



ユーザーガイド

# Amazon Managed Service for Prometheus



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

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

# Table of Contents

Amazon Managed Service for Prometheus とは .....	1
サポートされるリージョン .....	1
料金 .....	3
プレミアムサポート .....	4
開始 .....	5
セットアップ .....	5
にサインアップする AWS アカウント .....	5
管理アクセスを持つユーザーを作成する .....	6
ワークスペースの作成 .....	7
ワークスペースへの Prometheus メトリクスの取り込み .....	8
ステップ 1: 新しい Helm チャートリポジトリを追加する .....	9
ステップ 2: Prometheus 名前空間を作成する .....	9
ステップ 3: サービスアカウントの IAM ロールを設定する .....	10
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 .....	10
Prometheus メトリクスに対するクエリの実行 .....	11
ワークスペースの管理 .....	13
ワークスペースの作成 .....	13
ワークスペースの編集 .....	16
ワークスペースの ARN の確認 .....	17
ワークスペースの削除 .....	17
取り込みメトリクス .....	19
AWS マネージドコレクター .....	20
マネージドコレクターの使用 .....	21
Prometheus と互換性のあるメトリクス .....	35
カスターマネージドコレクター .....	36
メトリクスの取り込みの保護 .....	37
ADOT コレクター .....	38
Prometheus コレクター .....	54
高可用性データ .....	63
メトリクスのクエリ .....	72
メトリクスクエリの保護 .....	72
Amazon Managed Service for Prometheus AWS PrivateLink での の使用 .....	37
認証と認可 .....	37
Amazon Managed Grafana のセットアップ .....	73

プライベート VPC での Amazon Managed Grafana への接続 .....	74
Grafana オープンソースのセットアップ .....	74
AWS SigV4 の設定 .....	75
Grafana での Prometheus データソースの追加 .....	76
[保存してテスト] が機能しない場合のトラブルシューティング .....	78
Amazon EKS で動作する Grafana の設定 .....	79
AWS SigV4 をセットアップする .....	79
サービスアカウントの IAM ロールの設定 .....	80
Helm を使用した Grafana サーバーのアップグレード .....	81
Grafana での Prometheus データソースの追加 .....	82
Prometheus 互換 API を使用したクエリ .....	83
awscurl を使用した Prometheus 互換 API のクエリ .....	83
クエリ API レスポンス内のクエリ統計情報 .....	85
記録ルールとアラートルール .....	89
必要な IAM アクセス許可 .....	90
ルールファイルの作成 .....	91
Amazon Managed Service for Prometheus へのルール設定ファイルのアップロード .....	92
ルール設定ファイルの編集 .....	94
ルーラーのトラブルシューティング .....	95
アラートマネージャー .....	97
必要な IAM アクセス許可 .....	98
アラートマネージャー設定ファイルの作成 .....	99
アラートレシーバーの設定 .....	101
(オプション) 新しい Amazon SNS トピックの作成 .....	102
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の Amazon Managed Service for Prometheus への付与 .....	102
アラートマネージャー設定ファイルでの Amazon SNS トピックの指定 .....	104
(オプション) Amazon SNS に JSON を出力するためのアラートマネージャーの構成 .....	106
(オプション) Amazon SNS から他の宛先への送信 .....	108
SNS レシーバーのメッセージ検証と切り捨てルール .....	109
アラートマネージャー設定ファイルのアップロード .....	110
アラートと Grafana の統合 .....	112
前提条件 .....	113
Amazon Managed Grafana のセットアップ .....	114
アラートマネージャーのトラブルシューティング .....	115
空のコンテンツに関する警告 .....	115

非 ASCII 文字に関する警告 .....	116
無効な key/value に関する警告 .....	116
メッセージの制限に関する警告 .....	117
リソースベースのポリシーがないことによるエラー .....	117
ロギングとモニタリング .....	119
CloudWatch メトリクス .....	119
CloudWatch アラームの設定 .....	124
CloudWatch ログ .....	125
CloudWatch ログの設定 .....	125
コストの理解と最適化 .....	128
コストに影響する要因は何ですか? .....	128
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? .....	128
クエリコストを削減する最善の方法は何ですか? .....	128
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? .....	129
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? .....	129
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? .....	130
請求書はいつでも確認できますか? .....	130
月初めの請求額が月末よりも高いのはなぜですか? .....	130
Prometheus 用 Amazon マネージドサービスの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。何が起きているのでしょうか? .....	131
統合 .....	132
Amazon EKS コストモニタリング .....	132
AWS Observability Accelerator .....	133
前提条件 .....	133
インフラストラクチャモニタリングのサンプルの使用 .....	134
AWS Kubernetes 用コントローラー .....	135
前提条件 .....	136
ワークスペースのデプロイ .....	137
リモート書き込みのためのクラスターの構成 .....	141
Firehose での Amazon CloudWatch メトリクス .....	143
インフラストラクチャ .....	143
Amazon CloudWatch ストリームの作成 .....	146
クリーンアップ .....	147
セキュリティ .....	148
データ保護 .....	149

Amazon Managed Service for Prometheus によって収集されるデータ .....	150
保管中の暗号化 .....	151
Identity and Access Management .....	164
対象者 .....	164
アイデンティティを使用した認証 .....	165
ポリシーを使用したアクセスの管理 .....	169
Amazon Managed Service for Prometheus と IAM の連携 .....	171
アイデンティティベースポリシーの例 .....	179
AWS マネージドポリシー .....	182
トラブルシューティング .....	194
IAM のアクセス許可とポリシー .....	196
Amazon Managed Service for Prometheus のアクセス許可 .....	196
IAM ポリシーのサンプル .....	199
コンプライアンス検証 .....	200
レジリエンス .....	201
インフラストラクチャセキュリティ .....	202
サービスリンクロールの使用 .....	202
メトリクスクレイピングロール .....	203
CloudTrail ログ .....	205
の Amazon Managed Service for Prometheus 情報 CloudTrail .....	205
Amazon Managed Service for Prometheus のログファイルエントリについて .....	207
サービスアカウントの IAM ロールの設定 .....	211
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 .....	212
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 .....	215
インターフェイス VPC エンドポイント .....	218
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 .....	219
トラブルシューティング .....	223
429 または制限超過エラー .....	223
サンプルが重複している .....	224
サンプルタイムスタンプに関するエラーが表示される .....	224
制限に関するエラーメッセージが表示される .....	225
ローカル Prometheus サーバーの出力が制限を超えている .....	225
データの一部が表示されない .....	226
タグ付け .....	228
ワークスペースのタグ付け .....	229

ワークスペースへのタグの追加 .....	229
ワークスペースのタグの表示 .....	231
ワークスペースのタグの編集 .....	232
ワークスペースからのタグの削除 .....	233
ルールグループ名前空間のタグ付け .....	235
ルールグループ名前空間へのタグの追加 .....	235
ルールグループ名前空間のタグの表示 .....	237
ルールグループ名前空間のタグの編集 .....	238
ルールグループ名前空間からのタグの削除 .....	239
Service Quotas .....	241
Service Quotas .....	241
アクティブなシリーズ数のデフォルト .....	247
取り込みスロットリング .....	248
取り込まれるデータに関する追加の制限 .....	249
API リファレンス .....	250
Amazon Managed Service for Prometheus API .....	250
AWS SDK での Amazon Managed Service for Prometheus の使用 .....	250
Prometheus 互換 API .....	251
CreateAlertManagerAlerts .....	251
DeleteAlertManagerSilence .....	253
GetAlertManagerStatus .....	254
GetAlertManagerSilence .....	255
GetLabels .....	256
GetMetricMetadata .....	258
GetSeries .....	260
ListAlerts .....	262
ListAlertManagerAlerts .....	263
ListAlertManagerAlertGroups .....	264
ListAlertManagerReceivers .....	266
ListAlertManagerSilences .....	267
ListRules .....	268
PutAlertManagerSilences .....	270
QueryMetrics .....	271
RemoteWrite .....	273
ドキュメント履歴 .....	276
AWS 用語集 .....	281

---

..... cclxxxii



# 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 サービスと統合して、データへの高速かつ安全なアクセスを可能にします。

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、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。

## 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
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
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
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
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
アジアパシフィック (シンガポール)	ap-southeast-1	aps.ap-southeast-1.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-1.amazonaws.com	HTTPS
アジアパシフィック (シドニー)	ap-southeast-2	aps.ap-southeast-2.amazonaws.com	HTTPS
		aps-workspaces.ap-southeast-2.amazonaws.com	HTTPS
アジアパシフィック (東京)	ap-northeast-1	aps.ap-northeast-1.amazonaws.com	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
		aps-workspaces.ap-northeast-1.amazonaws.com	
欧州 (フランクフルト)	eu-central-1	aps.eu-central-1.amazonaws.com	HTTPS
		aps-workspaces.eu-central-1.amazonaws.com	HTTPS
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
欧州 (ロンドン)	eu-west-2	aps.eu-west-2.amazonaws.com	HTTPS
		aps-workspaces.eu-west-2.amazonaws.com	HTTPS
欧州 (パリ)	eu-west-3	aps.eu-west-3.amazonaws.com	HTTPS
		aps-workspaces.eu-west-3.amazonaws.com	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
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

## 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。

Cost Explorer と AWS コストと使用状況レポートを使用して、の料金をモニタリングできます。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

任意のレベルの AWS プレミアムサポートプランを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。

# 開始

このセクションでは、Amazon Managed Service for Prometheus ワークスペースをすばやく作成する方法、ワークスペースへの Prometheus メトリクスの取り込みを設定する方法、それらのメトリクスにクエリを実行する方法について説明します。

また、を初めて使用する場合に備えて AWS アカウント、の設定に関する情報も含まれています AWS。

トピック

- [セットアップ](#)
- [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

このセクションのタスクを完了して、AWS を初めてセットアップします。AWS アカウントをすでにお持ちの場合は、「」に進んでください[ワークスペースの作成](#)。

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

トピック

- [にサインアップする AWS アカウント](#)
- [管理アクセスを持つユーザーを作成する](#)

## にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

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

サインアップ手順の一環として、通話呼び出しを受け取り、電話キーパッドで検証コードを入力するように求められます。

にサインアップすると AWS アカウント、AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用して [ルートユーザーアクセスが必要なタスク](#) を実行してください。

