eu-north-1.api.aws	HTTPS
		aps.eu-north-1.api.aws	HTTPS
南米 (サンパウロ)	sa-east-1	aps.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.amazonaws.com	HTTPS
		aps-workspaces.sa-east-1.api.aws	HTTPS
		aps.sa-east-1.api.aws	HTTPS

料金

メトリクスの取り込みと保存には料金がかかります。ストレージ料金は、メトリクスサンプルとメタデータの圧縮サイズに基づきます。詳細については、「Amazon Managed Service for Prometheus の料金」を参照してください。

AWS Cost Explorer および AWS コストと使用状況レポートを使用して、料金を監視できます。詳細については、Cost Explorer を使用したデータの探索」およびAWS 「コストと使用状況レポートとは」を参照してください。

プレミアムサポート

AWS プレミアムサポートプランの任意のレベルにサブスクライブする場合、プレミアムサポートは Amazon Managed Service for Prometheus に適用されます。

Amazon Managed Service for Prometheus の使用を開始する

Amazon Managed Service for Prometheus は、コンテナメトリクスをモニタリングするためのサーバーレスの Prometheus 互換サービスであり、大規模なコンテナ環境を安全にモニタリングすることを容易にします。このセクションでは、Amazon Managed Service for Prometheus を使用する際の 3 つの主要領域について説明します。

- ワークスペースの作成 – メトリクスを保存およびモニタリングする Amazon Managed Service for Prometheus ワークスペースを作成します。
- メトリクスの取り込み — メトリクスをワークスペースに取り込むまで、ワークスペースは空です。Amazon Managed Service for Prometheus にメトリクスを送信するか、Amazon Managed Service for Prometheus でメトリクスを自動的にスクレイプできます。
- クエリメトリクス – ワークスペースにデータとしてメトリクスを作成したら、データをクエリしてそれらのメトリクスを探索またはモニタリングする準備が整います。

を初めて使用する場合、 AWSこのセクションには の設定に関する詳細 AWS アカウントも含まれています。

トピック

- セットアップする AWS
- Amazon Managed Service for Prometheus ワークスペースの作成
- ワークスペースへの Prometheus メトリクスの取り込み
- Prometheus メトリクスに対するクエリの実行

セットアップする AWS

このセクションのタスクを完了して をセットアップする AWS を初めて使用します。が既にある場合 AWS アカウント、「」に進みますAmazon Managed Service for Prometheus ワークスペースの作成。

にサインアップするとき AWS、 AWS アカウントは、 のすべてのサービスに自動的にアクセスできます。 AWS Amazon Managed Service for Prometheus を含む 。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。

トピック

にサインアップする AWS アカウント

をお持ちでない場合 AWS アカウントで、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/サインアップ 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き 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。ルートユーザーはすべての にアクセスできます AWS のサービス アカウントの および リソース。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/ に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示し、アカウントを管理できます。

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後 AWS アカウント、 をセキュリティで保護する AWS アカウントのルートユーザー、有効化 AWS IAM Identity Center、および 管理ユーザーを作成して、日常的なタスクにルートユーザーを使用しないようにします。

のセキュリティ保護 AWS アカウントのルートユーザー

1. にサインインします。AWS Management Console 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。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、「」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。 AWS サインイン ユーザーガイド。

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、「 の仮想MFAデバイスの有効化」を参照してください。 AWS アカウントIAM ユーザーガイドのルートユーザー (コンソール) 。

管理アクセスを持つユーザーを作成する

1. IAM Identity Center を有効にします。

   手順については、「 の有効化」を参照してください。 AWS IAM Identity Center ()AWS IAM Identity Center ユーザーガイド。

2. IAM Identity Center で、ユーザーに管理アクセス権を付与します。

   の使用に関するチュートリアル IAM アイデンティティセンターディレクトリ ID ソースとして、「デフォルトを使用してユーザーアクセスを設定する」を参照してください。 IAM アイデンティティセンターディレクトリ ()AWS IAM Identity Center ユーザーガイド。

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 Identity Center ユーザーでサインインするには、IAMIdentity Center ユーザーの作成時に E メールアドレスにURL送信されたサインインを使用します。

  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、「 へのサインイン」を参照してください。 AWS の アクセスポータル AWS サインイン ユーザーガイド。

追加のユーザーにアクセス権を割り当てる

1. IAM Identity Center で、最小特権のアクセス許可を適用するベストプラクティスに従うアクセス許可セットを作成します。

   手順については、「」の「アクセス許可セットの作成」を参照してください。 AWS IAM Identity Center ユーザーガイド。

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、「」の「グループの追加」を参照してください。 AWS IAM Identity Center ユーザーガイド。

Amazon Managed Service for Prometheus ワークスペースの作成

ワークスペースは、Prometheus メトリクスの保存とクエリに使用される専用の論理スペースです。ワークスペースでは、更新、一覧表示、記述、削除、メトリクスの取り込みとクエリなど、管理操作を認可するためのきめ細かいアクセスコントロールがサポートされます。アカウント内のリージョンごとに 1 つ以上のワークスペースを持つことができます。

ワークスペースをセットアップするには、次の手順に従います。

> **Note**
>
> ワークスペースの作成と使用可能なオプションの詳細については、「」を参照してくださいAmazon Managed Service for Prometheus ワークスペースを作成する。

Amazon Managed Service for Prometheus ワークスペースを作成するには

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) を開きます。
2. [WorkSpace エイリアス] に、新しいワークスペースのエイリアスを入力します。

   ワークスペースエイリアスは、ワークスペースの識別に役立つわかりやすい名前です。これは一意でなくても構いません。2 つのワークスペースに同じエイリアスを付けることもできます。ただし、すべてのワークスペースには Amazon Managed Service for Prometheus によって生成された一意のワークスペース ID が割り当てられます。
3. (オプション) 名前空間にタグを追加するには、[新しいタグを追加] を選択します。

   [キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。

   別のタグを追加するには、[新しいタグを追加] を再度選択します。
4. [ワークスペースを作成する] 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。ここには、このワークスペースのステータス、ARN、ワークスペース ID、リモート書き込み用とクエリ用のエンドポイント URL などの情報が表示されます。

   最初はステータスが [作成中] になる可能性があります。ステータスが [アクティブ] になるまで待ってから、メトリクスの取り込みの設定に進んでください。

[エンドポイント - リモート書き込み URL] と [エンドポイント - クエリ URL] に表示された URL を書き留めます。これらの URL は、このワークスペースにメトリクスをリモートで書き込むように Prometheus サーバーを構成するときと、それらのメトリクスにクエリを実行するときに必要になります。

ワークスペースへの Prometheus メトリクスの取り込み

メトリクスを取り込む方法の 1 つは、スタンドアロンの Prometheus エージェント (エージェントモードで実行されている Prometheus インスタンス) を使用してクラスターからメトリクスを取得し、Amazon Managed Service for Prometheus に転送してストレージとモニタリングを行うことです。このセクションでは、Helm を使用して Prometheus エージェントの新しいインスタンスをセットアップすることにより、Amazon EKS から Amazon Managed Service for Prometheus ワークスペースへのメトリクスの取り込みを設定する方法について説明します。

メトリクスを保護する方法や可用性の高いメトリクスを作成する方法など、Amazon Managed Service for Prometheus にデータを取り込む他の方法については、「Amazon Managed Service for Prometheus ワークスペースにメトリクスを取り込む」を参照してください。

> **Note**
>
> ワークスペースに取り込まれたメトリクスは、デフォルトで 150 日間保存され、その後自動的に削除されます。この長さは調整可能なクォータ です。

このセクションの手順に従うと、Amazon Managed Service for Prometheus を迅速に設定して稼働させることができます。ワークスペース を既に作成していることを前提としています。このセクションでは、Amazon EKS クラスターに新しい Prometheus サーバーをセットアップし、新しいサーバーはデフォルト設定を使用してエージェントとして機能し、Amazon Managed Service for Prometheus にメトリクスを送信します。この方法には次の前提条件があります。

- 新しい Prometheus サーバーがメトリクスを収集する Amazon EKS クラスターが必要です。
- Amazon EKS クラスターには、Amazon EBS CSI ドライバーがインストールされている必要があります (Helm で必要) 。
- Helm CLI 3.0 以降を使用する必要があります。
- 以下のセクションのステップを実行するには、Linux または MacOS コンピュータを使用する必要があります。

ステップ 1: 新しい Helm チャートリポジトリを追加する

次のコマンドを入力して、新しい Helm チャートリポジトリを追加します。これらのコマンドの詳細については、「Helm Repo」を参照してください。

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo add kube-state-metrics https://kubernetes.github.io/kube-state-metrics
helm repo update
```

ステップ 2: Prometheus 名前空間を作成する

次のコマンドを入力して、Prometheus サーバーとその他のモニタリングコンポーネント用の Prometheus 名前空間を作成します。を、この名前空間に使用する名前*prometheus-agent-namespace*に置き換えます。

```
kubectl create namespace prometheus-agent-namespace
```

ステップ 3: サービスアカウントの IAM ロールを設定する

この取り込み方法では、Prometheus エージェントが稼働している Amazon EKS クラスターのサービスアカウントの IAM ロールを使用する必要があります。

サービスアカウントの IAM ロールを使用すると、IAM ロールを Kubernetes サービスアカウントに関連付けることができます。このサービスアカウントは、そのサービスアカウントを使用するポッド内のコンテナに AWS アクセス許可を提供できます。詳細については、「サービスアカウントの IAM ロール」を参照してください。

これらのロールをまだ設定していない場合は、「Amazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定」の手順に従ってロールを設定します。そのセクションの手順では、`eksctl` を使用する必要があります。詳細については、「Amazon Elastic Kubernetes Service の開始方法 - `eksctl`」を参照してください。

> **Note**
>
> EKS または ではなく AWS 、アクセスキーとシークレットキーのみを使用して Amazon Managed Service for Prometheus にアクセスする場合、 `EKS-IAM-ROLE` ベースの SigV4 を使用することはできません。

ステップ 4: 新しいサーバーをセットアップしてメトリクスの取り込みを開始する

新しい Prometheus エージェントをインストールし、Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信するには、以下の手順に従います。

新しい Prometheus エージェントをインストールし、Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信するには

1. テキストエディタを使用して、`my_prometheus_values_yaml` という名前のファイルを作成し、次の内容を記述します。

   - *IAM_PROXY_PROMETHEUS_ROLE_ARN* を、 で作成した の ARN に置き換えamp-iamproxy-ingest-roleますAmazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定。
   - *WORKSPACE_ID* は、Amazon Managed Service for Prometheus ワークスペースの ID に置き換えます。
   - *REGION* は、Amazon Managed Service for Prometheus のリージョンに置き換えます。

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
## For the rest of prometheus helm chart values see: https://github.com/prometheus-
community/helm-charts/blob/main/charts/prometheus/values.yaml
##
serviceAccounts:
  server:
    name: amp-iamproxy-ingest-service-account
    annotations:
      eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
server:
  remoteWrite:
    - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
      sigv4:
        region: ${REGION}
      queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
```

2. 次のコマンドを入力して、Prometheus サーバーを作成します。

   - を Prometheus リリース名*prometheus-chart-name*に置き換えます。
   - を Prometheus 名前空間の名前*prometheus-agent-namespace*に置き換えます。

```
helm install prometheus-chart-name prometheus-community/prometheus -n prometheus-
agent-namespace \
-f my_prometheus_values_yaml
```

Prometheus メトリクスに対するクエリの実行

ワークスペースにメトリクスが取り込まれるようになったら、それらのメトリクスに対してクエリを実行できます。メトリクスをクエリする一般的な方法は、Grafana などのサービスを使用してメトリクスをクエリすることです。このセクションでは、Amazon Managed Grafana を使用して Amazon Managed Service for Prometheus のメトリクスをクエリする方法を学習します。

> **Note**
>
> Amazon Managed Service for Prometheus のメトリクスをクエリする他の方法や、Amazon Managed Service for Prometheus API を使用する方法については、「Prometheus メトリクスに対するクエリの実行」を参照してください。

このセクションでは、ワークスペースが既に作成されており、そのワークスペースにメトリクスを取り込んでいることを前提としています。

クエリの実行には、Prometheus の標準クエリ言語である PromQL を使用します。PromQL とその構文の詳細については、Prometheus ドキュメントの「Querying Prometheus」を参照してください。

Amazon Managed Grafana は、オープンソースの Grafana 向けのフルマネージドサービスで、オープンソースのサードパーティー ISV、およびデータソースを大規模に視覚化および分析するための AWS サービスへの接続を簡素化します。

Amazon Managed Service for Prometheus では、Amazon Managed Grafana を使用してワークスペース内のメトリクスにクエリを実行することがサポートされています。Amazon Managed Grafana コンソールで、既存の Amazon Managed Service for Prometheus アカウントを検出して、Amazon Managed Service for Prometheus ワークスペースをデータソースとして追加できま

す。Amazon Managed Grafana は、Amazon Managed Service for Prometheus にアクセスするために必要な認証情報の設定を管理します。Amazon Managed Grafana から Amazon Managed Service for Prometheus への接続を作成する方法の詳細については、「Amazon Managed Grafana User Guide」の手順を参照してください。

Amazon Managed Service for Prometheus のアラートを Amazon Managed Grafana で表示することもできます。アラートとの統合を設定する手順については、「アラートを Amazon Managed Grafana またはオープンソース Grafana と統合する」を参照してください。

> **Note**
>
> Amazon Managed Grafana ワークスペースがプライベート VPC を使用するように設定している場合は、Amazon Managed Service for Prometheus のワークスペースを同じ VPC に接続する必要があります。詳細については、「プライベートで Amazon Managed Grafana に接続する VPC」を参照してください。

Amazon Managed Service for Prometheus ワークスペースの管理

ワークスペースは、Prometheus メトリクスの保存とクエリに使用される専用の論理スペースです。ワークスペースでは、更新、一覧表示、記述、削除、メトリクスの取り込みとクエリなど、管理操作を認可するためのきめ細かいアクセスコントロールがサポートされます。アカウント内のリージョンごとに 1 つ以上のワークスペースを持つことができます。

Amazon Managed Service for Prometheus ワークスペースを作成して管理するには、このセクションの手順を使用します。

トピック

- Amazon Managed Service for Prometheus ワークスペースを作成する
- Amazon Managed Service for Prometheus ワークスペースを編集する
- Amazon Managed Service for Prometheus ワークスペースの詳細を検索するには、以下を含めます。 ARN
- Amazon Managed Service for Prometheus ワークスペースを削除する

Amazon Managed Service for Prometheus ワークスペースを作成する

Amazon Managed Service for Prometheus ワークスペースを作成するには、以下の手順に従います。または Amazon Managed Service for Prometheus コンソールを使用できます AWS CLI 。

> **Note**
>
> Amazon EKSクラスターを実行している場合は、AWS Controllers for Kubernetes を使用して新しいワークスペースを作成することもできます。

を使用してワークスペースを作成するには AWS CLI

1. 以下のコマンドを入力して、ワークスペースを作成します。この例では `my-first-workspace` という名前のワークスペースを作成しますが、必要に応じて別のエイリアスを使用することもできます。ワークスペースエイリアスは、ワークスペースの識別に役立つわかりやす

い名前です。これは一意でなくても構いません。2 つのワークスペースには同じエイリアスを含めることができますが、すべてのワークスペースには一意のワークスペース がありIDs、これは Amazon Managed Service for Prometheus によって生成されます。

(オプション) 独自のKMSキーを使用してワークスペースに保存されているデータを暗号化するには、使用する AWS KMS キーに kmsKeyArnパラメータを含めることができます。Amazon Managed Service for Prometheus はカスタマーマネージドキーの使用に対して課金しませんが、 のキーに関連するコストが発生する場合があります AWS Key Management Service。Amazon Managed Service for Prometheus によるワークスペース内のデータの暗号化、または独自のカスタマーマネージドキーの作成、管理、使用方法の詳細については、「保管中の暗号化」を参照してください。

括弧 ([]) 内のパラメータはオプションであり、コマンドには括弧を含めないでください。

```
aws amp create-workspace [--alias my-first-workspace] [--kmsKeyArn arn:aws:aps:us-west-2:111122223333:workspace/ws-sample-1234-abcd-56ef-7890abcd12ef] [--tags Status=Secret,Team=My-Team]
```

このコマンドは次のデータを返します。

- workspaceId は、このワークスペースの一意の ID です。この ID を書き留めてください。
- arn は、このワークスペースARNの です。
- status は、ワークスペースの現在のステータスです。ワークスペースの作成直後は CREATING になる可能性があります。
- kmsKeyArn は、ワークスペースデータの暗号化に使用されるカスタマーマネージドキーです (指定されている場合)。

> **Note**
>
> カスタマーマネージドキーで作成されたワークスペースは、取り込み用にAWS マネージドコレクターを使用することはできません。
> カスタマーマネージドキーと AWS 所有キーのどちらを慎重に使用するかを選択します。カスタマーマネージドキーを使用して作成されたワークスペースは、後で (またはその逆で) AWS 所有キーを使用するように変換することはできません。

- tags は、ワークスペースのタグ (ある場合) のリストを示します。

2. create-workspace コマンドが CREATING ステータスを返した場合は、後で次のコマンドを入力することで、ワークスペースの準備が整ったかどうかを確認できます。置換 *my-workspace-id* create-workspace コマンドが に返した値workspaceId。

```
aws amp describe-workspace --workspace-id my-workspace-id
```

describe-workspace コマンドから status として ACTIVE が返されたら、ワークスペースを使用する準備ができています。

Amazon Managed Service for Prometheus コンソールを使用してワークスペースを作成するには

1. で Amazon Managed Service for Prometheus コンソールを開きますhttps://console.aws.amazon.com/prometheus/。
2. [Create] (作成) を選択します。
3. [WorkSpace エイリアス] に、新しいワークスペースのエイリアスを入力します。

   ワークスペースエイリアスは、ワークスペースの識別に役立つわかりやすい名前です。これは一意でなくても構いません。2 つのワークスペースには同じエイリアスを含めることができますが、すべてのワークスペースには一意のワークスペース がありIDs、これは Amazon Managed Service for Prometheus によって生成されます。
4. (オプション) 独自のKMSキーを使用してワークスペースに保存されているデータを暗号化するには、暗号化設定のカスタマイズ を選択し、使用する AWS KMS キーを選択します (または新しいキーを作成します)。ドロップダウンリストからアカウントのキーを選択するか、アクセスできる任意のキーARNに を入力します。Amazon Managed Service for Prometheus はカスタマーマネージドキーの使用に対して課金しませんが、 のキーに関連するコストが発生する場合があります AWS Key Management Service。

   Amazon Managed Service for Prometheus によるワークスペース内のデータの暗号化や、お客様独自のカスタマーマネージドキーの作成、管理、使用方法の詳細については、「保管中の暗号化」を参照してください。

   > **Note**
   >
   > カスタマーマネージドキーで作成されたワークスペースは、取り込み用にAWS マネージドコレクターを使用することはできません。

カスタマーマネージドキーと AWS 所有キーのどちらを慎重に使用するかを選択します。カスタマーマネージドキーを使用して作成されたワークスペースは、後で (またはその逆で) AWS 所有キーを使用するように変換することはできません。

5. (オプション) ワークスペースに 1 つ以上のタグを追加するには、[新しいタグを追加] を選択します。[キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。

   別のタグを追加するには、[新しいタグを追加] を再度選択します。

6. [ワークスペースを作成する] 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。これにより、リモート書き込みとクエリの両方URLsについて、このワークスペースのステータスARN、、ワークスペース ID、エンドポイントなどの情報が表示されます。

   ステータスは、ワークスペースの準備ができるCREATINGまで返されます。ステータスが になるまで待ってACTIVEから、メトリクス取り込みの設定に進みます。

   Endpoint - リモート書き込み URLおよび Endpoint - クエリ URLに表示される を書き留めURLsます。これらの URL は、このワークスペースにメトリクスをリモートで書き込むようにPrometheus サーバーを構成するときと、それらのメトリクスにクエリを実行するときに必要になります。

ワークスペースにメトリクスを取り込む方法については、「ワークスペースへの Prometheus メトリクスの取り込み」を参照してください。

Amazon Managed Service for Prometheus ワークスペースを編集する

ワークスペースを編集してエイリアスを変更できます。 AWS CLIを使用してワークスペースエイリアスを変更するには、次のコマンドを入力します。

```
aws amp update-workspace-alias --workspace-id my-workspace-id --alias "new-alias"
```

Amazon Managed Service for Prometheus コンソールを使用してワークスペースを編集するには

1. で Amazon Managed Service for Prometheus コンソールを開きますhttps://console.aws.amazon.com/prometheus/。

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. 編集するワークスペースのワークスペース ID を選択し、[編集] を選択します。
4. ワークスペースの新しいエイリアスを入力し、[保存] を選択します。

Amazon Managed Service for Prometheus ワークスペースの詳細を検索するには、以下を含めます。 ARN

Amazon Managed Service for Prometheus ワークスペースの詳細については、 コンソールまたは を使用して AWS 確認できます AWS CLI。

Console

Amazon Managed Service for Prometheus コンソールを使用してワークスペースの詳細を検索するには

1. で Amazon Managed Service for Prometheus コンソールを開きますhttps://console.aws.amazon.com/prometheus/。
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. ワークスペースのワークスペース ID を選択します。これにより、次のようなワークスペースの詳細が表示されます。

   - 現在のステータス – アクティブ などのワークスペースのステータスがステータス の下に表示されます。
   - ARN – ワークスペースARNは の下に表示されますARN。
   - ID – ワークスペース ID は、ワークスペース ID の下に表示されます。
   - URLs – コンソールには、ワークスペースへの書き込みやワークスペースからのデータのクエリURLsのための など、ワークスペースURLsの複数の が表示されます。

   > **Note**
   >
   > デフォルトでは、URLs指定された は IPv4 ですURLs。デュアルスタック (IPv4 および IPv6をサポート) を使用することもできますURLs。これらは同じですが、デフォルトの `api.aws`ではなくドメインにありますamazonaws.com。例えば、次の (URL) IPv4 が表示される場合。

```
https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
```

デュアルスタック (のサポートを含むIPv6) はURL、次のように作成できます。

```
https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/api/v1/remote_write
```

AWS CLI

を使用してワークスペースの詳細を検索するには AWS CLI

次のコマンドは、ワークスペースの詳細を返します。を置き換える必要があります *my-workspace-id* に、詳細が必要なワークスペースのワークスペース ID を指定します。

```
aws amp describe-workspace --workspace-id my-workspace-id
```

これにより、次のようなワークスペースの詳細が返されます。

- 現在のステータス – ワークスペースのステータス、例えば は ACTIVEstatusCodeプロパティに返されます。
- ARN – ワークスペースARNは arnプロパティに返されます。
- URLs – は、 prometheusEndpointプロパティ内のワークスペースURLのベース AWS CLI を返します。

> **Note**
>
> デフォルトでは、URL返される は IPv4 ですURL。デフォルトの api.awsではなく、ドメインURLでデュアルスタック (IPv4 および IPv6をサポート) を使用することもできますamazonaws.com。例えば、次の (URL) IPv4 が表示される場合。
>
> ```
> https://aps-workspaces.us-east-1.amazonaws.com/workspaces/ws-abcd1234-ef56-7890-ab12-example/
> ```
>
> デュアルスタック (のサポートを含むIPv6) はURL、次のように作成できます。

```
https://aps-workspaces.us-east-1.api.aws/workspaces/ws-abcd1234-ef56-7890-ab12-example/
```

ワークスペースURLsのリモート書き込みとクエリを作成するには、/api/v1/queryそれぞれ /api/v1/remote_writeまたは を追加します。

Amazon Managed Service for Prometheus ワークスペースを削除する

ワークスペースを削除すると、ワークスペースに取り込まれたデータが削除されます。

> **Note**
>
> Amazon Managed Service for Prometheus ワークスペースを削除しても、メトリクスをスクレイピングしてワークスペースに送信している AWS マネージドコレクターは自動的に削除されません。詳細については、「スクレイパーの検出と削除」を参照してください。

を使用してワークスペースを削除するには AWS CLI

以下のコマンドを使用します。

```
aws amp delete-workspace --workspace-id my-workspace-id
```

Amazon Managed Service for Prometheus コンソールを使用してワークスペースを削除するには

1. で Amazon Managed Service for Prometheus コンソールを開きますhttps://console.aws.amazon.com/prometheus/。
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. 削除するワークスペースのワークスペース ID を選択し、[削除] を選択します。
4. 確認ボックスに **delete** と入力し、[削除] を選択します。

Amazon Managed Service for Prometheus ワークスペースにメトリクスを取り込む

メトリクスをクエリまたはアラートするには、Amazon Managed Service for Prometheus ワークスペースにメトリクスを取り込む必要があります。このセクションでは、ワークスペースへのメトリクスの取り込みを設定する方法について説明します。

> **Note**
>
> ワークスペースに取り込まれたメトリクスは、デフォルトで 150 日間保存され、その後自動的に削除されます。この長さは、調整可能なクォータ によって制御されます。

Amazon Managed Service for Prometheus ワークスペースにメトリクスを取り込むには、2 つの方法があります。

- AWS マネージドコレクターの使用 – Amazon Managed Service for Prometheus は、Amazon Elastic Kubernetes Service (Amazon EKS) クラスターからメトリクスを自動的にスクレイプする、フルマネージド型のエージェントレススクレイパーを提供します。スクレイピングは、Prometheus 互換エンドポイントからメトリクスを自動的にプルします。
- カスタマーマネージドコレクターの使用 — 独自のコレクターを管理するためのオプションは多数あります。使用する最も一般的なコレクターの 2 つは、Prometheus の独自のインスタンスのインストール、エージェントモードでの実行、または AWS Distro for の使用です OpenTelemetry。これらの 2 つについては、次のセクションで詳しく説明します。

  コレクターは Amazon Managed Service for Prometheus に Prometheus のリモート書き込み機能を使用してメトリクスを送信します。独自のアプリケーション内の Prometheus リモート書き込みを使用して、メトリクスを Amazon Managed Service for Prometheus に直接送信できます。リモート書き込みの直接使用とリモート書き込み設定の詳細については、「Prometheus ドキュメント」の「remote_write」を参照してください。

トピック

- AWS マネージドコレクターによるメトリクスの取り込み
- カスタマーマネージドコレクター

AWS マネージドコレクターによるメトリクスの取り込み

Amazon Managed Service for Prometheus の一般的なユースケースは、Amazon Elastic Kubernetes Service (Amazon EKS) によって管理される Kubernetes クラスターを監視することです。Kubernetes クラスターや Amazon EKS 内で実行される多くのアプリケーションは、Prometheus 互換のスクレイパーがアクセスできるようにメトリクスを自動的にエクスポートします。

> **Note**
>
> Kubernetes 環境で実行される多くのテクノロジーやアプリケーションは、Prometheus 互換のメトリクスを提供しています。十分に文書化されたエクスポーターのリストについては、「Prometheus ドキュメント」の「Exporters and integrations」を参照してください。

Amazon Managed Service for Prometheus は、完全マネージド型のエージェントレススクレイパー (コレクター) を提供し、Prometheus 互換のメトリクスを自動的に検出して取得します。エージェントやスクレイパーを管理、インストール、パッチ適用、または保守する必要はありません。Amazon Managed Service for Prometheus コレクターは Amazon EKS クラスター用に、信頼性が高く、安定性があり、可用性が高く、自動的にスケーリングされるメトリクスのコレクションを提供します。Amazon Managed Service for Prometheus マネージドコレクターは、EC2 や Fargate などの Amazon EKS クラスターで動作します。

Amazon Managed Service for Prometheus コレクターは、スクレイパーの作成時に指定されたサブネットごとに Elastic Network Interface (ENI) を作成します。コレクターはこれらの ENI を介してメトリクスをスクレイピングし、`remote_write` を使って、VPC エンドポイントを使用して Amazon Managed Service for Prometheus ワークスペースにデータをプッシュします。スクレイピングされたデータが、パブリックインターネット上を移動することはありません。

以下のトピックでは、Amazon EKS クラスターで Amazon Managed Service for Prometheus コレクターを使用する方法と、収集されたメトリクスについて詳しく説明します。

トピック

- AWS マネージドコレクターの使用
- Prometheus と互換性のあるメトリクスとはどのようなものですか。

AWS マネージドコレクターの使用

Amazon Managed Service for Prometheus コレクターを使用するには、Amazon EKS クラスター内のメトリクスを検出して取得するスクレイパーを作成する必要があります。

- Amazon EKS クラスターを作成するときに、スクレイパーを作成できます。スクレイパーの作成を含め、Amazon EKS クラスターの作成に関する詳細については、「Amazon EKS ユーザーガイド」の「Amazon EKS クラスターの作成」を参照してください。
- 独自のスクレイパーは、 AWS API または を使用してプログラムで作成できます AWS CLI。

> **Note**
>
> カスタマーマネージドキーで作成された Amazon Managed Service for Prometheus ワークスペースでは、 AWS マネージドコレクターを取り込みに使用することはできません。

Amazon Managed Service for Prometheus コレクターは、Prometheus と互換性のあるメトリクスをスクレイピングします。Prometheus 互換メトリクスの詳細については、「Prometheus と互換性のあるメトリクスとはどのようなものですか。」を参照してください。

> **Note**
>
> クラスターからメトリクスをスクレイピングすると、クロスリージョントラフィックなど、ネットワークの使用に対して料金が発生する可能性があります。これらのコストを最適化する 1 つの方法は、エンドポイントを設定`/metrics`して、提供されたメトリクス (gzip など) を圧縮し、ネットワーク上で移動する必要があるデータを減らすことです。これを行う方法は、メトリクスを提供するアプリケーションまたはライブラリによって異なります。一部のライブラリはデフォルトで gzip です。

以下のトピックでは、スクレイパーを作成、管理、および設定する方法について説明します。

トピック

- スクレイパーの作成
- Amazon EKS クラスターの設定
- スクレイパーの検出と削除
- スクレイパー設定

- スクレイパー設定のトラブルシューティング
- スクレイパーの制限事項

スクレイパーの作成

Amazon Managed Service for Prometheus コレクターは、Amazon EKS クラスターからメトリクスを検出して収集するスクレイパーで構成されています。Amazon Managed Service for Prometheus ではお客様に代わってスクレイパーが管理されます。インスタンス、エージェント、スクレイパーをご自身で管理しなくても、必要なスケーラビリティ、セキュリティ、信頼性を実現できます。

Amazon EKS コンソールから Amazon EKS クラスターを作成すると、スクレイパーが自動的に作成されます。ただし、状況によっては、ご自身でスクレイパーを作成したい場合もあるでしょう。例えば、 AWS マネージドコレクターを既存の Amazon EKS クラスターに追加する場合や、既存のコレクターの設定を変更する場合などです。

AWS API または を使用してスクレイパーを作成できます AWS CLI。

独自のスクレイパーを作成するには、いくつかの前提条件があります。

- Amazon EKS クラスターが作成済みである必要があります。
- Amazon EKS クラスターは、クラスターエンドポイントアクセスコントロールがプライベートアクセスを含むように設定されている必要があります。プライベートとパブリックを含めることができますが、プライベートを含める必要があります。

> **Note**
>
> クラスターは、Amazon リソースネーム (ARN) によってスクレイパーに関連付けられます。クラスターを削除し、同じ名前の新しいクラスターを作成すると、ARN は新しいクラスターに再利用されます。このため、スクレイパーは新しいクラスターのメトリクスの収集を試みます。スクレイパーは、クラスターの削除とは別に削除します。

AWS API

AWS API を使用してスクレイパーを作成するには

`CreateScraper` API オペレーションを使用して AWS API を含むスクレイパーを作成します。次の例では、`us-west-2` リージョンでスクレイパーを作成します。 AWS アカウント、ワーク

スペース、セキュリティ、Amazon EKS クラスターの情報を独自の IDsに置き換え、スクレイパーに使用する設定を指定する必要があります。

> **Note**
>
> セキュリティグループとサブネットは、接続先のクラスターのセキュリティグループとサブネットに設定する必要があります。
> 少なくとも 2 つ以上のアベイラビリティーゾーンにある 2 つ以上のサブネットを含める必要があります。

`scrapeConfiguration` は、base64 でエンコードされた Prometheus 設定 YAML ファイルです。`GetDefaultScraperConfiguration` API オペレーションで汎用設定をダウンロードできます。の形式の詳細については、`scrapeConfiguration`「」を参照してくださいスクレイパー設定。

```
POST /scrapers HTTP/1.1
Content-Length: 415
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: aws-cli/1.18.147 Python/2.7.18 Linux/5.4.58-37.125.amzn2int.x86_64
 botocore/1.18.6

{
    "alias": "myScraper",
    "destination":  {
        "ampConfiguration": {
            "workspaceArn": "arn:aws:aps:us-west-2:account-id:workspace/
ws-workspace-id"
        }
    },
    "source": {
        "eksConfiguration": {
            "clusterArn": "arn:aws:eks:us-west-2:account-id:cluster/cluster-name",
            "securityGroupIds": ["sg-security-group-id"],
            "subnetIds": ["subnet-subnet-id-1", "subnet-subnet-id-2"]
        }
    },
    "scrapeConfiguration": {
        "configurationBlob": <base64-encoded-blob>
    }
```

```
}
```

AWS CLI

を使用してスクレイパーを作成するには AWS CLI

`create-scraper` コマンドを使用して、 でスクレイパーを作成します AWS CLI。次の例では、`us-west-2` リージョンでスクレイパーを作成します。 AWS アカウント、ワークスペース、セキュリティ、Amazon EKS クラスターの情報を独自の IDsに置き換え、スクレイパーに使用する設定を指定する必要があります。

> **Note**
>
> セキュリティグループとサブネットは、接続先のクラスターのセキュリティグループとサブネットに設定する必要があります。
> 少なくとも 2 つ以上のアベイラビリティーゾーンにある 2 つ以上のサブネットを含める必要があります。

`scrape-configuration` は、base64 でエンコードされた Prometheus 設定 YAML ファイルです。`get-default-scraper-configuration` コマンドを使用して汎用設定をダウンロードできます。の形式の詳細については、`scrape-configuration`「」を参照してくださいスクレイパー設定。

```
aws amp create-scraper \
  --source eksConfiguration="{clusterArn='arn:aws:eks:us-west-2:account-
id:cluster/cluster-name', securityGroupIds=['sg-security-group-
id'],subnetIds=['subnet-subnet-id-1', 'subnet-subnet-id-2']}" \
  --scrape-configuration configurationBlob=<base64-encoded-blob> \
  --destination ampConfiguration="{workspaceArn='arn:aws:aps:us-west-2:account-
id:workspace/ws-workspace-id'}"
```

以下は、 AWS API で使用できるスクレイパーオペレーションの完全なリストです。

- CreateScraper API オペレーションを使用してスクレイパーを作成します。
- ListScrapers API オペレーションを使用して、既存のスクレイパーを一覧表示します。
- DeleteScraper API オペレーションを使用してスクレイパーを削除します。
- DescribeScraper API オペレーションを使用してスクレイパーの詳細を取得します。

- GetDefaultScraperConfiguration API オペレーションを使用してスクレイパーの汎用設定を取得します。

> **Note**
>
> スクレイピングする Amazon EKS クラスターは、Amazon Managed Service for Prometheus がメトリクスにアクセスできるように設定されている必要があります。次のトピックでは、クラスターの設定方法について説明します。

スクレイパー作成時の一般的なエラー

以下は、新しいスクレイパーを作成しようとするときに発生する最も一般的な問題です。

- 必要な AWS リソースは存在しません。指定されたセキュリティグループ 、サブネット 、および Amazon EKS クラスターが存在している必要があります。
- IP アドレス空間が不十分です。`CreateScraper` API に渡すサブネットごとに、少なくとも 1 つの IP アドレスが必要です。

Amazon EKS クラスターの設定

Amazon EKS クラスターは、スクレイパーがメトリクスにアクセスできるように設定する必要があります。この設定には 2 つのオプションがあります。

- Amazon EKS アクセスエントリを使用して、Amazon Managed Service for Prometheus コレクターにクラスターへのアクセスを自動的に提供します。
- マネージドメトリクススクレイピング用に Amazon EKS クラスターを手動で設定します。

以下のトピックでは、これらの各項目について詳しく説明します。

アクセスエントリを使用してスクレイパーアクセス用に Amazon EKS を設定する

Amazon EKS のアクセスエントリを使用することは、Amazon Managed Service for Prometheus にクラスターからメトリクスをスクレイプするためのアクセスを許可する最も簡単な方法です。

スクレイピングする Amazon EKS クラスターは、API 認証を許可するように設定する必要があります。クラスター認証モードは、 `API`または のいずれかに設定する必要がありますAPI_AND_CONFIG_MAP。これは、クラスターの詳細のアクセス設定タブの Amazon EKS コン

ソールで表示できます。詳細については、「Amazon EKS ユーザーガイド」の「Amazon EKS クラスターの Kubernetes オブジェクトへのアクセスを IAM ロールまたはユーザーに許可する」を参照してください。

クラスターの作成時または作成後にスクレイパーを作成できます。

- クラスターの作成時 – Amazon EKS コンソールを使用して Amazon EKS クラスターを作成するときに (クラスターの一部としてスクレイパーを作成する指示に従って)、このアクセスを設定できます。アクセスエントリポリシーが自動的に作成され、Amazon Managed Service for Prometheus にクラスターメトリクスへのアクセスが許可されます。
- クラスターの作成後に を追加する – Amazon EKS クラスターがすでに存在する場合は、認証モードを APIまたは に設定します。Amazon Managed Service for Prometheus API または CLI を使用して作成したスクレイパーにはAPI_AND_CONFIG_MAP、自動的に正しいアクセスエントリポリシーが作成され、スクレイパーはクラスターにアクセスできます。

アクセスエントリポリシーが作成されました

スクレイパーを作成し、Amazon Managed Service for Prometheus がアクセスエントリポリシーを生成できるようにすると、次のポリシーが生成されます。アクセスエントリの詳細については、「Amazon EKS ユーザーガイド」の「IAM ロールまたはユーザーに Kubernetes へのアクセスを許可する」を参照してください。

```
{
    "rules": [
        {
            "effect": "allow",
            "apiGroups": [
                ""
            ],
            "resources": [
                "nodes",
                "nodes/proxy",
                "nodes/metrics",
                "services",
                "endpoints",
                "pods",
                "ingresses",
                "configmaps"
            ],
            "verbs": [
```

```
                "get",
                "list",
                "watch"
            ]
        },
        {
            "effect": "allow",
            "apiGroups": [
                "extensions",
                "networking.k8s.io"
            ],
            "resources": [
                "ingresses/status",
                "ingresses"
            ],
            "verbs": [
                "get",
                "list",
                "watch"
            ]
        },
        {
            "effect": "allow",
            "nonResourceURLs": [
                "/metrics"
            ],
            "verbs": [
                "get"
            ]
        }
    ]
}
```

スクレイパーアクセス用に Amazon EKS を手動で設定する

を使用して kubernetes クラスターへのアクセス`aws-auth ConfigMap`を制御する場合でも、Amazon Managed Service for Prometheus スクレイパーにメトリクスへのアクセスを許可できます。次の手順では、Amazon Managed Service for Prometheus に Amazon EKS クラスターからメトリクスをスクレイプするためのアクセス権を付与します。

> **Note**
>
> ConfigMap およびアクセスエントリの詳細については、「Amazon EKS ユーザーガイド」の「IAM ロールまたはユーザーに Kubernetes へのアクセスを許可する」を参照してください。

この手順では、 kubectlと AWS CLI を使用します。kubectl のインストールの詳細については、「Amazon EKS ユーザーガイド」の「kubectl のインストール」を参照してください。

マネージドメトリクススクレイピング用に Amazon EKS クラスターを手動で設定するには

1. clusterrole-binding.yml という名前のファイルを作成し、次のテキストを記述します。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aps-collector-role
rules:
  - apiGroups: [""]
    resources: ["nodes", "nodes/proxy", "nodes/metrics", "services", "endpoints",
 "pods", "ingresses", "configmaps"]
    verbs: ["describe", "get", "list", "watch"]
  - apiGroups: ["extensions", "networking.k8s.io"]
    resources: ["ingresses/status", "ingresses"]
    verbs: ["describe", "get", "list", "watch"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aps-collector-user-role-binding
subjects:
- kind: User
  name: aps-collector-user
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aps-collector-role
  apiGroup: rbac.authorization.k8s.io
```

2. クラスターで次のコマンドを実行します。

```
kubectl apply -f clusterrole-binding.yml
```

これにより、クラスターのロールバインディングとルールが作成されます。この例では、aps-collector-role をロール名、aps-collector-user をユーザー名として使用しています。

3. 次のコマンドは、*scraper-id* という ID のスクレイパーに関する情報を提供します。これは、前のセクションのコマンドを使用して作成したスクレイパーです。

```
aws amp describe-scraper --scraper-id scraper-id
```

4. describe-scraper の結果から roleArn を探します。この形式は次のようになります。

```
arn:aws:iam::account-id:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper_unique-id
```

Amazon EKS では、この ARN に別の形式が必要です。次のステップで使用するために、返される ARN の形式を調整する必要があります。この形式に合わせて編集してください。

```
arn:aws:iam::account-id:role/AWSServiceRoleForAmazonPrometheusScraper_unique-id
```

例えば、この ARN の場合、

```
arn:aws:iam::111122223333:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-56ef-7
```

以下のように記述する必要があります。

```
arn:aws:iam::111122223333:role/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-56ef-7
```

5. 前のステップで変更した roleArn と、クラスター名およびリージョンを使用して、クラスター内で以下のコマンドを実行します。

```
eksctl create iamidentitymapping --cluster cluster-name --region region-id --
arn roleArn --username aps-collector-user
```

これにより、スクレイパーは clusterrole-binding.yml ファイルに作成したロールとユーザーを使用してクラスターにアクセスできます。

スクレイパーの検出と削除

AWS API または を使用して AWS CLI 、アカウント内のスクレイパーを一覧表示したり、削除したりできます。

> **Note**
>
> AWS CLI または SDK の最新バージョンを使用していることを確認してください。最新バージョンでは、最新の機能やセキュリティ更新プログラムを利用できます。または、常に up-to-date コマンドラインエクスペリエンスを提供する AWS Cloudshell を自動的に使用します。

アカウント内のすべてのスクレイパーを一覧表示するには、 ListScrapers API オペレーションを使用します。

または、 を使用して AWS CLIを呼び出します。

```
aws amp list-scrapers
```

ListScrapers は、アカウント内のすべてのスクレイパーを返します。例:

```
{
    "scrapers": [
        {
            "scraperId": "s-1234abcd-56ef-7890-abcd-1234ef567890",
            "arn": "arn:aws:aps:us-west-2:123456789012:scraper/s-1234abcd-56ef-7890-
abcd-1234ef567890",
            "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
AWSServiceRoleForAmazonPrometheusScraper_1234abcd-2931",
            "status": {
                "statusCode": "DELETING"
            },
            "createdAt": "2023-10-12T15:22:19.014000-07:00",
            "lastModifiedAt": "2023-10-12T15:55:43.487000-07:00",
            "tags": {},
            "source": {
```

```
                "eksConfiguration": {
                    "clusterArn": "arn:aws:eks:us-west-2:123456789012:cluster/my-
cluster",
                    "securityGroupIds": [
                        "sg-1234abcd5678ef90"
                    ],
                    "subnetIds": [
                        "subnet-abcd1234ef567890",
                        "subnet-1234abcd5678ab90"
                    ]
                }
            },
            "destination": {
                "ampConfiguration": {
                    "workspaceArn": "arn:aws:aps:us-west-2:123456789012:workspace/
ws-1234abcd-5678-ef90-ab12-cdef3456a78"
                }
            }
        }
    ]
}
```

スクレイパーを削除するには、 ListScrapersオペレーションを使用して削除するscraperIdスクレイパーの を見つけ、 DeleteScraperオペレーションを使用して削除します。

または、 を使用して AWS CLIを呼び出します。

```
aws amp delete-scraper --scraper-id scraperId
```

スクレイパー設定

Prometheus 互換のスクレイパー設定を使用して、スクレイパーがメトリクスを検出して収集する方法を制御できます。例えば、メトリクスをワークスペースに送信する間隔を変更できます。再ラベル付けを使用して、メトリクスのラベルを動的に書き換えることもできます。スクレイパー設定は、スクレイパーの定義の一部である YAML ファイルです。

新しいスクレイパーを作成したら、API コールで base64 でエンコードされた YAML ファイルを提供して設定を指定します。Amazon Managed Service for Prometheus API の GetDefaultScraperConfiguration オペレーションを含む汎用設定ファイルをダウンロードできます。

スクレイパーの設定を変更するには、スクレイパーを削除し、新しい設定で再作成します。

サポートされている設定

可能な値の詳細な内訳を含むスクレイパー設定形式の詳細については、Prometheus ドキュメントの「設定」を参照してください。グローバル設定オプションと <scrape_config> オプションには、最も一般的に必要なオプションが記載されています。

Amazon EKS はサポートされている唯一のサービスであるため、サポートされているサービス検出設定 (<*_sd_config>) は のみです<kubernetes_sd_config>。

許可される設定セクションの完全なリスト：

- <global>
- <scrape_config>
- <static_config>
- <relabel_config>
- <metric_relabel_configs>
- <kubernetes_sd_config>

これらのセクション内の制限事項は、サンプル設定ファイルの後に一覧表示されます。

設定ファイルの例

以下は、30 秒のスクレイプ間隔の YAML 設定ファイルのサンプルです。

```
global:
   scrape_interval: 30s
   external_labels:
     clusterArn: apiserver-test-2
scrape_configs:
  - job_name: pod_exporter
    kubernetes_sd_configs:
      - role: pod
  - job_name: cadvisor
    scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    kubernetes_sd_configs:
      - role: node
    relabel_configs:
```

