aws

開発者ガイド

Amazon OpenSearch サービス

Amazon OpenSearch サービス: 開発者ガイド

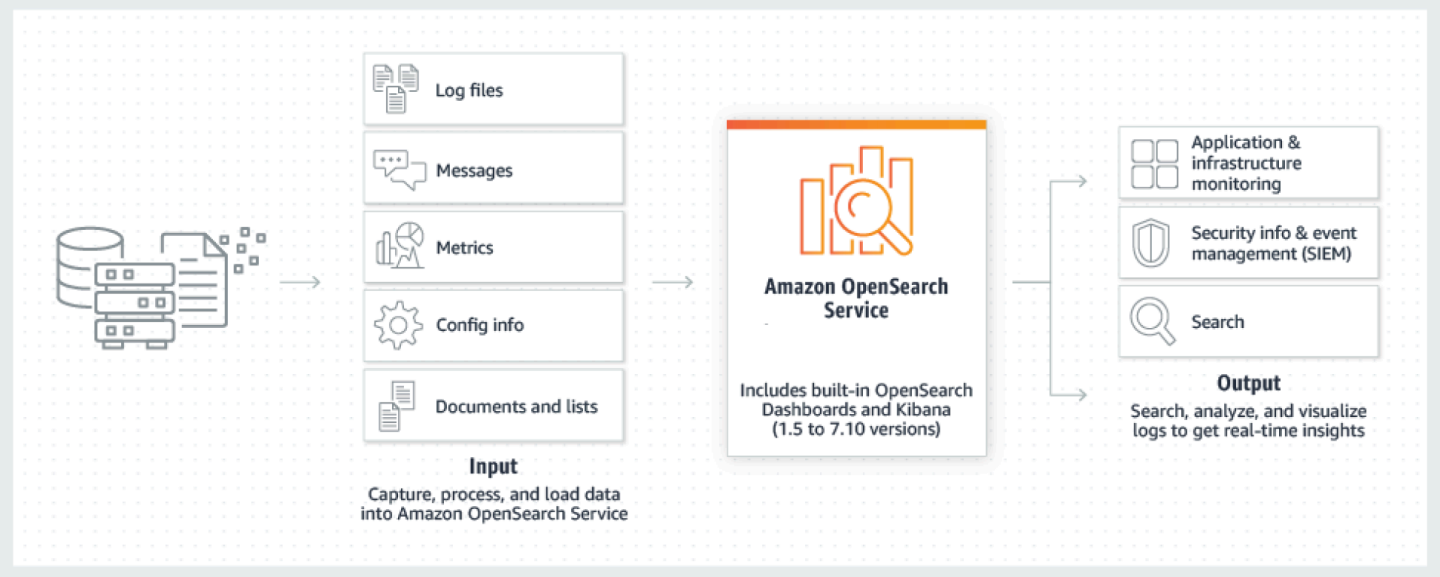
Table of Contents

Amazon OpenSearch Service とは

Amazon OpenSearch Service は、 AWS クラウドでの OpenSearch クラスターのデプロイ、運用、スケーリングを容易にするマネージドサービスです。Amazon OpenSearch Service は、OpenSearch およびレガシー Elasticsearch OSS (ソフトウェアの最終オープンソースバージョンである 7.10 まで) をサポートしています。クラスターを作成するときに、どの検索エンジンを使用するかのオプションがあります。

OpenSearch は、ログ分析、リアルタイムアプリケーションモニタリング、クリックストリーム分析などのユースケース向けの、完全にオープンソースの検索および分析エンジンです。詳細については、 OpenSearch ドキュメントを参照してください。

Amazon OpenSearch Service は、 OpenSearch クラスターのすべてのリソースをプロビジョニングし、クラスターを起動します。また、障害が発生した OpenSearch サービスノードを自動的に検出して置き換え、セルフマネージドインフラストラクチャに関連するオーバーヘッドを削減します。また、単一の API コールを使用するか、コンソールで数回クリックするだけで、クラスターを簡単にスケーリングできます。



OpenSearch サービスの使用を開始するには、クラスター に相当する OpenSearch サービスドメイン OpenSearch を作成します。クラスター内の各 EC2 インスタンスは、1 つの OpenSearch サービスノードとして機能します。

OpenSearch サービスコンソールを使用して、数分でドメインをセットアップおよび設定できます。プログラムによるアクセスを希望する場合は、、 AWS SDKsAWS CLI、または Terraform を使用できます。

Amazon OpenSearch Service の機能

OpenSearch サービスには以下の機能が含まれています。

[Scale] (スケール)

- 費用効率の高い Graviton インスタンスを含む、インスタンスタイプと呼ばれる、CPU、メモリ、ストレージ容量の多数の設定
- 最大 3 PB のアタッチ済みストレージ
- 読み取り専用データ用の費用対効果UltraWarmの高いコールドストレージ

セキュリティ

- AWS Identity and Access Management (IAM) アクセスコントロール
- Amazon VPC および VPC セキュリティグループと簡単に統合
- 保管中のデータの暗号化と node-to-node 暗号化
- OpenSearch Dashboards の Amazon Cognito 、HTTP 基本、または SAML 認証
- インデックスレベル、ドキュメントレベル、フィールドレベルのセキュリティ
- 監査ログ
- Dashboards マルチテナンシー

安定性

- リージョンおよびアベイラビリティーゾーンと呼ばれる、リソース用の複数の地理的場所
- マルチ AZ と呼ばれる、同じ AWS リージョン内の 2 つまたは 3 つのアベイラビリティーゾーンにまたがるノード割り当て
- クラスター管理タスクをオフロードする専用マスターノード
- OpenSearch サービスドメインをバックアップおよび復元するための自動スナップショット

柔軟性

- ビジネスインテリジェンス (BI) アプリケーションとの統合のための SQL サポート
- 検索結果を改善するためのカスタムパッケージ

人気のあるサービスとの統合

- OpenSearch Dashboards を使用したデータの視覚化
- OpenSearch サービスドメインメトリクスのモニタリングとアラームの設定 CloudWatch のための Amazon との統合
- OpenSearch サービスドメイン AWS CloudTrail への設定 API コールを監査するための との統合
- ストリーミングデータを OpenSearch サービスにロードするための Amazon S3、Amazon Kinesis 、および Amazon DynamoDB との統合
- データが特定のしきい値を超えたときの Amazon SNS からのアラート

と Amazon OpenSearch Service OpenSearch をいつ使用するか

次の表は、プロビジョニングされた Amazon OpenSearch Service とセルフマネージドのどちらが正しい選択 OpenSearch であるかを判断するのに役立ちます。

OpenSearch	Amazon OpenSearch サービス
• 組織には、自己プロビジョニングされたクラスターを手動でモニタリングして維持する適切なスキルを持つ人がいます。 • コードの完全なコンパイルレベルの制御が必要です。 • 組織はオープンソースソフトウェアを優先するか、独自に使用します。 • マルチクラウド戦略があり、ベンダー固有ではないテクノロジーが必要です。 • チームは、重要な本番稼働の問題に対処できます。 • 製品を自由に使用、変更、拡張できます。 • 新機能がリリースされたらすぐにアクセスできるようにする必要があります。	• インフラストラクチャを手動で管理、モニタリング、保守する必要はありません。 • Amazon S3 の耐久性と低コストを活用して、ストレージ階層にデータを階層化することで、増加する分析コストを管理する簡単な方法が必要です。 • DynamoDB 、Amazon DocumentDB (MongoDB 互換) 、IAM CloudWatch、 AWS のサービス などの他の との統合を利用したい CloudFormation。 • 予防メンテナンスや本番稼働中の問題 AWS Support が発生した場合は、 からのサポートに簡単にアクセスする必要があります。 • 自己修復、プロアクティブメンテナンス、回復力、バックアップなどの機能を活用したい。

Amazon OpenSearch サーバーレス

Amazon OpenSearch Serverless は、Amazon OpenSearch Service のオンデマンド、自動スケーリング、サーバーレス設定です。Serverless は、 OpenSearch クラスターのプロビジョニング、設定、チューニングの運用上の複雑さを排除します。詳細については、「Amazon OpenSearch Serverless」を参照してください。

Amazon OpenSearch Ingestion

Amazon Ingestion OpenSearch は、Data Prepper を搭載したフルマネージド型のデータコレクターで、Amazon OpenSearch Service ドメインと OpenSearch Serverless コレクションにリアルタイムのログとトレースデータを提供します。データのフィルタリング、強化、変換、正規化、集計を行い、下流の分析と可視化を可能にします。詳細については、「Amazon OpenSearch Ingestion」を参照してください。

OpenSearch および Elasticsearch でサポートされているバージョン

OpenSearch サービスは現在、次の OpenSearch バージョンをサポートしています。

- 2.13、2.11、2.9、2.7、2.5、2.3、1.3、1.2、1.1、1.0

OpenSearch サービスは、次のレガシー Elasticsearch OSS バージョンもサポートしています。

- 7.10、7.9、7.8、7.7、7.4、7.1
- 6.8、6.7、6.5、6.4、6.3、6.2、6.0
- 5.6、5.5、5.3、5.1
- 2.3
- 1.5

詳細については、the section called “サポートされているオペレーション”、the section called “エンジンバージョン別の機能”、およびthe section called “エンジンバージョンに応じたプラグイン”を参照してください。

新しい OpenSearch サービスプロジェクトを開始する場合は、サポートされている最新バージョンを選択することを強くお勧めします OpenSearch 。既存のドメインで以前の Elasticsearch バージョ

ンを使用している場合は、ドメインの維持またはデータの移行を選択できます。詳細については、「the section called “ドメインのアップグレード”」を参照してください。

Amazon OpenSearch Service の料金

OpenSearch サービスについては、EC2 インスタンスの使用時間ごと、およびインスタンスにアタッチされた EBS ストレージボリュームの累積サイズに対して料金を支払います。標準 AWS データ転送料金も適用されます。

ただし、いくつかの注意すべきデータ転送の例外があります。ドメインが複数のアベイラビリティーゾーン を使用している場合、 OpenSearch サービスはアベイラビリティーゾーン間のトラフィックに対して課金しません。シャードの割り当てと再調整中に、ドメイン内で重大なデータ転送が発生します。 OpenSearch サービスは、このトラフィックを計測も請求もしません。同様に、OpenSearch サービスは UltraWarm/cold ノードと Amazon S3 間のデータ転送には課金しません。

料金の詳細については、「Amazon OpenSearch Service の料金」を参照してください。設定変更中に発生する料金については、「the section called “設定変更に関連する料金”」を参照してください。

Amazon OpenSearch Service の開始方法

開始するには、AWS アカウント をまだお持ちでない場合はサインアップします。アカウントを設定したら、Amazon OpenSearch Service の開始方法チュートリアルを完了します。サービスについて学習中に詳しい情報が必要になった場合は、以下の概要トピックを参照してください。

- ドメインの作成
- ワークロードに適したドメインのサイジング
- ドメインアクセスポリシーまたはきめ細かなアクセスコントロールを使用したドメインへのアクセスの制御
- 手動または他の AWS のサービスからのデータのインデックス作成
- OpenSearch Dashboards を使用してデータを検索し、視覚化を作成する

セルフマネージド OpenSearch 型クラスターから OpenSearch サービスに移行する方法については、「」を参照してくださいthe section called “OpenSearch Service への移行”。

関連サービス

OpenSearch サービスは通常、以下のサービスで使用されます。

Amazon CloudWatch

OpenSearch サービスドメインは、ドメインのヘルスとパフォーマンスをモニタリング CloudWatch できるように、メトリクスを に自動的に送信します。詳細については、「Amazon による OpenSearch クラスターメトリクスのモニタリング CloudWatch」を参照してください。

CloudWatch ログは逆方向に進むこともできます。分析のためにデータを OpenSearch Service にストリーミングするように CloudWatch ログを設定できます。詳細については、「the section called “Amazon からストリーミングデータをロードする CloudWatch”」を参照してください。

AWS CloudTrail

を使用して AWS CloudTrail 、 アカウントの OpenSearch サービス設定 API コールおよび関連イベントの履歴を取得します。詳細については、「AWS CloudTrail での Amazon OpenSearch Service API 呼び出しのモニタリング」を参照してください。

Amazon Kinesis

Kinesis は、大規模なストリーミングデータをリアルタイムで処理するマネージドサービスです。詳細については、「the section called “Amazon Kinesis Data Streams からストリーミングデータをロードする”」および「the section called “Amazon Data Firehose からストリーミングデータをロードする”」を参照してください。

Amazon S3

Amazon Simple Storage Service (Amazon S3) は、インターネット用のストレージを提供します。このガイドでは、Amazon S3 と統合するための Lambda サンプルコードが提供されています。詳細については、「the section called “Amazon S3 からストリーミングデータをロードする”」を参照してください。

AWS IAM

AWS Identity and Access Management (IAM) は、サービスドメインへのアクセスを管理するために使用できるウェブ OpenSearch サービスです。詳細については、「the section called “Identity and Access Management”」を参照してください。

AWS Lambda

AWS Lambda は、サーバーのプロビジョニングや管理を行わずにコードを実行できるようにするコンピューティングサービスです。このガイドでは、DynamoDB、Amazon S3、および Kinesis からデータをストリーミングするための Lambda サンプルコードが提供されています。詳細につ

いては、「the section called “ストリーミングデータを OpenSearch サービスにロードする”」を参照してください。

Amazon DynamoDB

Amazon DynamoDB は、フルマネージド NoSQL データベースサービスであり、シームレスなスケーラビリティを備えた高速で予測可能なパフォーマンスを提供します。 OpenSearch サービスへのデータのストリーミングの詳細については、「」を参照してくださいthe section called “Amazon DynamoDB テーブルからストリーミングデータをロードする”。

Amazon QuickSight

Amazon QuickSight ダッシュボードを使用して、 OpenSearch サービスからデータを視覚化できます。詳細については、「Amazon ユーザーガイド」の「Amazon での Amazon OpenSearch サービス QuickSightの使用 QuickSight 」を参照してください。

> **Note**
>
> OpenSearch には、Elasticsearch B.V. からの特定の Apache ライセンス Elasticsearch コードとその他のソースコードが含まれています。Elasticsearch B.V. は、他のソースコードのソースではありません。ELASTICSEARCH は Elasticsearch B.V. の登録商標です。

Amazon OpenSearch サービスのセットアップ

トピック

にサインアップ AWS アカウント

をお持ちでない場合は AWS アカウント、次の手順を実行して作成してください。

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。セキュリティ上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、root ユーザーアクセスを必要とするタスクを実行するときは root ユーザーのみを使用してください。

AWS サインアッププロセスが完了すると、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。

管理者権限を持つユーザーを作成します。

にサインアップしたら AWS アカウント、日常のタスクに root ユーザーを使用しないように AWS IAM Identity Center、管理ユーザーを保護し、有効にしてから作成してください。 AWS アカウントのルートユーザー

セキュリティを確保してください。 AWS アカウントのルートユーザー

1. [Root user] を選択し、 AWS アカウント メールアドレスを入力して、AWS Management Consoleアカウント所有者としてにログインします。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 root ユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理者権限を持つユーザーを作成します。

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 Identity Center で、ユーザーに管理アクセスを付与します。

   IAM アイデンティティセンターディレクトリ をアイデンティティソースとして使用するチュートリアルについては、『ユーザーガイド』の「 IAM アイデンティティセンターディレクトリデフォルトでのユーザーアクセスの設定」を参照してください。AWS IAM Identity Center

管理者権限を持つユーザーとしてサインインします。

- 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ます。

1. IAM Identity Center で、最小権限権限を適用するというベストプラクティスに従った権限セットを作成します。

   手順については、『ユーザーガイド』の「権限セットの作成」を参照してください。AWS IAM Identity Center

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

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

許可を付与する

本番環境では、よりきめ細かなポリシーを使用することをお勧めします。アクセス管理の詳細については、IAM ユーザーガイドの「 AWS リソースのアクセス管理」を参照してください。

アクセス権限を付与するには、ユーザー、グループ、またはロールにアクセス許可を追加します。

- 以下のユーザーとグループ: AWS IAM Identity Center

  アクセス許可セットを作成します。「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

  ID フェデレーションのロールを作成します。詳細については、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールの作成」を参照してください。

- IAM ユーザー:
  - ユーザーが担当できるロールを作成します。手順については、「IAM ユーザーガイド」の「IAM ユーザー用ロールの作成」を参照してください。
  - (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加する。詳細については、「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス権限の追加」を参照してください。

プログラマチックアクセス権を付与する

AWS ユーザが外部とやりとりしたい場合、プログラムによるアクセスが必要です。 AWS Management Consoleプログラムによるアクセスを許可する方法は、アクセスするユーザーのタイプによって異なります。 AWS

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

プログラマチックアクセス権を必要とするユーザー	目的	方法
ワークフォースアイデンティティ (IAM Identity Center で管理されているユーザー)	一時的な認証情報を使用して、 AWS CLI、 AWS SDK、または API へのプログラムによるリクエストに署名します。 AWS	使用するインターフェイス用の手引きに従ってください。 • については AWS CLI、『AWS Command Line Interface ユーザガイド』の「AWS CLIAWS IAM Identity Centerを使用するように設定する」を参照してください。 • AWS SDK、ツール、 AWS API については、『AWS SDK およびツールリファレンスガイド』の「IAM ID センター認証」を参照してください。
IAM	一時的な認証情報を使用して、、 AWS SDK AWS CLI、または API へのプログラムによるリクエストに署名します。 AWS	IAM ユーザーガイドの「AWS リソースでの一時認証情報の使用」の指示に従います。
IAM	(非推奨) 長期認証情報を使用して、AWS CLI、 AWS SDK、また	使用するインターフェイス用の手引きに従ってください。

プログラマチックアクセス権を必要とするユーザー	目的	方法
	は API へのプログラムによるリクエストに署名します。AWS	• については AWS CLI、『ユーザーガイド』の「IAM ユーザー認証情報を使用した認証」を参照してください。AWS Command Line Interface • AWS SDK とツールについては、『AWS SDK およびツールリファレンスガイド』の「長期認証情報による認証」を参照してください。 • AWS API については、『IAM ユーザーガイド』の「IAM ユーザーのアクセスキーの管理」を参照してください。

をインストールして設定します。 AWS CLI

OpenSearch サービス API を使用する場合は、 AWS Command Line Interface (AWS CLI) の最新バージョンをインストールする必要があります。コンソールから OpenSearch Service AWS CLI を使用するためには必要ありません。で説明されている手順に従うと、CLI を使用しなくても開始できますAmazon OpenSearch Service の開始方法。

をセットアップするには AWS CLI

1. macOS、Linux、または Windows AWS CLI 用のの最新バージョンをインストールするには、「の最新バージョンのインストールまたは更新」を参照してください。 AWS CLI
2. AWS CLI OpenSearch サービスを含め、へのアクセスを安全に設定する方法については AWS のサービス、「aws configureクイック設定によるクイック設定」を参照してください。
3. セットアップを確認するには、 DataBrew コマンドプロンプトで次のコマンドを入力します。

```
aws opensearch help
```

AWS CLI パラメータまたはプロファイルで設定しない限り、 AWS リージョン コマンドは設定のデフォルトを使用します。 AWS リージョン パラメータを使用して設定するには、`--region`各コマンドにパラメータを追加します。

AWS リージョン プロファイルを使用して設定するには、`~/.aws/config%UserProfile%/.aws/config`まずファイルまたはファイル (Microsoft Windows の場合) に名前付きのプロファイルを追加します。「AWS CLI の名前付きプロファイル」のステップに従います。次に、以下の例のようなコマンドを使用して、ユーザー設定とその他の設定を行います。 AWS リージョン

```
[profile opensearch]
aws_access_key_id = ACCESS-KEY-ID-OF-IAM-USER
aws_secret_access_key = SECRET-ACCESS-KEY-ID-OF-IAM-USER
region = us-east-1
output = text
```

コンソールを開きます。

このセクションのコンソール関連のトピックのほとんどは、OpenSearch サービスコンソールから始まります。まだサインインしていない場合は AWS アカウント、OpenSearch ログインしてからサービスコンソールを開き、次のセクションに進んで Service の使用を続行してください。 OpenSearch

Amazon OpenSearch Service の開始方法

このチュートリアルでは、Amazon OpenSearch Service を使用してテストドメインを作成および設定する方法を示します。OpenSearch Service ドメインは、OpenSearch クラスターと同義です。ドメインは、指定した設定、インスタンスタイプ、インスタンスカウント、およびストレージリソースを持つクラスターです。

このチュートリアルでは、OpenSearch Service ドメインを準備してすぐに実行するための基本的な手順を説明します。詳細については、このガイドの「ドメインの作成と管理」とその他のトピックを参照してください。セルフマネージド OpenSearch クラスターから OpenSearch Service への移行の詳細については、「the section called “OpenSearch Service への移行”」を参照してください。

このチュートリアルのステップは、OpenSearch Service コンソール、AWS CLI、または AWS SDK を使用して完了できます。AWS CLI のインストールおよびセットアップの詳細については、「AWS Command Line Interface ユーザーガイド」を参照してください。

ステップ 1: Amazon OpenSearch Service ドメインを作成する

> **Important**
>
> これは、テスト Amazon OpenSearch Service ドメインを設定するための簡潔なチュートリアルです。本番稼働用ドメインを作成するには、このプロセスを使用しないでください。同じプロセスの包括的なバージョンについては、「ドメインの作成と管理」を参照してください。

OpenSearch Service ドメインは、OpenSearch クラスターと同義です。ドメインは、指定した設定、インスタンスタイプ、インスタンスカウント、およびストレージリソースを持つクラスターです。コンソール、AWS CLI、または AWS SDK を使用して OpenSearch Service ドメインを作成できます。

コンソールを使用して OpenSearch Service ドメインを作成するには

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。
2. [分析] の下で、[Amazon OpenSearch Service] を選択します。
3. [ドメインの作成] を選択します。

4. ドメインの名前を入力します。このチュートリアルの例では、movies という名前を使用します。
5. ドメインの作成方法は、[標準作成] を選択します。

> **Note**
>
> ベストプラクティスを使用して本番稼働用ドメインをすばやく設定するには、[簡易作成] を選択します。このチュートリアルの開発とテストには、[標準作成] を使用します。

6. テンプレートは、[開発/テスト] を選択します。
7. デプロイオプションは、[スタンバイが有効のドメイン] を選択します。
8. [Elasticsearch バージョン] で、最新バージョンを選択します。
9. 現時点では、[データノード]、[ウォームデータストレージとコールドデータストレージ]、[専用マスターノード]、[スナップショットの設定]、[カスタムエンドポイント] のセクションは無視してください。
10. このチュートリアルを簡単にするために、パブリックアクセスドメインを使用します。[ネットワーク] で [パブリックアクセス] を選択します。
11. きめ細かなアクセスコントロール設定で、[きめ細かなアクセスコントロールを有効化] チェックボックスをオンのままにします。[マスターユーザーの作成] をクリックして、ユーザー名とパスワードを入力します。
12. 今のところ、SAML 認証および Amazon Cognito 認証セクションは無視します。
13. [アクセスポリシー]で [きめ細かなアクセスコントロールのみを使用] を選択します。このチュートリアルでは、ドメインアクセスポリシーではなく、きめ細かなアクセスコントロールが認証を処理します。
14. 残りの設定は無視して、[作成] を選択します。通常、新しいドメインは初期化に 15 ~ 30 分かかりますが、設定によってはさらに時間がかかります。ドメインが初期化されたら、ドメインを選択して設定ペインを開きます。次の手順で使用する [一般情報] の下のドメインエンドポイント (例えば、`https://search-my-domain.us-east-1.es.amazonaws.com`) に注意してください。

次: インデックス作成のためにデータを OpenSearch Service ドメインにアップロードする

ステップ 2: インデックス作成のために Amazon OpenSearch Service にデータをアップロードする

> **Important**
>
> これは Amazon OpenSearch Service に少量のテストデータをアップロードするための簡潔なチュートリアルです。本番ドメインでのデータのアップロードの詳細については、「データのインデックス作成]を参照してください。

コマンドラインまたはほとんどのプログラミング言語を使用して、OpenSearch Service ドメインにデータをアップロードできます。

次の例のリクエストでは、簡潔にするため、および便宜上、一般的な HTTP クライアントである curl を使用しています。curl などのクライアントは、アクセスポリシーが IAM ユーザーあるいはロールを指定している場合に、必要なリクエスト署名を実行できません。このプロセスを正常に完了するには、ステップ 1 で設定したように、プライマリユーザー名とパスワードを使用してきめ細かなアクセスコントロールを使用する必要があります。

Windows に curl をインストールしてコマンドプロンプトから使用できますが、Cygwin または Windows Subsystem for Linux などのツールをお勧めします。macOS およびほとんどの Linux ディストリビューションでは、curl がプレインストールされています。

オプション 1: 単一のドキュメントをアップロードする

単一のドキュメントを movies ドメインに追加するには、次のコマンドを実行します。

```
curl -XPUT -u 'master-user:master-user-password' 'domain-endpoint/movies/_doc/1' -d
 '{"director": "Burton, Tim", "genre": ["Comedy","Sci-Fi"], "year": 1996, "actor":
 ["Jack Nicholson","Pierce Brosnan","Sarah Jessica Parker"], "title": "Mars Attacks!"}'
 -H 'Content-Type: application/json'
```

コマンドで、ステップ 1 で作成したユーザー名とパスワードを入力します。

このコマンドの詳細および OpenSearch Service への署名付きリクエストを作成する方法の詳細については、「データのインデックス作成」を参照してください。

オプション 2: 複数のドキュメントをアップロードする

複数のドキュメントを含む JSON ファイルを OpenSearch Service ドメインにアップロードするには

1. bulk_movies.json という名前のローカルファイルを作成します。以下のコンテンツをファイルに貼り付けます。末尾の改行も追加します。

```
{ "index" : { "_index": "movies", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller",
 "Crime"], "year": 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh,
 Janet", "Harvey, Laurence", "Silva, Henry", "Frees, Paul", "Gregory, James",
 "Bissell, Whit", "McGiver, John", "Parrish, Leslie", "Edwards, James", "Flowers,
 Bess", "Dhiegh, Khigh", "Payne, Julie", "Kleeb, Helen", "Gray, Joe", "Nalder,
 Reggie", "Stevens, Bert", "Masters, Michael", "Lowell, Tom"], "title": "The
 Manchurian Candidate"}
{ "index" : { "_index": "movies", "_id" : "3" } }
{"director": "Baird, Stuart", "genre": ["Action", "Crime", "Thriller"], "year":
 1998, "actor": ["Downey Jr., Robert", "Jones, Tommy Lee", "Snipes, Wesley",
 "Pantoliano, Joe", "Jacob, Ir\u00e8ne", "Nelligan, Kate", "Roebuck, Daniel",
 "Malahide, Patrick", "Richardson, LaTanya", "Wood, Tom", "Kosik, Thomas",
 "Stellate, Nick", "Minkoff, Robert", "Brown, Spitfire", "Foster, Reese",
 "Spielbauer, Bruce", "Mukherji, Kevin", "Cray, Ed", "Fordham, David", "Jett,
 Charlie"], "title": "U.S. Marshals"}
{ "index" : { "_index": "movies", "_id" : "4" } }
{"director": "Ray, Nicholas", "genre": ["Drama", "Romance"], "year": 1955, "actor":
 ["Hopper, Dennis", "Wood, Natalie", "Dean, James", "Mineo, Sal", "Backus, Jim",
 "Platt, Edward", "Ray, Nicholas", "Hopper, William", "Allen, Corey", "Birch,
 Paul", "Hudson, Rochelle", "Doran, Ann", "Hicks, Chuck", "Leigh, Nelson",
 "Williams, Robert", "Wessel, Dick", "Bryar, Paul", "Sessions, Almira", "McMahon,
 David", "Peters Jr., House"], "title": "Rebel Without a Cause"}
```

2. ファイルが保存されているローカルディレクトリで、次のコマンドを実行して、movies ドメインにそれをアップロードします。

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-
binary @bulk_movies.json -H 'Content-Type: application/json'
```

bulk ファイルの形式については、「データのインデックス作成」を参照してください。

次: ドキュメントの検索

ステップ 3: Amazon OpenSearch Service でドキュメントを検索する

Amazon OpenSearch Service ドメインでドキュメントを検索するには、OpenSearch 検索 API を使用します。または、OpenSearch Dashboards を使用してドメインのドキュメントを検索することもできます。

コマンドラインからドキュメントを検索する

movies ドメインで mars という単語を検索するには、次のコマンドを実行します。

```
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/movies/_search?
q=mars&pretty=true'
```

前のページでバルクデータを使用している場合は、代わりに rebel を検索してください。

次のようなレスポンスが表示されます。

```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.2876821,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.2876821,
        "_source" : {
          "director" : "Burton, Tim",
```

```
          "genre" : [
            "Comedy",
            "Sci-Fi"
          ],
          "year" : 1996,
          "actor" : [
            "Jack Nicholson",
            "Pierce Brosnan",
            "Sarah Jessica Parker"
          ],
          "title" : "Mars Attacks!"
        }
      }
    ]
  }
}
```

OpenSearch Dashboards を使用してドキュメントを検索する

OpenSearch Dashboards は OpenSearch で稼働するように設計された、ポピュラーなオープンソースの可視化ツールです。これは、インデックスを検索してモニタリングするための便利なユーザーインターフェイスを提供します。

Dashboards を使用して OpenSearch Service ドメインからドキュメントを検索するには

1. 使用しているドメインの OpenSearch Dashboards URL に移動します。この URL は、OpenSearch Service コンソールのドメインダッシュボードに表示されています。URL はこの形式に従います。

   ```
   domain-endpoint/_dashboards/
   ```

2. プライマリユーザー名とパスワードを使ってログインします。
3. Dashboards を使用するには、少なくとも 1 つのインデックスパターンを作成する必要があります。Dashboards は、そのパターンを使用して、どのインデックスを分析するかを特定します。左のナビゲーションパネルを開き、[スタック管理] 、[インデックスパターン] の順に選択してから、[インデックスパターンを作成する] を選択します。このチュートリアルでは、movies と入力します。
4. [次のステップ] を選択してから、[インデックスパターンの作成] を選択します。パターンが作成されたら、`actor` および `director` などのさまざまなドキュメントフィールドを表示できます。

5. [インデックスパターン] ページに戻り、movies がデフォルトとして設定されていることを確認します。そうでない場合は、パターンを選択し、星アイコンを選択してデフォルトにします。
6. データの検索を開始するには、左のナビゲーションパネルをもう一度開き、[検出] を選択します。
7. 検索バーに、単一のドキュメントをアップロードした場合は mars と入力するか、複数のドキュメントをアップロードした場合は rebel と入力してから [Enter] を押します。俳優名や監督名など、他の用語を検索してみることができます。

次: ドメインの削除

ステップ 4: Amazon OpenSearch Service ドメインを削除する

このチュートリアルの movies ドメインはテスト用のため、チュートリアルを終了したら、料金が発生しないようにこれを削除してください。

コンソールから OpenSearch Service ドメインを削除するには

1. [Amazon OpenSearch Service] コンソールにサインインします。
2. [ドメイン] で、[movies]ドメインを選択します。
3. [削除] を選択して、削除を確認します。

次のステップ

ドメインとインデックス作成データの作成方法がわかったので、次の演習のいくつかを行ってみてください。

- ドメインを作成するためのアドバンストオプションについて説明します。詳細については、「ドメインの作成と管理」を参照してください。
- ドメイン内のインデックスを管理する方法を見つけてください。詳細については、「インデックス管理」を参照してください。
- Amazon OpenSearch Service で作業するためのチュートリアルの 1 つを試してください。詳細については、「チュートリアル」を参照してください。

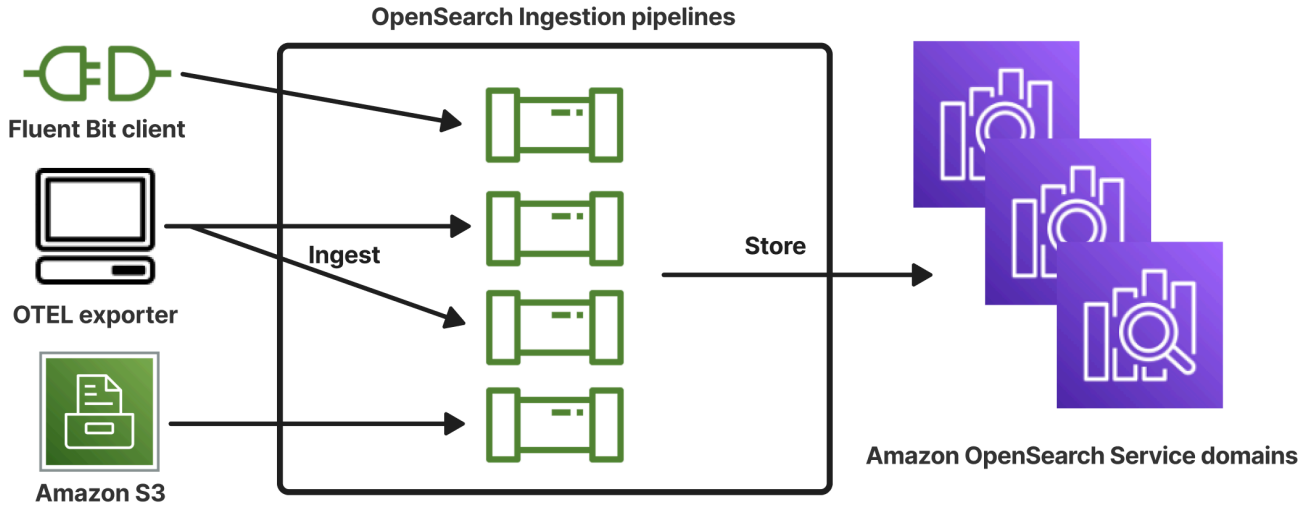
Amazon OpenSearch Ingestion

Amazon Ingestion OpenSearch は、Amazon OpenSearch Service ドメインと Serverless コレクションにリアルタイムのログ、メトリクス、トレースデータを配信する、フルマネージドの OpenSearch サーバーレスデータコレクターです。

OpenSearch Ingestion を使用すると、Logstash や Jaeger などのサードパーティーソリューションを使用して、 OpenSearch サービスドメインや OpenSearch サーバーレスコレクションにデータを取り込む必要がなくなります。データを Ingestion OpenSearch に送信するようにデータプロデューサーを設定します。その後、指定したドメインまたはコレクションにデータが自動的に配信されます。データを配信する前にデータを変換するように Ingestion OpenSearch を設定することもできます。

また、Ingestion OpenSearch では、サーバーのプロビジョニング、ソフトウェアの管理とパッチ適用、サーバーのクラスターのスケーリングについて心配する必要はありません。内で取り込みパイプラインを直接プロビジョニングすると AWS Management Console、Ingestion OpenSearch がそれらを管理およびスケーリングします。

OpenSearch 取り込みは Amazon OpenSearch Service のサブセットです。オープンソースのデータコレクターである Data Prepper を使用しており、データのフィルタリング、エンリッチ化、変換、正規化、集約を行って下流の分析や視覚化を可能にします。



トピック

- 主要なコンセプト
- OpenSearch 取り込みの利点
- 制限事項

- サポートされている Data Prepper のバージョン
- パイプラインのスケーリング
- OpenSearch 取り込み料金
- サポート対象 AWS リージョン
- OpenSearch 取り込みクォータ
- Amazon OpenSearch Ingestion のロールとユーザーの設定
- Amazon OpenSearch Ingestion の開始方法
- Amazon OpenSearch インジェストのパイプライン機能の概要
- Amazon Ingestion OpenSearch パイプラインの作成
- Amazon OpenSearch Ingestion パイプラインの表示
- Amazon OpenSearch インジェストパイプラインの更新
- Amazon OpenSearch Ingestion パイプラインの停止と開始
- Amazon OpenSearch Ingestion パイプラインの削除
- Amazon Ingestion OpenSearch パイプラインでサポートされているプラグインとオプション
- Amazon Ingestion OpenSearch パイプライン統合の使用
- Amazon OpenSearch Ingestion を使用してドメインとコレクション間でデータを移行する
- Amazon OpenSearch Ingestion を操作するための AWS SDKの使用
- Amazon OpenSearch Ingestion のセキュリティ
- Amazon OpenSearch Ingestion パイプラインのタグ付け
- Amazon CloudWatch を使用した Amazon OpenSearch Ingestion の口グ記録とモニタリング
- Amazon OpenSearch Ingestion のベストプラクティス

主要なコンセプト

OpenSearch 取り込みを開始すると、次の概念を理解しておくとメリットが得られます。

パイプライン

OpenSearch 取り込みの観点から見ると、パイプラインとは、 OpenSearch サービス内で作成する単一のプロビジョニングされたデータコレクターを指します。これは、1 つ以上のサブパイプラインを含む YAML 設定ファイル全体と考えることができます。取り込みパイプラインを作成する手順については、「the section called “パイプラインの作成”」を参照してください。

サブパイプライン

サブパイプラインは、YAML 設定ファイル内で定義します。各サブパイプラインは、ソース、バッファ、0 個以上のプロセッサ、1 個以上のシンクの組み合わせです。1 つの YAML ファイルで複数のサブパイプラインを定義することができます。各サブパイプラインには固有のソース、プロセッサ、シンクがあります。 CloudWatch およびその他の サービスによるモニタリングを支援するために、すべてのサブパイプラインとは異なるパイプライン名を指定することをお勧めします。

1 つ目のサブパイプラインのソースが 2 つ目のサブパイプラインで、シンクが 3 つ目のサブパイプラインとなるように、複数のサブパイプラインを 1 つの YAML ファイル内に紐づけることができます。例については、the section called “OpenTelemetry コレクター”を参照してください。

ソース

サブパイプラインの入力コンポーネントです。パイプラインがレコードを使用するメカニズムを定義します。ソースは、HTTPS 経由でイベントを受信するか、Amazon S3 などの外部エンドポイントから読み取ることでイベントを使用できます。ソースには、プッシュ型とプル型の 2 種類があります。HTTP や OTel logs などのプッシュ型のソースは、レコードを取り込みエンドポイントにストリーミングします。OTel trace や S3 などのプル型のソースは、ソースからデータを取得します。

Processors

レコードをシンクに発行する前に、レコードを目的の形式にフィルタリング、変換、エンリッチ化できる中間処理ユニットです。プロセッサはパイプラインのオプションコンポーネントです。プロセッサを定義しない場合、レコードはソースで定義されている形式で発行されます。複数のプロセッサを使用することができます。パイプラインは、定義した順序でプロセッサを実行します。

シンク

サブパイプラインの出力コンポーネントです。サブパイプラインがレコードを発行する 1 つ以上の送信先を定義します。Ingestion OpenSearch は OpenSearch 、サービスドメインをシンクとしてサポートします。また、サブパイプラインをシンクとしてサポートします。つまり、1 つの Ingestion Pipelines (YAML ファイル) OpenSearch 内で複数のサブパイプラインを文字列化できます。セルフマネージド型 OpenSearch クラスターはシンクとしてサポートされていません。

バッファ

プロセッサの一部で、ソースとシンクの間のレイヤーとして機能します。パイプライン内のバッファを手動で設定することはできません。 OpenSearch 取り込みでは、デフォルトのバッファ設定が使用されます。

ルート

プロセッサの一部で、パイプラインの作成者が、特定の条件に一致するイベントのみを異なるシンクに送信できるようにします。

有効なサブパイプラインの定義には、ソースとシンクが含まれている必要があります。こうしたパイプラインの各要素の詳細については、「設定リファレンス」を参照してください。

OpenSearch 取り込みの利点

OpenSearch 取り込みには主に次の利点があります。

- セルフプロビジョニングされたパイプラインを手動で管理する必要がなくなります。
- 定義したキャパシティ制限に基づいてパイプラインを自動的にスケーリングします。
- セキュリティパッチとバグに対するパッチで、パイプラインを最新の状態に維持します。
- オプションで、パイプラインを仮想プライベートクラウド (VPC) に接続し、セキュリティレイヤーを追加できます。
- コストを抑えるために、パイプラインの停止および開始が可能です。
- 一般的なユースケース用のパイプライン設定のブループリントを提供しているため、すみやかに起動して実行できます。
- さまざまな AWS SDKs と Ingestion API OpenSearch を使用して、プログラムでパイプラインとやり取りできます。
- Amazon でのパフォーマンスモニタリング CloudWatch と CloudWatch Logs でのエラーログ記録をサポートします。

制限事項

OpenSearch 取り込みには以下の制限があります。

- OpenSearch 1.0 以降、または Elasticsearch 6.8 以降を実行しているドメインにのみデータを取り込むことができます。OTel トレースソースを使用している場合は、OpenSearch Dashboards プラグイン を使用できるように、Elasticsearch 7.9 以降を使用することをお勧めします。
- パイプラインが VPC 内の OpenSearch サービスドメインに書き込む場合、パイプラインはドメイン AWS リージョン と同じ に作成する必要があります。
- パイプラインの定義内で設定できるデータソースは 1 つのみです。
- セルフマネージド型 OpenSearch クラスターをシンクとして指定することはできません。
- カスタムエンドポイントをシンクとして指定することはできません。カスタムエンドポイントが有効になっているドメインへの書き込みは可能ですが、標準エンドポイントを指定する必要があります。
- オプトインリージョン内のリソースをソースまたはシンクとして指定することはできません。
- パイプライン設定に含めることができるパラメータにはいくつかの制約があります。詳細については、「the section called “設定の要件と制限”」を参照してください。

サポートされている Data Prepper のバージョン

OpenSearch 取り込みは現在、Data Prepper の次のメジャーバージョンをサポートしています。

- 2.x

パイプラインを作成するときは、必要な `version` オプションを用いて、使用する Data Prepper のメジャーバージョンを指定します。例えば、`.version: "2"` OpenSearch Ingestion は、そのメジャーバージョンのサポートされている最新のマイナーバージョンを取得し、そのバージョンでパイプラインをプロビジョニングします。詳細については、「the section called “パイプラインのバージョンの指定”」を参照してください。

現在、 OpenSearch 取り込みパイプラインは Data Prepper のバージョン 2.7 でプロビジョニングされています。詳細については、「 2.7 リリースノート」を参照してください。Data Prepper の各バージョンに含まれる機能とバグ修正については、「リリース」ページを参照してください。特定のメジャーバージョンのすべてのマイナーバージョンが Ingestion OpenSearch でサポートされているわけではありません。

パイプラインの YAML 設定ファイルを更新するときに、Data Prepper の新しいマイナーバージョンがサポートされている場合、 OpenSearch Ingestion はパイプライン設定で指定された最新のサポートされているマイナーバージョンのメジャーバージョンにパイプラインを自動的にアップグレードします。例えば、パイプライン設定`version: "2"`に があり、Ingestion OpenSearch が最初にパイ

プラインをバージョン 2.6.0 でプロビジョニングしたとします。バージョン 2.7.0 のサポートが追加され、パイプライン設定を変更すると、Ingestion OpenSearch はパイプラインをバージョン 2.7.0 にアップグレードします。このプロセスにより、パイプラインは、最新のバグ修正とパフォーマンスの改善が施されて最新状態に保たれます。 OpenSearch パイプライン設定内の`version`オプションを手動で変更しない限り、取り込みはパイプラインのメジャーバージョンを更新できません。詳細については、「the section called “パイプラインの更新”」を参照してください。

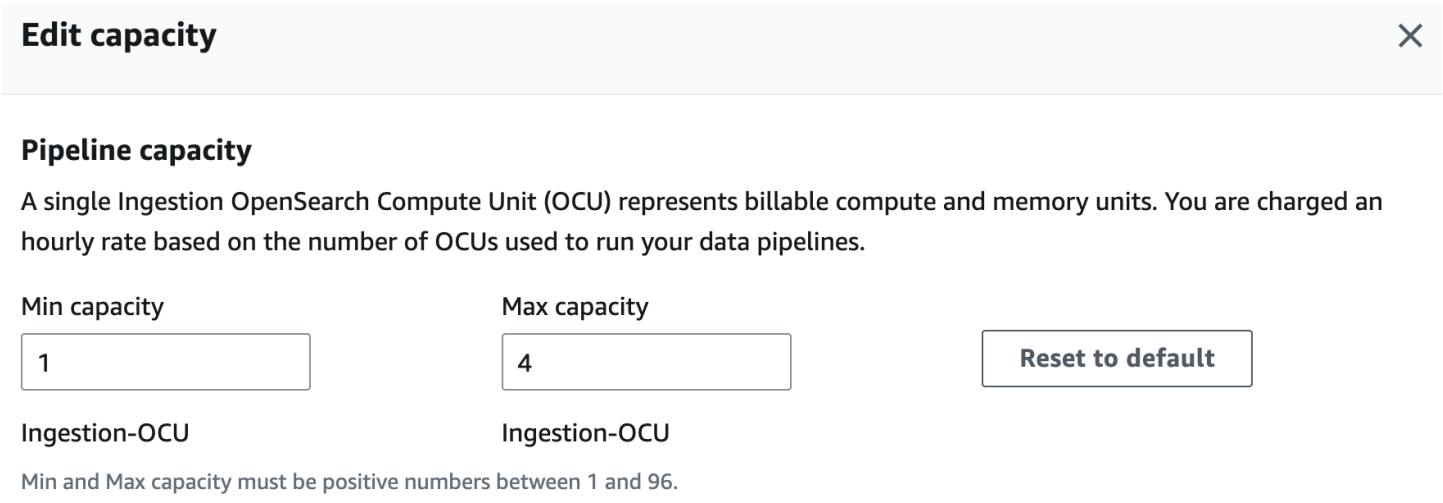
パイプラインのスケーリング

パイプライン容量を自分でプロビジョニングして管理する必要はありません。Ingestion OpenSearch は、指定した最小および最大の Ingestion OpenSearch Compute Units (Ingestion OCUs) に基づいて、推定ワークロードに応じてパイプライン容量を自動的にスケーリングします。

各 Ingestion OCU は、約 8 GiB のメモリと 2 個の vCPUs の組み合わせです。パイプラインの最小 OCU 値と最大 OCU 値を指定できます。Ingestion OpenSearch は、これらの制限に基づいてパイプライン容量を自動的にスケーリングします。

次の値を指定できます。

- 最小キャパシティ - パイプラインは、この数値の Ingestion OCU までキャパシティを減らすことができます。指定された最小キャパシティは、パイプラインの開始キャパシティでもあります。
- 最大キャパシティ - パイプラインは、この数値の Ingestion OCU までキャパシティを増やすことができます。



パイプラインの最大キャパシティがワークロードの急増に対処できる程度に十分な大きさであること、およびパイプラインがビジー状態でないときに、最小キャパシティがコストを最小限に抑える

のに十分な程度に低い値であるようにします。設定に基づいて、 OpenSearch Ingestion はパイプラインの Ingestion OCUs の数を自動的にスケーリングして、取り込みワークロードを処理します。いつの時点においても、課金されるのはパイプラインで実際に使用している Ingestion OCU 分のみです。

OpenSearch 取り込みパイプラインに割り当てられた容量は、パイプラインの処理要件とクライアントアプリケーションによって生成された負荷に基づいてスケールアップおよびスケールダウンします。容量が制約されている場合、Ingestion OpenSearch はより多くのコンピューティングユニット (GiB のメモリ) を割り当てることでスケールアップします。パイプラインが処理するワークロードが小さい場合や、データをまったく処理しない場合は、設定された最小限の Ingestion OCU までスケールダウンできます。

指定できる Ingestion OCU は最小 1 個で、最大ではステートレスパイプラインで 96 個、ステートフルパイプラインで 48 個です。プッシュ型のソースの場合、最小の Ingestion OCU は少なくとも 2 個をお勧めします。永続的バッファリングが有効な場合、最小 2 個および最大 384 個の取り込み OCU を指定できます。

単一のソース、シンプルな Grok パターン、シンクを持つ標準のログパイプラインを前提とする場合、各コンピューティングユニットは 1 秒あたり最大 2 MiB をサポートできます。複数のプロセッサを持つ複雑なログパイプラインの場合、各コンピューティングユニットがサポートする取り込み負荷は少なくなります。パイプラインの容量とリソース使用率に基づいて、 OpenSearch 取り込みスケーリングプロセスが開始されます。

高可用性を確保するために、Ingestion OCU はアベイラビリティーゾーン (AZ) に分散されます。AZ の数は、指定した最小キャパシティに依存します。

例えば、最低 2 つのコンピューティングユニットを指定した場合、任意の時点で使用される Ingestion OCU は 2 つの AZ に均等に分散されます。コンピューティングユニットを 3 つ以上指定すると、Ingestion OCU は 3 つの AZ に均等に分散されます。取り込みパイプラインの可用性が 99.9% になるように、少なくとも 2 つの Ingestion OCU をプロビジョニングすることをお勧めします。

パイプラインのステータスが `Create failed`、`Creating`、`Deleting`、`Stopped` の場合は、Ingestion OCU の料金は請求されません。

パイプラインのキャパシティ設定を行って取得する手順については、「the section called “パイプラインの作成”」を参照してください。

OpenSearch 取り込み料金

パイプラインを流れるデータがあるかどうかにかかわらず、特定の時点で、パイプラインに割り当てられた Ingestion OCU の数に基づく料金のみをお支払いいただきます。 OpenSearch 取り込みは、使用量に基づいてパイプライン容量をスケールアップまたはスケールダウンすることで、ワークロードに即座に対応します。

料金の詳細については、「Amazon OpenSearch Service の料金」を参照してください。

サポート対象 AWS リージョン

OpenSearch 取り込みは、 OpenSearch サービス AWS リージョン が利用可能な のサブセットで使用できます。サポートされているリージョンのリストについては、「」の「Amazon OpenSearch Service エンドポイントとクォータ」を参照してくださいAWS 全般のリファレンス。

OpenSearch 取り込みクォータ

取り込みリソースのデフォルトクォータのリストについては、「Amazon Service OpenSearch Quotas」を参照してください。 OpenSearch

Amazon OpenSearch Ingestion のロールとユーザーの設定

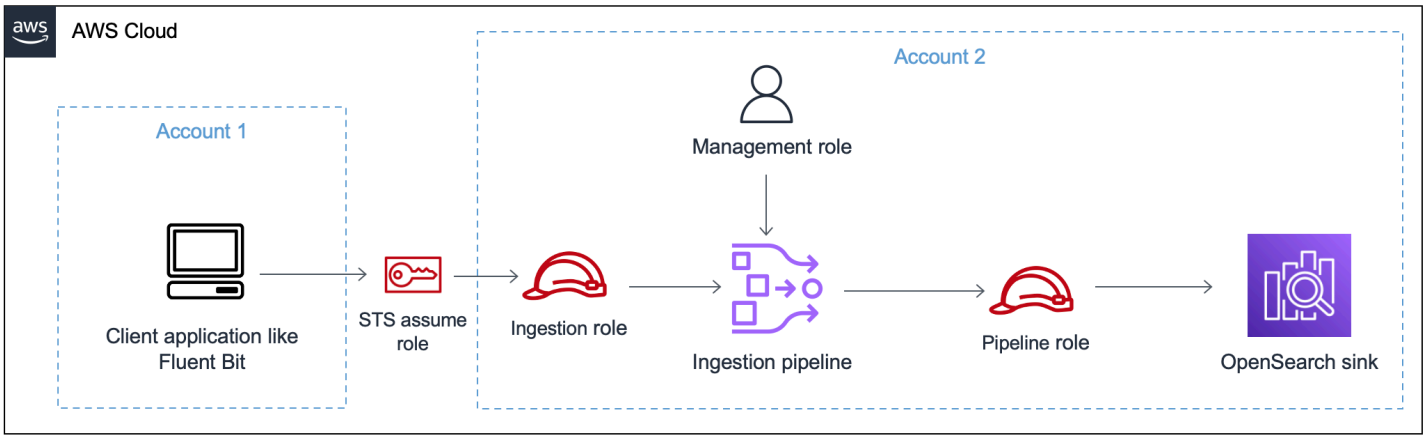
Amazon OpenSearch Ingestion では、ソースアプリケーションからパイプラインへの書き込みと、パイプラインからシンクへの書き込みを行えるように、さまざまなアクセス許可モデルと IAM ロールを使用しています。データの取り込みを開始する前に、特定のアクセス許可を持つ IAM ロールをユースケースに基づいて 1 つ以上作成する必要があります。

パイプラインを正常に設定するには、少なくとも次のロールが必要です。

名前	説明
管理ロール	パイプラインを管理するプリンシパル (一般的には「パイプライン管理者」) には、`osis:CreatePipeline` や `osis:UpdatePipeline` などのアクセス許可を含む管理アクセスが必要です。これらのアクセス許可により、ユーザーはパイプラインを管理できますが、必ずしもパイプラインにデータを書き込むことができるわけではありません。

名前	説明
パイプラインロール	パイプラインの YAML 設定内で指定するパイプラインロールは、パイプラインからドメインまたはコレクションシンクに書き込んだり、プル型のソースから読み込んだりするために必要なアクセス許可を付与します。詳細については、次のトピックを参照してください。 • the section called “パイプラインにドメインへのアクセス権を付与する” • the section called “パイプラインにコレクションへのアクセスを許可する”
取り込みロール	取り込みロールには、パイプラインリソースの `osis:Ingest` 許可が含まれています。これにより、プッシュ型のソースはパイプラインにデータを取り込むことができます。

次の図は、Amazon S3 や Fluent Bit などのデータソースから別のアカウントのパイプラインに書き込むときの一般的なパイプライン設定を示しています。この場合、パイプラインにアクセスするには、クライアントが取り込みロールを引き受けている必要があります。詳細については、「the section called “クロスアカウント取り込み”」を参照してください。



簡単な設定ガイドについては、「the section called “チュートリアル: ドメインにデータを取り込む”」を参照してください。

トピック

- the section called “管理ロール”
- the section called “取り込みロール”
- the section called “パイプラインロール”

- the section called “クロスアカウント取り込み”

管理ロール

パイプラインの作成と変更に必要な基本的な `osis:*` 許可に加えて、パイプラインロールリソース用の `iam:PassRole` 許可も必要です。ロールを受け入れる AWS のサービス は、必ずこの許可を使用します。OpenSearch Ingestion は、シンクにデータを書き込む必要があるたびに、このロールを引き受けます。これにより、承認済みのユーザーのみが、アクセス許可を付与するロールを使用して OpenSearch Ingestion を設定できるようになります。詳細については、「AWS のサービス のサービスにロールを渡すアクセス権限をユーザーに付与する」を参照してください。

AWS Management Consoleを使用している場合 (ブループリントの使用とパイプラインのチェック)、パイプラインを作成および更新するには次のアクセス許可が必要です。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:GetPipelineBlueprint",
            "osis:ListPipelineBlueprints",
            "osis:GetPipeline",
            "osis:ListPipelines",
            "osis:GetPipelineChangeProgress",
            "osis:ValidatePipeline",
            "osis:UpdatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/pipeline-role"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
```

```
}
```