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

## 管理アクセスを持つユーザーを作成する

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

### のセキュリティ保護 AWS アカウントのルートユーザー

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

ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「[ルートユーザーとしてサインインする](#)」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

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

### 管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

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

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

を ID ソース IAM アイデンティティセンターディレクトリとして使用する方法的チュートリアルについては、「[ユーザーガイド](#)」の「[デフォルトでユーザーアクセス IAM アイデンティティセンターディレクトリを設定するAWS IAM Identity Center](#)」を参照してください。

### 管理アクセス権を持つユーザーとしてサインインする

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

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

### 追加のユーザーにアクセス権を割り当てる

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

手順については、「[AWS IAM Identity Center ユーザーガイド](#)」の「[権限設定を作成する](#)」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

手順については、「[AWS IAM Identity Center ユーザーガイド](#)」の「[グループの参加](#)」を参照してください。

## 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

ワークスペース作成の詳細については、「[ワークスペースの作成](#)」を参照してください。

## 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。ここでは、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、「[ワークスペースへのメトリクスの取り込み](#)」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。ここでは、Amazon EKS クラスターに新しい Prometheus サーバーをセットアップします。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するエージェントとして稼働します。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。
- 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。*prometheus-agent-namespace* は、この名前空間に付ける名前に置き換えます。

```
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** 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](#)」で作成した amp-iamproxy-ingest-role の ARN に置き換えます。
  - **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。

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

```
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。

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 用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 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、「[アラートと Amazon Managed Grafana またはオープンソースの Grafana の統合](#)」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 VPC での Amazon Managed Grafana への接続](#)」を参照してください。

## 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

- [ワークスペースの作成](#)
- [ワークスペースの編集](#)
- [ワークスペースの ARN の確認](#)
- [ワークスペースの削除](#)

## 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、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。

(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、こ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の ARN 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で作成されたワークスペースは、後で 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 (その逆も同様です)。

- `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。を、`create-workspace` コマンドが返した値 `my-workspace-id` に置き換えます `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. Amazon Managed Service for Prometheus コンソール (<https://console.aws.amazon.com/prometheus/>) を開きます。
2. [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。

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の ARN を入力できます。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、こ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で作成されたワークスペースは、後で 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 (その逆も同様です)。

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。

ワークスペースの準備が整うまで、ステータスは [作成中] に戻ります。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なになります。

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。

## ワークスペースの編集

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. 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. 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。



## ワークスペースの ARN の確認

Amazon Managed Service for Prometheus ワークスペースの ARN は、コンソールまたは AWS CLI を使用して確認できます。

Amazon Managed Service for Prometheus コンソールを使用してワークスペースの ARN を確認するには

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 を選択します。

[ARN] の下にワークスペースの ARN が表示されます。

を使用してワークスペース ARN AWS CLI を検索するには、次のコマンドを入力します。

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

結果から arn の値を確認します。

## ワークスペースの削除

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. 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. 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。

# ワークスペースへのメトリクスの取り込み

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ための、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](#)
- [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

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トヤ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 `awscli` を使用してプログラムで作成できます AWS CLI。

### Note

[カスターマネージドキー](#)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。

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 [awscli](#) 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

CreateScrapper 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

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。GetDefaultScrapperConfiguration 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**

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

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は次のとおりです。

- 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権を付与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。

スクレイパーは、クラスターの作成時または作成後に作成できます。

- 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 API\\_AND\\_CONFIG\\_MAP スクレイパー](#) には、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/scrapper.aps.amazonaws.com/  
AWSServiceRoleForAmazonPrometheusScrapper_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/AWSServiceRoleForAmazonPrometheusScrapper_unique-id
```

例えば、この ARN の場合、

```
arn:aws:iam::111122223333:role/aws-service-role/scrapper.aps.amazonaws.com/  
AWSServiceRoleForAmazonPrometheusScrapper_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/  
AWSServiceRoleForAmazonPrometheusScrapper_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アップデートを利用できます。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`scrapersId`スクレイパーのを見つけ、[DeleteScrapers](#)オペレーションを使用して削除します。

または、`aws amp delete-scrapers`を使用して AWS CLI を呼び出します。

```
aws amp delete-scrapers --scrapers-id scrapersId
```

## 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 `GetDefaultScrapersConfiguration` 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 `<kubernetes_sd_config>` のみです。

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。
- 承認 — 承認が必要ない場合は省略します。必要に応じて、認証は `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せんが、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ー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。ADOT を使用して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](#)

- [Distro for AWS をコレクター OpenTelemetry として使用する](#)
- [Prometheus インスタンスをコレクターとして使用する](#)
- [高可用性データのための Amazon Managed Service for Prometheus の設定](#)

## 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、VPC から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を経由せず、AWS のネットワーク内で保護されます。Amazon Managed Service for Prometheus の AWS PrivateLink VPC エンドポイントを作成するには、「」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。IAM を使用して、誰を認証 (サインイン) し、誰にリソースの使用を認可する (アクセス許可を付与する) かを制御します。Amazon Managed Service for Prometheus は IAM と統合されているため、データを安全に保つことができます。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めるようにする IAM ロールと、Amazon Managed Service for Prometheus ワークスペースに保存されたメトリクスに対して Grafana サーバーからクエリを実行できるようにする IAM ロールを作成する必要があります。IAM の詳細については、「[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、HTTP によって送信される AWS リクエストに認証情報を追加するプロセスです。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

以下のトピックでは、Distro for AWS OpenTelemetry 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。

### トピック

- [Amazon Elastic Kubernetes Service AWS クラスターで Distro for Open Telemetry 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 を使用して Amazon ECS 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た Amazon EC2 インスタンスからのメトリクスの取り込みの設定](#)

### Amazon Elastic Kubernetes Service AWS クラスターで Distro for Open Telemetry を使用して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 OpenTelemetry (ADOT) Collector AWS を設定して Prometheus 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。

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

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。これらの設定を直接 ADOT コレクターの設定に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。データをエクスポートする 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 EKS 上の Prometheus メトリクスエンドポイントを自動的に検出し、[<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、[AWS 「SDK for Go 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手順を開始する前に、サービスアカウントの IAM ロールと信頼ポリシーを設定する必要があります。

サービスアカウントの IAM ロールと信頼ポリシーを設定するには

1. 「[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、サービスアカウントの IAM ロールを作成します。

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。IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
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. 信頼関係ポリシーの JSON で、`aws-amp` を `adot-col` に置き換えて [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 EKS で名前空間を作成すると、alertmanager とノードエクスポートはデフォルトで無効になっています。

Amazon EKS または Kubernetes クラスターで Prometheus の収集を有効にするには

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. ADOT コレクターのデフォルトのインスタンスを起動します。そのためには、まず次のコマンドを入力して、ADOT コレクターの Kubernetes 設定を取得します。

```
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 コレクターの設定で注釈 (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ています。ADOT コレクターは、これによってサンプルアプリのエンドポイントをクラスター内の kube-system エンドポイントから区別できます。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. 次のコマンドを入力して、ADOT コレクターをデプロイします。

```
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を持つ 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。これらの設定を直接 ADOT コレクターの設定に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、デフォルトの認証情報 AWS 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 を使用して Amazon ECS 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 (ADOT) を使用して Amazon AWS Elastic Container Service (Amazon ECS) からメトリクスを収集し、Amazon Managed Service for Prometheus に取り込む方法について説明します。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、AWS Fargate クラスター上のデフォルト設定の Amazon ECS 環境と、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。

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

## カスタム ADOT コレクターコンテナイメージの定義

以下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。*my-remote-URL* と *my-region* は、使用中の endpoint と region の値に置き換えます。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]
```

## Amazon ECR リポジトリへの ADOT コレクターコンテナイメージのプッシュ

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

1. Dockerfile をビルドして、コンテナイメージをコピーして OTEL Docker 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。*my-region* は、使用中の region の値に置き換えます。

```
# 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
```

Amazon Managed Service for Prometheus をスクレイピングする Amazon ECS タスク定義の作成

Amazon Managed Service for Prometheus をスクレイピングする 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。adot-collector の image の値は、リポジトリの URL とイメージタグ (\$COLLECTOR\_REPOSITORY:ecs) に置き換えます。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"
}
```

## タスク用の IAM ロールへの AWS マネージドポリシー **AmazonPrometheusRemoteWriteAccess** のアタッチ

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 AWS API オペレーションを呼び出すための正しいアクセス許可が必要です。タスク用の IAM ロールを作成し、そのロールに **AmazonPrometheusRemoteWriteAccess** ポリシーをアタッチする必要があります。このロールを作成してポリシーをアタッチする方法の詳細については、「[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 ロールに **AmazonPrometheusRemoteWriteAccess** をアタッチし、そのロールをタスクに使用したら、スクレイピングされたメトリクスを Amazon ECS によって Amazon Managed Service for Prometheus に送信できます。

## Amazon Managed Grafana でのメトリクスの視覚化

### Important

開始する前に、Amazon ECS タスク定義に対して Fargate 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た Amazon EC2 インスタンスからのメトリクスの取り込みの設定

このセクションでは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス上で、リモート書き込みを使用する 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。ユーザーが Prometheus、Amazon EC2、Amazon Managed Service for Prometheus に精通していることを前提としています。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 EC2 用の IAM ロールの作成

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. IAM コンソール (<https://console.aws.amazon.com/iam/>) を開きます。
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。ユースケースとして [EC2] を選択します。[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。[次へ: タグ] を選択します。
5. (オプション) IAM ロールに IAM タグを作成します。[次へ: 確認] を選択します。
6. ロールの名前を入力します。[ポリシーを作成] を選択します。

## Amazon EC2 インスタンスの起動

Amazon EC2 インスタンスを起動するには、「Amazon Elastic Compute Cloud Linux インスタンス用ユーザーガイド」の「[インスタンスの起動](#)」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、そのロールを使用して 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. 以下の YAML ファイルの例をテンプレートとして使用して、`prometheus.yaml` という名前の新しいファイルを作成します。の場合 `url`、`my-region` をリージョン値に置き換え、`my-`

*workspace-id*を Amazon Managed Service for Prometheus が生成したワークスペース ID に置き換えます。region については、*my-region* を使用中のリージョンの値に置き換えます。