```
      - action: labelmap
        regex: __meta_kubernetes_node_label_(.+)
      - replacement: kubernetes.default.svc:443
        target_label: __address__
      - source_labels: [__meta_kubernetes_node_name]
        regex: (.+)
        target_label: __metrics_path__
        replacement: /api/v1/nodes/$1/proxy/metrics/cadvisor
  # apiserver metrics
  - scheme: https
    authorization:
      type: Bearer
      credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
    job_name: kubernetes-apiservers
    kubernetes_sd_configs:
    - role: endpoints
    relabel_configs:
    - action: keep
      regex: default;kubernetes;https
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_service_name
      - __meta_kubernetes_endpoint_port_name
  # kube proxy metrics
  - job_name: kube-proxy
    honor_labels: true
    kubernetes_sd_configs:
    - role: pod
    relabel_configs:
    - action: keep
      source_labels:
      - __meta_kubernetes_namespace
      - __meta_kubernetes_pod_name
      separator: '/'
      regex: 'kube-system/kube-proxy.+'
    - source_labels:
      - __address__
      action: replace
      target_label: __address__
      regex: (.+?)(\\:\\d+)?
      replacement: $1:10249
```

以下は、 AWS マネージドコレクターに固有の制限です。

- スクレイプ間隔 — スクレイパー設定では、30 秒未満のスクレイプ間隔を指定できません。
- ターゲット — static_config 内のターゲットは IP アドレスとして指定する必要があります。
- DNS 解決 — ターゲット名に関連して、この設定で認識されるサーバー名は Kubernetes API サーバー のみですkubernetes.default.svc。他のすべてのマシン名は IP アドレスで指定する必要があります。
- 承認 – 承認が必要ない場合は省略します。必要な場合、認証は でなければならずBearer、ファイル を指す必要があります/var/run/secrets/kubernetes.io/serviceaccount/token。つまり、使用すると、承認セクションは次のようになります。

```
authorization:
  type: Bearer
  credentials_file: /var/run/secrets/kubernetes.io/serviceaccount/token
```

> **Note**
>
> type: Bearer がデフォルトであるため、省略できます。

スクレイパー設定のトラブルシューティング

Amazon Managed Service for Prometheus コレクターは、メトリクスの検出と収集を自動的に行います。しかし、予想したメトリクスが Amazon Managed Service for Prometheus ワークスペースに表示されない場合、どのようにトラブルシューティングできるでしょうか。

up メトリクスは便利なツールです。Amazon Managed Service for Prometheus コレクターが検出した各エンドポイントについて、このメトリクスは自動的に送信されます。このメトリクスには 3 つの状態があり、コレクター内で発生している問題のトラブルシューティングに役立ちます。

- up が存在しない — エンドポイントの up メトリクスが存在しない場合、コレクターがエンドポイントを検出できなかったことを意味します。

  エンドポイントが存在することが確実な場合は、スクレイプ設定を調整する必要がある可能性があります。検出の relabel_config の調整が必要な場合もあれば、検出に使用される role に問題がある可能性もあります。
- up は存在するものの、常に 0 — up が存在するが 0 の場合、コレクターはエンドポイントを検出できますが、Prometheus 互換のメトリクスを検出できません。

この場合は、curl エンドポイントに対して直接コマンドを実行してみるといいかもしれません。使用しているプロトコル (http または https) 、エンドポイント、ポートなど、詳細が正しいことを検証できます。エンドポイントが有効な200レスポンスで応答していること、および Prometheus 形式に従っていることを確認することもできます。最後に、レスポンスの本文を最大許容サイズより大きくすることはできません。(AWS マネージドコレクターの制限については、次のセクションを参照してください。)

- up が存在し、0 より大きい — up が存在し、かつ 0 より大きい場合、メトリクスは Amazon Managed Service for Prometheus に送信されています。

  Amazon Managed Service for Prometheus (または Amazon Managed Grafana などの代替ダッシュボード) で正しいメトリクスを検出していることを確認します。curl をもう一度使用して、/metrics エンドポイントに予想したデータがあるかどうかを確認できます。また、スクレイパーあたりのエンドポイント数など、他の制限を超えていないことも確認してください。を使用してメトリクスの数を確認することで、スクレイピングされるupメトリクスエンドポイントの数を確認できますcount(up)。

スクレイパーの制限事項

Amazon Managed Service for Prometheus が提供するフルマネージド型スクレイパーには、いくつかの制限があります。

- リージョン — EKS クラスター、マネージドスクレイパー、Amazon Managed Service for Prometheus ワークスペースはすべて同じ AWS リージョンにある必要があります。
- アカウント — EKS クラスター、マネージドスクレイパー、Amazon Managed Service for Prometheus ワークスペースはすべて同じ AWS アカウントにある必要があります。
- コレクター — 1 リージョンの 1 アカウントあたり、最大 10 個の Amazon Managed Service for Prometheus スクレイパーを設定できます。

  > **Note**
  >
  > クォータの引き上げをリクエストすることで、この上限を引き上げることができます。

- メトリクスレスポンス — 任意の 1 つの /metrics エンドポイントリクエストからのレスポンスの本文は 50 メガバイト (MB) を超えることはできません。
- スクレイパーあたりのエンドポイント — スクレイパーは最大 30,000 の /metrics エンドポイントをスクレイピングできます。

- スクレイプ間隔 — スクレイパー設定では、30 秒未満のスクレイプ間隔を指定できません。

Prometheus と互換性のあるメトリクスとはどのようなものですか。

Prometheus メトリクスをアプリケーションやインフラストラクチャからスクレイピングして Amazon Managed Service for Prometheus で使用するには、Prometheus 互換の `/metrics` エンドポイントから Prometheus 互換のメトリクスをインストルメントして公開する必要があります。独自のメトリクスを実装することができますが、必須ではありません。Kubernetes (Amazon EKS を含む) や他の多くのライブラリやサービスは、これらのメトリクスを直接実装しています。

Amazon EKS のメトリクスを Prometheus 互換のエンドポイントにエクスポートすると、それらのメトリクスを Amazon Managed Service for Prometheus コレクターで自動的にスクレイピングすることができます。

詳細については、次のトピックを参照してください。

- メトリクスを Prometheus メトリクスとしてエクスポートする既存のライブラリとサービスの詳細については、「Prometheus ドキュメント」の「Exporters and integrations」を参照してください。
- Prometheus 互換メトリクスを独自のコードからエクスポートする方法の詳細については、「Prometheus ドキュメント」の「Writing exporters」を参照してください。
- Amazon Managed Service for Prometheus コレクターを設定して Amazon EKS クラスターからメトリクスを自動的にスクレイピングする方法の詳細については、「AWS マネージドコレクターの使用」を参照してください。

カスタマーマネージドコレクター

このセクションには、Prometheus リモート書き込みを使用して Amazon Managed Service for Prometheus にメトリクスを送信する独自のコレクターを設定してデータを取り込む方法に関する情報が含まれています。

独自のコレクターを使用して Amazon Managed Service for Prometheus にメトリクスを送信する場合、メトリクスを保護し、取り込みプロセスが可用性のニーズを満たしていることを確認する責任はお客様にあります。

ほとんどのカスタマーマネージドコレクターは、以下のツールのいずれかを使用します。

- AWS Distro for OpenTelemetry （ ADOT ） – ADOTは、完全にサポートされ、安全で、本番環境に対応した のオープンソースディストリビューション OpenTelemetry であり、エージェントがメトリクスを収集できるようにします。を使用してメトリクスADOTを収集し、Amazon Managed Service for Prometheus ワークスペースに送信できます。ADOT コレクターの詳細については、AWS 「 Distro for OpenTelemetry」を参照してください。
- Prometheus エージェント — オープンソースの Prometheus サーバーの独自のインスタンスをセットアップし、エージェントとして実行することでメトリクスを収集し Amazon Managed Service for Prometheus ワークスペースに転送することができます。

以下のトピックは、これら両方のツールの使用方法について説明し、独自のコレクターの設定に関する一般的な情報も含んでいます。

トピック

メトリクスの取り込みの保護

Amazon Managed Service for Prometheus には、メトリクスの取り込みを保護するための手段が用意されています。

Amazon Managed Service for Prometheus AWS PrivateLink での の使用

Amazon Managed Service for Prometheus にメトリクスを取り込むネットワークトラフィックは、パブリックインターネットエンドポイントを介して、または を介してVPCエンドポイントによって実行できます AWS PrivateLink。 AWS PrivateLink を使用すると、 からのネットワークトラフィックVPCsが、パブリックインターネットを経由せずに AWS ネットワーク内で保護されます。Amazon Managed Service for Prometheus のエンドポイントを作成するには AWS PrivateLink VPC、「」を参照してくださいインターフェイス VPC エンドポイントでの Amazon Managed Service for Prometheus の使用。

認証と認可

AWS Identity and Access Management (IAM) は、 AWS リソースへのアクセスを安全に制御するのに役立つウェブサービスです。を使用してIAM、誰を認証 (サインイン) し、誰にリソースの使用を

承認する (アクセス許可を付与する) かを制御します。Amazon Managed Service for Prometheus はと統合されIAM、データの安全性の維持に役立ちます。Amazon Managed Service for Prometheus をセットアップするときは、Prometheus サーバーからメトリクスを取り込むためのIAMロールと、Grafana サーバーが Amazon Managed Service for Prometheus ワークスペースに保存されているメトリクスをクエリできるようにするロールを作成する必要があります。の詳細についてはIAM、「 IAMとは」を参照してください。

Amazon Managed Service for Prometheus のセットアップに役立つもう 1 つの AWS セキュリティ機能は、 AWS 署名バージョン 4 の署名プロセス (AWS SigV4) です。署名バージョン 4 は、 によって送信された AWS リクエストに認証情報を追加するプロセスですHTTP。セキュリティ上の理由から、 へのほとんどのリクエストは、アクセスキー ID とシークレットアクセスキーで構成されるアクセスキーで署名 AWS する必要があります。これらの 2 つのキーは、一般的にセキュリティ認証情報と呼ばれます。SigV4 の詳細については、「Signature Version 4 の署名プロセス」を参照してください。

Distro for AWS をコレクター OpenTelemetry として使用する

このセクションでは、Prometheus AWS で計測されたアプリケーションからスクレイプし、メトリクスを Amazon Managed Service for Prometheus に送信するように Distro for OpenTelemetry (ADOT) Collector を設定する方法について説明します。ADOT コレクターの詳細については、AWS「 Distro for OpenTelemetry」を参照してください。

以下のトピックでは、メトリクスが Amazon 、Amazon EKS、ECSまたは Amazon EC2インスタンスのいずれから来ているかに基づいて、メトリクスのコレクターADOTとして を設定する 3 つの異なる方法について説明します。

トピック

- Amazon Elastic Kubernetes Service AWS クラスター OpenTelemetry で Distro for を使用してメトリクスの取り込みを設定する
- Distro for Open Telemetry AWS ECSを使用して Amazon からのメトリクスの取り込みを設定する
- リモート書き込みを使用して Amazon EC2インスタンスからのメトリクスの取り込みを設定する

Amazon Elastic Kubernetes Service AWS クラスター OpenTelemetry で Distro for を使用してメトリクスの取り込みを設定する

AWS Distor for OpenTelemetry （ ADOT) コレクターを使用して、Prometheus で計測されたアプリケーションからメトリクスをスクレイプし、そのメトリクスを Amazon Managed Service for Prometheus に送信できます。

> **Note**
>
> ADOT コレクターの詳細については、AWS 「 Distro for OpenTelemetry」を参照してください。
> Prometheus で計測されたアプリケーションの詳細については、「」を参照してくださいPrometheus と互換性のあるメトリクスとはどのようなものですか。。

で Prometheus メトリクスを収集するには、Prometheus レシーバー、Prometheus Remote Write Exporter、および Sigv4 Authentication Extension の 3 つの OpenTelemetry コンポーネントADOTが必要です。

既存の Prometheus の設定を使用して Prometheus Receiver を構成して、サービス検出とメトリクスのスクレイピングを実行できます。Prometheus Receiver は、メトリクスを Prometheus 公開形式でスクレイピングします。スクレイピング対象のアプリケーションやエンドポイントは、Prometheus クライアントライブラリで構成する必要があります。Prometheus Receiver は、Prometheus ドキュメントの「Configuration」で説明されている Prometheus のスクレイピングと再ラベル付けの設定をすべてサポートしています。これらの設定は、ADOTコレクター設定に直接貼り付けることができます。

Prometheus Remote Write Exporter は、`remote_write` エンドポイントを使用して、スクレイピングされたメトリクスを管理ポータルワークスペースに送信します。データのエクスポートHTTPリクエストは、 AWS SigV4 Authentication Extension を使用して、安全な認証の AWS プロトコルである Sigv4 で署名されます。詳細については、「Signature Version 4 の署名プロセス」を参照してください。

コレクターは Amazon 上の Prometheus メトリクスエンドポイントを自動的に検出EKS し、<kubernetes_sd_config> にある設定を使用します。

以下のデモは、Amazon Elastic Kubernetes Service または自己管理型 Kubernetes を実行しているクラスターでのこの設定の例を示しています。これらのステップを実行するには、デフォルトの認証情報チェーンの潜在的なオプションのいずれかからの AWS 認証情報が必要です AWS 。詳細については、「Go の AWS SDK の設定」を参照してください。このデモでは、プロセスの統合テストに使用されるサンプルアプリを使用します。このサンプルアプリは、Prometheus クライアントライブラリのように、`/metrics` エンドポイントでメトリクスを公開します。

前提条件

以下の取り込み設定ステップを開始する前に、サービスアカウントと信頼ポリシーのIAMロールを設定する必要があります。

サービスアカウントと信頼ポリシーのIAMロールを設定するには

1. 「」の手順に従って、サービスアカウントのIAMロールを作成しますAmazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定。

   ADOT コレクターは、メトリクスをスクレイピングしてエクスポートするときにこのロールを使用します。

2. 次に、信頼ポリシーを編集します。でIAMコンソールを開きますhttps://console.aws.amazon.com/iam/。
3. 左側のナビゲーションペインでロールを選択し、ステップ 1 でamp-iamproxy-ingest-role作成した を見つけます。
4. [信頼関係] タブを選択し、[信頼関係の編集] を選択します。
5. 信頼関係ポリシー でJSON、 を aws-ampに置き換えadot-col、信頼ポリシー の更新を選択します。最終的なポリシーは次のようになります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::account-id:oidc-provider/
oidc.eks.region.amazonaws.com/id/openid"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region.amazonaws.com/id/openid:sub":
 "system:serviceaccount:adot-col:amp-iamproxy-ingest-service-account"
        }
      }
    }
  ]
}
```

6. [アクセス許可] タブを選択し、次のアクセス許可ポリシーがロールにアタッチされていることを確認します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:RemoteWrite",
                "aps:GetSeries",
                "aps:GetLabels",
                "aps:GetMetricMetadata"
            ],
            "Resource": "*"
        }
    ]
}
```

Prometheus メトリクスの収集の有効化

> **Note**
>
> Amazon で名前空間を作成するalertmanagerとEKS、ノードエクスポーターはデフォルトで無効になります。

Amazon EKSまたは Kubernetes クラスターで Prometheus コレクションを有効にするには

1. のリポジトリからサンプルアプリケーションをフォークしてクローンしますaws-otel-community。

   次に、以下のコマンドを実行します。

```
cd ./sample-apps/prometheus-sample-app
docker build . -t prometheus-sample-app:latest
```

2. このイメージを Amazon ECRや などのレジストリにプッシュします DockerHub。

3. 次のように Kubernetes 設定をコピーして適用し、サンプルアプリをクラスターにデプロイします。prometheus-sample-app.yaml ファイル内の {{PUBLIC_SAMPLE_APP_IMAGE}} は、先ほどプッシュしたイメージに置き換えます。

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/
main/examples/eks/aws-prometheus/prometheus-sample-app.yaml -o prometheus-sample-
app.yaml
kubectl apply -f prometheus-sample-app.yaml
```

4. 次のコマンドを入力して、サンプルアプリが起動したことを確認します。コマンドの出力で、NAME 列に prometheus-sample-app が表示されます。

```
kubectl get all -n aoc-prometheus-pipeline-demo
```

5. Collector ADOT のデフォルトインスタンスを起動します。これを行うには、まず次のコマンドを入力して、ADOTコレクターの Kubernetes 設定をプルします。

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
examples/eks/aws-prometheus/prometheus-daemonset.yaml -o prometheus-daemonset.yaml
```

次に、テンプレートファイルを編集して、YOUR_ENDPOINT を Amazon Managed Service for Prometheus ワークスペースの remote_write エンドポイントに、YOUR_REGION を使用中のリージョンに置き換えます。ワークスペースの詳細を確認したときに Amazon Managed Service for Prometheus コンソールに表示される remote_write エンドポイントを使用してください。

また、Kubernetes 設定のサービスアカウントセクションYOUR_ACCOUNT_IDの を AWS アカウント ID に変更する必要があります。

この例では、ADOTコレクター設定は注釈 (scrape=true) を使用して、スクレイピングするターゲットエンドポイントを指定します。これにより、ADOTコレクターはサンプルアプリケーションエンドポイントをクラスター内の kube-system エンドポイントと区別できます。別のサンプルアプリをスクレイピングする場合は、これを再ラベル付けの設定から削除できます。

6. ADOT コレクターをデプロイするには、次のコマンドを入力します。

```
kubectl apply -f prometheus-daemonset.yaml
```

7. 次のコマンドを入力して、ADOTコレクターが起動したことを確認します。NAMESPACE 列で adot-col を探してください。

```
kubectl get pods -n adot-col
```

8. ログエクスポーターを使用して、パイプラインが機能することを確認します。サンプルテンプレートは既にログエクスポーターと統合されています。次のコマンドを入力します。

```
kubectl get pods -A
kubectl logs -n adot-col name_of_your_adot_collector_pod
```

サンプルアプリからスクレイピングされたメトリクスの一部は、次の例のようになります。

```
Resource labels:
     -> service.name: STRING(kubernetes-service-endpoints)
     -> host.name: STRING(192.168.16.238)
     -> port: STRING(8080)
     -> scheme: STRING(http)
InstrumentationLibraryMetrics #0
Metric #0
Descriptor:
     -> Name: test_gauge0
     -> Description: This is my gauge
     -> Unit:
     -> DataType: DoubleGauge
DoubleDataPoints #0
StartTime: 0
Timestamp: 1606511460471000000
Value: 0.000000
```

9. Amazon Managed Service for Prometheus がメトリクスを受け取ったかどうかをテストするには、awscurl を使用します。このツールを使用すると、 AWS Sigv4 認証を使用してコマンドラインからHTTPリクエストを送信できるため、Amazon Managed Service for Prometheus からクエリを実行するための正しいアクセス許可で AWS 認証情報をローカルに設定する必要があります。 のインストール手順についてはawscurl、「awscurl」を参照してください。

次のコマンドの AMP_REGION と AMP_ENDPOINT は、Amazon Managed Service for Prometheus ワークスペースの情報に置き換えます。

```
awscurl --service="aps" --region="AMP_REGION" "https://AMP_ENDPOINT/api/v1/query?
query=adot_test_gauge0"
{"status":"success","data":{"resultType":"vector","result":[{"metric":
{"__name__":"adot_test_gauge0"},"value":[1606512592.493,"16.87214000011479"]}]}}
```

レスポンスとしてメトリクスを受け取れば、パイプラインの設定が成功し、サンプルアプリから Amazon Managed Service for Prometheus にメトリクスが正常に伝搬されたことを意味します。

クリーンアップ

このデモをクリーンアップするには、次のコマンドを入力します。

```
kubectl delete namespace aoc-prometheus-pipeline-demo
kubectl delete namespace adot-col
```

高度な設定

Prometheus Receiver は、Prometheus ドキュメントの「Configuration」で説明されている Prometheus のスクレイピングと再ラベル付けの設定をすべてサポートしています。これらの設定は、ADOTコレクター設定に直接貼り付けることができます。

Prometheus Receiver の設定には、サービス検出、スクレイピング設定、再ラベル設定が含まれます。レシーバーの設定は次のようになります。

```
receivers:
  prometheus:
    config:
      [Your Prometheus configuration]
```

設定ファイルの例を以下に示します。

```
receivers:
  prometheus:
    config:
      global:
        scrape_interval: 1m
        scrape_timeout: 10s

      scrape_configs:
      - job_name: kubernetes-service-endpoints
        sample_limit: 10000
        kubernetes_sd_configs:
        - role: endpoints
```

```
        tls_config:
          ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
          insecure_skip_verify: true
        bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
```

既存の Prometheus 設定がある場合は、値が環境変数で置き換えられないように、$ 文字を $$ に置き換える必要があります。*これは、relabel_configurations の replacement の値で特に重要です。例えば、次のような relabel_configurations があるとします。

```
relabel_configs:
- source_labels:
 [__meta_kubernetes_ingress_scheme,__address__,__meta_kubernetes_ingress_path]
  regex: (.+);(.+);(.+)
  replacement: ${1}://${2}${3}
  target_label: __param_target
```

これは次のように変更します。

```
relabel_configs:
- source_labels:
 [__meta_kubernetes_ingress_scheme,__address__,__meta_kubernetes_ingress_path]
  regex: (.+);(.+);(.+)
  replacement: $${1}://${2}${3}
  target_label: __param_target
```

Prometheus Remote Write Exporter と Sigv4 Authentication Extension

Prometheus Remote Write Exporter と Sigv4 Authentication Extension の設定は、Prometheus Receiver よりも簡単です。パイプラインのこの段階では、既にメトリクスが取り込まれていて、このデータを Amazon Managed Service for Prometheus にエクスポートする準備ができています。次の例は、Amazon Managed Service for Prometheus と通信するための適切な設定の最小要件を示しています。

```
extensions:
  sigv4auth:
    service: "aps"
    region: "user-region"
exporters:
  prometheusremotewrite:
    endpoint: "https://aws-managed-prometheus-endpoint/api/v1/remote_write"
    auth:
```

```
      authenticator: "sigv4auth"
```

この設定は、デフォルトの AWS 認証情報チェーンの AWS 認証情報を使用して AWS SigV4 によって署名されたHTTPSリクエストを送信します。詳細については、「Configuring the AWS SDK for Go」を参照してください。サービスには aps を指定する必要があります。

デプロイ方法に関係なく、ADOTコレクターはデフォルトの AWS 認証情報チェーンにリストされているオプションのいずれかにアクセスできる必要があります。Sigv4 Authentication Extension はに依存し AWS SDK for Go 、それを使用して認証情報を取得して認証します。これらの認証情報に、Amazon Managed Service for Prometheus のリモート書き込みアクセス許可があることを確認する必要があります。

Distro for Open Telemetry AWS ECSを使用して Amazon からのメトリクスの取り込みを設定する

このセクションでは、 Distro for Open Telemetry (ECS) を使用して Amazon Elastic Container Service (Amazon) からメトリクスを収集し、Amazon Managed Service for Prometheus AWS に取り込む方法について説明しますADOT。また、Amazon Managed Grafana でメトリクスを視覚化する方法についても説明します。

前提条件

> **Important**
>
> 開始する前に、デフォルト設定の クラスター、 AWS Fargate Amazon Managed Service for Prometheus ワークスペース、および Amazon Managed Grafana ワークスペースを持つ Amazon ECS環境が必要です。ユーザーがコンテナのワークロード、Amazon Managed Service for Prometheus、Amazon Managed Grafana に精通していることを前提としています。

詳細については、以下のリンクを参照してください。

- デフォルト設定で Fargate クラスターに Amazon ECS環境を作成する方法については、「Amazon デベロッパーガイド」の「クラスターの作成ECS」を参照してください。
- Amazon Managed Service for Prometheus ワークスペースを作成する方法については、「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 Grafana ワークスペースを作成する方法については、「Amazon Managed Grafana User Guide」の「Creating a workspace」を参照してください。

ステップ 1: カスタムADOTコレクターコンテナイメージを定義する

次の設定ファイルをテンプレートとして使用して、独自のADOTコレクターコンテナイメージを定義します。置換 *my-remote-URL* また、*my-region* endpoint と regionの値を使用します。設定を adot-config.yaml というファイルに保存します。

> **Note**
>
> この設定では、sigv4auth 拡張機能を使用して Amazon Managed Service for Prometheus への呼び出しを認証します。の設定の詳細についてはsigv4auth、 の「Authenticator - Sigv4」を参照してください GitHub。

```
receivers:
  prometheus:
    config:
      global:
        scrape_interval: 15s
        scrape_timeout: 10s
      scrape_configs:
      - job_name: "prometheus"
        static_configs:
        - targets: [ 0.0.0.0:9090 ]
  awsecscontainermetrics:
    collection_interval: 10s
processors:
  filter:
    metrics:
      include:
        match_type: strict
        metric_names:
          - ecs.task.memory.utilized
          - ecs.task.memory.reserved
          - ecs.task.cpu.utilized
          - ecs.task.cpu.reserved
          - ecs.task.network.rate.rx
          - ecs.task.network.rate.tx
          - ecs.task.storage.read_bytes
```

```
          - ecs.task.storage.write_bytes
exporters:
  prometheusremotewrite:
    endpoint: my-remote-URL
    auth:
      authenticator: sigv4auth
  logging:
    loglevel: info
extensions:
  health_check:
  pprof:
    endpoint: :1888
  zpages:
    endpoint: :55679
  sigv4auth:
    region: my-region
    service: aps
service:
  extensions: [pprof, zpages, health_check, sigv4auth]
  pipelines:
    metrics:
      receivers: [prometheus]
      exporters: [logging, prometheusremotewrite]
    metrics/ecs:
      receivers: [awsecscontainermetrics]
      processors: [filter]
      exporters: [logging, prometheusremotewrite]
```

ステップ 2: ADOTコレクターコンテナイメージを Amazon ECRリポジトリにプッシュする

Dockerfile を使用してコンテナイメージを作成し、Amazon Elastic Container Registry (ECR) リポジトリにプッシュします。

1. Dockerfile を構築してコンテナイメージをコピーし、Docker OTEL イメージに追加します。

```
FROM public.ecr.aws/aws-observability/aws-otel-collector:latest
COPY adot-config.yaml /etc/ecs/otel-config.yaml
CMD ["--config=/etc/ecs/otel-config.yaml"]
```

2. Amazon ECRリポジトリを作成します。

```
# create repo:
COLLECTOR_REPOSITORY=$(aws ecr create-repository --repository aws-otel-collector \
```

```
                                --query repository.repositoryUri --output text)
```

3. コンテナイメージを作成します。

```
# build ADOT collector image:
docker build -t $COLLECTOR_REPOSITORY:ecs .
```

> **Note**
>
> コンテナのビルドは、そのコンテナが実行される環境と同じ環境で行うことを前提としています。そうでない場合、イメージのビルド時に `--platform` パラメータの使用が必要になることがあります。

4. Amazon ECRリポジトリにサインインします。置換 *my-region* を自分のregion値で指定します。

```
# sign in to repo:
aws ecr get-login-password --region my-region | \
        docker login --username AWS --password-stdin $COLLECTOR_REPOSITORY
```

5. コンテナイメージをプッシュします。

```
# push ADOT collector image:
docker push $COLLECTOR_REPOSITORY:ecs
```

ステップ 3: Amazon Managed Service for Prometheus をスクレイプする Amazon ECSタスク定義を作成する

Amazon Managed Service for Prometheus をスクレイプする Amazon ECSタスク定義を作成します。タスク定義には、`adot-collector` という名前のコンテナと、`prometheus` という名前のコンテナを含める必要があります。`prometheus` はメトリクスを生成し、`adot-collector` は `prometheus` をスクレイピングします。

> **Note**
>
> Amazon Managed Service for Prometheus はサービスとして実行され、コンテナからメトリクスを収集します。この場合のコンテナは、Prometheus をエージェントモードでローカル

で実行し、ローカルのメトリクスを Amazon Managed Service for Prometheus に送信します。

例: タスク定義

以下の例は、タスク定義がどのようなものかを示しています。この例をテンプレートとして使用して、独自のタスク定義を作成できます。のimage値をリポジトリURLとイメージタグ () adot-collectorに置き換えます$COLLECTOR_REPOSITORY:ecs。adot-collector と prometheus の region の値は、使用中の region の値に置き換えます。

```
{
  "family": "adot-prom",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "adot-collector",
      "image": "account_id.dkr.ecr.region.amazonaws.com/image-tag",
      "essential": true,
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/ecs-adot-collector",
          "awslogs-region": "my-region",
          "awslogs-stream-prefix": "ecs",
          "awslogs-create-group": "True"
        }
      }
    },
    {
      "name": "prometheus",
      "image": "prom/prometheus:main",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/ecs-prom",
          "awslogs-region": "my-region",
          "awslogs-stream-prefix": "ecs",
          "awslogs-create-group": "True"
        }
      }
    }
```

```
  ],
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "1024"
}
```

ステップ 4: Amazon Managed Service for Prometheus にアクセスするためのアクセス許可をタスクに付与する

スクレイピングされたメトリクスを Amazon Managed Service for Prometheus に送信するには、Amazon ECSタスクに オペレーションを AWS API呼び出すための正しいアクセス許可が必要です。タスクの IAMロールを作成し、`AmazonPrometheusRemoteWriteAccess`ポリシーをアタッチする必要があります。このロールの作成とポリシーのアタッチの詳細については、「タスク用のIAMロールとポリシーの作成」を参照してください。

IAM ロール`AmazonPrometheusRemoteWriteAccess`に をアタッチし、そのロールをタスクに使用すると、Amazon ECSはスクレイピングされたメトリクスを Amazon Managed Service for Prometheus に送信できます。

ステップ 5: Amazon Managed Grafana でメトリクスを視覚化する

> **Important**
>
> 開始する前に、Amazon タスク定義で Fargate ECSタスクを実行する必要があります。そうしないと、Amazon Managed Service for Prometheus でメトリクスを使用することができません。

1. Amazon Managed Grafana ワークスペースのナビゲーションペインで、 AWS アイコンの下にあるデータソースを選択します。
2. [データソース] タブの [サービス] で、[Amazon Managed Service for Prometheus] を選択し、[デフォルトのリージョン] を選択します。
3. [データソースを追加する] を選択する。
4. `ecs` および `prometheus` プレフィックスを使用して、メトリクスのクエリと表示を行います。

リモート書き込みを使用して Amazon EC2インスタンスからのメトリクスの取り込みを設定する

このセクションでは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスでリモート書き込みを使用して Prometheus サーバーを実行する方法について説明します。Go で記述されたデモアプリケーションからメトリクスを収集し、それらを Amazon Managed Service for Prometheus ワークスペースに送信する方法について説明します。

前提条件

> **Important**
>
> 開始する前に、Prometheus v2.26 以降をインストールしておく必要があります。Prometheus、Amazon 、Amazon Managed Service for Prometheus に精通していることを前提EC2としています。Prometheus のインストール方法については、Prometheus ウェブサイトの「Getting started」を参照してください。

Amazon EC2または Amazon Managed Service for Prometheus に慣れていない場合は、まず以下のセクションを読むことをお勧めします。

- Amazon Elastic Compute Cloud とは
- Amazon Managed Service for Prometheus とは

Amazon の IAMロールを作成する EC2

メトリクスをストリーミングするには、まず AWS マネージドポリシー を使用して IAMロールを作成する必要がありますAmazonPrometheusRemoteWriteAccess。その後、そのロールを持つインスタンスを起動し、Amazon Managed Service for Prometheus ワークスペースにメトリクスをストリーミングできます。

1. でIAMコンソールを開きますhttps://console.aws.amazon.com/iam/。
2. ナビゲーションペインで [ロール] を選択し、[ロールを作成] を選択します。
3. 信頼されたエンティティの種類として [AWS のサービス] を選択します。ユースケースに [EC2] を選択します。[Next: Permissions] (次へ: アクセス許可) を選択します。
4. [Search] バーに「AmazonPrometheusRemoteWriteAccess」と入力します。ポリシー名 で を選択しAmazonPrometheusRemoteWriteAccess、ポリシー をアタッチ を選択します。[Next:Tags] (次のステップ: タグ) を選択します。

5. （オプション) IAMロールのIAMタグを作成します。[次へ: レビュー] を選択します。
6. ロールの名前を入力します。[Create policy] を選択します。

Amazon EC2インスタンスを起動する

Amazon EC2インスタンスを起動するには、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスを起動する」の手順に従います。

デモアプリケーションの実行

IAM ロールを作成し、ロールを使用してEC2インスタンスを起動したら、デモアプリケーションを実行して動作を確認できます。

デモアプリケーションとテストメトリクスを実行するには

1. 以下のテンプレートを使用して、`main.go` という名前の Go ファイルを作成します。

```
package main

import (
    "github.com/prometheus/client_golang/prometheus/promhttp"
    "net/http"
)

func main() {
    http.Handle("/metrics", promhttp.Handler())

    http.ListenAndServe(":8000", nil)
}
```

2. 次のコマンドを実行して、適切な依存関係をインストールします。

```
sudo yum update -y
sudo yum install -y golang
go get github.com/prometheus/client_golang/prometheus/promhttp
```

3. デモアプリケーションを実行します。

```
go run main.go
```

デモアプリケーションはポート 8000 で実行され、公開されているすべての Prometheus メトリクスを表示します。これらのメトリクスの例を以下に示します。

```
curl -s http://localhost:8000/metrics
...
process_max_fds 4096# HELP process_open_fds Number of open file descriptors.# TYPE
 process_open_fds gauge
process_open_fds 10# HELP process_resident_memory_bytes Resident memory size in
 bytes.# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 1.0657792e+07# HELP process_start_time_seconds Start
 time of the process since unix epoch in seconds.# TYPE process_start_time_seconds
 gauge
process_start_time_seconds 1.61131955899e+09# HELP process_virtual_memory_bytes
 Virtual memory size in bytes.# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 7.77281536e+08# HELP process_virtual_memory_max_bytes
 Maximum amount of virtual memory available in bytes.# TYPE
 process_virtual_memory_max_bytes gauge
process_virtual_memory_max_bytes -1# HELP
 promhttp_metric_handler_requests_in_flight Current number of scrapes being
 served.# TYPE promhttp_metric_handler_requests_in_flight gauge
promhttp_metric_handler_requests_in_flight 1# HELP
 promhttp_metric_handler_requests_total Total number of scrapes by HTTP status
 code.# TYPE promhttp_metric_handler_requests_total counter
promhttp_metric_handler_requests_total{code="200"} 1
promhttp_metric_handler_requests_total{code="500"} 0
promhttp_metric_handler_requests_total{code="503"} 0
```

Amazon Managed Service for Prometheus ワークスペースの作成

Amazon Managed Service for Prometheus ワークスペースを作成するには、「Create a workspace」の手順に従います。

Prometheus サーバーの実行

1. 次のサンプルYAMLファイルをテンプレートとして使用して、 という名前の新しいファイルを作成しますprometheus.yaml。の場合はurl、 を置き換えます。*my-region* リージョン値と *my-workspace-id* Amazon Managed Service for Prometheus が生成したワークスペース ID を入力します。の場合はregion、 を置き換えます。*my-region* リージョンの値を指定します。

   例: YAML ファイル

```
global:
  scrape_interval: 15s
  external_labels:
    monitor: 'prometheus'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:8000']

remote_write:
  -
    url: https://aps-workspaces.my-region.amazonaws.com/workspaces/my-workspace-id/
api/v1/remote_write
    queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
    sigv4:
         region: my-region
```

2. Prometheus サーバーを実行して、デモアプリケーションのメトリクスを Amazon Managed Service for Prometheus ワークスペースに送信します。

```
prometheus --config.file=prometheus.yaml
```

これで、Prometheus サーバーによってデモアプリケーションのメトリクスが Amazon Managed Service for Prometheus ワークスペースに送信されます。

Prometheus インスタンスをコレクターとして使用する

エージェントモード (Prometheus エージェント と呼ばれる) で実行されている Prometheus インスタンスを使用して、メトリクスをスクレイプし、Amazon Managed Service for Prometheus ワークスペースに送信できます。

以下のトピックでは、エージェントモードで実行されている Prometheus インスタンスをメトリクスのコレクターとして設定するさまざまな方法について説明します。

> **Warning**
>
> Prometheus エージェントを作成する場合、その設定とメンテナンスはお客様の責任となります。セキュリティ機能 を有効にすることで、Prometheus スクレイプエンドポイントをパブリックインターネットに公開しないようにします。

同じメトリクスセットをモニタリングする複数の Prometheus インスタンスをセットアップし、それらを 1 つの Amazon Managed Service for Prometheus ワークスペースに送信して高可用性を実現する場合は、重複排除を設定する必要があります。重複排除を設定する手順に従わない場合、Amazon Managed Service for Prometheus に送信されるすべてのデータサンプルが、重複サンプルも含めて課金対象になります。重複排除の設定方法については、「Amazon Managed Service for Prometheus に送信される高可用性メトリクスの重複排除」を参照してください。

トピック

- Helm を使用した新しい Prometheus サーバーからの取り込みの設定
- で Kubernetes の既存の Prometheus サーバーからの取り込みを設定する EC2
- Fargate 上の Kubernetes にある既存の Prometheus サーバーからの取り込みの設定

Helm を使用した新しい Prometheus サーバーからの取り込みの設定

このセクションの手順に従うと、Amazon Managed Service for Prometheus を迅速に設定して稼働させることができます。Amazon EKSクラスターに新しい Prometheus サーバーをセットアップすると、新しいサーバーはデフォルト設定を使用して Amazon Managed Service for Prometheus にメトリクスを送信します。この方法には次の前提条件があります。

- 新しい Prometheus サーバーがメトリクスを収集する Amazon EKSクラスターが必要です。
- Amazon EKSクラスターには、Amazon EBSCSIドライバーがインストールされている必要があります (Helm で必要) 。
- Helm 3.0 CLI 以降を使用する必要があります。
- 以下のセクションの手順を実行するには、Linux または macOS コンピュータを使用する必要があります。

ステップ 1: 新しい Helm チャートリポジトリを追加する

次のコマンドを入力して、新しい Helm チャートリポジトリを追加します。これらのコマンドの詳細については、「Helm Repo」を参照してください。

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
helm repo add kube-state-metrics https://kubernetes.github.io/kube-state-metrics
helm repo update
```

ステップ 2: Prometheus 名前空間を作成する

次のコマンドを入力して、Prometheus サーバーとその他のモニタリングコンポーネント用の Prometheus 名前空間を作成します。置換 *prometheus-namespace* この名前空間に必要な名前の 。

```
kubectl create namespace prometheus-namespace
```

ステップ 3: サービスアカウントのIAMロールを設定する

ドキュメント化しているオンボーディング方法については、Prometheus サーバーが実行されている Amazon EKSクラスターのサービスアカウントにIAMロールを使用する必要があります。

サービスアカウントのIAMロールを使用すると、 IAMロールを Kubernetes サービスアカウントに関連付けることができます。このサービスアカウントは、そのサービスアカウントを使用するポッド内のコンテナに AWS アクセス許可を提供できます。詳細については、IAM「 サービスアカウントのロール」を参照してください。

これらのロールをまだ設定していない場合は、「Amazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定」の手順に従ってロールを設定します。そのセクションの手順では、`eksctl` を使用する必要があります。詳細については、「Amazon Elastic Kubernetes Service の開始方法 - `eksctl`」を参照してください。

> **Note**
>
> EKS または ではなく AWS 、アクセスキーとシークレットキーのみを使用して Amazon Managed Service for Prometheus にアクセスする場合、 `EKS-IAM-ROLE` ベースの SigV4 を使用することはできません。

ステップ 4: 新しいサーバーをセットアップしてメトリクスの取り込みを開始する

Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信する新しい Prometheus サーバーをインストールするには、以下の手順に従います。

Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信する新しい Prometheus サーバーをインストールするには

1. テキストエディタを使用して、`my_prometheus_values_yaml` という名前のファイルを作成し、次の内容を記述します。

   - 置換 *IAM_PROXY_PROMETHEUS_ROLE_ARN* でamp-iamproxy-ingest-role作成した ARN の を使用しますAmazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定。
   - 置換 *WORKSPACE_ID* を Amazon Managed Service for Prometheus ワークスペースの ID で指定します。
   - 置換 *REGION* Amazon Managed Service for Prometheus ワークスペースの リージョン。

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
## For the rest of prometheus helm chart values see: https://github.com/prometheus-
community/helm-charts/blob/main/charts/prometheus/values.yaml
##
serviceAccounts:
  server:
    name: amp-iamproxy-ingest-service-account
    annotations:
      eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
server:
  remoteWrite:
    - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
      sigv4:
        region: ${REGION}
      queue_config:
        max_samples_per_send: 1000
        max_shards: 200
        capacity: 2500
```

2. 次のコマンドを入力して、Prometheus サーバーを作成します。

- 置換 *prometheus-chart-name* Prometheus リリース名を入力します。
- 置換 *prometheus-namespace* Prometheus 名前空間の名前を入力します。

```
helm install prometheus-chart-name prometheus-community/prometheus -n prometheus-
namespace \
-f my_prometheus_values_yaml
```

> **Note**
>
> `helm install` コマンドはさまざまな方法でカスタマイズできます。詳細については、「Helm ドキュメント」の「Helm install」を参照してください。

で Kubernetes の既存の Prometheus サーバーからの取り込みを設定する EC2

Amazon Managed Service for Prometheus は、Amazon を実行しているクラスターEKSと Amazon で実行されているセルフマネージド Kubernetes クラスター内の Prometheus サーバーからのメトリクスの取り込みをサポートしますEC2。このセクションの詳細な手順は、Amazon EKSクラスター内の Prometheus サーバー用です。Amazon のセルフマネージド Kubernetes クラスターの手順は同じEC2ですが、Kubernetes クラスターでサービスアカウントのOIDCプロバイダーとIAMロールを自分で設定する必要がある点が異なります。

このセクションの手順では、Kubernetes パッケージマネージャーとして Helm を使用します。

トピック

- ステップ 1: サービスアカウントのIAMロールを設定する
- ステップ 2: Helm を使用して既存の Prometheus サーバーをアップグレードする

ステップ 1: サービスアカウントのIAMロールを設定する

ドキュメント化しているオンボーディング方法については、Prometheus サーバーが実行されているAmazon EKSクラスターのサービスアカウントにIAMロールを使用する必要があります。これらのロールはサービスロールとも呼ばれます。

サービスロールを使用すると、 IAMロールを Kubernetes サービスアカウントに関連付けることができます。このサービスアカウントは、そのサービスアカウントを使用する任意のポッドのコンテナに

アクセス AWS 許可を付与できます。詳細については、IAM「 サービスアカウントの ロール」を参照してください。

これらのロールをまだ設定していない場合は、「Amazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定」の手順に従ってロールを設定します。

ステップ 2: Helm を使用して既存の Prometheus サーバーをアップグレードする

このセクションの手順には、リモート書き込みと sigv4 を設定して、Amazon Managed Service for Prometheus ワークスペースへのリモート書き込みを行えるように Prometheus サーバーを認証および認可する方法が含まれます。

Prometheus バージョン 2.26.0 以降を使用している場合

バージョン 2.26.0 以降の Prometheus サーバーイメージで Helm チャートを使用している場合は、以下の手順に従います。

Helm チャートを使用して Prometheus サーバーからのリモート書き込みを設定するには

1. Helm 設定ファイルに新しいリモート書き込みセクションを作成します。

   - を、 でamp-iamproxy-ingest-role作成した ARNの ${IAM_PROXY_PROMETHEUS_ROLE_ARN}に置き換えますステップ 1: サービスアカウントのIAMロールを設定する。ロールの形式は ARNである必要がありますarn:aws:iam::*your account ID*:role/amp-iamproxy-ingest-role。
   - ${WORKSPACE_ID} は、Amazon Managed Service for Prometheus のワークスペース ID に置き換えます。
   - ${REGION} は、Amazon Managed Service for Prometheus ワークスペースのリージョン (us-west-2 など) に置き換えます。

```
## The following is a set of default values for prometheus server helm chart which
 enable remoteWrite to AMP
    ## For the rest of prometheus helm chart values see: https://github.com/
prometheus-community/helm-charts/blob/main/charts/prometheus/values.yaml
    ##
    serviceAccounts:
      server:
        name: amp-iamproxy-ingest-service-account
        annotations:
          eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}
```

```
    server:
      remoteWrite:
        - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
          sigv4:
            region: ${REGION}
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

2. Helm を使用して既存の Prometheus サーバーの構成を更新します。

   - `prometheus-chart-name` は、Prometheus リリース名に置き換えます。
   - `prometheus-namespace` は、Prometheus サーバーがインストールされている Kubernetes 名前空間に置き換えます。
   - `my_prometheus_values_yaml` は、Helm 設定ファイルのパスに置き換えます。
   - `current_helm_chart_version` は、Prometheus サーバーの Helm チャートの現在のバージョンに置き換えます。現在のチャートのバージョンは、helm list コマンドを使用して確認できます。

```
helm upgrade prometheus-chart-name prometheus-community/prometheus \
       -n prometheus-namespace \
       -f my_prometheus_values_yaml \
       --version current_helm_chart_version
```

以前のバージョンの Prometheus 使う

2.26.0 より前のバージョンの Prometheus を使用している場合は、以下の手順に従います。これらのステップではサイドカーアプローチを使用します。これは、以前のバージョンの Prometheus では AWS 署名バージョン 4 の署名プロセス (AWS SigV4) がネイティブにサポートされていないためです。