AWS CLIを使用している場合 (ブループリントの事前検証とブループリントの使用以外)、パイプラインを作成および更新するには次のアクセス許可が必要です。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:UpdatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/pipeline-role"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

パイプラインロール

パイプラインからシンクに書き込むには、特定のアクセス許可が必要です。これらのアクセス許可は、シンクが OpenSearch Service ドメインか OpenSearch Serverless コレクションかによって異なります。

さらに、パイプラインには、ソースアプリケーションからプルするための許可 (ソースがプルベースのプラグインの場合)、および S3 デッドレターキュー (設定されている場合) に書き込むための許可が必要になる場合があります。

トピック

- ドメインシンクへの書き込み
- コレクションシンクへの書き込み

- デッドレターキューへの書き込み

ドメインシンクへの書き込み

OpenSearch Ingestion パイプラインには、シンクとして設定されている OpenSearch Service ドメインに書き込むためのアクセス許可が必要です。これらのアクセス許可には、ドメインを記述して、そこに HTTP リクエストを送信できることが含まれます。

シンクへの書き込みに必要なアクセス許可をパイプラインに付与するには、まず必要なアクセス許可を持つAWS Identity and Access Management (IAM) ロールを作成します。これらのアクセス許可は、パブリックパイプラインと VPC パイプラインのどちらも同じです。次に、ドメインがパイプラインからの書き込みリクエストを受け入れることができるように、ドメインアクセスポリシーでパイプラインロールを指定します。

最後に、パイプライン設定の sts_role_arn オプションの値にロール ARN を指定します。

```
version: "2"
source:
  http:
    ...
processor:
  ...
sink:
  - opensearch:
      ...
      aws:
        sts_role_arn: arn:aws:iam::{your-account-id}:role/pipeline-role
```

これらの各ステップを完了する手順については、「パイプラインにドメインへのアクセスを許可する」を参照してください。

コレクションシンクへの書き込み

シンクとして設定されている OpenSearch Serverless コレクションに書き込みを行うには、OpenSearch Ingestion パイプラインにアクセス許可を付与する必要があります。これらのアクセス権限には、コレクションを記述しそこに HTTP リクエストを送信できることが含まれます。

まず、すべてのリソース (*) に対する aoss:BatchGetCollection アクセス権限を持つ、IAM ロールを作成します。次に、このロールをデータアクセスポリシーに追加し、インデックスの作成、インデックスの更新、インデックスの記述、コレクション内でのドキュメントの書き込みを行うた

めのアクセス権限をそれに付与します。最後に、パイプライン設定の sts_role_arn オプションの値にロール ARN を指定します。

これらの各ステップを完了する手順については、「パイプラインにコレクションへのアクセスを許可する」を参照してください。

デッドレターキューへの書き込み

デッドレターキュー (DLQ) に書き込むようにパイプラインを設定する場合は、DLQ 設定内に sts_role_arn オプションを含める必要があります。このロールに含まれる許可により、DLQ イベントの宛先として指定した S3 バケットにパイプラインがアクセスできるようになります。

すべてのパイプラインコンポーネントで同じ sts_role_arn を使用する必要があります。したがって、DLQ アクセスを提供するパイプラインロールに別の許可ポリシーをアタッチする必要があります。少なくとも、ロールにはバケットリソースに対する S3:PutObject アクションが許可されている必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteToS3DLQ",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-dlq-bucket/*"
    }
  ]
}
```

その後、パイプラインの DLQ 設定内でロールを指定できます。

```
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/pipeline-role"
```

取り込みロール

OpenSearch Ingestion が現在サポートしているすべてのソースプラグイン (S3 を除く) は、プッシュ型のアーキテクチャを使用しています。つまり、パイプラインがソースからデータを取得するのではなく、ソースアプリケーションがデータをパイプラインにプッシュします。

そのため、OpenSearch Ingestion パイプラインにデータを取り込むために必要なアクセス許可を、ソースアプリケーションに付与する必要があります。少なくとも、リクエストに署名するロールには、パイプラインにデータを送信できるようにする `osis:Ingest` アクションの許可を付与する必要があります。パブリックパイプラインと VPC パイプラインのエンドポイントにも、同じアクセス許可が必要です。

次のサンプルポリシーでは、関連するプリンシパルが `my-pipeline` という名前の 1 つのパイプラインにデータを取り込むことを許可しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermitsWriteAccessToPipeline",
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-west-2:{your-account-id}:pipeline/my-pipeline"
    }
  ]
}
```

詳細については、「the section called “パイプライン統合を操作する”」を参照してください。

クロスアカウント取り込み

アプリケーションアカウントなど、異なる AWS アカウント からデータをパイプラインに取り込む必要がある場合があります。クロスアカウント取り込みを設定するには、パイプラインと同じアカウント内で取り込みロールを定義し、その取り込みロールとアプリケーションアカウント間に信頼関係を確立します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
```

```
        "AWS": "arn:aws:iam::{external-account-id}:root"
      },
      "Action": "sts:AssumeRole"
  }]
}
```

次に、取り込みロールを引き受けるようにアプリケーションを設定します。アプリケーションアカウントは、パイプラインアカウントの取り込みロールに対する AssumeRole 許可を、アプリケーションロールに付与する必要があります。

詳細な手順と IAM ポリシーの例については、「the section called “クロスアカウント取り込みアクセスの提供”」を参照してください。

Amazon Ingestion OpenSearch パイプラインにドメインへのアクセスを許可する

Amazon OpenSearch Ingestion パイプラインには、シンクとして設定された OpenSearch サービスドメインに書き込むためのアクセス許可が必要です。アクセスを提供するには、パイプラインがデータを送信しているドメインへのアクセスを制限する制限付きアクセス許可ポリシーで AWS Identity and Access Management (IAM) ロールを設定します。例えば、このユースケースをサポートするために必要なドメインとインデックスのみに、取り込みパイプラインを制限するとします。

パイプラインの設定でロールを指定する前に、適切な信頼関係を使ってこれを設定し、ドメインのアクセスポリシーでこのドメインへのアクセス許可を付与します。

トピック

- ステップ 1: パイプラインロールを作成する
- ステップ 2: パイプラインのロールをドメインアクセスポリシーに追加する
- ステップ 3: パイプラインロールをマッピングする (きめ細かいアクセスコントロールを使用するドメインについてのみ)
- ステップ 4: パイプライン設定でロールを指定する

ステップ 1: パイプラインロールを作成する

パイプライン設定の sts_role_arn パラメータで指定するロールには、ドメインシンクへのデータ送信を許可する、アクセス許可ポリシーが添付されている必要があります。また、Ingestion OpenSearch がロールを引き受けることを許可する信頼関係も必要です。ポリシーをロールにアタッチする方法については、「IAM ユーザーガイド」の「IAM ID アクセス許可の追加」を参照してください。

次のポリシー例では、パイプライン設定の sts_role_arn ロールに、単一ドメインへの書き込みを可能にする最小特権が付与されています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "es:DescribeDomain",
            "Resource": "arn:aws:es:*:{your-account-id}:domain/*"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:*:{your-account-id}:domain/{domain-namedomain}/*"
        }
    ]
}
```

ロールを再利用して複数のドメインに書き込む場合は、ドメイン名をワイルドカード文字 (*) に置き換えることでポリシーの範囲を広げることができます。

ロールには、次の信頼関係 が必要です。これにより、Ingestion OpenSearch はパイプラインロールを引き受けることができます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"osis-pipelines.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

さらに、いわゆる混乱した使節の問題からご自身を守るために、aws:SourceAccount と aws:SourceArn の条件キーをポリシーに追加することが推奨されています。送信元アカウントは、このパイプラインの所有者です。

例えば、次の条件ブロックをポリシーに追加できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "{your-account-id}"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:osis:{region}:{your-account-id}:pipeline/*"
    }
}
```

ステップ 2: パイプラインのロールをドメインアクセスポリシーに追加する

パイプラインがドメインにデータを書き込むには、sts_role_arn パイプラインロールにドメインへのアクセスを許可するドメインレベルのアクセスポリシーが、このドメインに必要になります。

次のドメインアクセスポリシーの例では、前のステップで作成した pipeline-role と言う名前のパイプラインロールに、ingestion-domain と言う名前のドメインへの、データの書き込みが許可されています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

ステップ 3: パイプラインロールをマッピングする (きめ細かいアクセスコントロールを使用するドメインについてのみ)

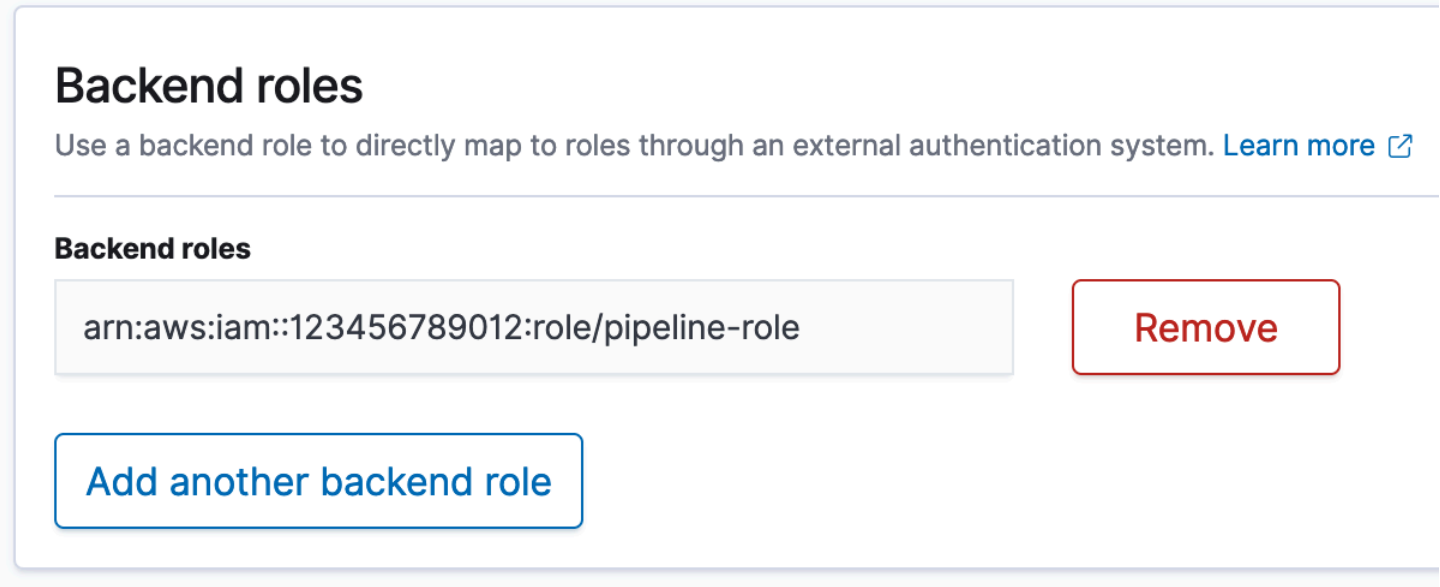
ドメインが、認証用に詳細なアクセス制御を使用している場合は、パイプラインにドメインへのアクセスを許可するには、追加の手順が必要になります。この手順は、ドメインの設定によって異なります。

シナリオ 1: マスターロールとパイプラインロールが異なる – IAM Amazon リソースネーム (ARN) をマスターユーザーとして使用していて、パイプラインロール (`sts_role_arn`) とは異なる場合は、パイプラインロールを`all_access`バックエンドロールに OpenSearchマッピングする必要があります。これにより、パイプラインロールが追加のマスターユーザーとして、実質的に追加されます。詳細については、「追加のマスターユーザー」を参照してください。

シナリオ 2: 内部ユーザーデータベースのマスターユーザー – ドメインが内部ユーザーデータベースのマスターユーザーと OpenSearch Dashboards の HTTP 基本認証を使用している場合、マスターユーザー名とパスワードをパイプライン設定に直接渡すことはできません。代わりに、パイプラインロール (`sts_role_arn`) を`all_access`バックエンドロールに OpenSearchマッピングする必要があります。これにより、パイプラインロールが追加のマスターユーザーとして、実質的に追加されます。詳細については、「追加のマスターユーザー」を参照してください。

シナリオ 3: マスターロールとパイプラインロールが同じ (まれなシナリオ) — IAM ARN をマスターユーザーとして使用し、パイプラインロール (`sts_role_arn`) として使用している ARN がこれと同じである場合、追加の操作は必要ありません。このパイプラインは、ドメインへの書き込みに必要なアクセス許可がすでにあります。ほとんどの環境では、管理者ロールまたは他のロールをマスターロールとして使用するため、これはまれなシナリオです。

次の図は、パイプラインのロールをバックエンドロールにマッピングする方法を示したものです。



ステップ 4: パイプライン設定でロールを指定する

パイプラインを作成するには、ステップ 1 で作成したパイプラインロールを、パイプライン設定の sts_role_arn パラメータとして指定する必要があります。パイプラインは、 OpenSearch サービスドメインシンクへのリクエストに署名するために、このロールを引き受けます。

sts_role_arn フィールドで、IAM パイプラインのロールの ARN を指定します。

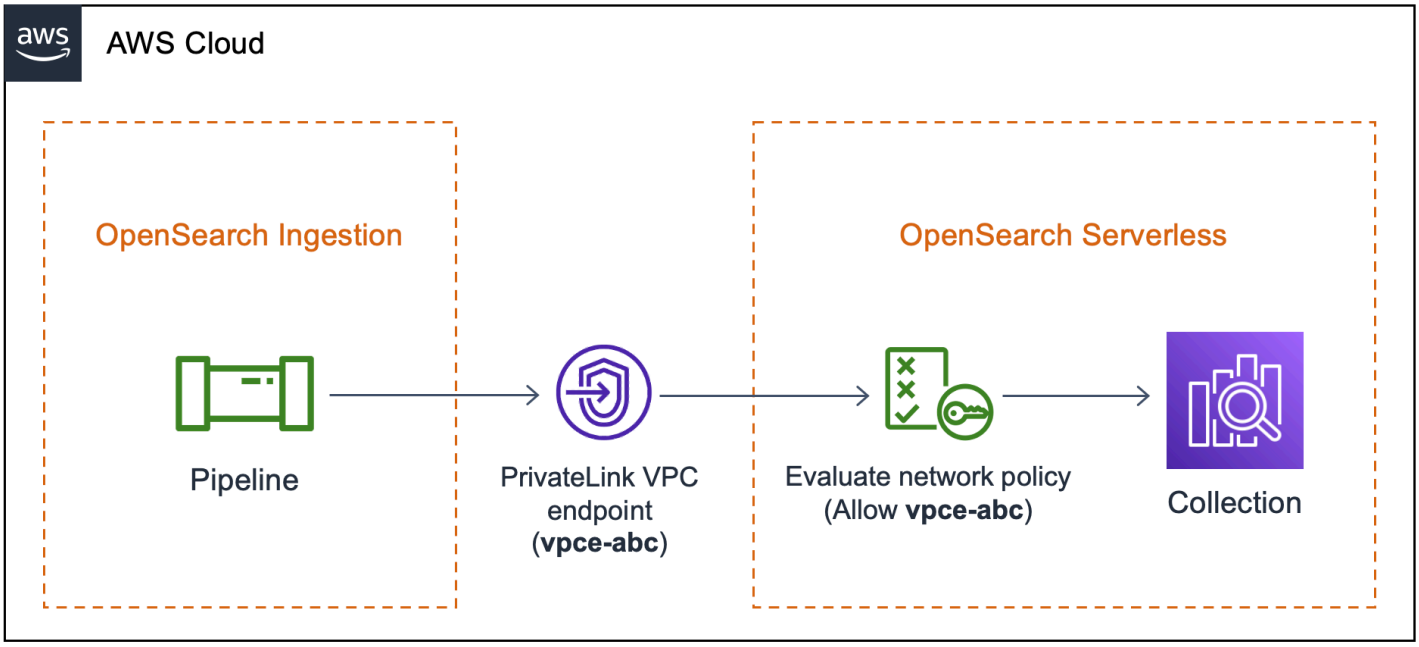
```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG}" ]
  sink:
    - opensearch:
        hosts: [ "https://search-{domain-name}.us-east-1.es.amazonaws.com" ]
        index: "my-index"
        aws:
          region: "{region}"
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
```

必要なパラメータとサポートされていないパラメータの詳細については、「the section called “サポートされているプラグインとオプション”」を参照してください。

Amazon OpenSearch Ingestion パイプラインにコレクションへのアクセス権を付与する

Amazon OpenSearch Ingestion パイプラインは、 OpenSearch サーバーレスのパブリックコレクションまたは VPC コレクションに書き込むことができます。コレクションへのアクセスを提供するには、コレクションへのアクセスを許可するアクセス権限ポリシーを使用して AWS Identity and Access Management (IAM) パイプラインロールを設定します。パイプライン設定でロールを指定する前に、適切な信頼関係を使用してロールを設定し、データアクセスポリシーを通じてデータアクセス権限を付与する必要があります。

パイプラインの作成時に、 OpenSearch Ingestion AWS PrivateLink OpenSearch はパイプラインとサーバーレスコレクション間の接続を作成します。パイプラインからのすべてのトラフィックは、この VPC エンドポイントを経由し、コレクションにルーティングされます。コレクションに到達するには、エンドポイントにネットワークアクセスポリシーを通じてコレクションへのアクセスを許可する必要があります。



トピック

- 制限事項
- パイプラインへのネットワークアクセスの提供
- ステップ 1: パイプラインロールを作成する
- ステップ 2: コレクションを作成する
- ステップ 3: パイプラインを作成する

制限事項

OpenSearch サーバーレスコレクションに書き込むパイプラインには、以下の制限が適用されます。

- 現在、OTel OpenSearch トレースグループプロセッサはサーバーレスコレクションシンクでは動作しません。
- 現在、 OpenSearch Ingestion `_template` はレガシー操作のみをサポートしていますが、OpenSearch サーバーレスはコンポーザブル操作をサポートしています。`_index_template`したがって、パイプライン設定に `index_type` オプションが含まれる場合は、これを `management_disabled` に設定する必要があります。

パイプラインへのネットワークアクセスの提供

OpenSearch サーバーレスで作成した各コレクションには、少なくとも 1 つのネットワークアクセスポリシーが関連付けられています。ネットワークアクセスポリシーは、パブリックネットワークからインターネット経由でコレクションにアクセスできるかどうか、またはプライベートにアクセスする必要があるかどうかを決定します。ネットワークポリシーの詳細については、を参照してくださいthe section called “ネットワークアクセス”。

ネットワークアクセスポリシー内では、 OpenSearch サーバーレス管理の VPC エンドポイントのみを指定できます。詳細については、「the section called “VPC エンドポイント”」を参照してください。ただし、パイプラインがコレクションに書き込むためには、 OpenSearch Ingestion がパイプラインとコレクションの間に自動的に作成する VPC エンドポイントへのアクセス権もポリシーで付与する必要があります。そのため、 OpenSearch サーバーレスコレクションシンクを含むパイプラインを作成する場合は、オプションを使用して関連するネットワークポリシーの名前を指定する必要があります。`network_policy_name`

例:

```
...
sink:
   - opensearch:
       hosts: [ "https://{collection-id}.{region}.aoss.amazonaws.com" ]
       index: "my-index"
        aws:
          serverless: true
          serverless_options:
            network_policy_name: "{network-policy-name}"
```

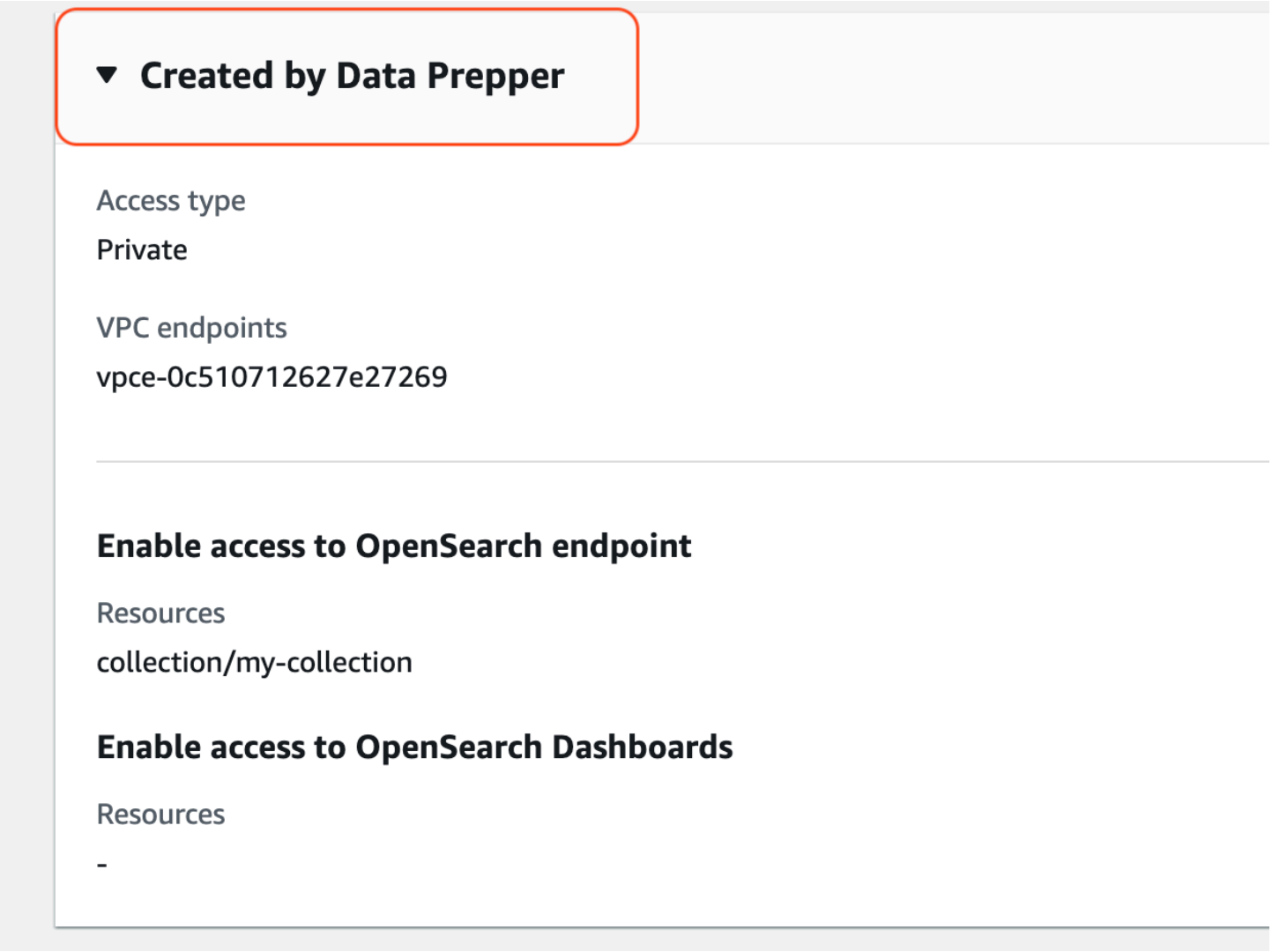
パイプラインの作成時に、 OpenSearch Ingestion は指定されたネットワークポリシーが存在するかどうかを確認します。存在しない場合、 OpenSearch Ingestion によって作成されます。存在する場合、 OpenSearch Ingestion は新しいルールを追加して更新します。このルールは、パイプラインとコレクションを接続する VPC エンドポイントへのアクセスを許可します。

例:

```
{
   "Rules":[
      {
         "Resource":[
            "collection/my-collection"
```

```
         ],
         "ResourceType":"collection"
      }
   ],
   "SourceVPCEs":[
      "vpce-0c510712627e27269" # The ID of the VPC endpoint that OpenSearch Ingestion
 creates between the pipeline and collection
   ],
   "Description":"Created by Data Prepper"
}
```

コンソールでは、 OpenSearch Ingestion によってネットワークポリシーに追加されるすべてのルールは Created by Data Prepper という名前になります。



> **Note**
>
> 一般に、コレクションへのパブリックアクセスを指定するルールは、プライベートアクセスを指定するルールよりも優先されます。したがって、ポリシーにすでにパブリックアクセスが設定されている場合、 OpenSearch Ingestion が追加したこの新しいルールによってポリシーの動作が実際に変わることはありません。詳細については、「the section called “ポリシーの優先順位”」を参照してください。

パイプラインを停止または削除すると、 OpenSearch Ingestion はパイプラインとコレクションの間の VPC エンドポイントを削除します。また、ネットワークポリシーを変更して、許可されたエンドポイントのリストから VPC エンドポイントを削除します。パイプラインを再起動すると、VPC エンドポイントが再作成され、エンドポイント ID を使用してネットワークポリシーが再更新されます。

ステップ 1: パイプラインロールを作成する

パイプライン設定の sts_role_arn パラメータで指定するロールには、コレクションシンクへのデータ送信を許可する、アクセス許可ポリシーが添付されている必要があります。また、 OpenSearch Ingestion がロールを引き継ぐことができる信頼関係も必要です。ポリシーをロールにアタッチする方法については、「IAM ユーザーガイド」の「IAM ID アクセス許可の追加」を参照してください。

次のポリシー例では、パイプライン設定の sts_role_arn ロールに、コレクションへの書き込みを可能にする最小特権が付与されています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "aoss:APIAccessAll",
                "aoss:BatchGetCollection",
                "aoss:CreateSecurityPolicy",
                "aoss:GetSecurityPolicy",
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

ロールには、 OpenSearch Ingestion が引き受けることができる以下の信頼関係が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "osis-pipelines.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

ステップ 2: コレクションを作成する

OpenSearch 以下の設定でサーバーレスコレクションを作成します。コレクションを作成する手順については、を参照してくださいthe section called “コレクションの作成”。

データアクセスポリシー

パイプラインロールに必要な権限を付与するコレクションのデータアクセスポリシーを作成します。
例:

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/{collection-name}/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
```

```
    "Principal": [
      "arn:aws:iam::{account-id}:role/{pipeline-role}"
    ],
    "Description": "Pipeline role access"
  }
]
```

> **Note**
>
> Principal の要素では、前のステップで作成したパイプラインロールの Amazon リソースネーム (ARN) を指定します。

ネットワークアクセスポリシー

コレクションのネットワークアクセスポリシーを作成します。パブリックコレクションまたは VPC コレクションにデータを取り込むことができます。たとえば、次のポリシーは、 OpenSearch サーバーレスが管理する単一の VPC エンドポイントへのアクセスを提供します。

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/{collection-name}"
            ]
         }
      ],
      "AllowFromPublic": false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   }
]
```

> **Important**
>
> network_policy_nameパイプライン設定のオプションにネットワークポリシーの名前を指定する必要があります。パイプラインの作成時に、 OpenSearch Ingestion はこのネットワー

クポリシーを更新して、パイプラインとコレクションの間に自動的に作成される VPC エンドポイントへのアクセスを許可します。パイプライン設定の例については、ステップ 3 を参照してください。詳細については、「the section called “パイプラインへのネットワークアクセスの提供”」を参照してください。

ステップ 3: パイプラインを作成する

最後に、パイプラインロールとコレクションの詳細を指定するパイプラインを作成します。パイプラインは、 OpenSearch サーバーレスコレクションシンクへのリクエストに署名するためにこの役割を引き受けます。

以下を実行するようにしてください。

- `hosts` のオプションで、ステップ 2 で作成したコレクションのエンドポイントを指定します。
- `sts_role_arn` のオプションで、ステップ 1 で作成したパイプラインロールの Amazon リソースネーム (ARN) を指定します。
- `serverless` のオプションは `true` に設定します。
- `network_policy_name`オプションをコレクションにアタッチされたネットワークポリシーの名前に設定します。 OpenSearch 取り込みでは、このネットワークポリシーが自動的に更新され、パイプラインとコレクションの間に作成される VPC からのアクセスが許可されます。詳細については、「the section called “パイプラインへのネットワークアクセスの提供”」を参照してください。

```
version: "2"
log-pipeline:
  source:
    http:
        path: "/log/ingest"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.{region}.aoss.amazonaws.com" ]
        index: "my-index"
        aws:
          serverless: true
```

```
          serverless_options:
            network_policy_name: "{network-policy-name}" # If the policy doesn't exist,
a new policy is created.
          region: "us-east-1"
          sts_role_arn: "arn:aws:iam::{account-id}:role/{pipeline-role}"
```

必要なパラメータとサポートされていないパラメータの詳細については、「the section called “サポートされているプラグインとオプション”」を参照してください。

Amazon OpenSearch Ingestion の開始方法

Amazon OpenSearch Ingestion は、マネージド型の OpenSearch Service ドメインと OpenSearch Serverless コレクションへのデータの取り込みに対応しています。次のチュートリアルでは、これらのユースケースごとに、パイプラインを起動して実行するための基本的な手順を説明します。

> **Note**
>
> 適切なアクセス許可を設定しなければ、パイプラインの作成は失敗します。パイプラインを作成する前に、必要なロールへの理解を深めるために「the section called “ロールとユーザーの設定”」を参照してください。

トピック

- チュートリアル: Amazon OpenSearch Ingestion を使用してドメインにデータを取り込む
- チュートリアル: Amazon Ingestion OpenSearch を使用してコレクションにデータを取り込む

チュートリアル: Amazon OpenSearch Ingestion を使用してドメインにデータを取り込む

このチュートリアルでは、Amazon Ingestion OpenSearch を使用してシンプルなパイプラインを設定し、Amazon OpenSearch Service ドメインにデータを取り込む方法を示します。パイプラインは、Ingestion OpenSearch がプロビジョニングして管理するリソースです。パイプラインを使用して、 OpenSearch Service でダウンストリームの分析と視覚化のためにデータをフィルタリング、強化、変換、正規化、集計できます。

このチュートリアルでは、パイプラインをすばやく起動するための基本的な手順を説明します。詳細については、「the section called “パイプラインの作成”」を参照してください。

このチュートリアルでは、次の手順を実行します。

1. パイプラインロールを作成する
2. ドメインを作成する
3. パイプラインを作成する
4. サンプルデータを取り込む

このチュートリアルでは、次のリソースを作成します。

- ingestion-pipeline という名前のパイプライン
- パイプラインの書き込み先となる、ingestion-domain という名前のドメイン
- パイプラインがドメインに書き込むときにこのロールを引き受ける、PipelineRole という名前の IAM ロール

必要なアクセス許可

このチュートリアルを完了するには、適切な IAM のアクセス許可を持っている必要があります。ユーザーまたはロールには、次の最低限の許可を含む ID ベースのポリシーが、アタッチされている必要があります。これらの許可により、パイプラインロールの作成 (iam:Create)、ドメインの作成または変更 (es:*)、パイプラインの使用 (osis:*) が可能になります。

さらに、パイプラインロールリソースには iam:PassRole 許可が必要です。このアクセス許可により、パイプラインロールを Ingestion OpenSearch に渡して、ドメインにデータを書き込むことができます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:*",
            "iam:Create*",
            "es:*"
         ]
      },
      {
         "Resource":[
```

```
            "arn:aws:iam::{your-account-id}:role/PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

ステップ 1: パイプラインロールを作成する

まず、 OpenSearch サービスドメインシンクにアクセスするためにパイプラインが引き受けるロールを作成します。これは、このチュートリアルの後半でパイプライン設定に含めます。

パイプラインロールを作成するには

1. https://console.aws.amazon.com/iamv2/ で AWS Identity and Access Management コンソールを開きます。
2. [ポリシー] を選択してから、[ポリシーを作成] を選択します。
3. このチュートリアルでは、次のステップで作成する ingestion-domain というドメインにデータを取り込みます。[JSON] を選択し、次のポリシーをエディタに貼り付けます。{your-account-id} を自分のアカウント ID に置き換え、必要に応じてリージョンを変更します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "es:DescribeDomain",
            "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-
domain"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-
domain/*"
        }
    ]
}
```

既存のドメインにデータを書き込む場合は、ingestion-domain を自分のドメイン名に置き換えます。

> **Note**
>
> このチュートリアルでは、わかりやすいようにかなり広範なアクセスポリシーを使用します。しかし、本番稼働用環境では、パイプラインロールに制限の厳しいアクセスポリシーを適用することをお勧めします。必要最低限のアクセス許可を付与するポリシーの例については、「the section called “パイプラインにドメインへのアクセス権を付与する”」を参照してください。

4. [次へ]、[次へ] の順にクリックし、ポリシーに pipeline-policy という名前を付けます。
5. [ポリシーの作成] を選択します。
6. 次に、ポリシーを作成してロールにアタッチします。[ロール]、[ロールの作成] の順に選択します。
7. [カスタム信頼ポリシー] をクリックし、次のポリシーをエディタに貼り付けます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"osis-pipelines.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

8. [次へ] をクリックします。その後、pipeline-policy (先ほど作成したもの) を検索して選択します。
9. Next を選択し、ロールに という名前を付けますPipelineRole。
10. [ロールの作成] を選択します。

ロールの Amazon リソースネーム (ARN) (例: `arn:aws:iam::{your-account-id}:role/PipelineRole`) を覚えておいてください。これは、パイプラインを作成するときに必要です。

ステップ 2: ドメインを作成する

次に、データを取り込む、ingestion-domain という名前のドメインを作成します。

https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールに移動し、次の要件を満たすドメインを作成します。

- OpenSearch 1.0 以降、または Elasticsearch 7.4 以降を実行している
- パブリックアクセスを使用する
- きめ細かいアクセスコントロールを使用しない

> **Note**
>
> これらの要件は、このチュートリアルをわかりやすくするためのものです。本番環境では、VPC アクセスを使用してドメインを設定できるほか、きめ細かなアクセスコントロールを使用することも可能です。きめ細かなアクセスコントロールを使用するには、「パイプラインロールのマッピング」を参照してください。

ドメインには、前のステップで作成した PipelineRole にアクセス許可を付与するアクセスポリシーが必要です。パイプラインは、 OpenSearch サービスドメインシンクにデータを送信するために、このロール (パイプライン設定では sts_role_arn という名前) を引き受けます。

ドメインが次のドメインレベルのアクセスポリシーを持っていることを確認します。このポリシーは、ドメインに PipelineRole アクセスを付与します。リージョンとアカウント ID を自分のものに置き換えます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/PipelineRole"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-domain/*"
    }
  ]
```