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

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**

[セキュリティ機能を有効にする](#)ことで、Prometheus Scrape エンドポイントがパブリックインターネットに公開されることを回避できます。

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](#)
- [EC2 上の Kubernetes 内にある既存の Prometheus サーバーからの取り込みの設定](#)
- [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。ここでは、Amazon EKS クラスターに新しい Prometheus サーバーをセットアップします。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。
- 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。`prometheus-namespace` は、この名前空間に付ける名前に置き換えます。

```
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

ここで説明するオンボーディング方法では、Prometheus サーバーが実行されている Amazon EKS クラスターで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 `eksctl` 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。
  - `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。
  - `prometheus-namespace` は、Prometheus 名前空間の名前に置き換えます。

```
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。

## EC2 上の Kubernetes 内にある既存の Prometheus サーバーからの取り込みの設定

Amazon Managed Service for Prometheus は、Amazon EKS を実行しているクラスターおよび Amazon EC2 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。このセクションの詳細な手順は、Amazon EKS クラスター内の Prometheus サーバーを対象としています。Amazon EC2 上の自己管理型 Kubernetes クラスターの場合も手順は同じですが、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 ロールを手動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

ここで説明するオンボーディング方法では、Prometheus サーバーが実行されている Amazon EKS クラスターで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。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、「[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。

## 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 の ARN `${IAM_PROXY_PROMETHEUS_ROLE_ARN}` に置き換えます [ステップ 1: サービスアカウントの IAM ロールを設定する](#)。ロールの ARN は `arn:aws:iam::your account ID:role/amp-iamproxy-ingest-role` という形式になります。
  - `${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。以前のバージョンの Prometheus は AWS 署名バージョン 4 の署名プロセス (AWS SigV4) をネイティブにサポートしていないため、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。

この YAML ファイルに次の値を追加します。

```
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 の ARN

`${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 ARN は `arn:aws:iam::your account ID:role/amp-iamproxy-ingest-role` という形式になります。

`${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
```

を、前のコマンドで返された Prometheus helm チャートの名前 *prometheus-helm-chart-name* に置き換えます。 *prometheus-namespace* は、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 上で動作する 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むには、`amp_ingest_override_values.yaml` という名前のファイルのデフォルト設定を次のようにオーバーライドします。

```
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
```

## 高可用性データのための Amazon Managed Service for Prometheus の設定

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

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、「[クロスリージョンでの可用性](#)」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 による Amazon Managed Service for Prometheus への高可用性データの送信](#)
- [Distro for Open Telemetry を使用して Amazon Managed Service for Prometheus AWS に高可用性データを送信する](#)
- [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](#)
- [よくある質問: 高可用性の構成](#)
- [クロスリージョンでの可用性](#)

## 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、[高可用性に関するよくある質問](#)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、[高可用性に関するよくある質問](#)を参照してください。

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

Prometheus Operator 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。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があります。

例: YAML ヘッダー

次の YAML ヘッダーでは、`externalLabel` に `cluster` が追加され、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します。また、`replicaExternalLabels` はグループ内の各レプリカを識別します。

```
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、[高可用性に関するよくある質問](#)を参照してください。

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

AWS Distro for Open Telemetry (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 は、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取集できるように、ソース API、ライブラリ、エージェントを提供します。ADOT の詳細については、[AWS 「Distro for Open Telemetry について」](#)を参照してください。

高可用性設定で ADOT 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、[高可用性に関するよくある質問](#)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、[高可用性に関するよくある質問](#)を参照してください。

## よくある質問: 高可用性の構成

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 のクォータ](#)」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るように見え、メトリクスを重複排除していません。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` でした。これにより、`cluster`ラベルに依存する Amazon Managed Service for Prometheus の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 `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](#)

## クロスリージョンでの可用性

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。Grafana などのサービスを使用してメトリクスにクエリを実行することも、Amazon Managed Service for Prometheus API を使用することもできます。

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。

## トピック

- [メトリクスクエリの保護](#)
- [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 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](#)
- [Amazon EKS クラスターで動作する Grafana を使用したクエリ](#)
- [Prometheus 互換 API を使用したクエリ](#)
- [クエリ API レスポンス内のクエリ統計情報](#)

## メトリクスクエリの保護

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、VPCs から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 AWS PrivateLink VPC エンドポイントを作成するには、「」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。IAM を使用して、誰を認証 (サインイン) し、誰にリソースの使用を認可する (アクセス許可を付与する) かを制御します。Amazon Managed Service for Prometheus は IAM と統合されているため、データを安全に保つことができます。Amazon Managed Service for Prometheus をセットアップするときは、Amazon Managed Service for Prometheus ワークスペースに保存されたメトリクスに対して Grafana サーバーからクエリを実行できるようにする IAM ロールを作成する必要があります。IAM の詳細については、「[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、HTTP によって送信される AWS リクエストに認証情報を追加するプロセスです。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、「[アラートと Amazon Managed Grafana またはオープンソースの Grafana の統合](#)」を参照してください。

## プライベート VPC での Amazon Managed Grafana への接続

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。

Amazon Managed Grafana は、プライベート VPC を使用するように構成できます (Grafana でプライベート VPC をセットアップする方法の詳細については、「Amazon Managed Grafana User Guide」の「[Connecting to Amazon VPC](#)」を参照してください)。設定によっては、この VPC から Amazon Managed Service for Prometheus のサービスエンドポイントにアクセスできない場合があります。

特定のプライベート VPC を使用するように構成されている Amazon Managed Grafana ワークスペースに、Amazon Managed Service for Prometheus をデータソースとして追加するには、まず VPC エンドポイントを作成して、Amazon Managed Service for Prometheus を同じ VPC 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、「[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](#)」を参照してください。

## 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

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。

tar.gz または zip ファイルを使用してスタンドアロンの 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「[Install Grafana](#)」を参照してください。新しいスタンドアロン Grafana をインストールすると、ユーザー名とパスワードの入力を求められます。デフォルトは **admin/admin** です。初回のログイン後、パスワードを変更するように求められます。詳細については、Grafana ドキュメントの「[Getting started with Grafana](#)」を参照してください。

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/bin/grafana-server -v
```

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 の設定

Amazon Managed Service for Prometheus は AWS Identity and Access Management (IAM) と連携して、Prometheus 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
```

## 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 URL として、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. 指定した HTTP URL から、URL に追加されている `/api/v1/query` という文字列を削除します。これは、Prometheus データソースによって自動的に追加されるためです。

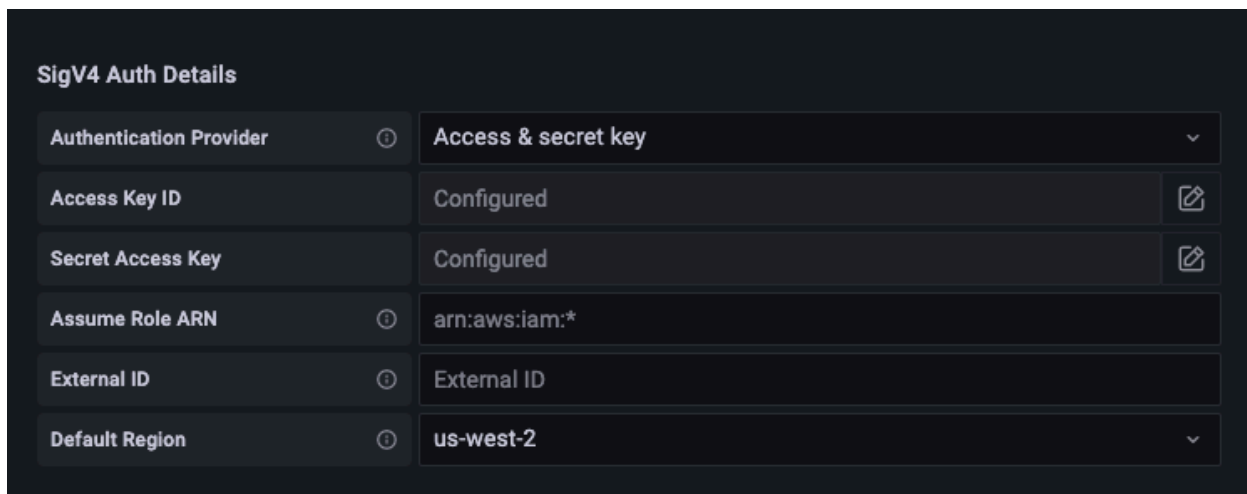
正しい URL は、`https://aps-workspaces.us-west-2.amazonaws.com/workspaces/ws-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` のようになります。

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. [引き受けロールの ARN] と [外部 ID] 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 で指定した URL に含まれてい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。

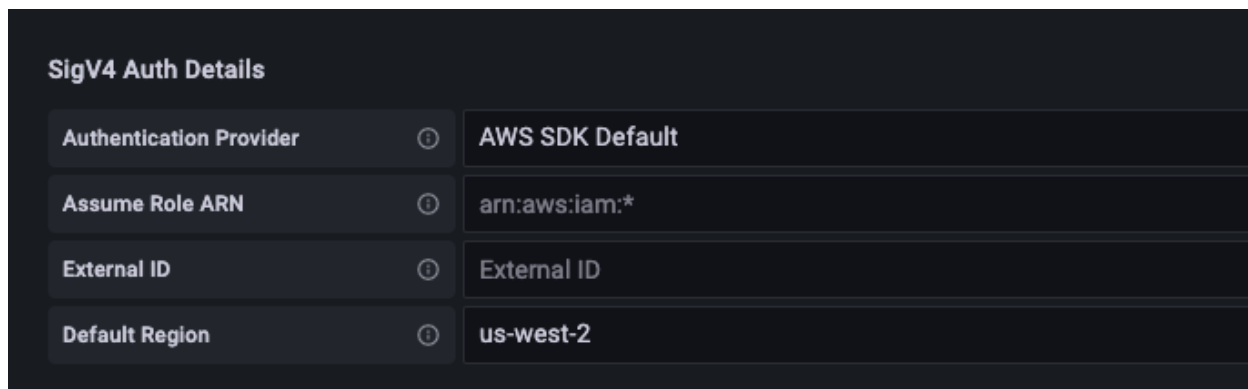


SigV4 Auth Details	
Authentication Provider	Access & secret key
Access Key ID	Configured
Secret Access Key	Configured
Assume Role ARN	arn:aws:iam:*
External ID	External ID
Default Region	us-west-2

- 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] として [AWS SDK のデフォルト] を選択します。
  - b. [引き受けロールの ARN] と [外部 ID] 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 で指定した URL に含まれてい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。

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



The screenshot shows a configuration panel titled "SigV4 Auth Details" with a dark background. It contains four rows, each with a label, an information icon, and a value:

SigV4 Auth Details		
Authentication Provider	ⓘ	AWS SDK Default
Assume Role ARN	ⓘ	arn:aws:iam:*
External ID	ⓘ	External ID
Default Region	ⓘ	us-west-2

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
```