これらの手順では、Prometheus のデプロイに Helm を使用しているものと想定します。

Prometheus サーバーからのリモート書き込みを設定するには

1. Prometheus サーバーで、新しいリモート書き込み設定を作成します。まず、新しい更新ファイルを作成します。このファイルの名前を `amp_ingest_override_values.yaml`. とします。

次の値を YAML ファイルに追加します。

```
serviceAccounts:
        server:
            name: "amp-iamproxy-ingest-service-account"
            annotations:
                eks.amazonaws.com/role-arn:
 "${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}"
    server:
        sidecarContainers:
            - name: aws-sigv4-proxy-sidecar
              image: public.ecr.aws/aws-observability/aws-sigv4-proxy:1.0
              args:
              - --name
              - aps
              - --region
              - ${REGION}
              - --host
              - aps-workspaces.${REGION}.amazonaws.com
              - --port
              - :8005
              ports:
              - name: aws-sigv4-proxy
                containerPort: 8005
        statefulSet:
            enabled: "true"
        remoteWrite:
            - url: http://localhost:8005/workspaces/${WORKSPACE_ID}/api/v1/
remote_write
```

${REGION} は、Amazon Managed Service for Prometheus ワークスペースのリージョンに置き換えます。

を、 でamp-iamproxy-ingest-role作成した ARNの ${SERVICE_ACCOUNT_IAM_INGEST_ROLE_ARN}に置き換えますステップ 1: サービスアカウントのIAMロールを設定する。ロールの形式は ARNである必要がありますarn:aws:iam::*your account ID*:role/amp-iamproxy-ingest-role。

${WORKSPACE_ID} は、ワークスペース ID に置き換えます。

2. Prometheus Helm チャートをアップグレードします。まず、以下のコマンドを入力して Helm チャート名を確認します。このコマンドの出力で、名前に prometheus という文字列を含むチャートを探します。

```
helm ls --all-namespaces
```

次に、以下のコマンドを入力します。

```
helm upgrade --install prometheus-helm-chart-name prometheus-community/prometheus -
n prometheus-namespace -f ./amp_ingest_override_values.yaml
```

置換 *prometheus-helm-chart-name* を、前のコマンドで返された Prometheus helm チャートの名前に置き換えます。置換 *prometheus-namespace* を名前空間の名前に置き換えます。

Helm チャートのダウンロード

Helm チャートをまだローカルにダウンロードしていない場合は、次のコマンドを使用してダウンロードできます。

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
    helm pull prometheus-community/prometheus --untar
```

Fargate 上の Kubernetes にある既存の Prometheus サーバーからの取り込みの設定

Amazon Managed Service for Prometheus は、Fargate 上で動作する自己管理型 Kubernetes クラスター内の Prometheus サーバーからのメトリクスの取り込みをサポートしています。Fargate で実行されている Amazon EKSクラスターの Prometheus サーバーからメトリクスを取り込むには、amp_ingest_override_values.yaml という名前の設定ファイルのデフォルトの設定を次のように上書きします。

```
prometheus-node-exporter:
        enabled: false

    alertmanager:
        enabled: false

    serviceAccounts:
      server:
        name: amp-iamproxy-ingest-service-account
```

```
        annotations:
          eks.amazonaws.com/role-arn: ${IAM_PROXY_PROMETHEUS_ROLE_ARN}

    server:
      persistentVolume:
        enabled: false
      remoteWrite:
        - url: https://aps-workspaces.${REGION}.amazonaws.com/workspaces/
${WORKSPACE_ID}/api/v1/remote_write
          sigv4:
            region: ${REGION}
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

次のコマンドを実行して、オーバーライドを使用して Prometheus をインストールします。

```
helm install prometheus-for-amp prometheus-community/prometheus \
                  -n prometheus \
                  -f amp_ingest_override_values.yaml
```

この Helm チャートの設定では、ノードエクスポーターとアラートマネージャーを無効にし、さらに Prometheus サーバーのデプロイの実行を無効にしています。

次のテストクエリの例を実行すると、インストールを確認できます。

```
$ awscurl --region region --service aps "https://aps-
workspaces.region_id.amazonaws.com/workspaces/workspace_id/api/v1/query?
query=prometheus_api_remote_read_queries"
    {"status":"success","data":{"resultType":"vector","result":[{"metric":
{"__name__":"prometheus_api_remote_read_queries","instance":"localhost:9090","job":"prometheus"
[1648461236.419,"0"]}]}}21
```

高可用性データ用に Amazon Managed Service for Prometheus をセットアップする

Amazon Managed Service for Prometheus にデータを送信すると、そのデータはリージョン内の AWS アベイラビリティーゾーン間で自動的にレプリケートされ、スケーラビリティ、可用性、セキュリティを提供するホストのクラスターから提供されます。特定の環境によっては、さらに可用性

を高めるフェイルセーフ機能を追加することが望ましい場合があります。環境に高可用性セーフティを追加する一般的な方法は 2 つあります。

- 同じデータを持つ複数のコンテナまたはインスタンスがある場合は、そのデータを Amazon Managed Service for Prometheus に送信し、データの重複排除を自動的に行わせることができます。これは、データを Amazon Managed Service for Prometheus ワークスペースに確実に送信するために役立ちます。

  高可用性データの重複排除の詳細については、「Amazon Managed Service for Prometheus に送信される高可用性メトリクスの重複排除」を参照してください。
- AWS リージョンが利用できない場合でもデータにアクセスできるようにする場合は、別のリージョンの 2 つ目のワークスペースにメトリクスを送信できます。

  メトリクスデータを複数のワークスペースに送信する方法の詳細については、「クロスリージョンワークスペースを使用して Amazon Managed Service for Prometheus で高可用性を追加する」を参照してください。

トピック

- Amazon Managed Service for Prometheus に送信される高可用性メトリクスの重複排除
- Prometheus による Amazon Managed Service for Prometheus への高可用性データの送信
- Prometheus Operator Helm チャートを使用して Amazon Managed Service for Prometheus への高可用性データをセットアップする
- Distro for を使用して Amazon Managed Service for Prometheus AWS に高可用性データを送信する OpenTelemetry
- Prometheus コミュニティ Helm チャートによる Amazon Managed Service for Prometheus への高可用性データの送信
- Amazon Managed Service for Prometheus での高可用性設定に関する一般的な質問への回答
- クロスリージョンワークスペースを使用して Amazon Managed Service for Prometheus で高可用性を追加する

Amazon Managed Service for Prometheus に送信される高可用性メトリクスの重複排除

複数の Prometheus エージェント (エージェントモードで実行されている Prometheus インスタンス) から、Amazon Managed Service for Prometheus ワークスペースにデータを送信できます。

これらのインスタンスのいくつかが同じメトリクスを記録して送信している場合、データの高可用性が確保されます (いずれかのエージェントがデータの送信を停止しても、Amazon Managed Service for Prometheus ワークスペースは別のインスタンスから引き続きデータを受信します)。ただし、Amazon Managed Service for Prometheus ワークスペースでは、メトリクスの重複が自動的に排除されるようにすることが望まれます。これにより、メトリクスが複数回表示されるのを防ぎ、データインジェストとストレージに対して複数回課金が発生することを回避できます。

Amazon Managed Service for Prometheus で複数の Prometheus エージェントからのデータを自動的に重複排除するには、重複データを送信しているエージェントのセットに単一のクラスター名を割り当て、各インスタンスにレプリカ名を割り当てます。クラスター名により、これらのインスタンスが共有データを持つものとして識別されます。レプリカ名により、Amazon Managed Service for Prometheus で各メトリクスのソースを識別することが可能になります。最終的に保存されるメトリクスにはクラスターラベルが含まれますが、レプリカは含まれないため、メトリクスは単一のソースから取得されているように見えます。

> **Note**
>
> Kubernetes の特定のバージョン (1.28 および 1.29) では、`cluster`ラベル付きの独自のメトリクスが出力される場合があります。これにより、Amazon Managed Service for Prometheus の重複排除に関する問題が発生する可能性があります。詳細については、高可用性FAQを参照してください。

以下のトピックでは、Amazon Managed Service for Prometheus がデータを自動的に重複解除できるように、データを送信し、 `cluster`および `__replica__`ラベルを含める方法を示します。

> **Important**
>
> 重複排除を設定しない場合、Amazon Managed Service for Prometheus に送信されるすべてのデータサンプルが課金対象になります。これらのデータサンプルには、重複するサンプルが含まれます。

Prometheus による Amazon Managed Service for Prometheus への高可用性データの送信

Prometheus で高可用性設定をセットアップするには、高可用性グループのすべてのインスタンスに外部ラベルを適用して、Amazon Managed Service for Prometheus でそれらを識別できるよ

うにする必要があります。Prometheus インスタンスのエージェントを高可用性グループの一部として識別するには、cluster ラベルを使用します。グループ内の各レプリカを個別に識別するには、__replica__ ラベルを使用します。重複排除を機能させるには、__replica__ と cluster の両方のラベルを適用する必要があります。

> **Note**
>
> __replica__ ラベルは、replica という単語の前後に 2 つのアンダースコア記号が付いた形式です。

例: コードスニペット

次のコードスニペットでは、cluster ラベルは Prometheus インスタンスのエージェント prom-team1 を識別し、_replica_ ラベルはレプリカ replica1 と replica2 を識別します。

```
cluster: prom-team1
__replica__: replica1
```

```
cluster: prom-team1
__replica__: replica2
```

Amazon Managed Service for Prometheus は、これらのラベルを持つ高可用性レプリカからのデータサンプルを保存する場合、サンプルを受け入れるときに replica ラベルを取り除きます。つまり、レプリカごとにシリーズが保存されるのではなく、現在のシリーズに対して 1:1 のシリーズマッピングが作成されます。cluster ラベルは保持されます。

> **Note**
>
> Kubernetes の特定のバージョン (1.28 および 1.29) では、clusterラベル付きの独自のメトリクスが出力される場合があります。これにより、Amazon Managed Service for Prometheus の重複排除に関する問題が発生する可能性があります。詳細については、「高可用性FAQ」を参照してください。

Prometheus Operator Helm チャートを使用して Amazon Managed Service for Prometheus への高可用性データをセットアップする

Helm で Prometheus Operator を使用して高可用性設定をセットアップするには、Amazon Managed Service for Prometheus が識別できるように、高可用性グループのすべてのインスタンスに外部ラベルを適用する必要があります。さらに、Prometheus Operator Helm チャートで `replicaExternalLabelName` および `externalLabels` 属性を設定する必要もあります。

例: YAMLヘッダー

次のYAMLヘッダーでは、 `cluster`が `externalLabel`に追加され、高可用性グループの一部として Prometheus インスタンスエージェントを識別し、グループ内の各レプリカ`replicaExternalLabels`を識別します。

```
replicaExternalLabelName: __replica__
externalLabels:
cluster: prom-dev
```

> **Note**
>
> Kubernetes の特定のバージョン (1.28 および 1.29) では、`cluster`ラベル付きの独自のメトリクスが出力される場合があります。これにより、Amazon Managed Service for Prometheus の重複排除に関する問題が発生する可能性があります。詳細については、「高可用性FAQ」を参照してください。

Distro for を使用して Amazon Managed Service for Prometheus AWS に高可用性データを送信する OpenTelemetry

AWS Distro for OpenTelemetry （ ADOT) は、 OpenTelemetry プロジェクトの安全で本番環境に対応したディストリビューションです。ADOT にはソース APIs、ライブラリ、エージェントが用意されているため、アプリケーションモニタリング用の分散トレースとメトリクスを収集できます。の詳細についてはADOT、AWS 「 Distro for Open Telemetry について」を参照してください。

高可用性設定ADOTで を設定するには、ADOTコレクターコンテナイメージを設定し、外部ラベル `cluster`と `__replica__`を AWS Prometheus リモート書き込みエクスポーターに適用する必要があります。このエクスポーターは、スクレイピングされたメトリクスを `remote_write` エンドポイント経由で Amazon Managed Service for Prometheus ワークスペースに送信します。これらのラ

ベルを Remote Write Exporter に設定すると、冗長レプリカの実行中に重複するメトリクスが保持されるのを防ぐことができます。 AWS Prometheus リモート書き込みエクスポーターの詳細については、「Amazon Managed Service for Prometheus の Prometheus リモート書き込みエクスポーターの開始方法」を参照してください。

> **Note**
>
> Kubernetes の特定のバージョン (1.28 および 1.29) では、`cluster`ラベル付きの独自のメトリクスが出力される場合があります。これにより、Amazon Managed Service for Prometheus の重複排除に関する問題が発生する可能性があります。詳細については、「高可用性FAQ」を参照してください。

Prometheus コミュニティ Helm チャートによる Amazon Managed Service for Prometheus への高可用性データの送信

Prometheus コミュニティ Helm チャートで高可用性設定をセットアップするには、高可用性グループのすべてのインスタンスに外部ラベルを適用して、Amazon Managed Service for Prometheus でそれらを識別できるようにする必要があります。以下は、Prometheus コミュニティ Helm チャートから Prometheus の 1 つのインスタンスに `external_labels` を追加する方法の例を示しています。

```
server:
global:
  external_labels:
      cluster: monitoring-cluster
      __replica__: replica-1
```

> **Note**
>
> 複数のレプリカが必要な場合は、異なるレプリカ値を使用してチャートを複数回デプロイする必要があります。Prometheus コミュニティ Helm チャートでは、コントローラーグループから直接レプリカの数を増やすとき、レプリカ値を動的に設定することができないためです。`replica` ラベルを自動設定するには、Prometheus Operator Helm チャートを使用します。

> **Note**
>
> Kubernetes の特定のバージョン (1.28 および 1.29) では、clusterラベル付きの独自のメトリクスが出力される場合があります。これにより、Amazon Managed Service for Prometheus の重複排除に関する問題が発生する可能性があります。詳細については、「高可用性FAQ」を参照してください。

Amazon Managed Service for Prometheus での高可用性設定に関する一般的な質問への回答

サンプルポイントを追跡するには、値 __replica__ を別のラベルに含める必要がありますか？

高可用性設定では、Amazon Managed Service for Prometheus は Prometheus インスタンスのクラスターからリーダーを選出することで、データサンプルが重複しないようにします。リーダーレプリカからのデータサンプルの送信が 30 秒間停止した場合、Amazon Managed Service for Prometheus は自動的に別の Prometheus インスタンスをリーダーレプリカに設定し、欠落したデータを含めてデータを新しいリーダーから取り込みます。したがって、答えは「いいえ」であり、推奨もされません。 これを行った場合、次のような問題が発生する可能性があります。

- PromQL で count のクエリを実行すると、新しいリーダーの選出期間中に、想定よりも高い値が返されることがあります。
- 新しいリーダーの選出期間中にそのリーダーが active series limits になると、active series の数が増加します。詳細については、AMP「クォータ」を参照してください。

Kubernetes には独自のクラスターラベルがあり、メトリクスの重複排除は行われていないようです。どうすればこの問題を解決できますか。

Kubernetes 1.28 では、clusterラベル付きの新しいメトリクスが導入されapiserver_storage_size_bytesました。これにより、clusterラベルに依存する Amazon Managed Service for Prometheus で重複排除の問題が発生する可能性があります。Kubernetes 1.3 では、ラベルの名前は に変更されます storage-cluster_id (1.28 および 1.29 の以降のパッチでも名前が変更されます)。クラスターがこのメトリクスを clusterラベルで出力している場合、Amazon Managed Service for Prometheus は関連する時系列を重複排除できません。この問題を回避するために、Kubernetes クラスターをパッチが適用された最新のバージョンにアップグレードすることをお勧めします。または、Amazon Managed Service for Prometheus に取り込む前

にapiserver_storage_size_bytes、メトリクスのclusterラベルを再ラベル付けすることもできます。

> **Note**
>
> Kubernetes への変更の詳細については、Kubernetes プロジェクト の「rename Label cluster to storage_cluster_id for apiserver_storage_size_bytes metric」を参照してください。GitHub

クロスリージョンワークスペースを使用して Amazon Managed Service for Prometheus で高可用性を追加する

クロスリージョン可用性をデータに追加するには、 AWS リージョン間で複数のワークスペースにメトリクスを送信できます。Prometheus では、複数のライターとクロスリージョンでの書き込みの両方がサポートされています。

以下の例は、Helm を使用してエージェントモードで動作する Prometheus サーバーをセットアップして、異なるリージョンの 2 つのワークスペースにメトリクスを送信する方法を示しています。

```
extensions:
      sigv4auth:
        service: "aps"

    receivers:
      prometheus:
        config:
          scrape_configs:
            - job_name: 'kubernetes-kubelet'
              scheme: https
              tls_config:
                ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
                insecure_skip_verify: true
              bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
              kubernetes_sd_configs:
              - role: node
              relabel_configs:
              - action: labelmap
                regex: __meta_kubernetes_node_label_(.+)
              - target_label: __address__
                replacement: kubernetes.default.svc.cluster.local:443
```

```
              - source_labels: [__meta_kubernetes_node_name]
                regex: (.+)
                target_label: __metrics_path__
                replacement: /api/v1/nodes/$${1}/proxy/metrics

    exporters:
      prometheusremotewrite/one:
        endpoint: "https://aps-workspaces.workspace_1_region.amazonaws.com/workspaces/
ws-workspace_1_id/api/v1/remote_write"
        auth:
          authenticator: sigv4auth
      prometheusremotewrite/two:
        endpoint: "https://aps-workspaces.workspace_2_region.amazonaws.com/workspaces/
ws-workspace_2_id/api/v1/remote_write"
        auth:
          authenticator: sigv4auth

    service:
      extensions: [sigv4auth]
      pipelines:
        metrics/one:
          receivers: [prometheus]
          exporters: [prometheusremotewrite/one]
        metrics/two:
          receivers: [prometheus]
          exporters: [prometheusremotewrite/two]
```

Prometheus メトリクスに対するクエリの実行

ワークスペースにメトリクスが取り込まれるようになったら、それらのメトリクスに対してクエリを実行できます。

メトリクスを視覚的に表現したダッシュボードを作成するには、Amazon Managed Grafana などのサービスを使用できます。Amazon Managed Grafana (または Grafana のスタンドアロンインスタンス) は、さまざまなディスプレイプレゼンテーションスタイルでメトリクスを表示するグラフィカルインターフェイスを構築できます。Amazon Managed Grafana の詳細については、「Amazon Managed Grafana ユーザーガイド」を参照してください。

直接クエリを使用して、1 回限りのクエリの作成、データの探索、メトリクスを使用する独自のアプリケーションの記述を行うこともできます。ダイレクトクエリは、Amazon Managed Service for Prometheus APIと標準の Prometheus クエリ言語である PromQL を使用して、Prometheus ワークスペースからデータを取得します。PromQL とその構文の詳細については、Prometheus ドキュメントの「Querying Prometheus」を参照してください。

トピック

メトリクスクエリを保護する

Amazon Managed Service for Prometheus には、メトリクスのクエリの実行を保護するための手段が用意されています。

Amazon Managed Service for Prometheus AWS PrivateLink での の使用

Amazon Managed Service for Prometheus でメトリクスをクエリするためのネットワークトラフィックは、パブリックインターネットエンドポイントを介して、または を介してVPCエンドポ

イントによって実行できます AWS PrivateLink。を使用すると AWS PrivateLink、 からのネットワークトラフィックVPCsは、パブリックインターネットを経由せずに AWS ネットワーク内で保護されます。Amazon Managed Service for Prometheus のエンドポイントを作成するには AWS PrivateLink VPC、「」を参照してくださいインターフェイス VPC エンドポイントでの Amazon Managed Service for Prometheus の使用。

認証と認可

AWS Identity and Access Management は、 リソースへのアクセス AWS を安全に制御するのに役立つウェブサービスです。を使用してIAM、誰を認証 (サインイン) し、誰に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。Amazon Managed Service for Prometheus は と統合されIAM、データの安全性の維持に役立ちます。Amazon Managed Service for Prometheus を設定するときは、Grafana サーバーが Amazon Managed Service for Prometheus ワークスペースに保存されているメトリクスをクエリできるようにするIAMロールをいくつか作成する必要があります。の詳細についてはIAM、「 IAMとは」を参照してください。

Amazon Managed Service for Prometheus のセットアップに役立つもう 1 つの AWS セキュリティ機能は、 AWS 署名バージョン 4 の署名プロセス (AWS SigV4) です。署名バージョン 4 は、 によって送信された AWS リクエストに認証情報を追加するプロセスですHTTP。セキュリティ上の理由から、 へのほとんどのリクエストは、アクセスキー ID とシークレットアクセスキーで構成されるアクセスキーで署名 AWS する必要があります。これらの 2 つのキーは、一般的にセキュリティ認証情報と呼ばれます。SigV4 の詳細については、「Signature Version 4 の署名プロセス」を参照してください。

Amazon Managed Service for Prometheus で使用するための Amazon Managed Grafana のセットアップ

Amazon Managed Grafana は、オープンソースの Grafana 向けのフルマネージドサービスで、オープンソース、サードパーティーの 、および AWS サービスへの接続を簡素化しISV、データソースを大規模に視覚化および分析します。

Amazon Managed Service for Prometheus では、Amazon Managed Grafana を使用してワークスペース内のメトリクスにクエリを実行することがサポートされています。Amazon Managed Grafana コンソールで、既存の Amazon Managed Service for Prometheus アカウントを検出して、Amazon Managed Service for Prometheus ワークスペースをデータソースとして追加できます。Amazon Managed Grafana は、Amazon Managed Service for Prometheus にアクセスするために必要な認証情報の設定を管理します。Amazon Managed Grafana から Amazon Managed Service

for Prometheus への接続を作成する方法の詳細については、「Amazon Managed Grafana User Guide」の手順を参照してください。

Amazon Managed Service for Prometheus のアラートを Amazon Managed Grafana で表示することもできます。アラートとの統合を設定する手順については、「アラートを Amazon Managed Grafana またはオープンソース Grafana と統合する」を参照してください。

プライベートで Amazon Managed Grafana に接続する VPC

Amazon Managed Service for Prometheus は、Amazon Managed Grafana がメトリクスやアラートのクエリを実行するときに接続するサービスエンドポイントを提供しています。

プライベートを使用するように Amazon Managed Grafana を設定できます VPC (Grafana VPC でのプライベートの設定の詳細については、「Amazon Managed Grafana ユーザーガイドVPC」の「Amazon への接続」を参照してください) 。設定によっては、Amazon Managed Service for Prometheus サービスエンドポイントにアクセスできないVPC場合があります。

特定のプライベート を使用するように設定された Amazon Managed Grafana ワークスペースにデータソースとして Amazon Managed Service for Prometheus を追加するにはVPC、まずVPCエンドポイントVPCを作成して Amazon Managed Service for Prometheus を同じ に接続する必要があります。VPC エンドポイントの作成の詳細については、「」を参照してくださいAmazon Managed Service for Prometheus 用のインターフェイス VPC エンドポイントの作成。

Amazon Managed Service for Prometheus で使用する Grafana オープンソースまたは Grafana Enterprise のセットアップ

Grafana のインスタンスを使用して、Amazon Managed Service for Prometheus でメトリクスをクエリできます。このトピックでは、Grafana のスタンドアロンインスタンスを使用して Amazon Managed Service for Prometheus からメトリクスをクエリする方法について説明します。

前提条件

Grafana インスタンス – Amazon Managed Service for Prometheus で認証できる Grafana インスタンスが必要です。

Amazon Managed Service for Prometheus では、Grafana バージョン 7.3.5 以降を使用してワークスペース内のメトリクスにクエリを実行することがサポートされています。バージョン 7.3.5 以降には、 AWS 署名バージョン 4 (SigV4) 認証のサポートが含まれています。

Grafana のバージョンを確認するには、次のコマンドを入力します。*grafana_install_directory* Grafana のインストールへのパス：

```
grafana_install_directory/bin/grafana-server -v
```

スタンドアロンの Grafana をまだお持ちでない場合、または新しいバージョンが必要な場合は、新しいインスタンスをインストールできます。スタンドアロンの Grafana をセットアップする手順については、Grafana ドキュメントの「Grafana のインストール」を参照してください。Grafana の開始方法については、Grafana ドキュメントの「Grafana の開始方法」を参照してください。

AWS アカウント – Amazon Managed Service for Prometheus メトリクスにアクセスするには、適切なアクセス許可 AWS アカウント を持つ が必要です。

Amazon Managed Service for Prometheus を使用するように Grafana を設定するには、AmazonPrometheusQueryAccessポリシーまたは `aps:QueryMetrics`、、`aps:GetMetricMetadata`、`aps:GetSeries`および アクセス`aps:GetLabels`許可を持つアカウントにログオンする必要があります。詳細については、「IAM のアクセス許可とポリシー」を参照してください。

次のセクションでは、Grafana からの認証の設定について詳しく説明します。

ステップ 1: SigV4 をセットアップ AWS する

Amazon Managed Service for Prometheus は AWS Identity and Access Management （ IAM) と連携して、Prometheus へのすべての呼び出しをIAM認証情報APIsで保護します。デフォルトでは、Grafana の Prometheus データソースは、Prometheus が認証を必要としないものと想定します。Grafana で Amazon Managed Service for Prometheus の認証および認可機能を利用できるようにするには、Grafana データソースで SigV4 認証サポートを有効にする必要があります。自己管理型の Grafana オープンソースサーバーまたは Grafana Enterprise サーバーを使用している場合は、このページの手順に従ってください。Amazon Managed Grafana を使用している場合、SIGv4 認証は完全に自動化されます。Amazon Managed Grafana の詳細については、「Amazon Managed Grafana とは」を参照してください。

Grafana で SigV4 を有効にするには、`AWS_SDK_LOAD_CONFIG` および `GF_AUTH_SIGV4_AUTH_ENABLED` 環境変数を `true` に設定して Grafana を起動します。`GF_AUTH_SIGV4_AUTH_ENABLED` 環境変数は、Grafana のデフォルト設定をオーバーライドして SigV4 サポートを有効にします。詳細については、Grafana ドキュメントの「Configuration」を参照してください。

Linux

Linux 上のスタンドアロン Grafana サーバーで SigV4 を有効にするには、次のコマンドを入力します。

```
export AWS_SDK_LOAD_CONFIG=true
```

```
export GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

```
./bin/grafana-server
```

Windows

Windows 上のスタンドアロン Grafana で SigV4 を有効にするには、Windows のコマンドプロンプトを使用して、次のコマンドを入力します。

```
set AWS_SDK_LOAD_CONFIG=true
```

```
set GF_AUTH_SIGV4_AUTH_ENABLED=true
```

```
cd grafana_install_directory
```

```
.\bin\grafana-server.exe
```

ステップ 2: Grafana に Prometheus データソースを追加する

以下の手順では、Grafana で Prometheus データソースを設定して、Amazon Managed Service for Prometheus メトリクスに対するクエリを実行する方法を説明します。

Grafana サーバーに Prometheus データソースを追加するには

1. Grafana コンソールを開きます。
2. [設定] で、[データソース] を選択します。

3. [データソースを追加] を選択します。
4. [Prometheus] を選択します。
5. HTTP にはURL、Amazon Managed Service for Prometheus コンソールのワークスペースの詳細ページに表示されるエンドポイント - クエリURLを指定します。
6. HTTP URL 先ほど指定した で、 に追加された`/api/v1/query`文字列を削除します。これはURL、Prometheus データソースによって自動的に追加されるためです。

   正しい URLは、https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9 のようになります。
7. [認証] で、[SigV4 認証] のトグルを選択して有効にします。
8. SigV4 認証は、Grafana で長期認証情報を直接指定するか、デフォルトのプロバイダーチェーンを使用して構成できます。長期認証情報を直接指定する方がすぐに開始できるため、以下では最初にその手順を説明します。Amazon Managed Service for Prometheus で Grafana を使用することに慣れてきたら、デフォルトのプロバイダーチェーンを使用することをお勧めします。これにより、柔軟性とセキュリティが向上します。デフォルトのプロバイダーチェーンを設定する方法の詳細については、「Specifying Credentials」を参照してください。

   - 長期認証情報を直接使用するには、以下を実行します。

     a. [SigV4 認証の詳細] で、[認証プロバイダー] として [アクセスとシークレットキー] を選択します。

     b. [アクセスキー ID] に、 AWS アクセスキー ID を入力します。

     c. [シークレットアクセスキー] に、 AWS シークレットアクセスキーを入力します。

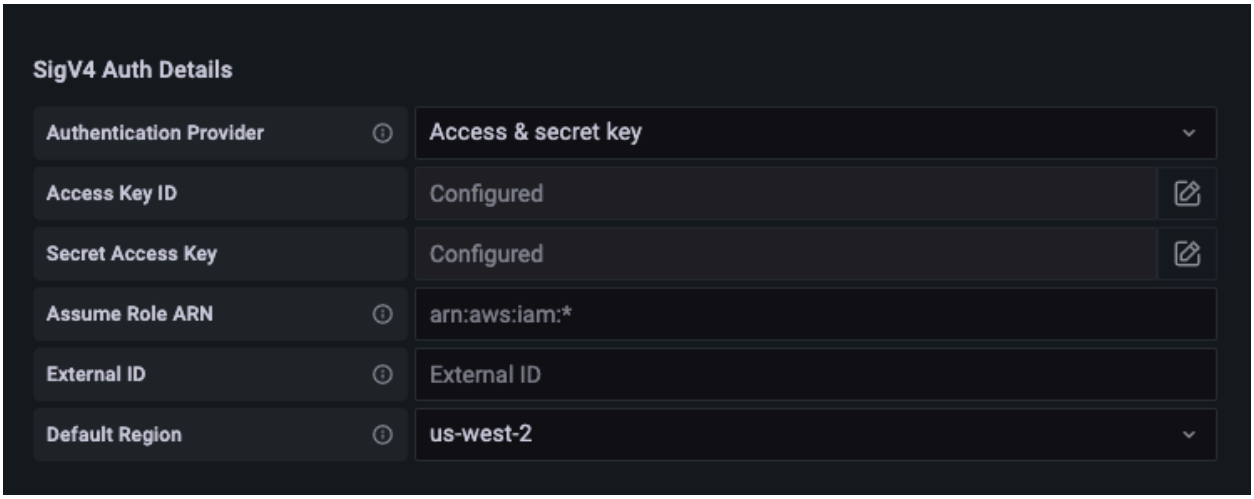
     d. ロールと外部 ID の引き受けARNフィールドは空白のままにします。

     e. [デフォルトのリージョン] で、Amazon Managed Service for Prometheus ワークスペースのリージョンを選択します。このリージョンは、ステップ 5 でリストURLした に含まれるリージョンと一致する必要があります。

     f. [保存してテスト] を選択します。

        「Data source is working」というメッセージが表示されます。

        次のスクリーンショットは、アクセスキーとシークレットキーを含む SigV4 認証の詳細設定を示しています。



- 代わりにデフォルトのプロバイダーチェーンを使用するには (本番環境に推奨)、以下を実行します。

  a. SigV4 認証の詳細 で、認証プロバイダー でAWS SDKデフォルト を選択します。

  b. ロールと外部 ID の引き受けARNフィールドは空白のままにします。

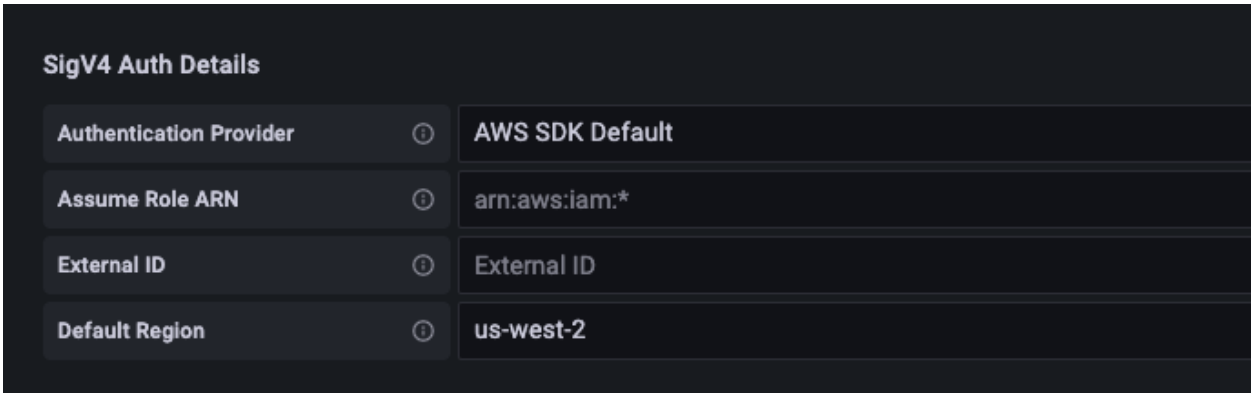
  c. [デフォルトのリージョン] で、Amazon Managed Service for Prometheus ワークスペースのリージョンを選択します。このリージョンは、ステップ 5 でリストURLした に含まれるリージョンと一致する必要があります。

  d. [保存してテスト] を選択します。

     「Data source is working」というメッセージが表示されます。

     このメッセージが表示されない場合は、次のセクションで接続に関するトラブルシューティングのヒントを提供します。

     次のスクリーンショットは、SDKデフォルトの SigV4 認証の詳細設定を示しています。



9. 新しいデータソースに対して PromQL クエリをテストします。

    a. [調査] を選択します。

    b. 次のようなサンプル PromQL クエリを実行します。

```
prometheus_tsdb_head_series
```

ステップ 3: (オプション) 保存とテストが機能しない場合のトラブルシューティング

前の手順で [保存してテスト] を選択したときにエラーが表示される場合は、以下を確認してください。

HTTP エラーが見つかりません

のワークスペース ID URLが正しいことを確認します。

HTTP エラーの禁止

このエラーは、認証情報が無効であることを意味します。以下をチェックしてください:

- [デフォルトのリージョン] に指定したリージョンが正しいことを確認します。
- 認証情報に誤字がないことを確認します。
- 使用している認証情報に AmazonPrometheusQueryAccessポリシーがあることを確認してください。詳細については、「IAM のアクセス許可とポリシー」を参照してください。
- 使用している認証情報に、この Amazon Managed Service for Prometheus ワークスペースへのアクセス権があることを確認します。

HTTP エラーの不正なゲートウェイ

このエラーのトラブルシューティングを行うには、Grafana サーバーのログを確認します。詳細については、Grafana ドキュメントの「Troubleshooting」を参照してください。

が表示された場合**`Error http: proxy error: NoCredentialProviders: no valid providers in chain`**、デフォルトの認証情報プロバイダーチェーンは、使用する有効な AWS 認証情報を見つけることができませんでした。「Specifying Credentials」に従って認証情報が設定されていることを確認してください。共有設定を使用する場合は、`AWS_SDK_LOAD_CONFIG` 環境が `true` に設定されていることを確認してください。

Amazon EKSクラスターで実行されている Grafana を使用したクエリ

Amazon Managed Service for Prometheus では、Grafana バージョン 7.3.5 以降を使用して Amazon Managed Service for Prometheus ワークスペース内のメトリクスにクエリを実行することがサポートされています。バージョン 7.3.5 以降には、 AWS 署名バージョン 4 (SigV4) 認証のサポートが含まれています。

Amazon Managed Service for Prometheus と連携するように Grafana を設定するには、AmazonPrometheusQueryAccessポリシーまたは `aps:QueryMetrics`、、`aps:GetMetricMetadata`、`aps:GetSeries`および アクセス`aps:GetLabels`許可を持つアカウントにログオンする必要があります。詳細については、「IAM のアクセス許可とポリシー」を参照してください。

AWS SigV4 をセットアップする

Grafana は、 AWS 署名バージョン 4 (SigV4) 認証をサポートする新機能を追加しました。詳細については、「Signature Version 4 の署名プロセス」を参照してください。Grafana サーバーでは、この機能はデフォルトで有効になっていません。ここでは、Kubernetes クラスターへの Grafana のデプロイに Helm を使用しているものと想定して、この機能を有効にする手順を説明します。

Grafana 7.3.5 以降のサーバーで SigV4 を有効にするには

1. Grafana の設定をオーバーライドする新しい更新ファイルを作成し、`amp_query_override_values.yaml` という名前を付けます。
2. 以下の内容をファイルに入力し、ファイルを保存します。置換 *`account-id`* Grafana サーバーが実行されている AWS アカウント ID を含む 。

```
serviceAccount:
    name: "amp-iamproxy-query-service-account"
    annotations:
        eks.amazonaws.com/role-arn: "arn:aws:iam::account-id:role/amp-iamproxy-
query-role"
grafana.ini:
  auth:
    sigv4_auth_enabled: true
```

そのYAMLファイルコンテンツでは、 `amp-iamproxy-query-role`は、次のセクションで作成するロールの名前ですサービスアカウントのIAMロールを設定する。ワークスペースに対してク

エリを実行するためのロールが既に作成されている場合は、ファイル内のロールを独自のロール名に置き換えることができます。

このファイルは、後の「Helm を使用した Grafana サーバーのアップグレード」で使用します。

サービスアカウントのIAMロールを設定する

Amazon EKSクラスターで Grafana サーバーを使用している場合は、アクセスコントロールにサービスIAMロールとも呼ばれるサービスアカウントのロールを使用することをお勧めします。これを実行して IAMロールを Kubernetes サービスアカウントに関連付けると、サービスアカウントはそのサービスアカウントを使用する任意のポッドのコンテナにアクセス AWS 許可を付与できます。詳細については、IAM「 サービスアカウントの ロール」を参照してください。

これらのクエリ用のサービスロールをまだ設定していない場合は、「メトリクスのクエリを実行するためのサービスアカウントの IAM ロールの設定」の手順に従ってロールを設定します。

その後、信頼関係の条件に Grafana サービスアカウントを追加する必要があります。

信頼関係の条件に Grafana サービスアカウントを追加するには

1. ターミナルウィンドウから、Grafana サーバーの名前空間とサービスアカウント名を確認します。例えば、次のコマンドを使用できます。

   ```
   kubectl get serviceaccounts -n grafana_namespace
   ```

2. Amazon EKSコンソールで、EKSクラスターに関連付けられているサービスアカウントのIAMロールを開きます。
3. [信頼関係の編集] を選択します。
4. Condition を更新して、ステップ 1 のコマンド出力で確認した Grafana 名前空間と Grafana サービスアカウント名を含めます。次に例を示します。

   ```
   {
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Federated": "arn:aws:iam::account-id:oidc-provider/
   oidc.eks.aws_region.amazonaws.com/id/openid"
         },
   ```

```
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "oidc.eks.region.amazonaws.com/id/openid:sub": [
            "system:serviceaccount:aws-amp:amp-iamproxy-query-service-account",
            "system:serviceaccount:grafana-namespace:grafana-service-account-name"
          ]
        }
      }
    }
  ]
}
```

5. [信頼ポリシーの更新] を選択します。

Helm を使用した Grafana サーバーのアップグレード

このステップでは、前のセクションで amp_query_override_values.yaml ファイルに追加したエントリを使用するように Grafana サーバーをアップグレードします。

以下のコマンドを実行します。Grafana 用の Helm チャートの詳細については、「Grafana Community Kubernetes Helm Charts」を参照してください。

```
helm repo add grafana https://grafana.github.io/helm-charts
```

```
helm upgrade --install grafana grafana/grafana -n grafana_namespace -f ./
amp_query_override_values.yaml
```

Grafana での Prometheus データソースの追加

以下の手順では、Grafana で Prometheus データソースを設定して、Amazon Managed Service for Prometheus メトリクスに対するクエリを実行する方法を説明します。

Grafana サーバーに Prometheus データソースを追加するには

1. Grafana コンソールを開きます。
2. [設定] で、[データソース] を選択します。
3. [データソースを追加] を選択します。
4. [Prometheus] を選択します。

5. HTTP にはURL、Amazon Managed Service for Prometheus コンソールのワークスペースの詳細ページに表示されるエンドポイント - クエリURLを指定します。
6. HTTP URL 先ほど指定した で、 に追加された`/api/v1/query`文字列を削除します。これはURL、Prometheus データソースによって自動的に追加されるためです。
7. [認証] で、[SigV4 認証] のトグルを選択して有効にします。

   ロールと外部 ID の引き受けARNフィールドは空白のままにします。 次に、[デフォルトのリージョン] で、Amazon Managed Service for Prometheus ワークスペースのあるリージョンを選択します。
8. [保存してテスト] を選択します。

   「Data source is working」というメッセージが表示されます。
9. 新しいデータソースに対して PromQL クエリをテストします。

   a. [調査] を選択します。

   b. 次のようなサンプル PromQL クエリを実行します。

   ```
   prometheus_tsdb_head_series
   ```

Prometheus 互換を使用したクエリ APIs

Amazon Managed Grafana などのツールを使用してメトリクスを表示およびクエリする最も簡単な方法ですが、Amazon Managed Service for Prometheus では、メトリクスのクエリAPIsに使用できる複数の Prometheus 互換もサポートしています。使用可能なすべての Prometheus 互換 の詳細についてはAPIs、「」を参照してくださいPrometheus 互換 API。

Prometheus 互換では、Prometheus クエリ言語 PromQL APIsを使用して、返すデータを指定します。PromQL とその構文の詳細については、Prometheus ドキュメントの「Prometheus のクエリ」を参照してください。

これらを使用してメトリクスAPIsをクエリする場合、リクエストは署名バージョン AWS 4 の署名プロセスで署名する必要があります。AWS Signature Version 4 をセットアップすると、署名プロセスを簡略化できます。詳細については、「aws-sigv4-proxy」を参照してください。

AWS SigV4 プロキシを介した署名は、 を使用して実行できますawscurl。次のトピックでは、awscurl を使用して Prometheus 互換のクエリAPIsを実行することで、 を使用して SigV4 をセットアップawscurl AWS する手順を説明します。

トピック

- awscurl を使用して Prometheus 互換のクエリを実行する APIs

awscurl を使用して Prometheus 互換のクエリを実行する APIs

API Amazon Managed Service for Prometheus の リクエストは、SigV4 で署名する必要があります。awscurl を使用すると、クエリのプロセスを簡略化できます。

awscurl をインストールするには、Python 3 と pip パッケージマネージャーがインストールされている必要があります。

Linux ベースのインスタンスでは、次のコマンドで awscurl をインストールします。

```
$ pip3 install awscurl
```

macOS マシンでは、次のコマンドで awscurl をインストールします。

```
$ brew install awscurl
```

次の例は、サンプルawscurlクエリです。を置き換える *Region*, *Workspace-id* また、*QUERY* ユースケースに適した値を持つ 入力 :

```
# Define the Prometheus query endpoint URL. This can be found in the Amazon Managed
 Service for Prometheus console page
# under the respective workspace.