```
}
```

ドメインレベルのアクセスポリシーの作成について、詳しくは「リソースベースのアクセスポリシー」を参照してください。

ドメインを既に作成済みの場合、その既存のアクセスポリシーを変更し、PipelineRole に上記のアクセス許可を付与します。

> **Note**
>
> ドメインエンドポイント (例:https://search-*ingestion-domain*.us-east-1.es.amazonaws.com) を覚えておいてください。次のステップでは、これを使用してパイプラインを設定します。

ステップ 3: パイプラインを作成する

適切なアクセス許可を持つドメインとロールが用意できたので、パイプラインを作成できます。

パイプラインを作成するには

1. Amazon OpenSearch Service コンソールで、左側のナビゲーションペインからパイプラインを選択します。
2. パイプラインの作成 を選択します。
3. パイプラインに ingestion-pipeline という名前を付け、キャパシティ設定はデフォルトのままにします。
4. このチュートリアルでは、log-pipeline というシンプルなサブパイプラインを作成し、Http ソースプラグインを使用します。このプラグインは、JSON 配列形式の口グデータを受け入れます。シンクとして単一の OpenSearch サービスドメインを指定し、すべてのデータをapplication_logsインデックスに取り込みます。

   [パイプライン設定] で、次の YAML 設定をエディタに貼り付けます。

   ```
   version: "2"
   log-pipeline:
     source:
       http:
         path: "/${pipelineName}/test_ingestion_path"
     processor:
       - date:
   ```

```
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://search-ingestion-domain.us-east-1.es.amazonaws.com" ]
        index: "application_logs"
        aws:
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
          region: "us-east-1"
```

> **Note**
>
> path オプションは取り込み用の URI パスを指定します。このオプションは、プル型のソースの場合に必要です。詳細については、「the section called “取り込みパスの指定”」を参照してください。

5. hosts URL を、前のセクションで作成 (または変更) したドメインのエンドポイントに置き換えます。sts_role_arn パラメータを PipelineRole の ARN に置き換えます。
6. [パイプラインを検証] をクリックし、検証が成功したことを確認します。
7. このチュートリアルでは、わかりやすいようにパイプラインにパブリックアクセスを設定します。[ネットワーク] で [パブリックアクセス] を選択します。

   VPC アクセスの設定の詳細については、「the section called “パイプラインの VPC アクセスの設定”」を参照してください。
8. このチュートリアルを完了する間に問題が発生した場合に備えて、ログ発行を有効にしておきます。詳細については、「the section called “パイプラインのログのモニタリング”」を参照してください。

   ロググループ名を次のように指定します: /aws/vendedlogs/OpenSearchIngestion/ingestion-pipeline/audit-logs
9. [次へ] をクリックします。パイプライン設定を確認し、[パイプラインの作成] をクリックします。パイプラインがアクティブになるまでに 5 ~ 10 分かかります。

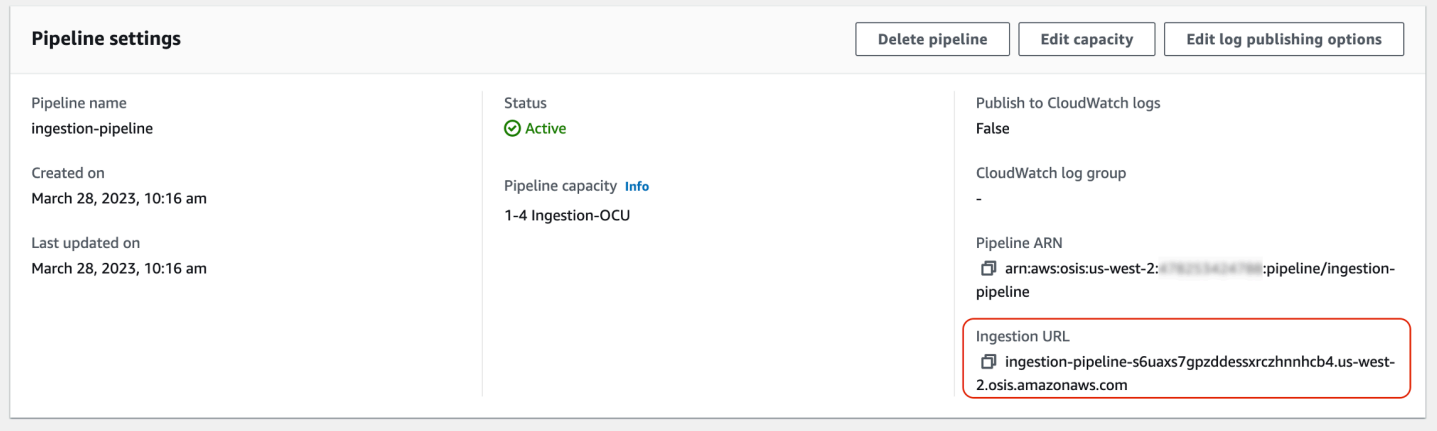
ステップ 4: サンプルデータを取り込む

パイプラインのステータスが Active になると、パイプラインへのデータの取り込みを開始できます。Signature Version 4 を使用して、パイプラインへのすべての HTTP リクエストに署名する必要があります。Postman や awscurl などの HTTP ツールを使用して、パイプラインにデータを送信し

ます。データのインデックス作成をドメインに直接行う場合と同様に、パイプラインにデータを取り込むには、常に IAM ロールまたは IAM アクセスキーとシークレットキーのいずれかが必要です。

> **Note**
>
> リクエストに署名するプリンシパルは、osis:Ingest という IAM アクセス許可を持っている必要があります。

まず、[パイプライン設定] ページから取り込み URL を取得します。



次に、いくつかのサンプルデータを取り込みます。次のリクエストでは、awscurl を使用して 1 つのログファイルを application_logs インデックスに送信します。

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
    https://{pipeline-endpoint}.us-east-1.osis.amazonaws.com/log-pipeline/
test_ingestion_path
```

200 OK というレスポンスが確認できるはずです。認証エラーが発生した場合は、パイプラインとは別のアカウントからデータを取り込んでいることが原因かもしれません。the section called “アクセス許可に関する問題に対応する” を参照してください。

次に、application_logs インデックスにクエリを実行して、ログエントリが正常に取り込まれたことを確認します。

```
awscurl --service es --region us-east-1 \
     -X GET \
     https://search-{ingestion-domain}.us-east-1.es.amazonaws.com/application_logs/
_search | json_pp
```

レスポンス例:


```
{
   "took":984,
   "timed_out":false,
   "_shards":{
      "total":1,
      "successful":5,
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"application_logs",
            "_type":"_doc",
            "_id":"z6VY_IMBRpceX-DU6V4O",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2022-10-21T21:00:25.502Z"
            }
         }
      ]
   }
}
```

アクセス許可に関する問題に対応する

チュートリアルのステップに従っても、データの取り込みを試みても認証エラーが表示される場合は、パイプラインに書き込むロールがパイプライン自体 AWS アカウント とは異なる にある可能性があります。この場合は、明確にデータの取り込みを可能にするロールを作成して引き受ける必要があります。手順については、「the section called “クロスアカウント取り込みアクセスの提供”」を参照してください。

関連リソース

このチュートリアルでは、HTTP 経由で 1 つのドキュメントを取り込むというシンプルなユースケースを紹介しました。本番シナリオでは、クライアントアプリケーション (Fluent Bit、Kubernetes、OpenTelemetry コレクターなど) を設定して、1 つ以上のパイプラインにデータを送信します。パイプラインは、このチュートリアルのシンプルな例よりも複雑になる可能性があります。

クライアントの設定とデータの取り込みを開始するには、次のリソースを参照してください。

- パイプラインの作成と管理
- Ingestion OpenSearch にデータを送信するようにクライアントを設定する
- Data Prepper のドキュメント

チュートリアル: Amazon Ingestion OpenSearch を使用してコレクションにデータを取り込む

このチュートリアルでは、Amazon OpenSearch Ingestion を使用してシンプルなパイプラインを設定し、Amazon OpenSearch Serverless コレクションにデータを取り込む方法を示します。パイプラインは、Ingestion OpenSearch がプロビジョニングおよび管理するリソースです。パイプラインを使用して、 OpenSearch サービスのダウンストリーム分析と可視化のためにデータをフィルタリング、強化、変換、正規化、集計できます。

プロビジョニングされた OpenSearch サービスドメインにデータを取り込む方法を示すチュートリアルについては、「」を参照してくださいthe section called “チュートリアル: ドメインにデータを取り込む”。

このチュートリアルでは、次の手順を実行します。

1. パイプラインロールを作成する
2. コレクションを作成します。

3. パイプラインを作成する
4. サンプルデータを取り込む

このチュートリアルでは、次のリソースを作成します。

- ingestion-pipeline-serverless という名前のパイプライン
- パイプラインの書き込み先となる、ingestion-collection という名前のコレクション
- パイプラインがコレクションに書き込むために引き受ける、PipelineRole という名前の IAM ロール

必要なアクセス許可

このチュートリアルを完了するには、適切な IAM のアクセス許可を持っている必要があります。ユーザーまたはロールには、次の最低限の許可を含む ID ベースのポリシーが、アタッチされている必要があります。これらの許可により、パイプラインロールの作成 (iam:Create*)、コレクションの作成または変更 (aoss:*)、パイプラインの使用 (osis:*) が可能になります。

さらに、パイプラインロールリソースには iam:PassRole 許可が必要です。このアクセス許可により、パイプラインロールを Ingestion OpenSearch に渡して、コレクションにデータを書き込むことができます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:*",
            "iam:Create*",
            "aoss:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
```

```
            "iam:PassRole"
         ]
      }
   ]
}
```

ステップ 1: パイプラインロールを作成する

まず、 OpenSearch Serverless コレクションシンクにアクセスするためにパイプラインが引き受けるロールを作成します。これは、このチュートリアルの後半でパイプライン設定に含めます。

パイプラインロールを作成するには

1. https://console.aws.amazon.com/iamv2/ で AWS Identity and Access Managementコンソールを開きます。
2. [ポリシー] を選択してから、[ポリシーを作成] を選択します。
3. [JSON] を選択し、次のポリシーをエディタに貼り付けます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:BatchGetCollection",
        "aoss:APIAccessAll"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:aoss:{region}:{your-account-id}:collection/{collection-
id}"
    },
    {
      "Action": [
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aoss:collection": "{collection-name}"
        }
```

```
      }
    }
  ]
}
```

4. 次へを選択し、次へを選択し、ポリシーに という名前を付けますcollection-pipeline-policy。
5. [ポリシーの作成] を選択します。
6. 次に、ポリシーを作成してロールにアタッチします。[ロール]、[ロールの作成] の順に選択します。
7. [カスタム信頼ポリシー] をクリックし、次のポリシーを エディタに貼り付けます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "Service":"osis-pipelines.amazonaws.com"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

8. [次へ] をクリックします。次にcollection-pipeline-policy、 (先ほど作成したもの) を検索して選択します。
9. Next を選択し、ロールに という名前を付けますPipelineRole。
10. [ロールの作成] を選択します。

ロールの Amazon リソースネーム (ARN) (例: `arn:aws:iam::{your-account-id}:role/PipelineRole`) を覚えておいてください。これは、パイプラインを作成するときに必要です。

ステップ 2: コレクションを作成する

次に、データを取り込むコレクションを作成します。コレクションに `ingestion-collection` という名前を付けます。

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールに移動します。

2. 左側のナビゲーションペインで [コレクション] をクリックし、次に [コレクションを作成] をクリックします。
3. コレクションに [ingestion-collection] という名前を付けます。
4. [ネットワークアクセスの設定] で、アクセスタイプを [パブリック] に変更します。
5. 他のすべての設定をデフォルトのままにして、[Next] (次へ) を選択します。
6. [定義方法] で [JSON] をクリックし、エディタに以下のポリシーを貼り付けます。このポリシーは次の 2 つを実行します。

   - パイプラインロールがコレクションに書き込むことを許可します。
   - コレクションから読み取ることを許可します。そして、いくつかのサンプルデータをパイプラインに取り込んだ後、コレクションにクエリを実行し、データが正常に取り込まれてインデックスに書き込まれたことを確認します。

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/ingestion-collection/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::{your-account-id}:role/PipelineRole",
      "arn:aws:iam::{your-account-id}:role/Admin"
    ],
    "Description": "Rule 1"
  }
]
```

7. `Principal` エレメントを置き換えます。最初のプリンシパルでは、作成したパイプラインロールを指定する必要があります。2 つ目のプリンシパルでは、後でコレクションをクエリするために使用できるユーザーまたはロールを指定する必要があります。
8. [次へ] をクリックします。アクセスポリシーに名前を付けpipeline-domain-access、もう一度次へを選択します。
9. コレクション設定を確認してから、[Submit] (送信) を選択します。

コレクションがアクティブになったら、 OpenSearch エンドポイントの下のエンドポイントを書き留めます (例: `https://{collection-id}.us-east-1.aoss.amazonaws.com`) 。これは、パイプラインを作成するときに必要です。

ステップ 3: パイプラインを作成する

コレクションと適切なアクセス許可を持つロールを作成できたので、パイプラインを作成できます。

パイプラインを作成するには

1. Amazon OpenSearch Service コンソールで、左側のナビゲーションペインからパイプラインを選択します。
2. パイプラインの作成 を選択します。
3. パイプラインに serverless-ingestion という名前を付け、キャパシティ設定はデフォルトのままにします。
4. このチュートリアルでは、`log-pipeline` というシンプルなサブパイプラインを作成し、HTTP ソースプラグインを使用します。プラグインは、JSON 配列形式のログデータを受け入れます。1 つの OpenSearch Serverless コレクションをシンクとして指定し、すべてのデータを`my_logs`インデックスに取り込みます。

   [パイプライン設定] で、次の YAML 設定をエディタに貼り付けます。

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/test_ingestion_path"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
```

```
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.us-east-1.aoss.amazonaws.com" ]
        index: "my_logs"
        aws:
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
          region: "us-east-1"
          serverless: true
```

5. hosts URL を、前のセクションで作成したコレクションのエンドポイントに置き換えます。sts_role_arn パラメータを PipelineRole の ARN に置き換えます。必要に応じて region を変更します。
6. [パイプラインを検証] をクリックし、検証が成功したことを確認します。
7. このチュートリアルでは、わかりやすいようにパイプラインにパブリックアクセスを設定します。[ネットワーク] で [パブリックアクセス] を選択します。

   VPC アクセスの設定の詳細については、「the section called “パイプラインの VPC アクセスの設定”」を参照してください。
8. このチュートリアルを完了する間に問題が発生した場合に備えて、ログ発行を有効にしておきます。詳細については、「the section called “パイプラインのログのモニタリング”」を参照してください。

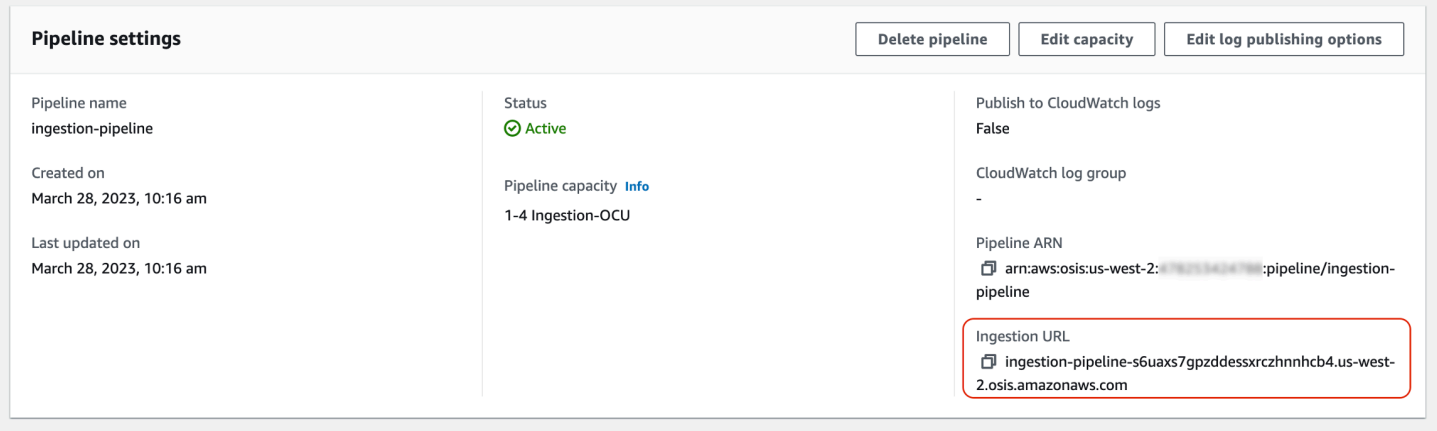
   ロググループ名を次のように指定します: /aws/vendedlogs/OpenSearchIngestion/serverless-ingestion/audit-logs
9. [次へ] をクリックします。パイプライン設定を確認し、[パイプラインの作成] をクリックします。パイプラインがアクティブになるまでに 5 ~ 10 分かかります。

ステップ 4: サンプルデータを取り込む

パイプラインのステータスが Active になると、パイプラインへのデータの取り込みを開始できます。Signature Version 4 を使用して、パイプラインへのすべての HTTP リクエストに署名する必要があります。Postman や awscurl などの HTTP ツールを使用して、パイプラインにデータを送信します。データのインデックス作成をコレクションに直接行う場合と同様に、パイプラインにデータを取り込むには、常に IAM ロールまたは IAM アクセスキーとシークレットキーのいずれかが必要です。

> **Note**
>
> リクエストに署名するプリンシパルは、`osis:Ingest` という IAM アクセス許可を持っている必要があります。

まず、[パイプライン設定] ページから取り込み URL を取得します。



次に、いくつかのサンプルデータを取り込みます。次のサンプルリクエストでは、awscurl を使用して 1 つのログファイルを my_logs インデックスに送信します。

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
    https://{pipeline-endpoint}.us-east-1.osis.amazonaws.com/log-pipeline/
test_ingestion_path
```

200 OK というレスポンスが確認できるはずです。

次に、my_logs インデックスにクエリを実行して、ログエントリが正常に取り込まれたことを確認します。

```
awscurl --service aoss --region us-east-1 \
     -X GET \
     https://{collection-id}.us-east-1.aoss.amazonaws.com/my_logs/_search | json_pp
```

レスポンス例:


```
{
   "took":348,
   "timed_out":false,
   "_shards":{
      "total":0,
      "successful":0,
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"my_logs",
            "_id":"1%3A0%3ARJgDvIcBTy5m12xrKE-y",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2023-04-26T05:22:16.204Z"
            }
         }
      ]
   }
}
```


関連リソース

このチュートリアルでは、HTTP 経由で 1 つのドキュメントを取り込むというシンプルなユースケースを紹介しました。本番シナリオでは、1 つ以上のパイプラインにデータを送信するようにクライアントアプリケーション (Fluent Bit、Kubernetes、 OpenTelemetry コレクターなど) を設定します。パイプラインは、このチュートリアルのシンプルな例よりも複雑になる可能性があります。

クライアントの設定とデータの取り込みを開始するには、次のリソースを参照してください。

- パイプラインの作成と管理
- Ingestion OpenSearch にデータを送信するようにクライアントを設定する
- Data Prepper のドキュメント

Amazon OpenSearch インジェストのパイプライン機能の概要

Amazon OpenSearch Ingestion は、ソース、バッファ、0 個以上のプロセッサ、1 つ以上のシンクで構成されるパイプラインをプロビジョニングします。取り込みパイプラインは、データエンジンとして Data Prepper を利用しています。パイプラインのさまざまなコンポーネントの概要については、「the section called “主要なコンセプト”」を参照してください。

以下のセクションでは、Amazon OpenSearch Ingestion で最も一般的に使用される機能の概要を説明します。

> **Note**
>
> これは、パイプラインで利用可能な機能をすべて網羅したリストではありません。使用可能なすべてのパイプライン機能に関する包括的なドキュメントについては、「Data Prepper のドキュメント」を参照してください。 OpenSearch Ingestion では、使用できるプラグインとオプションにいくつかの制約があることに注意してください。詳細については、「the section called “サポートされているプラグインとオプション”」を参照してください。

トピック

- 永続的バッファリング
- 分割
- チェーン
- デッドレターキュー
- インデックス管理
- E nd-to-end 確認応答
- ソースバックプレッシャー

永続的バッファリング

永続バッファは、データの耐久性を高めるため、複数のアベイラビリティーゾーンにまたがるディスクベースのバッファにデータを格納します。永続バッファリングを使用すると、スタンドアロンバッファを設定しなくても、サポートされているすべてのプッシュベースソースのデータを取り込むことができます。これらには HTTP のほか、ログ、トレース、 OpenTelemetry メトリクスのソースが含まれます。

永続的バッファリングを有効にするには、パイプラインを作成または更新するときに [永続的バッファを有効にする] を選択します。詳細については、「」を参照してくださいthe section called “パイプラインの作成”。 OpenSearch インジェストは、 OpenSearch パイプラインに指定したインジェストコンピュートユニット (Ingestion OCU) に基づいて、必要なバッファリング容量を自動的に決定します。

デフォルトでは、パイプラインはを使用してバッファーデータを暗号化します。 AWS 所有のキー これらのパイプラインには、パイプラインロールに追加の権限は必要ありません。または、カスタマー管理キーを指定して、パイプラインロールに次の IAM 権限を追加することもできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KeyAccess",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "arn:aws:kms:{region}:{aws-account-
id}:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

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。

> **Note**
>
> 永続的バッファリングを無効にすると、パイプラインが更新され、メモリ内バッファリングのみで実行されるようになります。

リクエストペイロードの最大サイズのチューニング

パイプラインの永続的バッファリングを有効にすると、リクエストペイロードの最大サイズはデフォルトで 1 MB になります。デフォルト値が最も高いパフォーマンスを発揮します。ただし、クライアントが 1 MB を超えるリクエストを送信する場合は、この値を増やすことができます。最大ペイロードサイズを調整するには、`max_request_length`ソース設定でオプションを設定します。永続的バッファリングと同様に、このオプションは HTTP とログ、トレース、 OpenTelemetry メトリクスのソースでのみサポートされます。

`max_request_length`このオプションで有効な値は、1 MB、1.5 MB、2 MB、2.5 MB、3 MB、3.5 MB、4 MB だけです。別の値を指定すると、エラーが表示されます。

次の例は、パイプライン設定内の最大ペイロードサイズを設定する方法を示しています。

```
...
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
      max_request_length: 4mb
  processor:
...
```

パイプラインの永続的バッファリングを有効にしない場合、`max_request_length`オプションの値はすべてのソースでデフォルトで 10 MB になり、変更できません。

分割

受信イベントをサブパイプラインに分割するように OpenSearch Ingestion パイプラインを設定すると、同じ受信イベントに対して異なる種類の処理を実行できます。

次のパイプラインの例では、受信イベントを 2 つのサブパイプラインに分割します。各サブパイプラインは独自のプロセッサを使用してデータを強化および操作し、データを異なるインデックスに送信します。 OpenSearch

```
version: "2"
log-pipeline:
  source:
    http:
    ...
  sink:
    - pipeline:
        name: "logs_enriched_one_pipeline"
    - pipeline:
        name: "logs_enriched_two_pipeline"

logs_enriched_one_pipeline:
  source:
    log-pipeline
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
        index: "enriched_one_logs"

logs_enriched_two_pipeline:
  source:
    log-pipeline
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_two_logs"
```

チェーン

複数のサブパイプラインを連結し、データ処理とエンリッチメントをチャンク単位で実行できます。つまり、受信イベントをあるサブパイプラインの特定の処理機能で強化し、それを別のサブパイプラインに送信して別のプロセッサでさらに強化し、最後にシンクに送信できます。 OpenSearch

次の例では、log_pipelineサブパイプラインは受信ログイベントを一連のプロセッサで強化し、そのイベントをという名前のインデックスに送信します。 OpenSearch enriched_logsパイプラインは同じイベントをサブパイプラインに送信し、log_advanced_pipelineサブパイプラインはそれを処理して、という名前の別のインデックスに送信します。 OpenSearch enriched_advanced_logs

```
version: "2"
log-pipeline:
  source:
    http:
    ...
  processor:
    ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_logs"
    - pipeline:
        name: "log_advanced_pipeline"

log_advanced_pipeline:
  source:
    log-pipeline
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
```

```
          index: "enriched_advanced_logs"
```

デッドレターキュー

デッドレターキュー (DLQ) とは、パイプラインがシンクへの書き込みに失敗したイベントの送信先です。 OpenSearch Ingestion では、DLQ として使用する適切な書き込み権限を持つ Amazon S3 バケットを指定する必要があります。パイプライン内のすべてのシンクに DLQ 設定を追加できます。パイプラインで書き込みエラーが発生すると、設定された S3 バケットに DLQ オブジェクトが作成されます。DLQ オブジェクトは、失敗したイベントの配列として JSON ファイル内に存在します。

次のいずれかの条件が満たされたとき、パイプラインは DLQ にイベントを書き込みます。

- `max_retries` OpenSearch シンク用のは使い果たされました。 OpenSearch このオプションでは、取り込みには最低でも 16 個必要です。
- エラー状態のため、イベントがシンクによって拒否されたとき。

構成

サブパイプラインのデッドレターキューを設定するには、`opensearch` シンク設定内で `dlq` オプションを指定します。

```
apache-log-pipeline:
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/dlq-role"
```

この S3 DLQ に書き込まれたファイルには、次の命名パターンが付けられます。

```
dlq-v${version}-${pipelineName}-${pluginId}-${timestampIso8601}-${uniqueId}
```

詳細については、「デッドレターキュー (DLQ)」を参照してください。

`sts_role_arn` ロールを設定する手順については、「the section called “デッドレターキューへの書き込み”」を参照してください。

例

次の DLQ ファイルの例を考えてみます。

```
dlq-v2-apache-log-pipeline-opensearch-2023-04-05T15:26:19.152938Z-e7eb675a-
f558-4048-8566-dac15a4f8343
```

次は、シンクへの書き込みに失敗したデータの例です。これは、さらなる分析のために DLQ S3 バケットに送信されます。

```
Record_0
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
failedData
index    "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log      "sample log"
timestamp     "2023-04-14T10:36:01.070Z"

Record_1
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
failedData
index               "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log                 "another sample log"
timestamp           "2023-04-14T10:36:01.071Z"
```

インデックス管理

Amazon OpenSearch Ingestion には、次のような多数のインデックス管理機能があります。

インデックスの作成

パイプラインシンクでインデックス名を指定でき、 OpenSearch Igestion はパイプラインをプロビジョニングするときにインデックスを作成します。インデックスが既に存在する場合、パイプラインはそれを使用して受信イベントのインデックスを作成します。インデックスがまだ存在しない場合、パイプラインを停止して再起動する、または YAML 設定を更新すると、パイプラインは新しいインデックスの作成を試みます。パイプラインではインデックスを一切削除できません。

次のシンクのサンプルでは、パイプラインがプロビジョニングされるときに 2 つのインデックスを作成します。

```
sink:
  - opensearch:
      index: apache_logs
  - opensearch:
      index: nginx_logs
```

インデックス名とパターンの生成

受信イベントのフィールドにある変数を使用すると、動的なインデックス名を生成できます。シンク設定では、形式 `string${}` を使用して文字列補間を示し、JSON ポインタを使用してイベントからフィールドを抽出します。`index_type` のオプションは `custom` または `management_disabled` です。`index_type` OpenSearch `management_disabled` OpenSearch ドメインとサーバーレスコレクションではデフォルトになるため、`custom`未設定のままでかまいません。

例えば、次のパイプラインは、受信イベントから `metadataType` フィールドを選択してインデックス名を生成します。

```
pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}"
```

次の設定では、1 日または 1 時間ごとに新しいインデックスを生成し続けます。

```
pipeline:
  ...
  sink:
```

```
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"

pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd.HH}"
```

インデックス名は、my-index-%{yyyy.MM.dd} のように、サフィックスとして日付/時刻パターンを持つプレーン文字列にすることもできます。シンクがにデータを送信すると OpenSearch、日時パターンが UTC 時間に置き換えられ、日ごとに新しいインデックス (など) が作成されます。my-index-2022.01.25詳細については、クラスを参照してください。DateTimeFormatter

このインデックス名は、my-${index}-name のように形式化された文字列にすることもできます (日付/時刻パターンのサフィックスの有無にかかわらず)。シンクはにデータを送信すると OpenSearch、"${index}"その部分を処理中のイベントの値に置き換えます。形式が "${index1/index2/index3}" の場合、フィールド index1/index2/index3 をイベント内の値に置き換えます。

ドキュメント ID の生成

パイプラインは、ドキュメントのインデックスを作成する際にドキュメント ID を生成できます。OpenSearchまた、それらのドキュメント ID を受信イベント内のフィールドから推測することも可能です。

次の例では、受信イベントの uuid フィールドを使用してドキュメント ID を生成します。

```
pipeline:
  ...
  sink:
    opensearch:
      index_type: custom
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"
      document_id_field: "uuid"
```

次の例では、Add entries プロセッサを使用し、受信イベントから uuid フィールドと other_field フィールドをマージしてドキュメント ID を生成します。

create アクションは、同じ ID のドキュメントが上書きされないようにします。パイプラインは再試行や DLQ イベントを必要とせずに、重複したドキュメントを削除します。ここでの目的は、既存

ドキュメントの更新を避けることなので、このアクションを使用するパイプライン作成者にとっては当然想定されるものです。

```
pipeline:
  ...
  processor:
   - add_entries:
      entries:
        - key: "my_doc_id_field"
          format: "${uuid}-${other_field}"
  sink:
    - opensearch:
       ...
       action: "create"
       document_id_field: "my_doc_id_field"
```

イベントのドキュメント ID をサブオブジェクトのフィールドに設定したい場合があります。次の例では、 OpenSearch `info/id`シンクプラグインはサブオブジェクトを使用してドキュメント ID を生成します。

```
sink:
  - opensearch:
       ...
       document_id_field: info/id
```

次のイベントが発生すると、パイプラインは `_id` フィールドに `json001` を設定したドキュメントを生成します。

```
{
   "fieldA":"arbitrary value",
   "info":{
      "id":"json001",
      "fieldA":"xyz",
      "fieldB":"def"
   }
}
```

ルーティング ID の生成

`routing_field` OpenSearch シンクプラグイン内のオプションを使用して、ドキュメントルーティングプロパティ (`_routing`) の値を受信イベントの値に設定できます。

ルーティングは JSON ポインタ構文をサポートしているため、最上位のフィールドだけでなく、ネストされたフィールドも使用できます。

```
sink:
  - opensearch:
        ...
        routing_field: metadata/id
        document_id_field: id
```

次のイベントが発生すると、プラグインは _routing フィールドに abcd を設定したドキュメントを生成します。

```
{
   "id":"123",
   "metadata":{
      "id":"abcd",
      "fieldA":"valueA"
   },
   "fieldB":"valueB"
}
```

インデックスを作成するときに、パイプラインで使用できるインデックステンプレートを作成する手順については、「インデックステンプレート」を参照してください。

E nd-to-end 確認応答

OpenSearch 取り込みでは、確認応答を使用してステートレスパイプラインのソースからシンクへのデータ配信を追跡することで、データの耐久性と信頼性を確保します。end-to-end現在、確認をサポートしているのは S3 ソースプラグインだけです。 end-to-end

end-to-end 承認されると、パイプラインソースプラグインは受信確認セットを作成してイベントのバッチを監視します。イベントがシンクに正常に送信された場合は肯定応答を受け取り、いずれかのイベントがシンクに送信できなかった場合は否定応答を受け取ります。

パイプラインコンポーネントに障害またはクラッシュが発生した場合、またはソースが確認応答を受け取れなかった場合、ソースはタイムアウトし、再試行や障害のログ記録などの必要なアクションを実行します。パイプラインに複数のシンクまたは複数のサブパイプラインが設定されている場合、イベントレベルの確認応答は、イベントが全サブパイプラインの全シンクに送信された後にのみ送信されます。シンクに DLQ が設定されている場合、 end-to-end 確認応答は DLQ に書き込まれたイベントも追跡します。

end-to-end 確認を有効にするには、ソース設定に以下のオプションを含めてください。acknowledgments

```
s3-pipeline:
  source:
    s3:
      acknowledgments: true
...
```

ソースバックプレッシャー

パイプラインは、データ処理が忙しい場合や、シンクが一時的にダウンしていたり、データの取り込みが遅い場合に、バックプレッシャーが発生する可能性があります。 OpenSearch インジェストでは、パイプラインが使用しているソースプラグインによって、バックプレッシャの処理方法が異なります。

HTTP ソース

HTTP ソースプラグインを使用するパイプラインでは、混雑しているパイプラインコンポーネントによってバックプレッシャーの処理方法が異なります。

- バッファ - バッファがいっぱいになると、パイプラインはエラーコード 408 の HTTP ステータス REQUEST_TIMEOUT をソースエンドポイントに返し始めます。バッファが解放されると、パイプラインは HTTP イベントの処理を再開します。
- ソーススレッド - すべての HTTP ソーススレッドがリクエストを実行中で負荷がかかっており、未処理のリクエストキューサイズがリクエストの最大許容数を超えた場合、パイプラインはエラーコード 429 の HTTP ステータス TOO_MANY_REQUESTS をソースエンドポイントに返し始めます。リクエストキューが最大許容キューサイズを下回ると、パイプラインはリクエストの処理を再開します。

OTel ソース

OpenTelemetry ソース (oTel ログ、OTel メトリクス、OTel トレース) を使用するパイプラインのバッファーがいっぱいになると、パイプラインはエラーコード 408 の HTTP REQUEST_TIMEOUT ステータスをソースエンドポイントに返し始めます。バッファが解放されると、パイプラインはイベントの処理を再開します。

S3 ソース

S3 ソースを使用するパイプラインのバッファがいっぱいになると、パイプラインは SQS 通知の処理を停止します。バッファが解放されると、パイプラインは通知の処理を再開します。

シンクがダウンしているか、データを取り込めず、 end-to-end ソースの確認が有効になっている場合、パイプラインはすべてのシンクから正常に受信確認を受け取るまで SQS 通知の処理を停止します。

Amazon Ingestion OpenSearch パイプラインの作成

パイプラインは、Amazon Ingestion OpenSearch がソース (データの出所) からシンク (データの出所) にデータを移動するために使用するメカニズムです。取り込み OpenSearch では、シンクは常に単一の Amazon OpenSearch Service ドメインになりますが、データのソースは Amazon S3、Fluent Bit、 OpenTelemetry コレクターなどのクライアントである可能性があります。

詳細については、 OpenSearch ドキュメントの「パイプライン」を参照してください。

トピック

- 前提条件と必要なロール
- 必要なアクセス許可
- パイプラインのバージョンの指定
- 取り込みパスの指定
- パイプラインの作成
- パイプラインの作成ステータスの追跡
- ブループリントを使用したパイプラインの作成

前提条件と必要なロール

OpenSearch 取り込みパイプラインを作成するには、次のリソースが必要です。

- Ingestion がシンクに書き込むために引き受ける OpenSearch IAM ロール。このロールの ARN をパイプライン設定に含めます。
- シンクとして機能する OpenSearch サービスドメインまたは OpenSearch サーバーレスコレクション。ドメインに書き込む場合は、 OpenSearch 1.0 以降、または Elasticsearch 7.4 以降を実行

している必要があります。シンクには、IAM パイプラインロールに適切な許可を付与するアクセスポリシーが必要です。

これらのリソースの作成手順については、次の各トピックを参照してください。

- the section called “パイプラインにドメインへのアクセス権を付与する”
- the section called “パイプラインにコレクションへのアクセスを許可する”

> **Note**
>
> きめ細かなアクセスコントロールを使用するドメインに書き込む場合は、追加の手順を実行する必要があります。the section called “ステップ 3: パイプラインロールをマッピングする (きめ細かいアクセスコントロールを使用するドメインについてのみ)” を参照してください。

必要なアクセス許可

OpenSearch 取り込みでは、パイプラインの作成に次の IAM アクセス許可を使用します。

- `osis:CreatePipeline` — パイプラインを作成します。
- `osis:ValidatePipeline` — パイプライン設定が有効かどうかを確認します。
- `iam:PassRole` — パイプラインロールを Ingestion OpenSearch に渡して、ドメインにデータを書き込めるようにします。このアクセス許可は、パイプラインロールリソース (パイプライン設定の `sts_role_arn` オプションで指定する ARN) にある必要があります。また、パイプラインごとに異なるロールを使用する場合は、`*` を使用します。

例えば、次のポリシーは、パイプラインを作成するためのアクセス許可を付与します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:ListPipelineBlueprints",
            "osis:ValidatePipeline"
```

```
            ]
        },
        {
            "Resource":[
                "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
            ],
            "Effect":"Allow",
            "Action":[
                "iam:PassRole"
            ]
        }
    ]
}
```

OpenSearch 取り込みにはosis:Ingest、署名バージョン 4 を使用して署名付きリクエストをパイプラインに送信するために必要な というアクセス許可も含まれています。詳細については、「the section called “取り込みロールの作成”」を参照してください。

> **Note**
>
> また、アカウントで最初にパイプラインを作成するユーザーは、iam:CreateServiceLinkedRole アクションの許可を有している必要があります。詳細については、「パイプラインロールリソース」を参照してください。

各アクセス許可の詳細については、「サービス認証リファレンス」の OpenSearch 「取り込みのアクション、リソース、および条件キー」を参照してください。

パイプラインのバージョンの指定

パイプラインを設定するときは、パイプラインを実行する Data Prepper のメジャーバージョンを指定する必要があります。バージョンを指定するには、パイプライン設定に version オプションを含めます。

```
version: "2"
log-pipeline:
  source:
    ...
```

作成 を選択すると、 OpenSearch Ingestion は指定したメジャーバージョンの利用可能な最新のマイナーバージョンを決定し、そのバージョンでパイプラインをプロビジョニングします。例え

ば、 を指定しversion: "2"、Data Prepper のサポートされている最新バージョンが 2.1.1 の場合、Ingestion OpenSearch はパイプラインをバージョン 2.1.1 でプロビジョニングします。パイプラインで実行されているマイナーバージョンは公開されません。

Data Prepper の新しいメジャーバージョンが利用可能になったときに、パイプラインをアップグレードするには、パイプライン設定を編集して新しいバージョンを指定します。パイプラインを以前のバージョンにダウングレードすることはできません。

> **Note**
>
> OpenSearch 取り込みでは、Data Prepper の新しいバージョンがリリースされるとすぐにはサポートされません。新しいバージョンが公開されてから Ingestion OpenSearch でサポートされるまで、多少の遅延が生じます。さらに、Ingestion OpenSearch は特定のメジャーバージョンまたはマイナーバージョンを完全にサポートしていない場合があります。詳細な一覧については、「the section called “サポートされている Data Prepper のバージョン”」を参照してください。

ブルー/グリーンデプロイを開始するパイプラインに変更を加えるたびに、Ingestion OpenSearch はパイプライン YAML ファイルで現在設定されているメジャーバージョンの最新のマイナーバージョンにアップグレードできます。詳細については、「」を参照してくださいthe section called “パイプライン更新用のブルー/グリーンデプロイ”。パイプライン設定内の versionオプションを明示的に更新しない限り、 OpenSearch 取り込みはパイプラインのメジャーバージョンを変更できません。

取り込みパスの指定

OTel トレースや OTel メトリクスなどのプルベースのソースの場合、Ingestion OpenSearch ではソース設定に追加pathのオプションが必要です。パスは /log/ingest のような文字列で、取り込み用の URI パスを示しています。このパスが、データをパイプラインに送信する際に使用する URI を定義します。

例えば、logs という名前の取り込みパイプラインに次のエントリサブパイプラインを指定したとします。

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

パイプラインにデータを取り込むときは、クライアント設定で次のエンドポイントを指定する必要があります: `https://logs-abcdefgh.us-west-2.osis.amazonaws.com/my/test_path`

パスはスラッシュ (/) から開始し、「-」、「_」、「.」、「/」などの特殊文字や `${pipelineName}` プレースホルダーを含めることができます。`${pipelineName}` (`path: "/${pipelineName}/test_path"` のように) を使用すると、変数は関連するサブパイプラインの名前に置き換えられます。この例では、`https://logs.us-west-2.osis.amazonaws.com/entry-pipeline/test_path` のようになります。

パイプラインの作成

このセクションでは、 OpenSearch サービスコンソールと OpenSearch を使用して取り込みパイプラインを作成する方法について説明します AWS CLI。

コンソール

パイプラインを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールにサインインします。
2. 左側のナビゲーションペインで [パイプライン] をクリックし、次に [パイプラインを作成] をクリックします。
3. パイプラインの名前を入力します。
4. (オプション) [永続的バッファリングを有効にする] を選択します。永続的バッファリングは、複数の AZ にまたがってディスクベースのバッファにデータを保存します。詳細については、「永続的バッファリング」を参照してください。永続バッファを有効にする場合は、 AWS Key Management Service キーを選択してバッファデータを暗号化します。
5. 取り込み OpenSearch コンピューティングユニット (OCUs) でパイプラインの最小容量と最大容量を設定します。詳細については、「the section called “パイプラインのスケーリング”」を参照してください。
6. [パイプライン設定] で、パイプライン設定を YAML 形式で入力します。1 個のパイプライン設定ファイルには、1 ~ 10 個のサブパイプラインを含めることができます。各サブパイプラインは、1 個のソース、0 個以上のプロセッサ、1 個のシンクの組み合わせです。 OpenSearch 取り込みの場合、シンクは常に OpenSearch サービスドメインである必要があります。サポートされているオプションのリストについては、「the section called “サポートされているプラグインとオプション”」を参照してください。

> **Note**
>
> 各サブパイプラインには sts_role_arn オプションと sigv4 オプションを含める必要があります。パイプラインは、ドメインへのリクエストに署名sts_role_arnするために で定義されたロールを引き受けます。詳細については、「the section called “パイプラインにドメインへのアクセス権を付与する”」を参照してください。

次のサンプル設定ファイルは、HTTP ソースプラグインと Grok プラグインを使用して非構造化ログデータを処理し、 OpenSearch それを サービスドメインに送信します。サブパイプラインには、log-pipeline という名前が付けられています。

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log: [ '%{COMMONAPACHELOG}' ]
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://search-my-domain.us-east-1.es.amazonaws.com" ]
        index: "apache_logs"
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/{pipeline-role}"
          region: "us-east-1"
```

> **Note**
>
> YAML パイプライン定義内で複数のシンクを指定する場合、それらはすべて同じ OpenSearch サービスドメインである必要があります。 OpenSearch 取り込みパイプラインは、複数の異なるドメインに書き込むことはできません。

独自のパイプライン設定を作成することもできますが、[ファイルをアップロード] をクリックして、セルフマネージド型の Data Prepper パイプラインの既存設定をインポートすることもできます。または、設定のブループリントを使用することもできます。

7. パイプラインを設定したら、[パイプラインを検証] をクリックして設定が正しいことを確認します。検証に失敗した場合、エラーを修正して検証を再実行します。
8. ネットワーク設定 で、VPC アクセス またはパブリックアクセス を選択します。[パブリックアクセス] を選択した場合は、次のステップに進みます。[VPC アクセス] を選択した場合は、次の設定を行います。

設定	説明
エンドポイント管理	VPC エンドポイントを自分で作成するか、 OpenSearch 取り込みで作成するかを選択します。エンドポイント管理のデフォルトは、Ingestion OpenSearch によって管理されるエンドポイントです。
VPC	使用する仮想プライベートクラウド (VPC) の ID を選択します。VPC とパイプラインは同じ AWS リージョンの中になければなりません。
サブネット	1 つ以上のサブネットを選択します。 OpenSearch サービスは VPC エンドポイントと Elastic Network Interface をサブネットに配置します。
セキュリティグループ	必要なアプリケーションがパイプラインによって公開されるポート (80 または OpenSearch 443) とプロトコル (HTTP または HTTP) の Ingestion パイプラインに到達できるようにする HTTPs1 つ以上選択します。
VPC アタッチメントオプション	ソースがセルフマネージドエンドポイントの場合は、パイプラインを VPC にアタッチします。提供されているデフォルトの CIDR オプションのいずれかを選択するか、カスタム CIDR を使用します。

詳細については、「the section called “パイプラインの VPC アクセスの設定”」を参照してください。

9. (オプション) [タグ] で、1 つ以上のタグ (キーと値のペア) をパイプラインに追加します。詳細については、「the section called “パイプラインのタグ付け”」を参照してください。

10. （オプション) ログ発行オプション で、Amazon CloudWatch Logs へのパイプラインログの発行を有効にします。パイプラインの問題をより簡単にトラブルシューティングできるように、ログの発行を有効にすることをお勧めします。詳細については、「the section called “パイプラインのログのモニタリング”」を参照してください。
11. [Next] (次へ) を選択します。
12. パイプライン設定を確認したら、[作成] をクリックします。

OpenSearch 取り込みは非同期プロセスを実行してパイプラインを構築します。パイプラインのステータスが `Active` になると、データの取り込みを開始できます。

AWS CLI

create-pipeline コマンドは、パイプライン設定を文字列として、またはは .yaml ファイル内で受け入れます。設定を文字列として入力する場合、改行するたびに \n でエスケープする必要があります。例えば、次のようになります: `"log-pipeline:\n source:\n http:\n processor:\n - grok:\n ...`

次のサンプルコマンドは、以下にある設定でパイプラインを作成します。

- 最低 4 つの取り込み OCU、最大 10 個の取り込み OCU
- 仮想プライベートクラウド (VPC) 内でプロビジョニング
- ログ発行の有効化