## [保存してテスト] 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: 見つかりません

URL に含まれているワークスペース ID が正しいことを確認してください。

HTTP エラー: 禁止されています

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: 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 クラスターで動作する 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。`account-id` は、Grafana サーバーが実行されている AWS アカウント ID に置き換えます。

```
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
```

この YAML ファイル内の `amp-iamproxy-query-role` は、次の「[サービスアカウントの IAM ロールの設定](#)」セクションで作成するロールの名前です。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 ロールの設定

Amazon EKS クラスターで Grafana サーバーを使用している場合は、アクセス制御にサービスアカウントの IAM ロール (サービスロールとも呼ばれます) 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、「[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 URL として、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. 指定した HTTP URL から、URL に追加されている `/api/v1/query` という文字列を削除します。これは、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。

[引き受けロールの ARN] と [外部 ID] 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 API を使用したクエリ

メトリクスの表示とクエリを行うには [Amazon Managed Grafana](#) などのツールを使用する方法が最も簡単ですが、Amazon Managed Service for Prometheus では、メトリクスのクエリに使用できる Prometheus 互換 API もいくつかサポートされています。使用可能なすべての Prometheus 互換 API の詳細については、「[Prometheus 互換 API](#)」を参照してください。

これらの APIs をクエリする場合、リクエストは Signature Version 4 AWS 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、`awscli` を使用して実行できます `awscurl`。次のトピック「[awscurl を使用した Prometheus 互換 API のクエリ](#)」では、`awscurl` を使用して AWS SigV4 をセットアップする方法について説明します。

### awscurl を使用した Prometheus 互換 API のクエリ

Amazon Managed Service for Prometheus の API 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。#####、`Workspace-id`、および `QUERY` 入力をユースケースに適した値に置き換えます。

```
# 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 inferred 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する。ロールのアクセスキーとシークレットキーは、<https://console.aws.amazon.com/iam/> で確認できます。

例:

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

- EC2 インスタンスに関連付けられているインスタンスプロファイルを使用する。

## awscli 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 `awscli` 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 `awscli` をデプロイしますが、OCI 準拠の任意のランタイムとイメージを使用できます。

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

## クエリ API レスポンス内のクエリ統計情報

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。query または queryRange API のクエリレスポンスには、処理されたクエリサン

プルに関する統計データが含まれます。リクエストで `stats=all` クエリパラメータが送信されると、`stats` オブジェクト内に `samples` オブジェクトが作成され、その `stats` データが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
<code>totalQueryableSamples</code>	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。
<code>totalQueryableSamplesPerStep</code>	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  
      ]  
    ]  
  }  
}  
}
```



# 記録ルールとアラートルール

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、頻繁に必要な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。ルールによってしきい値がトリガーされると、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、その通知を Amazon Simple Notification Service などのダウンストリームのレシーバーに転送します。

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

- [必要な IAM アクセス許可](#)
- [ルールファイルの作成](#)
- [Amazon Managed Service for Prometheus へのルール設定ファイルのアップロード](#)
- [ルール設定ファイルの編集](#)

## • [ルーラーのトラブルシューティング](#)

# 必要な IAM アクセス許可

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。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。

### 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>"
      ]
    }
  ]
}
```

## ルールファイルの作成

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 へのルール設定ファイルのアップロード](#)。

## Amazon Managed Service for Prometheus へのルール設定ファイルのアップロード

ルール設定ファイルに加える変更がわかったら、コンソール内で編集するか、コンソールまたはを使用して代替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 では、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の編集

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
```



```
    "level": "ERROR",
    "name": "failure",
    "group": "canary_long_running_v1_namespace",
    "namespace": "canary_long_running_v1_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

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

スタンドアロンの Prometheus では、個別のテンプレートファイルを使用したテンプレート化がサポートされます。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

- [必要な IAM アクセス許可](#)
- [アラートマネージャー設定ファイルの作成](#)
- [アラートレシーバーの設定](#)
- [Amazon Managed Service for Prometheus へのアラートマネージャー設定ファイルのアップロード](#)
- [アラートと Amazon Managed Grafana またはオープンソースの Grafana の統合](#)
- [アラートマネージャーのトラブルシューティング](#)

## 必要な IAM アクセス許可

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。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": "*"
}
]
```

## アラートマネージャー設定ファイルの作成

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 [Amazon Managed Service for Prometheus へのアラートマネージャー設定ファイルのアップロード](#)。

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 }}
```

## アラートレシーバーの設定

現在 Amazon Managed Service for Prometheus で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。詳細については、「[Amazon SNS とは](#)」を参照してください。

### トピック

- [\(オプション\) 新しい Amazon SNS トピックの作成](#)
- [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の Amazon Managed Service for Prometheus への付与](#)
- [アラートマネージャー設定ファイルでの Amazon SNS トピックの指定](#)
- [\(オプション\) Amazon SNS に JSON を出力するためのアラートマネージャーの構成](#)
- [\(オプション\) Amazon SNS から他の宛先への送信](#)
- [SNS レシーバーのメッセージ検証と切り捨てルール](#)

## (オプション) 新しい Amazon SNS トピックの作成

既存の Amazon SNS トピックを使用することも、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。

## 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の Amazon Managed Service for Prometheus への付与

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。以下のポリシーステートメントには、「混乱した代理」のセキュリティ上の問題を防ぐために役立つ 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) SNS トピックでサービス側の暗号化 (SSE) が有効になっている場合は、"Action" ブロックで KMS キーポリシーに次のアクセス許可を追加する必要があります。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。

```
kms:GenerateDataKey
kms:Decrypt
```

## 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 トピックにメッセージを送信するためのアクセス許可の Amazon Managed Service for Prometheus への付与](#)」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。

## アラートマネージャー設定ファイルでの Amazon SNS トピックの指定

次に、アラートマネージャーの設定に Amazon SNS のレシーバーを追加します。そのためには、Amazon SNS トピックの Amazon リソースネーム (ARN) を知っ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「[\(オプション\) Amazon SNS に JSON を出力するためのアラートマネージャーの構成](#)」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、「[Amazon Managed Service for Prometheus へのアラートマネージャー設定ファイルのアップロード](#)」を参照してください。

## (オプション) Amazon SNS に JSON を出力するためのアラートマネージャーの構成

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して、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でアラートを処理できます。Amazon Managed Service for Prometheus のアラートマネージャーに用意されているデフォルトのテンプレートでは、メッセージペイロードがテキストリスト形式で出力されますが、これは解析しにくい場合があります。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、「[アラートマネージャー設定ファイルの作成](#)」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、「[Amazon Managed Service for Prometheus へのアラートマネージャー設定ファイルのアップロード](#)」を参照してください。

## (オプション) Amazon SNS から他の宛先への送信

現在、Amazon Managed Service for Prometheus でアラートメッセージを直接送信できる先は Amazon SNS だけです。Amazon SNS を設定して、これらのメッセージを E メール、ウェブフック、Slack、などの他の宛先に送信できます OpsGenie。

### 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 チャン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 チャン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。

## SNS レシーバーのメッセージ検証と切り捨てルール

SNS レシーバーは、以下のルールに基づいて、必要に応じて SNS メッセージの検証、切り捨て、変更を行います。

- 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 because 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。

## Amazon Managed Service for Prometheus へのアラートマネージャー設定ファイルのアップロード

アラートマネージャー設定ファイルに加える変更がわかったら、コンソール内で編集するか、コンソールまたは を使用して代替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。

## アラートと Amazon Managed Grafana またはオープンソースの Grafana の統合

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。

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

AWS アカウントと IAM 認証情報の作成の詳細については、「[セットアップ](#)」を参照してください。

- 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、「[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](#)」を参照してください。ルールの詳細については、「[記録ルールとアラートルール](#)」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](#)

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「[Grafana での Prometheus データソースの追加](#)」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-xyz0000001/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 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。

## 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](#) 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](#)

### 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-efdc5b42-b051-11ec-b123-4567ac120002",
```

```
"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-efdc5b42-b051-11ec-b123-4567ac120002",
  "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-efdc5b42-b051-11ec-b123-4567ac120002",
"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-efdc5b42-b051-11ec-b123-4567ac120002",
  "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-efdc5b42-b051-11ec-b123-4567ac120002",
"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 が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](#)

## ロギングとモニタリング

Amazon Managed Service for Prometheus リソースの使用状況は、Amazon の CloudWatch ログ記録およびモニタリング機能を使用して管理できます。

- Amazon Managed Service for Prometheus のモニタリングには、[CloudWatch メトリクス](#)を使用します。
- Amazon Managed Service for Prometheus のアラートマネージャーとルーラーイベントのクエリと表示には、[CloudWatch ログ](#)を使用します。

## CloudWatch メトリクス

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	<p>サンプルの取り込みレート</p> <p>単位: 1 秒あたりのカウント</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>
ResourceCount	ActiveSeries	AWS/Usage	<p>ワークスペースごとのアクティブなシリーズの数</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>

CloudWatch メトリクス名	リソース名	CloudWatch 名前空間	説明
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	SizeOfアラート	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	<p>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。</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>
ResourceCount	AllAlerts	AWS/Usage	<p>ワークスペースごとのすべての状態のアラート数。</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>
AlertManagerAlertsReceived	-	AWS/Prometheus	<p>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</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>
AlertManagerNotificationsFailed	-	AWS/Prometheus	<p>失敗したアラート配信の数</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>

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
DiscardedSamples*	-	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
RuleEvaluation失敗	-	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	<p>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。</p> <p>単位: 数</p> <p>有効な統計: Average、Minimum、Maximum、Sum</p>

\* 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、許可されている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	メトリクス名は指定されません。
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 デイメンションがあります。

*RuleGroup#### ; 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

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。

## コストの理解と最適化

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収集 (スクレイピング) 間隔を増やすことができます。Prometheus サーバー (エージェントモードで実行) AWS と Distro for OpenTelemetry (ADOT) コレクターの両方がこの構成をサポートしています。scrape\_interval 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](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？

アラート機能を使うと、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、コストを抑えるために使用できる方法をいくつか紹介します。

- Amazon Managed Service for Prometheus アラートを使用する — Amazon Managed Service for Prometheus の外部アラートシステムでは、外部サービスが複数のアベイラビリティ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、耐障害性や高可用性を高めるために追加のクエリが必要になる場合があります。これには、高可用性を実現するための Grafana でのアラートも含まれます。これにより、コストが 3 倍以上になることがあります。Amazon Managed Service for Prometheus のアラートは最適化されており、最小限のクエリ数で高い可用性と耐障害性を実現します。

外部アラートシステムではなく、Amazon Managed Service for Prometheus の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簡単な方法の 1 つは、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るが、5 分ごとにしか必要ない場合は、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を 5 倍節約できます。
- 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、取得されるデータが増えるため、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は、アラートする必要があるデータに適したサイズであることを確認してください。たとえば、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ています。