$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
workspaces/Workspace-id/api/v1/query

# credentials are infered from the default profile
$ awscurl -X POST --region Region \
                  --service aps "${AMP_QUERY_ENDPOINT}" -d 'query=QUERY'  --header
 'Content-Type: application/x-www-form-urlencoded'
```

> **Note**
>
> クエリ文字列は URL エンコードされている必要があります。

のようなクエリではquery=up、次のような結果が得られます。

```
{
  "status": "success",
  "data": {
    "resultType": "vector",
    "result": [
      {
        "metric": {
          "__name__": "up",
          "instance": "localhost:9090",
          "job": "prometheus",
          "monitor": "monitor"
        },
        "value": [
          1652452637.636,
          "1"
        ]
      },
    ]
  }
}
```

指定したリクエストに awscurl で署名するには、有効な認証情報を以下のいずれかの方法で渡す必要があります。

- IAM ロールのアクセスキー ID とシークレットキーを指定します。ロールのアクセスキーとシークレットキーは、 にありますhttps://console.aws.amazon.com/iam/。

  例:

  ```
  $ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
  workspaces/Workspace_id/api/v1/query

  $ awscurl -X POST --region <Region> \
                    --access_key <ACCESS_KEY> \
                    --secret_key <SECRET_KEY> \
                    --service aps "$AMP_QUERY_ENDPOINT?query=<QUERY>"
  ```

- .aws/credentials および /aws/config ファイルに保存されている設定ファイルを参照する。使用するプロファイルの名前を指定することもできます。指定しない場合、 default ファイルが使用されます。例:

```
$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.<Region>.amazonaws.com/workspaces/
<Workspace_ID>/api/v1/query
$ awscurl -X POST --region <Region> \
                  --profile <PROFILE_NAME>
                  --service aps "$AMP_QUERY_ENDPOINT?query=<QUERY>"
```

- インスタンスに関連付けられたEC2インスタンスプロファイルを使用します。

awscurl コンテナを使用したクエリリクエストの実行

別のバージョンの Python がインストールされていて、関連する依存関係を満たすことができない場合は、コンテナを使用して awscurl アプリケーションとその依存関係をパッケージ化できます。次の例では、Docker ランタイムを使用して をデプロイしますがawscurl、OCI準拠しているランタイムとイメージは機能します。

```
$ docker pull okigan/awscurl
$ export AMP_QUERY_ENDPOINT=https://aps-workspaces.Region.amazonaws.com/
workspaces/Workspace_id/api/v1/query
$ docker run --rm -it okigan/awscurl --access_key $AWS_ACCESS_KEY_ID  --secret_key
 $AWS_SECRET_ACCESS_KEY \ --region Region --service aps "$AMP_QUERY_ENDPOINT?
query=QUERY"
```

各クエリのクエリ使用状況に関する統計を取得する

クエリの料金は、実行されたクエリから 1 か月間に処理されたクエリサンプルの合計数に基づいて計算されます。処理されたサンプルを追跡するために行った各クエリに関する統計を取得できます。query または のクエリレスポンスには、リクエストstats=allに クエリパラメータを含めることで処理されたクエリサンプルに関する統計データを含めるqueryRangeAPIことができます。samples オブジェクトは stats オブジェクトに作成され、statsデータはレスポンスで返されます。

samples オブジェクトは以下の属性で構成されます。

属性	説明
totalQueryableSamples	処理されたクエリサンプルの合計数。これが請求に使用される情報です。

属性	説明
totalQueryableSamplesPerStep	各ステップで処理されたクエリサンプルの数。これは、エポックタイムスタンプと、その特定のステップでロードされたサンプルの数から成る配列の配列として表されます。

サンプルのリクエストと、stats 情報を含むレスポンスの例を以下に示します。

query の例:

GET

```
endpoint/api/v1/query?query=up&time=1652382537&stats=all
```

レスポンス

```
{
    "status": "success",
    "data": {
        "resultType": "vector",
        "result": [
            {
                "metric": {
                    "__name__": "up",
                    "instance": "localhost:9090",
                    "job": "prometheus"
                },
                "value": [
                    1652382537,
                    "1"
                ]
            }
        ],
        "stats": {
            "timings": {
                "evalTotalTime": 0.00453349,
                "resultSortTime": 0,
                "queryPreparationTime": 0.000019363,
                "innerEvalTime": 0.004508405,
                "execQueueTime": 0.000008786,
```

```
                "execTotalTime": 0.004554219
            },
            "samples": {
                "totalQueryableSamples": 1,
                "totalQueryableSamplesPerStep": [
                    [
                        1652382537,
                        1
                    ]
                ]
            }
        }
    }
}
```

queryRange の例:

GET

```
endpoint/api/v1/query_range?query=sum+%28rate+%28go_gc_duration_seconds_count%5B1m%5D
%29%29&start=1652382537&end=1652384705&step=1000&stats=all
```

レスポンス

```
{
    "status": "success",
    "data": {
        "resultType": "matrix",
        "result": [
            {
                "metric": {},
                "values": [
                    [
                        1652383000,
                        "0"
                    ],
                    [
                        1652384000,
                        "0"
                    ]
                ]
            }
        ],
```

```
        "stats": {
            "samples": {
                "totalQueryableSamples": 8,
                "totalQueryableSamplesPerStep": [
                    [
                        1652382000,
                        0
                    ],
                    [
                        1652383000,
                        4
                    ],
                    [
                        1652384000,
                        4
                    ]
                ]
            }
        }
    }
}
```

ルールを使用して、受信時にメトリクスを変更またはモニタリングする

Amazon Managed Service for Prometheus によって受信されたメトリクスに基づいて動作するルールを設定できます。これらのルールは、メトリクスをモニタリングしたり、受信したメトリクスに基づいて新しい計算されたメトリクスを作成したりできます。

Amazon Managed Service for Prometheus は、定期的に評価される 2 種類のルールをサポートしています。

- 記録ルールでは、頻繁に必要になる式や計算負荷の高い式を事前に計算し、その結果を新しい時系列セットとして保存できます。多くの場合、事前に計算された結果に対してクエリを実行する方が、元の式を必要時に毎回実行するよりもはるかに高速です。
- アラートルールでは、PromQL としきい値に基づいてアラート条件を定義できます。ルールがしきい値をトリガーすると、アラートマネージャー に通知が送信されます。アラートマネージャーは、ルールを管理するように設定することも、Amazon Simple Notification Service などのレシーバーにダウンストリームの通知に転送することもできます。

Amazon Managed Service for Prometheus でルールを使用するには、ルールを定義する 1 つ以上の YAML ルールファイルを作成します。Amazon Managed Service for Prometheus のルールファイルの形式は、スタンドアロンの Prometheus のルールファイルと同じです。詳細については、Prometheus ドキュメントの「Defining Recording rules」と「Alerting rules」を参照してください。

ワークスペースには複数のルールファイルを含めることができます。それぞれのルールファイルは、別々の名前空間に格納されます。ルールファイルを複数にすれば、既存の Prometheus ルールファイルを変更したり結合したりする必要なく、そのままワークスペースにインポートできます。また、異なるルールグループ名前空間には、異なるタグを付けることができます。

ルールの順序

ルールファイル内では、ルールはルールグループに格納されます。ルールファイルの 1 つのルールグループ内のルールは、常に上から下に順番に評価されます。したがって、記録ルールでは、ある記録ルールの結果を、同じルールグループに含まれている後の記録ルールの計算やアラートルールで使用できます。ただし、個々のルールファイルの実行順序は指定できないため、ある記録ルールの結果を使用して別のルールグループまたは別のルールファイル内のルールを計算することはできません。

トピック

ルールの使用に必要な IAM アクセス許可について

ユーザーに、Amazon Managed Service for Prometheus でルールを使用するためのアクセス許可を付与する必要があります。以下のアクセス許可を含む AWS Identity and Access Management (IAM) ポリシーを作成し、そのポリシーをユーザー、グループ、ロールに割り当てます。

> **Note**
>
> IAM の詳細については、「Amazon Managed Service for Prometheus の Identity and Access Management」を参照してください。

ルールを使用するためのアクセス権を付与するポリシー

次のポリシーは、ルールを使用するためのアクセス権を、アカウント内のすべてのリソースに付与します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps: CreateRuleGroupsNamespace",
                "aps: ListRuleGroupsNamespaces",
                "aps: DescribeRuleGroupsNamespace",
                "aps: PutRuleGroupsNamespace",
                "aps: DeleteRuleGroupsNamespace",
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

1 つの名前空間にのみアクセス権を付与するポリシー

特定のポリシーにのみアクセス権を付与するポリシーを作成することもできます。次のサンプルポリシーは、指定された RuleGroupNamespace にのみアクセス権を付与します。このポリシーを使用するには、*<account>*、*<region>*、*<workspace-id>*、*<namespace-name>* を、アカウントに応じた適切な値に置き換えます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:ListRules",
                "aps:ListTagsForResource",
                "aps:GetLabels",
                "aps:CreateRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespaces",
                "aps:DescribeRuleGroupsNamespace",
                "aps:PutRuleGroupsNamespace",
                "aps:DeleteRuleGroupsNamespace"
            ],
            "Resource": [
                "arn:aws:aps:*:<account>:workspace/*",
                "arn:aws:aps:<region>:<account>:rulegroupnamespace/<workspace-
id>/<namespace-name>"
            ]
        }
    ]
}
```

ルールファイルを作成する

Amazon Managed Service for Prometheus でルールを使用するには、ルールを定義するルールファイルを作成します。Amazon Managed Service for Prometheus のルールファイルの形式は、スタンドアロンの Prometheus のルールファイルと同じです。詳細については、「Defining Recording rules」と「Alerting rules」を参照してください。

ルールファイルの基本的な例を以下に示します。

```
groups:
  - name: test
    rules:
    - record: metric:recording_rule
      expr: avg(rate(container_cpu_usage_seconds_total[5m]))
  - name: alert-test
    rules:
    - alert: metric:alerting_rule
      expr: avg(rate(container_cpu_usage_seconds_total[5m])) > 0
      for: 2m
```

アラートルールのその他の例については、「Alerting rule examples」を参照してください。

> **Note**
>
> ルール定義ファイルをローカルで作成して Amazon Managed Service for Prometheus にアップロードすることも、Amazon Managed Service for Prometheus コンソール内で直接定義を作成、編集、アップロードすることもできます。いずれの場合も、同じフォーマットルールが適用されます。ファイルのアップロードと編集の詳細については、「」を参照してくださいAmazon Managed Service for Prometheus にルール設定ファイルをアップロードする。

Amazon Managed Service for Prometheus にルール設定ファイルをアップロードする

ルール設定ファイルに必要なルールがわかったら、コンソール内で作成および編集するか、コンソールまたは を使用してファイルをアップロードできます AWS CLI。

> **Note**
>
> Amazon EKS クラスターを実行している場合は、 AWS Controllers for Kubernetes を使用してルール設定ファイルをアップロードすることもできます。

Amazon Managed Service for Prometheus コンソールを使用してルール設定を編集または置き換え、名前空間を作成するには

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. ワークスペースのワークスペース ID を選択し、[ルール管理] タブを選択します。
4. [名前空間を追加] を選択します。
5. [ファイルを選択] を選択し、ルール定義ファイルを選択します。

   または、設定の定義 を選択して、Amazon Managed Service for Prometheus コンソールでルール定義ファイルを直接作成および編集することもできます。これにより、アップロード前に編集するサンプルのデフォルト定義ファイルが作成されます。
6. (オプション) 名前空間にタグを追加するには、[新しいタグを追加] を選択します。

   [キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。

   別のタグを追加するには、[新しいタグを追加] を選択します。
7. [続行] を選択します。Amazon Managed Service for Prometheus は、選択したルールファイルと同じ名前で新しい名前空間を作成します。

を使用してアラートマネージャー設定を新しい名前空間のワークスペース AWS CLI にアップロードするには

1. アラートマネージャーファイルの内容を base64 でエンコードします。Linux では、次のコマンドを使用できます。

   ```
   base64 input-file output-file
   ```

   macOS では、次のコマンドを使用できます。

   ```
   openssl base64 input-file output-file
   ```
2. 以下のいずれかのコマンドを入力して、名前空間の作成とファイルのアップロードを行います。

   AWS CLI バージョン 2 では、次のように入力します。

   ```
   aws amp create-rule-groups-namespace --data file://path_to_base_64_output_file --
   name namespace-name  --workspace-id my-workspace-id --region region
   ```

   AWS CLI バージョン 1 で、次のように入力します。

```
aws amp create-rule-groups-namespace --data fileb://path_to_base_64_output_file --
name namespace-name  --workspace-id my-workspace-id --region region
```

3. アラートマネージャーの設定が有効になるまで数秒かかります。ステータスを確認するには、次のコマンドを入力します。

```
aws amp describe-rule-groups-namespace --workspace-id workspace_id --
name namespace-name --region region
```

`status` が `ACTIVE` であれば、ルールファイルが有効になっています。

ルール設定ファイルを編集または置き換える

Amazon Managed Service for Prometheus に既にアップロードしているルールファイルのルールを変更する場合は、新しいルールファイルをアップロードして既存の設定を置き換えるか、コンソールで現在の設定を直接編集できます。必要に応じて、現在のファイルをダウンロードし、テキストエディタで編集して、新しいバージョンをアップロードできます。

Amazon Managed Service for Prometheus コンソールを使用してルール設定を編集するには

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. ワークスペースのワークスペース ID を選択し、[ルール管理] タブを選択します。
4. 編集するルール設定ファイルの名前を選択します。
5. (オプション) 現在のルール設定ファイルをダウンロードする場合は、ダウンロードまたはコピーを選択します。
6. 変更 を選択して、コンソール内で設定を直接編集します。完了したら保存を選択します。

   または、設定の置き換えを選択して、新しい設定ファイルをアップロードすることもできます。その場合は、新しいルール定義ファイルを選択し、続行を選択してアップロードします。

を使用してルール設定ファイル AWS CLI を編集するには

1. ルールファイルの内容を base64 でエンコードします。Linux では、次のコマンドを使用できます。

```
base64 input-file output-file
```

macOS では、次のコマンドを使用できます。

```
openssl base64 input-file output-file
```

2. 以下のいずれかのコマンドを入力して、新しいファイルをアップロードします。

   AWS CLI バージョン 2 では、次のように入力します。

```
aws amp put-rule-groups-namespace --data file://path_to_base_64_output_file --
name namespace-name  --workspace-id my-workspace-id --region region
```

   AWS CLI バージョン 1 で、次のように入力します。

```
aws amp put-rule-groups-namespace --data fileb://path_to_base_64_output_file --
name namespace-name  --workspace-id my-workspace-id --region region
```

3. ルールファイルが有効になるまで数秒かかります。ステータスを確認するには、次のコマンドを入力します。

```
aws amp describe-rule-groups-namespace --workspace-id workspace_id --
name namespace-name --region region
```

   `status` が `ACTIVE` であれば、ルールファイルが有効になっています。それまでは、このルールファイルの以前のバージョンが有効なままになります。

ルーラーのトラブルシューティング

CloudWatch ログによる Amazon Managed Service for Prometheus イベントのモニタリング を使用すると、アラートマネージャーとルーラーに関する問題のトラブルシューティングを行うことができます。このセクションには、ルーラー関連のトラブルシューティングトピックが含まれています。

ログに次のルーラー失敗エラーが含まれている場合

```
{
    "workspaceId": "ws-12345c67-89c0-4d12-345b-f14db70f7a99",
```

```
    "message": {
        "log": "Evaluating rule failed, name=failure,
 group=canary_long_running_vl_namespace, namespace=canary_long_running_vl_namespace,
 err=found duplicate series for the match group {dimension1=\\\"1\\\"} on the right
 hand-side of the operation: [{__name__=\\\"fake_metric2\\\", dimension1=\\\"1\\
\", dimension2=\\\"b\\\"}, {__name__=\\\"fake_metric2\\\", dimension1=\\\"1\\\",
 dimension2=\\\"a\\\"}];many-to-many matching not allowed: matching labels must be
 unique on one side",
        "level": "ERROR",
        "name": "failure",
        "group": "canary_long_running_vl_namespace",
        "namespace": "canary_long_running_vl_namespace"
    },
    "component": "ruler"
}
```

これは、ルールの実行中に何らかのエラーが発生したことを示します。

実行するアクション

エラーメッセージを使用して、ルールの実行のトラブルシューティングを行います。

アラートマネージャーによる Amazon Managed Service for Prometheus でのアラートの管理と転送

Amazon Managed Service for Prometheus が実行するアラートルールが発動すると、送信されたアラートはアラートマネージャーによって処理されます。アラートマネージャーは、アラートの重複排除、グループ化、ダウンストリームのレシーバーへのルーティングを行います。Amazon Managed Service for Prometheus は、レシーバーとして Amazon Simple Notification Service のみをサポートし、同じアカウントの Amazon SNS トピックにメッセージをルーティングできます。アラートマネージャーを使用して、アラートを無音にしたり禁止したりすることもできます。

アラートマネージャーは、Prometheus の Alertmanager と同様の機能を提供します。

アラートマネージャーの設定ファイルを使用すると、次の機能を構成できます。

- グループ化 - 類似するアラートを 1 つの通知にまとめます。これは特に、多数のシステムに同時に障害が発生し、大量のアラートが同時に発生する可能性のある大規模なシステム停止時に役立ちます。例えば、ネットワーク障害により、多くのノードに同時に障害が発生したとします。これらのタイプのアラートがグループ化されていると、アラートマネージャーが送信する通知は 1 つになります。

  アラートのグループ化とグループ化された通知のタイミングは、アラートマネージャー設定ファイルのルーティングツリーによって構成されます。詳細については、「<route>」を参照してください。
- 禁止 - 他の特定のアラートが既に発動している場合に、特定のアラートの通知を抑制します。例えば、クラスターに到達できないというアラートが発動している場合に、そのクラスターに関する他のすべてのアラートをミュートするようにアラートマネージャーを構成できます。これにより、実際の問題とは関係のないアラートが大量に発生するのを防ぐことができます。禁止ルールの記述方法の詳細については、「<inhibit_rule>」を参照してください。
- サイレンス - メンテナンスの時間帯など、指定した時間だけアラートをミュートします。受信したアラートに対して、アクティブなサイレンスのすべての等価式または正規表現マッチャーと一致するかどうかのチェックが行われます。一致した場合、そのアラートに関する通知は送信されません。

  サイレンスを作成するには、`PutAlertManagerSilences` API を使用します。詳細については、「PutAlertManagerSilences」を参照してください。

Prometheus テンプレート

Standalone Prometheus は、個別のテンプレートファイルを使用したテンプレート作成をサポートしています。テンプレートでは、条件の使用、データのフォーマットど、さまざまな処理を行うことができます。

Amazon Managed Service for Prometheus では、テンプレートをアラートマネージャー設定 と同じアラートマネージャー設定ファイルに配置します。

トピック

- アラートマネージャーの使用に必要な IAM アクセス許可について
- Amazon Managed Service for Prometheus でアラートマネージャー設定を作成して、アラートを管理およびルーティングする
- Amazon Managed Service for Prometheus でアラートマネージャーを使用してアラートをアラートレシーバーに転送する
- アラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする
- アラートを Amazon Managed Grafana またはオープンソース Grafana と統合する
- CloudWatch Logs を使用したアラートマネージャーのトラブルシューティング

アラートマネージャーの使用に必要な IAM アクセス許可について

Amazon Managed Service for Prometheus でアラートマネージャーを使用するアクセス許可をユーザーに付与する必要があります。次のアクセス許可を持つ AWS Identity and Access Management (IAM) ポリシーを作成し、そのポリシーをユーザー、グループ、またはロールに割り当てます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps: CreateAlertManagerDefinition",
                "aps: DescribeAlertManagerSilence",
                "aps: DescribeAlertManagerDefinition",
                "aps: PutAlertManagerDefinition",
                "aps: DeleteAlertManagerDefinition",
                "aps: ListAlerts",
```

```
                "aps: ListRules",
                "aps: ListAlertManagerReceivers",
                "aps: ListAlertManagerSilences",
                "aps: ListAlertManagerAlerts",
                "aps: ListAlertManagerAlertGroups",
                "aps: GetAlertManagerStatus",
                "aps: GetAlertManagerSilence",
                "aps: PutAlertManagerSilences",
                "aps: DeleteAlertManagerSilence",
                "aps: CreateAlertManagerAlerts"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Managed Service for Prometheus でアラートマネージャー設定を作成して、アラートを管理およびルーティングする

Amazon Managed Service for Prometheus でアラートマネージャーとテンプレートを使用するには、アラートマネージャーの設定 YAML ファイルを作成します。Amazon Managed Service for Prometheus アラートマネージャーファイルには、次の 2 つの主要なセクションがあります。

- `template_files:` には、レシーバーから送信されたメッセージに使用されるテンプレートが含まれます。詳細については、Prometheus ドキュメントの「Template Reference」と「Template Examples」を参照してください。
- `alertmanager_config:` には、アラートマネージャーの設定が含まれます。これには、スタンドアロンの Prometheus のアラートマネージャー設定ファイルと同じ構造が使用されます。詳細については、Alertmanager ドキュメントの「Configuration」を参照してください。

> **Note**
>
> Amazon Managed Service for Prometheus では、上記の Prometheus ドキュメントで説明されている `repeat_interval` 設定に追加の制限があります。許容される最大値は 5 日間です。5 日より大きい値に設定しても 5 日間として扱われ、5 日間の期間が経過すると通知が再送信されます。

> **Note**
>
> Amazon Managed Service for Prometheus コンソールで設定ファイルを直接編集することもできますが、ここで指定した形式に従う必要があります。設定ファイルのアップロードまたは編集の詳細については、「」を参照してくださいアラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする。

Amazon Managed Service for Prometheus のアラートマネージャー設定ファイルでは、YAML ファイルのルートにある `alertmanager_config` キーの中に、アラートマネージャーの設定内容をすべて含める必要があります。

以下は、アラートマネージャー設定ファイルの基本的な例です。

```
alertmanager_config: |
  route:
    receiver: 'default'
  receivers:
    - name: 'default'
      sns_configs:
      - topic_arn: arn:aws:sns:us-east-2:123456789012:My-Topic
        sigv4:
          region: us-east-2
        attributes:
          key: key1
          value: value1
```

現在サポートされているレシーバーは、Amazon Simple Notification Service (Amazon SNS) だけです。設定に他の種類のレシーバーが指定されている場合、その設定は拒否されます。

以下は別のアラートマネージャー設定ファイルの例で、`template_files` ブロックと `alertmanager_config` ブロックの両方を使用しています。

```
template_files:
  default_template: |
    {{ define "sns.default.subject" }}[{{ .Status | toUpper }}{{ if eq .Status
 "firing" }}:{{ .Alerts.Firing | len }}{{ end }}]{{ end }}
    {{ define "__alertmanager" }}AlertManager{{ end }}
    {{ define "__alertmanagerURL" }}{{ .ExternalURL }}/#/alerts?receiver={{ .Receiver |
 urlquery }}{{ end }}
alertmanager_config: |
```

```
  global:
  templates:
    - 'default_template'
  route:
    receiver: default
  receivers:
    - name: 'default'
      sns_configs:
      - topic_arn: arn:aws:sns:us-east-2:accountid:My-Topic
        sigv4:
          region: us-east-2
        attributes:
          key: severity
          value: SEV2
```

デフォルトの Amazon SNS テンプレートブロック

デフォルトの Amazon SNS の設定では、明示的にオーバーライドしない限り、以下のテンプレートが使用されます。

```
{{ define "sns.default.message" }}{{ .CommonAnnotations.SortedPairs.Values | join "
 " }}
  {{ if gt (len .Alerts.Firing) 0 -}}
  Alerts Firing:
    {{ template "__text_alert_list" .Alerts.Firing }}
  {{- end }}
  {{ if gt (len .Alerts.Resolved) 0 -}}
  Alerts Resolved:
    {{ template "__text_alert_list" .Alerts.Resolved }}
  {{- end }}
{{- end }}
```

Amazon Managed Service for Prometheus でアラートマネージャーを使用してアラートをアラートレシーバーに転送する

アラートルールによってアラートが発生すると、アラートマネージャーに送信されます。アラートマネージャーは、アラートの重複排除、メンテナンス中のアラートの禁止、必要に応じてグループ化などの機能を実行します。次に、アラートをメッセージとしてアラート受信者 に転送します。オペレーターへの通知、自動応答、その他の方法でアラートに応答できるアラートレシーバーを設定できます。

Amazon Managed Service for Prometheus でサポートされているアラートレシーバーは、Amazon Simple Notification Service (Amazon SNS) のみです。詳細については、「Amazon SNS とは」を参照してください。Amazon SNS は、E メール、SMS、HTTP エンドポイントなどの他のシステムへの転送など、さまざまな方法でアラートに応答するために使用できます。

以下のトピックでは、Amazon SNS アラートレシーバーの作成と設定に関連するタスクについて説明します。

トピック

- Amazon Managed Service for Prometheus でアラートレシーバーとして使用する新しい Amazon SNS トピックの作成
- Amazon SNS トピックにアラートメッセージを送信するアクセス許可を Amazon Managed Service for Prometheus に付与する
- Amazon SNS トピックにメッセージを送信するようにアラートマネージャーを設定する
- Amazon SNS に JSON としてメッセージを送信するようにアラートマネージャーを設定する
- アラートのメッセージを他の送信先に送信するように Amazon SNS を設定する
- Amazon SNS メッセージ検証ルールについて

Amazon Managed Service for Prometheus でアラートレシーバーとして使用する新しい Amazon SNS トピックの作成

既存の Amazon SNS トピックを Amazon Managed Service for Prometheus のアラートレシーバーとして使用することも、新しいトピックを作成することもできます。[標準] タイプのトピックを使用することをお勧めします。このタイプでは、トピックから E メール、SMS、HTTP にアラートを転送できます。

アラートマネージャーのレシーバーとして使用する新しい Amazon SNS トピックを作成するには、「ステップ 1: トピックを作成する」の手順に従います。トピックのタイプとして、必ず [標準] を選択してください。

その Amazon SNS トピックにメッセージが送信されるたびに E メールを受信するには、「ステップ 2: トピックに対するサブスクリプションを作成する」の手順に従います。

新規または既存の Amazon SNS トピックを使用するかどうかにかかわらず、以下のタスクを完了するには、Amazon SNS トピックの Amazon リソースネーム (ARN) が必要です。

Amazon SNS トピックにアラートメッセージを送信するアクセス許可を Amazon Managed Service for Prometheus に付与する

Amazon SNS トピックにメッセージを送信するには、Amazon Managed Service for Prometheus にアクセス許可を付与する必要があります。次のポリシーステートメントは、そのアクセス許可を付与します。これには、混乱した代理問題と呼ばれるセキュリティ問題を防ぐのに役立つConditionステートメントが含まれています。Condition ステートメントは、Amazon SNS トピックへのアクセスを制限し、この特定のアカウントと Amazon Managed Service for Prometheus ワークスペースからのオペレーションのみを許可します。混乱した代理に関する問題の詳細については、「サービス間での不分別な代理処理の防止」を参照してください。

Amazon SNS トピックにメッセージを送信するためのアクセス許可を Amazon Managed Service for Prometheus に付与するには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. ナビゲーションペインで、[トピック] を選択します。
3. Amazon Managed Service for Prometheus で使用しているトピックの名前を選択します。
4. [編集] を選択します。
5. [アクセスポリシー] を選択し、次のポリシーステートメントを既存のポリシーに追加します。

```
{
    "Sid": "Allow_Publish_Alarms",
    "Effect": "Allow",
    "Principal": {
        "Service": "aps.amazonaws.com"
    },
    "Action": [
        "sns:Publish",
        "sns:GetTopicAttributes"
    ],
    "Condition": {
        "ArnEquals": {
            "aws:SourceArn": "workspace_ARN"
        },
        "StringEquals": {
            "AWS:SourceAccount": "account_id"
        }
    },
    "Resource": "arn:aws:sns:region:account_id:topic_name"
```

```
}
```

〔オプション] Amazon SNS トピックがサービス側の暗号化 (SSE) が有効になっている場合は、トピックの暗号化に使用される AWS KMS キーのキーポリシーに kms:GenerateDataKey*および アクセスkms:Decrypt許可を追加することで、Amazon Managed Service for Prometheus がこの暗号化されたトピックにメッセージを送信できるようにする必要があります。

例えば、ポリシーに以下を追加できます。

```
{
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "aps.amazonaws.com"
    },
    "Action": [
      "kms:GenerateDataKey*",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }]
}
```

詳細については、「SNS トピックに対するAWS KMS アクセス許可」を参照してください。

6. [変更を保存] を選択します。

> **Note**
>
> デフォルトでは、Amazon SNS は AWS:SourceOwner に条件を設定したアクセスポリシーを作成します。詳細については、「SNS アクセスポリシー」を参照してください。

> **Note**
>
> IAM は、最も制限の厳しいポリシーを優先するルールに従います。SNS トピックに、ドキュメント化された Amazon SNS ポリシーブロックよりも制限の厳しいポリシーブロックがあ

る場合、トピックポリシーのアクセス許可は付与されません。ポリシーを評価して何が許可されているかを確認するには、「ポリシーの評価論理」を参照してください。

サービス間での不分別な代理処理の防止

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、 は、アカウント内のリソースへのアクセスが許可されているサービスプリンシパルを持つすべてのサービスのデータを保護するのに役立つツール AWS を提供します。

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 Service for Prometheus が Amazon SNS に付与するリソースへのアクセス許可を制限することをお勧めします。両方のグローバル条件コンテキストキーを同じポリシーステートメントで使用する場合は、`aws:SourceAccount` 値と、`aws:SourceArn` 値に含まれるアカウントが、同じアカウント ID を示している必要があります。

`aws:SourceArn` の値は、Amazon Managed Service for Prometheus ワークスペースの ARN でなければなりません。

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:`*`servicename`*`::`*`123456789012`*`:*` のように指定します。

「Amazon SNS トピックにアラートメッセージを送信するアクセス許可を Amazon Managed Service for Prometheus に付与する」に記載されているポリシーは、Amazon Managed Service for Prometheus で `aws:SourceArn` および `aws:SourceAccount` グローバル条件コンテキストキーを使用して、混乱した代理問題を防止する方法を示しています。

Amazon SNS トピックにメッセージを送信するようにアラートマネージャーを設定する

(新規または既存の) 標準タイプの Amazon SNS トピックを作成したら、アラートマネージャー設定にアラートレシーバーとして追加できます。アラートマネージャーは、設定されたアラートレシーバーにアラートを転送できます。これを完了するには、Amazon SNS トピックの Amazon リソースネーム (ARN) を把握している必要があります。

Amazon SNS のレシーバー設定の詳細については、Prometheus の構成に関するドキュメントの「<sns_configs>」を参照してください。

サポートされないプロパティ

Amazon Managed Service for Prometheus では、アラートのレシーバーとして Amazon SNS がサポートされています。ただし、サービスの制約により、Amazon SNS レシーバーのすべてのプロパティがサポートされるわけではありません。以下のプロパティは、Amazon Managed Service for Prometheus のアラートマネージャー設定ファイルでは使用できません。

- `api_url:` - `api_url` は Amazon Managed Service for Prometheus によって自動的に設定されるため、このプロパティは使用できません。
- `Http_config` - このプロパティは、外部プロキシを設定できるようにするものです。Amazon Managed Service for Prometheus では、この機能は現在サポートされていません。

また、SigV4 設定にはリージョンプロパティを含める必要があります。リージョンプロパティを指定しない場合、認証リクエストを行うための十分な情報が Amazon Managed Service for Prometheus に提供されません。

Amazon SNS トピックをレシーバーとしてアラートマネージャーを構成するには

1. 既存のアラートマネージャー設定ファイルを使用している場合は、テキストエディタでそのファイルを開きます。
2. `receivers` ブロックに Amazon SNS 以外の現在のレシーバーがある場合は、それらを削除します。複数の Amazon SNS トピックをレシーバーとして構成するには、各トピックを `receivers` ブロック内の個別の `sns_config` ブロックに追加します。
3. `receivers` セクションに次の YAML ブロックを追加します。

```
- name: name_of_receiver
  sns_configs:
```

```
    - sigv4:
        region: region
      topic_arn: ARN_of_SNS_topic
      subject: somesubject
      attributes:
        key: somekey
        value: somevalue
```

subject を指定しない場合、デフォルトでは、ラベル名と値を使用するデフォルトテンプレートから件名が生成されますが、この値は SNS には長すぎる可能性があります。件名に適用されるテンプレートを変更するには、このガイドの「Amazon SNS に JSON としてメッセージを送信するようにアラートマネージャーを設定する」を参照してください。

この後は、アラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする必要があります。詳細については、「アラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする」を参照してください。

Amazon SNS に JSON としてメッセージを送信するようにアラートマネージャーを設定する

デフォルトでは、Amazon Managed Service for Prometheus アラートマネージャーはプレーンテキストリスト形式でメッセージを出力します。これは、他のサービスが解析するのがより難しい場合があります。代わりに JSON 形式でアラートを送信するようにアラートマネージャーを設定できます。JSON を使用すると、 AWS Lambda またはウェブフック受信エンドポイントで Amazon SNS のダウンストリームのメッセージを簡単に処理できます。デフォルトのテンプレートを使用する代わりに、メッセージの内容を JSON で出力するカスタムテンプレートを定義すると、ダウンストリーム関数での解析が容易になります。

アラートマネージャーから Amazon SNS に JSON 形式でメッセージを出力するには、アラートマネージャーの設定を更新して、template_files ルートセクション内に次のコードを含めます。

```
default_template: |
   {{ define "sns.default.message" }}{{ "{" }}"receiver": "{{ .Receiver }}","status":
 "{{ .Status }}","alerts": [{{ range $alertIndex, $alerts := .Alerts }}{{ if
 $alertIndex }}, {{ end }}{{ "{" }}"status": "{{ $alerts.Status }}"{{ if
 gt (len $alerts.Labels.SortedPairs) 0 -}},"labels": {{ "{" }}{{ range
 $index, $label := $alerts.Labels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $label.Name }}": "{{ $label.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ if gt (len $alerts.Annotations.SortedPairs )
 0 -}},"annotations": {{ "{" }}{{ range $index, $annotations :=
```

```
 $alerts.Annotations.SortedPairs }}{{ if $index }}, {{ end }}"{{ $annotations.Name }}":
 "{{ $annotations.Value }}"{{ end }}{{ "}" }}{{- end }},"startsAt":
 "{{ $alerts.StartsAt }}","endsAt": "{{ $alerts.EndsAt }}","generatorURL":
 "{{ $alerts.GeneratorURL }}","fingerprint": "{{ $alerts.Fingerprint }}"{{ "}" }}
{{ end }}]{{ if gt (len .GroupLabels) 0 -}},"groupLabels": {{ "{" }}{{ range
 $index, $groupLabels := .GroupLabels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $groupLabels.Name }}": "{{ $groupLabels.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ if gt (len .CommonLabels) 0 -}},"commonLabels": {{ "{" }}
{{ range $index, $commonLabels := .CommonLabels.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $commonLabels.Name }}": "{{ $commonLabels.Value }}"{{ end }}{{ "}" }}{{-
 end }}{{ if gt (len .CommonAnnotations) 0 -}},"commonAnnotations": {{ "{" }}{{ range
 $index, $commonAnnotations := .CommonAnnotations.SortedPairs }}{{ if $index }},
 {{ end }}"{{ $commonAnnotations.Name }}": "{{ $commonAnnotations.Value }}"{{ end }}
{{ "}" }}{{- end }}{{ "}" }}{{ end }}
   {{ define "sns.default.subject" }}[{{ .Status | toUpper }}{{ if eq .Status
 "firing" }}:{{ .Alerts.Firing | len }}{{ end }}]{{ end }}
```

> **Note**
>
> このテンプレートは、英数字データから JSON を作成します。データに特殊文字が含まれている場合は、このテンプレートを使用する前にそれらをエンコードしてください。

このテンプレートが送信通知で使用されるようにするには、次のように `alertmanager_config` ブロックでテンプレートを参照します。

```
alertmanager_config: |
  global:
  templates:
    - 'default_template'
```

> **Note**
>
> このテンプレートは、メッセージ本文全体を JSON として出力するものです。このテンプレートにより、メッセージ本文全体が上書きされます。この特定のテンプレートを使用する場合、メッセージ本文をオーバーライドすることはできません。手動で行ったオーバーライドは、テンプレートよりも優先されます。

詳細については、以下を参照してください。

- アラートマネージャー設定ファイルについては、「Amazon Managed Service for Prometheus でアラートマネージャー設定を作成して、アラートを管理およびルーティングする」を参照してください。
- 設定ファイルのアップロードについては、「アラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする」を参照してください。

アラートのメッセージを他の送信先に送信するように Amazon SNS を設定する

Amazon Managed Service for Prometheus は、Amazon Simple Notification Service (Amazon SNS) にのみアラートメッセージを送信できます。これらのメッセージを E メール、ウェブフック、Slack、または などの他の宛先に送信するには OpsGenie、それらのエンドポイントにメッセージを転送するように Amazon SNS を設定する必要があります。

以下のセクションでは、アラートを他の送信先に転送するように Amazon SNS を設定する方法について説明します。

トピック

Email(メール)

E メールにメッセージを出力するように Amazon SNS トピックを構成するには、サブスクリプションを作成します。Amazon SNS コンソールで、[サブスクリプション] タブを選択して、[サブスクリプション] リストページを開きます。[サブスクリプションの作成] を選択し、[E メール] を選択します。Amazon SNS から、指定した E メールアドレスに確認メールが送信されます。確認を受け入れると、サブスクライブしたトピックから Amazon SNS 通知を E メールとして受信できます。詳細については、「Amazon SNS トピックへサブスクライブする」を参照してください。

ウェブフック

ウェブフックエンドポイントにメッセージを出力するように Amazon SNS トピックを構成するには、サブスクリプションを作成します。Amazon SNS コンソールで、[サブスクリプション] タブ

を選択して、[サブスクリプション] リストページを開きます。[サブスクリプションの作成] を選択し、[HTTP/HTTPS] を選択します。サブスクリプションを作成したら、確認手順に従ってサブスクリプションをアクティブにする必要があります。アクティブになると、HTTP エンドポイントで Amazon SNS 通知が受信されます。詳細については、「Amazon SNS トピックへサブスクライブする」を参照してください。Slack ウェブフックを使用してさまざまな宛先にメッセージを発行する方法の詳細については、「ウェブフックを使用して Amazon SNS メッセージを Amazon Chime、Slack、または Microsoft Teams に発行するにはどうすればよいですか?」を参照してください。

Slack

Slack にメッセージを出力するように Amazon SNS トピックを構成するには、2 つの方法があります。Slack が E メールメッセージを受信して Slack チャネルに転送できるようにする Slack email-to-channel の統合と統合することも、Lambda 関数を使用して Amazon SNS 通知を Slack に書き換えることもできます。E メールを Slack チャネルに転送する方法の詳細については、「Slack Webhook の AWS SNS トピックサブスクリプションの確認」を参照してください。Lambda 関数を作成して Amazon SNS メッセージを Slack に変換する方法の詳細については、「How to integrate Amazon Managed Service for Prometheus with Slack」を参照してください。

OpsGenie

にメッセージを出力するように Amazon SNS トピックを設定する方法については OpsGenie、「Opsgenie を着信 Amazon SNS と統合する」を参照してください。

Amazon SNS メッセージ検証ルールについて

Amazon Simple Notification Service (Amazon SNS) では、特定の基準を満たすためにメッセージが必要です。これらの標準を満たしていないメッセージは、受信時に変更されます。アラートメッセージは、以下のルールに基づいて、必要に応じて Amazon SNS レシーバーによって検証、切り捨て、または変更されます。

- メッセージに UTF 以外の文字が含まれている場合。
  - メッセージは「Error - not a valid UTF-8 encoded string.」に置き換えられます。
  - キーが「truncated」で値が「true」のメッセージ属性が 1 つ追加されます。
  - キーが「modified」で値が「Message: Error - not a valid UTF-8 encoded string.」のメッセージ属性が 1 つ追加されます。
- メッセージが空の場合。

  - メッセージは「Error - Message should not be empty.」に置き換えられます。
  - キーが「modified」で値が「Message: Error - Message should not be empty.」のメッセージ属性が 1 つ追加されます。
- メッセージが切り捨てられた場合。
  - メッセージは切り捨てられたコンテンツになります。
  - キーが「truncated」で値が「true」のメッセージ属性が 1 つ追加されます。
  - キーが「modified」で値が「Message: Error - Message has been truncated from *X* KB, because it exceeds the 256 KB size limit.」のメッセージ属性が 1 つ追加されます。
- 件名が ASCII でない場合。
  - 件名は「Error - contains non printable ASCII characters.」に置き換えられます。
  - キーが「modified」で値が「Subject: Error - contains non-printable ASCII characters.」のメッセージ属性が 1 つ追加されます。
- 件名が切り捨てられた場合。
  - 件名は切り捨てられたコンテンツになります。
  - キーが「modified」で値が「Subject: Error - Subject has been truncated from *X* characters, because it exceeds the 100 character size limit.」のメッセージ属性が 1 つ追加されます。
- メッセージ属性のキー/値が無効な場合。
  - 無効なメッセージ属性は削除されます。
  - キーが「modified」、値がMessageAttribute「: Error - *X* of the message attributes have been removed becasue of invalid MessageAttributeKey or 」のメッセージ属性が 1 つ追加されます MessageAttributeValue。
- メッセージ属性が切り捨てられた場合。
  - 余分なメッセージ属性は削除されます。
  - キーが「modified」で値がMessageAttribute「: Error - *X* of the message attributes have been removed, because it exceeds the 256KB size limit.」のメッセージ属性が 1 つ追加されます。

アラートマネージャー設定ファイルを Amazon Managed Service for Prometheus にアップロードする

アラートマネージャー設定ファイルで何が必要かわかったら、コンソールで作成して編集するか、Amazon Managed Service for Prometheus コンソールまたは を使用して既存のファイルをアップロードできます AWS CLI。

> **Note**
>
> Amazon EKS クラスターを実行している場合は、 AWS Controllers for Kubernetes を使用してアラートマネージャー設定ファイルをアップロードすることもできます。

Amazon Managed Service for Prometheus コンソールを使用してアラートマネージャーの設定を編集または置き換えるには

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. ワークスペースのワークスペース ID を選択し、[アラートマネージャー] タブを選択します。
4. ワークスペースにまだアラートマネージャーの定義がない場合は、[定義を追加] を選択します。

   > **Note**
   >
   > ワークスペースに置換するアラートマネージャー定義がある場合は、代わりに変更を選択します。

5. [ファイルを選択] を選択し、アラートマネージャー定義ファイルを選択して、[続行] を選択します。

   > **Note**
   >
   > または、定義の作成オプションを選択して、新しいファイルを作成し、コンソールで直接編集することもできます。これにより、アップロード前に編集するサンプルデフォルト設定が作成されます。

を使用してアラートマネージャー設定をワークスペースに初めて AWS CLI アップロードするには

1. アラートマネージャーファイルの内容を base64 でエンコードします。Linux では、次のコマンドを使用できます。

   ```
   base64 input-file output-file
   ```

macOS では、次のコマンドを使用できます。

```
openssl base64 input-file output-file
```

2. 以下のいずれかのコマンドを入力して、ファイルをアップロードします。

   AWS CLI バージョン 2 では、次のように入力します。

   ```
   aws amp create-alert-manager-definition --data file://path_to_base_64_output_file
    --workspace-id my-workspace-id --region region
   ```

   AWS CLI バージョン 1 では、次のように入力します。

   ```
   aws amp create-alert-manager-definition --data fileb://path_to_base_64_output_file
    --workspace-id my-workspace-id --region region
   ```

3. アラートマネージャーの設定が有効になるまで数秒かかります。ステータスを確認するには、次のコマンドを入力します。

   ```
   aws amp describe-alert-manager-definition --workspace-id workspace_id --
   region region
   ```

   `status` が `ACTIVE` であれば、新しいアラートマネージャーの定義が有効になっています。

を使用してワークスペースのアラートマネージャー設定を新しい設定 AWS CLI に置き換えるには

1. アラートマネージャーファイルの内容を base64 でエンコードします。Linux では、次のコマンドを使用できます。

   ```
   base64 input-file output-file
   ```

   macOS では、次のコマンドを使用できます。

   ```
   openssl base64 input-file output-file
   ```

2. 以下のいずれかのコマンドを入力して、ファイルをアップロードします。

   AWS CLI バージョン 2 では、次のように入力します。

```
aws amp put-alert-manager-definition --data file://path_to_base_64_output_file --
workspace-id my-workspace-id --region region
```

AWS CLI バージョン 1 では、次のように入力します。

```
aws amp put-alert-manager-definition --data fileb://path_to_base_64_output_file --
workspace-id my-workspace-id --region region
```

3. 新しいアラートマネージャーの設定が有効になるまで数秒かかります。ステータスを確認するには、次のコマンドを入力します。

```
aws amp describe-alert-manager-definition --workspace-id workspace_id --
region region
```

`status` が `ACTIVE` であれば、新しいアラートマネージャーの定義が有効になっています。それまでは、以前のアラートマネージャーの設定が有効なままになります。

アラートを Amazon Managed Grafana またはオープンソース Grafana と統合する

Amazon Managed Service for Prometheus 内の Alertmanager で作成したアラートルールは、Amazon Managed Grafana や Grafana に転送して表示することができます。これにより、アラートルールとアラートを単一の環境に統合できます。Amazon Managed Grafana 内で、アラートルールと生成されたアラートを表示できます。

前提条件

Amazon Managed Service for Prometheus を Amazon Managed Grafana に統合する前に、以下の前提条件が満たされている必要があります。

- Amazon Managed Service for Prometheus AWS アカウント および IAM ロールをプログラムで作成するには、既存の および IAM 認証情報が必要です。

  AWS アカウント および IAM 認証情報の作成の詳細については、「」を参照してくださいセットアップする AWS。
- Amazon Managed Service for Prometheus ワークスペースがあり、そこにデータが取り込まれている必要があります。新しいワークスペースをセットアップするには、「Amazon Managed

Service for Prometheus ワークスペースの作成」を参照してください。また、Alertmanager やルーラーなどの Prometheus の概念を理解しておく必要もあります。これらのトピックの詳細については、Prometheus のドキュメントを参照してください。

- Amazon Managed Service for Prometheus で、Alertmanager の設定とルールファイルが既に構成されている必要があります。Amazon Managed Service for Prometheus での Alertmanager の詳細については、「アラートマネージャーによる Amazon Managed Service for Prometheus でのアラートの管理と転送」を参照してください。ルールの詳細については、「ルールを使用して、受信時にメトリクスを変更またはモニタリングする」を参照してください。
- Amazon Managed Grafana がセットアップされているか、オープンソースバージョンの Grafana が実行されている必要があります。
  - Amazon Managed Grafana を使用する場合は、Grafana アラートを使用している必要があります。詳細については、「Migrating legacy dashboard alerts to Grafana alerting」を参照してください。
  - オープンソースバージョンの Grafana を使用する場合は、バージョン 9.1 以降を実行している必要があります。

> **Note**
>
> 以前のバージョンの Grafana を使用することもできますが、統合アラート (Grafana アラート) 機能を有効にする必要があります。また、Grafana から Amazon Managed Service for Prometheus を呼び出すには sigv4 プロキシのセットアップが必要になる場合があります。詳細については、「Amazon Managed Service for Prometheus で使用する Grafana オープンソースまたは Grafana Enterprise のセットアップ」を参照してください。

- 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、https://docs.aws.amazon.com/grafana/latest/userguide/AMG-manage-permissions.html 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 Grafana のセットアップ

既に Amazon Managed Service for Prometheus インスタンスでルールとアラートが設定されている場合、Amazon Managed Grafana をそれらのアラートのダッシュボードとして使用するための構成は、すべて Amazon Managed Grafana 内で完結します。

Amazon Managed Grafana をアラートのダッシュボードとして構成するには

1. ワークスペースの Grafana コンソールを開きます。
2. [設定] で、[データソース] を選択します。
3. Prometheus データソースを作成するか開きます。まだ Prometheus データソースを設定していない場合、詳細については「ステップ 2: Grafana に Prometheus データソースを追加する」を参照してください。
4. Prometheus データソースで、[Alertmanager UI を使用してアラートを管理] を選択します。
5. [データソース] インターフェイスに戻ります。
6. 新しい Alertmanager データソースを作成します。
7. Alertmanager データソースの設定ページで、次の設定を追加します。

   - [実装] を `Prometheus` に設定します。
   - [URL] 設定には、Prometheus ワークスペースの URL を使用し、ワークスペース ID 以降の文字をすべて削除して、末尾に `/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`* のように指定します。
   - [認証] で、[SigV4Auth] をオンにします。これにより、リクエストに AWS 認証を使用するように Grafana に指示します。
   - [SigV4Auth の詳細] で、[デフォルトのリージョン] に Prometheus インスタンスのリージョンを指定します。例えば、`us-east-1` を指定します。
   - [デフォルト] オプションを `true` に設定します。
8. [保存してテスト] を選択します。
9. これで、Amazon Managed Service for Prometheus のアラートが Grafana インスタンスと連携するように構成されました。Amazon Managed Service for Prometheus インスタンスのアラートルール、アラートグループ (アクティブなアラートを含む)、サイレンスが、Grafana の [アラート] ページに表示されることを確認します。