```
aws osis create-pipeline \
  --pipeline-name my-pipeline \
  --min-units 4 \
  --max-units 10 \
  --log-publishing-options
 IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="MyLogGroup"} \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

OpenSearch 取り込みは非同期プロセスを実行してパイプラインを構築します。パイプラインのステータスが `Active` になると、データの取り込みを開始できます。パイプラインのステータスを確認するには、 GetPipeline コマンドを使用します。

OpenSearch 取り込み API

Ingestion API OpenSearch を使用して Ingestion OpenSearch パイプラインを作成するには、CreatePipelineオペレーションを呼び出します。

パイプラインが正常に作成されたら、クライアントを設定し、 OpenSearch サービスドメインへのデータの取り込みを開始できます。詳細については、「the section called “パイプライン統合を操作する”」を参照してください。

パイプラインの作成ステータスの追跡

Ingestion OpenSearch がパイプラインをプロビジョニングし、データを取り込む準備をするときに、パイプラインのステータスを追跡できます。

コンソール

パイプラインを最初に作成した後、Ingestion OpenSearch がデータを取り込む準備をするときに、パイプラインは複数の段階を経ます。パイプライン作成のさまざまな段階を確認するには、パイプライン名をクリックして[パイプラインの設定] ページを表示します。[ステータス] で、[詳細を表示] をクリックします。

パイプラインは、データの取り込みが可能になるまでに次の段階を経ます。

- 検証 — パイプライン設定を検証する。この段階が完了すると、すべての検証が成功したことになります。
- 環境の作成 — リソースの準備とプロビジョニングを行います。この段階が完了すると、新しいパイプライン環境が作成されます。
- パイプラインのデプロイ — パイプラインをデプロイします。この段階が完了すると、パイプラインは正常にデプロイされます。
- パイプラインのヘルスチェック — パイプラインのヘルス状態をチェックします。この段階が完了すると、すべてのヘルスチェックを通過したことになります。
- トラフィックの有効化 — パイプラインがデータを取り込むことができるようにします。この段階が完了すると、パイプラインへのデータの取り込みを開始できます。

CLI

get-pipeline-change-progress コマンドを使用して、パイプラインのステータスを確認します。次の AWS CLI リクエストは、 という名前のパイプラインのステータスをチェックしますmy-pipeline。

```
aws osis get-pipeline-change-progress \
    --pipeline-name my-pipeline
```

レスポンス:

```
{
   "ChangeProgressStatuses": {
      "ChangeProgressStages": [
         {
            "Description": "Validating pipeline configuration",
            "LastUpdated": 1.671055851E9,
            "Name": "VALIDATION",
            "Status": "PENDING"
         }
      ],
      "StartTime": 1.671055851E9,
      "Status": "PROCESSING",
      "TotalNumberOfStages": 5
   }
}
```

OpenSearch 取り込み API

Ingestion API を使用してパイプライン作成のステータスを追跡するには、 OpenSearch GetPipelineChangeProgressオペレーションを呼び出します。

ブループリントを使用したパイプラインの作成

パイプラインの定義をゼロから作成するのではなく、設定のブループリントを使用できます。設定のブループリントは、トレース分析や Apache ログなどの一般的な取り込みシナリオ用に事前設定された YAML テンプレートです。設定のブループリントを使用すると、設定をゼロから作成しなくても、パイプラインを簡単にプロビジョニングできます。

コンソール

パイプラインのブループリントを使用するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールにサインインします。
2. 左側のナビゲーションペインで [パイプライン] をクリックし、次に [パイプラインを作成] をクリックします。

3. ブループリントを選択します。パイプライン設定には、選択したユースケースのサブパイプラインが入力されます。
4. コメントアウトされたテキストを確認します。ここには、ブループリント設定についての説明が記載されています。

> **Important**
>
> パイプラインのブループリントは、そのままでは有効になりません。認証に使用する AWS リージョン やロール ARN を指定するなど、いくつかの変更を行う必要があります。そうしないと、パイプラインの検証は失敗します。

CLI

を使用して使用可能なすべてのブループリントのリストを取得するには AWS CLI、 list-pipeline-blueprintsリクエストを送信します。

```
aws osis list-pipeline-blueprints
```

このリクエストは、利用可能なすべてのブループリントのリストを返します。

特定のブループリントに関する詳細情報を取得するには、 get-pipeline-blueprint コマンドを使用します。

```
aws osis get-pipeline-blueprint --blueprint-name AWS-ApacheLogPipeline
```

このリクエストは、Apache ログパイプラインブループリントの内容を返します。

```
{
   "Blueprint":{
      "PipelineConfigurationBody":"###\n  # Limitations: https://docs.aws.amazon.com/
opensearch-service/latest/ingestion/ingestion.html#ingestion-limitations\n###\n###\n
 # apache-log-pipeline:\n    # This pipeline receives logs via http (e.g. FluentBit),
 extracts important values from the logs by matching\n    # the value in the 'log' key
 against the grok common Apache log pattern. The grokked logs are then sent\n    # to
 OpenSearch to an index named 'logs'\n###\n\nversion: \"2\"\napache-log-pipeline:\n
 source:\n    http:\n      # Provide the path for ingestion. ${pipelineName} will be
 replaced with pipeline name configured for this pipeline.\n      # In this case it
 would be \"/apache-log-pipeline/logs\". This will be the FluentBit output URI value.
```

```
\n      path: \"/${pipelineName}/logs\"\n  processor:\n    - grok:\n        match:\n
         log: [ \"%{COMMONAPACHELOG_DATATYPED}\" ]\n  sink:\n    - opensearch:\n
   # Provide an AWS OpenSearch Service domain endpoint\n        # hosts: [ \"https://
search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com\" ]\n
 aws:\n          # Provide a Role ARN with access to the domain. This role should have
 a trust relationship with osis-pipelines.amazonaws.com\n          # sts_role_arn:
 \"arn:aws:iam::123456789012:role/Example-Role\"\n          # Provide the region of the
 domain.\n          # region: \"us-east-1\"\n          # Enable the 'serverless' flag
 if the sink is an Amazon OpenSearch Serverless collection\n          # serverless:
 true\n        index: \"logs\"\n        # Enable the S3 DLQ to capture any failed
 requests in an S3 bucket\n        # dlq:\n          # s3:\n            # Provide an
 S3 bucket\n            # bucket: \"your-dlq-bucket-name\"\n            # Provide a key
 path prefix for the failed requests\n            # key_path_prefix: \"${pipelineName}/
logs/dlq\"\n            # Provide the region of the bucket.\n            # region:
 \"us-east-1\"\n            # Provide a Role ARN with access to the bucket. This role
 should have a trust relationship with osis-pipelines.amazonaws.com\n            #
 sts_role_arn: \"arn:aws:iam::123456789012:role/Example-Role\"\n",
      "BlueprintName":"AWS-ApacheLogPipeline"
   }
}
```

OpenSearch 取り込み API

Ingestion API を使用してパイプラインの設計図に関する情報を取得するには、ListPipelineBlueprintsおよび OpenSearch GetPipelineBlueprintオペレーションを使用します。

Amazon OpenSearch Ingestion パイプラインの表示

Amazon OpenSearch Ingestion パイプラインの詳細は、AWS Management Console、AWS CLI、または OpenSearch Ingestion API を使用して表示できます。

コンソール

パイプラインを表示するには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. 左側のナビゲーションペインで [パイプライン] をクリックします。
3. (オプション) 特定のステータスのパイプラインを表示するには、[任意のステータス] をクリックし、フィルタリングするステータスを選択します。

パイプラインには、次のステータスがあります。

- `Creating` - パイプラインを作成中です。
- `Active` - パイプラインはアクティブで、データを取り込む準備ができています。
- `Updating` - パイプラインを更新中です。
- `Deleting` - パイプラインを削除中です。
- `Create failed` - パイプラインを作成できませんでした。
- `Update failed` - パイプラインを更新できませんでした。
- `Starting` - パイプラインを開始中です。
- `Start failed` - パイプラインを開始できませんでした。
- `Stopping` - パイプラインを停止中です。
- `Stopped` - パイプラインは停止中で、いつでも再開できます。

パイプラインのステータスが `Create failed`、`Creating`、`Deleting`、`Stopped` の場合は、Ingestion OCU の料金は請求されません。

CLI

AWS CLI を使用してパイプラインを表示するには、list-pipelines リクエストを送信します。

```
aws osis list-pipelines
```

リクエストは、既存の全パイプラインのリストを返します。

```
{
    "NextToken": null,
    "Pipelines": [
        {,
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 4,
            "MinUnits": 2,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/log-pipeline",
            "PipelineName": "log-pipeline",
            "Status": "ACTIVE",
            "StatusReason": {
                "Description": "The pipeline is ready to ingest data."
```

```
            }
        },
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 2,
            "MinUnits": 8,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/another-
pipeline",
            "PipelineName": "another-pipeline",
            "Status": "CREATING",
            "StatusReason": {
                "Description": "The pipeline is being created. It is not able to ingest
 data."
            }
        }
    ]
}
```

1 つのパイプラインに関する情報を取得するには、get-pipeline コマンドを使用します。

```
aws osis get-pipeline --pipeline-name "my-pipeline"
```

リクエストは、指定されたパイプラインの設定情報を返します。

```
{
    "Pipeline": {
        "PipelineName": "my-pipeline",
        "PipelineArn": "arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline",
        "MinUnits": 9,
        "MaxUnits": 10,
        "Status": "ACTIVE",
        "StatusReason": {
            "Description": "The pipeline is ready to ingest data."
        },
        "PipelineConfigurationBody": "log-pipeline:\n source:\n http:\n processor:\n
 - grok:\n match:\nlog: [ '%{COMMONAPACHELOG}' ]\n - date:\n from_time_received: true
\n destination: \"@timestamp\"\n  sink:\n - opensearch:\n hosts: [ \"https://search-
mdp-performance-test-duxkb4qnycd63rpy6svmvyvfpi.us-east-1.es.amazonaws.com\" ]\n index:
 \"apache_logs\"\n aws_sts_role_arn: \"arn:aws:iam::123456789012:role/my-domain-role
\"\n  aws_region: \"us-east-1\"\n  aws_sigv4: true",,
        "CreatedAt": "2022-10-01T15:28:05+00:00",
        "LastUpdatedAt": "2022-10-21T21:41:08+00:00",
        "IngestEndpointUrls": [
```

```
            "my-pipeline-123456789012.us-east-1.osis.amazonaws.com"
        ]
    }
}
```

OpenSearch Ingestion API

OpenSearch Ingestion API を使用して OpenSearch Ingestion パイプラインを表示するには、ListPipelines オペレーションと GetPipeline オペレーションを呼び出します。

Amazon OpenSearch インジェストパイプラインの更新

、 AWS CLI、 OpenSearch またはインジェスト API を使用して Amazon インジェストパイプラインを更新できます。 AWS Management Console OpenSearch OpenSearch パイプラインの YAML 設定を更新すると、インジェストによってブルー/グリーンデプロイが開始されます。詳細については、「the section called “パイプライン更新用のブルー/グリーンデプロイ”」を参照してください。

トピック

- 考慮事項
- 必要なアクセス許可
- パイプラインの更新
- パイプライン更新用のブルー/グリーンデプロイ

考慮事項

パイプラインを更新するときは、次の点を考慮します。

- パイプラインの容量制限、ログ公開のオプション、YAML 設定は、編集が可能です。名前やネットワーク設定は、編集できません。
- パイプラインが VPC ドメインシンクに書き込みを行う場合、パイプラインを作成した後に、前に戻ってシンクを別の VPC ドメインに変更することはできません。そのパイプラインを削除し、新しいシンクを使って改めて作成する必要があります。シンクを VPC ドメインからパブリックドメイン、パブリックドメインから VPC ドメイン、パブリックドメインから別のパブリックドメインに、それぞれ切り替えることはできます。
- パイプラインシンクは、 OpenSearch パブリックサービスドメインとサーバーレスコレクションの間でいつでも切り替えることができます。 OpenSearch

- パイプラインの YAML 設定を更新すると、 OpenSearch Ingestion はブルー/グリーンデプロイを開始します。詳細については、「the section called “パイプライン更新用のブルー/グリーンデプロイ”」を参照してください。
- パイプラインの YAML 設定を更新すると、 OpenSearch Ingestion はパイプライン設定で指定されている Data Prepper のメジャーバージョンの、サポートされている最新のマイナーバージョンにパイプラインを自動的にアップグレードします。このプロセスにより、パイプラインは、最新のバグ修正とパフォーマンスの改善が施されて最新状態に保たれます。
- パイプラインは、停止した場合でも更新することができます。

必要なアクセス許可

OpenSearch インジェストでは、次の IAM 権限を使用してパイプラインを更新します。

- `osis:UpdatePipeline` - パイプラインを更新します。
- `osis:ValidatePipeline` — パイプライン設定が有効かどうかを確認します。
- `iam:PassRole`— パイプラインのロールを OpenSearch Ingestion に渡して、ドメインにデータを書き込めるようにします。このアクセス許可は、パイプラインの YAML 設定を更新するときのみ必要で、ログの公開や容量制限など、他の設定を変更する場合は必要ありません。

例えば、次のポリシーではパイプラインを更新するためのアクセス許可を付与しています。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:UpdatePipeline",
            "osis:ValidatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
         ],
         "Effect":"Allow",
         "Action":[
```

```
            "iam:PassRole"
         ]
      }
   ]
}
```

パイプラインの更新

、 AWS CLI、 OpenSearch またはインジェスト API を使用して Amazon インジェストパイプラインを更新できます。 AWS Management Console OpenSearch

コンソール

パイプラインを更新するには

1. https://console.aws.amazon.com/aos/home の Amazon OpenSearch サービスコンソールにサインインします。
2. 左側のナビゲーションペインで [パイプライン] をクリックします。
3. パイプラインを選択し、設定を開きます。パイプラインの容量制限、ログ公開のオプション、YAML 設定は、編集が可能です。名前やネットワーク設定は、編集できません。
4. 変更が完了したら、[保存] を選択します。

CLI

を使用してパイプラインを更新するには AWS CLI、パイプライン更新リクエストを送信してください。次のリクエスト例では、新しい設定ファイルをアップロードし、最小容量と最大容量値を更新しています。

```
aws osis update-pipeline \
  --pipeline-name "my-pipeline" \
  --pipline-configuration-body "file://new-pipeline-config.yaml" \
  --min-units 11 \
  --max-units 18
```

OpenSearch インジェスト API

OpenSearch OpenSearch Ingestion API を使用してインジェストパイプラインを更新するには、オペレーションを呼び出します。UpdatePipeline

パイプライン更新用のブルー/グリーンデプロイ

OpenSearch パイプラインの YAML 設定を更新すると、インジェストはブルー/グリーンデプロイプロセスを開始します。

ブルー/グリーンとは、パイプライン更新用の新しい環境を作成し、これらの更新が完了した後に新しい環境にトラフィックをルーティングする方法のことを指します。この方法では、新しい環境へのデプロイに失敗しても、ダウンタイムを最小限に抑えることができ、元の環境を維持することができます。ブルー/グリーンデプロイ自体はパフォーマンスに影響しませんが、パイプラインの設定が、パフォーマンスを変えるような形で変更されると、パフォーマンスが変わる可能性があります。

OpenSearch 取り込みにより、ブルー/グリーンデプロイ中のauto-scaling がブロックされます。新しいパイプラインにリダイレクトされるまで、古いパイプラインへのトラフィックに対してのみ、引き続き料金が請求されます。トラフィックがリダイレクトされると、新しいパイプラインに対してのみ料金が請求されます。2 つのパイプラインの料金が同時に請求されることはありません。

パイプラインの YAML 設定ファイルを更新すると、 OpenSearch Ingestion はパイプライン設定で指定されている Data Prepper のメジャーバージョンの、サポートされている最新のマイナーバージョンにパイプラインを自動的にアップグレードできます。たとえば、パイプライン設定に OpenSearch Igestion が最初にバージョン 2.1.0 `version: "2"` でパイプラインをプロビジョニングしたとします。バージョン 2.1.1 のサポートが追加され、パイプライン設定を変更すると、 OpenSearch Igestion はパイプラインをバージョン 2.1.1 にアップグレードします。

このプロセスにより、パイプラインは最新のバグ修正とパフォーマンスの向上によって常に最新の状態に保たれます。 OpenSearch `version`パイプライン設定内のオプションを手動で変更しない限り、インジェストではパイプラインのメジャーバージョンを更新できません。

Amazon OpenSearch Ingestion パイプラインの停止と開始

Amazon OpenSearch Ingestion パイプラインの停止と開始は、開発およびテスト環境のコスト管理に役立ちます。パイプラインを使用するたびに設定と終了処理を行うのではなく、一時的に停止することができます。

トピック

- OpenSearch Ingestion パイプラインの停止と開始の概要
- OpenSearch Ingestion パイプラインの停止
- OpenSearch Ingestion パイプラインの開始

OpenSearch Ingestion パイプラインの停止と開始の概要

パイプラインにデータを取り込む必要がないときは、パイプラインを停止することができます。使用する必要が出てきたら、いつでもパイプラインを開始できます。パイプラインを開始および停止することで、開発やテスト、その他、継続的な使用を必要としない類似のアクティビティに使用するパイプラインの、設定と終了の処理を簡素化できます。

パイプラインが停止している間は、Ingestion OCU の時間に対して料金は請求されません。停止したパイプラインは、引き続き更新することが可能で、マイナーバージョンの更新とセキュリティパッチは自動的に受け取られます。

パイプラインの実行は継続しなければならないが、必要以上の容量がある場合は、開始と停止を使用しないでください。パイプラインのコストがかかりすぎる場合、または混雑がさほど深刻でない場合は、最大容量制限を下げることを検討します。詳細については、「the section called “パイプラインのスケーリング”」を参照してください。

OpenSearch Ingestion パイプラインの停止

OpenSearch Ingestion パイプラインを使用したり、管理を実行したりするときは、必ずアクティブなパイプラインから開始します。次に、パイプラインを停止して、その後に、再びパイプラインを開始します。パイプラインが停止している間は、Ingestion OCU の時間に対して料金は請求されません。

コンソール

パイプラインを停止するには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. ナビゲーションペインで [パイプライン] を選択し、パイプラインを選択します。このページで停止オペレーションを実行するか、停止させるパイプラインの詳細ページに移動します。
3. [アクション] で [パイプラインを停止] を選択します。

   パイプラインを停止および開始できない場合は、パイプラインを停止のアクションは使用できません。

AWS CLI

AWS CLI を使用してパイプラインを停止するときは、次のパラメータで stop-pipeline コマンドを呼び出します。

- `--pipeline-name` – パイプラインの名前。

Example

```
aws osis stop-pipeline --pipeline-name my-pipeline
```

OpenSearch Ingestion API

OpenSearch Ingestion API を使用してパイプラインを停止するには、次のパラメータで StopPipeline オペレーションを呼び出します。

- `PipelineName` – パイプラインの名前。

OpenSearch Ingestion パイプラインの開始

OpenSearch Ingeestion パイプラインを開始するときは、必ず、常に停止状態になっているパイプラインから始めます。このパイプラインは、容量制限、ネットワーク設定、ログ公開オプションなどの構成設定が維持されています。

パイプラインを再度開始するときは、数分かかります。

コンソール

パイプラインを開始するには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. ナビゲーションペインで [パイプライン] を選択し、パイプラインを選択します。このページから開始オペレーションを実行するか、開始するパイプラインの詳細ページに移動します。
3. [アクション] で [パイプラインを開始] を選択します。

AWS CLI

AWS CLI を使用してパイプラインを停止するときは、次のパラメータで start-pipeline コマンドを呼び出します。

- `--pipeline-name` – パイプラインの名前。

Example

```
aws osis start-pipeline --pipeline-name my-pipeline
```

OpenSearch Ingestion API

OpenSearch Ingestion API を使用して OpenSearch Ingestion パイプラインを開始するには、次のパラメータで StartPipeline オペレーションを呼び出します。

- `PipelineName` – パイプラインの名前。

Amazon OpenSearch Ingestion パイプラインの削除

Amazon OpenSearch Ingestion パイプラインは、AWS Management Console、AWS CLI、または OpenSearch Ingestion API を使用して削除できます。ステータスが `Creating` または `Updating` の場合、パイプラインは削除できません。

コンソール

パイプラインを削除するには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. 左側のナビゲーションペインで [パイプライン] をクリックします。
3. 削除するパイプラインを選択し、[削除] をクリックします。
4. 削除を確認し、[Delete] (削除) を選択します。

CLI

AWS CLI を使用してパイプラインを削除するには、delete-pipeline リクエストを送信します。

```
aws osis delete-pipeline --pipeline-name "my-pipeline"
```

OpenSearch Ingestion API

OpenSearch Ingestion API を使用して OpenSearch Ingestion パイプラインを削除するには、次のパラメータを指定して DeletePipeline オペレーションを呼び出します。

- `PipelineName` – パイプラインの名前。

Amazon Ingestion OpenSearch パイプラインでサポートされているプラグインとオプション

Amazon Ingestion OpenSearch は、オープンソースの Data Prepper と比較して、ソース、プロセッサ、シンクのサブセットをサポートしています。さらに、サポートされている各プラグインで使用可能なオプションに Ingestion OpenSearch が課す制約がいくつかあります。以下のセクションでは、Ingestion OpenSearch がサポートするプラグインと関連するオプションについて説明します。

> **Note**
>
> OpenSearch 取り込みでは、デフォルトのバッファが自動的に設定されるため、バッファプラグインはサポートされていません。パイプラインの設定にバッファーを含めると、検証エラーが発生します。

トピック

サポートされているプラグイン

OpenSearch Ingestion は、以下の Data Prepper プラグインをサポートしています。

ソース:

- Amazon DocumentDB

- DynamoDB
- OpenSearch

- HTTP
- Kafka
- OTel logs
- OTel metrics
- OTel trace
- S3

プロセッサ:

- Aggregate
- Anomaly detector
- CSV
- 日付
- 解凍
- 解体
- Drop events
- Geo IP
- Grok
- Key value
- リストにマップする
- ミューテーションイベント (プロセッサのシリーズ)
- ミューテーション文字列 (プロセッサのシリーズ)
- Obfuscate
- OTel metrics
- OTel trace group
- OTel trace
- Ion を解析する

- Parse JSON
- XML を解析する
- エントリの選択
- Service-map
- Trace peer forwarder
- 切り捨て
- User agent

シンク:

- OpenSearch (OpenSearch Service、 OpenSearch Serverless、Elasticsearch 6.8 以降をサポート)
- S3

シンクコーデック:

- Avro
- NDJSON
- JSON
- Parquet

ステートレスプロセッサとステートフルプロセッサ

ステートレスプロセッサは変換やフィルタリングなどのオペレーションを実行し、ステートフルプロセッサは前回の実行結果を記憶する集計などのオペレーションを実行します。Ingestion OpenSearch はステートフルプロセッサの Aggregate と Service-map をサポートします。その他のサポートされているプロセッサは、すべてステートレスです。

ステートレスプロセッサのみを含むパイプラインの場合、最大容量制限は 96 Ingestion OCUs。パイプラインにステートフルプロセッサが含まれている場合、最大容量制限は 48 Ingestion OCUs。ただし、パイプラインで永続バッファリングが有効になっている場合、ステートレスプロセッサのみを搭載した最大 384 個の Ingestion OCUs、またはステートフルプロセッサが含まれている場合は 192 個の Ingestion OCUs を持つことができます。詳細については、「the section called “パイプラインのスケーリング”」を参照してください。

E nd-to-end 確認はステートレスプロセッサでのみサポートされます。詳細については、「the section called “E nd-to-end 確認応答”」を参照してください。

設定の要件と制限

以下で特に指定されていない限り、上記のサポートされているプラグインの Data Prepper 設定リファレンスで説明されているすべてのオプションが Ingestion OpenSearch パイプラインで許可されます。以下のセクションでは、Ingestion OpenSearch が特定のプラグインオプションに配置する制約について説明します。

> **Note**
>
> OpenSearch 取り込みでは、デフォルトのバッファが自動的に設定されるため、バッファプラグインはサポートされていません。パイプラインの設定にバッファーを含めると、検証エラーが発生します。

や など、多くのオプションは Ingestion OpenSearch によって内部的に設定`authentication`および管理されます`acm_certificate_arn`。`thread_count` や `request_timeout` など、その他のオプションは、手動で変更するとパフォーマンスに影響します。したがって、これらの値は、パイプラインの最適なパフォーマンスを確保するために内部で設定されます。

最後に、 `ism_policy_file`や OpenSearch などの一部のオプションは、オープンソースの Data Prepper で実行されるとローカルファイルになるため`sink_template`、取り込みに渡すことはできません。こうした値はサポートされていません。

トピック

- 一般的なパイプラインのオプション
- Grok プロセッサ
- HTTP ソース
- OpenSearch シンク
- OTel メトリクスソース、OTel トレースソース、OTel ログソース
- OTel トレースグループプロセッサ
- OTel トレースプロセッサ
- サービスマッププロセッサ

- S3 ソース

一般的なパイプラインのオプション

次の一般的なパイプラインオプションは Ingestion OpenSearch によって設定され、パイプライン設定ではサポートされていません。

- `workers`
- `delay`

Grok プロセッサ

以下の Grok プロセッサオプションはサポートされていません。

- `patterns_directories`
- `patterns_files_glob`

HTTP ソース

HTTP ソースプラグインには、次の要件および制限があります。

- `path` のオプションは必須です。このパスは `/log/ingest` のような文字列で、ログを取り込むための URI パスを表しています。このパスが、データをパイプラインに送信する際に使用する URI を定義します。例えば `https://log-pipeline.us-west-2.osis.amazonaws.com/log/ingest` です。パスは、スラッシュ (/) から開始し、「-」、「_」、「.」、「/」などの特殊文字や `${pipelineName}` プレースホルダーを含めることができます。
- 次の HTTP ソースオプションは Ingestion OpenSearch によって設定され、パイプライン設定ではサポートされていません。
  - `port`
  - `ssl`
  - `ssl_key_file`
  - `ssl_certificate_file`
  - `aws_region`
  - `authentication`
  - `unauthenticated_health_check`

- use_acm_certificate_for_ssl
- thread_count
- request_timeout
- max_connection_count
- max_pending_requests
- health_check_service
- acm_private_key_password
- acm_certificate_timeout_millis
- acm_certificate_arn

OpenSearch シンク

OpenSearch シンクプラグインには、次の要件と制限があります。

- aws のオプションは必須であり、次のオプションを含んでいる必要があります。
  - sts_role_arn
  - region
  - hosts
  - serverless (シンクが OpenSearch サーバーレスコレクションの場合)
- sts_role_arn のオプションは、YAML 定義ファイル内で、各シンクで同じロールを指している必要があります。
- hosts オプションでは、 OpenSearch サービスドメインエンドポイントまたは OpenSearch サーバーレスコレクションエンドポイントを指定する必要があります。YAML 定義ファイル内のホストは、すべて同じエンドポイントを指している必要があります。ドメインのカスタムエンドポイントは指定できません。こちらは、標準エンドポイントでなければなりません。
- hosts のオプションがサーバーレスコレクションのエンドポイントである場合は、serverless のオプションを true に設定する必要があります。さらに、YAML 定義ファイルに index_type のオプションが含まれている場合は、これを management_disabled に設定する必要があります。そうしないと検証に失敗します。
- 次のオプションはサポートされていません。
  - username
  - password
  - cert

- proxy
- dlq_file - 失敗したイベントをデッドレターキュー (DLQ) にオフロードする場合は、dlq オプションを使用して S3 バケットを指定します。
- ism_policy_file
- socket_timeout
- template_file
- insecure
- bulk_size

OTel メトリクスソース、OTel トレースソース、OTel ログソース

OTel メトリクスソース、OTel トレースソース、OTel ログソースプラグインには、次の要件および制限があります。

- path のオプションは必須です。このパスは /log/ingest のような文字列で、ログを取り込むための URI パスを表しています。このパスが、データをパイプラインに送信する際に使用する URI を定義します。例えば https://log-pipeline.us-west-2.osis.amazonaws.com*/log/ingest* です。パスは、スラッシュ (/) から開始し、「-」、「_」、「.」、「/」などの特殊文字や ${pipelineName} プレースホルダーを含めることができます。
- 以下のオプションは Ingestion OpenSearch によって設定され、パイプライン設定ではサポートされていません。
  - port
  - ssl
  - sslKeyFile
  - sslKeyCertChainFile
  - authentication
  - unauthenticated_health_check
  - useAcmCertForSSL
  - unframed_requests
  - proto_reflection_service
  - thread_count
  - request_timeout
  - max_connection_count

- acmPrivateKeyPassword
- acmCertIssueTimeOutMillis
- health_check_service
- acmCertificateArn
- awsRegion

OTel トレースグループプロセッサ

OTel トレースグループプロセッサには、次の要件および制限があります。

- aws のオプションは必須であり、次のオプションを含んでいる必要があります。
  - sts_role_arn
  - region
  - hosts
- sts_role_arn オプションは、 OpenSearch シンク設定で指定したパイプラインロールと同じロールを指定します。
- username、password、cert、insecure のオプションはサポートされていません。
- aws_sigv4 のオプションは必須であり、true に設定する必要があります。
- OpenSearch シンクプラグイン内の serverlessオプションはサポートされていません。Otel トレースグループプロセッサは現在、 OpenSearch サーバーレスコレクションでは動作しません。
- パイプライン設定本体内の otel_trace_group プロセッサの数は、8 を超えることはできません。

OTel トレースプロセッサ

OTel トレースプロセッサには、次の要件および制限があります。

- trace_flush_interval オプションの値は、300 秒を超えることはできません。

サービスマッププロセッサ

サービスマッププロセッサには、次の要件および制限があります。

- window_duration オプションの値は、300 秒を超えることはできません。

S3 ソース

S3 ソースのプラグインには、次の要件および制限があります。

- aws のオプションは必須であり、region と sts_role_arn のオプションを含んでいる必要があります。
- records_to_accumulate オプションの値は、200 を超えることはできません。
- maximum_messages オプションの値は、10 を超えることはできません。
- 指定する場合は、disable_bucket_ownership_validation オプションを false に設定する必要があります。
- 指定する場合は、input_serialization オプションを parquet に設定する必要があります。

Amazon Ingestion OpenSearch パイプライン統合の使用

Amazon OpenSearch Ingestion パイプラインにデータを正常に取り込むには、パイプラインエンドポイントにデータを送信するようにクライアントアプリケーション (ソース) を設定する必要があります。ソースは、Fluent Bit ログ、 OpenTelemetryコレクター、または単純な S3 バケットなどのクライアントです。正確な設定は、クライアントごとに異なります。

ソース設定 (OpenSearch サービスドメインまたは OpenSearch サーバーレスコレクションに直接データを送信する場合と比較して) における重要な違いは、パイプラインエンドポイントである必要がある AWS サービス名 (osis) とホストエンドポイントです。

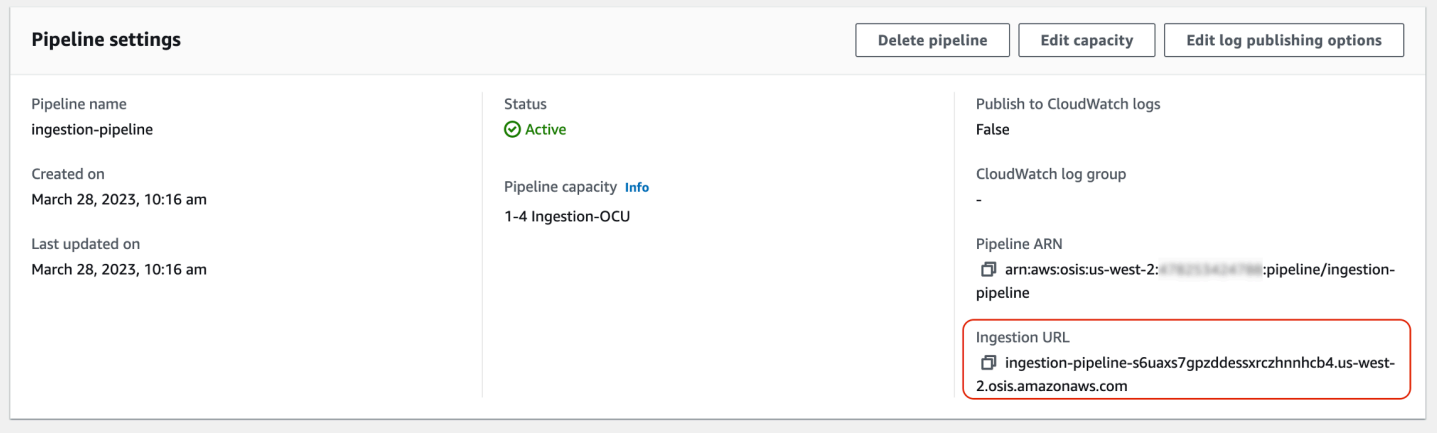
トピック

- 取り込みエンドポイントの構築
- 取り込みロールの作成
- Amazon DynamoDB OpenSearch での取り込みパイプラインの使用
- Amazon DocumentDB OpenSearch での取り込みパイプラインの使用
- Confluent Kafka クラウドでの OpenSearch Ingestion パイプラインの使用
- での OpenSearch 取り込みパイプラインの使用 Amazon Managed Streaming for Apache Kafka
- Amazon S3 OpenSearch での取り込みパイプラインの使用
- Amazon Security Lake OpenSearch での取り込みパイプラインの使用
- Fluent Bit OpenSearch での取り込みパイプラインの使用
- Fluentd OpenSearch での取り込みパイプラインの使用

- OpenTelemetry Collector OpenSearch での取り込みパイプラインの使用
- 次のステップ

取り込みエンドポイントの構築

データをパイプラインに取り込むには、取り込みエンドポイントにデータを送信します。取り込み URL を見つけるには、[パイプラインの設定] ページに移動して [取り込み URL] をコピーします。



OTel trace や OTel metrics などのプルベースのソースの完全な取り込みエンドポイントを構築するには、パイプライン設定からの取り込みパスを取り込み URL に追加します。

例えば、パイプライン設定に次のような取り込みパスがあるとします。

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

クライアント設定で指定した完全な取り込みエンドポイントは、次の形式になります: `https://ingestion-pipeline-abcdefg.us-west-2.osis.amazonaws.com/my/test_path`。

詳細については、「the section called “取り込みパスの指定”」を参照してください。

取り込みロールの作成

取り込みへのすべてのリクエストは、署名バージョン OpenSearch 4 で署名する必要があります。https://docs.aws.amazon.com/general/latest/gr/signature-version-4.html少なくとも、リクエストに署名するロールには、 `osis:Ingest`アクションのアクセス許可を付与する必要があります。これにより、 OpenSearch 取り込みパイプラインにデータを送信できるようになります。

例えば、次の AWS Identity and Access Management (IAM) ポリシーでは、対応するロールが単一のパイプラインにデータを送信することを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-east-1:{account-id}:pipeline/pipeline-name"
    }
  ]
}
```

> **Note**
>
> すべてのパイプラインにロールを使用するには、Resource 要素内の ARN をワイルドカード (*) に置き換えます。

クロスアカウント取り込みアクセスの提供

> **Note**
>
> クロスアカウント取り込みアクセスを提供できるのは、パブリックパイプラインに対してのみであり、VPC パイプラインには提供できません。

ソースアプリケーションを格納するアカウントなど AWS アカウント、別の からパイプラインにデータを取り込む必要がある場合があります。パイプラインに書き込むプリンシパルが、パイプライン自体とは別のアカウントにある場合は、パイプラインにデータを取り込むために別の IAM ロールを信頼するようにプリンシパルを設定する必要があります。

クロスアカウント取り込みアクセス許可の設定するには

1. パイプライン AWS アカウント と同じ 内に、 アクセスosis:Ingest許可 (前のセクションで説明) を持つ取り込みロールを作成します。手順については、「IAM ロールの作成」を参照してください。

2. 取り込みロールに信頼ポリシーをアタッチして、別のアカウントのプリンシパルがそのポリシーを引き受けることができるようにします。

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::{external-account-id}:root"
      },
     "Action": "sts:AssumeRole"
  }]
}
```

3. もう一方のアカウントで、取り込みロールを引き受けるようにクライアントアプリケーション (例: Fluent Bit) を設定します。これを機能させるには、アプリケーションアカウントは、アプリケーションユーザーまたはロールに取り込みロールを引き受ける許可を付与する必要があります。

   次のアイデンティティベースポリシーの例では、アタッチされたプリンシパルがパイプラインアカウントから ingestion-role を引き受けることを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::{account-id}:role/ingestion-role"
    }
  ]
}
```

その後、クライアントアプリケーションは AssumeRoleオペレーションを使用してデータを引き受けingestion-role、関連するパイプラインに取り込むことができます。

Amazon DynamoDB OpenSearch での取り込みパイプラインの使用

DynamoDB で Ingestion OpenSearch パイプラインを使用して、DynamoDB テーブルイベント (作成、更新、削除など) を Amazon OpenSearch Service ドメインとコレクションにストリーミングで

きます。 OpenSearch Ingestion パイプラインには、変更データキャプチャ (CDC) インフラストラクチャが組み込まれており、DynamoDB テーブルからデータを継続的にストリーミングするための、高スケールで低レイテンシーの手段を提供します。

データを処理するソースとして DynamoDB を使用するには、完全な初期スナップショットを使用する方法と使用しない方法の 2 つがあります。

完全な初期スナップショットは、DynamoDB がpoint-in-time リカバリ (PITR) 機能を使用して取得するテーブルのバックアップです。DynamoDB はこのスナップショットを Amazon S3 にアップロードします。そこから、取り込みパイプラインはドメイン内の 1 OpenSearch つのインデックスに送信するか、ドメイン内の複数のインデックスにパーティション化します。DynamoDB 内のデータを OpenSearch 一貫性に保つために、パイプラインは DynamoDB テーブル内のすべての作成、更新、削除イベントを OpenSearch 、インデックスまたはインデックスに保存されたドキュメントと同期します。

完全な初期スナップショットを使用すると、 OpenSearch 取り込みパイプラインは最初にスナップショットを取り込んでから、DynamoDB Streams からのデータの読み取りを開始します。最終的には、DynamoDB と の間のほぼリアルタイムのデータ整合性が追いついて維持されます OpenSearch。このオプションを選択する場合は、テーブルで PITR と DynamoDB ストリームの両方を有効にする必要があります。

Ingestion と OpenSearch DynamoDB の統合を使用して、スナップショットなしでイベントをストリーミングすることもできます。他のメカニズムからの完全なスナップショットが既にある場合、または DynamoDB Streams を使用して DynamoDB テーブルから現在のイベントをストリーミングするだけの場合は、このオプションを選択します。このオプションを選択した場合、必要なのは、テーブルで DynamoDB ストリームを有効にすることだけです。

この統合の詳細については、「 デベロッパーガイド」の「Amazon OpenSearch Service との DynamoDB ゼロ ETL 統合」を参照してください。 Amazon DynamoDB

トピック

- 前提条件
- ステップ 1: パイプラインロールを設定する
- ステップ 2: パイプラインを作成する
- データ整合性
- データ型のマッピング
- 制限事項

前提条件

パイプラインを設定するには、DynamoDB Streams が有効になっている DynamoDB テーブルが必要です。ストリームでは NEW_IMAGE ストリームビュータイプを使用する必要があります。ただし、このストリームビュータイプがユースケースに適合するNEW_AND_OLD_IMAGES場合、 OpenSearch 取り込みパイプラインは でイベントをストリーミングすることもできます。

スナップショットを使用している場合は、テーブルで point-in-time 復旧を有効にする必要もあります。詳細については、「Amazon DynamoDB デベロッパーガイド」の「テーブルの作成」、point-in-time 「復旧の有効化」、「ストリームの有効化」を参照してください。

ステップ 1: パイプラインロールを設定する

DynamoDB テーブルを設定したら、パイプライン設定で使用するパイプラインロールを設定し、そのロールに次の DynamoDB 許可を追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowRunExportJob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:ExportTableToPointInTime"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table"
            ]
        },
        {
            "Sid": "allowCheckExportjob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeExport"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table/export/*"
            ]
        },
        {
```

```
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords",
                "dynamodb:GetShardIterator"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table/stream/*"
            ]
        },
        {
            "Sid": "allowReadAndWriteToS3ForExport",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/{exportPath}/*"
            ]
        }
    ]
}
```

AWS KMS カスタマーマネージドキーを使用して、エクスポートデータファイルを暗号化することもできます。エクスポートされたオブジェクトを復号するには、パイプラインのエクスポート設定において、次の形式でキー ID として s3_sse_kms_key_id を指定します: arn:aws:kms:*us-west-2*:*{account-id}*:key/*my-key-id*。次のポリシーには、カスタマーマネージドキーを使用するために必要なアクセス許可が含まれています。

```
{
    "Sid": "allowUseOfCustomManagedKey",
    "Effect": "Allow",
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": arn:aws:kms:us-west-2:{account-id}:key/my-key-id
}
```

ステップ 2: パイプラインを作成する

その後、DynamoDB OpenSearch をソースとして指定する Ingestion パイプラインを次のように設定できます。このサンプルパイプラインは、PITR スナップショットを使用して table-a からデータを取り込み、続いて DynamoDB Streams からイベントを取り込みます。開始位置が LATEST であることは、パイプラインが DynamoDB Streams から最新のデータを読み取る必要があることを示します。