```
- alert: metric:alerting_rule
  expr: avg(rate(container_cpu_usage_seconds_total[10m])) > 0
  for: 2m
```

`expr`を変更すると、`avg(rate(container_cpu_usage_seconds_total[5m])) > 0`クエリのコストを削減できます。

一般的には、アラートルールを見て、サービスに最適なメトリクスでアラートを出していることを確認してください。同じメトリックに対して重複するアラートを作成したり、同じ情報を提供する複数のアラートを作成したりするのは簡単です。特に、時間の経過とともにアラートを追加する場合はなおさらです。複数のアラートが同時に発生することが多い場合は、すべてのアラートを含めるのではなく、アラートを最適化することができます。

これらの提案はコスト削減に役立ちます。最終的には、システムの状態を把握するための適切なアラートセットを作成することと、コストのバランスを取る必要があります。

Prometheus 用 Amazon マネージドサービスのアラートに関する詳細については、[を参照してください](#)。 [アラートマネージャー](#)

## 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 でモニタリングすれば CloudWatch、取り込みコストを追跡できます。の Amazon マネージドサービスの Prometheus メトリクスのモニタリングの詳細については、[を参照してください](#)。 CloudWatch [CloudWatch メトリクス](#)

## 請求書はいつでも確認できますか？

AWS Cost and Usage Report AWS はユーザーの使用状況を追跡し、請求期間内のアカウントに関連する推定請求額を表示します。詳細については、「[AWS コストと使用状況レポートとは](#)」を参照してください。『AWS コストと使用状況レポートのユーザーガイド』の

## 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**

- 階層は地域内での使用を目的としており、地域間での使用を目的としたものではありません。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レベルに達する必要があります。
- の組織では AWS Organizations、ティア使用量はアカウントごとではなく支払いアカウントごとに集計されます (支払いアカウントは常に組織の管理アカウントです)。組織内のすべてのアカウントの (リージョン内の) 取り込まれた指標の合計が次の階層に達すると、すべてのアカウントに低い方のレートが課金されます。

## Prometheus 用 Amazon マネージドサービスの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。何が起きているのでしょうか？

この場合の 1 つの可能性は、AWS 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ているマネージドスクレイパーがまだ存在していることです。の指示に従ってください。[スクレイパーの検出と削除](#)

## 他の AWS のサービスとの統合

Amazon Managed Service for Prometheus は、他の AWS の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) コストモニタリング (Kubecost を使用) との統合方法や、Terraform モジュールを使用して EKS プロジェクト用に AWS Observability Accelerator による完全なオブザーバビリティソリューションを作成する方法について説明します。

### トピック

- [Amazon EKS コストモニタリングとの統合](#)
- [AWS Observability Accelerator の使用](#)
- [AWS Controllers for Kubernetes との統合](#)
- [CloudWatch メトリクスと Firehose 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、シヨ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。

## AWS Observability Accelerator の使用

AWS には、Amazon Elastic Kubernetes Service (Amazon EKS) のプロジェクト向けに、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が用意されています。これには、Amazon Managed Service for Prometheus、[Amazon Managed Grafana](#)、[AWS Distro for OpenTelemetry](#)、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 には、インフラストラクチャのモニタリング、[NGINX](#) のデプロイ、その他のシナリオに向けた例が用意されています。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 の GitHub ページ](#)を参照してください。[AWS Observability Accelerator の発表に関するブログ記事](#)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](#) - AWS の機能をコマンドラインから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 の作成と管理を行うアクセス権を持つ IAM ロールを使用して、AWS プロバイダーをセットアップする必要があります。Terraform 用の AWS プロバイダーを構成する方法の詳細については、Terraform ドキュメントの「[AWS 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のセットアップと構成を行う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方法と、テンプレートに含まれている追加機能の詳細については、GitHub の「[Existing Cluster with the AWS Observability Accelerator base and Infrastructure monitoring](#)」ページ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 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 を使用してください。

```
# (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。`grafana-workspace-id` は、Grafana ワークスペースの ID に置き換えます。

```
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 terraform.tfvars ファイルに ID を追加します。 *prometheus-workspace-id* は Prometheus ワークスペース ID に置き換えます。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、AWS アカウントに以下のリソースが作成されます。

- 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 クラスター内にセットアップされた [AWS Distro for OpenTelemetry](#) オペレーター。Amazon Managed Service for Prometheus ワークスペースにメトリクスを送信するために使用されます。

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。

## AWS Controllers for Kubernetes との統合

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 オブジェクトは、クラスター外のリソースを定義しなくても使用できます。

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 をセットアップする方法について説明します。

Controllers [AWS for Kubernetes](#) を紹介し、[Amazon Managed Service for Prometheus の ACK コントローラーを紹介](#)するブログ記事もお読みください。

## 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、次の前提条件を満たす必要があります。

- 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 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「[OIDC アイデンティティプロバイダー認証](#)」と「[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アラートを送信する先の [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 Controller 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て、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「Install an [ACK](#) AWS Controller」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 Controllers for Kubernetes AWS Controller 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` に関する情報が返され、`status: deployed` というステータスを確認できます。

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** は通知を送信する Amazon SNS トピックの ARN に置き換え、**REGION** は使用しているに置き換え AWS リージョンです。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「[AlertManagerDefinitionData](#)」を参照してください。

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、yml 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`、および `o` を使用して `PrometheusNamespace`、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、`awscli` 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` を使用中 AWS リージョンの `region` に置き換え、`workspaceURL/` をステップ 1 で見つけた URL に置き換えます。

```
awscli --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 Controller を使用して、すべての Amazon Managed Service for Prometheus 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 Firehose の統合

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設定し、Firehose 配信ストリーム、Lambda、および Amazon S3 バケットを作成して、完全なシナリオをデモンストレーションします。

### 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 が環境に [インストール](#) および [設定されています](#)。
- [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トの github リポジトリ](#) (CWMetricsStreamExporter) がローカルマシンに複製されました。

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きな名前に変更します。

2. AWS CDK と Lambda コードをビルドするには、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 CloudFormation にデプロイされます。

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. <https://console.aws.amazon.com/cloudwatch/home#metric-streams:streamsList> で CloudWatch コンソールに移動し、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

- [Amazon Managed Service for Prometheus でのデータ保護](#)
- [Amazon Managed Service for Prometheus の Identity and Access Management](#)
- [IAM のアクセス許可とポリシー](#)
- [Amazon Managed Service for Prometheus のコンプライアンス検証](#)
- [Amazon Managed Service for Prometheus の耐障害性](#)
- [Amazon Managed Service for Prometheus のインフラストラクチャセキュリティ](#)
- [Amazon Managed Service for Prometheus のサービスリンクロールの使用](#)

- [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

責任 AWS [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に対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「[データプライバシーのよくある質問](#)」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された記事「[AWS 責任共有モデルおよび GDPR](#)」を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、内のすべてのデフォルトのセキュリティコントロールを使用します AWS 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。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「[連邦情報処理規格 \(FIPS\) 140-2](#)」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、コンソール、API、または AWS CLI SDK を使用して Amazon Managed Service for Prometheus または他の AWS のサービスを使用する場合も同様です。AWS SDKs 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。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

## トピック

- [Amazon Managed Service for Prometheus によって収集されるデータ](#)
- [保管中の暗号化](#)

## 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 ID により、さまざまな顧客からのデータが分離されます。これらの ID は、どの顧客データにアクセスできるかを制限します。顧客がテナント ID を変更することはできません。

Amazon Managed Service for Prometheus は、AWS Key Management Service (AWS KMS) キーで保存するデータを暗号化します。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、独自のカスターマネージドキーの使用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 HTTPS で自動的に暗号化されます。Amazon Managed Service for Prometheus は、内部的に HTTPS を使用して、AWS リージョン内のアベイラビリティーゾーン間の接続を保護します。

## 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、AWS 所有の暗号化キーを使用してこれを行います。

- 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。AWS 所有キーを表示、管理、使用したり、その使用を監査したりすることはできません。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、AWS Key Management Service デベロッパーガイドの「[AWS 所有キー](#)」を参照してください。

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 ポリシーとグラントの策定と維持
  - キーポリシーの有効化と無効化
  - 暗号化素材のローテーション
  - タグの追加
  - キーエイリアスの作成
  - 削除のためのキースケジューリング

詳細については、「AWS Key Management Service デベロッパーガイド」の「[カスタマーマネージドキー](#)」を参照してください。

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で作成されたワークスペースは、後で 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 (その逆も同様です)。

**Note**

Amazon Managed Service for Prometheus は、AWS 所有キーを使用して保管中の暗号化を自動的に有効にし、データを無料で保護します。

ただし、カスタマーマネージドキーの使用には AWS KMS 料金が適用されます。料金の詳細については、「[AWS Key Management Service の料金](#)」を参照してください。

の詳細については 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、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。

## 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 [CreateGrant](#) リクエストを送信して、ユーザーに代わって 3 つの許可を作成します AWS KMS。の許可 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ピング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。