CloudWatch Logs を使用したアラートマネージャーのトラブルシューティング

CloudWatch ログによる Amazon Managed Service for Prometheus イベントのモニタリング を使用すると、アラートマネージャーとルーラーに関する問題のトラブルシューティングを行うことができます。このセクションには、アラートマネージャー関連のトラブルシューティングトピックが含まれています。

トピック

- 空のコンテンツに関する警告
- 非 ASCII 文字に関する警告
- 無効な key/value に関する警告
- メッセージの制限に関する警告
- リソースベースのポリシーがないことによるエラー
- KMS を呼び出す権限がありません

空のコンテンツに関する警告

ログに次の警告が含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Message has been modified because the content was empty."
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

これは、アラートマネージャーテンプレートにより、送信アラートが空のメッセージに解決されたことを意味します。

実行するアクション

アラートマネージャーのテンプレートを検証し、すべてのレシーバーのパスに有効なテンプレートがあることを確認します。

非 ASCII 文字に関する警告

ログに次の警告が含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Subject has been modified because it contains control or non-ASCII
 characters."
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

これは、件名に非 ASCII 文字が含まれていることを意味します。

実行するアクション

テンプレートの件名フィールドから、非 ASCII 文字を含む可能性のあるラベルへの参照を削除します。

無効な key/value に関する警告

ログに次の警告が含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "MessageAttributes has been removed because of invalid key/value,
 numberOfRemovedAttributes=1"
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

これは、キーや値が無効なため、メッセージ属性の一部が削除されたことを意味します。

実行するアクション

メッセージ属性の設定に使用しているテンプレートを再評価し、有効な SNS メッセージ属性に解決されることを確認します。Amazon SNS トピックに送信するメッセージの検証の詳細については、「SNS トピックの検証」を参照してください。

メッセージの制限に関する警告

ログに次の警告が含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Message has been truncated because it exceeds size limit,
 originSize=266K, truncatedSize=12K"
       "level": "WARN"
   },
   "component": "alertmanager"
}
```

これは、メッセージサイズの一部が大きすぎることを意味します。

実行するアクション

アラートレシーバーのメッセージテンプレートを確認し、サイズ制限に収まるように変更します。

リソースベースのポリシーがないことによるエラー

ログに次のエラーが含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Notify for alerts failed, AMP is not authorized to perform: SNS:Publish
 on resource: arn:aws:sns:us-west-2:12345:testSnsReceiver because no resource-based
 policy allows the SNS:Publish action"
       "level": "ERROR"
   },
   "component": "alertmanager"
}
```

これは、指定された SNS トピックにアラートを送信するためのアクセス許可が Amazon Managed Service for Prometheus にないことを意味します。

実行するアクション

トピックに SNS メッセージを送信する許可が、Amazon SNS トピックのアクセスポリシーによって Amazon Managed Service for Prometheus に与えられていることを確認します。サービス

aps.amazonaws.com (Amazon Managed Service for Prometheus) に Amazon SNS トピックへのアクセスを許可する Amazon SNS アクセスポリシーを作成します。SNS アクセスポリシーの詳細については、「Amazon Simple Notification Service デベロッパーガイド」の「アクセスポリシー言語の使用」および「Amazon SNS アクセスコントロールの例」を参照してください。 Amazon SNS

KMS を呼び出す権限がありません

ログに次の AWS KMS エラーが含まれている場合

```
{
   "workspaceId": "ws-abcd1234-ef56-78ab-cd90-1234abcd0000",
   "message": {
       "log": "Notify for alerts failed, AMP is not authorized to call KMS",
       "level": "ERROR"
   },
   "component": "alertmanager"
}
```

実行するアクション

Amazon SNS トピックの暗号化に使用されるキーのキーポリシーで、Amazon Managed Service for Prometheus サービスプリンシパルが kms:GenerateDataKey*、、および のアクションを実行aps.amazonaws.comできることを確認しますkms:Decrypt。詳細については、「SNS トピックに対するAWS KMS アクセス許可」を参照してください。

Amazon Managed Service for Prometheus ワークスペースのログ記録とモニタリング

Amazon Managed Service for Prometheus は、Amazon CloudWatch を使用してオペレーションに関するデータを提供します。 CloudWatch メトリクスを使用して、リソースの使用状況と Amazon Managed Service for Prometheus ワークスペースへのリクエストを確認できます。 CloudWatch Logs サポートを有効にして、ワークスペースで発生したイベントのログを取得できます。

以下のトピック CloudWatch では、 の使用について詳しく説明します。

CloudWatch メトリクスを使用して Amazon Managed Service for Prometheus リソースをモニタリングする

Amazon Managed Service for Prometheus は、使用状況メトリクスを に供給します CloudWatch。これらのメトリクスにより、ワークスペースの使用状況が可視化されます。提供されたメトリクスは、 の AWS/Usageおよび AWS/Prometheus名前空間にあります CloudWatch。これらのメトリクスは で CloudWatch無料で利用できます。使用状況メトリクスの詳細については、「CloudWatch 使用状況メトリクス」を参照してください。

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
ResourceCount	IngestionRate	AWS/Usage	サンプルの取り込みレート 単位: 1 秒あたりのカウント 有効な統計: Average、Minimum、Maximum、Sum
ResourceCount	ActiveSeries	AWS/Usage	ワークスペースごとのアクティブなシリーズの数 単位: 数

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
			有効な統計: Average、Minimum、Maximum、Sum
ResourceCount	ActiveAlerts	AWS/Usage	ワークスペースごとのアクティブなアラートの数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
ResourceCount	SizeOfAlerts	AWS/Usage	ワークスペース内のすべてのアラートの合計サイズ、バイト単位 単位: バイト 有効な統計: Average、Minimum、Maximum、Sum
ResourceCount	SuppressedAlerts	AWS/Usage	ワークスペースごとの抑制状態にあるアラートの数。アラートは、無音や禁止にすることで抑制できます。 単位: 数 有効な統計: Average、Minimum、Maximum、Sum

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
ResourceCount	UnprocessedAlerts	AWS/Usage	ワークスペースごとの未処理状態のアラートの数。アラートは、 によって受信されると未処理の状態になりますが AlertManager、次の集約グループ評価を待っています。 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
ResourceCount	AllAlerts	AWS/Usage	ワークスペースごとのすべての状態のアラート数。 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
AlertManagerAlerts Received	-	AWS/Prometheus	アラートマネージャーによって正常に受信されたアラートの合計数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
AlertManagerNotifications Failed	-	AWS/Prometheus	失敗したアラート配信の数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
AlertManagerNotificationsThrottled	-	AWS/Prometheus	スロットリングされたアラートの数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
Discarded Samples*	-	AWS/Prometheus	破棄されたサンプルの数 (理由別) 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
RuleEvaluations	-	AWS/Prometheus	ルール評価の合計数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum
RuleEvaluationFailures	-	AWS/Prometheus	特定の間隔におけるルール評価の失敗の数 単位: 数 有効な統計: Average、Minimum、Maximum、Sum

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
RuleGroup IterationsMissed	-	AWS/Prometheus	特定の間隔における欠落したルールグループイテレーションの数。 単位: 数 有効な統計: Average、Minimum、Maximum、Sum

* サンプルが破棄される理由には、次のようなものがあります。

理由	意味
greater_than_max_sample_age	1 時間以上経過したサンプルを破棄します。
new-value-for-timestamp	重複したサンプルは、以前に記録されたものとは異なるタイムスタンプで送信されます。
per_metric_series_limit	ユーザーがメトリクスあたりのアクティブなシリーズの制限に達しました。
per_user_series_limit	ユーザーがアクティブなシリーズ制限の合計数に達しました。
rate_limited	取り込みレートには制限があります。
sample-out-of-order	サンプルは順不同で送信され、処理できません。
label_value_too_long	ラベルの値が文字数の制限を超えています。
max_label_names_per_series	ユーザーがメトリクスあたりのラベル名に達しました。
missing_metric_name	メトリクス名が指定されていません。
metric_name_invalid	無効なメトリクス名が指定されました。

理由	意味
label_invalid	無効なラベルが指定されました。
duplicate_label_names	重複するラベル名が指定されました。

> **Note**
>
> メトリクスがない場合は、そのメトリクスの値が 0 であることと同じ意味になります。

> **Note**
>
> `RuleGroupIterationsMissed`、`RuleEvaluations`、`RuleEvaluationFailures` には、次の構造の `RuleGroup` ディメンションがあります。
> *`RuleGroupNamespace`*;*`RuleGroup`*

Prometheus が販売したメトリクスに CloudWatch アラームを設定する

CloudWatch アラームを使用して Prometheus リソースの使用状況をモニタリングできます。

Prometheus ActiveSeriesの の数にアラームを設定するには

1. グラフ化されたメトリクスタブを選択し、ActiveSeriesラベルまで下にスクロールします。

   [グラフ化したメトリクス] ビューには、現在取り込まれているメトリクスのみが表示されます。
2. [アクション] 列の [通知] アイコンを選択します。
3. [メトリクスと条件の指定] で、[条件値] フィールドにしきい値の条件を入力し、[次へ] を選択します。
4. [アクションの設定] で、通知の送信先となる既存の SNS トピックを選択するか、新しいトピックを作成します。
5. [名前と説明を追加] に、アラームの名前と、必要に応じて説明を追加します。
6. [アラームを作成] を選択します。

CloudWatch ログによる Amazon Managed Service for Prometheus イベントのモニタリング

Amazon Managed Service for Prometheus は、アラートマネージャーとルーラーのエラーイベントと警告イベントを Amazon CloudWatch Logs のロググループに記録します。アラートマネージャーとルーラーの詳細については、このガイドの「アラートマネージャー」トピックを参照してください。ワークスペースログデータを CloudWatch Logs のログストリームに発行できます。モニタリングするログは、Amazon Managed Service for Prometheus コンソールまたは AWS CLIを使用して構成できます。これらのログは、 コンソールで表示またはクエリできます CloudWatch。コンソールでの CloudWatch ログログストリームの表示の詳細については、 CloudWatch ユーザーガイドの「 でのロググループとログストリームの操作 CloudWatch」を参照してください。

CloudWatch 無料利用枠では、最大 5Gb のログを CloudWatch Logs に発行できます。無料利用枠枠を超えたログは、CloudWatch 料金プラン に基づいて課金されます。

トピック

- CloudWatch ログの設定

CloudWatch ログの設定

Amazon Managed Service for Prometheus は、アラートマネージャーとルーラーのエラーイベントと警告イベントを Amazon CloudWatch Logs のロググループに記録します。

`create-logging-configuration` API リクエストを呼び出す AWS CLI ことで、Amazon Managed Service for Prometheus コンソールまたは で CloudWatch ログログ記録設定を設定できます。

前提条件

を呼び出す前に`create-logging-configuration`、 CloudWatch ログの設定に使用する ID またはロールに次のポリシーまたは同等のアクセス許可をアタッチします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
```

```
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups",
                "aps:CreateLoggingConfiguration",
                "aps:UpdateLoggingConfiguration",
                "aps:DescribeLoggingConfiguration",
                "aps:DeleteLoggingConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

CloudWatch ログを設定するには

Amazon Managed Service for Prometheus のログ記録は、 AWS コンソールまたは を使用して設定できます AWS CLI。

Console

Amazon Managed Service for Prometheus コンソールでログ記録を構成するには

1. ワークスペースの詳細パネルの [ログ] タブに移動します。
2. [ログ] パネルの右上にある [ログを管理] を選択します。
3. [ログレベル] ドロップダウンリストで、[すべて] を選択します。
4. [ロググループ] ドロップダウンリストで、ログを発行する先のロググループを選択します。

   CloudWatch コンソールで新しいロググループを作成することもできます。
5. [変更の保存] を選択します。

AWS CLI

ログ記録設定は、 を使用して設定できます AWS CLI。

を使用してログ記録を設定するには AWS CLI

- を使用して AWS CLI、次のコマンドを実行します。

```
aws amp create-logging-configuration --workspace-id my_workspace_ID
                                --log-group-arn my-log-group-arn
```

制限事項

- すべてのイベントが記録されるわけではない

  Amazon Managed Service for Prometheus は、`warning` または `error` レベルのイベントのみをログに記録します。
- ポリシーサイズの制限

  CloudWatch ログリソースポリシーは 5,120 文字に制限されています。 CloudWatch Logs は、ポリシーがこのサイズ制限に近づいていることを検出すると、 で始まるロググループを自動的に有効にします`/aws/vendedlogs/`。

  ログ記録を有効にしてアラートルールを作成する場合、Amazon Managed Service for Prometheus は指定したロググループで CloudWatch Logs リソースポリシーを更新する必要があります。 CloudWatch Logs リソースポリシーのサイズ制限に達しないようにするには、Logs CloudWatch ロググループ名の前に を付けます`/aws/vendedlogs/`。Amazon Managed Service for Prometheus コンソールでロググループを作成する場合は、ロググループ名に `/aws/vendedlogs/` プレフィックスが付けられます。詳細については、 CloudWatch 「 ログユーザーガイド」の「特定の AWS サービスからのログ記録の有効化」を参照してください。

Amazon Managed Service for Prometheus のコストを理解して最適化する

以下のよくある質問とその回答は、Amazon Managed Service for Prometheus に関連するコストを理解して最適化するために役立つ可能性があります。

コストに影響する要因は何ですか?

ほとんどのお客様にとって、コストの大部分はメトリクスの取り込みに由来します。クエリの使用量が多いお客様の場合、処理されたクエリサンプル数に基づくコストも発生しますが、メトリクスのストレージがコスト全体に及ぼす影響は小規模です。これらのそれぞれにかかる料金の詳細については、Amazon Managed Service for Prometheus の製品ページの「料金」を参照してください。

コストを削減する最善の方法は何ですか? どうすれば取り込みコストを下げることができますか?

ほとんどのお客様にとって、コストの大部分は取り込みレートです (メトリクスのストレージではありません)。取り込みレートを下げるには、収集頻度を低くする (収集間隔を大きくする) か、取り込むアクティブなシリーズの数を少なくします。

コレクションエージェントからコレクション (スクレイピング) 間隔を延長できます。Prometheus サーバー (エージェントモードで実行) と AWS Distro for OpenTelemetry (ADOT) コレクターの両方が`scrape_interval`設定をサポートします。例えば、収集間隔を 30 秒から 60 秒に増やすと、取り込みの使用量が半減します。

`<relabel_config>` を使用して、Amazon Managed Service for Prometheus に送信されるメトリクスをフィルタリングすることもできます。Prometheus エージェントの設定における再ラベル付けの詳細については、Prometheus ドキュメントの https://prometheus.io/docs/prometheus/latest/configuration/configuration/#relabel_config を参照してください。

クエリコストを削減する最善の方法は何ですか?

クエリコストは、処理されたサンプルの数に基づきます。クエリの頻度を低くすると、クエリコストを削減できます。

クエリコストに最も大きく影響しているクエリをより明確に把握するには、サポート担当者に連絡してチケットを提出してください。Amazon Managed Service for Prometheus チームが、コストに最も大きく影響しているクエリを特定できるように支援します。

メトリクスの保持期間を短くした場合、合計請求額の削減につながりますか?

保持期間を短縮することはできますが、それによってコストが大幅に削減される可能性はほとんどありません。

保持期間を短縮 (または延長) する必要がある場合は、`Retention time for ingested data` クォータを変更するサービス制限リクエストを提出できます。

アラートクエリのコストを低く抑えるにはどうすればよいですか？

アラートによりデータに対するクエリが作成され、クエリコストが増加します。アラートクエリを最適化し、コストを削減するために使用できる戦略をいくつか紹介します。

- Amazon Managed Service for Prometheus アラートの使用 – Amazon Managed Service for Prometheus の外部にあるアラートシステムには、複数のアベイラビリティーゾーンまたはリージョンからメトリクスをクエリするため、耐障害性または高可用性を追加するための追加のクエリが必要になる場合があります。これには、Grafana での高可用性に関するアラートが含まれます。これにより、コストが 3 倍以上増加する可能性があります。Amazon Managed Service for Prometheus のアラートは最適化されており、クエリ数を最小限に抑えながら高い可用性と回復性を提供します。

  外部アラートシステムではなく、Amazon Managed Service for Prometheus でネイティブアラートを使用することをお勧めします。
- アラート間隔の最適化 – アラートクエリを最適化する 1 つの簡単な方法は、自動更新間隔を長くすることです。1 分ごとにクエリを実行するアラートがあっても、5 分ごとにのみ必要である場合、自動更新間隔を長くすると、そのアラートのクエリコストが 5 倍節約される可能性があります。
- 最適なルックバックを使用する – クエリのルックバックウィンドウを大きくすると、より多くのデータを取得するため、クエリのコストが増加します。PromQL クエリのルックバックウィンドウが、アラートが必要なデータに対して適切なサイズであることを確認します。例えば、次のルールでは、式に 10 分のルックバックウィンドウが含まれます。

```
  - alert: metric:alerting_rule
    expr: avg(rate(container_cpu_usage_seconds_total[10m])) > 0
    for: 2m
```

`expr` を に変更`avg(rate(container_cpu_usage_seconds_total[5m])) > 0`すると、クエリコストを削減できます。

一般的に、アラートルールを確認し、サービスに最適なメトリクスをアラートしていることを確認します。特に時間の経過とともにアラートを追加する場合、同じメトリクスまたは同じ情報を提供する複数のアラートに重複するアラートを簡単に作成できます。アラートのグループが同時に発生することがよくある場合は、アラートを最適化し、すべてを含めることはできません。

これらの提案は、コスト削減に役立ちます。最終的には、システムの状態を理解するための適切なアラートセットの作成とコストのバランスを取る必要があります。

Amazon Managed Service for Prometheus でのアラートの詳細については、「」を参照してくださいアラートマネージャーによる Amazon Managed Service for Prometheus でのアラートの管理と転送。

コストのモニタリングにはどのようなメトリクスを使用できますか?

Amazon `IngestionRate`で をモニタリング CloudWatch して、取り込みコストを追跡します。での Amazon Managed Service for Prometheus メトリクスのモニタリングの詳細についてはCloudWatch、「」を参照してください CloudWatch メトリクスを使用して Amazon Managed Service for Prometheus リソースをモニタリングする。

請求書はいつでも確認できますか?

は AWS 使用状況 AWS Cost and Usage Report を追跡し、請求期間内にアカウントに関連する推定請求額を提供します。詳細については、「 AWS コストと使用状況レポートユーザーガイド」のAWS 「 コストと使用状況レポートとは」を参照してください。

月初めの請求額が月末よりも高いのはなぜですか?

Amazon Managed Service for Prometheus では、取り込み量に応じた階層型の価格モデルが採用されているため、初期の使用量のコストの方が高い結果となります。使用量が上位の取り込み階層に達するにつれて、コストが下がります。取り込み階層を含む料金の詳細については、Amazon Managed Service for Prometheus の製品ページの「料金」を参照してください。

> **Note**
>
> - 階層はリージョン 内で使用するためのもので、リージョン間では使用できません。より低いレートを使用するには、リージョン内の使用量が次の階層に到達する必要があります。
> - の組織では AWS Organizations、階層の使用状況は、アカウントごとではなく、支払者アカウントごとに集計されます (支払者アカウントは常に組織管理アカウントです) 。組織内のすべてのアカウントで取り込まれたメトリクスの合計 (リージョン内) が次の階層に達すると、すべてのアカウントにより低い料金が課金されます。

Amazon Managed Service for Prometheus ワークスペースをすべて削除しましたが、まだ課金されているようです。何が起こっている可能性がありますか?

この場合の 1 つの可能性は、削除したワークスペースにメトリクスを送信するように設定された AWS マネージドスクレイパーがまだあることです。「」の手順に従いますスクレイパーの検出と削除。

他の AWS サービスとの統合

Amazon Managed Service for Prometheus は、他の AWS サービスと統合されます。このセクションでは、Amazon Elastic Kubernetes Service (Amazon EKS) のコストモニタリング (Kubecost を使用) との統合と、Amazon Data Firehose を使用して から CloudWatchメトリクスを取り込む方法について説明します。また、 AWS Observability Accelerator Terraform モジュールを使用するか、 AWS Controllers for Kubernetes を使用して Amazon Managed Service for Prometheus をセットアップおよび管理する方法についても説明します。

トピック

- Amazon EKS コストモニタリングとの統合
- Observability Accelerator で Amazon Managed Service for Prometheus AWS をセットアップする
- Controllers for Kubernetes で Amazon Managed Service for AWS Prometheus を管理する
- CloudWatch メトリクスと Amazon Managed Service for Prometheus の統合

Amazon EKS コストモニタリングとの統合

Amazon Managed Service for Prometheus は Amazon Elastic Kubernetes Service (Amazon EKS) のコストモニタリング (Kubecost を使用) と統合され、コスト配分を計算し、Kubernetes クラスターの最適化に関するインサイトを提供します。Amazon Managed Service for Prometheus を Kubecost と共に使用すると、信頼性の高い方法でコストモニタリングをスケールして、より大規模なクラスターをサポートできます。

Kubecost との統合により、Amazon EKS クラスターのコストが細かく可視化されます。コンテナレベルからクラスターレベル、さらにはマルチクラスターレベルまで、Kubernetes コンテキストの大部分でコストを集計できます。複数のコンテナやクラスターにまたがるレポートを生成して、ショーバックまたはチャージバックの目的でコストを追跡できます。

単一クラスターまたはマルチクラスターのシナリオで Kubecost と統合する手順を以下に示します。

- 単一クラスター統合 - Amazon EKS コストモニタリングを単一のクラスターと統合する方法については、AWS ブログ記事の「Integrating Kubecost with Amazon Managed Service for Prometheus」を参照してください。
- マルチクラスター統合 - Amazon EKS コストモニタリングを複数のクラスターと統合する方法については、AWS ブログ記事の「Multi-cluster cost monitoring for Amazon EKS using Kubecost and Amazon Managed Service for Prometheus」を参照してください。

> **Note**
>
> Kubecost の使用方法の詳細については、「Amazon EKS ユーザーガイド」の「コストモニタリング」を参照してください。

Observability Accelerator で Amazon Managed Service for Prometheus AWS をセットアップする

AWS は、Amazon Elastic Kubernetes Service (Amazon EKS) プロジェクトのモニタリング、ログ記録、アラート、ダッシュボードなどのオブザーバビリティツールを提供します。これには、Amazon Managed Service for Prometheus、Amazon Managed Grafana、 AWS Distro for OpenTelemetry、およびその他のツールが含まれます。これらのツールの連携を支援するために、 AWS では、これらのサービスでオブザーバビリティを構成するための AWS Observability Accelerator と呼ばれる Terraform モジュールを提供しています。

AWS Observability Accelerator は、インフラストラクチャ、NGINX デプロイメント、およびその他のシナリオをモニタリングするための例を提供します。このセクションでは、Amazon EKS クラスターのインフラストラクチャをモニタリングする例を示します。

Terraform テンプレートと詳細な手順については、「AWS Observability Accelerator for Terraform GitHub 」ページを参照してください。AWS Observability Accelerator を発表するブログ記事を読むこともできます。

前提条件

AWS Observability Accelerator を使用するには、既存の Amazon EKS クラスターと以下の前提条件が必要です。

- AWS CLI – コマンドラインから AWS 機能を呼び出すために使用されます。
- kubectl - コマンドラインから EKS クラスターを制御するために使用されます。
- Terraform - このソリューションのリソースの作成を自動化するために使用されます。 AWS アカウント内で Amazon Managed Service for Prometheus、Amazon Managed Grafana、IAM を作成および管理するためのアクセス権を持つ IAM ロールを使用して AWS プロバイダーを設定する必要があります。Terraform の AWS プロバイダーを設定する方法の詳細については、Terraform ドキュメントAWSの「 provider」を参照してください。

インフラストラクチャモニタリングのサンプルの使用

AWS Observability Accelerator には、付属の Terraform モジュールを使用して Amazon EKS クラスターのオブザーバビリティをセットアップおよび設定するサンプルテンプレートが用意されています。この例は、 AWS Observability Accelerator を使用してインフラストラクチャのモニタリングを設定する方法を示しています。このテンプレートの使用とそれに含まれる追加機能の詳細については、「」のAWS 「オブザーバビリティアクセラレーターのベースとインフラストラクチャのモニタリングページを含む既存のクラスター」を参照してください GitHub。

インフラストラクチャモニタリングの Terraform モジュールを使用するには

1. プロジェクトを作成するフォルダーから、次のコマンドを使用してリポジトリをクローンします。

```
git clone https://github.com/aws-observability/terraform-aws-observability-
accelerator.git
```

2. 次のコマンドを実行して Terraform を初期化します。

```
cd examples/existing-cluster-with-base-and-infra

terraform init
```

3. 新しい `terraform.tfvars` ファイルを作成し、以下の例を記述します。Amazon EKS クラスターの AWS リージョンとクラスター ID を使用します。

```
# (mandatory) AWS Region where your resources will be located
aws_region = "eu-west-1"

# (mandatory) EKS Cluster name
eks_cluster_id = "my-eks-cluster"
```

4. 使用する Amazon Managed Grafana ワークスペースがない場合は作成します。新しいワークスペースを作成する方法については、「Amazon Managed Grafana User Guide」の「Create your first workspace」を参照してください。
5. コマンドラインで次のコマンドを実行して、Terraform で Grafana ワークスペースを使用するために必要な 2 つの変数を作成します。を Grafana ワークスペース*grafana-workspace-id*の ID に置き換える必要があります。

```
export TF_VAR_managed_grafana_workspace_id=grafana-workspace-id
```

```
export TF_VAR_grafana_api_key=`aws grafana create-workspace-api-key --key-name
 "observability-accelerator-$(date +%s)" --key-role ADMIN --seconds-to-live 1200 --
workspace-id $TF_VAR_managed_grafana_workspace_id --query key --output text`
```

6. 〔オプション] 既存の Amazon Managed Service for Prometheus ワークスペースを使用するには、次の例のように ID を `terraform.tfvars` ファイルに追加し、 を Prometheus ワークスペース ID に置き換え*prometheus-workspace-id*ます。既存のワークスペースを指定しない場合は、新しい Prometheus ワークスペースが自動的に作成されます。

```
# (optional) Leave it empty for a new workspace to be created
managed_prometheus_workspace_id = "prometheus-workspace-id"
```

7. 次のコマンドを使用してソリューションをデプロイします。

```
terraform apply -var-file=terraform.tfvars
```

これにより、以下を含むリソースが AWS アカウントに作成されます。

- 新しい Amazon Managed Service for Prometheus ワークスペース (既存のワークスペースを使用する場合を除く)。
- Prometheus ワークスペース内のアラートマネージャーの設定、アラート、ルール。
- Amazon Managed Grafana の現在のワークスペース内の新しいデータソースとダッシュボード。データソースは `aws-observability-accelerator` という名前になります。ダッシュボードは [Observability Accelerator ダッシュボード] の下に一覧表示されます。
- 提供された Amazon EKS クラスターに AWS Distro for OpenTelemetry演算子がセットアップされ、Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信します。

新しいダッシュボードを表示するには、Amazon Managed Grafana ワークスペースでそのダッシュボードを開きます。Amazon Managed Grafana の使用方法の詳細については、「Amazon Managed Grafana User Guide」の「Working in your Grafana workspace」を参照してください。

Controllers for Kubernetes で Amazon Managed Service for AWS Prometheus を管理する

Amazon Managed Service for Prometheus は、AWS Controllers for Kubernetes (ACK) と統合され、Amazon EKS のワークスペース、アラートマネージャー、ルーラーリソースの管理をサポート

します。 AWS Controllers for Kubernetes カスタムリソース定義 (CRDsとネイティブ Kubernetes オブジェクトを使用できます。クラスターの外部にあるリソースは定義する必要はありません。

このセクションでは、既存の Amazon EKS クラスターで AWS Controllers for Kubernetes と Amazon Managed Service for Prometheus を設定する方法について説明します。

AWS Controllers for Kubernetes と Amazon Managed Service for Prometheus の ACK コントローラーを紹介するブログ記事も読むことができます。

前提条件

AWS Controllers for Kubernetes と Amazon Managed Service for Prometheus を Amazon EKS クラスターと統合する前に、次の前提条件が必要です。

- Amazon Managed Service for Prometheus AWS アカウント および IAM ロールをプログラムで作成するには、既存の および アクセス許可が必要です。
- OpenID Connect (OIDC) を有効にした既存の Amazon EKS クラスターが必要です。

  OIDC が有効でない場合、次のコマンドを使用して有効にすることができます。*YOUR_CLUSTER_NAME* と *AWS_REGION* は、アカウントに応じた適切な値に置き換えてください。

  ```
  eksctl utils associate-iam-oidc-provider \
      --cluster ${YOUR_CLUSTER_NAME} --region ${AWS_REGION} \
      --approve
  ```

  Amazon EKS で OIDC を使用する方法の詳細については、「Amazon EKS ユーザーガイド」の「OpenID Connect アイデンティティプロバイダーからクラスターのユーザーを認証する」と「クラスター用の IAM OIDC プロバイダーの作成」を参照してください。
- Amazon EKS クラスターに Amazon EBS CSI ドライバーがインストールされている必要があります。
- AWS CLI がインストールされている必要があります。 AWS CLI は、コマンドラインから AWS 機能を呼び出すために使用されます。
- Kubernetes のパッケージマネージャーである Helm がインストールされている必要があります。
- Amazon EKS クラスターで、Prometheus のコントロールプレーンメトリクスがセットアップされている必要があります。

- 新しいワークスペースからのアラートの送信先となる Amazon Simple Notification Service (Amazon SNS) トピックが必要です。トピックにメッセージを送信するためのアクセス許可が Amazon Managed Service for Prometheus に付与されていることを確認してください。

Amazon EKS クラスターが正しく構成されたら、`kubectl get --raw /metrics` を呼び出して、Prometheus 用にフォーマットされたメトリクスを確認できます。これで、 AWS Controllers for Kubernetes サービスコントローラーをインストールし、それを使用して Amazon Managed Service for Prometheus リソースをデプロイする準備が整いました。

AWS Controllers for Kubernetes を使用したワークスペースのデプロイ

新しい Amazon Managed Service for Prometheus ワークスペースをデプロイするには、 AWS Controllers for Kubernetes コントローラーをインストールし、それを使用してワークスペースを作成します。

AWS Controllers for Kubernetes を使用して新しい Amazon Managed Service for Prometheus ワークスペースをデプロイするには

1. 以下のコマンドを実行し、Helm を使用して Amazon Managed Service for Prometheus サービスコントローラーをインストールします。詳細については、 の Controllers for Kubernetes ドキュメントの AWS 「ACK Controller をインストールする」を参照してください GitHub。*region* には、`us-east-1` など、システムに適したリージョンを使用してください。

   ```
   export SERVICE=prometheusservice
   export RELEASE_VERSION=`curl -sL https://api.github.com/repos/aws-controllers-k8s/
   $SERVICE-controller/releases/latest | grep '"tag_name":' | cut -d'"' -f4`
   export ACK_SYSTEM_NAMESPACE=ack-system
   export AWS_REGION=region

   aws ecr-public get-login-password --region us-east-1 | helm registry login --
   username AWS --password-stdin public.ecr.aws
   helm install --create-namespace -n $ACK_SYSTEM_NAMESPACE ack-$SERVICE-controller \
     oci://public.ecr.aws/aws-controllers-k8s/$SERVICE-chart --version=
   $RELEASE_VERSION --set=aws.region=$AWS_REGION
   ```

   しばらくすると、成功を示す次のようなレスポンスが表示されます。

   ```
   You are now able to create Amazon Managed Service for Prometheus (AMP) resources!
   The controller is running in "cluster" mode.
   ```

```
The controller is configured to manage AWS resources in region: "us-east-1"
```

オプションで、次のコマンドを使用して AWS Controllers for Kubernetes コントローラーが正常にインストールされたことを確認できます。

```
helm list --namespace $ACK_SYSTEM_NAMESPACE -o yaml
```

これにより、コントローラー ack-prometheusservice-controller に関する情報 (status: deployed など) が返されます。

2. workspace.yaml という名前のファイルを作成し、次のテキストを記述します。これは、作成するワークスペースの設定として使用されます。

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: Workspace
metadata:
  name: my-amp-workspace
spec:
  alias: my-amp-workspace
  tags:
    ClusterName: EKS-demo
```

3. 次のコマンドを実行してワークスペースを作成します (このコマンドでは、ステップ 1 で設定したシステム変数が使用されます)。

```
kubectl apply -f workspace.yaml -n $ACK_SYSTEM_NAMESPACE
```

しばらくすると、アカウントに my-amp-workspace という新しいワークスペースが表示されます。

次のコマンドを実行して、ワークスペース ID などのワークスペースの詳細とステータスを確認します。または、Amazon Managed Service for Prometheus コンソールで新しいワークスペースを確認することもできます。

```
kubectl describe workspace my-amp-workspace -n $ACK_SYSTEM_NAMESPACE
```

> **Note**
>
> 新しいワークスペースを作成する代わりに、既存のワークスペースを使用することもできます。

4. Rulegroups の設定として 2 つの新しい yaml ファイルを作成し、次の設定を使用して次に AlertManager 作成します。

   次の設定を `rulegroup.yaml` として保存します。*`WORKSPACE-ID`* は、前のステップで確認したワークスペース ID に置き換えます。

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: RuleGroupsNamespace
metadata:
  name: default-rule
spec:
  workspaceID: WORKSPACE-ID
  name: default-rule
  configuration: |
    groups:
    - name: example
      rules:
      - alert: HostHighCpuLoad
        expr: 100 - (avg(rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) > 60
        for: 5m
        labels:
          severity: warning
          event_type: scale_up
        annotations:
          summary: Host high CPU load (instance {{ $labels.instance }})
          description: "CPU load is > 60%\n  VALUE = {{ $value }}\n  LABELS =
 {{ $labels }}"
      - alert: HostLowCpuLoad
        expr: 100 - (avg(rate(node_cpu_seconds_total{mode="idle"}[2m])) * 100) < 30
        for: 5m
        labels:
          severity: warning
          event_type: scale_down
        annotations:
          summary: Host low CPU load (instance {{ $labels.instance }})
```

```
          description: "CPU load is < 30%\n  VALUE = {{ $value }}\n  LABELS =
 {{ $labels }}"
```

次の設定を alertmanager.yaml として保存します。*WORKSPACE-ID* は、前のステップで確認したワークスペース ID に置き換えます。*TOPIC-ARN* を通知を送信する Amazon SNS トピックの ARN に、*REGION* を使用中の に置き換え AWS リージョン ます。Amazon Managed Service for Prometheus に、Amazon SNS トピックへのアクセス許可が必要であることを忘れないでください。

```
apiVersion: prometheusservice.services.k8s.aws/v1alpha1
kind: AlertManagerDefinition
metadata:
  name: alert-manager
spec:
  workspaceID: WORKSPACE-ID
  configuration: |
    alertmanager_config: |
      route:
         receiver: default_receiver
      receivers:
        - name: default_receiver
          sns_configs:
          - topic_arn: TOPIC-ARN
            sigv4:
              region: REGION
            message: |
              alert_type: {{ .CommonLabels.alertname }}
              event_type: {{ .CommonLabels.event_type }}
```

> **Note**
>
> これらの設定ファイルの形式の詳細については、RuleGroupsNamespaceData「」および「」を参照してくださいAlertManagerDefinitionData。

5. 次のコマンドを実行して、ルールグループとアラートマネージャーの設定を作成します (このコマンドでは、ステップ 1 で設定したシステム変数が使用されます)。

```
kubectl apply -f rulegroup.yaml -n $ACK_SYSTEM_NAMESPACE
kubectl apply -f alertmanager.yaml -n $ACK_SYSTEM_NAMESPACE
```

しばらくすると変更が有効になります。

> **Note**
>
> リソースを作成するのではなく更新する場合は、yaml ファイルを更新し、kubectl apply コマンドを再実行するだけです。
> リソースを削除するには、次のコマンドを実行します。
> を、、Workspace、AlertManagerDefinitionまたは を削除するリソースのタイプ*ResourceType*に置き換えますRuleGroupNamespace。を削除するリソースの名前*ResourceName*に置き換えます。
>
> ```
> kubectl delete ResourceType ResourceName -n $ACK_SYSTEM_NAMESPACE
> ```

これで、新しいワークスペースのデプロイは完了です。次のセクションでは、このワークスペースにメトリクスを送信するようにクラスターを構成する方法を説明します。

Amazon Managed Service for Prometheus ワークスペースに書き込むための Amazon EKS クラスターの構成

このセクションでは、Helm を使用して、Amazon EKS クラスターで実行されている Prometheus を構成し、前のセクションで作成した Amazon Managed Service for Prometheus ワークスペースへのメトリクスのリモートで書き込みを行う方法について説明します。

この手順では、メトリクスの取り込みに使用するために作成した IAM ロールの名前が必要です。まだ作成していない場合は、「Amazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定」を参照して、詳細と手順を確認してください。これらの手順に従うと、amp-iamproxy-ingest-role という IAM ロールが作成されます。

Amazon EKS クラスターをリモート書き込み用に構成するには

1. 次のコマンドを使用して、ワークスペースの prometheusEndpoint を取得します。*WORKSPACE-ID* は、前のセクションで確認したワークスペース ID に置き換えます。

   ```
   aws amp describe-workspace --workspace-id WORKSPACE-ID
   ```

   返される結果には prometheusEndpoint が含まれ、次のような形式になります。

```
https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-a1b2c3d4-a123-b456-
c789-ac1234567890/
```

後のステップで使用するために、この URL を保存しておきます。

2. 次のテキストで新しいファイルを作成し、prometheus-config.yaml という名前を付けます。*account* は自分のアカウント ID に、*workspaceURL/* は先ほど確認した URL に、*region* はシステムの適切な AWS リージョン に置き換えます。

```
serviceAccounts:
        server:
            name: "amp-iamproxy-ingest-service-account"
            annotations:
                eks.amazonaws.com/role-arn: "arn:aws:iam::account:role/amp-
iamproxy-ingest-role"
server:
    remoteWrite:
        - url: workspaceURL/api/v1/remote_write
          sigv4:
            region: region
          queue_config:
            max_samples_per_send: 1000
            max_shards: 200
            capacity: 2500
```

3. 次の Helm コマンドを使用して、Prometheus のチャート名、名前空間の名前、チャートのバージョンを確認します。

```
helm ls --all-namespaces
```

ここまでの手順に基づくと、Prometheus チャートと名前空間にはどちらも prometheus という名前が付いていて、チャートのバージョンは 15.2.0 のようになります。

4. 前のステップで*PrometheusChartVersion*確認した *PrometheusChartName*、*PrometheusNamespace*、および を使用して、次のコマンドを実行します。

```
helm upgrade PrometheusChartName prometheus-community/prometheus -
n PrometheusNamespace -f prometheus-config.yaml --version PrometheusChartVersion
```

数分後に、アップグレードが成功したことを示すメッセージが表示されます。

5. 必要に応じて、awscurl を使用して Amazon Managed Service for Prometheus エンドポイントにクエリを実行して、メトリクスが正常に送信されていることを確認します。*Region* AWS リージョン を使用中の に、*workspaceURL* をステップ 1 で見つけた URL に置き換えます。

```
awscurl --service="aps" --region="Region" "workspaceURL/api/v1/query?
query=node_cpu_seconds_total"
```

これで、YAML ファイルを設定として使用して、Amazon Managed Service for Prometheus ワークスペースを作成し、そのワークスペースに Amazon EKS クラスターから接続することができました。これらのファイルはカスタムリソース定義 (CRD) と呼ばれ、Amazon EKS クラスター内に配置されます。 AWS Controllers for Kubernetes コントローラーを使用して、Amazon Managed Service for Prometheus のすべてのリソースをクラスターから直接管理できます。

CloudWatch メトリクスと Amazon Managed Service for Prometheus の統合

これは、すべてのメトリクスを 1 か所に配置するのに役立ちます。Amazon Managed Service for Prometheus は Amazon CloudWatch メトリクスを自動的に取り込みません。ただし、Amazon Data Firehose と を使用して AWS Lambda Amazon Managed Service for Prometheus に CloudWatch メトリクスをプッシュできます。

このセクションでは、Amazon CloudWatch メトリクスストリームを計測し、Amazon Data Firehose と AWS Lambda を使用して Amazon Managed Service for Prometheus にメトリクスを取り込む方法について説明します。

AWS Cloud Development Kit (CDK) を使用してスタックをセットアップし、Firehose 配信ストリーム、Lambda、Amazon S3 バケットを作成して、完全なシナリオを示します。

インフラストラクチャ

まず、このレシピのインフラストラクチャをセットアップする必要があります。

CloudWatch メトリクスストリームでは、ストリーミングメトリクスデータを HTTP エンドポイントまたは Amazon S3 バケット に転送できます。

インフラストラクチャのセットアップは、次の 4 つのステップで構成されます。

- 前提条件を構成する
- Amazon Managed Service for Prometheus ワークスペースを作成する
- 依存関係をインストールする
- スタックをデプロイする

前提条件

- AWS CLI は環境にインストールされ、設定されます。
- AWS CDK TypeScript が環境にインストールされている。
- Node.js と Go が環境にインストールされている。
- AWS オブザーバビリティ CloudWatch メトリクスエクスポーター github リポジトリ (`CWMetricsStreamExporter`) がローカルマシンにクローンされました。

Amazon Managed Service for Prometheus ワークスペースを作成するには

1. このレシピのデモアプリケーションは、Amazon Managed Service for Prometheus 上で実行されます。次のコマンドを使用して、Amazon Managed Service for Prometheus Workspace ワークスペースを作成します。

   ```
   aws amp create-workspace --alias prometheus-demo-recipe
   ```

2. 次のコマンドを使用して、ワークスペースが作成されたことを確認します。

   ```
   aws amp list-workspaces
   ```

   Amazon Managed Service for Prometheus の詳細については、「Amazon Managed Service for Prometheus ユーザーガイド」を参照してください。

依存関係をインストールするには

1. 依存関係のインストール

   `aws-o11y-recipes` リポジトリのルートから、次のコマンドを使用してディレクトリを `CWMetricStreamExporter` に変更します。

   ```
   cd sandbox/CWMetricStreamExporter
   ```

以降では、このディレクトリをリポジトリのルートと見なします。

2. 次のコマンドを実行して、ディレクトリを /cdk に変更します。

```
cd cdk
```

3. 次のコマンドを実行して、CDK の依存関係をインストールします。

```
npm install
```

4. ディレクトリをリポジトリのルートに戻してから、次のコマンドを使用してディレクトリを /lambda に変更します。

```
cd lambda
```

5. /lambda フォルダに移動したら、次のコマンドを使用して Go の依存関係をインストールします。

```
go get
```

これですべての依存関係がインストールされました。

スタックをデプロイするには

1. リポジトリのルートで config.yaml を開き、Amazon Managed Service for Prometheus ワークスペース URL を変更して、{workspace} を新しく作成したワークスペース ID に置き換えます。さらに、リージョンを変更して、Amazon Managed Service for Prometheus ワークスペースのあるリージョンを指定します。

例えば、以下の部分を変更します。

```
AMP:
    remote_write_url: "https://aps-workspaces.us-east-2.amazonaws.com/workspaces/
{workspaceId}/api/v1/remote_write"
    region: us-east-2
```

Firehose 配信ストリームと Amazon S3 バケットの名前を好みに変更します。

2. AWS CDK と Lambda コードを構築するには、リポジトリのルートで次のコマンドを実行します。

```
npm run build
```

このビルドステップにより、Go Lambda バイナリが構築され、CDK が にデプロイされます CloudFormation。

3. スタックに必要とされる IAM の変更を確認して承認し、デプロイを完了します。
4. (オプション) 次のコマンドを実行すると、スタックが作成されたことを確認できます。

```
aws cloudformation list-stacks
```

リストに `CDK Stack` という名前のスタックが表示されます。

Amazon CloudWatch ストリームの作成

メトリクスを処理する Lambda 関数ができたので、Amazon からメトリクスストリームを作成できます CloudWatch。

CloudWatch メトリクスストリームを作成するには

1. CloudWatch コンソール https://console.aws.amazon.com/cloudwatch/home#metric-streams:streamsList に移動し、メトリクスストリームの作成 を選択します。
2. 必要なメトリクスを選択します。すべてのメトリクスを選択することも、特定の名前空間からのメトリクスのみを選択することもできます。
3. `Configuration` で、[アカウントが所有している既存の Firehose を選択] を選択します。
4. CDK によって以前に作成された Firehose を使用します。[Kinesis Data Firehose ストリームを選択] ドロップダウンで、以前に作成したストリームを選択します。これは `CdkStack-KinesisFirehoseStream123456AB-sample1234` のような名前になります。
5. 出力形式を [JSON] に変更します。
6. メトリクスストリームにわかりやすい名前を付けます。
7. [メトリクスストリームの作成] を選択します。
8. (オプション) Lambda 関数の呼び出しを検証するには、Lambda コンソールに移動して `KinesisMessageHandler` 関数を選択します。[モニタリング] タブと [ログ] サブタブを選択すると、[最近の呼び出し] に、トリガーされている Lambda 関数のエントリが表示されます。

> **Note**
>
> 呼び出しが [モニタリング] タブに表示されるようになるまでに、最大で 5 分ほどかかることがあります。

これで、メトリクスが Amazon から Amazon Managed Service for Prometheus CloudWatch にストリーミングされます。

クリーンアップ

この例で使用したリソースのクリーンアップが必要になる場合があります。以下の手順では、その方法を説明します。これにより、作成したメトリクスストリームが停止します。

リソースをクリーンアップするには

1. まず、次のコマンドを使用して CloudFormation スタックを削除します。

```
cd cdk
cdk destroy
```

2. Amazon Managed Service for Prometheus ワークスペースを削除します。

```
aws amp delete-workspace --workspace-id \
    `aws amp list-workspaces --alias prometheus-sample-app --query
 'workspaces[0].workspaceId' --output text`
```

3. 最後に、Amazon CloudWatch コンソール を使用して Amazon CloudWatch メトリクスストリームを削除します。

Amazon Managed Service for Prometheus でのセキュリティ

AWS でのクラウドセキュリティは最優先事項です。AWS のユーザーは、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を利用できます。

セキュリティは、AWS とユーザーの間の責任共有です。責任共有モデルではこれを、クラウドのセキュリティ、およびクラウド内で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 - AWS は、AWS 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を負います。また AWS は、安全に使用できるサービスを提供します。AWS コンプライアンスプログラムの一環として、サードパーティー監査者が定期的にセキュリティの有効性をテストおよび検証します。Amazon Managed Service for Prometheus 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による対象範囲内の AWS のサービス」を参照してください。
- クラウド内のセキュリティ - ユーザーの責任は、使用する AWS サービスに応じて異なります。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 Service for Prometheus を使用する際に責任共有モデルを適用する方法を理解するために役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目標を達成するように Amazon Managed Service for Prometheus を構成する方法を説明します。また、Amazon Managed Service for Prometheus resources リソースのモニタリングと保護に役立つ他の 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

- AWS CloudTrailを使用した Amazon Managed Service for Prometheus API コールのログ記録
- サービスアカウントの IAM ロールの設定
- インターフェイス VPC エンドポイントでの Amazon Managed Service for Prometheus の使用

Amazon Managed Service for Prometheus でのデータ保護

- AWS 責任共有モデル、Amazon Managed Service for Prometheus 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の制御を維持する責任があります。また、 のセキュリティ設定と管理タスクについても責任を負います。 AWS のサービス 使用する 。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。欧州でのデータ保護の詳細については、「」を参照してください。 AWS の責任共有モデルとGDPRブログ記事 AWS セキュリティブログ 。

データ保護の目的で、 を保護することをお勧めします。 AWS アカウント 認証情報と を使用して個々のユーザーをセットアップする AWS IAM Identity Center または AWS Identity and Access Management （ 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 と通信する AWS リソースの使用料金を見積もることができます。1TLS.2 が必要で、1.3 TLS をお勧めします。
- で APIとユーザーアクティビティの口グ記録を設定する AWS CloudTrail。 CloudTrail 証跡を使用してキャプチャする方法については、「」を参照してください。 AWS アクティビティ、「」の「証 CloudTrail 跡の使用」を参照してください。 AWS CloudTrail ユーザーガイド。
- 使用アイテム 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 AWS のサービス.
- 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。
- アクセス時に FIPS 140-3 検証済みの暗号化モジュールが必要な場合 AWS コマンドラインインターフェイスまたは を介して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、Amazon Managed

Service for Prometheus またはその他の を使用する場合も同様です。 AWS のサービス コンソール、API、 AWS CLI、または 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。

トピック

Amazon Managed Service for Prometheus によって収集されるデータ

Amazon Managed Service for Prometheus は、アカウントで実行されている Prometheus サーバーから Amazon Managed Service for Prometheus に送信するように構成された運用メトリクスを収集して保存します。このデータには以下が含まれています。

- メトリクス値
- データの識別と分類に役立つメトリクスラベル (任意のキーと値のペア)
- データサンプルのタイムスタンプ

一意のテナントは、さまざまな顧客からデータをIDs分離します。これにより、どの顧客データにアクセスできるかIDsが制限されます。お客様はテナント を変更できませんIDs。

Amazon Managed Service for Prometheus は、保存するデータを暗号化します。 AWS Key Management Service (AWS KMS) キー。これらのキーは Amazon Managed Service for Prometheus によって管理されます。

> **Note**
>
> Amazon Managed Service for Prometheus は、データを暗号化するためのカスタマーマネージドキーの作成をサポートしています。Amazon Managed Service for Prometheus がデフォルトで使用するキーと、独自のカスタマーマネージドキーの使用方法の詳細については、「」を参照してください保管中の暗号化。

転送中のデータは でHTTPS自動的に暗号化されます。Amazon Managed Service for Prometheus は、 内のアベイラビリティーゾーン間の接続を保護します。 AWS HTTPS 内部で を使用するリージョン。

保管中の暗号化

デフォルトでは、Amazon Managed Service for Prometheus は保管時の暗号化を自動的に提供し、を使用してこれを行います。 AWS が所有する暗号化キー。

- AWS 所有キー – Amazon Managed Service for Prometheus は、これらのキーを使用して、ワークスペースにアップロードされたデータを自動的に暗号化します。を表示、管理、または使用できない AWS が所有するキー、またはそれらの使用を監査します。ただし、データを暗号化するキーを保護するためのアクションの実施やプログラムの変更を行う必要はありません。詳細については、「」を参照してくださいAWS の 所有キー AWS Key Management Service デベロッパーガイド 。

保管中のデータの暗号化は、個人を特定できる情報など、顧客の機密データを保護するにあたって伴う運用上のオーバーヘッドと複雑さを軽減するために役立ちます。これにより、厳格な暗号化のコンプライアンスと規制要件に対応する安全なアプリケーションを構築できます。

ワークスペースの作成時にカスタマーマネージドキーを使用することもできます。

- カスタマーマネージドキー — Amazon Managed Service for Prometheus では、ワークスペース内のデータを暗号化するために、ユーザーが作成、所有、管理する対称型カスタマーマネージドキーの使用をサポートします。この暗号化は完全に制御できるため、次のようなタスクを実行できます。
  - キーポリシーの策定と維持
  - IAM ポリシーと許可の確立と維持
  - キーポリシーの有効化と無効化
  - キー暗号化マテリアルのローテーション
  - タグの追加
  - キーエイリアスの作成
  - キー削除のスケジュール設定

  詳細については、「」の「カスタマーマネージドキー」を参照してください。 AWS Key Management Service デベロッパーガイド。

カスタマーマネージドキーを使用するか、 を使用するかを選択します。 AWS 所有キーは慎重に。 カスタマーマネージドキーで作成されたワークスペースは、 の使用に変換できません。 AWS 後で (またはその逆で) 所有キー。

> **Note**
>
> Amazon Managed Service for Prometheus は、 を使用して保管時の暗号化を自動的に有効にします。 AWS データを無料で保護するための 所有キー。
> ただし、 AWS KMS カスタマーマネージドキーの使用には 料金が適用されます。料金の詳細については、「」を参照してください。 AWS Key Management Service の料金

の詳細については、「」を参照してください。 AWS KMS、「 とは」を参照してください。 AWS Key Management Service?