```
version: "2"
cdc-pipeline:
  source:
    dynamodb:
      tables:
      - table_arn: "arn:aws:dynamodb:us-west-2:{account-id}:table/table-a"
        export:
          s3_bucket: "my-bucket"
          s3_prefix: "export/"
        stream:
          start_position: "LATEST"
      aws:
        region: "us-west-2"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${getMetadata(\"table_name\")}"
      index_type: custom
      normalize_index: true
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
```

事前設定された DynamoDB ブループリントを使用して、このパイプラインを作成できます。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

データ整合性

OpenSearch 取り込みでは、データの耐久性を確保するため end-to-end の承認がサポートされています。パイプラインがスナップショットまたはストリームを読み取る際に、並列処理用のパーティ

ションが動的に作成されます。パイプラインは、 OpenSearch ドメインまたはコレクション内のすべてのレコードを取り込んだ後に確認を受け取ると、パーティションを完了としてマークします。

OpenSearch Serverless 検索コレクションに取り込む場合は、パイプラインでドキュメント ID を生成できます。 OpenSearch Serverless 時系列コレクションに取り込む場合は、パイプラインがドキュメント ID を生成しないことに注意してください。

Ingestion OpenSearch パイプラインは、受信イベントアクションを対応する一括インデックス作成アクションにマッピングして、ドキュメントの取り込みにも役立ちます。これによりデータの整合性が保たれるため、DynamoDB のすべてのデータ変更が、 の対応するドキュメントの変更と照合されます OpenSearch。

データ型のマッピング

OpenSearch サービスは、各受信ドキュメントのデータ型を DynamoDB の対応するデータ型に動的にマッピングします。次の表は、 OpenSearch サービスがさまざまなデータ型を自動的にマッピングする方法を示しています。

データ型	OpenSearch	DynamoDB
数	OpenSearch は数値データを自動的にマッピングします。数値が整数の場合、はそれを長い値として OpenSearch マッピングします。数値が小数の場合、 はそれを浮動小数点値として OpenSearch マッピングします。 OpenSearch は、最初に送信されたドキュメントに基づいて、さまざまな属性を動的にマッピングします。DynamoDB の同じ属性についてデータ型が混在している場合 (整数と小数の両方など)、マッピングは失敗する可能性があります。 例えば、最初のドキュメントに整数の属性があり、後のドキュメントに小数と同じ属性がある場合、 は 2 番目のドキュメントの取り込みに OpenSearch 失敗します。このような場合は、次のような明	DynamoDB は数値をサポートしています。

データ型	OpenSearch	DynamoDB
	示的なマッピングテンプレートを提供する必要があります: `{ "template": { "mappings": { "properties": { "MixedNumberAttribute": { "type": "float" } } } } }` 倍精度が必要な場合は、文字列型のフィールドマッピングを使用します。で 38 桁の精度をサポートする同等の数値型はありません OpenSearch。	
数値セット	OpenSearch は、数値セットを長い値または浮動小数点値の配列に自動的にマッピングします。スカラー数値の場合と同様、これは、取り込まれた最初の数値が整数または小数のいずれであるかによって異なります。スカラー文字列をマッピングするのと同じ方法で、数値セットのマッピングを提供できます。	DynamoDB は、数値のセットを表す型をサポートしています。

データ型	OpenSearch	DynamoDB
文字列	OpenSearch は文字列値をテキストとして自動的にマッピングします。状況によっては (列挙型の値など)、キーワード型にマッピングできます。 次の例は、 という名前の DynamoDB 属性を OpenSearch キーワードPartTypeにマッピングする方法を示しています。 `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	DynamoDB は文字列をサポートします。
文字列セット	OpenSearch は、文字列セットを文字列の配列に自動的にマッピングします。スカラー文字列をマッピングするのと同じ方法で、文字列セットのマッピングを提供できます。	DynamoDB は、文字列のセットを表す型をサポートしています。

データ型	OpenSearch	DynamoDB
バイナリ	OpenSearch はバイナリデータをテキストとして自動的にマッピングします。マッピングを指定して、これらを のバイナリフィールドとして記述できます OpenSearch。 次の例は、 という名前の DynamoDB 属性`ImageData` を OpenSearch バイナリフィールドにマッピングする方法を示しています。 `{ "template": { "mappings": { "properties": { "ImageData": { "type": "binary" } } } } }`	DynamoDB はバイナリ型の属性をサポートしています。
バイナリセット	OpenSearch は、バイナリセットをテキストとしてバイナリデータの配列に自動的にマッピングします。スカラーバイナリをマッピングするのと同じ方法で、数値セットのマッピングを提供できます。	DynamoDB は、バイナリ値のセットを表す型をサポートしています。
ブール値	OpenSearch は、DynamoDB ブール型を OpenSearch ブール型にマッピングします。	DynamoDB は、ブール型属性をサポートします。

データ型	OpenSearch	DynamoDB
Null	OpenSearch は、DynamoDB null タイプのドキュメントを取り込むことができます。値は null 値としてドキュメントに保存されます。この型にはマッピングがなく、このフィールドにはインデックスが付けられず、検索もできません。 NULL 型に同じ属性名が使用され、後で文字列などの異なる型に変更されると、は最初の NULL 以外の値の動的マッピング OpenSearch を作成します。後続の値は引き続き DynamoDB の null 値にすることができます。	DynamoDB は、null 型属性をサポートします。

データ型	OpenSearch	DynamoDB
マッピング	OpenSearch は、DynamoDB マップ属性をネストされたフィールドにマッピングします。同じマッピングがネストされたフィールド内でも適用されます。 次の例では、ネストされたフィールドの文字列を のキーワードタイプにマッピングします OpenSearch。 `{ "template": { "mappings": { "properties": { "AdditionalDescriptions": { "properties": { "PartType": { "type": "keyword" } } } } } } }`	DynamoDB は、マッピング型属性をサポートします。

データ型	OpenSearch	DynamoDB
リスト	OpenSearch は、リストの内容に応じて、DynamoDB リストに異なる結果を提供します。 リストに同じタイプのスカラー型がすべて含まれている場合 (すべての文字列のリストなど)、 はそのタイプの配列としてリストを取り OpenSearch 込みます。これは、文字列、数値、ブール型、および null 型について機能します。これらの各型についての制限は、その型のスカラーについての制限と同じです。 また、マップに使用するのと同じマッピングを使用して、マップのリストのマッピングを提供することもできます。 混合型のリストを提供することはできません。	DynamoDB は、リスト型属性をサポートします。

データ型	OpenSearch	DynamoDB
設定	OpenSearch は、セットの内容に応じて、DynamoDB セットに対して異なる結果を提供します。 セットに同じタイプのスカラー型がすべて含まれている場合 (例えば、すべての文字列のセット) 、 はそのセットをその型の配列として OpenSearch 取り込みます。これは、文字列、数値、ブール型、および null 型について機能します。これらの各型についての制限は、その型のスカラーについての制限と同じです。 また、マップに使用するのと同じマッピングを使用して、マップのセットのマッピングを提供することもできます。 混合型のセットを提供することはできません。	DynamoDB は、セットを表す型をサポートします。

取り込みパイプラインでデッドレターキュー (DLQ) OpenSearch を設定することをお勧めします。キューを設定した場合、動的マッピングの失敗により取り込むことができない失敗したドキュメントはすべて、 OpenSearch サービスからキューに送信されます。

自動マッピングが失敗した場合は、パイプライン設定で `template_type` と `template_content` を使用して明示的なマッピングルールを定義できます。あるいは、パイプラインを開始する前に、検索ドメインまたはコレクションにマッピングテンプレートを直接作成することもできます。

制限事項

DynamoDB OpenSearch の取り込みパイプラインを設定するときは、次の制限事項を考慮してください。

- DynamoDB との OpenSearch Ingestion 統合は現在、クロスリージョン取り込みをサポートしていません。DynamoDB テーブルと Ingestion OpenSearch パイプラインは同じ にある必要があります AWS リージョン。

- DynamoDB テーブルと Ingestion OpenSearch パイプラインは同じ にある必要があります AWS アカウント。
- 取り込みパイプラインは、ソースとして OpenSearch 1 つの DynamoDB テーブルのみをサポートします。
- DynamoDB Streams は、最大 24 時間ログにデータを格納します。大きなテーブルの最初のスナップショットからの取り込みに 24 時間以上かかる場合、初期データの一部が失われます。このデータ損失を軽減するには、テーブルのサイズを見積もり、適切な Ingestion OpenSearch パイプラインのコンピューティングユニットを設定します。

Amazon DocumentDB OpenSearch での取り込みパイプラインの使用

Amazon DocumentDB OpenSearch で Ingestion パイプラインを使用して、ドキュメントの変更 (作成、更新、削除など) を Amazon OpenSearch Service ドメインとコレクションにストリーミングできます。 OpenSearch Ingestion パイプラインは、Amazon DocumentDB クラスターで利用可能な場合は変更データキャプチャ (CDC) メカニズム、または API ポーリングを活用して、Amazon DocumentDB クラスターからデータを継続的にストリーミングするための高スケールで低レイテンシーの方法を提供できます。

Amazon DocumentDB をソースとして使用してデータを処理する方法は 2 つあります。つまり、完全な初期スナップショット がある場合とない場合です。

完全な初期スナップショットは、Amazon DocumentDB コレクション全体の一括クエリです。Amazon DocumentDB は、このスナップショットを Amazon S3 にアップロードします。そこから、取り込みパイプラインはドメイン内の 1 OpenSearch つのインデックスに送信するか、ドメイン内の複数のインデックスにパーティション化します。Amazon DocumentDB 内のデータを OpenSearch 整合性に保つために、パイプラインは Amazon DocumentDB コレクション内のすべての作成、更新、削除イベントを OpenSearch 、インデックスまたはインデックスに保存されたドキュメントと同期します。

完全な初期スナップショットを使用する場合、 OpenSearch 取り込みパイプラインはまずスナップショットを取り込んでから、Amazon DocumentDB 変更ストリームからのデータの読み取りを開始します。最終的には、Amazon DocumentDB と の間のほぼリアルタイムのデータ整合性が追いついて維持されます OpenSearch。

Amazon DocumentDB OpenSearch との Ingestion 統合を使用して、スナップショットなしでイベントをストリーミングすることもできます。他のメカニズムからの完全なスナップショットが既にある場合、または変更ストリームを使用して Amazon DocumentDB コレクションから現在のイベントをストリーミングする場合は、このオプションを選択します。

パイプライン設定で でストリームを有効にする場合は、これらのオプションの両方で Amazon DocumentDB コレクションで変更ストリームを有効にする必要があります。 Amazon DocumentDB 全ロードまたはエクスポートのみを使用する場合は、変更ストリームを有効にする必要はありません。

前提条件

OpenSearch 取り込みパイプラインを作成する前に、次のステップを実行します。

1. Amazon DocumentDB デベロッパーガイド」のAmazon DocumentDB クラスターを作成する」の手順に従って、データを読み取るアクセス許可を持つ Amazon DocumentDB クラスターを作成します。 Amazon DocumentDB CDC インフラストラクチャを使用する場合は、変更ストリームを発行するように Amazon DocumentDB クラスターを設定してください。
2. を使用して Amazon DocumentDB クラスターで認証を設定します AWS Secrets Manager。Amazon DocumentDB のパスワードの自動ローテーション」の手順に従って、シークレットのローテーションを有効にします。詳細については、Amazon DocumentDB でのロールベースのアクセス制御とセキュリティを使用したデータベースアクセス」を参照してください。
3. 変更ストリームを使用して Amazon DocumentDB コレクションのデータ変更をサブスクライブする場合は、 `change_stream_log_retention_duration`パラメータを使用して保持期間を最大 7 日間に延長することで、データ損失を回避します。変更ストリームイベントは、イベントが記録されてからデフォルトで 3 時間保存されます。これは、大規模なコレクションでは十分な時間ではありません。変更ストリームの保持期間を変更するには、「変更ストリームログの保持期間の変更」を参照してください。
4. OpenSearch サービスドメインまたは OpenSearch サーバーレスコレクションを作成します。詳細については、 OpenSearch 「サービスドメインの作成」および「コレクションの作成」を参照してください。
5. リソースベースのポリシーをドメインにアタッチするか、コレクションにデータアクセスポリシーをアタッチします。これらのアクセスポリシーにより、Ingestion OpenSearch は Amazon DocumentDB クラスターからドメインまたはコレクションにデータを書き込むことができます。

   次のサンプルドメインアクセスポリシーでは、次のステップで作成するパイプラインロールがドメインにデータを書き込むことを許可します。必ず独自の ARN で `resource` を更新してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{pipeline-account-id}:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:{region}:{account-id}:domain/domain-name"
      ]
    }
  ]
}
```

コレクションまたはドメインへの書き込みデータにアクセスするための正しいアクセス許可を持つ IAM ロールを作成するには、「ドメインに必要なアクセス許可」および「コレクションに必要なアクセス許可」を参照してください。

ステップ 1: パイプラインロールを設定する

Amazon DocumentDB パイプラインの前提条件を設定したら、パイプライン設定で使用するパイプラインロールを設定し、ロールに次の Amazon DocumentDB アクセス許可を追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowS3ListObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::{s3_bucket}"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": "{s3_prefix}/*"
                }
            }
        },
```

```
        {
            "Sid": "allowReadAndWriteToS3ForExportStream",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::{s3_bucket}/{s3_prefix}/*"
            ]
        },
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": ["arn:aws:secretsmanager:{region}:{account-id}:secret:secret-
name"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:{account-id}:network-interface/*",
                "arn:aws:ec2:*:{account-id}:subnet/*",
                "arn:aws:ec2:*:{account-id}:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
```

```
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals":
                    {
                        "aws:RequestTag/OSISManaged": "true"
                    }
            }
        }
    ]
}
```

パイプラインはこれらのアクセス許可を使用して VPC 内のネットワークインターフェイスを作成および削除するため、取り込みパイプラインの作成に使用する OpenSearch IAM ロールに対して上記の Amazon EC2 アクセス許可を指定する必要があります。パイプラインは、このネットワークインターフェイスを介してのみ Amazon DocumentDB クラスターにアクセスできます。

ステップ 2: パイプラインを作成する

その後、Amazon DocumentDB OpenSearch をソースとして指定する Ingestion パイプラインを次のように設定できます。インデックス名を入力するために、getMetadata関数はメタデータキー*documentdb_collection*として を使用することに注意してください。getMetadata メソッドを使用せずに別のインデックス名を使用する場合は、設定 を使用できますindex: "*my_index_name*"。

```
version: "2"
documentdb-pipeline:
  source:
    documentdb:
      acknowledgments: true
      host: "https://docdb-cluster-id.us-east-1.docdb.amazonaws.com"
      port: 27017
```

```
      authentication:
        username: ${aws_secrets:secret:username}
        password: ${aws_secrets:secret:password}
      aws:
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      s3_bucket: "bucket-name"
      s3_region: "bucket-region"
      s3_prefix: "path" #optional path for storing the temporary data
      collections:
        - collection: "dbname.collection"
          export: true
          stream: true
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${getMetadata(\"documentdb_collection\")}"
      index_type: custom
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
extension:
  aws:
    secrets:
      secret:
        secret_id: "my-docdb-secret"
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
        refresh_interval: PT1H
```

事前設定された Amazon DocumentDB ブループリントを使用して、このパイプラインを作成できます。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

を使用してパイプライン AWS Management Console を作成する場合は、Amazon DocumentDB をソースとして使用するため、パイプラインを VPC にアタッチする必要もあります。これを行うには、ネットワーク設定セクションを検索し、VPC にアタッチチェックボックスをオンにして、提供されているデフォルトオプションのいずれかから CIDR を選択するか、独自のものを選択します。

カスタム CIDR を指定するには、ドロップダウンメニューからその他を選択します。Ingestion と Amazon DocumentDB OpenSearch 間の IP アドレスの衝突を回避するには、Amazon DocumentDB VPC CIDR が Ingestion の CIDR OpenSearch と異なることを確認します。

詳細については、「パイプラインの VPC アクセスの設定」を参照してください。

データ整合性

パイプラインは、Amazon DocumentDB クラスターから変更を継続的にポーリングまたは受信し、インデックス内の対応するドキュメントを更新することで、データの整合性を確保します OpenSearch。

OpenSearch 取り込みでは、データの耐久性を確保するため end-to-end の確認がサポートされています。パイプラインがスナップショットまたはストリームを読み取る際に、並列処理用のパーティションが動的に作成されます。パイプラインは、 OpenSearch ドメインまたはコレクション内のすべてのレコードを取り込んだ後に確認を受け取ると、パーティションを完了としてマークします。

OpenSearch Serverless 検索コレクションに取り込む場合は、パイプラインでドキュメント ID を生成できます。 OpenSearch Serverless 時系列コレクションに取り込む場合は、パイプラインがドキュメント ID を生成しないため、パイプラインシンク設定`document_id: "${getMetadata(\"primary_key\")}"`で を省略する必要があることに注意してください。

Ingestion OpenSearch パイプラインは、受信イベントアクションを対応する一括インデックス作成アクションにマッピングして、ドキュメントの取り込みにも役立ちます。これによりデータの整合性が保たれるため、Amazon DocumentDB のすべてのデータ変更が、 の対応するドキュメントの変更と照合されます OpenSearch。

データ型のマッピング

OpenSearch サービスは、各受信ドキュメントのデータ型を Amazon DocumentDB の対応するデータ型に動的にマッピングします。次の表は、 OpenSearch サービスがさまざまなデータ型を自動的にマッピングする方法を示しています。

データ型	OpenSearch	Amazon DocumentDB
整数	OpenSearch は、Amazon DocumentDB 整数値を OpenSearch整数に自動的にマッピングします。 OpenSearch は、最初に送信されたドキュメントに基づいて フィールドを動的にマッピングします。Amazon DocumentDB で同じ属性のデータ型が混	Amazon DocumentDB は整数 をサポートしています。

データ型	OpenSearch	Amazon DocumentDB
	在している場合、自動マッピングが失敗する可能性があります。 例えば、最初のドキュメントに長い属性があり、後のドキュメントに整数と同じ属性がある場合、 は 2 番目のドキュメントの取り込みに OpenSearch 失敗します。このような場合は、次のような最も柔軟な番号タイプを選択する明示的なマッピングテンプレートを指定する必要があります。 `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	

データ型	OpenSearch	Amazon DocumentDB
Long	OpenSearch は、Amazon DocumentDB の長い値を長い値 OpenSearchに自動的にマッピングします。 OpenSearch は、最初に送信されたドキュメントに基づいて フィールドを動的にマッピングします。Amazon DocumentDB で同じ属性のデータ型が混在している場合、自動マッピングが失敗する可能性があります。 例えば、最初のドキュメントに長い属性があり、後のドキュメントに整数と同じ属性がある場合、 は 2 番目のドキュメントの取り込みに OpenSearch 失敗します。このような場合は、次のような最も柔軟な番号タイプを選択する明示的なマッピングテンプレートを指定する必要があります。 `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB は longs をサポートしています。

データ型	OpenSearch	Amazon DocumentDB
文字列	OpenSearch は文字列値をテキストとして自動的にマッピングします。状況によっては (列挙型の値など)、キーワード型にマッピングできます。 次の例は、 という名前の Amazon DocumentDB 属性を OpenSearch キーワード`PartType`にマッピングする方法を示しています。 `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	Amazon DocumentDB は文字列 をサポートします。

データ型	OpenSearch	Amazon DocumentDB
ダブル	OpenSearch は、Amazon DocumentDB の倍精度値を倍精度に自動的にOpenSearchマッピングします。 OpenSearch は、最初に送信されたドキュメントに基づいて フィールドを動的にマッピングします。Amazon DocumentDB で同じ属性のデータ型が混在している場合、自動マッピングが失敗する可能性があります。 例えば、最初のドキュメントに長い属性があり、後のドキュメントに整数と同じ属性がある場合、 は 2 番目のドキュメントの取り込みに OpenSearch 失敗します。このような場合は、次のような最も柔軟な番号タイプを選択する明示的なマッピングテンプレートを指定する必要があります。 `{ "template": { "mappings": { "properties": { "MixedNumberField": { "type": "float" } } } } }`	Amazon DocumentDB は、 を 2 倍サポートしています。

データ型	OpenSearch	Amazon DocumentDB
日付	デフォルトでは、日付は の整数にマッピングされます OpenSearch。日付を日付にマッピングするカスタムマッピングテンプレートを定義できます OpenSearch 。 `{ "template": { "mappings": { "properties": { "myDateField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB は日付 をサポートしています。
タイムスタンプ	デフォルトでは、タイムスタンプは の整数にマッピングされます OpenSearch。日付を日付に OpenSearchマッピングするカスタムマッピングテンプレートを定義できます。 `{ "template": { "mappings": { "properties": { "myTimestampField": { "type": "date", "format": "epoch_second" } } } } }`	Amazon DocumentDB はタイムスタンプ をサポートしています。

データ型	OpenSearch	Amazon DocumentDB
ブール値	OpenSearch は Amazon DocumentDB ブール型を OpenSearch ブール型にマッピングします。	Amazon DocumentDB はブール型の属性をサポートしています。
10 進数	OpenSearch は Amazon DocumentDB マップ属性をネストされたフィールドにマッピングします。同じマッピングがネストされたフィールド内でも適用されます。 次の例では、ネストされたフィールドの文字列を のキーワードタイプにマッピングします OpenSearch。 `{ "template": { "mappings": { "properties": { "myDecimalField": { "type": "double" } } } } }` このカスタムマッピングを使用すると、フィールドをクエリして二重レベルの精度で集計できます。元の値は、ドキュメントの `_source`プロパティで完全な精度を保持します OpenSearch。このマッピングがない場合、 はデフォルトでテキスト OpenSearch を使用します。	Amazon DocumentDB は小数 をサポートしています。

データ型	OpenSearch	Amazon DocumentDB
正規表現	正規表現タイプは、ネストされたフィールドを作成します。これらには、*<myFieldName>* .patternと が含まれます*<myFieldName>* .options。	Amazon DocumentDB は正規表現 をサポートしています。
バイナリデータ	OpenSearch は、Amazon DocumentDB バイナリデータをテキストに自動的にマッピングします OpenSearch。マッピングを指定して、これらを のバイナリフィールドとして記述できます OpenSearch。 次の例は、 という名前の Amazon DocumentDB フィールドimageData をOpenSearch バイナリフィールドにマッピングする方法を示しています。 `{ "template": { "mappings": { "properties": { "imageData": { "type": "binary" } } } } }`	Amazon DocumentDB はバイナリデータフィールド をサポートしています。
ObjectId	objectId タイプのフィールドは、OpenSearch テキストフィールドにマッピングされます。値は objectId の文字列表現になります。	Amazon DocumentDB は objectIdsをサポートしています。

データ型	OpenSearch	Amazon DocumentDB
Null	OpenSearch は、Amazon DocumentDB null タイプのドキュメントを取り込むことができます。値は null 値としてドキュメントに保存されます。この型にはマッピングがなく、このフィールドにはインデックスが付けられず、検索もできません。 NULL 型に同じ属性名が使用され、後で文字列などの異なる型に変更されると、は最初の NULL 以外の値の動的マッピング OpenSearch を作成します。それ以降の値はAmazon DocumentDB null 値のままにすることができます。	Amazon DocumentDB は null 型フィールド をサポートしています。
未定義	OpenSearch は、Amazon DocumentDB 未定義タイプのドキュメントを取り込むことができます。値は null 値としてドキュメントに保存されます。この型にはマッピングがなく、このフィールドにはインデックスが付けられず、検索もできません。 同じフィールド名が未定義の型に使用され、後で文字列などの別の型に変更された場合、 は最初の未定義の値の動的マッピング OpenSearch を作成します。後続の値はAmazon DocumentDB の未定義の値のままにすることができます。	Amazon DocumentDB は、未定義の型フィールド をサポートしています。

データ型	OpenSearch	Amazon DocumentDB
MinKey	OpenSearch は、Amazon DocumentDB minKey タイプのドキュメントを取り込むことができます。値は null 値としてドキュメントに保存されます。この型にはマッピングがなく、このフィールドにはインデックスが付けられず、検索もできません。 minKey タイプに同じフィールド名が使用され、後で文字列などの異なるタイプに変更された場合、 は最初の minKey 以外の値の動的マッピング OpenSearch を作成します。後続の値は、Amazon DocumentDB minKey 値のままにすることができます。	Amazon DocumentDB は minKey タイプフィールド をサポートしています。
MaxKey	OpenSearch は、Amazon DocumentDB maxKey タイプのドキュメントを取り込むことができます。値は null 値としてドキュメントに保存されます。この型にはマッピングがなく、このフィールドにはインデックスが付けられず、検索もできません。 maxKey 型に同じフィールド名が使用され、後で文字列などの異なる型に変更された場合、 は最初の maxKey 以外の値の動的マッピング OpenSearch を作成します。それ以降の値は Amazon DocumentDB maxKey 値のままにすることができます。	Amazon DocumentDB は maxKey タイプフィールド をサポートしています。

取り込みパイプラインでデッドレターキュー (DLQ) OpenSearch を設定することをお勧めします。キューを設定した場合、動的マッピングの失敗により取り込むことができない失敗したドキュメントはすべて、 OpenSearch サービスからキューに送信されます。

自動マッピングが失敗した場合は、パイプライン設定で `template_type` と `template_content` を使用して明示的なマッピングルールを定義できます。あるいは、パイプラインを開始する前に、検索ドメインまたはコレクションにマッピングテンプレートを直接作成することもできます。

制限事項

Amazon DocumentDB OpenSearch の取り込みパイプラインを設定するときは、次の制限事項を考慮してください。

- Amazon DocumentDB との OpenSearch Ingestion 統合は現在、クロスリージョン取り込みをサポートしていません。Amazon DocumentDB クラスターと Ingestion OpenSearch パイプラインは同じ にある必要があります AWS リージョン。
- Amazon DocumentDB との OpenSearch Ingestion 統合は現在、クロスアカウント取り込みをサポートしていません。Amazon DocumentDB クラスターと Ingestion OpenSearch パイプラインは同じ にある必要があります AWS アカウント。
- OpenSearch 取り込みパイプラインは、ソースとして 1 つの Amazon DocumentDB クラスターのみをサポートします。
- Amazon DocumentDB との OpenSearch Ingestion 統合は、Amazon DocumentDB インスタンスベースのクラスターを特にサポートしています。Amazon DocumentDB エラスティッククラスターはサポートされていません。
- OpenSearch Ingestion 統合は、Amazon DocumentDB クラスターの認証メカニズム AWS Secrets Manager としてのみ をサポートします。
- 既存のパイプライン設定を更新して、別のデータベースまたはコレクションからデータを取り込むことはできません。代わりに、新しいパイプラインを作成する必要があります。

Confluent Kafka クラウドでの OpenSearch Ingestion パイプラインの使用

Confluent Kafka を Ingestion OpenSearch のソースとして使用して、Confluent Kafka クラスターから Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションにデータをストリーミングできます。 OpenSearch Ingestion は、パブリックネットワークスペースとプライベートネットワークスペースのセルフマネージド Kafka からのストリーミングデータの処理をサポートします。

Confluent パブリック Kafka クラウドへの接続

Ingestion OpenSearch パイプラインを使用して、パブリック設定の Confluent Kafka クラスターからデータをストリーミングできます (ブートストラップサーバーの DNS 名はパブリックに解決す

る必要があります）。これを行うには、取り込みパイプライン、ソースとしての Confluent Kafka OpenSearch クラスター、および送信先としての Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションが必要です。

データを移行するには、以下が必要です。

- ソースとして機能する Confluent Kafka クラスター。クラスターには、移行するデータが含まれている必要があります。
- 送信先として機能する Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクション。
- Kafka クラスターでは、 の認証情報を使用して認証を有効にする必要があります AWS Secrets Manager。

要件

セルフマネージド OpenSearch 型または Elasticsearch ソースクラスターで AWS Secrets Manager ベース認証を有効にするには、

- 「シークレットのローテーション」の手順に従って AWS Secrets Manager 、Confluent Kafka クラスターで認証を設定します。 AWS Secrets Manager
- Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションに書き込むアクセス許可を持つパイプラインロールを IAM に作成します。また、 から認証情報を読み取るアクセス許可を指定する必要があります AWS Secrets Manager。これを実行するには:
  - リソースベースのポリシーを Amazon OpenSearch Service ドメインにアタッチするか、コレクションにデータアクセスポリシーをアタッチします。これらのアクセスポリシーにより、Ingestion OpenSearch は自己管理 OpenSearch 型または Elasticsearch ソースクラスターから Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションにデータを書き込むことができます。
- 設計図を参照して Ingestion OpenSearch パイプラインを作成します。

  これらのステップを完了すると、パイプラインはソースクラスターからのデータの処理を自動的に開始し、Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションの送信先に取り込みます。Ingestion OpenSearch パイプラインのさまざまなプロセッサを使用して、取り込まれたデータに対して変換を実行できます。

IAM ロールとアクセス許可

次のサンプルドメインアクセスポリシーでは、次のステップで作成するパイプラインロールが Amazon OpenSearch Service ドメインにデータを書き込むことを許可します。リソースは、必ず独自の ARN で更新してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{pipeline-account-id}:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:{region}:{account-id}:domain/domain-name"
      ]
    }
  ]
}
```

ネットワークインターフェイスを管理するには、次のアクセス許可が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:{account-id}:network-interface/*",
```

```
                "arn:aws:ec2:*:{account-id}:subnet/*",
                "arn:aws:ec2:*:{account-id}:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [ "ec2:CreateTags" ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals": { "aws:RequestTag/OSISManaged": "true" }
            }
        }
    ]
}
```

以下は、 AWS Secrets Manager サービスからシークレットを読み取るために必要なアクセス許可です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": ["arn:aws:secretsmanager:<region>:<account-id>:secret:<secret-
name>"]
        }
    ]
}
```

Amazon OpenSearch Service ドメインに書き込むには、次のアクセス許可が必要です。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

パイプラインの作成

ポリシーをパイプラインロールにアタッチしたら、Confluent Kafka データ移行パイプラインの設計図を使用してパイプラインを作成します。このブループリントには、Kafka と送信先の間でデータを移行するためのデフォルト設定が含まれています。

- 複数の Amazon OpenSearch Service ドメインをデータの送信先として指定できます。この機能を使用すると、受信データを複数の Amazon OpenSearch Servicedomain に条件付きでルーティングまたはレプリケーションできます。
- ソース Confluent Kafka クラスターから Amazon OpenSearch Serverless VPC コレクションにデータを移行できます。パイプライン設定内でネットワークアクセスポリシーを指定していることを確認します。
- confluent スキーマレジストリを使用して、 と confluent スキーマを定義できます。

次のサンプルパイプラインは、Confluent Kafka クラスターから Amazon OpenSearch Service ドメインにデータを取り込みます。

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      # Encryption is always required
      encryption:
        type: "ssl"
      topics:
```

```
        - name: "topic_4"
          group_id: "demoGroup"
      bootstrap_servers:
        # TODO: for public confluent kafka use public booststrap server dns
        - "<<bootstrap-server>>.us-west-2.aws.private.confluent.cloud:9092"
      authentication:
        sasl:
          plain:
            username: "${{aws_secrets:confluent-kafka-secret:username}}"
            password: "${{aws_secrets:confluent-kafka-secret:password}}"
      # Schema is optional
      schema:
        type: confluent
        registry_url: https://<<registry-url>>.us-west-2.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
sink:
    - opensearch:
        hosts: [ "https://<<opensearchdomain>>.us-west-2.es.amazonaws.com" ]
        index: "enterprise-confluent-demo"
        aws:
          sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
          region: "<<aws-region>>"
 extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "enterprise-kafka-credentials"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
      schema-secret:
        secret_id: "self-managed-kafka-schema"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
```

VPC 内の Confluent Kafka クラウドへの接続

Ingestion OpenSearch パイプラインを使用して、パブリック設定で Confluent Kafka クラスターからデータをストリーミングできます。これを行うには、Confluent Kafka OpenSearch をソースとして、Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションを送信先として Ingestion パイプラインを設定します。パイプラインは、kafka クラスターからのすべてのストリーミングデータを処理し、そのデータを宛先クラスターに取り込みます。

Confluent Kafka ネットワーク設定

OpenSearch Ingestion は、Confluent でサポートされているすべてのネットワークモードで設定された Confluent Kafka クラスターをサポートします。以下のネットワーク設定モードは、 OpenSearch 取り込みのソースとしてサポートされています。

- AWS VPC ピアリング接続
- AWS PrivateLink 専用クラスター用の
- AWS PrivateLink for Enterprise クラスター
- AWS Transit Gateway

Confluent マネージド Kafka は、Confluent クラウドからデータを取り込むためのソースとして使用できます。これを実現するには、Kafka をソースとして設定し、Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションをシンクとして設定するパイプラインを設定します。これにより、Kafka から指定された宛先へのデータの移行が容易になります。移行では、confluent レジストリの使用もサポートされます。レジストリはまったく使用されません。

データ移行を実行するには、次のリソースが必要です。

- 移行する予定のデータを含む、ソースとして機能する Confluent Kafka クラスター。
- Amazon OpenSearch Service ドメインやシンクとしての Amazon OpenSearch Serverless コレクションなどのターゲット送信先。
- Confluent VPC にアクセスできる Amazon VPC の VPC ID。
- Kafka クラスターでは、 の認証情報を使用して認証を有効にする必要があります AWS Secrets Manager。

要件

Kafka クラスターで取り込みを設定するには、以下が必要です。

- Kafka クラスターで AWS Secrets Manager ベースの認証を有効にする必要があります。
  - を使用して Kafka クラスターで認証を設定します AWS Secrets Manager。「シークレットのローテーション」の手順に従って、AWS Secrets Manager シークレットのローテーションを有効にします。
- Ingestion サービスで使用する VPC CIDR OpenSearch を指定する必要があります。

  - AWS マネジメントコンソールを使用してパイプラインを作成する場合は、Confluent Kafka OpenSearch をソースとして使用するには、VPC に Amazon Ingestion パイプラインもアタッチする必要があります。これを行うには、ネットワーク設定セクションを見つけ、VPC にアタッチチェックボックスを選択し、CIDR を選択するか、取り込みで使用する /24 CIDR OpenSearch を手動で入力します。Ingestion で使用するように選択した CIDR OpenSearch は、Confluent マネージド Kafka が実行されている VPC CIDR とは異なる必要があります。Confluent Kafka CIDR の詳細については、「」を参照してください。以下は、Ingestion Service がネットワーク接続を作成するために使用できるデフォルトの CIDR OpenSearch オプションです。
    - 10.99.20.0/24
    - 192.168.36.0/24
    - 172.21.56.0/24
- Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションへのアクセス許可と、 からシークレットを読み取るアクセス許可を持つパイプラインロールを IAM で作成する必要があります AWS Secrets Manager。
  - リソースベースのポリシーを Amazon OpenSearch Servicedomain にアタッチするか、Amazon OpenSearch Serverless データアクセスポリシーをコレクションにアタッチします。これらのアクセスポリシーにより、Ingestion OpenSearch は Kafka から Amazon OpenSearch Service ドメインまたは Amazon OpenSearch Serverless コレクションにデータを書き込むことができます。

- AWS PrivateLink 接続可能な Confluent Kafka の場合は、 を設定します。

  VPC DHCP オプション 。DNS ホスト名と DNS 解決を有効にする必要があります。
  - domain-name: aws.private.confluent.cloud

    domain-name-servers: Amazon が提供する DNS

IAM ロールとアクセス許可

次のサンプルドメインアクセスポリシーでは、パイプラインロールが Amazon OpenSearch Service ドメインにデータを書き込むことを許可します。

> **Note**
>
> を独自の ARN `resource`で更新する必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{pipeline-account-id}:role/pipeline-role"
      },
      "Action": [
        "es:DescribeDomain",
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:{region}:{account-id}:domain/domain-name"
      ]
    }
  ]
}
```

次のサンプルは、ネットワークインターフェイスを管理するために必要なアクセス許可を提供します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DetachNetworkInterface",
                "ec2:DescribeNetworkInterfaces"
            ],
            "Resource": [
                "arn:aws:ec2:*:{account-id}:network-interface/*",
                "arn:aws:ec2:*:{account-id}:subnet/*",
                "arn:aws:ec2:*:{account-id}:security-group/*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [ "ec2:CreateTags" ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
               "StringEquals": { "aws:RequestTag/OSISManaged": "true" }
            }
        }
    ]
```

次のサンプルは、 からシークレットを読み取るために必要なアクセス許可を提供します AWS Secrets Manager。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "SecretsManagerReadAccess",
            "Effect": "Allow",
            "Action": ["secretsmanager:GetSecretValue"],
            "Resource": ["arn:aws:secretsmanager:<region>:<account-id>:secret:<secret-name>"]
        }
    ]
}
```

次のサンプルは、Amazon OpenSearch Service ドメインへの書き込みに必要なアクセス許可を提供します。

```
{
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

パイプラインの作成

ポリシーをパイプラインロールにアタッチしたら、Confluent Kafka データ移行パイプラインの設計図を使用してパイプラインを作成できます。このブループリントには、Kafka と送信先の間でデータを移行するためのデフォルト設定が含まれています。

- 複数の Amazon OpenSearch Service ドメインをデータの送信先として指定できます。この機能により、受信データを複数の Amazon OpenSearch Service に条件付きでルーティングまたはレプリケーションできます。
- ソース Confluent Kafka クラスターから Amazon OpenSearch Serverless VPC コレクションにデータを移行できます。パイプライン設定内でネットワークアクセスポリシーを指定していることを確認します。
- Confluent スキーマレジストリを使用して、 と Confluent スキーマを定義できます。

サンプルパイプライン設定

```
version: "2"
kafka-pipeline:
  source:
    kafka:
      # Encryption is always required
      encryption:
        type: "ssl"
      topics:
        - name: "topic_4"
          group_id: "demoGroup"
      bootstrap_servers:
        # TODO: for public confluent kafka use public booststrap server dns
        - "<<bootstrap-server>>.us-west-2.aws.private.confluent.cloud:9092"
```

```
      authentication:
        sasl:
          plain:
            username: "${{aws_secrets:confluent-kafka-secret:username}}"
            password: "${{aws_secrets:confluent-kafka-secret:password}}"
      # Schema is optional
      schema:
        type: confluent
        registry_url: https://<<registry-url>>.us-west-2.aws.confluent.cloud
        api_key: "${{aws_secrets:schema-secret:schema_registry_api_key}}"
        api_secret: "${{aws_secrets:schema-secret:schema_registry_api_secret}}"
        basic_auth_credentials_source: "USER_INFO"
  sink:
    - opensearch:
        hosts: [ "https://<<opensearchdomain>>.us-west-2.es.amazonaws.com" ]
        index: "enterprise-confluent-demo"
        aws:
          sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
          region: "<<aws-region>>"
extension:
  aws:
    secrets:
      confluent-kafka-secret:
        secret_id: "enterprise-kafka-credentials"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
      schema-secret:
        secret_id: "self-managed-kafka-schema"
        region: "<<aws-region>>"
        sts_role_arn: "arn:aws:iam::1234567890:role/os-os-test-role"
```

での OpenSearch 取り込みパイプラインの使用 Amazon Managed Streaming for Apache Kafka

Kafka プラグインを使用して、Amazon Managed Streaming for Apache Kafka (Amazon MSK) から Ingestion OpenSearch パイプラインにデータを取り込むことができます。Amazon MSK を使用すると、Apache Kafka をストリーミングデータの処理に使用するアプリケーションを構築および実行できます。 OpenSearch 取り込みでは AWS PrivateLink 、 を使用して Amazon MSK に接続します。Amazon MSK クラスターと Amazon MSK Serverless クラスターの両方からデータを取り込むことができます。2 つのプロセスの違いは、パイプラインを設定する前に実行する必要がある前提条件の手順だけです。

トピック

Amazon MSK の前提条件

OpenSearch 取り込みパイプラインを作成する前に、次のステップを実行します。

1. 「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「クラスターの作成」の手順に従って、Amazon MSK でプロビジョニングされたクラスターを作成します。ブローカータイプ では、t3タイプ以外のオプションを選択します。これらは Ingestion OpenSearch ではサポートされていないためです。
2. クラスターのステータスが Active になったら、「マルチ VPC 接続を有効にする」の手順に従います。
3. クラスターとパイプラインが同じ AWS アカウントにあるかどうかに応じて、「クラスターポリシーを MSK クラスターにアタッチする」のステップに従い、以下のポリシーのいずれかをアタッチします。このポリシーにより、Ingestion OpenSearch は Amazon MSK クラスター AWS PrivateLink への接続を作成し、Kafka トピックからデータを読み取ることができます。必ず独自の ARN で `resource` を更新してください。

   クラスターとパイプラインが同じ AWS アカウントにある場合は、次のポリシーが適用されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
```

```
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    }
  ]
}
```

Amazon MSK クラスターがパイプライン AWS アカウント とは異なる にある場合は、代わりに次のポリシーをアタッチします。クロスアカウントアクセスは、プロビジョニングされた Amazon MSK クラスターでのみ可能で、Amazon MSK サーバーレスクラスターではできないことに注意してください。の AWS `principal` ARN は、パイプライン YAML 設定に提供するのと同じパイプラインロールの ARN である必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-
name/cluster-id"
```

```
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{pipeline-account-id}:role/pipeline-role"
      },
      "Action": [
        "kafka-cluster:*",
        "kafka:*"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-name/cluster-id",
        "arn:aws:kafka:us-east-1:{msk-account-id}:topic/cluster-name/cluster-id/*",
        "arn:aws:kafka:us-east-1:{msk-account-id}:group/cluster-name/*"
      ]
    }
  ]
}
```

4. 「トピックの作成」の手順に従って Kafka トピックを作成します。*BootstrapServerString* がプライベートエンドポイント (単一 VPC) のブートストラップ URL の 1 つであることを確認してください。の値は3、Amazon MSK クラスターのゾーン数に基づいて、 2または `--replication-factor`である必要があります。`--partitions` の値は少なくとも `10` である必要があります。

5. 「データの生成と消費」の手順に従って、データを生成して使用します。*BootstrapServerString* がプライベートエンドポイント (単一 VPC) のブートストラップ URL の 1 つであることを確認してください。

Amazon MSK Serverless の前提条件

OpenSearch 取り込みパイプラインを作成する前に、次のステップを実行します。

1. 「Amazon Managed Streaming for Apache Kafka デベロッパーガイド」の「MSK サーバーレスクラスターを作成する」の手順に従って、Amazon MSK サーバーレスクラスターを作成します。
2. クラスターのステータスがアクティブになったら、「クラスターポリシーを MSK クラスターにアタッチする」の手順に従って、次のポリシーをアタッチします。必ず独自の ARN で `resource` を更新してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    }
  ]
}
```

このポリシーにより、Ingestion OpenSearch は Amazon MSK Serverless クラスター AWS PrivateLink への接続を作成し、Kafka トピックからデータを読み取ることができます。このポリシーは、クラスターとパイプラインが同じ にある場合に適用されます。これは AWS アカウント、Amazon MSK Serverless がクロスアカウントアクセスをサポートしていないためです。

3. 「トピックの作成」の手順に従って Kafka トピックを作成します。*BootstrapServerString* が Simple Authentication and Security Layer (SASL) IAM ブートストラップ URLs の 1 つであることを確認します。の値は3、Amazon MSK Serverless クラスターのゾーン数に基づいて、 2または `--replication-factor`である必要があります。`--partitions` の値は少なくとも `10` である必要があります。
4. 「データの生成と消費」の手順に従って、データを生成して使用します。繰り返しになりますが、 *BootstrapServerString*が Simple Authentication and Security Layer (SASL) IAM ブートストラップ URLs の 1 つであることを確認してください。

ステップ 1: パイプラインロールを設定する

Amazon MSK プロビジョンドクラスターまたはサーバーレスクラスターを設定したら、パイプライン設定で使用するパイプラインロールに次の Kafka アクセス許可を追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka:DescribeClusterV2",
                "kafka:GetBootstrapBrokers"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-id"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:ReadData"
```

```
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:topic/cluster-name/cluster-
id/topic-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:group/cluster-name/*"
            ]
        }
    ]
}
```

ステップ 2: パイプラインを作成する

その後、Kafka OpenSearch をソースとして指定する Ingestion パイプラインを次のように設定できます。

```
version: "2"
log-pipeline:
  source:
    kafka:
      acknowledgements: true
      topics:
      - name: "topic-name"
        group_id: "group-id"
      aws:
        msk:
          arn: "arn:aws:kafka:{region}:{account-id}:cluster/cluster-name/cluster-id"
        region: "us-west-2"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  processor:
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
```

```
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
      index: "index_name"
      aws_sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      aws_region: "us-east-1"
      aws_sigv4: true
```

事前設定された Amazon MSK ブループリントを使用して、このパイプラインを作成できます。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

ステップ 3: (オプション) AWS Glue スキーマレジストリを使用する

Amazon MSK OpenSearch で Ingestion を使用する場合、Schema AWS Glue Registry でホストされているスキーマに AVRO データ形式を使用できます。AWS Glue スキーマレジストリを使用すると、データストリームスキーマを一元的に検出、制御、および展開できます。

このオプションを使用するには、パイプライン設定で type スキーマを有効にします。

```
schema:
  type: "aws_glue"
```

また、パイプラインロール AWS Glue で読み取りアクセス許可を に提供する必要があります。という AWS マネージドポリシーを使用できますAWSGlueSchemaRegistryReadonlyAccess。さらに、レジストリは、取り込みパイプラインと同じ OpenSearch AWS アカウント およびリージョンに存在する必要があります。

ステップ 4: (オプション) Amazon MSK パイプラインの推奨コンピューティングユニット (OCU) を設定する

各コンピューティングユニットには、トピックごとに 1 つのコンシューマーがあります。ブローカーは、特定のトピックについて、これらのコンシューマー間でパーティションのバランスを取ります。ただし、パーティションの数がコンシューマーの数よりも多い場合、Amazon MSK は各コンシューマーで複数のパーティションをホストします。 OpenSearch 取り込みには、CPU 使用率またはパイプライン内の保留中のレコード数に基づいてスケールアップまたはスケールダウンする自動スケーリングが組み込まれています。

最適なパフォーマンスを得るには、パーティションを多くのコンピューティングユニットに分散して並列処理を行います。トピックに多くのパーティションがある場合 (パイプラインあたりの最大数である 96 以上の OCU がある場合など)、1 ~ 96 個の OCU でパイプラインを設定することをお勧めします。これは、必要に応じて自動的にスケールするためです。トピックのパーティション数が少ない場合 (96 未満の場合など)、最大コンピューティングユニットをパーティションの数と同じにします。

パイプラインに複数のトピックがある場合は、最大コンピューティングユニットを設定する参照としてパーティション数が最も多いトピックを選択します。新しい OCU セットを含むパイプラインを同じトピックとコンシューマーグループに追加すると、スループットをほぼ直線的にスケールすることができます。

Amazon S3 OpenSearch での取り込みパイプラインの使用

OpenSearch 取り込みでは、Amazon S3 をソースまたは送信先として使用できます。Amazon S3 をソースとして使用する場合は、 OpenSearch データを取り込みパイプラインに送信します。Amazon S3 を送信先として使用する場合は、取り込みパイプラインから 1 OpenSearch つ以上の S3 バケットにデータを書き込みます。

トピック

- ソースとしての Amazon S3
- 送信先としての Amazon S3
- ソースとしての Amazon S3 クロスアカウント

ソースとしての Amazon S3

Amazon S3 をデータ処理のソースとして使用する方法には、S3-SQS 処理とスケジュールされたスキャンの 2 つがあります。

S3 にファイルが書き込まれた後にほぼリアルタイムでファイルをスキャンする必要がある場合は、S3-SQS 処理を使用します。バケット内でオブジェクトが保存または変更されるたびにイベントを発生させるように Amazon S3 バケットを設定できます。S3 バケット内のデータをバッチ処理するには、1 回限りのスケジュールされたスキャンまたは反復的なスケジュールされたスキャンを使用します。

トピック

- 前提条件
- ステップ 1: パイプラインロールを設定する

- ステップ 2: パイプラインを作成する

前提条件

スケジュールされたスキャンまたは S3-SQS 処理の両方の取り込みパイプラインのソースとして Amazon S3-SQS を使用するには、まず S3 バケット を作成します。 OpenSearch

> **Note**
>
> 取り込みパイプラインのソースとして使用される S3 OpenSearch バケットが別の にある場合は AWS アカウント、バケットでクロスアカウント読み取りアクセス許可を有効にする必要もあります。これにより、パイプラインはデータを読み取って処理できるようになります。クロスアカウントアクセス許可を有効にするには、「Amazon S3 ユーザーガイド」の「バケット所有者がクロスアカウントのバケットのアクセス許可を付与する」を参照してください。
> S3 バケットが複数のアカウントにある場合は、`bucket_owners`マップを使用します。例については、 ドキュメントの「クロスアカウント S3 アクセス OpenSearch」を参照してください。

S3-SQS 処理をセットアップするには、以下のステップも実行する必要があります。

1. Amazon SQS キューを作成します。
2. SQS キューを送信先とする S3 バケットでイベント通知を有効にします。

ステップ 1: パイプラインロールを設定する

パイプラインにデータをプッシュする他のソースプラグインとは異なり、S3 ソースプラグインは、パイプラインがソースからデータを取得する読み取りベースのアーキテクチャを使用しています。

したがって、パイプラインが S3 から読み取るには、パイプラインの S3 ソース設定で、S3 バケットと Amazon SQS キューの両方にアクセスできるロールを指定する必要があります。キューからデータを読み取るために、パイプラインはこのロールを引き受けます。

> **Note**
>
> S3 ソース設定内で指定するロールは、パイプラインロールである必要があります。したがって、パイプラインロールには 2 つの異なるアクセス許可ポリシーを含める必要があり

ます。1 つはシンクに書き込むポリシーで、もう 1 つは S3 ソースから取得するポリシーです。すべてのパイプラインコンポーネントで同じ sts_role_arn を使用する必要があります。

次のサンプルポリシーは、S3 をソースとして使用するために必要なアクセス許可を示しています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":[
          "s3:ListBucket",
          "s3:GetBucketLocation",
          "s3:GetObject"
       ],
      "Resource": "arn:aws:s3:::my-bucket/*"
    },
    {
       "Effect":"Allow",
       "Action":"s3:ListAllMyBuckets",
       "Resource":"arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility"
      ],
      "Resource": "arn:aws:sqs:us-west-2:{account-id}:MyS3EventSqsQueue"
    }
  ]
}
```