- 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する [DescribeKey](#) リクエストを AWS KMS に送信します。
- 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 AWS KMS には、[GenerateDataKey](#) リクエストを送信します。
- [Decrypt](#) リクエストを AWS KMS に送信して、暗号化されたデータキーを復号し、データの暗号化に使用できます。



Amazon Managed Service for Prometheus は、Amazon Managed Service for Prometheus がユーザーに代わって AWS KMS キーを使用できるようにする 3 つの許可をキーに作成します。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

対称カスタマーマネージドキーを作成するには AWS Management Console、[AWS CLI](#)、または AWS KMS APIs を使用します。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

AWS Key Management Service デベロッパーガイドにある [対称カスタマーマネージドキーの作成](#) ステップを実行します。

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、AWS Key Management Service デベロッパーガイドの「[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。

Amazon Managed Service for Prometheus 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、「AWS Key Management Service デベロッパーガイド」の「[許可の使用](#)」を参照してください。

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、保存されている暗号化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。
- [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>
]
```

- [ポリシーでの許可の指定](#)に関する詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。
- [キーアクセスのトラブルシューティング](#)に関する詳細については、「AWS Key Management Service デベロッパーガイド」を参照してください。

## ステップ 2: Amazon Managed Service for Prometheus のカスタ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は、Amazon Managed Service for Prometheus がワークスペースに保存されているデータを暗号化するために使用する KMS キー ARN を入力して、カスタ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にアクセスする

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、AWS KMS キー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することはできません。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 `AmazonGrafanaServiceRole-  
<unique-id>`。コンソールには、ロールの完全な ARN が表示されます。この名前は、後のステップでコンソールで指定します AWS KMS 。
3. [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、IAM ロールをキーユーザーのリストに追加します。

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。データを暗号化するリクエストに暗号化コンテキストを含めると、 は暗号化コンテ

キストを暗号化されたデータに AWS KMS 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、すべての暗号化オペレーションで同じ AWS KMS 暗号化コンテキストを使用します。キーは `aws:aps: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、[AWS CloudTrail](#) または [Amazon CloudWatch Logs](#) によって生成されたログにも表示されます。

### 暗号化コンテキストを使用してカスターマネージドキーへのアクセスを制御する

対称カスターマネージドキー (CMK) へのアクセスを制御するための `conditions` として、キーポリシーと IAM ポリシー内の暗号化コンテキストを使用することができます。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

Amazon Managed Service for Prometheus ワークスペースで AWS KMS カスタマーマネージドキーを使用する場合、[AWS CloudTrail](#)または[Amazon CloudWatch Logs](#)を使用して、Amazon Managed Service for Prometheus が送信するリクエストを追跡できます AWS KMS。

次の例はCreateGrant、Amazon Managed Service for Prometheus がカスタマーマネージドキーで暗号化されたデータにアクセスするために呼び出す KMS オペレーションをモニタリングDescribeKeyするための GenerateDataKeyDecrypt、、、およびの AWS CloudTrail イベントです。

### CreateGrant

AWS KMS カスタマーマネージドキーを使用してワークスペースを暗号化すると、Amazon Managed Service for Prometheus は、指定した KMS キーにアクセスするために、ユーザーに代わって 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、AWS KMS カスタマーマネージドキーに関連付けられたリソースに固有のものであります。

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

ワークスペースの AWS KMS カスタマーマネージドキーを有効にすると、Amazon Managed Service for Prometheus によって一意のキーが作成されます。リソースの AWS KMS カスタマーマネージドキー AWS KMS を指定する GenerateDataKey リクエストを に送信します。

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": "57f5dbec-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、ワークスペースに関連付けられている AWS KMS カスタマーマネージドキーがアカウントとリージョンに存在するかどうかを確認します。

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"  
}
```

## 詳細はこちら

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。

- 詳細については、「AWS Key Management Service デベロッパーガイド」の「[AWS Key Management Service の概念](#)」を参照してください。
- [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、「AWS Key Management Service AWS Key Management Service デベロッパーガイド」](#)を参照してください。

## 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、誰を認証 (サインイン) し、誰に Amazon Managed Service for Prometheus リソースの使用を認可する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで AWS 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 と IAM の連携](#)
- [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 방법은、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、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点検して、IAM の基本概念を理解してください。会社で Amazon Managed Service for Prometheus と IAM を併用する方法の詳細については、「[Amazon Managed Service for Prometheus と IAM の連携](#)」を参照してください。

IAM 管理者 - IAM 管理者は、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を詳しく理解することが必要になる場合があります。IAM 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確認するには、「[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。

## アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証 ( にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーテッド ID AWS として にサインインできます。AWS IAM Identity Center ( IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーテッド ID の例です。フェデレーテッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることとなります。

ユーザーのタイプに応じて、AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「ユーザーガイド」の「[へのサインイン AWS アカウント](#)方法AWS サインイン」を参照してください。

AWS プログラムで にアクセスする場合、は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストに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 API AWS リクエスト](#) の署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させることをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「[多要素認証](#)」および「IAM ユーザーガイド」の「[AWSでの多要素認証 \(MFA\) の使用](#)」を参照してください。

## AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての AWS のサービス およびリソースへの完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「IAM ユーザーガイド」の「[ルートユーザー認証情報が必要なタスク](#)」を参照してください。

## フェデレーテッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービス します。

フェデレーテッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーテッド ID が にアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。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](#)は、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウントを持つ内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM ユーザーガイドの「[長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする](#)」を参照してください。

[IAM グループ](#)は、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、「IAM ユーザーガイド」の「[IAM ユーザー \(ロールではなく\) の作成が適している場合](#)」を参照してください。

## IAM ロール

[IAM ロール](#)は、特定のアクセス許可 AWS アカウントを持つ内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロールを切り替える AWS Management Console ことで、[IAM ロール](#)を一時的に引き受けることができます。ロールを引き受けるには、または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「[IAM ロールの使用](#)」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「[Creating a role for a third-party Identity Provider](#)」(サードパーティーアイデンティティプロバイダー向けロールの作成)を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスでき

るものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center 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 ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の AWS サービス、(ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、IAM ユーザーガイドの「[IAM でのクロスアカウントリソースアクセス](#)」を参照してください。
- クロスサービスアクセス — 一部の AWS の機能は、他の AWS のサービスを使用します。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
- 転送アクセスセッション (FAS) — IAM ユーザーまたはロールを使用してアクションを実行する場合 AWS、ユーザーは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、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と組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービスまたはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。
- サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「[AWS のサービスにアクセス許可を委任するロールの作成](#)」を参照してください。
- サービスにリンクされたロール - サービスにリンクされたロールは、にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールはに表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスリンクロールの許可を表示できますが、編集することはできません。



- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「[Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する](#)」を参照してください。

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「[\(IAM ユーザーではなく\) IAM ロールをいつ作成したら良いのか?](#)」を参照してください。

## ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、AWS ID またはリソースにアタッチします。ポリシーは、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) AWS 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 として保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「[JSON ポリシー概要](#)」を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、AWS Management Console、AWS CLI または AWS API からロール情報を取得できます。

### アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティ

ベースのポリシーを作成する方法については、「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。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「[マネージドポリシーとインラインポリシーの比較](#)」を参照してください。

## リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#)必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーテッドユーザー、またはを含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

## アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、『Amazon Simple Storage Service デベロッパーガイド』の「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。

## その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- **アクセス許可の境界** - アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。Principal フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可の境界](#)」を参照してください。
- **サービスコントロールポリシー (SCPs)** – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。AWS Organizations は、AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するためのサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「[SCP の仕組み](#)」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、「IAM ユーザーガイド」の「[セッションポリシー](#)」を参照してください。

## 複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「[ポリシー評価ロジック](#)」を参照してください。

## Amazon Managed Service for Prometheus と IAM の連携

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 で利用できる IAM の機能を理解しておく必要があります。

## Amazon Managed Service for Prometheus で使用できる IAM 機能

IAM 機能	Amazon Managed Service for Prometheus でのサポート
<a href="#">アイデンティティベースのポリシー</a>	Yes
<a href="#">リソースベースのポリシー</a>	No
<a href="#">ポリシーアクション</a>	Yes
<a href="#">ポリシーリソース</a>	Yes
<a href="#">ポリシー条件キー</a>	いいえ
<a href="#">ACL</a>	No
<a href="#">ABAC (ポリシー内のタグ)</a>	はい
<a href="#">一時的な認証情報</a>	はい
<a href="#">転送アクセスセッション (FAS)</a>	いいえ
<a href="#">サービスロール</a>	いいえ
<a href="#">サービスリンクロール</a>	はい

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 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」の「IAM [AWS と連携する のサービス](#)」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

アイデンティティベースポリシーをサポートする	Yes
------------------------	-----

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティ

ベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAM ユーザーガイド」の「[IAM JSON ポリシーの要素のリファレンス](#)」を参照してください。

## Amazon Managed 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	No
-------------------	----

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、[プリンシパルを指定する](#) 必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる がある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与

する必要はありません。詳細については、[「IAM ユーザーガイド」の「IAM でのクロスアカウントリソースアクセス」](#)を参照してください。

## Amazon Managed Service for Prometheus のポリシーアクション

ポリシーアクションに対するサポート	はい
-------------------	----

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

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

ポリシーリソースに対するサポート	はい
------------------	----

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件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。

オペレーションの一覧表示など、リソースレベルのアクセス許可をサポートしないアクションでは、ワイルドカード (\*) を使用して、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します。

```
"Resource": "*"
```

Amazon Managed Service for Prometheus リソースタイプとその ARN 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 で定義されるリソース](#)」を参照してください。どのアクションで各リソースの ARN を指定できるかについては、「[Amazon Managed 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のサポート	いいえ
----------------------	-----

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件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 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、AWS は AND 論理演算子を使用してそれら进行评估します。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 を评估します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「[IAM ポリシーの要素: 変数およびタグ](#)」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」の [AWS 「グローバル条件コンテキストキー」](#) を参照してください。

Amazon Managed 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 (ACL)

ACL のサポート	No
-----------	----

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

## Amazon Managed Service for Prometheus での属性ベースのアクセス制御 (ABAC)

ABAC のサポート (ポリシー内のタグ)	はい
-----------------------	----



属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義するアクセス許可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの [条件要素](#) でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、IAM ユーザーガイドの「[ABAC とは?](#)」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「[属性ベースのアクセス制御 \(ABAC\) を使用する](#)」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	はい
---------------	----