> **Note**
>
> カスタマーマネージドキーで作成されたワークスペースは を使用できません AWS 取り込み用の マネージドコレクター。

Amazon Managed Service for Prometheus が で許可を使用する方法 AWS KMS

Amazon Managed Service for Prometheus には、カスタマーマネージドキーを使用するための許可が 3 つ必要です。

カスタマーマネージドキーで暗号化された Amazon Managed Service for Prometheus ワークスペースを作成すると、Amazon Managed Service for Prometheus は にCreateGrantリクエストを送信することで、ユーザーに代わって 3 つの許可を作成します。 AWS KMS。 でのグラント AWS KMS は、ユーザーに代わって直接呼び出されない場合でも (Amazon EKSクラスターからスクレイプされたメトリクスデータを保存する場合などに)、Amazon Managed Service for Prometheus にアカウントのKMSキーへのアクセスを許可するために使用されます。

Amazon Managed Service for Prometheus は、以下の内部オペレーションのためにユーザーのカスタマーマネージドキーを使用する許可を必要とします。

- へのDescribeKeyリクエストの送信 AWS KMS ワークスペースの作成時に指定された対称カスタマーマネージドKMSキーが有効であることを確認します。

- へのGenerateDataKeyリクエストの送信 AWS KMS カスタマーマネージドキーで暗号化されたデータキーを生成するには、 を使用します。
- Decrypt リクエストを に送信する AWS KMS 暗号化されたデータキーを復号して、データの暗号化に使用できるようにします。

Amazon Managed Service for Prometheus は、 に 3 つの許可を作成します。 AWS KMS Amazon Managed Service for Prometheus がユーザーに代わって キーを使用できるようにする キー。キーポリシーを変更するか、キーを無効にするか、または許可を取り消すことで、キーへのアクセスを削除できます。これらのアクションを実行する前に、その結果を理解しておく必要があります。これにより、ワークスペース内のデータが失われる可能性があります。

何らかの方法で許可へのアクセスを削除すると、Amazon Managed Service for Prometheus は、カスタマーマネージドキーによって暗号化されたすべてのデータにアクセスできなくなり、ワークスペースに送信された新しいデータを保存することもできなくなります。これにより、そのデータに依存するオペレーションが影響を受けます。ワークスペースに送信された新しいデータにはアクセスできなくなり、永久に失われる可能性があります。

> ⚠ Warning
>
> - キーを無効にするか、キーポリシーで Amazon Managed Service for Prometheus へのアクセスを削除すると、ワークスペースデータにはアクセスできなくなります。ワークスペースに送信される新しいデータにはアクセスできなくなり、永久に失われる可能性があります。
>
> Amazon Managed Service for Prometheus のキーへのアクセスを復元することで、ワークスペースデータにアクセスできるようになり、新しいデータの受信を再開できます。
> - 許可を取り消すと、再作成することはできず、ワークスペース内のデータは永久に失われます。

ステップ 1：カスタマーマネージドキーを作成する

を使用して、対称カスタマーマネージドキーを作成できます。 AWS Management Console、または AWS KMS APIs。以下に説明するように、ポリシーを通じて正しいアクセスを提供している限り、キーは Amazon Managed Service for Prometheus ワークスペースと同じアカウントにある必要はありません。

対称カスタマーマネージドキーを作成するには

「」の「対称カスタマーマネージドキーの作成」の手順に従います。 AWS Key Management Service デベロッパーガイド 。

キーポリシー

キーポリシーは、カスタマーマネージドキーへのアクセスを制御します。すべてのカスタマーマネージドキーには、キーポリシーが 1 つだけ必要です。このポリシーには、そのキーを使用できるユーザーとその使用方法を決定するステートメントが含まれています。カスタマーマネージドキーを作成する際に、キーポリシーを指定することができます。詳細については、「」の「カスタマーマネージドキーへのアクセスの管理」を参照してください。 AWS Key Management Service デベロッパーガイド 。

Amazon Managed Service for Prometheus ワークスペースでカスタマーマネージドキーを使用するには、キーポリシーで次のAPIオペレーションを許可する必要があります。

- kms:CreateGrant - カスタマーマネージドキーに許可を追加します。指定されたKMSキーへのアクセスを制御する権限を付与します。これにより、Amazon Managed Service for Prometheus が必要とする許可オペレーションへのアクセスを許可します。詳細については、「」の「許可の使用」を参照してください。 AWS Key Management Service デベロッパーガイド 。

  これにより、Amazon Managed Service for Prometheus は以下を実行できるようになります。

  - GenerateDataKey を呼び出して、暗号化されたデータキーを生成して保存します。データキーは暗号化にすぐには使用されないからです。
  - Decrypt を呼び出して、保存された暗号化データキーを使用して暗号化データにアクセスします。
- kms:DescribeKey — カスタマーマネージドキーの詳細を指定し、Amazon Managed Service for Prometheus がキーを検証できるようにします。

Amazon Managed Service for Prometheus に追加できるポリシーステートメントの例を以下に示します。

```
 "Statement" : [
   {
     "Sid" : "Allow access to Amazon Managed Service for Prometheus principal within
your account",
     "Effect" : "Allow",
     "Principal" : {
```

```
        "AWS" : "*"
      },
      "Action" : [
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "kms:ViaService" : "aps.region.amazonaws.com",
          "kms:CallerAccount" : "111122223333"
        }
    },
    {
      "Sid": "Allow access for key administrators - not required for Amazon Managed
Service for Prometheus",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
       },
      "Action" : [
        "kms:*"
       ],
      "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    <other statements needed for other non-Amazon Managed Service for Prometheus
scenarios>
 ]
```

- ポリシー でのアクセス許可の指定の詳細については、「」を参照してください。 AWS Key Management Service デベロッパーガイド 。
- キーアクセスのトラブルシューティングの詳細については、「」を参照してください。 AWS Key Management Service デベロッパーガイド 。

ステップ 2: Amazon Managed Service for Prometheus のカスタマーマネージドキーを指定する

ワークスペースを作成するときに、Amazon Managed Service for Prometheus がワークスペースに保存されているデータの暗号化に使用するKMSキー ARNを入力して、カスタマーマネージドキーを指定できます。

ステップ 3: Amazon Managed Grafana などの他の のサービスからのデータへのアクセス

このステップはオプションです。別のサービスから Amazon Managed Service for Prometheus データにアクセスする必要がある場合にのみ必要です。

暗号化されたデータには、 を使用するためのアクセス権限がない限り、他の のサービスからアクセスできません。 AWS KMS キー。例えば、Amazon Managed Grafana を使用してデータに対してダッシュボードまたはアラートを作成する場合は、Amazon Managed Grafana に キーへのアクセス権を付与する必要があります。

Amazon Managed Grafana にカスタマーマネージドキーへのアクセスを許可するには

1. Amazon Managed Grafana ワークスペースのリストで、Amazon Managed Service for Prometheus へのアクセスを許可するワークスペースの名前を選択します。Amazon Managed Grafana ワークスペースに関する概要情報が表示されます。
2. ワークスペースで使用されるIAMロールの名前を書き留めます。名前は の形式で すAmazonGrafanaServiceRole-<unique-id>。コンソールには、ロールARNの完全な が表示されます。この名前は、 で指定します。 AWS KMS 後のステップの コンソール。
3. の AWS KMS カスタマーマネージドキーリスト で、Amazon Managed Service for Prometheus ワークスペースの作成時に使用したカスタマーマネージドキーを選択します。キー設定の詳細ページが開きます。
4. キーユーザー の横にある「追加」ボタンを選択します。
5. 名前のリストから、上記の Amazon Managed Grafana IAMロールを選択します。検索しやすくするために、名前で検索することもできます。
6. 追加 を選択して、キーユーザーのリストにIAMロールを追加します。

Amazon Managed Grafana ワークスペースから Amazon Managed Service for Prometheus ワークスペースのデータにアクセスできるようになりました。他のユーザーまたはロールをキーユーザーに追加して、他の のサービスがワークスペースにアクセスできるようにします。

Amazon Managed Service for Prometheus 暗号化コンテキスト

暗号化コンテキストは、データに関する追加のコンテキスト情報が含まれたキーバリューペアのオプションのセットです。

AWS KMS は、追加の認証データとして暗号化コンテキストを使用して、認証された暗号化 をサポートします。データの暗号化リクエストに暗号化コンテキストを含めると、 AWS KMS は、暗号化コンテキストを暗号化されたデータにバインドします。データを復号化するには、そのリクエストに (暗号化時と) 同じ暗号化コンテキストを含めます。

Amazon Managed Service for Prometheus 暗号化コンテキスト

Amazon Managed Service for Prometheus は、すべての で同じ暗号化コンテキストを使用します。AWS KMS 暗号化オペレーション。キーは `aws:amp:arn`で、値はワークスペースの Amazon リソースネーム (ARN) です。

Example

```
"encryptionContext": {
    "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-sample-1234-
abcd-56ef-7890abcd12ef"
}
```

モニタリングに暗号化コンテキストを使用する

対称カスタマーマネージドキーを使用してワークスペースデータを暗号化する場合は、監査レコードとログで暗号化コンテキストを使用して、カスタマーマネージドキーがどのように使用されているかを特定することもできます。暗号化コンテキストは、 によって生成されたログにも表示されます。AWS CloudTrail または Amazon CloudWatch Logs 。

暗号化コンテキストを使用してカスタマーマネージドキーへのアクセスを制御する

暗号化コンテキストをキーポリシーと IAMポリシーで使用`conditions`して、対称カスタマーマネージドキーへのアクセスを制御できます。付与する際に、暗号化コンテキストの制約を使用することもできます。

Amazon Managed Service for Prometheus は、許可で暗号化コンテキスト制約を使用して、アカウントまたはリージョン内のカスタマーマネージドキーへのアクセスを制御します。権限の制約では、権限によって許可されるオペレーションで指定された暗号化コンテキストを使用する必要があります。

Example

次に、特定の暗号化コンテキストのカスタマーマネージドキーへのアクセスを付与するキーポリシーステートメントの例を示します。このポリシーステートメントの条件では、権限に暗号化コンテキストを指定する暗号化コンテキスト制約が必要です。

```
{
    "Sid": "Enable DescribeKey",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:DescribeKey",
     "Resource": "*"
},
{
     "Sid": "Enable CreateGrant",
     "Effect": "Allow",
     "Principal": {
         "AWS": "arn:aws:iam::111122223333:role/ExampleReadOnlyRole"
     },
     "Action": "kms:CreateGrant",
     "Resource": "*",
     "Condition": {
         "StringEquals": {
             "kms:EncryptionContext:aws:aps:arn": "arn:aws:aps:us-
west-2:111122223333:workspace/ws-sample-1234-abcd-56ef-7890abcd12ef"
          }
     }
}
```

Amazon Managed Service for Prometheus の暗号化キーを監視

を使用する場合 AWS KMS Amazon Managed Service for Prometheus ワークスペースで カスタマーマネージドキーを使用すると、 AWS CloudTrail Amazon Managed Service for Prometheus が に送信するリクエストを追跡するための または Amazon CloudWatch Logs AWS KMS.

以下の例を示します。 AWS CloudTrail Amazon Managed Service for Prometheus によって呼び出されたKMSオペレーションをモニタリングDescribeKeyしてCreateGrant、カスタマーマネージドキーによって暗号化されたデータにアクセスするための、GenerateDataKey、Decrypt、、および の イベント。

CreateGrant

を使用する場合 AWS KMS ワークスペースを暗号化するための カスタマーマネージドキー。Amazon Managed Service for Prometheus は、ユーザーに代わって、指定したKMSキーにアクセスするための 3 つのCreateGrantリクエストを送信します。Amazon Managed Service for Prometheus が作成する許可は、 に関連付けられたリソースに固有です。 AWS KMS カスタマーマネージドキー。

以下のイベント例では CreateGrant オペレーションを記録しています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "TESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-KEY-ID1",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "TESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "retiringPrincipal": "aps.region.amazonaws.com",
        "operations": [
```

```
            "GenerateDataKey",
            "Decrypt",
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "granteePrincipal": "aps.region.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey

を有効にする場合 AWS KMS ワークスペースの カスタマーマネージドキー、Amazon Managed Service for Prometheus は一意のキーを作成します。GenerateDataKey リクエストを に送信します。 AWS KMS を指定する AWS KMSリソースの カスタマーマネージドキー。

以下のイベント例では GenerateDataKey オペレーションを記録しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "aps.amazonaws.com"
    },
```

```
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-
sample-1234-abcd-56ef-7890abcd12ef"
        },
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "57f5dbee-16da-413e-979f-2c4c6663475e"
}
```

Decrypt

暗号化されたワークスペースでクエリが生成されると、Amazon Managed Service for Prometheus は `Decrypt` オペレーションを呼び出し、保存されている暗号化されたデータキーを使用して暗号化されたデータにアクセスします。

以下のイベント例では `Decrypt` オペレーションを記録しています。

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:10:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:aps:arn": "arn:aws:aps:us-west-2:111122223333:workspace/ws-
sample-1234-abcd-56ef-7890abcd12ef"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```

DescribeKey

Amazon Managed Service for Prometheus は DescribeKeyオペレーションを使用して、 AWS KMS ワークスペースに関連付けられた カスタマーマネージドキーは、アカウントとリージョンに存在します。

以下のイベント例では、DescribeKey オペレーションを記録しています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "TESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-KEY-ID1",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "TESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-04-22T17:02:00Z"
            }
        },
        "invokedBy": "aps.amazonaws.com"
    },
    "eventTime": "2021-04-22T17:07:02Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
```

```
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


詳細

次のリソースは、保管時のデータ暗号化についての詳細を説明しています。

- の詳細AWS Key Management Service の基本概念 については、「」を参照してください。 AWS Key Management Service デベロッパーガイド
- のセキュリティのベストプラクティスの詳細については、「」を参照してください。 AWS Key Management Service、「」を参照してください。 AWS Key Management Service デベロッパーガイド

Amazon Managed Service for Prometheus の Identity and Access Management

AWS Identity and Access Management （ IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、Amazon Managed Service for Prometheus リソースを使用するための認証 (サインイン) および認可 (アクセス許可を持つ) できるユーザーを制御します。IAM は、追加料金なしで AWS のサービス 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 Service for Prometheus の仕組み IAM
- Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例
- AWS Amazon Managed Service for Prometheus の マネージドポリシー
- Amazon Managed Service for Prometheus のアイデンティティとアクセスに関するトラブルシューティング

対象者

AWS Identity and Access Management （ IAM) の使用方法は、Amazon Managed Service for Prometheus で行う作業によって異なります。

サービスユーザー - Amazon Managed Service for Prometheus サービスを使用してジョブを実行する場合は、必要な認証情報とアクセス許可を管理者が用意します。さらに多くの Amazon Managed Service for Prometheus 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と、適切なアクセス許可を管理者にリクエストするうえで役立ちます。Amazon Managed Service for Prometheus の機能にアクセスできない場合は、「Amazon Managed Service for Prometheus のアイデンティティとアクセスに関するトラブルシューティング」を参照してください。

サービス管理者 - 社内の Amazon Managed Service for Prometheus リソースを担当している場合は、通常、Amazon Managed Service for Prometheus へのフルアクセスが付与されます。サービスユーザーが Amazon Managed Service for Prometheus のどの機能やリソースにアクセスする必要があるかを決定するのは、サービス管理者の仕事です。その後、IAM管理者にリクエストを送信して、サービスユーザーのアクセス許可を変更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。会社が Amazon Managed Service for Prometheus IAMで 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 Service for Prometheus の仕組み IAM。

IAM 管理者 – IAM管理者の場合は、Amazon Managed Service for Prometheus へのアクセスを管理するポリシーの作成方法の詳細を知りたい場合があります。で使用できる Amazon Managed Service for Prometheus 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 Service for Prometheus 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

認証は、アイデンティティ認証情報 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で、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーティッド ID の例です。フェデレーティッド ID としてサインインすると、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になります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、 AWS サインイン ユーザーガイドの「 へのサインイン方法 AWS アカウント」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化して署名します。 AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。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「リクエストの署名 AWS API」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、 では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、AWS IAM Identity Center 「 ユーザーガイド」の「多要素認証の使用」および「 ユーザーガイド」の「多要素認証の使用 (MFA ） AWS」を参照してください。 IAM

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての および リソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインイン ID 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してアクセスします。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。

フェデレーティッドアイデンティティ

ベストプラクティスとして、 では、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アクセスするために ID プロバイダーとのフェデレーション AWS のサービス の使用を要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブアイデンティティプロバイダー、 AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを通じて提供された認証情報 AWS のサービス を使用してアクセスするすべてのユーザーのユーザーです。フェデレーティッド ID が にアクセスすると AWS アカウント、それらはロールを引き受け、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したり、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ます。IAM Identity Center の詳細については、AWS IAM Identity Center 「 ユーザーガイド」のIAM「Identity Center とは」を参照してください。

IAM ユーザーとグループ

IAM ユーザーとは、1 人のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成する代わりに、一時的な認証情報に依存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、IAM「 ユーザーガイド」の「長期的な認証情報を必要とするユースケースのアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAMユーザーのコレクションを指定する ID 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があり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、 IAM ユーザーガイドの （ロールではなく) IAM ユーザーを作成するタイミングを参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内の ID です。ユーザーと似ていますがIAM、特定の人物には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。または オペレーションを AWS CLI AWS API呼び出すか、カスタム を使用してロールを引き受けることができますURL。ロールを使用する方法の詳細については、 IAM ユーザーガイドの「ロールを引き受ける方法」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 – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、 IAM ユーザーガイドの「サードパーティー ID プロバイダーのロールの作成」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 が認証された後にアクセスできる内容を制御するために、IAMIdentity 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、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IAMロールを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、別のアカウントの誰か (信頼できるプリンシパル) 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 では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセス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、IAM「 ユーザーガイド」の「 のクロスアカウントリソースアクセスIAM」を参照してください。
- クロスサービスアクセス — 他の の機能 AWS のサービス を使用するものもあります AWS のサービス。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 のサービス、ダウンストリームサービス AWS のサービス へのリクエスト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 のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「アクセスセッションの転送」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、IAM「 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 のサービス」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示することはできますが、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。EC2インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルには ロールが含まれており、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、IAM「 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」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、IAM「 ユーザーガイド」の「 (ユーザーではなく) IAMロールを作成するタイミング」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 アイデンティティ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オブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーはJSONドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、IAM「 ユーザーガイド」のJSON「ポリシーの概要」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には、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方法に関係なく、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。ID 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」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。マネージドポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには AWS 、管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーを選択する方法については、IAM ユーザーガイドの「マネージド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ロール信頼ポリシー と Amazon S3 バケットポリシー 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、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーIAMでは、 から AWS 管理ポリシーを使用することはできません。

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、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) にリソースへのアクセス許可があるかを制御します。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は使用しません。

Amazon S3、および Amazon VPCは AWS WAF、 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) 概要」を参照してください。

その他のポリシータイプ

AWS は、追加の低頻度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) に付与できる最大アクセス許可を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM「 ユーザーガイド」のIAM「エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定するJSONポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) を任意のアカウントまたはすべてのアカウントに適用できます。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もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「 ユーザーガイド」の「セッションポリシー」を参照してください。 IAM

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係してい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」を参照してください。 IAM

Amazon Managed Service for Prometheus の仕組み IAM

IAM を使用して Amazon Managed Service for Prometheus へのアクセスを管理する前に、Amazon Managed Service for Prometheus で使用できるIAM機能について説明します。

IAM Amazon Managed Service for Prometheus で使用できる機能

IAM 機能	Amazon Managed Service for Prometheus でのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	Yes
ポリシー条件キー	不可
ACLs	なし
ABAC (ポリシーのタグ)	可能
一時的な認証情報	可能
転送アクセスセッション (FAS)	不可
サービスロール	いいえ
サービスリンクロール	可能

Amazon Managed Service for Prometheus およびその他の AWS のサービスがほとんどのIAM機能でどのように機能するかの概要については、 IAM ユーザーガイドのAWS 「 で機能する のサービス IAM」を参照してください。

Amazon Managed Service for Prometheus 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。ID 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」を参照してください。 IAM

IAM ID 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、 IAM ユーザーガイドのIAMJSON「ポリシー要素リファレンス」を参照してください。

Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例

Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例については、「Amazon Managed Service for Prometheus 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。

Amazon Managed Service for Prometheus 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ロール信頼ポリシー と Amazon S3 バケットポリシー 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、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、リソースベースのポリシーのプリンシパルとして、別のアカウントのアカウントまたはIAMエンティティ全体を指定できます。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 にある場合 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、IAM「 ユーザーガイド」の「 のクロスアカウントリソースアクセスIAM」を参照してください。

Amazon Managed Service for Prometheus 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、ポリシー内のアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションは通常、関連付けられた AWS APIオペレーションと同じ名前です。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。

ポリシーにアクションを含めることで、関連するオペレーションを実行するためのアクセス許可を付与します。

Amazon Managed Service for Prometheus アクションの一覧については、「サービス認可リファレンス」の「Amazon Managed Service for Prometheus によって定義されるアクション」を参照してください。

Amazon Managed Service for Prometheus のポリシーアクションでは、アクションの前に次のプレフィックスが使用されます。

```
aps
```

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": [
```

```
      "aps:action1",
      "aps:action2"
         ]
```

Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例については、「Amazon Managed Service for Prometheus 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。

Amazon Managed Service for Prometheus 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。

`Resource` JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource` 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

Amazon Managed Service for Prometheus 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 Service for Prometheus 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 Service for Prometheus で定義されるアクション」を参照してください。

Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例については、「Amazon Managed Service for Prometheus 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。

Amazon Managed Service for Prometheus 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、IAMユーザー名でタグ付けされている場合にのみ、リソースにアクセスするアクセス許可を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」を参照してください。 IAM

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」を参照してください。 IAM

Amazon Managed Service for Prometheus 条件キーの一覧については、「サービス認可リファレンス」の「Amazon Managed Service for Prometheus によって定義される条件キー」を参照してください。どのアクションおよびリソースで条件キーを使用できるかについては、「Amazon Managed Service for Prometheus で定義されるアクション」を参照してください。

Amazon Managed Service for Prometheus のアイデンティティベースのポリシーの例については、「Amazon Managed Service for Prometheus 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。

Amazon Managed Service for Prometheus のアクセスコントロールリスト (ACLs)

をサポートACLs： なし

アクセスコントロールリスト (ACLs) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) にリソースへのアクセス許可があるかを制御します。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は使用しません。

Amazon Managed Service for Prometheus 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 (ポリシーのタグ): はい

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証戦略です。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ときに、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で役立ち、ポリシー管理が面倒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、 IAM ユーザーガイドの「 とはABAC」を参照してください。を設定する手順を含むチュートリアルを表示するにはABAC、 IAM ユーザーガイドの「属性ベースのアクセスコントロールを使用する (ABAC)」を参照してください。

Amazon Managed Service for Prometheus での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する方法などの詳細については、 IAM ユーザーガイドの AWS のサービス を使用する方法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。

AWS CLI または を使用して、一時的な認証情報を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 にアクセスできます AWS。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「」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 Service for Prometheus 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): いいえ

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 のサービス、ダウンストリームサービス AWS のサービス へのリクエスト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 のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「アクセスセッションの転送」を参照してください。

Amazon Managed Service for Prometheus のサービスロール

サービスロールのサポート: なし

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、IAM「 ユーザーガイド」の「 へのアクセス許可を委任するロールの作成 AWS のサービス」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 Service for Prometheus の機能が破損する可能性があります。Amazon Managed Service for Prometheus 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 Service for Prometheus 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 のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービ

スにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示することはできますが、編集することはできません。

Amazon Managed Service for Prometheus サービスリンクロールの作成または管理の詳細については、「Amazon Managed Service for Prometheus のサービスリンクロールの使用」を参照してください。

Amazon Managed Service for Prometheus 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、ユーザーおよびロールには、Amazon Managed Service for Prometheus リソースを作成または変更する許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または を使用してタスクを実行することはできません AWS API。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には、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。

これらのポリシードキュメント例を使用して IAM ID ベースのJSON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。

Amazon Managed Service for Prometheus で定義されるアクションとリソースタイプの詳細については、ARNs各リソースタイプの の形式など、「サービス認証リファレンス」の「Amazon Managed Service for Prometheus 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 Service for Prometheus コンソールの使用
- 自分の権限の表示をユーザーに許可する

ポリシーのベストプラクティス

アイデンティティ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 Service for Prometheus リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS マネージドポリシーを開始し、最小権限のアクセス許可に移行 – ユーザーとワークロードへのアクセス許可の付与を開始するには、多くの一般的なユースケースのアクセス許可を付与するAWS マネージドポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM「 ユーザーガイド」の「 管理AWS ポリシー」またはジョブ機能の 管理ポリシーを参照してください。 AWS
- 最小権限のアクセス許可を適用する - IAMポリシーでアクセス許可を設定する場合、タスクの実行に必要なアクセス許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用してアクセス許可を適用する方法の詳細については、IAM「 ユーザーガイド」の「 のポリシーとアクセス許可IAM」を参照してください。
- IAM ポリシーの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、アクションと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 を使用して送信する必要があることを指定できますSSL。また、 などの特定の を通じてサービスアクションが使用されている場合 AWS のサービス、 条件を使用して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IAMJSON「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、安全で機能的なアクセス許可を確保する – IAM Access Analyzer は、ポリシーがポリシー言語 (JSON) とIAMベストプラクティスに準拠するように、新規および既存のIAMポリシーを検証します。IAM Access Analyzer には、安全で機能的なポリシーの作成に役立つ 100 を超えるポリシーチェックと実用的なレコメンデーションが用意されています。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。
- 多要素認証が必要 (MFA) – 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、 をオンにMFAしてセキュリティを強化します。API オペレーションが呼び出されるMFAタイミングを要求するには、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。

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「 のセキュリティのベストプラクティスIAM」を参照してください。 IAM

Amazon Managed Service for Prometheus コンソールの使用

Amazon Managed Service for Prometheus コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可で、 AWS アカウント内の Amazon Managed Service for Prometheus リソースの一覧表示と詳細表示を許可する必要があります。最小限必要な許可より

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ません。

AWS CLI または のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません AWS API。代わりに、実行しようとしているAPIオペレーションに一致するアクションのみへのアクセスを許可します。

ユーザーとロールが引き続き Amazon Managed Service for Prometheus コンソールを使用できるようにするには、Amazon Managed Service for Prometheus `ConsoleAccess`または `ReadOnly` AWS マネージドポリシーをエンティティにアタッチします。詳細については、「 ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。 IAM

自分の権限の表示をユーザーに許可する

この例では、IAMユーザーがユーザー ID にアタッチされているインラインポリシーとマネージドポリシーを表示できるようにするポリシーを作成する方法を示します。このポリシーには、コンソールで、または AWS CLI または を使用してプログラムでこのアクションを実行するアクセス許可が含まれています AWS API。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
```

```
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Amazon Managed Service for Prometheus 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 のサービス 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。

AmazonPrometheusFullAccess

`AmazonPrometheusFullAccess` ポリシーは IAM ID にアタッチできます。

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。

- `aps` — Amazon Managed Service for Prometheus へのフルアクセスを許可します
- `eks` — Amazon Managed Service for Prometheus が Amazon EKS クラスターに関する情報を読み取れるようにします。これは、クラスター内のマネージドスクレイパーの作成とメトリクスの検出を可能にするために必要です。

- ec2 — Amazon Managed Service for Prometheus が Amazon EC2 ネットワークに関する情報を読み取れるようにします。これは、Amazon EKS メトリクスにアクセスできるマネージドスクレイパーを作成できるようにするために必要です。
- iam - マネージドメトリクススクレイパーのサービスリンクロールの作成をプリンシパルに許可します。

の内容AmazonPrometheusFullAccessは次のとおりです。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllPrometheusActions",
   "Effect": "Allow",
   "Action": [
    "aps:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DescribeCluster",
   "Effect": "Allow",
   "Action": [
    "eks:DescribeCluster",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": [
      "aps.amazonaws.com"
     ]
    }
   },
   "Resource": "*"
  },
  {
   "Sid": "CreateServiceLinkedRole",
   "Effect": "Allow",
   "Action": "iam:CreateServiceLinkedRole",
   "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*",
```

```
    "Condition": {
     "StringEquals": {
      "iam:AWSServiceName": "scraper.aps.amazonaws.com"
     }
    }
   }
  ]
}
```

AmazonPrometheusConsoleFullAccess

AmazonPrometheusConsoleFullAccess ポリシーは IAM ID にアタッチできます。

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。

- aps — Amazon Managed Service for Prometheus へのフルアクセスを許可します
- tag — プリンシパルが Amazon Managed Service for Prometheus コンソールでタグの候補を確認できるようにします。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "TagSuggestions",
   "Effect": "Allow",
   "Action": [
    "tag:GetTagValues",
    "tag:GetTagKeys"
   ],
   "Resource": "*"
  },
  {
   "Sid": "PrometheusConsoleActions",
   "Effect": "Allow",
   "Action": [
    "aps:CreateWorkspace",
    "aps:DescribeWorkspace",
    "aps:UpdateWorkspaceAlias",
    "aps:DeleteWorkspace",
```

```
    "aps:ListWorkspaces",
    "aps:DescribeAlertManagerDefinition",
    "aps:DescribeRuleGroupsNamespace",
    "aps:CreateAlertManagerDefinition",
    "aps:CreateRuleGroupsNamespace",
    "aps:DeleteAlertManagerDefinition",
    "aps:DeleteRuleGroupsNamespace",
    "aps:ListRuleGroupsNamespaces",
    "aps:PutAlertManagerDefinition",
    "aps:PutRuleGroupsNamespace",
    "aps:TagResource",
    "aps:UntagResource",
    "aps:CreateLoggingConfiguration",
    "aps:UpdateLoggingConfiguration",
    "aps:DeleteLoggingConfiguration",
    "aps:DescribeLoggingConfiguration"
   ],
   "Resource": "*"
  }
 ]
}
```

AmazonPrometheusRemoteWriteAccess

の内容AmazonPrometheusRemoteWriteAccessは次のとおりです。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:RemoteWrite"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AmazonPrometheusQueryAccess

の内容AmazonPrometheusQueryAccessは次のとおりです。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aps:GetLabels",
                "aps:GetMetricMetadata",
                "aps:GetSeries",
                "aps:QueryMetrics"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

AWS マネージドポリシー： AmazonPrometheusScraperServiceRolePolicy

IAM エンティティ AmazonPrometheusScraperServiceRolePolicy に をアタッチすることはできません。このポリシーは、ユーザーに代わって Amazon Managed Service for Prometheus がアクションを実行することを許可する、サービスリンクロールにアタッチされます。詳細については、「EKS からメトリクスをスクレイピングするためのロールの使用」を参照してください。

このポリシーは、Amazon EKS クラスターからの読み取りと Amazon Managed Service for Prometheus ワークスペースへの書き込みを許可する権限を寄稿者に付与します。

> **Note**
>
> このユーザーガイドは、以前に誤ってこのポリシーと呼ばれていました。`AmazonPrometheusScraperServiceLinkedRolePolicy`

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。

- aps — サービスプリンシパルが Amazon Managed Service for Prometheus ワークスペース にメトリクスを書き込むことを許可します。
- ec2 — サービスプリンシパルがネットワーク設定を読み取って変更し、Amazon EKS クラスターを含むネットワークに接続できるようにします。

- eks — サービスプリンシパルが Amazon EKS クラスターにアクセスできるようにします。これは、メトリクスを自動的にスクレイプできるようにするために必要です。また、スクレイパーが削除されたときにプリンシパルが Amazon EKS リソースをクリーンアップできるようにします。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "DeleteSLR",
   "Effect": "Allow",
   "Action": [
    "iam:DeleteRole"
   ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/scraper.aps.amazonaws.com/
AWSServiceRoleForAmazonPrometheusScraper*"
  },
  {
   "Sid": "NetworkDiscovery",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ENIManagement",
   "Effect": "Allow",
   "Action": "ec2:CreateNetworkInterface",
   "Resource": "*",
   "Condition": {
    "ForAllValues:StringEquals": {
     "aws:TagKeys": [
      "AMPAgentlessScraper"
     ]
    }
   }
  },
  {
   "Sid": "TagManagement",
   "Effect": "Allow",
```

```
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
     },
     "Null": {
      "aws:RequestTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "ENIUpdating",
    "Effect": "Allow",
    "Action": [
     "ec2:DeleteNetworkInterface",
     "ec2:ModifyNetworkInterfaceAttribute"
    ],
    "Resource": "*",
    "Condition": {
     "Null": {
      "ec2:ResourceTag/AMPAgentlessScraper": "false"
     }
    }
   },
   {
    "Sid": "EKSAccess",
    "Effect": "Allow",
    "Action": "eks:DescribeCluster",
    "Resource": "arn:aws:eks:*:*:cluster/*"
   },
   {
    "Sid": "DeleteEKSAccessEntry",
    "Effect": "Allow",
    "Action": "eks:DeleteAccessEntry",
    "Resource": "arn:aws:eks:*:*:access-entry/*/role/*",
    "Condition": {
     "StringEquals": {
      "aws:PrincipalAccount": "${aws:ResourceAccount}"
     },
     "ArnLike": {
      "eks:principalArn": "arn:aws:iam::*:role/aws-service-role/
scraper.aps.amazonaws.com/AWSServiceRoleForAmazonPrometheusScraper*"
     }
```

```
    }
   },
   {
    "Sid": "APSWriting",
    "Effect": "Allow",
    "Action": "aps:RemoteWrite",
    "Resource": "arn:aws:aps:*:*:workspace/*",
    "Condition": {
     "StringEquals": {
      "aws:PrincipalAccount": "${aws:ResourceAccount}"
     }
    }
   }
  ]
}
```

マネージド AWS ポリシーに対する Amazon Managed Service for Prometheus の更新

Amazon Managed Service for Prometheus 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通知を受け取るには、Amazon Managed Service for Prometheus ドキュメントの履歴ページから、RSS フィードにサブスクライブしてください。

変更	説明	日付
AmazonPrometheusScraperServiceRolePolicy - 既存ポリシーへの更新	Amazon Managed Service for Prometheus は、Amazon EKS でのアクセスエントリの使用をサポートする新しいアクセス許可AmazonPrometheusScraperServiceRolePolicyを に追加しました。 スクレイパーが削除されたときにリソースをクリーンアップできるように、Amazon EKS アクセスエントリを管理するためのアクセス許可が含まれています。	2024 年 5 月 2 日

変更	説明	日付
	Note このユーザーガイドは、以前に誤ってこのポリシーと呼ばれていました。 `AmazonPrometheusScraperServiceLinkedRolePolicy`	
AmazonPrometheusFullAccess – 既存ポリシーへの更新	Amazon Managed Service for Prometheus では、Amazon EKS クラスター内のメトリクスのマネージドスクレイパーの作成をサポートする新しい権限が AmazonPrometheusFullAccess に追加されました。 Amazon EKS クラスターへの接続、Amazon EC2 ネットワークの読み取り、およびスクレイパー用のサービスリンクロールを作成するための権限が含まれます。	2023 年 11 月 26 日

変更	説明	日付
AmazonPrometheusScraperServiceLinkedRolePolicy - 新しいポリシー	Amazon Managed Service for Prometheus では、Amazon EKS コンテナから読み取る新しいサービスリンクロールポリシーが追加され、メトリクスを自動的にスクレイピングできるようになりました。 Amazon EKS クラスターへの接続、Amazon EC2 ネットワークの読み取り、`AMPAgentlessScraper` としてタグ付けされたネットワークの作成と削除、および Amazon Managed Service for Prometheus ワークスペースへの書き込みを行うアクセス許可が含まれます。	2023 年 11 月 26 日

変更	説明	日付
AmazonPrometheusConsoleFullAccess – 既存ポリシーへの更新	Amazon Managed Service for Prometheus は、アラートマネージャーとルーラーイベントの CloudWatch ログ記録をサポートするためにAmazonPrometheusConsoleFullAccess、 に新しいアクセス許可を追加しました。 `aps:CreateLoggingConfiguration` 、`aps:UpdateLoggingConfiguration` 、`aps:DeleteLoggingConfiguration` 、`aps:DescribeLoggingConfiguration` のアクセス許可が追加されました。	2022 年 10 月 24 日

変更	説明	日付
AmazonPrometheusConsoleFullAccess – 既存ポリシーへの更新	Amazon Managed Service for Prometheus で、Amazon Managed Service for Prometheus の新機能をサポートするための新しいアクセス許可が AmazonPrometheusConsoleFullAccess に追加されました。これにより、このポリシーを持つユーザーは、Amazon Managed Service for Prometheus リソースにタグを適用するときに、タグの候補のリストを表示できます。 `tag:GetTagKeys 、tag:GetTagValues 、aps:CreateAlertManagerDefinition 、aps:CreateRuleGroupsNamespace 、aps:DeleteAlertManagerDefinition 、aps:DeleteRuleGroupsNamespace 、aps:DescribeAlertManagerDefinition 、aps:DescribeRuleGroupsNamespace 、aps:ListRuleGroupsNamespaces 、aps:PutAlertManagerDefinition 、aps:PutRuleGroupsN`	2021 年 9 月 29 日

変更	説明	日付
	amespace 、aps:TagResource 、aps:UntagResource のアクセス許可が追加されました。	
Amazon Managed Service for Prometheus で変更の追跡を開始	Amazon Managed Service for Prometheus 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 15 日

Amazon Managed Service for Prometheus のアイデンティティとアクセスに関するトラブルシューティング

Amazon Managed Service for Prometheus と を使用する際に発生する可能性のある一般的な問題を診断して修正するには、以下の情報を使用しますIAM。

トピック

- Amazon Managed Service for Prometheus 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 アカウント外のユーザーに Amazon Managed Service for Prometheus リソースへのアクセスを許可したい

Amazon Managed Service for Prometheus 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、mateojacksonIAMユーザーがコンソールを使用して架空の*my-example-widget*リソースの詳細を表示しようとしたときに、架空のaps:*GetWidget*アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aps:GetWidget on resource: my-example-widget
```

この場合、aps:*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 Service for Prometheus にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、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。

次のエラー例は、 という名前のIAMユーザーがコンソールを使用して Amazon Managed Service for Prometheus marymajor でアクションを実行しようとすると発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

AWS アカウント外のユーザーに Amazon Managed Service for Prometheus 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 (ACLs) をサポートするサービスでは、これらのポリシーを使用して、リソースへのアクセスをユーザーに許可できます。

詳細については、以下を参照してください。

- Amazon Managed Service for Prometheus がこれらの機能をサポートしているかどうかを確認するには、「Amazon Managed Service for Prometheus の仕組み IAM」を参照してください。

- 所有 AWS アカウント している リソースへのアクセスを提供する方法については、 IAM ユーザーガイドの「所有 AWS アカウント している別の のIAMユーザーへのアクセスを提供する」を参照してください。
- サードパーティー にリソースへのアクセスを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通じ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「外部認証されたユーザーへのアクセスを提供する (ID フェデレーション)」を参照してください。
- クロスアカウントアクセスにロールとリソースベースのポリシーを使用する違いについては、IAM ユーザーガイドの「 のクロスアカウントリソースアクセスIAM」を参照してください。