S3 ソースプラグイン設定の sts_role_arn オプションで指定した IAM ロールに、これらのアクセス許可をアタッチする必要があります。

```
version: "2"
source:
  s3:
```

```
    ...
    aws:
      ...
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
processor:
  ...
sink:
  - opensearch:
      ...
```

ステップ 2: パイプラインを作成する

アクセス許可を設定したら、Amazon OpenSearch S3 のユースケースに応じて取り込みパイプラインを設定できます。 Amazon S3

S3-SQS 処理

S3-SQS 処理をセットアップするには、パイプラインを設定して S3 をソースとして指定し、Amazon SQS 通知を設定します。

```
version: "2"
s3-pipeline:
  source:
    s3:
      notification_type: "sqs"
      codec:
        newline: null
      sqs:
        queue_url: "https://sqs.us-east-1.amazonaws.com/{account-id}/ingestion-queue"
      compression: "none"
      aws:
        region: "us-east-1"
        # IAM role that the pipeline assumes to read data from the queue. This role
 must be the same as the pipeline role.
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  processor:
  - grok:
      match:
        message:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
```

```
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
      index: "index-name"
      aws:
        # IAM role that the pipeline assumes to access the domain sink
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
        region: "us-east-1"
```

Amazon S3 で小さなファイルを処理するときに CPU 使用率が低い場合は、 workersオプションの値を変更してスループットを上げることを検討してください。詳細については、S3 プラグインの設定オプション」を参照してください。

スケジュールされたスキャン

スケジュールされたスキャンをセットアップするには、すべての S3 バケットまたはバケットレベルに適用されるスキャンレベルで、パイプラインにスケジュールを設定します。バケットレベルのスケジュールまたはスキャン間隔の設定は、常にスキャンレベルの設定を上書きします。

スケジュールれたスキャンは、データ移行に理想的な 1 回限りのスキャン、またはバッチ処理に理想的な反復的スキャンで設定できます。

Amazon S3 から読み取るようにパイプラインを設定するには、事前設定された Amazon S3 ブループリントを使用します。パイプライン設定の scan の部分は、スケジュールのニーズに合わせて編集できます。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

1 回限りのスキャン

1 回限りのスケジュールされたスキャンは 1 回実行されます。YAML 設定で、start_time および end_time を使用して、バケット内のオブジェクトをスキャンするタイミングを指定できます。range を使用して、バケット内のオブジェクトをスキャンする現在の時刻に相対的な間隔を指定できます。

例えば、範囲を PT4H に設定すると、過去 4 時間に作成されたすべてのファイルがスキャンされます。1 回限りのスキャンを 2 回目に実行するように設定するには、パイプラインを停止して再起動する必要があります。範囲を設定しない場合、開始時間と終了時間も更新する必要があります。

次の設定では、すべてのバケット、およびそれらのバケット内のすべてのオブジェクトを 1 回だけスキャンするようにセットアップされます。

```
version: "2"
```

```
log-pipeline:
  source:
    s3:
      codec:
        csv:
      compression: "none"
      aws:
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      acknowledgments: true
      scan:
        buckets:
          - bucket:
              name: my-bucket-1
              filter:
                include_prefix:
                  - Objects1/
                exclude_suffix:
                  - .jpeg
                  - .png
          - bucket:
              name: my-bucket-2
              key_prefix:
                include:
                  - Objects2/
                exclude_suffix:
                  - .jpeg
                  - .png
      delete_s3_objects_on_read: false
  processor:
    - date:
        destination: "@timestamp"
        from_time_received: true
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index: "index-name"
        aws:
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
        dlq:
          s3:
            bucket: "my-bucket-1"
            region: "us-east-1"
```

```
            sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
```

次の設定では、指定した時間ウィンドウですべてのバケットを 1 回だけスキャンするようにセットアップされます。したがって、S3 は、作成時間がこのウィンドウ内に収まるオブジェクトのみを処理します。

```
scan:
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

次の設定では、スキャンレベルとバケットレベルの両方で 1 回限りのスキャンがセットアップされます。バケットレベルの開始時間と終了時間は、スキャンレベルの開始時間と終了時間よりも優先されます。

```
scan:
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-1
        filter:
          include:
            - Objects1/
```

```
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

パイプラインを停止すると、停止前にパイプラインによってスキャンされたオブジェクトの既存の参照が削除されます。1 つのスキャンパイプラインが停止すると、すべてのオブジェクトが既にスキャンされていても、その開始後に再スキャンされます。1 つのスキャンパイプラインを停止する必要がある場合は、パイプラインを再開する前に時間枠を変更することをお勧めします。

開始時刻と終了時刻でオブジェクトをフィルタリングする必要がある場合は、パイプラインの停止と開始が唯一のオプションです。開始時刻と終了時刻でフィルタリングする必要がない場合は、名前でオブジェクトをフィルタリングできます。名前で文字起こしをしても、パイプラインを停止して開始する必要はありません。これを行うには、 include_prefixと を使用しますexclude_suffix。

反復的スキャン

反復的なスケジュールされたスキャンでは、指定した S3 バケットのスキャンがスケジュールされた間隔で定期的に実行されます。個別のバケットレベルの設定はサポートされていないため、これらの間隔はスキャンレベルでのみ設定できます。

YAML 設定で、interval は反復的スキャンの頻度を 30 秒から 365 日の間で指定します。これらのスキャンのうちの最初のスキャンは、パイプラインの作成時に実行されます。count はスキャンインスタンスの合計数を定義します。

次の設定では、12 時間のスキャン間隔での反復的スキャンがセットアップされます。

```
scan:
  scheduling:
    interval: PT12H
    count: 4
  buckets:
```

```
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

送信先としての Amazon S3

取り込みパイプラインから S3 OpenSearch バケットにデータを書き込むには、事前設定された S3 ブループリントを使用して S3 シンク でパイプラインを作成します。このパイプラインは、選択的データを OpenSearch シンクにルーティングし、同時にすべてのデータを S3 にアーカイブします。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

S3 シンクを作成するとき、さまざまなシンクコーデックから優先フォーマットを指定できます。例えば、列形式でデータを書き込む場合は、Parquet コーデックまたは Avro コーデックを選択します。行ベースの形式の場合は、JSON または ND-JSON を選択します。指定したスキーマで S3 にデータを書き込むには、Avro 形式を使用してシンクコーデック内にインラインスキーマを定義することもできます。

次の例では、S3 シンクのインラインスキーマが定義されます。

```
- s3:
  codec:
    parquet:
      schema: >
        {
           "type" : "record",
           "namespace" : "org.vpcFlowLog.examples",
           "name" : "VpcFlowLog",
           "fields" : [
```

```
            { "name" : "version", "type" : "string"},
            { "name" : "srcport", "type": "int"},
            { "name" : "dstport", "type": "int"},
            { "name" : "start", "type": "int"},
            { "name" : "end", "type": "int"},
            { "name" : "protocol", "type": "int"},
            { "name" : "packets", "type": "int"},
            { "name" : "bytes", "type": "int"},
            { "name" : "action", "type": "string"},
            { "name" : "logStatus", "type" : "string"}
        ]
    }
```

このスキーマを定義するとき、パイプラインがシンクに配信するさまざまなタイプのイベントに存在する可能性のあるすべてのキーのスーパーセットを指定します。

例えば、あるイベントでキーが欠落している可能性がある場合は、そのキーに null 値を付けてスキーマに追加します。null 値を宣言すると、(キーを持つイベントと持たないイベントがある) 不均一なデータをスキーマで処理できます。受信イベントにこれらのキーがある場合、その値はシンクに書き込まれます。

このスキーマ定義は、定義済みのキーのみをシンクに送信し、未定義のキーを受信イベントから削除するフィルターとして機能します。

include_keys と exclude_keys をシンクで使用して、他のシンクにルーティングされるデータをフィルタリングすることもできます。この 2 つのフィルターは相互に排他的であるため、スキーマでは一度に 1 つしか使用できません。また、ユーザー定義のスキーマと一緒に使用することもできません。

このようなフィルターでパイプラインを作成するには、事前設定されたシンクフィルターの設計図を使用します。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

ソースとしての Amazon S3 クロスアカウント

Amazon S3 のアカウント間でアクセスを許可して、取り込みパイプラインが別のアカウントの OpenSearch S3 バケットにソースとしてアクセスできるようにします。クロスアカウントアクセスを有効にするには、Amazon S3 ユーザーガイド」の「バケット所有者がクロスアカウントバケットのアクセス許可を付与する」を参照してください。アクセスを許可したら、パイプラインロールに必要なアクセス許可があることを確認します。

次に、 を使用して YAML 設定を作成しbucket_owners、Amazon S3 バケットへのクロスアカウントアクセスをソースとして有効にできます。

```
s3-pipeline:
 source:
  s3:
   notification_type: "sqs"
   codec:
    csv:
     delimiter: ","
     quote_character: "\""
     detect_header: True
   sqs:
    queue_url: "https://sqs.ap-northeast-1.amazonaws.com/401447383613/test-s3-queue"
   bucket_owners:
    my-bucket-01: 123456789012
    my-bucket-02: 999999999999
   compression: "gzip"
```

Amazon Security Lake OpenSearch での取り込みパイプラインの使用

S3 ソースプラグインを使用して、Amazon Security Lake から Ingestion OpenSearch パイプラインにデータを取り込むことができます。Security Lake は、 AWS 環境、オンプレミス環境、SaaS プロバイダーのセキュリティデータを専用のデータレイクに自動的に一元化します。Security Lake から Ingestion OpenSearch パイプラインにデータをレプリケートするサブスクリプションを作成して、それを OpenSearch サービスドメインまたは OpenSearch Serverless コレクションに書き込むことができます。

Security Lake から読み取るようにパイプラインを設定するには、事前設定された Security Lake ブループリントを使用します。ブループリントには、Security Lake から Open Cybersecurity Schema Framework (OCSF) Parquet ファイルを取り込むデフォルトの設定が含まれています。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

トピック

- 前提条件
- ステップ 1: パイプラインロールを設定する
- ステップ 2: パイプラインを作成する

前提条件

OpenSearch 取り込みパイプラインを作成する前に、次のステップを実行します。

- Security Hub を有効にします。
- Security Lake でサブスクライバーを作成します。
  - パイプラインに取り込むソースを選択します。
  - [サブスクライバーの認証情報] には、パイプラインを作成する AWS アカウント の ID を追加します。外部 ID には OpenSearchIngestion-*{accountid}* を指定します。
  - [データアクセスメソッド] には [S3] を選択します。
  - 通知の詳細には、SQS キューを選択します。

サブスクライバーを作成すると、Security Lake は 2 つのインライン許可ポリシーを自動的に作成します。1 つは S3 用、もう 1 つは SQS 用です。ポリシーの形式は AmazonSecurityLake-*{12345}*-S3 および AmazonSecurityLake-*{12345}*-SQS です。パイプラインがサブスクライバーソースにアクセスできるようにするには、必要なアクセス許可をパイプラインロールに関連付ける必要があります。

ステップ 1: パイプラインロールを設定する

Security Lake が自動的に作成した 2 つのポリシーから必要なアクセス許可のみを組み合わせた新しいアクセス許可ポリシーを IAM で作成します。次のポリシー例は、Ingestion パイプラインが複数の Security Lake OpenSearch ソースからデータを読み取るために必要な最小特権を示しています。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/
LAMBDA_EXECUTION/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/S3_DATA/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/VPC_FLOW/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/ROUTE53/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/SH_FINDINGS/1.0/*"
```

```
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sqs:ReceiveMessage",
            "sqs:DeleteMessage"
         ],
         "Resource":[
            "arn:aws:sqs:{region}:{account-id}:AmazonSecurityLake-abcde-Main-Queue"
         ]
      }
   ]
}
```

> **Important**
>
> Security Lake はパイプラインロールポリシーを管理しません。Security Lake サブスクリプションにソースを追加またはそこからソースを削除する場合は、ポリシーを手動で更新する必要があります。Security Lake はログソースごとにパーティションを作成するため、パイプラインロールのアクセス許可を手動で追加または削除する必要があります。

sqs において、S3 ソースプラグイン設定内の sts_role_arn オプションで指定した IAM ロールに、これらの許可をアタッチする必要があります。

```
version: "2"
source:
  s3:
    ...
    sqs:
      queue_url: "https://sqs.{region}.amazonaws.com/{account-id}/
AmazonSecurityLake-abcde-Main-Queue"
    aws:
      ...
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
processor:
  ...
sink:
  - opensearch:
      ...
```

ステップ 2: パイプラインを作成する

パイプラインロールにアクセス許可を追加したら、事前設定された S3 ブループリントを使用してパイプラインを作成します。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

s3 ソース設定内の queue_url オプションを指定する必要があります (これは、読み取り元の Amazon SQS キューの URL です)。URL をフォーマットするには、サブスクライバー設定で [サブスクリプションエンドポイント] を探し、arn:aws: を https:// に変更します。例えば https://sqs.*{region}*.amazonaws.com/*{account-id}*/AmazonSecurityLake-*abdcef*-Main-Queue です。

S3 ソース設定内で指定した sts_role_arn は、パイプラインロールの ARN でなければなりません。

Fluent Bit OpenSearch での取り込みパイプラインの使用

このサンプル Fluent Bit 設定ファイルは、Fluent Bit から Ingestion OpenSearch パイプラインにログデータを送信します。ログデータの取り込みの詳細については、Data Prepper ドキュメントの「ログ分析」を参照してください。

次の点に注意してください。

- host の値はパイプラインエンドポイントにする必要があります。例えば *pipeline-endpoint*.us-east-1.osis.amazonaws.com です。
- aws_service の値は osis にする必要があります。
- aws_role_arn 値は、クライアントが引き受け、署名バージョン 4 認証に使用する AWS IAM ロールの ARN です。

```
[INPUT]
  name                  tail
  refresh_interval      5
  path                  /var/log/test.log
  read_from_head        true

[OUTPUT]
  Name http
  Match *
  Host pipeline-endpoint.us-east-1.osis.amazonaws.com
```

```
  Port 443
  URI /log/ingest
  Format json
  aws_auth true
  aws_region us-east-1
  aws_service osis
  aws_role_arn arn:aws:iam::{account-id}:role/ingestion-role
  Log_Level trace
  tls On
```

その後、HTTP OpenSearch をソースとする Ingestion パイプラインを次のように設定できます。

```
version: "2"
unaggregated-log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log:
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
    - grok:
        match:
          details:
            - "'%{NOTSPACE:http_method} %{NOTSPACE:http_uri}' %{NOTSPACE:protocol}"
            - "TLS%{NOTSPACE:tls_version} %{GREEDYDATA:encryption}"
            - "%{NUMBER:status_code:int} %{NUMBER:response_size:int}"
    - delete_entries:
        with_keys: ["details", "log"]

  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index: "index_name"
        index_type: custom
        bulk_size: 20
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

Fluentd OpenSearch での取り込みパイプラインの使用

Fluentd は、Fluent Bit などのさまざまな言語やサブプロジェクト用の SDKs を提供するオープンソースのデータ収集エコシステムです。このサンプル Fluentd 設定ファイルは、Fluentd から Ingestion OpenSearch パイプラインにログデータを送信します。ログデータの取り込みの詳細については、Data Prepper ドキュメントの「ログ分析」を参照してください。

次の点に注意してください。

- endpoint の値はパイプラインエンドポイントにする必要があります。例えば *pipeline-endpoint*.us-east-1.osis.amazonaws.com/apache-log-pipeline/logs です。
- aws_service の値は osis にする必要があります。
- aws_role_arn 値は、クライアントが引き受け、署名バージョン 4 認証に使用する AWS IAM ロールの ARN です。

```
<source>
  @type tail
  path logs/sample.log
  path_key log
  tag apache
  <parse>
    @type none
  </parse>
</source>

<filter apache>
  @type record_transformer
  <record>
    log ${record["message"]}
  </record>
</filter>

<filter apache>
  @type record_transformer
  remove_keys message
</filter>

<match apache>
  @type http
  endpoint pipeline-endpoint.us-east-1.osis.amazonaws.com/apache-log-pipeline/logs
  json_array true
```

```
  <auth>
    method aws_sigv4
    aws_service osis
    aws_region us-east-1
    aws_role_arn arn:aws:iam::{account-id}:role/ingestion-role
  </auth>

  <format>
    @type json
  </format>

  <buffer>
    flush_interval 1s
  </buffer>
</match>
```

その後、HTTP OpenSearch をソースとする Ingestion パイプラインを次のように設定できます。

```
version: "2"
apache-log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log:
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index: "index_name"
        aws_sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
        aws_region: "us-east-1"
        aws_sigv4: true
```

OpenTelemetry Collector OpenSearch での取り込みパイプラインの使用

このサンプルOpenTelemetry 設定ファイルは、コレクターから OpenTelemetryトレースデータをエクスポートし、 OpenSearch 取り込みパイプラインに送信します。トレースデータの取り込みの詳細については、Data Prepper ドキュメントの「トレース分析」を参照してください。

次の点に注意してください。

- endpoint の値にはパイプラインエンドポイントを含める必要があります。例えば https://*pipeline-endpoint*.us-east-1.osis.amazonaws.com です。
- service の値は osis にする必要があります。
- OTLP/HTTP Exporter の compressionオプションは、パイプラインの OpenTelemetry ソースの compressionオプションと一致する必要があります。

```
extensions:
  sigv4auth:
    region: "us-east-1"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/v1/traces"
    auth:
      authenticator: sigv4auth
    compression: none

service:
  extensions: [sigv4auth]
  pipelines:
    traces:
      receivers: [jaeger]
      exporters: [otlphttp]
```

その後、OTel OpenSearch トレースプラグインをソースとして指定する Ingestion パイプラインを次のように設定できます。 OTel

```
version: "2"
otel-trace-pipeline:
  source:
    otel_trace_source:
      path: "/v1/traces"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace-pipeline"
    - pipeline:
        name: "service-map-pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"

service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-service-map
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

別のパイプラインの例については、事前設定されたトレース分析ブループリントを参照してください。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

次のステップ

データをパイプラインにエクスポートしたら、パイプラインのシンクとして設定された OpenSearch サービスドメインからデータをクエリできます。次のリソースを参照してください。

- オブザーバビリティ
- the section called “トレース分析”
- the section called “パイプ処理言語”

Amazon OpenSearch Ingestion を使用してドメインとコレクション間でデータを移行する

OpenSearch Ingestion パイプラインを使用して、Amazon OpenSearch Service OpenSearch ドメインまたはサーバーレス VPC コレクション間でデータを移行できます。そのためには、1 つのドメインまたはコレクションをソースとして設定し、別のドメインまたはコレクションをシンクとして設定するパイプラインを設定します。これにより、あるドメインまたはコレクションから別のドメインまたはコレクションにデータが効果的に移行されます。

データを移行するには、以下のリソースが必要です。

- OpenSearch OpenSearch ソースサービスドメインまたはサーバーレス VPC コレクション。このドメインまたはコレクションには、移行するデータが含まれています。ドメインを使用している場合は、 OpenSearch バージョン 1.0 以降、または Elasticsearch バージョン 7.4 以降を実行している必要があります。ドメインには、パイプラインロールに適切な権限を付与するアクセスポリシーも必要です。
- データの移行先となる別のドメインまたは VPC コレクション。このドメインまたはコレクションはパイプラインシンクとして機能します。
- OpenSearch Ingestion がコレクションまたはドメインの読み取りと書き込みに使用するパイプラインロール。このロールの Amazon リソースネーム (ARN) をパイプライン設定に含めます。詳細については、以下のリソースを参照してください。
  - the section called “パイプラインにドメインへのアクセス権を付与する”
  - the section called “パイプラインにコレクションへのアクセスを許可する”

トピック

制限事項

OpenSearch OpenSearch サービスドメインまたはサーバーレスコレクションをシンクとして指定する場合、以下の制限が適用されます。

- パイプラインは複数の VPC ドメインに書き込むことはできません。
- VPC OpenSearch アクセスを使用するサーバーレスコレクションとの間でのみデータを移行できます。パブリックコレクションはサポートされていません。
- 1 つのパイプライン設定で VPC とパブリックドメインの組み合わせを指定することはできません。
- 1 つのパイプライン設定には、パイプライン以外のシンクを最大 20 個設定できます。
- 1 AWS リージョン つのパイプライン設定で最大 3 つの異なるシンクを指定できます。
- 複数のシンクがあるパイプラインでは、いずれかのシンクが長時間ダウンしていたり、受信データを受信するのに十分な容量がプロビジョニングされていなかったりすると、時間の経過とともに処理速度が低下する可能性があります。

OpenSearch ソースとしてのサービス

ソースとして指定したドメインまたはコレクションは、データの移行元です。

IAM でパイプラインロールを作成する

OpenSearch Ingestion パイプラインを作成するには、まずパイプラインロールを作成して、ドメインまたはコレクション間の読み取り/書き込みアクセスを許可する必要があります。これを作成するには、次のステップを実行します。

1. IAM で新しいアクセス権限ポリシーを作成し、パイプラインロールにアタッチします。ソースからの読み取りとシンクへの書き込みの権限を必ず許可してください。 OpenSearch サービスドメインの IAM パイプライン権限の設定について詳しくは、「」the section called “パイプラインにド

メインへのアクセス権を付与する” と the section called “パイプラインにコレクションへのアクセスを許可する”「」を参照してください。

2. ソースから読み取りを行うには、パイプラインロール内で以下の権限を指定します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"es:ESHttpGet",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_cat/indices",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/scroll",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpPost",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/
point_in_time",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/scroll"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpDelete",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/
point_in_time",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/scroll"
         ]
      }
   ]
}
```

パイプラインの作成

ポリシーをパイプラインロールにアタッチしたら、AWSOpenSearchDataMigrationPipeline移行ブループリントを使用してパイプラインを作成します。このブループリントには、 OpenSearch サービスドメインまたはコレクション間でデータを移行するためのデフォルト設定が含まれています。詳細については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

> **Note**
>
> OpenSearch 取り込みでは、ソースドメインのバージョンとディストリビューションを使用して、移行に使用するメカニズムを決定します。`point_in_time`一部のバージョンではこのオプションがサポートされています。 OpenSearch `point_in_timescroll`サーバーレスはまたはをサポートしていないため、`search_after`このオプションを使用します。

移行プロセス中に新しいインデックスが作成中である場合や、移行中にドキュメントの更新が進行中である場合があります。そのため、新しいデータや更新されたデータを取得するために、ドメインインデックスデータを 1 回だけでなく複数回スキャンする必要がある場合があります。

パイプライン設定で `index_read_count` と `interval` を設定して、スキャンを実行する回数を指定します。次の例では、複数回のスキャンを実行する方法を示しています。

```
scheduling:
    interval: "PT2H"
    index_read_count: 3
    start_time: "2023-06-02T22:01:30.00Z"
```

OpenSearch Ingestion では以下の設定を使用して、データが同じインデックスに書き込まれ、同じドキュメント ID が維持されるようにします。

```
index: "${getMetadata(\"opensearch-index\")}"
document_id: "${getMetadata(\"opensearch-document_id\")}"
```

OpenSearch 複数のサービスドメインシンクを指定する

データの送信先として、 OpenSearch 複数のパブリックサービスドメインを指定できます。この機能を使用して、条件付きルーティングを実行したり、 OpenSearch 受信データを複数のサービスド

メインに複製したりできます。最大 10 OpenSearch 個の異なるパブリックサービスドメインをシンクとして指定できます。

以下の例では、 OpenSearch 受信データは条件付きで異なるサービスドメインにルーティングされます。

```
...
  route:
    - 2xx_status: "/response >= 200 and /response < 300"
    - 5xx_status: "/response >= 500 and /response < 600"
  sink:
    - opensearch:
        hosts: [ "https://search-response-2xx.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          region: "us-east-1"
        index: "response-2xx"
        routes:
          - 2xx_status
    - opensearch:
        hosts: [ "https://search-response-5xx.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          region: "us-east-1"
        index: "response-5xx"
        routes:
          - 5xx_status
```

OpenSearch サーバーレス VPC コレクションへのデータの移行

OpenSearch Ingestion を使用して、 OpenSearch OpenSearch ソースのサービスドメインまたはサーバーレスコレクションから VPC コレクションシンクにデータを移行できます。パイプライン設定内にネットワークアクセスポリシーを指定する必要があります。 OpenSearch サーバーレス VPC コレクションへのデータ取り込みの詳細については、を参照してください。the section called “チュートリアル: コレクションにデータを取り込む”

データを VPC コレクションに移行するには

1. OpenSearch サーバーレスコレクションを作成します。手順については、「the section called “チュートリアル: コレクションにデータを取り込む”」を参照してください。

2. コレクションエンドポイントと Dashboard エンドポイントの両方に対する VPC アクセスを指定するコレクションのネットワークポリシーを作成します。手順については、「the section called “ネットワークアクセス”」を参照してください。
3. パイプラインロールをまだ持っていない場合は作成します。手順については、「the section called “パイプラインロール”」を参照してください。
4. パイプラインを作成します。手順については、「the section called “ブループリントを使用したパイプラインの作成”」を参照してください。

Amazon OpenSearch Ingestion を操作するための AWS SDKの使用

このセクションには、AWS SDK を使用して Amazon OpenSearch Ingestion を操作する方法の例が含まれています。コード例は、ドメインとパイプラインを作成し、パイプラインにデータを取り込む方法を示しています。

トピック

- Python

Python

次のサンプルスクリプトでは、AWS SDK for Python (Boto3) を使用して IAM パイプラインロール、データ書き込み先ドメイン、およびデータを取り込むパイプラインを作成します。次に、`requests` HTTP ライブラリを使用してサンプルログファイルをパイプラインに取り込みます。

必要な従属関係をインストールには、次のコマンドを実行します。

```
pip install boto3
pip install botocore
pip install requests
pip install requests-auth-aws-sigv4
```

スクリプト内で、アクセスポリシーのアカウント ID を自分の AWS アカウント ID に置き換えます。必要に応じて、`region` を変更することもできます。

```
import boto3
import botocore
from botocore.config import Config
```

```
import requests
from requests_auth_aws_sigv4 import AWSSigV4
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

opensearch = boto3.client('opensearch', config=my_config)
iam = boto3.client('iam', config=my_config)
osis = boto3.client('osis', config=my_config)

domainName = 'test-domain'  # The name of the domain
pipelineName = 'test-pipeline' # The name of the pipeline

def createPipelineRole(iam, domainName):
    """Creates the pipeline role"""
    response = iam.create_policy(
        PolicyName='pipeline-policy',
        PolicyDocument=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect
\":\"Allow\",\"Action\":\"es:DescribeDomain\",\"Resource\":\"arn:aws:es:us-
east-1:123456789012:domain\/{domainName}\"}},{{\"Effect\":\"Allow\",\"Action\":
\"es:ESHttp*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain\/{domainName}\/*
\"}}]}}'
    )
    policyarn = response['Policy']['Arn']

    response = iam.create_role(
        RoleName='PipelineRole',
        AssumeRolePolicyDocument='{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"Service\":\"osis-pipelines.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}]}'
    )
    rolename=response['Role']['RoleName']

    response = iam.attach_role_policy(
        RoleName=rolename,
        PolicyArn=policyarn
    )
```

```
    print('Creating pipeline role...')
    time.sleep(10)
    print('Role created: ' + rolename)

def createDomain(opensearch, domainName):
    """Creates a domain to ingest data into"""
    response = opensearch.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_2.3',
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect\":
\"Allow\",\"Principal\":{{\"AWS\":\"arn:aws:iam::123456789012:role\/PipelineRole
\"}},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain\/
{domainName}\/*\"}}]}}',
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    return(response)

def waitForDomainProcessing(opensearch, domainName):
    """Waits for the domain to be active"""
    try:
        response = opensearch.describe_domain(
            DomainName=domainName
        )
        # Every 30 seconds, check whether the domain is processing.
        while 'Endpoint' not in response['DomainStatus']:
            print('Creating domain...')
            time.sleep(60)
            response = opensearch.describe_domain(
```

```
                DomainName=domainName)

        # Once we exit the loop, the domain is ready for ingestion.
        endpoint = response['DomainStatus']['Endpoint']
        print('Domain endpoint ready to receive data: ' + endpoint)
        createPipeline(osis, endpoint)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found.')
        else:
            raise error

def createPipeline(osis, endpoint):
    """Creates a pipeline using the domain and pipeline role"""
    try:
        definition = f'version: \"2\"\nlog-pipeline:\n  source:\n    http:\n      path:
 \"/${{pipelineName}}/logs\"\n  processor:\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://{endpoint}\" ]\n        index: \"application_logs\"\n        aws:\n
     sts_role_arn: \"arn:aws:iam::123456789012:role/PipelineRole\"\n          region:
 \"us-east-1\"'
        response = osis.create_pipeline(
            PipelineName=pipelineName,
            MinUnits=4,
            MaxUnits=9,
            PipelineConfigurationBody=definition
        )

        response = osis.get_pipeline(
                PipelineName=pipelineName
        )

        # Every 30 seconds, check whether the pipeline is active.
        while response['Pipeline']['Status'] == 'CREATING':
            print('Creating pipeline...')
            time.sleep(30)
            response = osis.get_pipeline(
                PipelineName=pipelineName)

        # Once we exit the loop, the pipeline is ready for ingestion.
        ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
        print('Pipeline ready to ingest data at endpoint: ' + ingestionEndpoint)
        ingestData(ingestionEndpoint)
```

```
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceAlreadyExistsException':
            print('Pipeline already exists.')
            response = osis.get_pipeline(
                PipelineName=pipelineName
            )
            ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
            ingestData(ingestionEndpoint)
        else:
            raise error


def ingestData(ingestionEndpoint):
    """Ingests a sample log file into the pipeline"""
    endpoint = 'https://' + ingestionEndpoint
    r = requests.request('POST', f'{endpoint}/log-pipeline/logs',

 data='[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","requ
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]',
    auth=AWSSigV4('osis'))
    print('Ingesting sample log file into pipeline')
    print('Response: ' + r.text)

def main():
    createPipelineRole(iam, domainName)
    createDomain(opensearch, domainName)
    waitForDomainProcessing(opensearch, domainName)

if __name__ == "__main__":
    main()
```

Amazon OpenSearch Ingestion のセキュリティ

AWS では、クラウドのセキュリティが最優先事項です。AWS の顧客は、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から利点を得られます。

セキュリティは、AWS と顧客の間の責任共有です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS の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、使用するサービスを安全に提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査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ています。
- 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 のサービスに応じて異なります。また、ユーザーは、データの機密性、企業要件、および適用法令と規制などのその他要因に対する責任も担います。

このドキュメントは、OpenSearch Ingestion を使用するときに、責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するように OpenSearch Ingestion を設定する方法について説明します。また、OpenSearch Ingestion リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

トピック

- Amazon Ingestion OpenSearch パイプラインの VPC アクセスの設定
- Amazon OpenSearch Ingestion の Identity and Access Management
- AWS CloudTrail を使用した Amazon OpenSearch Ingestion API コールのログ記録

Amazon Ingestion OpenSearch パイプラインの VPC アクセスの設定

インターフェイス VPC エンドポイントを使用して Amazon OpenSearch Ingestion パイプラインにアクセスできます。VPC は、 専用の仮想ネットワークです AWS アカウント。 AWS クラウド内の他の仮想ネットワークから論理的に分離されます。VPC エンドポイントを介してパイプラインにアクセスすると、インターネットゲートウェイ、NAT デバイス、VPN OpenSearch 接続を必要とせずに、VPC 内の取り込みと他の のサービス間の安全な通信が可能になります。すべてのトラフィックは AWS クラウド内で安全に保持されます。

OpenSearch 取り込みは、 を使用するインターフェイスエンドポイント を作成することで、このプライベート接続を確立します AWS PrivateLink。パイプラインの作成時に指定した各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、 OpenSearch 取り込みパイプライン宛てのトラフィックのエントリポイントとして機能するリクエスタ管理のネットワークインターフェイスです。また、インターフェイスエンドポイントを自分で作成および管理することもできます。

VPC を使用すると、パブリックインターネットではなく VPC OpenSearch の境界内の取り込みパイプラインを介してデータフローを適用できます。VPC 内にないパイプラインは、一般向けのエンドポイントとインターネットを経由してデータを送受信します。

VPC アクセスを持つパイプラインは、パブリックドメインまたは VPC OpenSearch サービスドメイン、およびパブリックコレクションまたは VPC OpenSearch サーバーレスコレクションに書き込むことができます。

トピック

考慮事項

パイプラインの VPC アクセスを設定する際は、以下の点に注意してください。

- パイプラインはシンクと同じ VPC に存在する必要はありません。また、2 つの VPCs間の接続を確立する必要はありません。Ingestion OpenSearch が接続を処理します。
- パイプラインに指定できる VPC は 1 つのみです。
- パブリックパイプラインとは異なり、VPC パイプラインは書き込み先のドメインまたはコレクションシンクと同じ AWS リージョン にある必要があります。
- パイプラインは、VPC の 1 つ、2 つ、3 つのいずれかのサブネットにデプロイできます。サブネットは、Ingestion OpenSearch Compute Units (OCUs) がデプロイされているのと同じアベイラビリティーゾーンに分散されます。
- パイプラインを 1 つのサブネットにのみデプロイした場合、そのアベイラビリティーゾーンがダウンするとデータを取り込めなくなります。高可用性を確保するために、パイプラインを 2 つまたは 3 つのサブネットで構成することが推奨されます。
- セキュリティグループの指定はオプションです。セキュリティグループを指定しない場合、Ingestion OpenSearch は VPC で指定されたデフォルトのセキュリティグループを使用します。

制限事項

VPC アクセスを持つパイプラインには、次の制限があります。

- パイプラインの作成後は、パイプラインのネットワーク設定を変更できません。VPC 内でパイプラインを起動した場合、後からこれをパブリックエンドポイントに変更することはできず、その逆もまた同様です。
- インターフェイス VPC エンドポイントまたはパブリックエンドポイントを使用してパイプラインを起動することはできますが、両方を行うことはできません。パイプラインを作成するときに、いずれかを選択する必要があります。
- VPC アクセスを使用してパイプラインをプロビジョニングした後は、別の VPC に移動したり、サブネットやセキュリティグループ設定を変更したりすることはできません。
- パイプラインが VPC アクセスを使用するドメインまたはコレクションシンクに書き込む場合、パイプラインの作成後に後で戻ってシンク (VPC またはパブリック) を変更することはできません。そのパイプラインを削除し、新しいシンクを使って改めて作成する必要があります。パブリックシンクから VPC アクセスのあるシンクに切り替えることはできます。
- VPC パイプラインへのアカウントを横断した取り込みアクセスを提供することはできません。

前提条件

VPC アクセスを使用してパイプラインをプロビジョニングする前に、以下を実行する必要があります。

- 「VPC を作成する」

  VPC を作成するには、Amazon VPC コンソール、 AWS CLI、または AWS SDKsのいずれかを使用できます。詳細については、Amazon VPC ユーザーガイドの「VPC の使用」を参照してください。VPC が既にある場合、このステップは省略できます。

- IP アドレスのリザーブ

  OpenSearch 取り込みでは、パイプラインの作成時に指定した各サブネットに Elastic Network Interface を配置します。各ネットワークインターフェースは 1 つの IP アドレスに関連付けられます。ネットワークインターフェイスのサブネットごとに 1 つの IP アドレスを予約する必要があります。

パイプラインの VPC アクセスを設定する

OpenSearch サービスコンソール内または を使用して、パイプラインの VPC アクセスを有効にできます AWS CLI。

コンソール

VPC アクセスの設定は、パイプラインの作成時に行います。[ネットワーク] で [VPC アクセス] を選択し、次のように設定します。

設定	説明
エンドポイント管理	VPC エンドポイントを自分で作成するか、 OpenSearch 取り込みで作成するかを選択します。
VPC	使用する仮想プライベートクラウド (VPC) の ID を選択します。VPC とパイプラインは同じ AWS リージョンの中になければなりません。
サブネット	1 つ以上のサブネットを選択します。 OpenSearch サービスは VPC エンドポイントと Elastic Network Interface をサブネットに配置します。
セキュリティグループ	必要なアプリケーションがパイプラインによって公開されるポート (80 または OpenSearch 443) とプロトコル (HTTP または HTTP) の Ingestion パイプラインに到達できるようにする HTTPs1 つ以上選択します。
VPC アタッチメントオプション	ソースがセルフマネージドエンドポイントの場合は、パイプラインを VPC にアタッチします。提供されているデフォルトの CIDR オプションのいずれかを選択するか、カスタム CIDR を使用します。

CLI

を使用して VPC アクセスを設定するには AWS CLI、 `--vpc-options`パラメータを指定します。

```
aws osis create-pipeline \
  --pipeline-name vpc-pipeline \
  --min-units 4 \
  --max-units 10 \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
```

```
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

セルフマネージド VPC エンドポイント

パイプラインを作成するときは、エンドポイント管理を使用して、セルフマネージドエンドポイントまたはサービスマネージドエンドポイントでパイプラインを作成できます。エンドポイント管理はオプションであり、デフォルトでは Ingestion OpenSearch によって管理されるエンドポイントになります。

でセルフマネージド VPC エンドポイントを使用してパイプラインを作成するには AWS Management Console、 OpenSearch 「サービスコンソール を使用したパイプラインの作成」を参照してください。でセルフマネージド VPC エンドポイントを使用してパイプラインを作成するには AWS CLI、create-pipeline コマンドで `--vpc-options`パラメータを使用できます。

```
--vpc-options SubnetIds=subnet-abcdef01234567890,VpcEndpointManagement=CUSTOMER
```

エンドポイントサービスを指定するときに、パイプラインへのエンドポイントを自分で作成できます。エンドポイントサービスを検索するには、get-pipeline コマンドを使用します。このコマンドは、次のようなレスポンスを返します。

```
"vpcEndpointService" : "com.amazonaws.osis.us-east-1.pipeline-
id-1234567890abcdef1234567890",
"vpcEndpoints" : [
  {
    "vpcId" : "vpc-1234567890abcdef0",
    "vpcOptions" : {
      "subnetIds" : [ "subnet-abcdef01234567890", "subnet-021345abcdef6789" ],
      "vpcEndpointManagement" : "CUSTOMER"
    }
  }
```

レスポンス`vpcEndpointService`の を使用して、 AWS Management Console または で VPC エンドポイントを作成します AWS CLI。

セルフマネージド VPC エンドポイントを使用する場合は、VPC `enableDnsHostnames`で DNS 属性 `enableDnsSupport` と を有効にする必要があります。を停止して再起動するセルフマネージドエンドポイントを持つパイプラインがある場合は、アカウントに VPC エンドポイントを再作成する必要があることに注意してください。

VPC アクセス用の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、サービスに権限を委任する一意のタイプの IAM ロールであり、ユーザーに代わってリソースを作成して管理できます。サービスマネージド VPC エンドポイントを選択した場合、Ingestion OpenSearch では、VPC にアクセスし、パイプラインエンドポイントを作成し、VPC のサブネットにネットワークインターフェイスを配置AWSServiceRoleForAmazonOpenSearchIngestionServiceするために、 と呼ばれるサービスにリンクされたロールが必要です。

セルフマネージド VPC エンドポイントを選択した場合、Ingestion OpenSearch には というサービスにリンクされたロールが必要ですAWSServiceRoleForOpensearchIngestionSelfManagedVpce。これらのロール、そのアクセス許可、および削除方法の詳細については、「」を参照してくださいthe section called “パイプライン作成ロール”。

OpenSearch 取り込みパイプラインを作成すると、取り込みによってロールが自動的に作成されます。この自動作成が成功するには、アカウントに最初のパイプラインを作成するユーザーが、`iam:CreateServiceLinkedRole` アクションに対するアクセス許可を持っている必要があります。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許可」を参照してください。ロールは、作成後に AWS Identity and Access Management (IAM) コンソールで表示できます。

Amazon OpenSearch Ingestion 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 Ingestion リソースの使用を承認する (アクセス許可を付与する) OpenSearch 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

- OpenSearch 取り込み用のアイデンティティベースのポリシー
- OpenSearch 取り込みのポリシーアクション
- OpenSearch 取り込みのポリシーリソース
- Amazon OpenSearch Ingestion のポリシー条件キー
- OpenSearch 取り込みによる ABAC
- OpenSearch 取り込みでの一時的な認証情報の使用
- OpenSearch Ingestion のサービスにリンクされたロール

- OpenSearch 取り込みのアイデンティティベースのポリシーの例

OpenSearch 取り込み用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	あり

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学ぶには、IAM ユーザーガイドのIAM JSON ポリシーの要素のリファレンスを参照してください。

OpenSearch 取り込みのアイデンティティベースのポリシーの例

取り込みアイデンティティベースのポリシーの例を表示するには、 OpenSearch 「」を参照してくださいthe section called “アイデンティティベースポリシーの例”。

OpenSearch 取り込み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	あり

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。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Ingestion OpenSearch のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
osis
```

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": [
      "osis:action1",
      "osis:action2"
         ]
```