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「IAM と連携する」](#) を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で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 Single Sign-On (SSO) リンク AWS を使用してアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「[ロールへの切り替え \(コンソール\)](#)」を参照してください。

一時的な認証情報は、AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して、AWS recommends にアクセスできます AWS。これは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することを推奨しています。詳細については、「[IAM の一時的セキュリティ認証情報](#)」を参照してください。

## Amazon Managed 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 (FAS) をサポート いいえ

IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「[転送アクセスセッション](#)」を参照してください。

## Amazon Managed Service for Prometheus のサービスロール

サービスロールのサポート いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける [IAM ロール](#)です。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「[AWS のサービスにアクセス許可を委任するロールの作成](#)」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 はい

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービ

スにリンクされたロールは に表示され AWS アカウント、サービスによって所有されます。IAM 管理者は、サービスリンクロールの許可を表示できますが、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 AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースに必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

Amazon Managed Service for Prometheus で定義されるアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「[Amazon Managed 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、または削除できるかどうかを決定します。これらのアクションを実行すると、AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらは使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「[AWS マネージドポリシー](#)」または「[AWS ジョブ機能の管理ポリシー](#)」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、「IAM ユーザーガイド」の「[IAM でのポリシーとアクセス許可](#)」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、などの特定の を介してサービスアクションが使用される場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、「IAM ユーザーガイド」の「[IAM JSON policy elements: Condition](#)」(IAM JSON ポリシー要素: 条件) を参照してください。
- IAM Access Analyzer を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM Access Analyzer は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、「IAM ユーザーガイド」の「[IAM Access Analyzer ポリシーの検証](#)」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、「IAM ユーザーガイド」の「[MFA 保護 API アクセスの設定](#)」を参照してください。

IAM でのベストプラクティスの詳細については、「IAM ユーザーガイド」の「[IAM でのセキュリティのベストプラクティス](#)」を参照してください。

## Amazon Managed 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 AWS API 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。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、エンティティに Amazon Managed Service for Prometheus ConsoleAccess または ReadOnly AWS 管理ポリシーもアタッチします。詳細については、「IAM 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。

## 自分の権限の表示をユーザーに許可する

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API 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。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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/scrapper.aps.amazonaws.com/AWSServiceRoleForAmazonPrometheusScrapper*",
    }
  ]
}
```

```
"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 : AmazonPrometheusScrapperServiceRolePolicy

IAM エンティティ AmazonPrometheusScrapperServiceRolePolicy 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。  
AmazonPrometheusScrapperServiceLinkedRolePolicy

## 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/scrapper.aps.amazonaws.com/AWSServiceRoleForAmazonPrometheusScrapper*"
    },
    {
      "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": [
            "AMPAgentlessScrapper"
          ]
        }
      }
    },
    {
      "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
<a href="#">AmazonPrometheusScrapingServiceRolePolicy</a> - 既存ポリシーへの更新	<p>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 AmazonPrometheusScrapingServiceRolePolicy を追加しました。</p> <p>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。</p>	2024 年 5 月 2 日

変更	説明	日付
	<p> Note</p> <p>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</p>	
<p><a href="#">AmazonPrometheusFullAccess</a> – 既存ポリシーへの更新</p>	<p>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。</p> <p>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。</p>	2023 年 11 月 26 日

変更	説明	日付
<p><a href="#">AmazonPrometheusScrapingServiceLinkedRolePolicy</a> - 新しいポリシー</p>	<p>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。</p> <p>Amazon EKS クラスターへの接続、Amazon EC2 ネットワークの読み取り、AMPAgentlessScrape r 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。</p>	2023 年 11 月 26 日

変更	説明	日付
<a href="#">AmazonPrometheusConsoleFullAccess</a> – 既存ポリシーへの更新	<p>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。</p> <p>aps:CreateLoggingConfiguration、aps:UpdateLoggingConfiguration、aps:DeleteLoggingConfiguration、aps:DescribeLoggingConfiguration のアクセス許可が追加されました。</p>	2022 年 10 月 24 日



変更	説明	日付
<a href="#">AmazonPrometheusConsoleFullAccess</a> – 既存ポリシーへの更新	<p>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。</p> <p>tag:GetTagKeys 、 tag:GetTagValues 、 aps:CreateAlertManagerDefinition 、 aps:CreateRuleGroupsNamespace 、 aps&gt;DeleteAlertManagerDefinition 、 aps&gt;DeleteRuleGroupsNamespace 、 aps:DescribeAlertManagerDefinition 、 aps:DescribeRuleGroupsNamespace 、 aps:ListRuleGroupsNamespaces 、 aps:PutAlertManagerDefinition 、 aps:PutRuleGroupsN</p>	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

以下の情報は、Amazon Managed Service for Prometheus と IAM の使用時に発生する可能性のある一般的な問題の診断と修復に役立ちます。

### 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 アカウント以外のユーザーに 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、mateojackson IAM ユーザーがコンソールを使用して、ある *my-example-widget* リソースに関する詳細情報を表示しようとしたことを想定して、その際に必要な *aps:GetWidget* アクセス許可を持っていない場合に発生するものです。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
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。

次の例は、`marymajor` という IAM ユーザーがコンソールを使用して Amazon Managed Service for Prometheus でアクションを実行しようとした場合に発生するエラーを示しています。ただし、このアクションをサービスで実行するには、サービスロールから付与されたアクセス許可が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

## AWS アカウント以外のユーザーに Amazon Managed 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 (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- Amazon Managed 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 と IAM の連携](#)」を参照してください。

- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、[IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」](#)を参照してください。
- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「[サードパーティー AWS アカウント が所有する へのアクセスを提供する](#)」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの[外部認証されたユーザーへのアクセスの提供 \(ID フェデレーション\)](#)を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いについては、IAM ユーザーガイドの「[IAM でのクロスアカウントリソースアクセス](#)」を参照してください。

## 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。	<code>aps:CreateAlertManagerAlerts</code>
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、「 <a href="#">アラートマネージャー</a> 」を参照してください。	<code>aps:CreateAlertManagerDefinition</code>

[アクション]	必要なアクセス許可
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、「 <a href="#">記録ルールとアラートルール</a> 」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
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

[アクション]	必要なアクセス許可
Amazon Managed Service for Prometheus メトリクスのメタデータを取得する。	<code>aps:GetMetricMetadata</code>
時系列データを取得する。	<code>aps:GetSeries</code>
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。	<code>aps:ListAlertManagerAlertGroups</code>
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。	<code>aps:ListAlertManagerAlerts</code>
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。	<code>aps:ListAlertManagerReceivers</code>
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。	<code>aps:ListAlertManagerSilences</code>
アクティブなアラートのリストを取得する。	<code>aps:ListAlerts</code>
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。	<code>aps:ListRules</code>
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。	<code>aps:ListRuleGroupsNamespaces</code>
Amazon Managed Service for Prometheus に関連付けられているタグを取得する。	<code>aps:ListTagsForResource</code>
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。	<code>aps:ListWorkspaces</code>
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。	<code>aps:PutAlertManagerDefinition</code>

[アクション]	必要なアクセス許可
アラートサイレンスを作成する。	<code>aps:PutAlertManagerSilences</code>
既存のルールグループ名前空間を更新する。	<code>aps:PutRuleGroupsNamespace</code>
Amazon Managed Service for Prometheus メトリクスに対するクエリを実行する。	<code>aps:QueryMetrics</code>
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。	<code>aps:RemoteWrite</code>
Amazon Managed Service for Prometheus リソースにタグを割り当てる。	<code>aps:TagResource</code>
Amazon Managed Service for Prometheus リソースからタグを削除する。	<code>aps:UntagResource</code>
既存のワークスペースのエイリアスを変更する。	<code>aps:UpdateWorkspaceAlias</code>
ログ記録設定を作成する。	<code>aps:CreateLoggingConfiguration</code>
ログ記録設定を削除する。	<code>aps&gt;DeleteLoggingConfiguration</code>
ログ記録設定を記述する。	<code>aps:DescribeLoggingConfiguration</code>
ログ記録設定を更新する。	<code>aps:UpdateLoggingConfiguration</code>

## 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 サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャー](#) – 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 AWS のサービス HIPAA の対象となるわけではありません。詳細については、「[HIPAA 対応サービスのリファレンス](#)」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) を含む) 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確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「[Security Hub のコントロールリファレンス](#)」を参照してください。
- [Amazon GuardDuty](#) – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS 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、PCI DSS 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ゾーンがあります。アベイラビリティゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリ

ティーゾーンは、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 のセキュリティサービスと、AWS がインフラストラクチャをどのように保護するかについては、「[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、「セキュリティの柱 - AWS Well-Architected フレームワーク」の「[インフラストラクチャ保護](#)」を参照してください。

ネットワーク経由で Amazon Managed Service for Prometheus にアクセスするには、AWS が公開している API コールを使用します。クライアントは以下をサポートする必要があります：

- Transport Layer Security (TLS)。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイート。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。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) AWS STS を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

## 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 が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ある場合は、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 AWS CloudTrail、Amazon Managed Service for Prometheus のユーザー、ロール、または AWS のサービス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。は、Amazon Managed Service for Prometheus のすべての API コールをイベントとして CloudTrail キャプチャします。キャプチャされたコールには、Amazon Managed Service for Prometheus コンソールからのコールと、コードからの Amazon Managed Service for Prometheus API オペレーションのコールが含まれます。証跡を作成する場合は、Amazon Managed Service for Prometheus の CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。Amazon S3 証跡を設定しない場合でも、コンソールのイベント履歴で最新の CloudTrail イベントを表示できます。で収集された情報を使用して CloudTrail、Amazon Managed Service for Prometheus に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

の詳細については CloudTrail、「[AWS CloudTrail ユーザーガイド](#)」を参照してください。

## の Amazon Managed Service for Prometheus 情報 CloudTrail

CloudTrail AWS アカウントを作成すると、がアカウントで有効になります。Amazon Managed Service for Prometheus でアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴の他の AWS サービスイベントとともにイベントに記録されます。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、「[イベント履歴を使用した CloudTrail イベントの表示](#)」を参照してください。