IAM のアクセス許可とポリシー

Amazon Managed Service for Prometheus のアクションとデータにアクセスするには、認証情報が必要です。これらの認証情報には、クラウドリソースに関する Amazon Managed Service for Prometheus データを取得するなど、アクションを実行したり、AWS リソースにアクセスしたりするためのアクセス許可が付与されている必要があります。以下のセクションでは、AWS Identity and Access Management (IAM) と Amazon Managed Service for Prometheus 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 でのポリシーとアクセス許可」を参照してください。

Amazon Managed Service for Prometheus のアクセス許可

次の表は、Amazon Managed Service for Prometheus で実行される可能性のあるアクションと、そのアクションに必要なアクセス許可を示しています。アクションには他のサービスからのアクセス許可も必要になる場合がありますが、ここでは詳しく説明しません。

[アクション]	必要なアクセス許可
アラートを作成する。	`aps:CreateAlertManagerAlerts`
ワークスペースにアラートマネージャーの定義を作成する。詳細については、「アラートマネージャーによる Amazon Managed Service for Prometheus でのア	`aps:CreateAlertManagerDefinition`

[アクション]	必要なアクセス許可
ラートの管理と転送」を参照してください。	
ワークスペースにルールグループ名前空間を作成する。詳細については、「ルールを使用して、受信時にメトリクスを変更またはモニタリングする」を参照してください。	`aps:CreateRuleGroupsNamespace`
Amazon Managed Service for Prometheus ワークスペースを作成する。ワークスペースは、Prometheus メトリクスの保存とクエリに使用される専用の論理スペースです。	`aps:CreateWorkspace`
ワークスペースからアラートマネージャーの定義を削除する。	`aps:DeleteAlertManagerDefinition`
アラートサイレンスを削除する。	`aps:DeleteAlertManagerSilence`
Amazon Managed Service for Prometheus ワークスペースを削除する。	`aps:DeleteWorkspace`
アラートマネージャーの定義に関する詳細情報を取得する。	`aps:DescribeAlertManagerDefinition`
ルールグループ名前空間に関する詳細情報を取得する。	`aps:DescribeRuleGroupsNamespace`
Amazon Managed Service for Prometheus ワークスペースに関する詳細情報を取得する。	`aps:DescribeWorkspace`
アラートサイレンスに関する詳細情報を取得する。	`aps:GetAlertManagerSilence`

[アクション]	必要なアクセス許可
ワークスペース内のアラートマネージャーのステータスを取得する。	aps:GetAlertManagerStatus
ラベルを取得する。	aps:GetLabels
Amazon Managed Service for Prometheus メトリクスのメタデータを取得する。	aps:GetMetricMetadata
時系列データを取得する。	aps:GetSeries
アラートマネージャーの定義に含まれているアラートグループのリストを取得する。	aps:ListAlertManagerAlertGroups
アラートマネージャーで定義されているアラートのリストを取得する。	aps:ListAlertManagerAlerts
アラートマネージャーの定義に含まれているレシーバーのリストを取得する。	aps:ListAlertManagerReceivers
定義されているアラートサイレンスのリストを取得する。	aps:ListAlertManagerSilences
アクティブなアラートのリストを取得する。	aps:ListAlerts
ワークスペース内のルールグループ名前空間にあるルールのリストを取得する。	aps:ListRules
ワークスペース内のルールグループ名前空間のリストを取得する。	aps:ListRuleGroupsNamespaces
Amazon Managed Service for Prometheus に関連付けられているタグを取得する。	aps:ListTagsForResource

[アクション]	必要なアクセス許可
アカウント内に存在する Amazon Managed Service for Prometheus ワークスペースのリストを取得する。	`aps:ListWorkspaces`
ワークスペース内の既存のアラートマネージャーの定義を更新する。	`aps:PutAlertManagerDefinition`
アラートサイレンスを作成する。	`aps:PutAlertManagerSilences`
既存のルールグループ名前空間を更新する。	`aps:PutRuleGroupsNamespace`
Amazon Managed Service for Prometheus メトリクスに対するクエリを実行する。	`aps:QueryMetrics`
リモート書き込みオペレーションを実行して Prometheus サーバーから Amazon Managed Service for Prometheus へのメトリクスのストリーミングを開始する。	`aps:RemoteWrite`
Amazon Managed Service for Prometheus リソースにタグを割り当てる。	`aps:TagResource`
Amazon Managed Service for Prometheus リソースからタグを削除する。	`aps:UntagResource`
既存のワークスペースのエイリアスを変更する。	`aps:UpdateWorkspaceAlias`
ログ記録設定を作成する。	`aps:CreateLoggingConfiguration`
ログ記録設定を削除する。	`aps:DeleteLoggingConfiguration`
ログ記録設定を記述する。	`aps:DescribeLoggingConfiguration`
ログ記録設定を更新する。	`aps:UpdateLoggingConfiguration`

IAM ポリシーのサンプル

このセクションでは、ユーザーが作成できるセルフマネージドポリシーの例を示します。

次の IAM ポリシーでは、Amazon Managed Service for Prometheus へのフルアクセスを許可するとともに、ユーザーが Amazon EKS クラスターを検出してその詳細を確認できるようにします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aps:*",
                "eks:DescribeCluster",
                "eks:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon Managed Service for Prometheus のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス 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 のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 は、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- 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 のサービス 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 のサービス 、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 のサービス 、 内のセキュリティ状態を包括的に把握できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、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 のサービス 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 のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Amazon Managed Service for Prometheus の耐障害性

AWS のグローバルインフラストラクチャは、AWS リージョンとアベイラビリティーゾーンを中心として構築されています。AWSリージョンには、低レイテンシー、高いスループット、そして高

度の冗長ネットワークで接続されている複数の物理的に独立および隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

AWS のグローバルインフラストラクチャに加えて、Amazon Managed Service for Prometheus にも、高可用性データのサポートなど、データの耐障害性とバックアップのニーズに対応するための機能が用意されています。

Amazon Managed Service for Prometheus 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 Service for Prometheus は、 AWS グローバルネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「 Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

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 Service for Prometheus 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。

Amazon Managed Service for Prometheus のサービスリンクロールの使用

Amazon Managed Service for Prometheus 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。サービスリンクロールは、Amazon Managed Service for Prometheus に直接リンクされた特殊な IAM ロールです。サービスリンクロールは Amazon Managed Service for Prometheus によって事前に定義されており、サービスがユーザーに代わって他の 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。

必要なアクセス許可を手動で追加する必要がないため、サービスリンクロールは Amazon Managed Service for Prometheus のセットアップを容易にします。サービスリンクロールのアクセス許可は Amazon Managed Service for Prometheus が定義し、別段の定義がない限り、Amazon Managed Service for Prometheus 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。

EKS からメトリクスをスクレイピングするためのロールの使用

Amazon Managed Service for Prometheus マネージドコレクターを使用してメトリクスを自動的にスクレイピングする場合、 AWSServiceRoleForAmazonPrometheusScraper必要なアクセス許可を手動で追加する必要がないため、サービスにリンクされたロールを使用してマネージドコレクターを簡単にセットアップできます。Amazon Managed Service for Prometheus がアクセス許可を定義し、Amazon Managed Service for Prometheus のみがロールを引き受けることができます。

サービスリンク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 role (サービスリンクロール)] 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 Service for Prometheus 用のサービスリンクロールのアクセス許可

Amazon Managed Service for Prometheus は、プレフィックス が付いた という名前のサービスにリンクされたロールを使用してAWSServiceRoleForAmazonPrometheusScraper、Amazon Managed Service for Prometheus が Amazon EKS クラスター内のメトリクスを自動的にスクレイプできるようにします。

AWSServiceRoleForAmazonPrometheusScraper 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。

- `scraper.aps.amazonaws.com`

という名前のロールアクセス許可ポリシーAmazonPrometheusScraperServiceRolePolicyにより、Amazon Managed Service for Prometheus 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。

- Amazon EKS クラスターを含むネットワークに接続するためのネットワーク設定を準備し、変更します。
- Amazon EKS クラスターからメトリクスを読み取り、Amazon Managed Service for Prometheus ワークスペースにメトリクスを書き込みます。

ユーザー、グループ、ロールなどがサービスリンクロールを作成できるようにするには、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 Service for Prometheus のサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。、、または AWS CLI AWS API で Amazon EKS または Amazon Managed Service for Prometheus を使用してマネージドコレクターインスタンスを作成する AWS Management Consoleと、Amazon Managed Service for Prometheus 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。詳細については、「 に新しいロールが表示されました AWS アカウント」を参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。Amazon EKS または Amazon Managed Service for Prometheus を使用してマネージドコレクターインスタンスを作成する際に、Amazon Managed Service for Prometheus によってサービスリンクロールが再作成されます。

Amazon Managed Service for Prometheus のサービスリンクロール

Amazon Managed Service for Prometheus では、 AWSServiceRoleForAmazonPrometheusScraper 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 Service for Prometheus のサービスリンクロールの削除

AWSServiceRoleForAmazonPrometheusScraper ロールを手動で削除する必要はありません。 AWS Management Console、、 AWS CLIまたは AWS API のロールに関連付けられているすべてのマネージドコレクターインスタンスを削除すると、Amazon Managed Service for Prometheus はリソースをクリーンアップし、サービスにリンクされたロールを削除します。

Amazon Managed Service for Prometheus のサービスリンクロールがサポートされているリージョン

Amazon Managed Service for Prometheus 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リンクロールの使用をサポートします。詳細については、「サポートされるリージョン」を参照してください。

AWS CloudTrailを使用した Amazon Managed Service for Prometheus API コールのログ記録

Amazon Managed Service for Prometheus は、ユーザーAWS CloudTrail、ロール、または 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 のサービス。 CloudTrail は、Amazon Managed Service for Prometheus のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、Amazon Managed Service for Prometheus コンソールからの呼び出しと、Amazon Managed Service for Prometheus API オペレーションへのコード呼び出しが含まれます。で収集された情報を使用して CloudTrail、Amazon Managed Service for Prometheus に対して行われたリクエスト、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
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

CloudTrail アカウント AWS アカウント を作成すると、 は 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、 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 でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成 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、行ベースの JSON 形式の既存のイベントを Apache ORC 形式に変換します。ORC は、データを高速に取得するために最適化された単票ストレージ形式です。イベントはイベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基いた、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。

の Amazon Managed Service for Prometheus 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 CloudTrail を記録します。

Amazon Managed Service for Prometheus は、すべての Amazon Managed Service for Prometheus 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 Service for Prometheus が に記録する Amazon Managed Service for Prometheus 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 Service for Prometheus API リファレンス」を参照してください。

Amazon Managed Service for Prometheus 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。

例： CreateWorkspace

次の例は、 アクションを示す CloudTrail CreateWorkspaceログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
```

```
            },
            "webIdFederationData": {

            },
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-11-30T23:39:29Z"
            }
        }
    },
    "eventTime": "2020-11-30T23:43:21Z",
    "eventSource": "aps.amazonaws.com",
    "eventName": "CreateWorkspace",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.1",
    "userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
    "requestParameters": {
        "alias": "alias-example",
        "clientToken": "12345678-1234-abcd-1234-12345abcd1"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "arn": "arn:aws:aps:us-west-2:123456789012:workspace/ws-abc123456-
abcd-1234-5678-1234567890",
        "status": {
            "statusCode": "CREATING"
        },
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

例： CreateAlertManagerDefinition

次の例は、 CreateAlertManagerDefinition アクションを示す CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {

            },
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-09-23T20:20:14Z"
            }
        }
    },
    "eventTime": "2021-09-23T20:22:43Z",
    "eventSource": "aps.amazonaws.com",
    "eventName": "CreateAlertManagerDefinition",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.1",
    "userAgent": "Boto3/1.17.46 Python/3.6.14 Linux/4.14.238-182.422.amzn2.x86_64 exec-
env/AWS_ECS_FARGATE Botocore/1.20.46",
    "requestParameters": {
        "data":
 "YWxlcnRtYW5hZ2VyX2NvbmZpZzogfAogIGdsb2JhbDoKICAgIHNtdHBfc21hcnRob3N0OiAnbG9jYWxob3N0OjI1JwogI
        "clientToken": "12345678-1234-abcd-1234-12345abcd1",
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "status": {
            "statusCode": "CREATING"
```

```
        }
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```

例： CreateRuleGroupsNamespace

次の例は、 CreateRuleGroupsNamespace アクションを示す CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE123EXAMPLE123-1234567890616",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/admin",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {

            },
            "attributes": {
                "creationDate": "2021-09-23T20:22:19Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2021-09-23T20:25:08Z",
    "eventSource": "aps.amazonaws.com",
```

```
    "eventName": "CreateRuleGroupsNamespace",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "34.212.33.165",
    "userAgent": "Boto3/1.17.63 Python/3.6.14 Linux/4.14.238-182.422.amzn2.x86_64 exec-
env/AWS_ECS_FARGATE Botocore/1.20.63",
    "requestParameters": {
        "data":
 "Z3JvdXBzOgogIC0gbmFtZTogdGVzdFJ1bGVHcm91cHNOYW1lc3BhY2UKICAgIHJ1bGVzOgogICAgLSBhbGVydDogdGVzd
        "clientToken": "12345678-1234-abcd-1234-12345abcd1",
        "name": "exampleRuleGroupsNamespace",
        "workspaceId": "ws-12345678-1234-abcd-1234-1234567890"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
trace-id,x-amzn-errormessage,x-amz-apigw-id,date",
        "name": "exampleRuleGroupsNamespace",
        "arn": "arn:aws:aps:us-west-2:492980759322:rulegroupsnamespace/ws-
ae46a85c-1609-4c22-90a3-2148642c3b6c/exampleRuleGroupsNamespace",
        "status": {
            "statusCode": "CREATING"
        },
        "tags": {}
    },
    "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
    "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "123456789012"
}
```


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AWS CloudTrail 」を参照してください。

サービスアカウントの IAM ロールの設定

サービスアカウントの IAM ロールを使用すると、IAM ロールを Kubernetes サービスアカウントに関連付けることができます。このサービスアカウントは、そのサービスアカウントを使用するポッド内のコンテナに AWS アクセス許可を提供できます。詳細については、「サービスアカウントの IAM ロール」を参照してください。

サービスアカウントの IAM ロールは、サービスロールとも呼ばれます。

Amazon Managed Service for Prometheus でサービスロールを使用すると、Amazon Managed Service for Prometheus、Prometheus サーバー、Grafana サーバー間での認証と認可に必要なロールを取得するために役立ちます。

前提条件

このページの手順では、AWS CLI と EKSCTL コマンドラインインターフェイスがインストールされている必要があります。

Amazon EKS クラスターからメトリクスを取り込むためのサービスロールの設定

サービスロールを設定して、Amazon Managed Service for Prometheus で Amazon EKS クラスター内の Prometheus サーバーからメトリクスを取り込めるようにするには、次のアクセス許可を持つアカウントにログオンする必要があります。

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Amazon Managed Service for Prometheus への取り込みのためのサービスロールを設定するには

1. `createIRSA-AMPIngest.sh` という名前のファイルを作成し、次の内容を記述します。<my_amazon_eks_clustername> はクラスターの名前に、<my_prometheus_namespace> は Prometheus 名前空間に置き換えます。

```
#!/bin/bash -e
CLUSTER_NAME=<my_amazon_eks_clustername>
SERVICE_ACCOUNT_NAMESPACE=<my_prometheus_namespace>
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
OIDC_PROVIDER=$(aws eks describe-cluster --name $CLUSTER_NAME --query
 "cluster.identity.oidc.issuer" --output text | sed -e "s/^https:\/\///")
SERVICE_ACCOUNT_AMP_INGEST_NAME=amp-iamproxy-ingest-service-account
SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE=amp-iamproxy-ingest-role
SERVICE_ACCOUNT_IAM_AMP_INGEST_POLICY=AMPIngestPolicy
```

```
#
# Set up a trust policy designed for a specific combination of K8s service account
 and namespace to sign in from a Kubernetes cluster which hosts the OIDC Idp.
#
cat <<EOF > TrustPolicy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-provider/
${OIDC_PROVIDER}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_PROVIDER}:sub": "system:serviceaccount:
${SERVICE_ACCOUNT_NAMESPACE}:${SERVICE_ACCOUNT_AMP_INGEST_NAME}"
        }
      }
    }
  ]
}
EOF
#
# Set up the permission policy that grants ingest (remote write) permissions for
 all AMP workspaces
#
cat <<EOF > PermissionPolicyIngest.json
{
  "Version": "2012-10-17",
   "Statement": [
       {"Effect": "Allow",
        "Action": [
           "aps:RemoteWrite",
           "aps:GetSeries",
           "aps:GetLabels",
           "aps:GetMetricMetadata"
        ],
        "Resource": "*"
      }
   ]
}
```

```
EOF

function getRoleArn() {
  OUTPUT=$(aws iam get-role --role-name $1 --query 'Role.Arn' --output text 2>&1)

  # Check for an expected exception
  if [[ $? -eq 0 ]]; then
    echo $OUTPUT
  elif [[ -n $(grep "NoSuchEntity" <<< $OUTPUT) ]]; then
    echo ""
  else
    >&2 echo $OUTPUT
    return 1
  fi
}

#
# Create the IAM Role for ingest with the above trust policy
#
SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN=$(getRoleArn
 $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE)
if [ "$SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN" = "" ];
then
  #
  # Create the IAM role for service account
  #
  SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN=$(aws iam create-role \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE \
  --assume-role-policy-document file://TrustPolicy.json \
  --query "Role.Arn" --output text)
  #
  # Create an IAM permission policy
  #
  SERVICE_ACCOUNT_IAM_AMP_INGEST_ARN=$(aws iam create-policy --policy-name
 $SERVICE_ACCOUNT_IAM_AMP_INGEST_POLICY \
  --policy-document file://PermissionPolicyIngest.json \
  --query 'Policy.Arn' --output text)
  #
  # Attach the required IAM policies to the IAM role created above
  #
  aws iam attach-role-policy \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE \
  --policy-arn $SERVICE_ACCOUNT_IAM_AMP_INGEST_ARN
else
```

```
    echo "$SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN IAM role for ingest already
 exists"
fi
echo $SERVICE_ACCOUNT_IAM_AMP_INGEST_ROLE_ARN
#
# EKS cluster hosts an OIDC provider with a public discovery endpoint.
# Associate this IdP with AWS IAM so that the latter can validate and accept the
 OIDC tokens issued by Kubernetes to service accounts.
# Doing this with eksctl is the easier and best approach.
#
eksctl utils associate-iam-oidc-provider --cluster $CLUSTER_NAME --approve
```

2. 次のコマンドを入力して、スクリプトに必要な権限を付与します。

```
chmod +x createIRSA-AMPIngest.sh
```

3. スクリプトを実行します。

メトリクスのクエリを実行するためのサービスアカウントの IAM ロールの設定

サービスアカウントの IAM ロール (サービスロール) を設定して、Amazon Managed Service for Prometheus ワークスペースからメトリクスのクエリを実行できるようにするには、次のアクセス許可を持つアカウントにログオンする必要があります。

- `iam:CreateRole`
- `iam:CreatePolicy`
- `iam:GetRole`
- `iam:AttachRolePolicy`
- `iam:GetOpenIDConnectProvider`

Amazon Managed Service for Prometheus メトリクスのクエリを実行するためのサービスロールを設定するには

1. `createIRSA-AMPQuery.sh` という名前のファイルを作成し、次の内容を記述します。<my_amazon_eks_clustername> はクラスターの名前に置き換え、<my_prometheus_namespace> は Prometheus 名前空間に置き換えます。

```
#!/bin/bash -e
CLUSTER_NAME=<my_amazon_eks_clustername>
SERVICE_ACCOUNT_NAMESPACE=<my_prometheus_namespace>
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
OIDC_PROVIDER=$(aws eks describe-cluster --name $CLUSTER_NAME --query
 "cluster.identity.oidc.issuer" --output text | sed -e "s/^https:\/\///")
SERVICE_ACCOUNT_AMP_QUERY_NAME=amp-iamproxy-query-service-account
SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE=amp-iamproxy-query-role
SERVICE_ACCOUNT_IAM_AMP_QUERY_POLICY=AMPQueryPolicy
#
# Setup a trust policy designed for a specific combination of K8s service account
 and namespace to sign in from a Kubernetes cluster which hosts the OIDC Idp.
#
cat <<EOF > TrustPolicy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-provider/
${OIDC_PROVIDER}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_PROVIDER}:sub": "system:serviceaccount:
${SERVICE_ACCOUNT_NAMESPACE}:${SERVICE_ACCOUNT_AMP_QUERY_NAME}"
        }
      }
    }
  ]
}
EOF
#
# Set up the permission policy that grants query permissions for all AMP workspaces
#
cat <<EOF > PermissionPolicyQuery.json
{
  "Version": "2012-10-17",
   "Statement": [
       {"Effect": "Allow",
        "Action": [
```

```
            "aps:QueryMetrics",
            "aps:GetSeries",
            "aps:GetLabels",
            "aps:GetMetricMetadata"
        ],
        "Resource": "*"
      }
   ]
}
EOF

function getRoleArn() {
  OUTPUT=$(aws iam get-role --role-name $1 --query 'Role.Arn' --output text 2>&1)

  # Check for an expected exception
  if [[ $? -eq 0 ]]; then
    echo $OUTPUT
  elif [[ -n $(grep "NoSuchEntity" <<< $OUTPUT) ]]; then
    echo ""
  else
    >&2 echo $OUTPUT
    return 1
  fi
}

#
# Create the IAM Role for query with the above trust policy
#
SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN=$(getRoleArn
 $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE)
if [ "$SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN" = "" ];
then
  #
  # Create the IAM role for service account
  #
  SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN=$(aws iam create-role \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE \
  --assume-role-policy-document file://TrustPolicy.json \
  --query "Role.Arn" --output text)
  #
  # Create an IAM permission policy
  #
  SERVICE_ACCOUNT_IAM_AMP_QUERY_ARN=$(aws iam create-policy --policy-name
 $SERVICE_ACCOUNT_IAM_AMP_QUERY_POLICY \
```

```
  --policy-document file://PermissionPolicyQuery.json \
  --query 'Policy.Arn' --output text)
  #
  # Attach the required IAM policies to the IAM role create above
  #
  aws iam attach-role-policy \
  --role-name $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE \
  --policy-arn $SERVICE_ACCOUNT_IAM_AMP_QUERY_ARN
else
    echo "$SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN IAM role for query already
 exists"
fi
echo $SERVICE_ACCOUNT_IAM_AMP_QUERY_ROLE_ARN
#
# EKS cluster hosts an OIDC provider with a public discovery endpoint.
# Associate this IdP with AWS IAM so that the latter can validate and accept the
 OIDC tokens issued by Kubernetes to service accounts.
# Doing this with eksctl is the easier and best approach.
#
eksctl utils associate-iam-oidc-provider --cluster $CLUSTER_NAME --approve
```

2. 次のコマンドを入力して、スクリプトに必要な権限を付与します。

```
chmod +x createIRSA-AMPQuery.sh
```

3. スクリプトを実行します。

インターフェイス VPC エンドポイントでの Amazon Managed Service for Prometheus の使用

Amazon Virtual Private Cloud (Amazon VPC) を使用して AWS リソースをホストする場合、VPC と Amazon Managed Service for Prometheus の間にプライベート接続を確立できます。これらの接続を使用すると、Amazon Managed Service for Prometheus はパブリックインターネットを経由せずに VPC のリソースと通信できます。

Amazon VPC は、ユーザー定義の仮想ネットワークで AWS リソースを起動するために使用できる AWS のサービスです。VPC を使用することで、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 Service for Prometheus に接続するには、VPC を AWS のサービスに接続するためのインターフェイス VPC エンドポイントを定義します。このエンドポイントは、インターネットゲートウェイ、

ネットワークアドレス変換 (NAT) インスタンス、または VPN 接続を必要とすることなく、Amazon Managed Service for Prometheus へのスケーラブルで信頼性の高い接続を提供します。詳細については、「Amazon VPC ユーザーガイド」の「Amazon VPC とは」を参照してください。

インターフェイス VPC エンドポイントは AWS PrivateLink を利用しています。これは、Elastic Network Interface とプライベート IP アドレスを使用して AWS のサービス間のプライベート通信を可能にする AWS のテクノロジーです。詳細については、ブログ記事の「New - AWS PrivateLink for AWS Services」を参照してください。

以下の情報は Amazon VPC ユーザーを対象としています。詳細については、「Amazon VPC ユーザーガイド」の「開始方法」を参照してください。

Amazon Managed Service for Prometheus 用のインターフェイス VPC エンドポイントの作成

インターフェイス VPC エンドポイントを作成して、Amazon Managed Service for Prometheus の使用を開始します。次のいずれかのサービス名エンドポイントを選択します。

- `com.amazonaws.region.aps-workspaces`

  Prometheus 互換 API を使用するには、このサービス名を選択します。詳細については、「Amazon Managed Service for Prometheus ユーザーガイド」の「Prometheus 互換 API」を参照してください。

- `com.amazonaws.region.aps`

  ワークスペースの管理タスクを実行するには、このサービス名を選択します。詳細については、「Amazon Managed Service for Prometheus ユーザーガイド」の「Amazon Managed Service for Prometheus API」を参照してください。

> **Note**
>
> インターネットに直接アクセスできない VPC で remote_write を使用している場合は、AWS Security Token Service のインターフェイス VPC エンドポイントも作成して、そのエンドポイント経由で sigv4 が機能できるようにする必要があります。AWS STS の VPC エンドポイントを作成する方法の詳細については、「AWS Identity and Access Management ユーザーガイド」の「AWS STS インターフェイス VPC エンドポイントの使用」を参照してくださ

い。AWS STS は、リージョン化されたエンドポイントを使用するように設定する必要があります。

インターフェイス VPC エンドポイントの作成手順を含む詳細については、「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。

> **Note**
>
> VPC エンドポイントポリシーを使用すると、Amazon Managed Service for Prometheus インターフェイス VPC エンドポイントへのアクセスを制御できます。詳細については、次のセクションを参照してください。

Amazon Managed Service for Prometheus のインターフェイス VPC エンドポイントを作成済みで、VPC に配置されたワークスペースに既にデータが流れている場合、メトリクスはデフォルトでインターフェイス VPC エンドポイントを通じて送信されます。Amazon Managed Service for Prometheus は、パブリックエンドポイントまたはプライベートインターフェイスエンドポイント (どちらか使用中のもの) を使用してこのタスクを実行します。

Amazon Managed Service for Prometheus VPC エンドポイントへのアクセスの制御

VPC エンドポイントポリシーを使用すると、Amazon Managed Service for Prometheus インターフェイス VPC エンドポイントへのアクセスを制御できます。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。

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のアクセス制御」を参照してください。

Amazon Managed Service for Prometheus のエンドポイントポリシーの例を次に示します。このポリシーは、`PromUser` というロールを持ち、VPC 経由で Amazon Managed Service for Prometheus に接続するユーザーに、ワークスペースとルールグループの表示を許可しますが、例えば、ワークスペースの作成や削除は許可しません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AmazonManagedPrometheusPermissions",
            "Effect": "Allow",
            "Action": [
                "aps:DescribeWorkspace",
                "aps:DescribeRuleGroupsNamespace",
                "aps:ListRuleGroupsNamespace",
                "aps:ListWorkspaces"
            ],
            "Resource": "arn:aws:aps:*:*:/workspaces*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::111122223333:role/PromUser"
                ]
            }
        }
    ]
}
```

次の例は、指定した VPC 内の指定した IP アドレスから送信されたリクエストのみが成功するように許可するポリシーを示しています。他の IP アドレスからのリクエストは失敗します。

```
{
    "Statement": [
        {
            "Action": "aps:*",
            "Effect": "Allow",
            "Principal": "*",
            "Resource": "*",
            "Condition": {
                "IpAddress": {
                    "aws:VpcSourceIp": "192.0.2.123"
                },
        "StringEquals": {
                    "aws:SourceVpc": "vpc-555555555555"
                }
            }
        }
    ]
```

```
}
```

Amazon Managed Service for Prometheus エラーのトラブルシューティング

Amazon Managed Service for Prometheus に関する問題のトラブルシューティングを行うには、以下のセクションが役立ちます。

トピック

429 または制限超過エラー

次の例のような 429 エラーが表示される場合は、リクエストが Amazon Managed Service for Prometheus の取り込みのクォータを超えています。

```
ts=2020-10-29T15:34:41.845Z caller=dedupe.go:112 component=remote level=error
 remote_name=e13b0c
url=http://iamproxy-external.prometheus.uswest2-prod.eks:9090/workspaces/workspace_id/
api/v1/remote_write
msg="non-recoverable error" count=500 err="server returned HTTP status 429
Too Many Requests: ingestion rate limit (6666.666666666667) exceeded while adding 499
 samples and 0 metadata
```

次の例のような 429 エラーが表示される場合は、リクエストが Amazon Managed Service for Prometheus のワークスペースあたりのアクティブなメトリクス数のクォータを超えています。

```
ts=2020-11-05T12:40:33.375Z caller=dedupe.go:112 component=remote level=error
 remote_name=aps
url=http://iamproxy-external.prometheus.uswest2-prod.eks:9090/workspaces/workspace_id/
api/v1/remote_write
msg="non-recoverable error" count=500 err="server returned HTTP status 429 Too Many
 Requests: user=accountid_workspace_id:
```

```
per-user series limit (local limit: 0 global limit: 3000000 actual local limit: 500000)
 exceeded
```

次の例のような 429 エラーが表示された場合、リクエストが Prometheus 互換 API を使用してワークスペースにデータを送信できるレート (1 秒あたりのトランザクション数) の Amazon Managed Service for Prometheus `RemoteWrite` クォータを超えています。

```
ts=2024-03-26T16:50:21.780708811Z caller=dedupe.go:112 component=remote level=error
 remote_name=ab123c
url=https://aps-workspaces.us-east-1.amazonaws.com/workspaces/workspace_id/api/v1/
remote_write
msg="non-recoverable error" count=1000 exemplarCount=0 err="server returned HTTP status
 429 Too Many Requests: {\"message\":\"Rate exceeded\"}"
```

次の例のような 400 エラーが表示された場合、リクエストはアクティブな時系列の Amazon Managed Service for Prometheus クォータを超えています。アクティブな時系列クォータの処理方法の詳細については、「」を参照してくださいアクティブなシリーズ数のデフォルト。

```
ts=2024-03-26T16:50:21.780708811Z caller=push.go:53 level=warn
url=https://aps-workspaces.us-east-1.amazonaws.com/workspaces/workspace_id/api/v1/
remote_write
msg="non-recoverable error" count=500 exemplarCount=0
err="server returned HTTP status 400 Bad Request: maxFailure (quorum) on a given error
 family, rpc error: code = Code(400)
desc = addr=10.1.41.23:9095 state=ACTIVE zone=us-east-1a, rpc error: code = Code(400)
desc = user=accountid_workspace_id: per-user series limit of 10000000 exceeded,
Capacity from 2,000,000 to 10,000,000 is automatically adjusted based on the last 30
 min of usage.
If throttled above 10,000,000 or in case of incoming surges, please contact
 administrator to raise it.
(local limit: 0 global limit: 10000000 actual local limit: 92879)"
```

Amazon Managed Service for Prometheus のサービスクォータの詳細については、「Amazon Managed Service for Prometheus のサービスクォータ」を参照してください。

サンプルが重複している

高可用性の Prometheus グループを使用している場合は、Prometheus インスタンスで外部ラベルを使用して重複排除を設定する必要があります。詳細については、「Amazon Managed Service for Prometheus に送信される高可用性メトリクスの重複排除」を参照してください。

重複データに関するその他の問題については、次のセクションで説明します。

サンプルタイムスタンプに関するエラーが表示される

Amazon Managed Service for Prometheus はデータを順番に取り込み、各サンプルに前のサンプルよりも後のタイムスタンプがあることを期待します。

データが順番に到着しない場合、`out-of-order samples`、、`duplicate sample for timestamp`または に関するエラーが表示されます`samples with different value but same timestamp`。これらの問題は通常、Amazon Managed Service for Prometheus にデータを送信しているクライアントの誤った設定が原因で発生します。エージェントモードで実行されている Prometheus クライアントを使用している場合は、重複するシリーズ名または重複するターゲットを持つルールの設定を確認します。メトリクスでタイムスタンプが直接指定されている場合は、それらが順序が間違っていないことを確認します。

これがどのように機能するか、またはセットアップを確認する方法の詳細については、ブログ記事「Understanding Duplicate Samples and O ut-of-order Timestamp Errors in Prometheus from Prom Labs」を参照してください。

制限に関するエラーメッセージが表示される

> **Note**
>
> Amazon Managed Service for Prometheus は、Prometheus リソースCloudWatch の使用状況をモニタリングするための使用状況メトリクスを提供します。使用状況メトリクスアラーム機能を使用すると CloudWatch、Prometheus のリソースと使用状況をモニタリングして、制限エラーを防ぐことができます。

次のいずれかのエラーメッセージが表示される場合は、Amazon Managed Service for Prometheus のいずれかのクォータの引き上げをリクエストすることで問題を解決できます。詳細については、「Amazon Managed Service for Prometheus のサービスクォータ」を参照してください。

- ユーザーあたりのシリーズ制限 *<value>* を超えました。管理者に連絡して制限を引き上げてください
- メトリクスあたりのシリーズ制限 *<value>* を超えました。管理者に連絡して制限を引き上げてください
- ingestion rate limit (...) exceeded

- series has too many labels (...) series: '%s'
- the query time range exceeds the limit (query length: xxx, limit: yyy)
- the query hit the max number of chunks limit while fetching chunks from ingesters
- Limit exceeded. Maximum workspaces per account.

ローカル Prometheus サーバーの出力が制限を超えている

Amazon Managed Service for Prometheus には、ワークスペースが Prometheus サーバーから受信できるデータ量に対するサービスクォータがあります。Prometheus サーバーが Amazon Managed Service for Prometheus に送信しているデータ量を特定するには、Prometheus サーバーに対して以下のクエリを実行します。Prometheus の出力が Amazon Managed Service for Prometheus の制限を超えていると判明した場合は、対応するサービスクォータの引き上げをリクエストできます。詳細については、「Amazon Managed Service for Prometheus のサービスクォータ」を参照してください。

出力制限を確認するためにローカル自己実行 Prometheus サーバーに対して実行するクエリ

データの種類	使用するクエリ	
現在のアクティブなシリーズ数	`prometheus_tsdb_head_series`	
現在の取り込みレート	`rate(prometheus_tsdb_head_samples_appended_total[5m])`	
メトリクス名あたりのアクティブなシリーズの M ost-to-least リスト	`sort_desc(count by(__name__)({__name__!=""}))`	

データの種類	使用するクエリ	
メトリクスシリーズごとのラベル数	`group by(mylabelname) ({__name__!=""})`	

データの一部が表示されない

Amazon Managed Service for Prometheus に送信されるデータは、さまざまな理由で破棄できます。次の表は、データが取り込まれるのではなく破棄される理由を示しています。

Amazon を使用して、データが破棄される量と理由を追跡できます CloudWatch。詳細については、「CloudWatch メトリクスを使用して Amazon Managed Service for Prometheus リソースをモニタリングする」を参照してください。

理由	意味
greater_than_max_sample_age	現在の時刻より古いログ行の破棄
new-value-for-timestamp	重複するサンプルが以前の記録とは異なるタイムスタンプで送信されました
per_metric_series_limit	ユーザーがメトリクスごとのアクティブなシリーズ数の上限に達しました
per_user_series_limit	ユーザーがアクティブなシリーズの合計数の上限に達しました
rate_limited	取り込みレートが制限されました
sample-out-of-order	サンプルが順不同で送信されたため、処理できません
label_value_too_long	ラベル値の長さが許容される文字数の上限を超えています

理由	意味
max_label_names_per_series	ユーザーがメトリクスごとのラベル名の数の上限に達しました
missing_metric_name	メトリクス名が指定されていません
metric_name_invalid	無効なメトリクス名が指定されました
label_invalid	無効なラベルが指定されました
duplicate_label_names	重複するラベル名が指定されました

Amazon Managed Service for Prometheus でのタグ付け

タグは、 AWS リソース AWS に割り当てるカスタム属性ラベルです。各 AWS タグには 2 つの部分があります。

- タグキー (CostCenter、Environment、Project、Secret など)。タグキーでは、大文字と小文字が区別されます。
- タグ値と呼ばれるオプションのフィールド (111122223333、Production、チーム名など)。タグ値を省略すると、空の文字列を使用した場合と同じになります。タグキーと同様に、タグ値では大文字と小文字が区別されます。

これらを合わせて、キーと値のペアと呼ばれます。各ワークスペースには、最大 50 個のタグを割り当てることができます。

タグは、 AWS リソースの識別と整理に役立ちます。多くの AWS 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、Amazon S3 バケットに割り当てたものと同じタグを、Amazon Managed Service for Prometheus ワークスペースに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、「Tagging AWS Resources」を参照してください。

Amazon Managed Service for Prometheus では、ワークスペースとルールグループ名前空間の両方にタグを付けることができます。コンソール、、API AWS CLI、または SDKs 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。 APIs タグは、ワークスペースやルールグループの識別、整理、追跡に使用できるほか、IAM ポリシーで使用すると、Amazon Managed Service for Prometheus リソースを表示および操作できるユーザーを制御するために役立ちます。

タグの制限

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。
- キーまたは値のプレフィックスとして、aws:、AWS:、またはその大文字と小文字の組み合わせを変えたものは使用しないでください。これらは AWS 専用に予約されています。このプレフィックスを持つタグのキーや値を編集または削除することはできません。このプレフィックスが付いたタグは、 tags-per-resource 制限にはカウントされません。

トピック

Amazon Managed Service for Prometheus ワークスペースにタグを付ける

タグは、リソースに割り当てることができるカスタムラベルです。一意のキーとオプションの値 (キーと値のペア) が含まれます。タグは、 AWS リソースの識別や整理に役立ちます。Amazon Managed Service for Prometheus では、ワークスペース (およびルールグループ名前空間) にタグを付けることができます。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 Service for Prometheus 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。

Amazon Managed Service for Prometheus 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。

トピック

- ワークスペースからのタグの削除

ワークスペースへのタグの追加

Amazon Managed Service for Prometheus ワークスペースにタグを追加すると、 AWS リソースの識別と整理、アクセスの管理に役立ちます。まず、ワークスペースに 1 つ以上のタグ (キーと値のペア) を追加します。タグを追加したら、そ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 を使用して AWS CLI 、Amazon Managed Service for Prometheus ワークスペースにタグを追加できます。

> **Important**
>
> ワークスペースにタグを追加すると、そのワークスペースへのアクセスに影響が生じる可能性があります。ワークスペースにタグを追加する前に、タグを使用してリソースへのアクセスを制御している可能性のある IAM ポリシーをすべて確認してください。

Amazon Managed Service for Prometheus ワークスペースの作成時にタグを追加する方法の詳細については、「Amazon Managed Service for Prometheus ワークスペースを作成する」を参照してください。

トピック

- ワークスペースへのタグの追加 (コンソール)
- ワークスペースへのタグの追加 (AWS CLI)

ワークスペースへのタグの追加 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus ワークスペースに 1 つ以上のタグを追加できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. Amazon Managed Service for Prometheus ワークスペースにタグが追加されていない場合は、[タグを作成] を選択します。それ以外の場合は、[タグを管理] を選択します。
7. [キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。
8. (オプション) 別のタグを追加するには、[タグを追加] を再度選択します。
9. タグの追加が完了したら、[変更を保存] を選択します。

ワークスペースへのタグの追加 (AWS CLI)

を使用して Amazon Managed Service for Prometheus ワークスペースにタグ AWS CLI を追加するには、次の手順に従います。ワークスペースの作成時にタグを追加するには、「Amazon Managed Service for Prometheus ワークスペースを作成する」を参照してください。

これらのステップでは、 の最新バージョンが既にインストールされているか、最新バージョンに AWS CLI 更新されていることを前提としています。詳細については、「AWS Command Line Interfaceのインストール」を参照してください。

ターミナルまたはコマンドラインで、tag-resource コマンドを実行して、タグを追加するワークスペースの Amazon リソースネーム (ARN) と、追加するタグのキーおよび値を指定します。ワークスペースには 1 つ以上のタグを追加できます。例えば、My-Workspace という名前の Amazon Managed Service for Prometheus ワークスペースを、タグキーが *Status* でタグ値が *Secret* のタグと、タグキーが *Team* でタグ値が *My-Team* のタグの 2 つでタグ付けするには、次のように入力します。

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring
--tags Status=Secret,Team=My-Team
```

成功した場合、このコマンドは何も返しません。

ワークスペースのタグの表示

タグは、 AWS リソースを識別して整理し、リソースへのアクセスを管理するのに役立ちます。タグ付け戦略の詳細については、「 AWS リソースのタグ付け」を参照してください。

Amazon Managed Service for Prometheus ワークスペースのタグの表示 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus ワークスペースに関連付けられているタグを表示できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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。

Amazon Managed Service for Prometheus ワークスペースのタグの表示 (AWS CLI)

を使用してワークスペースの AWS タグ AWS CLI 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、次のように入力します。

```
aws amp list-tags-for-resource --resource-arn arn:aws:aps:us-
west-2:123456789012: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

ワークスペースに関連付けられているタグの値を変更できます。キーの名前を変更することもできます。これは、現在のタグを削除し、新しい名前を使用して、元のキーと同じ値を持つタグを追加することと同等です。

> **Important**
>
> Amazon Managed Service for Prometheus 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が生じる可能性があります。ワークスペースのタグの名前

(キー) または値を編集する前に、タグのキーや値を使用してリポジトリなどのリソースへのアクセスを制御する可能性のある IAM ポリシーを必ず確認してください。

Amazon Managed Service for Prometheus ワークスペースのタグの編集 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus ワークスペースに関連付けられているタグを編集できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. ワークスペースにタグが追加されていない場合は、[タグを作成] を選択します。それ以外の場合は、[タグを管理] を選択します。
7. [キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。
8. (オプション) 別のタグを追加するには、[タグを追加] を再度選択します。
9. タグの追加が完了したら、[変更を保存] を選択します。

Amazon Managed Service for Prometheus ワークスペースのタグの編集 (AWS CLI)

を使用してワークスペースのタグ AWS CLI を更新するには、次の手順に従います。既存のキーの値を変更したり、別のキーを追加したりできます。

ターミナルまたはコマンドラインで、tag-resource コマンドを実行して、タグを更新する Amazon Managed Service for Prometheus ワークスペースの Amazon リソースネーム (ARN) と、タグキーおよびタグ値を指定します。

```
aws amp tag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tags Team=New-Team
```

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 Service for Prometheus ワークスペースのタグを削除すると、そのワークスペースへのアクセスに影響が生じ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、タグのキーや値を使用してリポジトリなどのリソースへのアクセスを制御する可能性のある IAM ポリシーを必ず確認してください。

Amazon Managed Service for Prometheus ワークスペースからのタグの削除 (コンソール)

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. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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、[削除] を選択します。

Amazon Managed Service for Prometheus ワークスペースからのタグの削除 (AWS 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。タグを削除してもそのタグがなくなるわけではありません。タグとワークスペースとの関連付けが解除されるだけです。

> **Note**
>
> Amazon Managed Service for Prometheus ワークスペースを削除すると、削除されたワークスペースからタグの関連付けがすべて解除されます。ワークスペースを削除する前にタグを削除する必要はありません。

ターミナルまたはコマンドラインで、untag-resource コマンドを実行して、タグを削除するワークスペースの Amazon リソースネーム (ARN) と、削除するタグのタグキーを指定します。例えば、My-Workspace という名前のワークスペースから *Status* というタグキーを持つタグを削除するには、次のように入力します。

```
aws amp untag-resource --resource-arn arn:aws:aps:us-
west-2:123456789012:workspace/IDstring --tag-keys Status
```

成功した場合、このコマンドは何も返しません。ワークスペースに関連付けられているタグを確認するには、list-tags-for-resource コマンドを実行します。

ルールグループ名前空間のタグ付け

タグは、リソースに割り当てることができるカスタムラベルです。一意のキーとオプションの値 (キーと値のペア) が含まれます。タグは、 AWS リソースの識別や整理に役立ちます。Amazon Managed Service for Prometheus では、ルールグループ名前空間 (およびワークスペース) にタグを付けることができます。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 タグを使用してルールグループ名前空間を識別、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 Service for Prometheus 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。

Amazon Managed Service for Prometheus のルールグループ名前空間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。

トピック

- ルールグループ名前空間へのタグの追加
- ルールグループ名前空間のタグの表示
- ルールグループ名前空間のタグの編集
- ルールグループ名前空間からのタグの削除

ルールグループ名前空間へのタグの追加

Amazon Managed Service for Prometheus ルールグループ名前空間にタグを追加すると、 AWS リソースを識別して整理し、リソースへのアクセスを管理するのに役立ちます。まず、ルールグループ名前空間に 1 つ以上のタグ (キーと値のペア) を追加します。タグを追加した後、IAM ポリシーを作成して、それらのタグに基づいて名前空間へのアクセスを管理できます。コンソールまたは を使用

して、Amazon Managed Service for Prometheus ルールグループ名前空間にタグ AWS CLI を追加できます。

> **Important**
>
> ルールグループ名前空間にタグを追加すると、そのルールグループ名前空間へのアクセスに影響が生じる可能性があります。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。

ルールグループ名前空間の作成時にタグを追加する方法の詳細については、「ルールファイルを作成する」を参照してください。

トピック

- ルールグループ名前空間へのタグの追加 (コンソール)
- ルールグループ名前空間へのタグの追加 (AWS CLI)

ルールグループ名前空間へのタグの追加 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus のルールグループ名前空間に 1 つ以上のタグを追加できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. [ルール管理] タブを選択します。
6. 名前空間名の横にあるボタンを選択し、[編集] を選択します。
7. [タグを作成]、[新しいタグを追加] を選択します。
8. [キー] にタグの名前を入力します。[値] では、任意でタグに値を追加できます。
9. (オプション) 別のタグを追加するには、[新しいタグを追加] を再度選択します。
10. タグの追加が完了したら、[変更を保存] を選択します。