ワイルドカード文字 (*) を使用すると、複数のアクションを指定することができます。例えば、List という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "osis:List*"
```

取り込みアイデンティティベースのポリシーの例を表示するには、 OpenSearch 「」を参照してください OpenSearch Serverless での ID ベースのポリシー例。

OpenSearch 取り込み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	あり

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。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

Amazon OpenSearch Ingestion 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	なし

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 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

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

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。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。

取り込み条件キーのリストを確認するには、 OpenSearch 「サービス認証リファレンス」の「Amazon OpenSearch Ingestion の条件キー」を参照してください。条件キーを使用できるアクションとリソースについては、「Amazon Ingestion OpenSearch で定義されるアクション」を参照してください。

OpenSearch 取り込みによる 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ありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、IAM ユーザーガイドのABAC 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に基づくアクセスコントロール (ABAC) を使用するを参照してください。

取り込みリソース OpenSearch のタグ付けの詳細については、「」を参照してくださいthe section called “パイプラインのタグ付け”。

OpenSearch 取り込みでの一時的な認証情報の使用

一時的な認証情報のサポート	あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する 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 にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、IAM ユーザーガイドの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。

OpenSearch Ingestion のサービスにリンクされたロール

サービスリンクロールのサポート	あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

OpenSearch 取り込みでは、 というサービスにリンクされたロールを使用しますAWSServiceRoleForAmazonOpenSearchIngestionService。という名前のサービスにリンクされたロールAWSServiceRoleForOpensearchIngestionSelfManagedVpceは、セルフマネージド VPC エンドポイントを持つパイプラインでも使用できます。 OpenSearch Ingestion サービスにリンクされたロールの作成と管理の詳細については、「」を参照してくださいthe section called “パイプライン作成ロール”。

OpenSearch 取り込みのアイデンティティベースのポリシーの例

デフォルトでは、ユーザーとロールには Ingestion OpenSearch リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

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

各リソースタイプの ARN の形式など、Amazon OpenSearch Ingestion で定義されるアクションとリソースタイプの詳細については、「サービス認証リファレンス」の「Amazon OpenSearch Ingestion 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。 ARNs

トピック

- ポリシーのベストプラクティス
- コンソールでの OpenSearch Ingestion の使用
- OpenSearch 取り込みパイプラインの管理

- 取り込みパイプラインへのデータの OpenSearch 取り込み

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ポリシーは非常に強力です。アカウント内で誰かが Ingestion OpenSearch リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

ID ベースのポリシーは、ユーザーのアカウントで誰かが Ingestion OpenSearch リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可を付与するには、多くの一般的なユースケースにアクセス許可を付与する AWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可をさらに減らすことをお勧めします。詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」または「ジョブ機能のAWS マネージドポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイドのIAM でのポリシーとアクセス許可を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、 などの特定の を通じ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。

コンソールでの OpenSearch Ingestion の使用

OpenSearch サービスコンソール内の OpenSearch 取り込み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により、 AWS アカウントの OpenSearch 取り込みリソースの詳細を一覧表示および表示できます。必要最低限の許可よりも制限が厳しいアイデンティティベース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 (IAM ロールなど) に対してコンソールが意図したとおりに機能しなくなります。

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が許可されます。

次のポリシーでは、ユーザーが OpenSearch サービスコンソール内の Ingestion OpenSearch にアクセスすることを許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
    ]
}
```

または、 のすべての取り込みリソースへの読み取り専用アクセスを許可する OpenSearch the section called “AmazonOpenSearchIngestionReadOnlyAccess” AWS マネージドポリシーを使用することもできます AWS アカウント。

OpenSearch 取り込みパイプラインの管理

このポリシーは、ユーザーが Amazon OpenSearch Ingestion パイプラインを管理および管理できるようにする「パイプライン管理者」ポリシーの例です。ユーザーはパイプラインの作成、表示、削除を行うことができます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:CreatePipeline",
                "osis:DeletePipeline",
                "osis:UpdatePipeline",
                "osis:ValidatePipeline",
                "osis:StartPipeline",
                "osis:StopPipeline"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ],
            "Effect": "Allow"
        }
    ]
}
```

取り込みパイプラインへのデータの OpenSearch 取り込み

このポリシー例では、ユーザーまたは他のエンティティが自分のアカウントの Amazon Ingestion OpenSearch パイプラインにデータを取り込むことを許可します。ユーザーがパイプラインを変更することはできません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CloudTrail を使用した Amazon OpenSearch Ingestion API コールのログ記録

Amazon OpenSearch Ingestion は、AWS CloudTrail (ユーザー、ロール、または OpenSearch Ingestion 内で AWS のサービスが実行したアクションを記録するためのサービス) と統合されています。

CloudTrail は、OpenSearch Ingestion のすべての API コールをイベントとしてキャプチャします。キャプチャされる対象には、OpenSearch Service コンソールの OpenSearch Ingestion セクションからの呼び出しや、OpenSearch Ingestion API オペレーションへのコード呼び出しなどが含まれます。

証跡を作成すると、OpenSearch Ingestion のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にできます。追跡を設定しない場合でも、CloudTrail コンソールの [Event history] (イベント履歴) で最新のイベントを表示できます。

CloudTrail で収集された情報を使用すると、OpenSearch Ingestion に対して発行されたリクエスト、リクエスト元の IP アドレス、リクエスト作成者、リクエスト作成日時、その他の詳細情報などを確認できます。

CloudTrail の詳細については、AWS CloudTrail ユーザーガイドを参照してください。

CloudTrail での OpenSearch Ingestion の情報

CloudTrail は、アカウント作成時に AWS アカウント で有効になります。OpenSearch Ingestion でアクティビティが発生すると、そのアクティビティは、[イベント履歴] にある別の AWS サービスのイベントとともに、CloudTrail イベントに記録されます。最近のイベントは、AWS アカウント で表示、検索、ダウンロードできます。詳細については、 CloudTrail イベント履歴でのイベントの表示を参照してください。

OpenSearch Ingestion のイベントを含む AWS アカウント のイベントを継続的に記録するには、証跡を作成します。追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての 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 ログで収集し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ためにその他の AWS の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。

- 「追跡を作成するための概要」
- CloudTrail が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 CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての OpenSearch Ingestion アクションは CloudTrail によってログが記録されます。それらのアクションは「OpenSearch Ingestion API リファレンス」に記載されています。例えば、`CreateCollection`、`ListCollections`、`DeleteCollection` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー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ティッドユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

OpenSearch Ingestion のログファイルエントリの理解

[トレイル] は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように構成できます。CloudTrail ログファイルには、1 つ以上のログエントリがあります。

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ます。これには、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。

次の例は、DeletePipelineアクションを示す CloudTrail ログエントリです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/test-user",
        "accountId": "123456789012",
        "accessKeyId": "access-key",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-04-21T16:48:33Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-04-21T16:49:22Z",
    "eventSource": "osis.amazonaws.com",
    "eventName": "UpdatePipeline",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
```

```
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36",
    "requestParameters": {
        "pipelineName": "my-pipeline",
        "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
 http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:\n
        log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received: true
\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n"
    },
    "responseElements": {
        "pipeline": {
            "pipelineName": "my-pipeline",sourceIPAddress
            "pipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/my-pipeline",
            "minUnits": 1,
            "maxUnits": 1,
            "status": "UPDATING",
            "statusReason": {
                "description": "An update was triggered for the pipeline. It is still
 available to ingest data."
            },
            "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
    http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:
\n          log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n",
            "createdAt": "Mar 29, 2023 1:03:44 PM",
            "lastUpdatedAt": "Apr 21, 2023 9:49:21 AM",
            "ingestEndpointUrls": [
                "my-pipeline-tu33ldsgdltgv7x7tjqiudvf7m.us-west-2.osis.amazonaws.com"
            ]
        }
    },
    "requestID": "12345678-1234-1234-1234-987654321098",
    "eventID": "12345678-1234-1234-1234-987654321098",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
```

```
    "recipientAccountId": "709387180454",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "osis.us-west-2.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

Amazon OpenSearch Ingestion パイプラインのタグ付け

タグを使用すると、Amazon OpenSearch Ingestion パイプラインに任意の情報を割り当て、その情報を分類しフィルタリングすることができます。タグとは、ユーザーまたは AWS が AWS リソースに割り当てるメタデータラベルです。各タグは、キーと値から構成されます。ユーザーが割り当てるタグでは、ユーザーがキーと値を定義します。たとえば、1 つのリソースのキーを `stage` と定義し、値を `test` と定義します。

タグは、以下のことに役立ちます。

- AWS リソースの特定と整理。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 OpenSearch Service ドメインに割り当てる OpenSearch Ingestion パイプラインに、同じタグを割り当てることができます。
- AWS のコストの追跡。これらのタグは、AWS Billing and Cost Management ダッシュボードで有効にします。AWS では、タグを使用してコストを分類し、毎月のコスト配分レポートを提供します。詳細については、「AWS Billing ユーザーガイド」の「Use Cost Allocation Tags」(コスト配分タグの使用) を参照してください。
- 属性ベースのアクセス制御を使用して、パイプラインへのアクセスを制限します。詳細については、IAM ユーザーガイドのタグキーに基づいたアクセス制御を参照してください。

OpenSearch Ingestion では、主要なリソースはパイプラインです。OpenSearch Service コンソール、AWS CLI、OpenSearch Ingestion API、AWS SDK を使用することで、パイプラインのタグを追加、管理、削除できます。

トピック

- 必要な許可
- タグの操作 (コンソール)

- タグの操作 (AWS CLI)

必要な許可

OpenSearch Ingestion は、パイプラインのタグ付けに次の AWS Identity and Access Management Access Analyzer (IAM) アクセス許可を使用します。

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

各アクセス許可の詳細については、「サービス認証リファレンス」の「OpenSearch Ingestion 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。

タグの操作 (コンソール)

コンソールは、パイプラインにタグ付けする際の最も簡単な方法です。

タグを作成するには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. ナビゲーションペインの [取り込み] を選択します。
3. タグを追加するパイプラインを選択し、[タグ] タブに移動します。
4. [管理] を選択して、[新しいタグを追加] を選択します。
5. タグキーとオプションの値を入力します。
6. [Save (保存)] を選択します。

タグを削除するには、同じ手順に従って、[タグを管理] ページで [削除] を選択します。

タグを操作するコンソールを使用する方法の詳細については、AWS マネジメントコンソール入門ガイドの「タグエディター」を参照してください。

タグの操作 (AWS CLI)

AWS CLI を使用してパイプラインにタグを付けるには、`TagResource` リクエストを送信します。

```
aws osis tag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tags Key=service,Value=osis Key=source,Value=otel
```

UntagResource コマンドを使用してパイプラインからタグを削除します。

```
aws osis untag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tag-keys service
```

ListTagsForResource コマンドを使用してパイプラインの既存のタグを表示します。

```
aws osis list-tags-for-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

Amazon CloudWatch を使用した Amazon OpenSearch Ingestion のログ記録とモニタリング

Amazon OpenSearch Ingestion は、メトリクスとログを Amazon CloudWatch にパブリッシュします。

トピック

- パイプラインのログのモニタリング
- パイプラインメトリクスのモニタリング

パイプラインのログのモニタリング

Amazon OpenSearch Ingestion パイプラインのログ記録を有効にすることで、パイプラインのオペレーション中や取り込みアクティビティ中に発生した、エラーや警告メッセージを公開できます。OpenSearch Ingestion は、すべてのログを Amazon CloudWatch Logs にパブリッシュします。CloudWatch Logs は、ログファイル内の情報をモニタリングし、特定のしきい値が満たされたときに通知します。高い耐久性を備えたストレージにログデータをアーカイブすることも可能です。詳細については、「Amazon CloudWatch Logs ユーザーガイド」を参照してください。

OpenSearch Ingestion のログに、リクエスト処理の失敗、送信元からシンクへの認証エラー、その他、トラブルシューティングに役立つ警告が表示されることがあります。OpenSearch Ingestion

は、ログでは、INFO、WARN、ERROR、FATAL のログレベルを使用します。ログのパブリッシュは、すべてのパイプラインで有効にすることが推奨されています。

必要な許可

OpenSearch Ingestion を有効にしてログを CloudWatch Logs へ送信するには、特定の IAM アクセス権限を持つユーザーとしてサインインする必要があります。

ログ配信リソースを作成し更新するには、次の CloudWatch Logs のアクセス権限が必要です。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries"
            ]
        }
    ]
}
```

ログパブリッシュの有効化

ログのパブリッシュは、既存のパイプラインで有効にするか、パイプラインの作成中に有効にすることができます。パイプラインの作成中にログのパブリッシュを有効にする手順については、「the section called “パイプラインの作成”」を参照してください。

コンソール

既存のパイプラインでログのパブリッシュを有効にするには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. ナビゲーションペインで [取り込み] を選択し、ログを有効にするパイプラインを選択します。

3. [ログパブリッシュのオプションを編集] を選択します。
4. [CloudWatch Logs へパブリッシュ] を選択します。
5. 新しいロググループを作成するか、既存のロググループを選択します。名前は、`/aws/vendedlogs/OpenSearchIngestion/pipeline-name/audit-logs` のようなパスの形式にすることが推奨されます。この形式を使用すれば、`/aws/vendedlogs/OpenSearchService/OpenSearchIngestion` のような特定のパスを持つすべてのロググループにアクセス権限を付与する、CloudWatch アクセスポリシーの適用が容易になります。

   > **Important**
   >
   > ロググループ名には、プレフィックス `vendedlogs` を含めます。さもないと作成に失敗します。

6. [Save (保存)] を選択します。

CLI

AWS CLI を使用してログのパブリッシュを有効にするには、次のリクエストを送信します。

```
aws osis update-pipeline \
  --pipeline-name my-pipeline \
  --log-publishing-options  IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="/aws/vendedlogs/OpenSearchIngestion/pipeline-name"}
```

パイプラインメトリクスのモニタリング

Amazon CloudWatch を使って Amazon OpenSearch Ingestion パイプラインをモニタリングすることができます。Amazon CloudWatch が raw データを収集し、読み取り可能な、ほぼリアルタイムのメトリクスに加工します。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をより的確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

OpenSearch Ingestion のコンソールでは、CloudWatch の raw データに基づく一連のグラフが、各パイプラインの [パフォーマンス] タブに表示されます。

OpenSearch Ingestion は、サポートされているほとんどのプラグインのメトリクスを報告します。特定のプラグインに、以下の、固有の表がなければ、プラグインに固有のメトリクスは報告されてい

ないということを意味します。パイプラインのメトリクスは、AWS/OSIS 名前空間にパブリッシュされます。

トピック

- 一般的なメトリクス
- バッファのメトリクス
- Signature V4 メトリクス
- 制限付きブロッキングバッファのメトリクス
- Otel トレースソースメトリクス
- Otel メトリクスソースメトリクス
- HTTP メトリクス
- S3 メトリクス
- メトリクス集約
- 日付メトリクス
- Grok メトリクス
- Otel トレース raw メトリクス
- Otel トレースグループメトリクス
- サービスマップステートフルメトリクス
- OpenSearch メトリクス
- システムと測定メトリクス

一般的なメトリクス

以下は、すべてのプロセッサーとシンクに共通のメトリクスです。

各メトリクスの先頭には、サブパイプライン名とプラグイン名が <*sub_pipeline_name*><*plugin*><*metric_name*> の形式で付きます。例えば、サブパイプライン名が `my-pipeline` である `recordsIn.count` メトリクスと date プロセッサの完全名は `my-pipeline.date.recordsIn.count` になります。

メトリクスのサフィックス	説明
recordsIn.count	パイプラインのコンポーネントへのレコードのイングレス。このメトリクスは、プロセッサとシンクに適用されます。 関連する統計情報: Sum ディメンション:PipelineName
recordsOut.count	パイプラインのコンポーネントへのレコードのエグレス。このメトリクスは、プロセッサと送信元に適用されます。 関連する統計情報: Sum ディメンション:PipelineName
timeElapsed.count	パイプラインコンポーネントの実行中に記録されたデータポイントの数。このメトリクスは、プロセッサとシンクに適用されます。 関連する統計情報: Sum ディメンション:PipelineName
timeElapsed.sum	パイプラインコンポーネントの実行中に経過した合計時間。このメトリクスは、プロセッサとシンクにミリ秒単位で適用されます。 関連する統計情報: Sum ディメンション:PipelineName
timeElapsed.max	パイプラインコンポーネントの実行中に経過した最大時間。このメトリクスは、プロセッサとシンクにミリ秒単位で適用されます。 関連する統計情報: Max ディメンション:PipelineName

バッファのメトリクス

次のメトリクスは、OpenSearch Ingestion がすべてのパイプラインに自動的に設定するデフォルトの制限のあるブロッキングバッファに適用されます。

各メトリクスの先頭には、サブパイプライン名とバッファ名が <*sub_pipeline_name*><*buffer_name*><*metric_name*> の形式で付きます。例えば、サブパイプライン名が `my-pipeline` である `recordsWritten.count` メトリクスの完全名は `my-pipeline.BlockingBuffer.recordsWritten.count` になります。

メトリクスのサフィックス	説明
`recordsWritten.count`	バッファに書き込まれたレコードの数。 関連する統計情報: Sum ディメンション:`PipelineName`
`recordsRead.count`	バッファから読み取られたレコードの数。 関連する統計情報: Sum ディメンション:`PipelineName`
`recordsInFlight.value`	バッファから読み取られた未チェックのレコードの数。 関連する統計: Average ディメンション:`PipelineName`
`recordsInBuffer.value`	現在バッファに入っているレコードの数。 関連する統計: Average ディメンション:`PipelineName`
`recordsProcessed.count`	バッファから読み取られパイプラインによって処理されたレコードの数。 関連する統計情報: Sum ディメンション:`PipelineName`

メトリクスのサフィックス	説明
recordsWriteFailed.count	パイプラインがシンクへの書き込みに失敗したレコードの数。 関連する統計情報: Sum ディメンション:PipelineName
writeTimeElapsed.count	バッファへの書き込み中に記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName
writeTimeElapsed.sum	バッファへの書き込み中に経過した合計時間 (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
writeTimeElapsed.max	バッファへの書き込み中に経過した最大時間 (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
writeTimeouts.count	バッファへの書き込みタイムアウトの数。 関連する統計情報: Sum ディメンション:PipelineName
readTimeElapsed.count	バッファへの読み取り中に記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
readTimeElapsed.sum	バッファからの読み取り中に経過した合計時間 (ミリ秒単位) 。 関連する統計情報: Sum ディメンション:PipelineName
readTimeElapsed.max	バッファからの読み取り中に経過した最大時間 (ミリ秒単位) 。 関連する統計情報: Max ディメンション:PipelineName
checkpointTimeElapsed.count	チェックポイント中に記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName
checkpointTimeElapsed.sum	チェックポイント中に経過した合計時間 (ミリ秒単位)。 関連する統計情報: Sum ディメンション:PipelineName
checkpointTimeElapsed.max	チェックポイント中に経過した最大時間 (ミリ秒単位)。 関連する統計情報: Max ディメンション:PipelineName

Signature V4 メトリクス

次のメトリクスは、パイプラインの取り込みエンドポイントに適用され、ソースのプラグイン (http、otel_trace、otel_metrics) に関連付けられています。取り込みエンドポイントへのリクエストでは、すべて、Signature Version 4 を使用して署名する必要があります。これらのメトリクスは、パイプラインに接続する際の認証の問題を特定したり、認証が正常に行われていることを確認したりするときに役立ちます。

各メトリクスの先頭に、サブパイプライン名と osis_sigv4_auth が付きます。例えば、*sub_pipeline_name*.osis_sigv4_auth.httpAuthSuccess.count です。

メトリクスのサフィックス	説明
httpAuthSuccess.count	Signature V4 を使用したパイプラインへのリクエストで成功した数。 関連する統計情報: Sum ディメンション:PipelineName
httpAuthFailure.count	Signature V4 を使用したパイプラインへのリクエストで失敗した数。 関連する統計情報: Sum ディメンション:PipelineName
httpAuthServerError.count	Signature V4 を使用したパイプラインへのリクエストでサーバーエラーが返された数。 関連する統計情報: Sum ディメンション:PipelineName

制限付きブロッキングバッファのメトリクス

次のメトリクスは、制限付きブロッキングバッファに適用されます。各メトリクスの先頭に、サブパイプライン名と BlockingBuffer が付きます。例えば、*sub_pipeline_name*.BlockingBuffer.bufferUsage.value です。

メトリクスのサフィックス	説明
bufferUsage.value	バッファ内のレコードの数に基づく buffer_size の使用率です。 buffer_size は、バッファに書き込まれるレコードの最大数、およびチェックが済んでいない実行中のレコードの最大数を表します。

メトリクスのサフィックス	説明
	関連する統計: Average ディメンション:PipelineName

Otel トレースソースメトリクス

次のメトリクスは、OTel トレースソースに適用されます。各メトリクスの先頭に、サブパイプライン名と otel_trace_source が付きます。例えば、*sub_pipeline_name*.otel_trace_source.requestTimeouts.count です。

メトリクスのサフィックス	説明
requestTimeouts.count	タイムアウトしたリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
requestsReceived.count	プラグインによって受信されたリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
successRequests.count	プラグインによって正常に処理されたリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
badRequests.count	プラグインによって処理された、無効な形式のリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
requestsTooLarge.count	コンテンツ内のスパンの数がバッファ容量よりも大きいリクエストの数。

メトリクスのサフィックス	説明
	関連する統計情報: Sum ディメンション:PipelineName
internalServerError.count	カスタムの例外タイプを持つプラグインによって処理されたリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.count	リクエストをプラグインで処理している最中に記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.sum	プラグインによって処理されたリクエストの合計レイテンシー (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.max	プラグインによって処理されたリクエストの最大レイテンシー (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
payloadSize.count	受信リクエストのペイロードサイズの分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
payloadSize.sum	受信リクエストのペイロードサイズの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
payloadSize.max	受信リクエストのペイロードサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName

Otel メトリクスソースメトリクス

次のメトリクスは、OTel メトリクスソースに適用されます。各メトリクスの先頭に、サブパイプライン名と otel_metrics_source が付きます。例えば、*sub_pipeline_name*.otel_metrics_source.requestTimeouts.count です。

メトリクスのサフィックス	説明
requestTimeouts.count	タイムアウトしたプラグインへのリクエストの合計数。 関連する統計情報: Sum ディメンション:PipelineName
requestsReceived.count	プラグインによって受信されたリクエストの合計数。 関連する統計情報: Sum ディメンション:PipelineName
successRequests.count	プラグインによって正常に処理されたリクエストの数 (200 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
requestProcessDuration.count	プラグインによって処理されたリクエストのレイテンシー数 (秒)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.sum	プラグインによって処理されたリクエストの合計レイテンシー (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.max	プラグインによって処理されたリクエストの最大レイテンシー (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
payloadSize.count	受信リクエストのペイロードサイズの分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
payloadSize.sum	受信リクエストのペイロードサイズの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
payloadSize.max	受信リクエストのペイロードサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName

HTTP メトリクス

次のメトリクスは、HTTP ソースに適用されます。各メトリクスの先頭に、サブパイプライン名と http が付きます。例えば、*sub_pipeline_name*.http.requestsReceived.count です。

メトリクスのサフィックス	説明
requestsReceived.count	/log/ingest エンドポイントが受信したリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
requestsRejected.count	プラグインによって拒否されたリクエストの数 (429 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName
successRequests.count	プラグインによって正常に処理されたリクエストの数 (200 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName
badRequests.count	プラグインによって処理された、コンテンツタイプまたは形式が無効なリクエストの数 (400 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName
requestTimeouts.count	HTTP ソースサーバーでタイムアウトしたリクエストの数 (415 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
requestsTooLarge.count	コンテンツ内のイベントサイズがバッファ容量よりも大きいリクエストの数 (413 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName
internalServerError.count	カスタムの例外タイプを持つプラグインによって処理されたリクエストの数 (500 応答状態コード)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.count	プラグインによって処理されたリクエストのレイテンシー数 (秒)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.sum	プラグインによって処理されたリクエストの合計レイテンシー (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
requestProcessDuration.max	プラグインによって処理されたリクエストの最大レイテンシー (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
payloadSize.count	受信リクエストのペイロードサイズの分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
payloadSize.sum	受信リクエストのペイロードサイズの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
payloadSize.max	受信リクエストのペイロードサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName

S3 メトリクス

次のメトリクスは、S3 ソースに適用されます。各メトリクスの先頭に、サブパイプライン名と s3 が付きます。例えば、*sub_pipeline_name*.s3.s3ObjectsFailed.count です。

メトリクスのサフィックス	説明
s3ObjectsFailed.count	プラグインが読み取れなかった S3 オブジェクトの合計数。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectsNotFound.count	S3 の Not Found エラーによりプラグインが読み取れなかった S3 オブジェクトの数。これらのメトリクスも、s3ObjectsFailed メトリクスとしてカウントされます。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectsAccessDenied.count	S3 の Access Denied または Forbidden エラーによりプラグインが読み取れなかった S3 オブジェクトの数。

メトリクスのサフィックス	説明
	これらのメトリクスも、s3ObjectsFailed メトリクスとしてカウントされます。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectReadTimeElapsed.count	プラグインが S3 オブジェクトの GET リクエストを実行し、これを解析して、イベントをバッファに書き込むまでに要する時間。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectReadTimeElapsed.sum	プラグインが S3 オブジェクトの GET リクエストを実行し、これを解析して、イベントをバッファに書き込むまでに要する合計時間 (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectReadTimeElapsed.max	プラグインが S3 オブジェクトの GET リクエストを実行し、これを解析して、イベントをバッファに書き込むまでに要する最大時間 (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
s3ObjectSizeBytes.count	S3 オブジェクトサイズの分布の数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
s3ObjectSizeBytes.sum	S3 オブジェクトサイズの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectSizeBytes.max	S3 オブジェクトサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName
s3ObjectProcessedBytes.count	プラグインによって処理された S3 オブジェクトの分布の数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectProcessedBytes.sum	プラグインによって処理された S3 オブジェクトの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectProcessedBytes.max	プラグインによって処理された S3 オブジェクトの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName
s3ObjectsEvents.count	プラグインが受信した S3 イベントの分布の数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
s3ObjectsEvents.sum	プラグインが受信した S3 イベントの合計分布。 関連する統計情報: Sum ディメンション:PipelineName
s3ObjectsEvents.max	プラグインが受信した S3 イベントの最大分布。 関連する統計情報: Max ディメンション:PipelineName
sqsMessageDelay.count	S3 が、オブジェクトの作成からそれが完全に解析されるまでのイベント時間を記録している最中に、記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName
sqsMessageDelay.sum	S3 が、オブジェクトの作成からそれが完全に解析されるまでのイベント時間を記録している間の合計時間 (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
sqsMessageDelay.max	S3 が、オブジェクトの作成からそれが完全に解析されるまでのイベント時間を記録している間の最大時間 (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName

メトリクスのサフィックス	説明
s3ObjectsSucceeded.count	プラグインが正常に読み取った S3 オブジェクトの数。 関連する統計情報: Sum ディメンション:PipelineName
sqsMessagesReceived.count	プラグインがキューから受信した Amazon SQS メッセージの数。 関連する統計情報: Sum ディメンション:PipelineName
sqsMessagesDeleted.count	プラグインがキューから削除した Amazon SQS メッセージの数。 関連する統計情報: Sum ディメンション:PipelineName
sqsMessagesFailed.count	プラグインが解析に失敗した Amazon SQS メッセージの数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクス集約

次のメトリクスは、Aggregate プロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と aggregate が付きます。例えば、*sub_pipeline_name*.aggregate.actionHandleEventsOut.count です。

メトリクスのサフィックス	説明
actionHandleEvents Out.count	handleEvent 呼び出しから設定済みアクションに返されたイベントの数。

メトリクスのサフィックス	説明
	関連する統計情報: Sum ディメンション:PipelineName
actionHandleEvents Dropped.count	handleEvent 呼び出しから設定済みアクションに返されたイベントの数。 関連する統計情報: Sum ディメンション:PipelineName
actionHandleEvents ProcessingErrors.count	設定済みアクションのために handleEvent に対して実行され、エラーになった呼び出しの数。 関連する統計情報: Sum ディメンション:PipelineName
actionConcludeGrou pEventsOut.count	concludeGroup 呼び出しから設定済みアクションに返されたイベントの数。 関連する統計情報: Sum ディメンション:PipelineName
actionConcludeGrou pEventsDropped.count	condludeGroup 呼び出しから設定済みアクションに返されなかったイベントの数。 関連する統計情報: Sum ディメンション:PipelineName
actionConcludeGrou pEventsProcessingE rrors.count	設定済みアクションのために concludeGroup に対して実行され、エラーになった呼び出しの数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
currentAggregateGroups.value	現在のグループの数。この数値は、グループが完了すると減り、イベントが新しいグループの作成を始めると増えます。 関連する統計: Average ディメンション:PipelineName

日付メトリクス

次のメトリクスは、Date プロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と date が付きます。例えば、*sub_pipeline_name*.date.dateProcessingMatchSuccess.count です。

メトリクスのサフィックス	説明
dateProcessingMatchSuccess.count	match の設定オプションで指定されたパターンのうちの 1 つ以上に一致しているレコード数。 関連する統計情報: Sum ディメンション:PipelineName
dateProcessingMatchFailure.count	match の設定オプションで指定されたパターンのいずれにも一致していなかったレコード数。 関連する統計情報: Sum ディメンション:PipelineName

Grok メトリクス

次のメトリクスは、Grok プロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と grok が付きます。例えば、*sub_pipeline_name*.grok.grokProcessingMatch.count です。

メトリクスのサフィックス	説明
grokProcessingMatch.count	match の設定オプションのうち、1 つ以上と一致していることがわかったレコード数。 関連する統計情報: Sum ディメンション:PipelineName
grokProcessingMismatch.count	match の設定オプションで指定されたパターンのいずれにも一致していなかったレコード数。 関連する統計情報: Sum ディメンション:PipelineName
grokProcessingErrors.count	レコード処理エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
grokProcessingTimeouts.count	マッチング中にタイムアウトしたレコードの数。 関連する統計情報: Sum ディメンション:PipelineName
grokProcessingTime.count	個々のレコードが match 設定オプションのパターンと照合している間に記録されたデータポイントの数。 関連する統計情報: Sum ディメンション:PipelineName
grokProcessingTime.sum	個々のレコードが match 設定オプションのパターンとの照合に要する合計時間 (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
grokProcessingTime.max	個々のレコードが match 設定オプションのパターンとの照合に要する最大時間 (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName

Otel トレース raw メトリクス

次のメトリクスは、OTel トレース raw プロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と otel_trace_raw が付きます。例えば、*sub_pipeline_name*.otel_trace_raw.traceGroupCacheCount.value です。

メトリクスのサフィックス	説明
traceGroupCacheCount.value	トレースグループキャッシュ内のトレースグループの数。 関連する統計情報: Sum ディメンション:PipelineName
spanSetCount.value	スパンセットコレクション内のスパンセットの数。 関連する統計情報: Sum ディメンション:PipelineName

Otel トレースグループメトリクス

次のメトリクスは、OTel トレースグループプロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と otel_trace_group が付きます。例えば、*sub_pipeline_name*.otel_trace_group.recordsInMissingTraceGroup.count です。

メトリクスのサフィックス	説明
recordsInMissingTraceGroup.count	トレースグループフィールドが欠落しているイングレスレコードの数。 関連する統計情報: Sum ディメンション:PipelineName
recordsOutFixedTraceGroup.count	トレースグループフィールドが正常に入力されているエグレスレコードの数。 関連する統計情報: Sum ディメンション:PipelineName
recordsOutMissingTraceGroup.count	トレースグループフィールドが欠落しているエグレスレコードの数。 関連する統計情報: Sum ディメンション:PipelineName

サービスマップステートフルメトリクス

次のメトリクスは、サービスマップステートフルプロセッサに適用されます。各メトリクスの先頭に、サブパイプライン名と service-map-stateful が付きます。例えば、*sub_pipeline_name*.service-map-stateful.spansDbSize.count です。

メトリクスのサフィックス	説明
spansDbSize.value	現在と前のウィンドウ期間における MapDB のスパンのメモリ内バイトサイズ。 関連する統計: Average ディメンション:PipelineName

メトリクスのサフィックス	説明
traceGroupDbSize.value	現在と前のウィンドウ期間における MapDB のトレースグループのメモリ内バイトサイズ。 関連する統計: Average ディメンション:PipelineName
spansDbCount.value	現在と前のウィンドウ期間における MapDB のスパンの数。 関連する統計情報: Sum ディメンション:PipelineName
traceGroupDbCount.value	現在と前のウィンドウ期間における MapDB のトレースグループの数。 関連する統計情報: Sum ディメンション:PipelineName
relationshipCount.value	現在と前のウィンドウ期間に保存された関係の数。 関連する統計情報: Sum ディメンション:PipelineName

OpenSearch メトリクス

次のメトリクスは、OpenSearch シンクに適用されます。各メトリクスの先頭に、サブパイプライン名と opensearch が付きます。例えば、*sub_pipeline_name*.opensearch.bulkRequestErrors.count です。

メトリクスのサフィックス	説明
bulkRequestErrors.count	一括リクエストの送信中に発生したエラーの合計数。 関連する統計情報: Sum

メトリクスのサフィックス	説明
	ディメンション:PipelineName
documentsSuccess.count	一括リクエストにより OpenSearch Service に正常に送信されたドキュメントの数 (再試行を含む)。 関連する統計情報: Sum ディメンション:PipelineName
documentsSuccessFirstAttempt.count	一括リクエストにより OpenSearch Service に一回で正常に送信されたドキュメントの数。 関連する統計情報: Sum ディメンション:PipelineName
documentErrors.count	一括リクエストで送信できなかったドキュメントの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestFailed.count	送信できなかった一括リクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestNumberOfRetries.count	失敗した一括リクエストの再試行数。 関連する統計情報: Sum ディメンション:PipelineName
bulkBadRequestErrors.count	一括リクエストの送信中に発生した Bad Request エラーの数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
bulkRequestNotAllowedErrors.count	一括リクエストの送信中に発生した Request Not Allowed エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestInvalidInputErrors.count	一括リクエストの送信中に発生した Invalid Input エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestNotFoundErrors.count	一括リクエストの送信中に発生した Request Not Found エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestTimeoutErrors.count	一括リクエストの送信中に発生した Request Timeout エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestServerErrors.count	一括リクエストの送信中に発生した Server Error エラーの数。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestSizeBytes.count	一括リクエストのペイロードサイズの分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
bulkRequestSizeBytes.sum	一括リクエストのペイロードサイズの合計分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestSizeBytes.max	一括リクエストのペイロードサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName
bulkRequestLatency.count	リクエストがプラグインに送信されている間に記録されたデータポイントの数 (再試行含む)。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestLatency.sum	プラグインに送信されたリクエストの合計レイテンシー (再試行含む) (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName
bulkRequestLatency.max	プラグインに送信されたリクエストの最大レイテンシー (再試行含む) (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
s3.dlqS3RecordsSuccess.count	S3 デッドレターキューに正常に送信されたレコードの数。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
s3.dlqS3RecordsFailed.count	S3 デッドレターキューへの送信に失敗したレコードの数。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestSuccess.count	S3 デッドレターキューに正常に送信されたリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestFailed.count	S3 デッドレターキューへの送信に失敗したリクエストの数。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestLatency.count	リクエストが S3 デッドレターキューに送信されている間に記録されたデータポイントの数 (再試行含む)。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestLatency.sum	S3 デッドレターキューに送信されたリクエストの合計レイテンシー (再試行含む) (ミリ秒)。 関連する統計情報: Sum ディメンション:PipelineName

メトリクスのサフィックス	説明
s3.dlqS3RequestLatency.max	S3 デッドレターキューに送信されたリクエストの最大レイテンシー (再試行含む) (ミリ秒)。 関連する統計情報: Max ディメンション:PipelineName
s3.dlqS3RequestSizeBytes.count	S3 デッドレターキューへのリクエストの、ペイロードサイズの分布数 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestSizeBytes.sum	S3 デッドレターキューへのリクエストの、ペイロードサイズの合計分布 (バイト)。 関連する統計情報: Sum ディメンション:PipelineName
s3.dlqS3RequestSizeBytes.max	S3 デッドレターキューへのリクエストの、ペイロードサイズの最大分布 (バイト)。 関連する統計情報: Max ディメンション:PipelineName

システムと測定メトリクス

次のメトリクスは、OpenSearch Ingestion システム全体に適用されます。これらのメトリクスの前には、何も付いていません。

メトリクス	説明
system.cpu.usage.value	すべてのデータノードの、使用可能な CPU 使用量の割合。

メトリクス	説明
	関連する統計: Average ディメンション: PipelineName 、area、id
system.cpu.count.value	すべてのデータノードの CPU 使用量の合計。 関連する統計: Average ディメンション: PipelineName 、area、id
jvm.memory.max.value	メモリ管理に使用できるメモリの最大数 (バイト)。 関連する統計: Average ディメンション: PipelineName 、area、id
jvm.memory.used.value	使用されているメモリの合計量 (バイト)。 関連する統計: Average ディメンション: PipelineName 、area、id signa
jvm.memory.committed.value	Java 仮想マシン (JVM) で使用するためにコミットされたメモリーの量 (バイト)。 関連する統計: Average ディメンション: PipelineName 、area、id
computeUnits	パイプラインで使用されている Ingestion OpenSearch Compute Units (Ingestion OCU) の数。 関連する統計: Max、Sum、Average ディメンション:PipelineName

Amazon OpenSearch Ingestion のベストプラクティス

この章では、Amazon OpenSearch Ingestion パイプラインを作成して管理するときのベストプラクティスと、多くのユースケースに適用される一般的なガイドラインを提供します。各ワークロードは一意で、固有の特性があるため、すべてのユースケースに完全に適した一般的な推奨事項はありません。

トピック

一般的なベストプラクティス

パイプラインの作成と管理には、次の一般的なベストプラクティスが適用されます。

- 高可用性を確保するために、VPC パイプラインを 2 つまたは 3 つのサブネットで構成します。パイプラインを 1 つのサブネットにのみデプロイした場合、そのアベイラビリティーゾーンがダウンするとデータを取り込めなくなります。
- 各パイプライン内では、サブパイプラインの数を 5 つ以下に制限することをお勧めします。
- S3 ソースプラグインを使用している場合は、パフォーマンスを最適化するために均等なサイズの S3 ファイルを使用します。
- S3 ソースプラグインを使用している場合は、パフォーマンスを最適化するために、S3 バケットのファイルサイズ 0.25 GB ごとに 30 秒の可視性タイムアウトを追加します。
- パイプライン設定にデッドレターキュー (DLQ) を含めると、失敗したイベントをオフロードし、分析のためにアクセスできるようになります。誤ったマッピングや他の問題によりシンクがデータを拒否した場合、トラブルシューティングして問題を解決するために、データを DLQ にルーティングできます。

推奨される CloudWatch アラーム

CloudWatch アラームは、CloudWatch メトリクスがある程度の時間にわたって指定された値を超えたときにアクションを実行します。たとえば、クラスターの状態が `red` になって 1 分を超えたときに AWS から E メールを受け取ることができます。このセクションでは、Amazon OpenSearch Ingestion で推奨されるいくつかのアラームとそのアラームへの対応方法について説明します。

アラームの設定の詳細については、Amazon CloudWatch ユーザーガイドの「Amazon CloudWatch でのアラームの使用」を参照してください。

アラーム	問題
`computeUnits` maximum is = the configured `maxUnits` for 15 minute, 3 consecutive times	パイプラインが最大キャパシティに達したため、`maxUnits` の更新が必要になります。パイプラインの最大キャパシティを増やしてください。
`opensearch.documentErrors.count` sum is = *`{sub_pipeline_name}`* `.opensearch.recordsIn.count` sum for 1 minute, 1 consecutive time	パイプラインから OpenSearch シンクに書き込めません。パイプラインのアクセス許可をチェックし、ドメインまたはコレクションが正常であることを確かめます。デッドレターキュー (DLQ) が設定されている場合は、失敗したイベントがないかどうかを確認することもできます。
`bulkRequestLatency.max` max is >= x for 1 minute, 1 consecutive time	パイプラインで OpenSearch シンクにデータを送信するときに高いレイテンシーが発生しています。これは、シンクのサイズが小さすぎるか、シャーディング戦略が不十分でシンクの処理が遅れていることが原因と考えられます。高いレイテンシーが続くとパイプラインのパフォーマンスに影響が及び、クライアントのバックプレッシャーにつながる可能性があります。
`httpAuthFailure.count` sum >= 1 for 1 minute, 1 consecutive time	取り込みリクエストが認証されていません。すべてのクライアントで Signature Version 4 認証が正しく有効になっていることを確認します。
`system.cpu.usage.value` average >= 80% for 15	CPU 使用率の高い状態が続くと、問題が生じる可能性があります。パイプラインの最大キャパシティを増やすことを検討してください。

アラーム	問題
minutes, 3 consecutive times	
bufferUsage.value average >= 80% for 15 minutes, 3 consecutive times	バッファ使用量の多い状態が続くと、問題が生じる可能性があります。パイプラインの最大キャパシティを増やすことを検討してください。

検討した方が良いその他のアラーム

定期的に使用する Amazon OpenSearch Ingestion の機能に応じて、次のアラームの設定を検討します。

アラーム	問題
dynamodb.exportJobFailure.count sum 1	Amazon S3 へのエクスポートのトリガーの試行が失敗しました。
opensearch.EndtoEndLatency.avg average > X(15 分間)、連続 4 回	EndtoEndLatency が DynamoDB ストリームからの読み取りに必要な値よりも大きくなっています。これは、OpenSearch クラスターのスケールが不十分であるか、またはパイプライン OCU の最大キャパシティが DynamoDB テーブルの WCU スループットに鑑みて少なすぎることが原因である可能性があります。EndtoEndLatency は、エクスポート後には大きくなりますが、最新の DynamoDB ストリームに追いつくにつれて時間の経過とともに小さくなるはずです。
dyanmodb.changeEventsProcessed.count sum == 0 (X 分間)	DynamoDB ストリームからレコードは収集されていません。これは、テーブル上でアクティビティがないこと、または DynamoDB ストリームに対するアクセスに関する問題が原因である可能性があります。

アラーム	問題
opensearch.s3.dlqS3RecordsSuccess.count sum >= opensearch.documentSuccess.count sum for 1 minute, 1 consecutive time	OpenSearch シンクよりも多くのレコードが DLQ に送信されています。OpenSearch シンクプラグインのメトリクスを確認し、根本原因を調査して特定してください。
grok.grokProcessingTimeouts.count sum = recordsIn.count sum for 1 minute, 5 consecutive times	Grok プロセッサがパターンマッチングを試みている間、すべてのデータがタイムアウトしています。これはパフォーマンスに影響を与え、パイプラインの速度を低下させている可能性があります。タイムアウトを減らすために、パターンの調整を検討してください。
grok.grokProcessingErrors.count sum is >= 1 for 1 minute, 1 consecutive time	Grok プロセッサがパイプライン内のデータとのパターンマッチングに失敗したため、エラーが発生しています。データと Grok プラグインの設定を確認し、パターンマッチングが想定どおりであることを確認してください。
grok.grokProcessingMismatch.count sum = recordsIn.count sum for 1 minute, 5 consecutive times	Grok プロセッサは、パイプライン内のデータにパターンを一致させることができません。データと Grok プラグインの設定を確認し、パターンマッチングが想定どおりであることを確認してください。

アラーム	問題
date.dateProcessingMatchFailure.count sum = recordsIn.count sum for 1 minut, 5 consecutive times	Date プロセッサは、パイプライン内のデータにどのパターンも一致させることができません。データと Date プラグインの設定を確認し、パターンが想定どおりであることを確認してください。
s3.s3ObjectsFailed.count sum >= 1 for 1 minute, 1 consecutive time	この問題は、S3 オブジェクトが存在しないか、パイプラインのアクセス許可が不十分であるために発生しています。s3ObjectsNotFound.count メトリクスと s3ObjectsAccessDenied.count メトリクスを確認し、根本原因を特定してください。S3 オブジェクトが存在することを確認するか、アクセス許可を更新します。
s3.sqsMessagesFailed.count sum >= 1 for 1 minute, 1 consecutive time	S3 プラグインは Amazon SQS メッセージの処理に失敗しました。SQS キューで DLQ が有効になっている場合、失敗のメッセージを確認してください。パイプラインが処理しようとしている無効なデータをキューが受け取っている可能性があります。
http.badRequests.count sum >= 1 for 1 minute, 1 consecutive times	クライアントが不正なリクエストを送信しています。すべてのクライアントが適切なペイロードを送信していることを確認してください。
http.requestsTooLarge.count sum >= 1 for 1 minute, 1 consecutive time	HTTP ソースプラグインからのリクエストに含まれるデータが多すぎるため、バッファ容量を超えています。クライアントのバッチサイズを調整してください。

アラーム	問題
`http.internalServerError.count` sum >= 0 for 1 minute, 1 consecutive time	HTTP ソースプラグインに問題が発生し、イベントを受信できません。
`http.requestTimeouts.count` sum >= 0 for 1 minute, 1 consecutive time	ソースタイムアウトは、パイプラインのプロビジョニングが不十分であることが原因と考えられます。追加のワークロードを処理するために、パイプラインの `maxUnits` を増やすことを検討してください。
`otel_trace.badRequests.count` sum >= 1 for 1 minute, 1 consecutive time	クライアントが不正なリクエストを送信しています。すべてのクライアントが適切なペイロードを送信していることを確認してください。
`otel_trace.requestsTooLarge.count` sum >= 1 for 1 minute, 1 consecutive time	Otel Trace ソースプラグインからのリクエストに含まれるデータが多すぎるため、バッファ容量を超えています。クライアントのバッチサイズを調整してください。
`otel_trace.internalServerError.count` sum >= 0 for 1 minute, 1 consecutive time	Otel Trace ソースプラグインに問題が発生し、イベントを受信できません。

アラーム	問題
otel_trace.requestTimeouts.count sum >= 0 for 1 minute, 1 consecutive time	ソースタイムアウトは、パイプラインのプロビジョニングが不十分であることが原因と考えられます。追加のワークロードを処理するために、パイプラインの maxUnits を増やすことを検討してください。
otel_metrics.requestTimeouts.count sum >= 0 for 1 minute, 1 consecutive time	ソースタイムアウトは、パイプラインのプロビジョニングが不十分であることが原因と考えられます。追加のワークロードを処理するために、パイプラインの maxUnits を増やすことを検討してください。

Amazon OpenSearch Serverless

Amazon OpenSearch Serverless は、Amazon OpenSearch Service のオンデマンドの自動スケーリング設定です。 OpenSearch サーバーレスコレクションは、アプリケーションのニーズに基づいてコンピューティング性能をスケーリングする OpenSearch クラスターです。これは、手動で容量を管理する OpenSearch サービスプロビジョニング OpenSearch ドメイン とは対照的です。

OpenSearch Serverless は、低頻度、断続的、または予測不可能なワークロードに対して、シンプルで費用対効果の高いオプションを提供します。このコスト効率が高いのは、アプリケーションの使用状況に合わせてコンピューティング性能を自動的にスケーリングするためです。

OpenSearch サーバーレスコレクションには、プロビジョニングされた OpenSearch サービスドメインで使用されるのと同じ種類の大容量、分散型、高可用性のストレージボリュームがあります。

OpenSearch サーバーレスコレクションは常に暗号化されます。暗号化キーは選択できますが、暗号化を無効にすることはできません。詳細については、「the section called “暗号化”」を参照してください。

トピック

利点

OpenSearch サーバーレスには次の利点があります。

- プロビジョニングよりもシンプル – OpenSearch サーバーレスは、 OpenSearch クラスターと容量の管理の複雑さの大部分を排除します。クラスターのサイズ設定および調整を自動的に行い、シャードとインデックスのライフサイクルを管理します。また、サービスソフトウェアの更新とOpenSearchバージョンアップグレードも管理します。すべての更新とアップグレードは中断されません。
- 費用対効果 – OpenSearch Serverless を使用する場合、消費したリソースに対してのみ料金が発生します。これにより、ピーク時のワークロードに対する事前のプロビジョニングやオーバープロビジョニングが不要になります。
- 高可用性 — OpenSearch サーバーレスは、アベイラビリティーゾーンの停止やインフラストラクチャの障害から保護するために、冗長性を備えた本稼働ワークロードをサポートします。
- スケーラブル — OpenSearch サーバーレスは、リソースを自動的にスケーリングして、一貫した高速なデータ取り込み速度とクエリ応答時間を維持します。

Amazon OpenSearch Serverless とは

Amazon OpenSearch Serverless は、Amazon OpenSearch Service のオンデマンドサーバーレス設定です。Serverless は、クラスターのプロビジョニング、設定、チューニングの運用上の複雑さを排除します OpenSearch。これは、 OpenSearch クラスターを自己管理したくない組織や、大規模なクラスターを運用するための専用のリソースや専門知識を持たない組織に適しています。OpenSearch Serverless を使用すると、基盤となるインフラストラクチャやデータ管理を気にすることなく、大量のデータを簡単に検索および分析できます。

OpenSearch サーバーレスコレクションは、特定のワークロードまたはユースケースをサポートするために連携するインデックスの OpenSearchグループです。コレクションは、手動プロビジョニングを必要とするセルフマネージド OpenSearch 型クラスターよりも使いやすいです。

コレクションには、プロビジョニングされた OpenSearch サービスドメインで使用されるのと同じ種類の大容量、分散型、可用性の高いストレージボリュームがありますが、手動設定やチューニングを必要としないため、複雑さが軽減されます。データはコレクション内で転送中に暗号化されます。 OpenSearch Serverless は、データを分析するための直感的なインターフェイスを提供する OpenSearch Dashboards もサポートしています。

サーバーレスコレクションは現在、 OpenSearch バージョン 2.0.x を実行しています。新しいバージョンがリリースされると、 OpenSearch Serverless はコレクションを自動的にアップグレードして、新機能、バグ修正、パフォーマンスの向上を利用します。

トピック

OpenSearch Serverless のユースケース

OpenSearch Serverless は、主に次の 2 つのユースケースをサポートしています。

- ログ分析 – ログ分析セグメントは、オペレーションインサイトとユーザーの行動に関するインサイトを得るために、マシン生成による大量の半構造化された時系列データを分析することに重点を置きます。
- 全文検索 – 全文検索セグメントが、社内ネットワーク内のアプリケーション (コンテンツ管理システム、法的文書) や、e コマースウェブサイトのコンテンツ検索などのインターネット向けアプリケーションを強化します。

コレクションを作成するときは、これらのユースケースのいずれかを選択します。詳細については、「the section called “コレクションタイプを選択する”」を参照してください。

開始

OpenSearch Serverless の使用を開始するには、 OpenSearch サービスコンソール、、または AWS SDKs のいずれかを使用して AWS CLI 1 つ以上のコレクションを作成します。コレクションをすばやく起動して実行するのに役立つチュートリアルについては、「the section called “Serverless OpenSearch の開始方法”」を参照してください。

OpenSearch Serverless は OpenSearch 、オープンソーススイートと同じ取り込みおよびクエリ API オペレーションをサポートしているため、既存のクライアントとアプリケーションを引き続き使用できます。 OpenSearch Serverless を使用するには、クライアントが OpenSearch 2.x と互換性がある必要があります。詳細については、「the section called “コレクションへのデータの取り込み”」を参照してください。

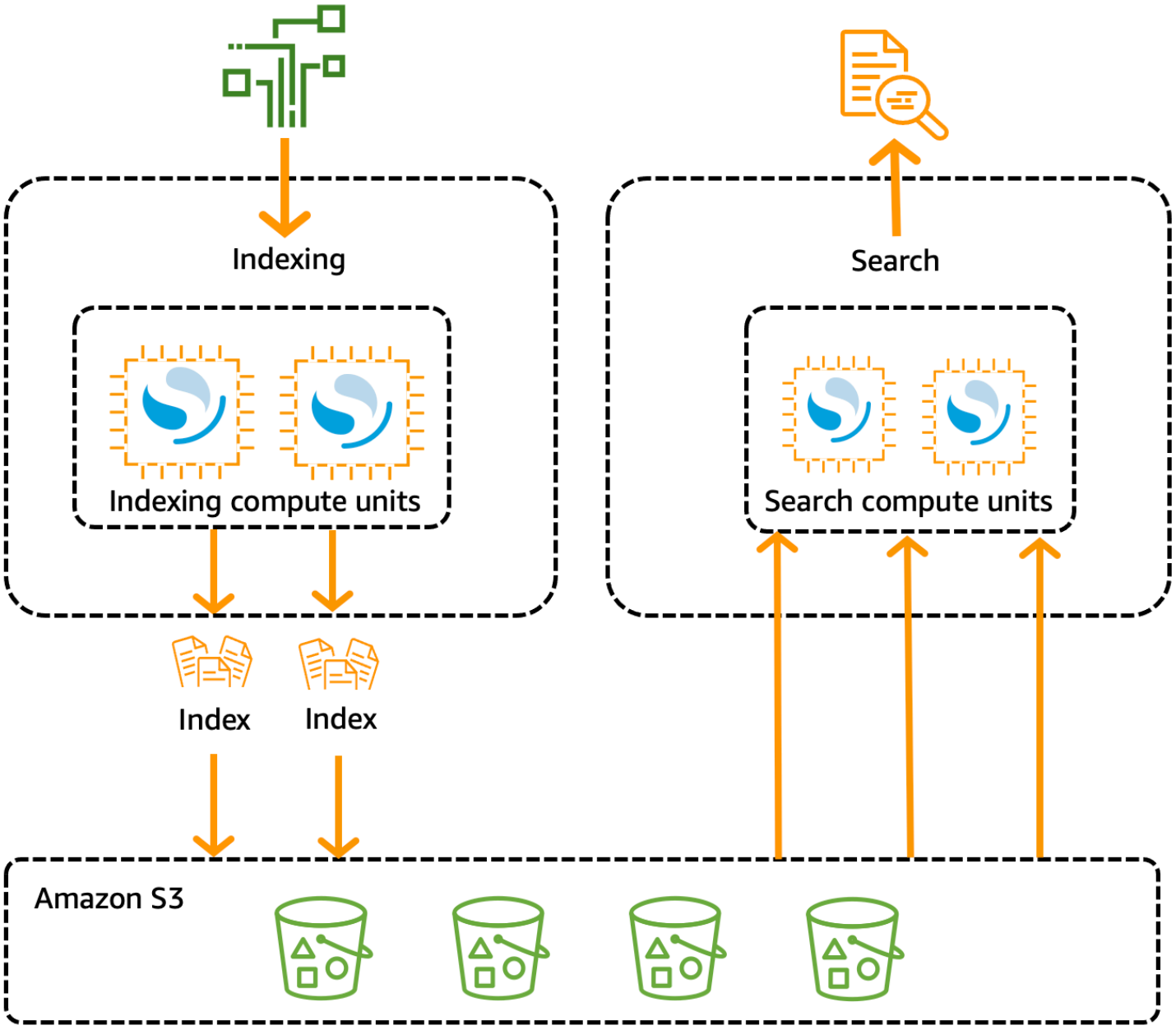
仕組み

従来の OpenSearch クラスターには、インデックス作成オペレーションと検索オペレーションの両方を実行するインスタンスのセットが 1 つあり、インデックスストレージはコンピューティング性能と密接に結合されています。対照的に、 OpenSearch Serverless は、インデックス作成 (取り込み) コンポーネントと検索 (クエリ) コンポーネントを分離するクラウドネイティブアーキテクチャを使用します。Amazon S3 はインデックスのプライマリデータストレージです。

この分離されたアーキテクチャでは、検索機能とインデックス作成機能を相互に独立して、さらに S3 のインデックス化されたデータとも無関係にスケーリングできます。また、このアーキテクチャでは、取り込みオペレーションとクエリオペレーションを分離できるため、リソースを競合させることなく同時に実行できます。

コレクションにデータを書き込むと、 OpenSearch Serverless はそれをインデックス作成コンピューティングユニットに分散します。インデックス作成用コンピューティングユニットは、受信データを取り込み、インデックスを S3 に移動します。コレクションデータに対して検索を実行すると、 OpenSearch Serverless はクエリ対象のデータを保持する検索コンピューティングユニットにリクエストをルーティングします。検索コンピューティングユニットは、インデックス化されたデータを S3 から直接ダウンロードし (まだローカルにキャッシュされていない場合)、検索オペレーションを実行し、集計を実行します。

次の図は、この分離されたアーキテクチャを示しています。



OpenSearch データインジェスト、検索、クエリのサーバーレスコンピューティング容量は、コンピューティングユニット (OCU) OpenSearch で測定されます。 OCUs 各 OCU は、6 GiB のメモリと対応する仮想 CPU (vCPU)、および Amazon S3 へのデータ転送を組み合わせたものです。各 OCU には、120 GiB のインデックスデータを保存するのに十分なホットエフェメラルストレージが含まれています。

最初のコレクションを作成すると、 OpenSearch Serverless は 2 つの OCUs をインスタンス化します。1 つはインデックス作成用、もう 1 つは検索用です。高可用性を確保するために、他のアベイラビリティーゾーンでスタンバイノードのセットも起動します。開発およびテストの目的で、コレクションの冗長性を有効にする設定を無効にすることができます。これにより、2 つのスタンバイレプ

リカがなくなり、2 つの OCUs。デフォルトでは、冗長アクティブレプリカが有効になっています。これは、アカウントの最初のコレクションのために合計で 4 つの OCU がインスタンス化されることを意味します。

これらの OCU は、すべてのコレクションエンドポイントにアクティビティがない場合でも存在します。後続のすべてのコレクションは、これらの OCU を共有します。同じアカウントで追加のコレクションを作成すると、 OpenSearch Serverless は、指定した容量制限に従って、コレクションをサポートするために必要な場合にのみ検索および取り込み用の OCUs を追加します。コンピューティング使用量が減少すると、容量はスケールダウンします。

これらの OCU に対する課金方法については、「the section called “ OpenSearch Serverless の料金”」を参照してください。

コレクションタイプを選択する

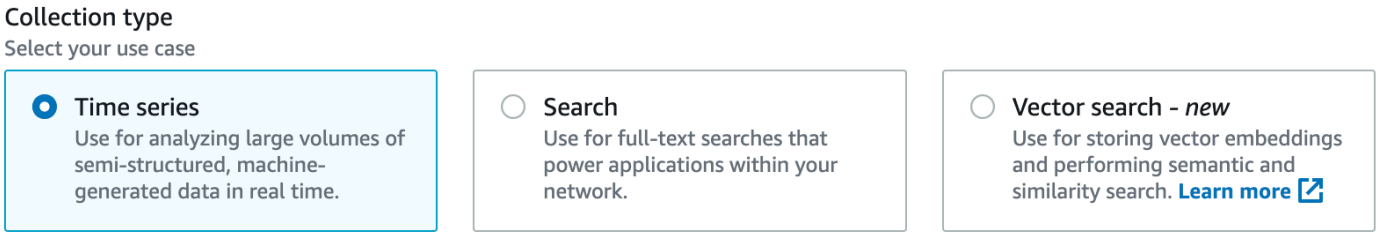
OpenSearch サーバーレスは、次の 3 つの主要なコレクションタイプをサポートしています。

[Time series] (時系列) – 運用、セキュリティ、ユーザー行動、およびビジネスに関するインサイトを得るために、マシン生成による大量の半構造化されたデータをリアルタイムで分析することに重点を置いたログ分析セグメント。

[Search] (検索) – 社内ネットワーク内のアプリケーション (コンテンツ管理システム、法的文書) や、e コマースウェブサイト検索やコンテンツ検索などのインターネット向けアプリケーションを強化する全文検索。

[ベクトル検索] – ベクトルデータ管理を簡素化し、機械学習 (ML) によって拡張された検索エクスペリエンスと、チャットボット、パーソナルアシスタント、不正検出などの生成 AI アプリケーションを強化する、ベクトル埋め込みのセマンティック検索。

コレクションを初めて作成するときに、コレクションタイプを選択します。



選択するコレクションタイプは、コレクションに取り込む予定のデータの種類と、そのデータへのクエリ実行方法によって異なります。コレクションの作成後にコレクションタイプを変更することはできません。

コレクションタイプには、次の大きな違いがあります。

- 検索およびベクトル検索コレクションでは、迅速なクエリ応答時間を確保するために、すべてのデータがホットストレージに保存されます。時系列コレクションでは、ホットストレージとウォームストレージを組み合わせて使用します。この場合、より頻繁にアクセスされるデータのクエリ応答時間を最適化するために、最新のデータがホットストレージに保存されます。
- 時系列およびベクトル検索コレクションの場合、カスタムドキュメント ID によるインデックス付けや、UPSERT リクエストによる更新はできません。この操作は検索のユースケース専用です。代わりにドキュメント ID を使用して更新できます。詳細については、「the section called “サポートされている OpenSearch API オペレーションとアクセス許可”」を参照してください。
- 検索および時系列コレクションの場合、k-NN タイプのインデックスは使用できません。

OpenSearch Serverless の料金

OpenSearch Serverless では、以下のコンポーネントに対して課金されます。

- データインジェスト用コンピューティング
- 検索およびクエリ用コンピューティング
- ストレージは Amazon S3 に保持されます

OCU は 1 時間ごとに、秒単位の粒度で課金されます。アカウントのステートメントには、データインジェスト用のラベルと検索用のラベルが付いた OCU 時間単位のコンピューティングに関するエントリがあります。また、Amazon S3 に保存されているデータに対しても月単位で請求されます。OpenSearch Dashboards の使用に対して課金されることはありません。

コレクションを作成し、冗長アクティブレプリカを有効にすると、取り込みに 2 OCUs [0.5 OCU x 2]、検索に 1 OCU [0.5 OCU x 2] 以上の料金が請求されます。冗長アクティブレプリカを無効にすると、アカウント内の最初のコレクションに対して最低 1 OCU [0.5 OCU x 2] の料金が請求されます。後続のすべてのコレクションは、これらの OCU を共有できます。

OpenSearch Serverless はOCUsコレクションをサポートするために必要なコンピューティング能力とストレージに基づいて、1 OCU 単位で OCU を追加します。コストを抑えるために、アカウントの OCU の最大数を設定できます。

> **Note**
>
> 一意の を持つコレクション AWS KMS keys は、OCUs他のコレクションと共有できません。

OpenSearch サーバーレスは、ワークロードの変化を考慮して、最小限必要なリソースの使用を試みます。いつでもプロビジョニングされる OCUs の数はさまざまであり、正確ではありません。時間の経過とともに、 OpenSearch サーバーレスが使用するアルゴリズムは、システムの使用量を最小限に抑えるために改善され続けます。

料金の詳細については、「Amazon OpenSearch Service の料金」を参照してください。

サポート対象 AWS リージョン

OpenSearch Serverless は、その OpenSearch サービス AWS リージョン が利用可能な のサブセットで使用できます。サポートされているリージョンのリストについては、「」の「Amazon OpenSearch Service エンドポイントとクォータ」を参照してくださいAWS 全般のリファレンス。

制限事項

OpenSearch Serverless には以下の制限があります。

- 一部の OpenSearch API オペレーションはサポートされていません。the section called “サポートされている OpenSearch API オペレーションとアクセス許可” を参照してください。
- 一部の OpenSearch プラグインはサポートされていません。the section called “サポートされている OpenSearch プラグイン” を参照してください。
- 現在、マネージド OpenSearch サービスドメインからサーバーレスコレクションにデータを自動的に移行する方法はありません。ドメインからコレクションにデータを再インデックスする必要があります。
- コレクションへのクロスアカウントアクセスはサポートされていません。他のアカウントからのコレクションを暗号化またはデータアクセスポリシーに含めることはできません。
- カスタム OpenSearch プラグインはサポートされていません。
- OpenSearch Serverless コレクションのスナップショットを作成または復元することはできません。
- クロスリージョン検索およびレプリケーションはサポートされていません。

- 1 つのアカウントおよびリージョンで使用できるサーバーレスリソースの数には制限があります。OpenSearch 「サーバーレスクォータ」を参照してください。
- ベクトル検索コレクション内のインデックスの更新間隔は約 60 秒です。検索コレクションと時系列コレクションのインデックスの更新間隔は約 10 秒です。
- シャードの数、間隔の数、および更新間隔は変更できず、 OpenSearch サーバーレスによって処理されます。シャーディング戦略はコレクションタイプとトラフィックに基づいています。例えば、時系列コレクションでは、書き込みトラフィックのボトルネックに基づいてプライマリシャードをスケーリングします。
- 2.1 までの OpenSearch バージョンで利用可能な地理空間機能がサポートされています。

OpenSearch サービスと OpenSearch サーバーレスの比較

OpenSearch Serverless では、一部の概念と機能は、プロビジョニングされた OpenSearch サービスドメインに対応する機能とは異なります。例えば、重要な違いの 1 つは、 OpenSearch サーバーレスにはクラスターやノードの概念がないことです。

次の表は、 OpenSearch Serverless の重要な機能と概念が、プロビジョニングされた OpenSearch サービスドメインの同等の機能とどの程度異なるかを示しています。

機能	OpenSearch サービス	OpenSearch サーバーレス
ドメインとコレクション	インデックスは、事前プロビジョニングされた OpenSearch クラスターであるドメイン に保持されます。 詳細については、「ドメインの作成と管理」を参照してください。	インデックスはコレクションで保持されます。コレクションとは、特定のワークロードやユースケースを表すインデックスを論理的にグループ化したものです。 詳細については、「the section called “コレクションの作成、一覧表示、および削除”」を参照してください。
ノードタイプとキャパシティー管理	コストとパフォーマンスの仕様を満たすノードタイプを使用してクラスターを構築します。独自のストレージ要件を計算し、ドメインのインスタンスタイプを選択する必要があります。	OpenSearch Serverless は、容量の使用量に基づいて、アカウントの追加コンピューティングユニットを自動的にスケーリングおよびプロビジョニングします。 詳細については、「the section called “キャパシティ制限の管理”」を参照してください。

機能	OpenSearch サービス	OpenSearch サーバーレス
	詳細については、「the section called “ドメインのサイジング”」を参照してください。	
「請求」	1 時間ごとの EC2 インスタンスの使用およびインスタンスにアタッチされたすべての EBS ストレージボリュームの累積サイズに対して料金が発生します。 詳細については、「the section called “料金”」を参照してください。	データインジェストのコンピューティング、検索とクエリのコンピューティング、および S3 に保持されるストレージに対して、OCU 時間単位で課金されます。 詳細については、「the section called “OpenSearch Serverless の料金”」を参照してください。
暗号化	ドメインの保管時の暗号化はオプションです。 詳細については、「the section called “保管中の暗号化”」を参照してください。	コレクションには保管時の暗号化が必要です。 詳細については、「the section called “暗号化”」を参照してください。
データアクセスコントロール	ドメイン内のデータへのアクセスは、IAM ポリシーときめ細かなアクセスコントロールによって決定されます。	コレクション内のデータへのアクセスは、データアクセスポリシーによって決定されます。
サポートされている OpenSearch オペレーション	OpenSearch サービスは、すべての OpenSearch API オペレーションのサブセットをサポートします。 詳細については、「the section called “サポートされているオペレーション”」を参照してください。	OpenSearch Serverless は、 OpenSearch API オペレーションの異なるサブセットをサポートしています。 詳細については、「the section called “サポートされているオペレーションとプラグイン”」を参照してください。

機能	OpenSearch サービス	OpenSearch サーバーレス
Dashboards サインイン	ユーザー名とパスワードを使用してサインインします。 詳細については、「the section called “マスターユーザーとしてダッシュボードにアクセスする OpenSearch 。”」を参照してください。	AWS コンソールにログインして Dashboard URL に移動すると、自動的にログインします。 詳細については、「the section called “OpenSearch ダッシュボードへのアクセス”」を参照してください。
API	OpenSearch サービス OpenSearch API オペレーション を使用して、プログラムでサービスとやり取りします。	OpenSearch Serverless OpenSearch API オペレーション を使用して、プログラムで Serverless とやり取りします。
ネットワークアクセス	ドメインのネットワーク設定は、ドメインエンドポイントと OpenSearch Dashboards エンドポイントに適用されます。両方のネットワークアクセスは密接に連携しています。	ドメインエンドポイントと OpenSearch Dashboards エンドポイントのネットワーク設定は分離されます。 OpenSearch Dashboards のネットワークアクセスを設定しないことを選択できます。 詳細については、「the section called “ネットワークアクセス”」を参照してください。
リクエストへの署名	リクエストに署名するには、OpenSearch 高レベルと低レベルの REST クライアントを使用します。サービス名を `es` と指定してください。	現時点では、 OpenSearch Serverless は OpenSearch Service がサポートするクライアントのサブセットをサポートしています。 リクエストに署名するときは、サービス名を `aoss` と指定します。`x-amz-content-sha256` ヘッダーは必須です。詳細については、「the section called “その他のクライアント”」を参照してください。

機能	OpenSearch サービス	OpenSearch サーバーレス
OpenSearch バージョンアップグレード	新しいバージョンの OpenSearch が利用可能になったら、ドメインを手動でアップグレードします。ユーザーには、ドメインがアップグレード要件を満たしていること、および重要な変更点すべてに対処したことを確認する責任があります。	OpenSearch サーバーレスは、コレクションを新しい OpenSearch バージョンに自動的にアップグレードします。新しいバージョンがリリースされるとすぐにアップグレードが行われるとは限りません。
サービスソフトウェア更新	サービスソフトウェア更新が利用可能になったら、手動でドメインに適用してください。	OpenSearch Serverless はコレクションを自動的に更新して、最新のバグ修正、機能、パフォーマンスの向上を利用します。
VPC アクセス	VPC 内でドメインをプロビジョニングできます。 また、追加のOpenSearch サービスマネージド VPC エンドポイントを作成して、ドメインにアクセスすることもできます。	アカウントのサーバーOpenSearch レスマネージド VPC エンドポイントを 1 つ以上作成します。次に、これらのエンドポイントをネットワークポリシー内に含めます。
SAML 認証	SAML 認証はドメインごとに有効にします。 詳細については、「the section called “ OpenSearch Dashboards の SAML 認証”」を参照してください。	アカウントレベルで 1 つまたは複数の SAML プロバイダーを設定し、関連するユーザー ID とグループ ID をデータアクセスポリシーに含めます。 詳細については、「the section called “SAML 認証”」を参照してください。
Transport Layer Security (TLS)	OpenSearch サービスは TLS 1.2 をサポートしていますが、TLS 1.3 を使用することをお勧めします。	OpenSearch Serverless は TLS 1.2 をサポートしていますが、TLS 1.3 を使用することをお勧めします。

Amazon OpenSearch Serverless の開始方法

このチュートリアルでは、Amazon OpenSearch Serverless 検索コレクションをすばやく起動して実行するための基本的な手順について説明します。検索コレクションを使用すると、内部ネットワーク内のアプリケーション、および e コマースウェブサイトの検索やコンテンツ検索などのインターネットに接続されたアプリケーションを強化できます。

ベクトル検索コレクションの使用方法については、「the section called “ベクトル検索コレクションの使用”」を参照してください。コレクションの使用に関する詳細については、このガイドの「the section called “コレクションの作成、一覧表示、および削除”」とその他のトピックを参照してください。

このチュートリアルでは、次の手順を実行します。

1. アクセス許可を設定する
2. コレクションを作成する
3. データをアップロードおよび検索する
4. コレクションを削除する

ステップ 1: アクセス許可を設定する

このチュートリアルを完了し、一般的に OpenSearch Serverless を使用するには、正しい IAM アクセス許可が必要です。このチュートリアルでは、コレクションを作成し、アップロードしたデータの検索を行い、最後にそのコレクションを削除します。

ユーザーまたはロールには、以下の最低限の許可を含む ID ベースのポリシーが、アタッチされている必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
```

```
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

OpenSearch Serverless IAM アクセス許可の詳細については、「」を参照してくださいthe section called “ID とアクセス管理”。

ステップ 2: コレクションを作成する

コレクションは、特定のワークロードまたはユースケースをサポートするために連携する OpenSearch インデックスのグループです。

OpenSearch サーバーレスコレクションを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインで [Collections] (コレクション) 、[Create collection] (コレクションを作成) を選択します。
3. コレクションに movies という名前を付けます。
4. コレクションタイプでは、[Search] (検索) を選択します。詳細については、「Choosing a collection type」(コレクションタイプの選択) を参照してください。
5. セキュリティ で、標準作成 を選択します。
6. 暗号化 で、 の使用 AWS 所有のキーを選択します。これは、 OpenSearch サーバーレス AWS KMS key がデータの暗号化に使用する です。
7. [Network] (ネットワーク) で、コレクションのネットワーク設定を行います。

    - アクセスタイプには、[Public] (パブリック) を選択します。
    - リソースタイプ で、 OpenSearch エンドポイントへのアクセスを有効にする と OpenSearch Dashboards へのアクセスを有効にする の両方を選択します。 OpenSearch Dashboards を使用してデータをアップロードおよび検索するため、両方を有効にする必要があります。

8. [次へ] をクリックします。
9. [Configure data access] (データアクセスの設定) で、コレクションのアクセス設定をセットアップします。データアクセスポリシーにより、コレクション内のデータに対し、ユーザーとロールがアクセスできるようになります。このチュートリアルでは、movies コレクション内にあるデータのインデックス作成と検索に必要なアクセス許可を、1 人のユーザーに提供します。

   movies コレクションに対するアクセスを提供する単一のルールを作成します。このルールに、「Movies collection access」権という名前を付けます。
10. 「プリンシパルの追加」、「IAM ユーザーとロール」を選択し、 OpenSearch ダッシュボードへのサインインとデータのインデックス作成に使用するユーザーまたはロールを選択します。[保存] を選択します。
11. [Index permissions] (インデックス作成の許可) で、すべての許可を選択します。
12. [次へ] をクリックします。
13. アクセスポリシーの設定では、[Create a new data access policy] (新しいデータアクセスポリシーとして作成する) を選択し、ポリシーに [movies] という名前を付けます。
14. [次へ] をクリックします。
15. コレクションの設定を確認して、[Submit] (送信) を選択します。コレクションステータスが `Active` になるまで数分待機します。

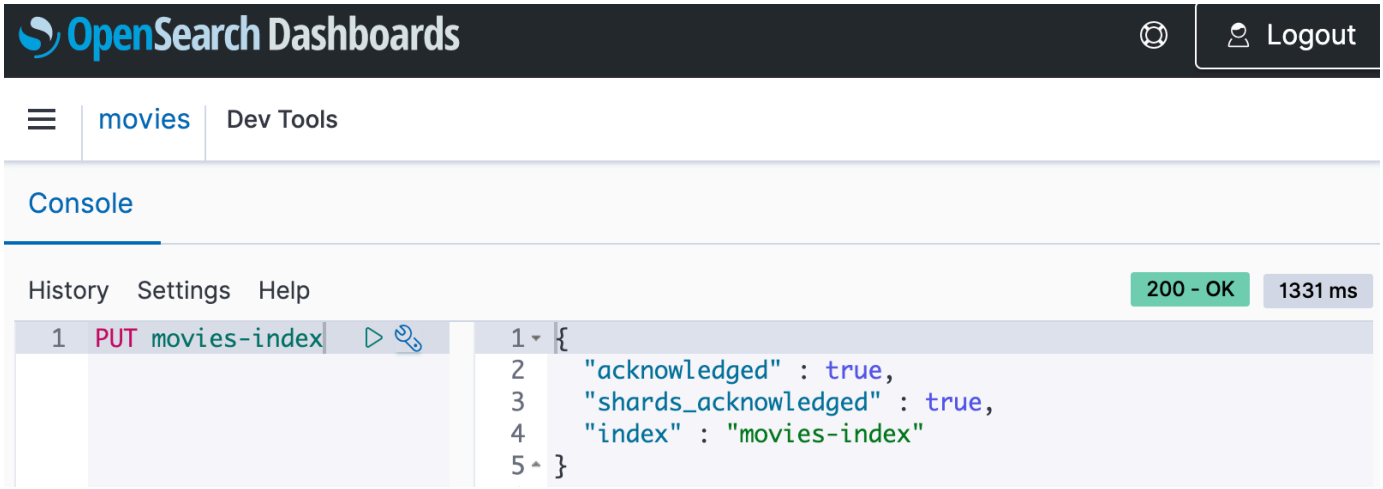
ステップ 3: データをアップロードして検索する

Postman または cURL を使用して、 OpenSearch サーバーレスコレクションにデータをアップロードできます。簡略化のため、これらの例では OpenSearch Dashboards コンソール内で開発ツールを使用します。

movies コレクションのデータをインデックス化して検索するには

1. 左側のナビゲーションペインで [Collections] (コレクション) を選択した後、「movies」コレクションを選択してその詳細ページを開きます。
2. コレクションの OpenSearch Dashboards URL を選択します。この URL の形式は、`https://dashboards.{region}.aoss.amazonaws.com/_login/?collectionId={collection-id}` になります。
3. OpenSearch ダッシュボードで、左側のナビゲーションペインを開き、開発ツール を選択します。
4. 次のリクエストを送信し、「movies-index」というインデックスを 1 つ作成します。