Amazon Managed Service for Prometheus のイベントなど、AWS アカウントのイベントの継続的な記録については、証跡を作成します。証跡により、はログファイル CloudTrail を Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成すると、証跡はすべての AWS リージョンに適用されます。証跡は、AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをより詳細に分析し、それに基づく対応するように、他の AWS サービスを設定できます。詳細については、次を参照してください:

- [「追跡の作成の概要」](#)
- [CloudTrail でサポートされているサービスと統合](#)
- [CloudTrail の Amazon SNS 通知の設定](#)
- [複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信](#)

Amazon Managed Service for Prometheus では、次のアクションのログ記録がサポートされています。

- [CreateAlertManagerAlerts](#)
- [CreateAlertManagerDefinition](#)
- [CreateRuleGroupsNamespace](#)
- [CreateWorkspace](#)
- [DeleteAlertManagerDefinition](#)
- [DeleteAlertManagerSilence](#)
- [DeleteWorkspace](#)
- [DeleteRuleGroupsNamespace](#)
- [DescribeAlertManagerDefinition](#)
- [DescribeRulesGroupsNamespace](#)
- [DescribeWorkspace](#)
- [ListRuleGroupsNamespaces](#)
- [ListWorkspaces](#)
- [PutAlertManagerDefinition](#)
- [PutAlertManagerSilences](#)
- [PutRuleGroupsNamespace](#)

- [UpdateWorkspaceAlias](#)

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、以下を判別するために役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザーの認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって送信されたかどうか。

詳細については、「[CloudTrail userIdentity 要素](#)」を参照してください。

## Amazon Managed Service for Prometheus のログファイルエントリについて

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、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、特定の順序では表示されません。

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":
"YWxlcnRtYW5hZ2VyX2NvbWZpZzogfAogIGdsb2JhbDoKICAgIHNTdHBfc21hcnRob3N00iAnbG9jYWxob3N00jI1JwogI
    "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":
      "Z3JvdXBz0gogIC0gYmFtZTogdGVzdFJ1bGVHcm91cHN0YW1lc3BhY2UKICAgIHJ1bGVz0gogICAgLSBhbGVydDogdGVzd
      "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"
}
```

## 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 EKSCONTROL 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イ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\_clusternamespace> 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_clusternamespace>
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\_clusternamespace> 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。

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"
        }
      }
    }
  ]
}
```

```
}
```

# 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](#)
- [サンプルが重複している](#)
- [サンプルタイムスタンプに関するエラーが表示される](#)
- [制限に関するエラーメッセージが表示される](#)
- [ローカル Prometheus サーバーの出力が制限を超えている](#)
- [データの一部が表示されない](#)

## 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 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。メトリクスがタイムスタンプを直接提供する場合は、それらが順序どおりに設定され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 Out-of-order Timestamp Errors in Prometheus](#) from Prometheus 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	<code>prometheus_tsdb_head_series</code>
現在の取り込みレート	<code>rate(prometheus_tsdb_head_samples_appended_total[5m])</code>
メトリクス名あたりのアクティブなシリーズの Most-to-least リスト	<code>sort_desc(count by(__name__))</code> <code>({__name__!=""})</code>
メトリクスシリーズごとのラベル数	<code>group by(mylabelname)</code> <code>({__name__!=""})</code>

## 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
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

# タグ付け

タグは、ユーザーまたは AWS が AWS リソースに割り当てるカスタム属性ラベルです。各 AWS タグは 2 つの部分で構成されます。

- 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 リソースの識別や整理に役立ちます。多くの AWS のサービスではタグ付けがサポートされるため、さまざまなサービスからリソースに同じタグを割り当てて、リソースの関連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。これらのリソースのタグを追加、管理、削除するには、コンソール、AWS CLI、API、または SDK を使用できます。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

タグには以下のような基本制限があります。

- 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。

- 複数の AWS のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数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 でのみ使用するように予約されています。このプレフィックスを持つタグのキーや値を編集または削除することはできません。このプレフィックスを持つタグは、リソースごとのタグ数の制限にはカウントされません。

## トピック

- [ワークスペースのタグ付け](#)
- [ルールグループ名前空間のタグ付け](#)

## ワークスペースのタグ付け

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

- [ワークスペースへのタグの追加](#)
- [ワークスペースのタグの表示](#)
- [ワークスペースのタグの編集](#)
- [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。Amazon Managed Service for Prometheus ワークスペースにタグを追加するには、コンソールまたは AWS CLI を使用できます。

#### 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、「[ワークスペースの作成](#)」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)

AWS CLI を使用して Amazon Managed Service for Prometheus ワークスペースにタグを追加するには、以下の手順に従います。ワークスペースの作成時にタグを追加するには、「[ワークスペースの作成](#)」を参照してください。

以下のステップでは、AWS CLI の最新版を既にインストールしているか、最新版に更新しているものと想定します。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 リソースの識別と整理、アクセスの管理に役立ちます。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 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)

AWS CLI を使用してワークスペースの AWS タグを表示するには、以下の手順に従います。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がある 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)

AWS CLI を使用してワークスペースのタグを更新するには、以下の手順に従います。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た他の AWS リソースからタグを削除することにはなりません。

### 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)

AWS CLI を使用してワークスペースからタグを削除するには、以下の手順に従います。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

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 のルールグループ名前空間にタグを追加すると、AWS リソースの識別と整理、アクセスの管理に役立ちます。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。Amazon Managed Service for Prometheus のルールグループ名前空間にタグを追加するには、コンソールまたは AWS CLI を使用できます。

### 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、「[ルールファイルの作成](#)」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)

AWS CLI を使用して Amazon Managed Service for Prometheus のルールグループ名前空間にタグを追加するには、以下の手順に従います。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 へのルール設定ファイルのアップロード](#)」を参照してください。

以下のステップでは、AWS CLI の最新版を既にインストールしているか、最新版に更新しているものと想定します。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 リソースの識別と整理、アクセスの管理に役立ちます。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 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)

AWS CLI を使用してルールグループ名前空間の AWS タグを表示するには、以下の手順に従います。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 arn: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)

AWS CLI を使用してルールグループ名前空間のタグを更新するには、以下の手順に従います。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た他の AWS リソースからタグを削除することにはなりません。

### 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. [タグを管理] 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)

AWS CLI を使用してルールグループ名前空間からタグを削除するには、以下の手順に従います。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 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。

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。ワークスペースごとの取り込みレートとワークスペースごとの取り込みバーストサイズは、ワークスペースにデータを取り込む速度を制御します。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	いいえ	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。
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	<u>はい</u>	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	<u>はい</u>	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。
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	いいえ	アラートマネージャー定義ファイルの最大サイズ。

名前	デフォルト	引き上げ可能	説明
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	はい	ワークスペースあたりのすべてのアラートマネージャー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	<u>はい</u>	ワークスペースあたりの同時アラートマネージャーアラートの最大数。
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	はい	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	<u>はい</u>	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	<u>はい</u>	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	<a href="#">はい</a>	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	はい え	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	<a href="#">はい</a>	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	はい え	メトリクスのメタデータに許容される最大長。メタデータとは、Metric Name、HELP、UNIT 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	はい え	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	<a href="#">はい</a>	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内の 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	<a href="#">はい</a>	リージョンごとの 1 秒あたりの API オペレーションの最大数。これには、ワークスペースの CRUD API、タグ付け API、名前空間 CRUD API ルールグループ、アラートマネージャ定義 CRUD API が含まれます。
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	はい はい え	1 つのインスタントクエリでスキャンできる最大バイト数。
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	はい はい え	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	はい はい え	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	はい はい え	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	はい はい え	1 つのクエリでスキャンできるシリーズの最大数。

名前	デフォルト	引き上げ可能	説明
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	いいえ	PromQL クエリの最大時間範囲。
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	いいえ	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	<a href="#">はい</a>	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	<a href="#">はい</a>	ワークスペースあたりの、ルールグループのルール評価間隔の最小値。
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	いいえ	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	<a href="#">はい</a>	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	<a href="#">はい</a>	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。

名前	デフォルト	引き上げ可能	説明
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	<u>はい</u>	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、特定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 (ワークスペースごと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 秒間に 100 万個のデータサンプルを取り込むことができます。取り込むサンプルが 100 万を超えると、スロットリングされ、それ以上レコードは取り込まれません。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場合、設定された数のトークンが最大容量に達するまで毎秒追加されます。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、それらは破棄されます。バケットは最大トークン数を超えるトークンを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制限によって設定されます。ワークスペースごとの取り込みレートが 170,000 に設定されている場合、バケットの補充レートは 1 秒あたり 170,000 トークンになります。

ワークスペースが 1 秒間に 1,000,000 個のデータサンプルを取り込む場合、バケットはすぐにゼロトークンに削減されます。バケットは、最大容量の 1,000,000 トークンに達するまで、毎秒 170,000 トークンずつ補充されます。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。100 個のトークンが使用可能で、101 個のサンプルを含む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は受け付けません。コレクターを作成する場合は、再試行を管理できます (小さなバッチを使用するか、しばらく時間が経過した後)。



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前に、バケットがいっぱいになるのを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、バケットを使い切ると、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、取り込むデータサンプルが1秒あたり170,000個未満の場合にのみ、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。

# API リファレンス

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。

これらの API オペレーションと、シリーズ、ラベル、API リクエストに対する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 のサービスクォータ](#)」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 には、Amazon Managed Service for Prometheus ワークスペースを作成および管理する API オペレーションが用意されています。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、デベロッパーが希望する言語で 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 やその他の 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 API を使用したクエリ](#)。

## 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。アラートオ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": {
```



```
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
<a href="#">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</a>	Amazon Managed Service for Prometheus では、Amazon Managed Service for Prometheus コンソール内から <a href="#">アラートマネージャー設定ファイル</a> と <a href="#">ルール定義ファイル</a> 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
<a href="#">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</a>	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、 <a href="#">AWS マネージドコレクター</a> の設定が簡素化されました。 <a href="#">AmazonPrometheusScrapingServiceRolePolicy</a> 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
<a href="#">AWS API を別の API リファレンスガイドに移動する</a>	Amazon Managed Service for Prometheus AWS APIs <a href="#">「Amazon Managed Service for Prometheus API</a>	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

[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



# AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「[AWS 用語集](#)」を参照してください。

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。