ルールグループ名前空間へのタグの追加 (AWS CLI)

を使用して Amazon Managed Service for Prometheus ルールグループ名前空間にタグ AWS CLI を追加するには、次の手順に従います。ルールグループ名前空間の作成時にタグを追加するには、「Amazon Managed Service for Prometheus にルール設定ファイルをアップロードする」を参照してください。

これらのステップでは、 の最新バージョンが既にインストールされているか、最新バージョンに AWS CLI 更新されていることを前提としています。詳細については、「AWS Command Line Interfaceのインストール」を参照してください。

ターミナルまたはコマンドラインで、tag-resource コマンドを実行して、タグを追加するルールグループ名前空間の Amazon リソースネーム (ARN) と、追加するタグのキーおよび値を指定します。ルールグループ名前空間には複数のタグを追加できます。例えば、My-Workspace という名前の Amazon Managed Service for Prometheus 名前空間を、タグキーが *Status* でタグ値が *Secret* のタグと、タグキーが *Team* でタグ値が *My-Team* のタグの 2 つでタグ付けするには、次のように入力します。

```
aws amp tag-resource \
    --resource-arn arn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name \
    --tags Status=Secret,Team=My-Team
```

成功した場合、このコマンドは何も返しません。

ルールグループ名前空間のタグの表示

タグは、 AWS リソースを識別して整理し、リソースへのアクセスを管理するのに役立ちます。タグ付け戦略の詳細については、「 AWS リソースのタグ付け」を参照してください。

Amazon Managed Service for Prometheus ルールグループ名前空間のタグの表示 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus のルールグループ名前空間に関連付けられているタグを表示できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. [ルール管理] タブを選択します。
6. 名前空間名を選択します。

Amazon Managed Service for Prometheus ワークスペースのタグの表示 (AWS CLI)

を使用してルールグループ名前空間の AWS タグ AWS CLI 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、ルールグループ名前空間のタグキーとタグ値の一覧を表示するには、次のように入力します。

```
aws amp list-tags-for-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name
```

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"
    }
}
```

ルールグループ名前空間のタグの編集

ルールグループ名前空間に関連付けられているタグの値を変更できます。キーの名前を変更することもできます。これは、現在のタグを削除し、新しい名前を使用して、元のキーと同じ値を持つタグを追加することと同等です。

> **Important**
>
> ルールグループ名前空間のタグを編集すると、その名前空間へのアクセスに影響が生じる可能性があります。リソースのタグの名前 (キー) または値を編集する前に、タグのキーや値を使用して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。

Amazon Managed Service for Prometheus ルールグループ名前空間のタグの編集 (コンソール)

コンソールを使用して、Amazon Managed Service for Prometheus のルールグループ名前空間に関連付けられているタグを編集できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. [ルール管理] タブを選択します。
6. 名前空間の名前を選択します。
7. [タグを管理] を選択し、[新しいタグを追加] を選択します。
8. 既存のタグの値を変更するには、[値] に新しい値を入力します。
9. 他のタグを追加するには、[新しいタグを追加] を選択します。
10. タグの追加と編集が完了したら、[変更を保存] を選択します。

Amazon Managed Service for Prometheus ルールグループ名前空間のタグの編集 (AWS CLI)

を使用してルールグループ名前空間のタグ AWS CLI を更新するには、次の手順に従います。既存のキーの値を変更したり、別のキーを追加したりできます。

ターミナルまたはコマンドラインで、tag-resource コマンドを実行して、タグを更新するリソースの Amazon リソースネーム (ARN) と、タグキーおよびタグ値を指定します。

```
aws amp tag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tags Team=New-Team
```

ルールグループ名前空間からのタグの削除

ルールグループ名前空間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**
>
> リソースのタグを削除すると、そのリソースへのアクセスに影響が生じる可能性があります。リソースからタグを削除する前に、タグのキーや値を使用して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。

Amazon Managed Service for Prometheus ルールグループ名前空間のタグの削除 (コンソール)

コンソールを使用して、タグとルールグループ名前空間との関連付けを解除できます。

1. Amazon Managed Service for Prometheus コンソール (https://console.aws.amazon.com/prometheus/) 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. 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. [ルール管理] タブを選択します。
6. 名前空間の名前を選択します。
7. [Manage tags (タグの管理)] を選択します。
8. 削除するタグの横にある [削除] を選択します。
9. 完了したら、[変更を保存] を選択します。

Amazon Managed Service for Prometheus ルールグループ名前空間のタグの削除 (AWS CLI)

を使用してルールグループ名前空間からタグ AWS CLI を削除するには、次の手順に従います。タグを削除してもそのタグがなくなるわけではありません。タグとルールグループ名前空間との関連付けが解除されるだけです。

> **Note**
>
> Amazon Managed Service for Prometheus のルールグループ名前空間を削除すると、削除された名前空間からタグの関連付けがすべて解除されます。名前空間を削除する前にタグを削除する必要はありません。

ターミナルまたはコマンドラインで、untag-resource コマンドを実行して、タグを削除するルールグループ名前空間の 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、My-Workspace という名前のワークスペースから *Status* というタグキーを持つタグを削除するには、次のように入力します。

```
aws amp untag-resource --resource-arn rn:aws:aps:us-
west-2:123456789012:rulegroupsnamespace/IDstring/namespace_name --tag-keys Status
```

成功した場合、このコマンドは何も返しません。リソースに関連付けられているタグを確認するには、list-tags-for-resource コマンドを実行します。

Amazon Managed Service for Prometheus のサービスクォータ

以下の 2 つのセクションでは、Amazon Managed Service for Prometheus に関連付けられているクォータと制限について説明します。

Service Quotas

Amazon Managed Service for Prometheus には、以下に示すクォータがあります。Amazon Managed Service for Prometheus は、Prometheus リソースCloudWatch の使用状況をモニタリングするための使用状況メトリクスを提供します。 CloudWatch 使用状況メトリクスアラーム機能を使用すると、Prometheus リソースと使用状況をモニタリングして、制限エラーを防ぐことができます。

プロジェクトやワークスペースの拡大に伴い、モニタリングや引き上げリクエストが必要になる最も一般的なクォータは、ワークスペースごとのアクティブなシリーズ数、ワークスペースごとの取り込みレート、ワークスペースごとの取り込みバーストサイズです。

すべての調整可能なクォータでは、[調整可能] 列のリンクを選択するか、クォータの引き上げをリクエストすることで、クォータの引き上げをリクエストできます。

ワークスペースごとのアクティブシリーズ数の制限は動的に適用されます。詳細については、「アクティブなシリーズ数のデフォルト」を参照してください。ワークスペースあたりの取り込み速度とワークスペースあたりの取り込みバーストサイズは、ワークスペースへのデータ取り込み速度をまとめて制御します。詳細については、「取り込みスロットリング」を参照してください。

> **Note**
>
> 特に明記されていない限り、これらのクォータはワークスペースごとに設定されます。

名前	デフォルト	引き上げ可能	説明
ワークスペースごとのメタデータを持つアクティブなメトリクス数	サポートされている各リージョン: 20,000	不可	ワークスペースごとの、メタデータを持つ一意のアクティブなメトリクスの数。注: 制限に達すると、メトリクスサンプルが記録されますが、制限を超えるメタデータは削除されます。
ワークスペースごとのアクティブなシリーズ数	サポートされている各リージョン: 10,000,000/2 時間	可能	ワークスペースごとの一意のアクティブなシリーズの数。過去 2 時間以内にサンプルが報告された場合、そのシリーズはアクティブです。2 M から 10 M までのキャパシティは、過去 30 分間の使用状況に基づいて自動的に調整されます。
アラートマネージャー定義ファイル内のアラート集約グループのサイズ	サポートされている各リージョン: 1,000	可能	アラートマネージャー定義ファイル内のアラート集約グループの最大サイズ。group_by のラベル値の組み合わせごとに集約グループが作成されます。

名前	デフォルト	引き上げ可能	説明
アラートマネージャー定義ファイルのサイズ	サポートされている各リージョン: 1 メガバイト	不可	アラートマネージャー定義ファイルの最大サイズ。
Alert Manager のアラートペイロードサイズ	サポートされている各リージョン: 20 メガバイト	不可	ワークスペースあたりのすべての Alert Manager アラートの最大アラートペイロードサイズ。アラートのサイズは、ラベルと注釈に依存します。
アラートマネージャーのアラート	サポートされている各リージョン: 1,000	可能	ワークスペースあたりの同時 Alert Manager アラートの最大数。
HA トラッカーのクラスター数	サポートされている各リージョン: 500	不可	ワークスペースごとの、取り込まれたサンプルについて HA トラッカーが追跡するクラスターの最大数。
ワークスペースごとの取り込みバーストサイズ	サポートされている各リージョン: 1,000,000	可能	ワークスペースごとに 1 回のバーストで 1 秒あたりに取り込むことができるサンプルの最大数。
ワークスペースごとの取り込みレート	サポートされている各リージョン: 170,000	可能	ワークスペースごとの、1 秒あたりのメトリクスサンプルの取り込みレート。

名前	デフォルト	引き上げ可能	説明
アラートマネージャー定義ファイル内の禁止ルール数	サポートされている各リージョン: 100	可能	アラートマネージャー定義ファイル内の禁止ルールの最大数。
ラベルサイズ	サポートされている各リージョン: 7 キロバイト	不可	1 つのシリーズで許容される、すべてのラベルとラベル値を合わせた最大サイズ。
メトリクスシリーズごとのラベル数	サポートされている各リージョン: 70	可能	メトリクスシリーズごとのラベルの数。
メタデータの長さ	サポートされている各リージョン: 1 キロバイト	不可	メトリクスのメタデータに許容される最大長。メタデータは、メトリクス名、タイプ、単位、ヘルプテキストを指します。
メトリクスごとのメタデータ数	サポートされている各リージョン: 10	不可	メトリクスごとのメタデータの数。
アラートマネージャーのルーティングツリー内のノード数	サポートされている各リージョン: 100	可能	アラートマネージャーのルーティングツリー内の最大ノード数。

名前	デフォルト	引き上げ可能	説明
1 秒あたりのトランザクションにおけるリージョンあたりのAPIオペレーション数	サポートされている各リージョン: 10	可能	リージョンあたりの 1 秒あたりのAPIオペレーションの最大数。これにはAPIs、ワークスペース CRUD 、タグ付け APIs、ルールグループ名前空間 CRUD APIs、アラートマネージャー定義 CRUD が含まれますAPIs。
1 秒あたりのトランザクションのワークスペースあたりの GetSeries、GetLabels 、および GetMetricMetadata API オペレーションの数	サポートされている各リージョン: 10	不可	ワークスペースあたりの 1 秒あたりの GetSeries 、 GetLabels および GetMetricMetadata Prometheus 互換APIオペレーションの最大数。
1 秒あたりのトランザクションにおけるワークスペースあたりのオペレーション数 QueryMetrics API	サポートされている各リージョン: 300	不可	ワークスペースあたりの 1 秒あたりの QueryMetrics Prometheus 互換APIオペレーションの最大数。
1 秒あたりのトランザクションにおけるワークスペースあたりのオペレーション数 RemoteWrite API	サポートされている各リージョン: 3,000	不可	ワークスペースあたりの 1 秒あたりの RemoteWrite Prometheus 互換APIオペレーションの最大数。

名前	デフォルト	引き上げ可能	説明
1 秒あたりのトランザクションにおけるワークスペースあたりの他の Prometheus 互換APIオペレーションの数	サポートされている各リージョン: 100	不可	ListAlerts ListRulesなどAPIsを含む他のすべての Prometheus 互換のワークスペースあたりの 1 秒あたりのAPIオペレーションの最大数。
インスタントクエリのクエリバイト数	サポートされている各リージョン: 5 ギガバイト	不可	750MB は 1 回のインスタントクエリでスキャンできます。
範囲クエリのクエリバイト数	サポートされている各リージョン: 5 ギガバイト	不可	1 つの範囲クエリで 24 時間ごとにスキャンできる最大バイト数。
フェッチされるクエリチャンク数	サポートされている各リージョン: 20,000,000	不可	1 つのクエリでスキャンできるチャンクの最大数。
クエリサンプル数	サポートされている各リージョン: 50,000,000	不可	1 つのクエリでスキャンできるサンプルの最大数。
フェッチされるクエリシリーズ数	サポートされている各リージョン: 12,000,000	不可	1 つのクエリでスキャンできるシリーズの最大数。
クエリ時間範囲の日数	サポートされている各リージョン: 32	不可	QueryMetrics、GetSeries、および の最大時間範囲 GetLabels APIs。

名前	デフォルト	引き上げ可能	説明
リクエストサイズ	サポートされている各リージョン: 1 メガバイト	不可	取り込みまたはクエリの最大リクエストサイズ。
取り込んだデータの保持期間の日数	サポートされている各リージョン: 150	可能	ワークスペースのデータが保持される日数。これより古いデータは削除されます。クォータの変更をリクエストして、この値を増減できます。
ルール評価間隔	サポートされている各リージョン: 30 秒	可能	最小ルール評価間隔。
ルールグループ名前空間定義ファイルのサイズ	サポートされている各リージョン: 1 メガバイト	不可	ルールグループ名前空間定義ファイルの最大サイズ。
ワークスペースごとのルール数	サポートされている各リージョン: 2,000	可能	ワークスペースごとのルールの最大数。
アラートマネージャー定義ファイル内のテンプレート数	サポートされている各リージョン: 100	可能	アラートマネージャー定義ファイル内のテンプレートの最大数。
アカウントごとのリージョンあたりのワークスペース数	サポートされている各リージョン: 25	可能	リージョンあたりのワークスペースの最大数。

アクティブなシリーズ数のデフォルト

Amazon Managed Service for Prometheus では、デフォルトでアクティブな時系列数のクォータまで使用することが許容されます。

Amazon Managed Service for Prometheus のワークスペースは、取り込み量に自動的に適応します。使用量の増加に応じて、Amazon Managed Service for Prometheus は時系列のキャパシティを自動的に増やし、デフォルトのクォータを上限としてベースラインの使用量を 2 倍にします。例えば、過去 30 分間のアクティブな時系列数の平均が 350 万の場合、最大 700 万の時系列がスロットリングなしで使用可能になります。

前のベースラインの 2 倍以上が必要な場合、Amazon Managed Service for Prometheus は取り込み量の増加に合わせてクォータまで自動的にキャパシティの割り当てを増やし、ワークロードでスロットリングが発生し続けないようにします。ただし、過去 30 分間に計算された前のベースラインの 2 倍を超えると、スロットリングが発生する可能性があります。スロットリングを避けるため、Amazon Managed Service for Prometheus では、以前のアクティブな時系列数の 2 倍以上に増加する場合は、取り込み量を徐々に増やすことを推奨しています。

> **Note**
>
> アクティブな時系列の最小キャパシティは 200 万です。時系列が 200 万未満の場合、スロットリングは発生しません。
> デフォルトのクォータを超えるには、クォータの引き上げをリクエストできます。

取り込みスロットリング

Amazon Managed Service for Prometheus は、現在の制限に基づいて、各ワークスペースの取り込みを調整します。これにより、ワークスペースのパフォーマンスを維持できます。制限を超えると、 CloudWatch メトリクス`DiscardedSamples`に が表示されます (`rate_limited`理由付き)。Amazon を使用して取り込みを CloudWatch モニタリングし、スロットリング制限に近づいたときに警告するアラームを作成できます。詳細については、「CloudWatch メトリクスを使用して Amazon Managed Service for Prometheus リソースをモニタリングする」を参照してください。

Amazon Managed Service for Prometheus は、トークンバケットアルゴリズムを使用して取り込みスロットリングを実装します。このアルゴリズムでは、アカウントには、特定の数のトークンを保持するバケットがあります。バケット内のトークンの数は、任意の秒の取り込み制限を表します。

取り込まれた各データサンプルは、バケットから 1 つのトークンを削除します。バケットサイズ (ワークスペースあたりの取り込みバーストサイズ) が 1,000,000 の場合、ワークスペースは 1 秒あたり 100 万個のデータサンプルを取り込みます。取り込むサンプルが 100 万個を超えると、スロットリングされ、それ以上レコードを取り込むことはありません。追加のデータサンプルは破棄されます。

バケットは、設定されたレートで自動的に補充されます。バケットが最大容量を下回ると、最大容量に達するまで 1 秒ごとに一定数のトークンがバケットに追加されます。リフィルトークンが到着したときにバケットがいっぱいになると、バケットは破棄されます。バケットはトークンの最大数を超えて保持することはできません。サンプル取り込みのリフィルレートは、ワークスペースあたりの取り込みレートの制限によって設定されます。ワークスペースあたりの取り込みレートが 170,000 に設定されている場合、バケットのリフィルレートは 170,000 トークン/秒です。

ワークスペースが 1 秒あたり 1,000,000 個のデータサンプルを取り込みた場合、バケットはすぐにゼロトークンに削減されます。その後、バケットは最大容量の 1,000,000 トークンに達するまで、1 秒あたり 170,000 トークン補充されます。それ以上の取り込みがない場合、以前に空のバケットは 6 秒で最大容量に戻ります。

> **Note**
>
> 取り込みはバッチリクエストで行われます。使用可能なトークンが 100 個あり、サンプルが 101 個のリクエストを送信すると、リクエスト全体が拒否されます。Amazon Managed Service for Prometheus はリクエストを部分的に受け入れません。コレクターを記述する場合は、再試行を管理できます (バッチが小さいか、しばらく経過した後) 。

ワークスペースがより多くのデータサンプルを取り込むまで、バケットがいっぱいになるまで待つ必要はありません。トークンはバケットに追加されたときに使用できます。すぐにリフィルトークンを使用すると、バケットは最大容量に達しません。例えば、バケットを枯渇させた場合、1 秒あたり 170,000 個のデータサンプルを引き続き取り込むことができます。バケットは、1 秒あたり 170,000 個未満のデータサンプルを取り込む場合にのみ、最大容量まで補充できます。

取り込まれるデータに関する追加の制限

Amazon Managed Service for Prometheus には、ワークスペースに取り込まれるデータに関して次の要件があります。これらは調整できません。

- 1 時間以上経過したメトリクスサンプルは取り込まれません。

- すべてのサンプルとメタデータにメトリクス名が必要です。

Amazon Managed Service for Prometheus API リファレンス

Amazon Managed Service for Prometheus には、次の 2 種類の APIs。

1. Amazon Managed Service for Prometheus APIs – これらの APIs を使用すると、ワークスペース、スクレイパー、アラートマネージャー定義、ルールグループ名前空間、ログ記録のオペレーションなど、Amazon Managed Service for Prometheus ワークスペースを作成および管理できます。これらの API を操作するには、さまざまなプログラミング言語で使用できる AWS SDKs を使用します。 APIs
2. Prometheus 互換 APIs – Amazon Managed Service for Prometheus は、Prometheus と互換性のある HTTP APIs をサポートしています。これらの APIsにより、カスタムアプリケーションの構築、ワークフローの自動化、他の のサービスやツールとの統合、Prometheus クエリ言語 (PromQL) を使用したモニタリングデータのクエリと操作が可能になります。

このセクションでは、Amazon Managed Service for Prometheus でサポートされる API オペレーションとデータ構造の一覧を示します。

シリーズ、ラベル、API リクエストのクォータについては、Amazon Managed Service for Prometheus ユーザーガイドの「Amazon Managed Service for Prometheus サービスクォータ」を参照してください。

トピック

- Amazon Managed Service for Prometheus API
- Prometheus 互換 API

Amazon Managed Service for Prometheus API

Amazon Managed Service for Prometheus は、Amazon Managed Service for Prometheus ワークスペースを作成および管理する API オペレーションを提供します。これには、ワークスペース、スクレイパー、アラートマネージャー定義、ルールグループ名前空間、ログ記録用の APIs が含まれます。

Amazon Managed Service for Prometheus APIs「Amazon Managed Service for Prometheus API リファレンス」を参照してください。

AWS SDK での Amazon Managed Service for Prometheus の使用

AWS Software Development Kit (SDKsは、多くの一般的なプログラミング言語で使用できます。各 SDK には、開発者が好みの言語で AWS アプリケーションを簡単に構築できるようにする API、コード例、およびドキュメントが用意されています。言語別の SDKs「 デベロッパーセンター」の「 で構築するツール AWS AWS 」を参照してください。

> SDK のバージョン
>
> プロジェクトで使用する AWS SDK および他の SDKsの最新のビルドを使用し、SDKsを最新の状態に保つことをお勧めします。 AWS SDK には、最新の特長と機能に加え、セキュリティアップデートも含まれています。

Prometheus 互換 API

Amazon Managed Service for Prometheus では、以下の Prometheus 互換 API がサポートされています。

Prometheus 互換 APIs「」を参照してくださいPrometheus 互換を使用したクエリ APIs。

トピック

- CreateAlertManagerAlerts
- DeleteAlertManagerSilence
- GetAlertManagerStatus
- GetAlertManagerSilence
- GetLabels
- GetMetricMetadata
- GetSeries
- ListAlerts
- ListAlertManagerAlerts
- ListAlertManagerAlertGroups
- ListAlertManagerReceivers
- ListAlertManagerSilences

- ListRules
- PutAlertManagerSilences
- QueryMetrics
- RemoteWrite

CreateAlertManagerAlerts

`CreateAlertManagerAlerts` オペレーションは、ワークスペースにアラートを作成します。

有効な HTTP 動詞:

`POST`

有効な URI:

`/workspaces/workspaceId/alertmanager/api/v2/alerts`

URL クエリパラメータ:

`alerts` オブジェクトの配列。各オブジェクトは 1 つのアラートを表します。アラートブジェクトの例を以下に示します。

```
[
  {
    "startsAt": "2021-09-24T17:14:04.995Z",
    "endsAt": "2021-09-24T17:14:04.995Z",
    "annotations": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "labels": {
      "additionalProp1": "string",
      "additionalProp2": "string",
      "additionalProp3": "string"
    },
    "generatorURL": "string"
  }
]
```

リクエスト例

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts
 HTTP/1.1
Content-Length: 203,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0

[
  {
    "labels": {
      "alertname": "test-alert"
    },
    "annotations": {
      "summary": "this is a test alert used for demo purposes"
    },
    "generatorURL": "https://www.amazon.com/"
  }
]
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

DeleteAlertManagerSilence

`DeleteSilence` は、1 つのアラートサイレンスを削除します。

有効な HTTP 動詞:

`DELETE`

有効な URI:

`/workspaces/`*`workspaceId`*`/alertmanager/api/v2/silence/`*`silenceID`*

URL クエリパラメータ: なし

リクエスト例

```
DELETE /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silence/
d29d9df3-9125-4441-912c-70b05f86f973 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

GetAlertManagerStatus

GetAlertManagerStatus は、アラートマネージャーのステータスに関する情報を取得します。

有効な HTTP 動詞:

GET

有効な URI:

/workspaces/*workspaceId*/alertmanager/api/v2/status

URL クエリパラメータ: なし

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/status
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 941
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "cluster": null,
    "config": {
        "original": "global:\n  resolve_timeout: 5m\n  http_config:\n
 follow_redirects: true\n  smtp_hello: localhost\n  smtp_require_tls: true\nroute:
\n  receiver: sns-0\n  group_by:\n  - label\n  continue: false\nreceivers:\n-
 name: sns-0\n  sns_configs:\n  - send_resolved: false\n    http_config:\n
      follow_redirects: true\n    sigv4: {}\n    topic_arn: arn:aws:sns:us-
west-2:123456789012:test\n    subject: '{{ template \"sns.default.subject\" . }}'\n
    message: '{{ template \"sns.default.message\" . }}'\n    workspace_arn:
 arn:aws:aps:us-west-2:123456789012:workspace/ws-58a6a446-5ec4-415b-9052-a449073bbd0a
\ntemplates: []\n"
    },
    "uptime": null,
    "versionInfo": null
}
```

GetAlertManagerSilence

GetAlertManagerSilence は、1 つのアラートサイレンスに関する情報を取得します。

有効な HTTP 動詞:

GET

有効な URI:

/workspaces/*workspaceId*/alertmanager/api/v2/silence/*silenceID*

URL クエリパラメータ: なし

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silence/
d29d9df3-9125-4441-912c-70b05f86f973 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 310
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "id": "d29d9df3-9125-4441-912c-70b05f86f973",
    "status": {
        "state": "active"
    },
    "updatedAt": "2021-10-22T19:32:11.763Z",
    "comment": "hello-world",
    "createdBy": "test-person",
    "endsAt": "2023-07-24T01:05:36.000Z",
    "matchers": [
        {
            "isEqual": true,
            "isRegex": true,
            "name": "job",
            "value": "hello"
        }
    ],
    "startsAt": "2021-10-22T19:32:11.763Z"
}
```

GetLabels

`GetLabels` オペレーションは、時系列に関連付けられているラベルを取得します。

有効な HTTP 動詞:

GET, POST

有効な URI:

/workspaces/*workspaceId*/api/v1/labels

/workspaces/*workspaceId*/api/v1/label/*label-name*/values この URI は GET リクエストのみをサポートします。

URL クエリパラメータ:

match[]=<series_selector> ラベル名を読み取るシリーズを選択する、シリーズセレクターの繰り返しを含む引数。オプション。

start=<rfc3339 | unix_timestamp> 開始タイムスタンプ。オプション。

end=<rfc3339 | unix_timestamp> 終了タイムスタンプ。オプションです。

/workspaces/workspaceId/api/v1/labels のサンプルリクエスト

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/labels HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

/workspaces/workspaceId/api/v1/labels のサンプル応答

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 1435
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "status": "success",
    "data": [
        "__name__",
```

```
        "access_mode",
        "address",
        "alertname",
        "alertstate",
        "apiservice",
        "app",
        "app_kubernetes_io_instance",
        "app_kubernetes_io_managed_by",
        "app_kubernetes_io_name",
        "area",
        "beta_kubernetes_io_arch",
        "beta_kubernetes_io_instance_type",
        "beta_kubernetes_io_os",
        "boot_id",
        "branch",
        "broadcast",
        "buildDate",
        ...
    ]
}
```

/workspaces/workspaceId/api/v1/label/label-name/values のサンプルリクエスト

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/label/access_mode/values
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

/workspaces/workspaceId/api/v1/label/label-name/values のサンプル応答

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 74
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "status": "success",
```

```
    "data": [
        "ReadWriteOnce"
    ]
}
```

GetMetricMetadata

GetMetricMetadata オペレーションは、現在ターゲットからスクレイピングされているメトリクスに関するメタデータを取得します。ターゲット情報は提供されません。

クエリ結果のデータセクションはオブジェクトで構成されます。各オブジェクトのキーはメトリクス名を表し、値には、そのメトリクス名で公開されている固有のメタデータオブジェクトが、すべてのターゲットにわたるリストとして含まれます。

有効な HTTP 動詞:

GET

有効な URI:

/workspaces/*workspaceId*/api/v1/metadata

URL クエリパラメータ:

limit=<number> 取得するメトリクスの最大数。

metric=<string> メタデータをフィルタリングするメトリクス名。空のままにすると、すべてのメトリクスメタデータが取得されます。

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/metadata HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
```

```
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
Transfer-Encoding: chunked

{
    "status": "success",
    "data": {
        "aggregator_openapi_v2_regeneration_count": [
            {
                "type": "counter",
                "help": "[ALPHA] Counter of OpenAPI v2 spec regeneration count broken
 down by causing APIService name and reason.",
                "unit": ""
            }
        ],
        ...
    }
}
```

GetSeries

GetSeries オペレーションは、特定のラベルセットに一致する時系列のリストを取得します。

有効な HTTP 動詞:

GET, POST

有効な URI:

/workspaces/*workspaceId*/api/v1/series

URL クエリパラメータ:

match[]=<series_selector> 取得するシリーズを選択する、シリーズセレクターの繰り返しを含む引数。少なくとも 1 つの match[] 引数を指定する必要があります。

start=<rfc3339 | unix_timestamp> 開始タイムスタンプ。オプションです。

end=<rfc3339 | unix_timestamp> 終了タイムスタンプ。オプションです。

リクエスト例

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/series --data-urlencode
 'match[]=node_cpu_seconds_total{app="prometheus"}' --data-urlencode 'start=1634936400'
 --data-urlencode 'end=1634939100' HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
content-encoding: gzip

{
    "status": "success",
    "data": [
        {
            "__name__": "node_cpu_seconds_total",
            "app": "prometheus",
            "app_kubernetes_io_managed_by": "Helm",
            "chart": "prometheus-11.12.1",
            "cluster": "cluster-1",
            "component": "node-exporter",
            "cpu": "0",
            "heritage": "Helm",
            "instance": "10.0.100.36:9100",
            "job": "kubernetes-service-endpoints",
            "kubernetes_name": "servicesstackprometheuscf14a6d7-node-exporter",
            "kubernetes_namespace": "default",
            "kubernetes_node": "ip-10-0-100-36.us-west-2.compute.internal",
            "mode": "idle",
            "release": "servicesstackprometheuscf14a6d7"
        },
        {
            "__name__": "node_cpu_seconds_total",
            "app": "prometheus",
            "app_kubernetes_io_managed_by": "Helm",
            "chart": "prometheus-11.12.1",
```

```
            "cluster": "cluster-1",
            "component": "node-exporter",
            "cpu": "0",
            "heritage": "Helm",
            "instance": "10.0.100.36:9100",
            "job": "kubernetes-service-endpoints",
            "kubernetes_name": "servicesstackprometheuscf14a6d7-node-exporter",
            "kubernetes_namespace": "default",
            "kubernetes_node": "ip-10-0-100-36.us-west-2.compute.internal",
            "mode": "iowait",
            "release": "servicesstackprometheuscf14a6d7"
        },
        ...
    ]
}
```

ListAlerts

ListAlerts オペレーションは、ワークスペースで現在アクティブなアラートを取得します。

有効な HTTP 動詞:

```
GET
```

有効な URI:

```
/workspaces/workspaceId/api/v1/alerts
```

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/alerts HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 386
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
```

```
Content-Type: application/json
Server: amazon
vary: Origin

{
  "status": "success",
  "data": {
    "alerts": [
      {
        "labels": {
          "alertname": "test-1.alert",
          "severity": "none"
        },
        "annotations": {
          "message": "message"
        },
        "state": "firing",
        "activeAt": "2020-12-01T19:37:25.429565909Z",
        "value": "1e+00"
      }
    ]
  },
  "errorType": "",
  "error": ""
}
```

ListAlertManagerAlerts

`ListAlertManagerAlerts` は、ワークスペースのアラートマネージャーで現在発生しているアラートに関する情報を取得します。

有効な HTTP 動詞:

```
GET
```

有効な URI:

```
/workspaces/workspaceId/alertmanager/api/v2/alerts
```

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts
 HTTP/1.1
```

```
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 354
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "annotations": {
            "summary": "this is a test alert used for demo purposes"
        },
        "endsAt": "2021-10-21T22:07:31.501Z",
        "fingerprint": "375eab7b59892505",
        "receivers": [
            {
                "name": "sns-0"
            }
        ],
        "startsAt": "2021-10-21T22:02:31.501Z",
        "status": {
            "inhibitedBy": [],
            "silencedBy": [],
            "state": "active"
        },
        "updatedAt": "2021-10-21T22:02:31.501Z",
        "labels": {
            "alertname": "test-alert"
        }
    }
]
```

ListAlertManagerAlertGroups

`ListAlertManagerAlertGroups` オペレーションは、ワークスペースのアラートマネージャーで構成されているアラートグループのリストを取得します。

有効な HTTP 動詞:

`GET`

有効な URI:

`/workspaces/workspaceId/alertmanager/api/v2/alerts/groups`

URL クエリパラメータ:

`active` ブール値。true の場合、返されるリストにはアクティブなアラートが含まれます。デフォルトは true です。オプションです。

`silenced` ブール値。true の場合、返されるリストには無音のアラートが含まれます。デフォルトは true です。オプションです。

`inhibited` ブール値。true の場合、返されるリストには禁止されたアラートが含まれます。デフォルトは true です。オプションです。

`filter` 文字列の配列。アラートをフィルタリングするマッチャーのリスト。オプションです。

`receiver` 文字列。アラートをフィルタリングするレシーバーに一致する正規表現。オプションです。

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/alerts/
groups HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 443
```

```
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "alerts": [
            {
                "annotations": {
                    "summary": "this is a test alert used for demo purposes"
                },
                "endsAt": "2021-10-21T22:07:31.501Z",
                "fingerprint": "375eab7b59892505",
                "receivers": [
                    {
                        "name": "sns-0"
                    }
                ],
                "startsAt": "2021-10-21T22:02:31.501Z",
                "status": {
                    "inhibitedBy": [],
                    "silencedBy": [],
                    "state": "unprocessed"
                },
                "updatedAt": "2021-10-21T22:02:31.501Z",
                "generatorURL": "https://www.amazon.com/",
                "labels": {
                    "alertname": "test-alert"
                }
            }
        ],
        "labels": {},
        "receiver": {
            "name": "sns-0"
        }
    }
]
```

ListAlertManagerReceivers

`ListAlertManagerReceivers` オペレーションは、アラートマネージャーで構成されているレシーバーに関する情報を取得します。

有効な HTTP 動詞:

```
GET
```

有効な URI:

```
/workspaces/workspaceId/alertmanager/api/v2/receivers
```

URL クエリパラメータ: なし

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/receivers
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 19
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "name": "sns-0"
    }
]
```

ListAlertManagerSilences

`ListAlertManagerSilences` は、ワークスペースに構成されているアラートサイレンスに関する情報を取得します。

有効な HTTP 動詞:

```
GET
```

有効な URI:

```
/workspaces/workspaceId/alertmanager/api/v2/silences
```

リクエスト例

```
GET /workspaces/ws-58a6a446-5ec4-415b-9052-a449073bbd0a/alertmanager/api/v2/silences
 HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 312
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

[
    {
        "id": "d29d9df3-9125-4441-912c-70b05f86f973",
        "status": {
            "state": "active"
        },
        "updatedAt": "2021-10-22T19:32:11.763Z",
        "comment": "hello-world",
        "createdBy": "test-person",
```

```
        "endsAt": "2023-07-24T01:05:36.000Z",
        "matchers": [
            {
                "isEqual": true,
                "isRegex": true,
                "name": "job",
                "value": "hello"
            }
        ],
        "startsAt": "2021-10-22T19:32:11.763Z"
    }
]
```

ListRules

ListRules は、ワークスペースに構成されているルールに関する情報を取得します。

有効な HTTP 動詞:

GET

有効な URI:

/workspaces/*workspaceId*/api/v1/rules

リクエスト例

```
GET /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/rules HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 423
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
```

```
vary: Origin

{
    "status": "success",
    "data": {
        "groups": [
            {
                "name": "test-1.rules",
                "file": "test-rules",
                "rules": [
                    {
                        "name": "record:1",
                        "query": "sum(rate(node_cpu_seconds_total[10m:1m]))",
                        "labels": {},
                        "health": "ok",
                        "lastError": "",
                        "type": "recording",
                        "lastEvaluation": "2021-10-21T21:22:34.429565909Z",
                        "evaluationTime": 0.001005399
                    }
                ],
                "interval": 60,
                "lastEvaluation": "2021-10-21T21:22:34.429563992Z",
                "evaluationTime": 0.001010504
            }
        ]
    },
    "errorType": "",
    "error": ""
}
```

PutAlertManagerSilences

PutAlertManagerSilences オペレーションは、新しいアラートサイレンスの作成または既存のアラートサイレンスの更新を行います。

有効な HTTP 動詞:

POST

有効な URI:

/workspaces/*workspaceId*/alertmanager/api/v2/silences

URL クエリパラメータ:

silence サイレンスを表すオブジェクト。形式を次に示します。

```
{
  "id": "string",
  "matchers": [
    {
      "name": "string",
      "value": "string",
      "isRegex": Boolean,
      "isEqual": Boolean
    }
  ],
  "startsAt": "timestamp",
  "endsAt": "timestamp",
  "createdBy": "string",
  "comment": "string"
}
```

リクエスト例

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/alertmanager/api/v2/silences
 HTTP/1.1
Content-Length: 281,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0

{
   "matchers":[
      {
         "name":"job",
         "value":"up",
         "isRegex":false,
         "isEqual":true
      }
   ],
   "startsAt":"2020-07-23T01:05:36+00:00",
   "endsAt":"2023-07-24T01:05:36+00:00",
   "createdBy":"test-person",
   "comment":"test silence"
```

```
}
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 53
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin

{
    "silenceID": "512860da-74f3-43c9-8833-cec026542b32"
}
```

QueryMetrics

QueryMetrics オペレーションは、特定の時点において、または一定期間にわたってインスタントクエリを評価します。

有効な HTTP 動詞:

GET, POST

有効な URI:

/workspaces/*workspaceId*/api/v1/query この URI は、特定の時点でインスタントクエリを評価します。

/workspaces/*workspaceId*/api/v1/query_range この URI は、一定期間にわたってインスタントクエリを評価します。

URL クエリパラメータ:

query=<string> Prometheus 式のクエリ文字列。query と query_range の両方で使用されます。

time=<rfc3339 | unix_timestamp> (オプション) query を使用して特定の時点でインスタントクエリを評価する場合、評価のタイムスタンプ。

timeout=<duration> (オプション) 評価のタイムアウト。デフォルトは -query.timeout フラグの値で、この値が上限になります。query と query_range の両方で使用されます。

start=<rfc3339 | unix_timestamp> query_range を使用して一定期間にわたってクエリを評価する場合、開始タイムスタンプ。

end=<rfc3339 | unix_timestamp> query_range を使用して一定期間にわたってクエリを評価する場合、終了タイムスタンプ。

step=<duration | float> クエリの解決ステップ幅 (duration 形式または float の秒数)。query_range を使用して一定期間にわたってクエリを評価するときに、そのクエリで必要とされる場合にのみ使用します。

duration

Prometheus 互換 API の duration には、数値に続けて以下の単位のいずれかを指定します。

- ms ミリ秒
- s 秒
- m 分
- h 時間
- d 日 (1 日は常に 24h と想定)
- w 週 (1 週間は常に 7d と想定)
- y 年 (1 年は常に 365d と想定)

リクエスト例

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/query?
query=sum(node_cpu_seconds_total) HTTP/1.1
Content-Length: 0,
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Grafana/8.1.0
```

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length: 132
Connection: keep-alive
```

```
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
content-encoding: gzip

{
    "status": "success",
    "data": {
        "resultType": "vector",
        "result": [
            {
                "metric": {},
                "value": [
                    1634937046.322,
                    "252590622.81000024"
                ]
            }
        ]
    }
}
```

RemoteWrite

RemoteWrite オペレーションは、Prometheus サーバーからリモート URL にメトリクスを標準化された形式で書き込みます。通常、このオペレーションを呼び出すには、Prometheus サーバーなどの既存のクライアントを使用します。

有効な HTTP 動詞:

POST

有効な URI:

/workspaces/*workspaceId*/api/v1/remote_write

URL クエリパラメータ:

なし

RemoteWrite での取り込みレートは 70,000 サンプル/秒、取り込みバーストサイズは 1,000,000 サンプルです。

リクエスト例

```
POST /workspaces/ws-b226cc2a-a446-46a9-933a-ac50479a5568/api/v1/remote_write --data-
binary "@real-dataset.sz" HTTP/1.1
Authorization: AUTHPARAMS
X-Amz-Date: 20201201T193725Z
User-Agent: Prometheus/2.20.1
Content-Type: application/x-protobuf
Content-Encoding: snappy
X-Prometheus-Remote-Write-Version: 0.1.0

body
```

> **Note**
>
> リクエスト本文の構文については、https://github.com/prometheus/prometheus/blob/1c624c58ca934f618be737b4995e22051f5724c1/prompb/remote.pb.go#L64 のプロトコルバッファの定義を参照してください。

レスポンス例

```
HTTP/1.1 200 OK
x-amzn-RequestId: 12345678-abcd-4442-b8c5-262b45e9b535
Content-Length:0
Connection: keep-alive
Date: Tue, 01 Dec 2020 19:37:25 GMT
Content-Type: application/json
Server: amazon
vary: Origin
```

Amazon Managed Service for Prometheus ユーザーガイドのドキュメント履歴

次の表は、「Amazon Managed Service for Prometheus ユーザーガイド」のドキュメントの重要な更新情報をまとめたものです。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
コンソールにルール定義ファイルとアラートマネージャー設定ファイルの編集を追加	Amazon Managed Service for Prometheus では、Amazon Managed Service for Prometheus コンソール内からアラートマネージャー設定ファイルとルール定義ファイルを編集するためのサポートが追加されました。	2024 年 5 月 16 日
Amazon EKS のアクセスエントリを含む、よりシンプルな AWS マネージドコレクターのセットアップを追加	Amazon Managed Service for Prometheus では、Amazon EKS アクセスエントリのサポートが追加され、AWS マネージドコレクター の設定が簡素化されました。AmazonPrometheusScraperServiceRolePolicy マネージドコレクターの AWS マネージドポリシーが更新され、使用されなくなったアクセスエントリを削除できるようになりました。	2024 年 5 月 2 日
AWS API を別の API リファレンスガイドに移動する	Amazon Managed Service for Prometheus AWS APIs「Amazon Managed Service for Prometheus API	2024 年 2 月 7 日

	リファレンス」で利用可能になりました。Prometheus 互換 APIs「Amazon Managed Service for Prometheus ユーザーガイド」に引き続き記載されています。	
ワークスペース暗号化用のカスタマーマネージドキーを追加	Amazon Managed Service for Prometheus では、ワークスペース暗号化用のカスタマーマネージドキーのサポートが追加されました。詳細については、「保管時の暗号化」を参照してください。	2023 年 12 月 21 日
に新しいアクセス許可を追加 AmazonPrometheusFullAccess	Amazon EKS クラスターのAmazonPrometheusFullAccessマネージドコレクターの作成をサポートする新しいアクセス許可が AWS マネージドポリシーに追加されました。	2023 年 11 月 26 日
新しい マネージドポリシーを追加しました。 AmazonPrometheusScraperServiceLinkedRolePolicy	マネージドコレクターAmazonPrometheusScraperServiceLinkedRolePolicyが Amazon EKS クラスターからメトリクスを収集するための新しい AWS マネージドポリシーを追加しました。	2023 年 11 月 26 日
取り込み方法として AWS マネージドコレクターを追加	Amazon Managed Service for Prometheus では、AWS マネージドコレクターのサポートが追加されました。	2023 年 11 月 26 日

Amazon Managed Grafana との統合のサポートが追加されました。	Amazon Managed Service for Prometheus に Amazon Managed Grafana アラートとの統合のサポートが追加されました。	2022 年 11 月 23 日
に新しいアクセス許可を追加 AmazonPrometheusConsoleFullAccess	Logs でのアラートマネージャーとルーラーイベントのログ記録をサポートする新しいアクセス許可が CloudWatch AmazonPrometheusConsoleFullAccessマネージドポリシーに追加されました。	2022 年 10 月 24 日
Amazon EKS オブザーバビリティソリューションが追加されました。	Amazon Managed Service for Prometheus は、 AWS Observability Accelerator を使用する新しいソリューションを追加します。詳細については、「AWS Observability Accelerator の使用」を参照してください。	2022 年 10 月 14 日
Amazon EKS コストモニタリングとの統合のサポートが追加されました。	Amazon Managed Service for Prometheus に、Amazon EKS コストモニタリングとの統合のサポートが追加されました。詳細については、「Amazon EKS コストモニタリングとの統合」を参照してください。	2022 年 9 月 22 日

Amazon CloudWatch Logs でアラートマネージャーログとルーラーログのサポートを開始しました。	Amazon Managed Service for Prometheus が、Amazon CloudWatch Logs のアラートマネージャーとルーラーのエラーログのサポートを開始しました。詳細については、「Amazon CloudWatch Logs」を参照してください。	2022 年 9 月 1 日
カスタムのストレージ保持のサポートが追加されました。	Amazon Managed Service for Prometheus で、ワークスペースのクォータを変更することにより、ワークスペースごとにカスタムのストレージ保持がサポートされるようになりました。Amazon Managed Service for Prometheus のクォータの詳細については、「サービスクォータ」を参照してください。	2022 年 8 月 12 日
Amazon に使用状況メトリクスを追加しました CloudWatch。	Amazon Managed Service for Prometheus で、Amazon への使用状況メトリクスの送信のサポートが追加されました CloudWatch。詳細については、「Amazon CloudWatch メトリクス」を参照してください。	2022 年 5 月 6 日
欧州 (ロンドン) リージョンのサポートが追加されました。	Amazon Managed Service for Prometheus に欧州 (ロンドン) リージョンのサポートが追加されました。	2022 年 5 月 4 日

Amazon Managed Service for Prometheus が一般公開され、ルールとアラートマネージャーのサポートが追加されました。	Amazon Managed Service for Prometheus が一般公開されました。また、ルールとアラートマネージャーのサポートも追加されました。詳細については、「記録ルールとアラートルール」および「アラートマネージャーとテンプレート」を参照してください。	2021 年 9 月 29 日
タグ付けのサポートが追加されました。	Amazon Managed Service for Prometheus で、Amazon Managed Service for Prometheus ワークスペースのタグ付けがサポートされました。	2021 年 9 月 7 日
アクティブなシリーズ数と取り込みレートのクォータが増加しました。	アクティブなシリーズ数のクォータが 1,000,000 に増加し、取り込みレートのクォータが 1 秒あたり 70,000 サンプルに増加しました。	2021 年 2 月 22 日
Amazon Managed Service for Prometheus のプレビューがリリースされました。	Amazon Managed Service for Prometheus 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。