```
PUT movies-index
```



5. 1 つのドキュメントを 「movies-index」 にインデックスするために、次のリクエストを送信します。

```
PUT movies-index/_doc/1
{
  "title": "Shawshank Redemption",
  "genre": "Drama",
  "year": 1994
}
```

6. OpenSearch Dashboards でデータを検索するには、少なくとも 1 つのインデックスパターンを設定する必要があります。 OpenSearch はこれらのパターンを使用して、分析するインデックスを識別します。左のナビゲーションパネルを開き、[Stack Management] (スタック管理)、[Index Patterns] (インデックスパターン) の順に選択してから、[Create index pattern] (インデックスパターンを作成) を選択します。このチュートリアルでは、movies と入力します。
7. [次のステップ] を選択してから、[インデックスパターンの作成] を選択します。パターンが作成されたら、`title` および `genre` などのさまざまなドキュメントフィールドを表示できます。
8. データの検索を開始するには、左のナビゲーションパネルを再度開き [Discover] (検出) を選択するか、開発ツールの検索 API を使用します。

ステップ 4: コレクションを削除する

「movies」コレクションはテスト用のため、試用を終了したら確実に削除します。

OpenSearch Serverless コレクションを削除するには

1. Amazon OpenSearch Service コンソールに戻ります。
2. 左側のナビゲーションペインで [Collections] (コレクション) を選択した後、「movies」コレクションを選択します。
3. [削除] を選択して、削除を確認します。

次のステップ

ここまでで、コレクションおよびデータインデックスの作成に関する練習が終わっています。さらに、以下の演習のいくつかも有用です。

- コレクションを作成するための、高度なオプションを確認してみてください。詳細については、「the section called “コレクションの作成、一覧表示、および削除”」を参照してください。
- コレクションのセキュリティを大規模に管理するための、セキュリティポリシーの設定方法を説明しています。詳細については、「the section called “ OpenSearch サーバーレスのセキュリティ”」を参照してください。
- コレクション内でデータをインデックス化するための、他の方法を学びます。詳細については、「the section called “コレクションへのデータの取り込み”」を参照してください。

Amazon OpenSearch Serverless コレクションの作成と管理

Amazon OpenSearch Serverless コレクションは、コンソール、AWS CLI と API、AWS SDK、および AWS CloudFormation を使用して作成できます。

トピック

- Amazon OpenSearch サーバーレスコレクションの作成、一覧表示、削除
- ベクトル検索コレクションの使用
- Amazon OpenSearch Serverless でデータライフサイクルポリシーを使用する
- AWS SDK を使用した Amazon OpenSearch Serverless の操作
- AWS CloudFormation を使用した Amazon OpenSearch Serverless コレクションの作成

Amazon OpenSearch サーバーレスコレクションの作成、一覧表示、削除

Amazon OpenSearch Serverless のコレクションは、分析ワークロードを表す 1 つ以上のインデックスを論理的にグループ化したものです。 OpenSearch サービスはコレクションを自動的に管理および調整するため、手動入力は最小限で済みます。

トピック

必要なアクセス許可

OpenSearch サーバーレスでは、次の AWS Identity and Access Management (IAM) 権限を使用してコレクションを作成および管理します。IAM 条件を指定して、ユーザーを特定のコレクションに制限できます。

- `aoss:CreateCollection` – コレクションを作成します。
- `aoss:ListCollections` – 現在のアカウントのコレクションを一覧表示します。
- `aoss:BatchGetCollection` – 1 つまたは複数のコレクションに関する詳細情報を取得します。
- `aoss:UpdateCollection` – コレクションを変更します。
- `aoss:DeleteCollection` – コレクションを削除します。

次の ID ベースのアクセスポリシーのサンプルでは、ユーザーが `Logs` という名前の 1 つのコレクションを管理するのに必要な最小限のアクセス許可が付与されています。

```
[
   {
      "Sid":"Allows managing logs collections",
      "Effect":"Allow",
      "Action":[
         "aoss:CreateCollection",
         "aoss:ListCollections",
         "aoss:BatchGetCollection",
```

```
            "aoss:UpdateCollection",
            "aoss:DeleteCollection",
            "aoss:CreateAccessPolicy",
            "aoss:CreateSecurityPolicy"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"Logs"
            }
         }
      }
]
```

コレクションを正常に機能させるには、暗号化、ネットワーク、およびデータアクセスポリシーが必要なため、aoss:CreateAccessPolicy と aoss:CreateSecurityPolicy が含まれています。詳細については、「the section called “ID とアクセス管理”」を参照してください。

> **Note**
>
> アカウントで最初のコレクションを作成する場合は、iam:CreateServiceLinkedRole アクセス許可も必要です。詳細については、「the section called “コレクション作成ロール”」を参照してください。

コレクションの作成

AWS CLI コンソールまたはを使用してサーバーレスコレクションを作成できます。以下のステップは、検索コレクションまたは時系列コレクションの作成方法について説明します。ベクター検索コレクションを作成するには、「the section called “ベクトル検索コレクションの使用”」を参照してください。

コレクションを作成する (コンソール)

コンソールを使用してコレクションを作成するには

1. https://console.aws.amazon.com/aos/home/ にある Amazon OpenSearch サービスコンソールに移動します。
2. 左側のナビゲーションペインで [Serverless] (サーバーレス) を展開し、[Collections] (コレクション) を選択します。

3. [Create collection] (コレクションの作成) を選択します。
4. コレクションの名前と説明を入力します。名前は次の基準を満たしている必要があります。

   - お客様のアカウントに固有であり、 AWS リージョン
   - 先頭が小文字
   - 3～32 文字
   - 小文字の a～z、0～9 の数字、ハイフン (-) のみ含まれる
5. コレクションタイプを次の中から選択します。

   - [Search] (検索) – 社内ネットワークのアプリケーションやインターネットに接続するアプリケーションに使用される全文検索。すべての検索データはホットストレージに保存され、クエリの応答時間を短縮できます。
   - [Time series] (時系列) – マシン生成の大量の半構造化データの分析に焦点を当てたログ分析セグメント。少なくとも 24 時間のデータはホットインデックスに保存され、残りはウォームストレージに残ります。
   - [ベクトル検索] – ベクトルデータ管理を簡素化するベクトル埋め込みのセマンティック検索。機械学習 (ML) 拡張検索エクスペリエンスと、チャットボット、パーソナルアシスタント、不正行為検出などの生成 AI アプリケーションを強化します。

   詳細については、「the section called “コレクションタイプを選択する”」を参照してください。
6. 「デプロイタイプ」で、コレクションの冗長設定を選択します。デフォルトでは、各コレクションは冗長性をもって作成されます。つまり、インデックス作成と検索の OpenSearch Compute Unit (OCU) は、それぞれ異なるアベイラビリティーゾーンに独自のスタンバイレプリカを持っています。開発とテストの目的で冗長性を無効にして、コレクション内の OCU の数を 2 つに減らすことができます。詳細については、「the section called “仕組み”」を参照してください。
7. 「暗号化」で、 AWS KMS データを暗号化する鍵を選択します。 OpenSearch サーバーレスは、入力したコレクション名が暗号化ポリシーで定義されたパターンと一致する場合に通知します。このマッチングを維持するか、独自の暗号化設定でオーバーライドするかを選択できます。詳細については、「the section called “暗号化”」を参照してください。
8. [Network access settings] (ネットワークアクセス設定) で、コレクションのネットワークアクセスを設定します。

   - [アクセスタイプ] には、[パブリック] または [プライベート] を選択します。次に、どの VPC AWS のサービス エンドポイントとコレクションにアクセスできるかを指定します。

- アクセス用 VPC エンドポイント — アクセスを許可する VPC エンドポイントを 1 つ以上指定します。VPC エンドポイントの作成については、「the section called “VPC エンドポイント”」を参照してください。
- AWS のサービス プライベートアクセス — アクセスを許可するサポート対象のサービスを 1 つ以上選択します。

- リソースタイプでは、OpenSearchコレクションにエンドポイント (curl、Postman などを通じて API 呼び出しを行う場合)、OpenSearch ダッシュボードエンドポイント (ビジュアライゼーションを操作し、コンソールから API 呼び出しを行う場合)、あるいはその両方からアクセス可能にするかを選択します。

> **Note**
>
> AWS のサービス プライベートアクセスはエンドポイントにのみ適用され、Dashboards OpenSearch エンドポイントには適用されません。 OpenSearch

OpenSearch サーバーレスは、入力したコレクション名がネットワークポリシーで定義されたパターンと一致する場合に通知します。このマッチングを維持するか、カスタムネットワーク設定でオーバーライドするかを選択できます。詳細については、「the section called “ネットワークアクセス”」を参照してください。

9. (オプション) コレクションに 1 つまたは複数のタグを追加します。詳細については、「the section called “タグコレクション”」を参照してください。
10. [次へ] を選択します。
11. コレクションのデータアクセスルールを設定します。これは、コレクション内のデータに誰がアクセスできるかを定義します。作成するルールごとに、以下のステップを実行します。

    - [Add principals] (プリンシパルの追加) を選択し、データアクセス権が付与される IAM ロール、または SAML ユーザーとグループを 1 つ、または複数選択します。
    - [Grant permissions] (許可の付与) で、関連付けられたプリンシパルに付与するエイリアス、テンプレート、およびインデックス許可を選択します。アクセス許可とそれによって許可されるアクセスの完全なリストについては、「the section called “サポートされている OpenSearch API オペレーションとアクセス許可”」を参照してください。

    OpenSearch サーバーレスは、入力したコレクション名がデータアクセスポリシーで定義されたパターンと一致するかどうかを通知します。この一致を維持するか、固有のデータアクセス設定

で上書きするかを選択できます。詳細については、「the section called “データアクセスコントロール”」を参照してください。

12. [次へ] を選択します。
13. [Data access policy settings] (データアクセスポリシー設定) で、先ほど作成したルールで実行する事柄を選択します。これらを使用して新しいデータアクセスポリシーを作成する、またはこれらを既存のポリシーに追加することができます。
14. コレクション設定を確認してから、[Submit] (送信) を選択します。

OpenSearch Serverless がコレクションを作成すると、`Creating`コレクションのステータスはに変わります。

コレクションを作成する (CLI)

を使用してコレクションを作成する前に AWS CLI、コレクションの意図した名前と一致するリソースパターンを含む暗号化ポリシーが必要です。例えば、コレクションに「logs-application」という名前を付ける場合は、次のような暗号化ポリシーを作成します。

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource
\":[\"collection\/logs-application\"]}],\"AWSOwnedKey\":true}"
```

このポリシーを別のコレクションにも使用する場合は、`collection/logs*` や `collection/*` のようにルールを拡張しておくことができます。

また、コレクションのネットワーク設定をネットワークポリシーの形式で設定する必要があります。前述の「logs-application」の例を使用すると、次のネットワークポリシーを作成できます。

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type network --policy "[{\"Description\":\"Public access for logs collection
\",\"Rules\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/logs-
application\"]},{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/logs-
application\"]}],\"AllowFromPublic\":true}]"
```

> **Note**
>
> ネットワークポリシーは、コレクションを作成した後でも作成できますが、事前に作成することをお勧めします。

コレクションを作成するには、CreateCollection以下のリクエストを送信してください。

```
aws opensearchserverless create-collection --name "logs-application" --type SEARCH --
description "A collection for storing log data"
```

type の場合、SEARCH または TIMESERIES のいずれかを指定します。詳細については、「the section called “コレクションタイプを選択する”」を参照してください。

レスポンス例

```
{
    "createCollectionDetail": {
        "id": "07tjusf2h91cunochc",
        "name": "books",
        "description":"A collection for storing log data",
        "status": "CREATING",
        "type": "SEARCH",
        "kmsKeyArn": "auto",
        "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
        "createdDate": 1665952577473
    }
}
```

リクエストでコレクションタイプを指定しない場合、デフォルトは TIMESERIES です。コレクションが AWS 所有のキーで暗号化されている場合、kmsKeyArn は ARN ではなく auto です。

> **Important**
>
> コレクション作成後は、データアクセスポリシーと一致しない限りアクセスできません。データアクセスポリシーを作成する手順については、「the section called “データアクセスコントロール”」を参照してください。

OpenSearch ダッシュボードへのアクセス

を使用してコレクションを作成すると AWS Management Console、 OpenSearch コレクションのダッシュボード URL に移動できます。Dashboards URL を確認するには、ナビゲーションペインで [Collections] (コレクション) を選択し、コレクションを選択して詳細ページを開きます。この URL の形式は、`https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc` になります。URL に移動すると、Dashboards に自動的ログインします。

OpenSearch ダッシュボード URL はすでにあるがにアクセスしていない場合は AWS Management Console、ブラウザからダッシュボード URL を呼び出すと、コンソールにリダイレクトされます。AWS 認証情報を入力すると、自動的にダッシュボードにログインします。SAML のコレクションへのアクセスについて詳しくは、「SAML OpenSearch によるダッシュボードへのアクセス」を参照してください。

OpenSearch ダッシュボードコンソールのタイムアウトは 1 時間で、設定できません。

> **Note**
>
> 2023 年 5 月 10 日、 OpenSearch ダッシュボード用の共通グローバルエンドポイントが導入されました。 OpenSearch この形式の URL を使用して、 OpenSearch ブラウザーでダッシュボードに移動できるようになりました。`https://dashboards.us-east-1.aoss.amazonaws.com/_login/?collectionId=07tjusf2h91cunochc`下位互換性を確保するため、 OpenSearch この形式ではコレクション固有の既存のダッシュボードエンドポイントを引き続きサポートします。`https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/_dashboards`

コレクションの表示

Amazon OpenSearch サービスコンソールの [コレクション] タブで既存のコレクションを表示できます。 AWS アカウント

コレクションとその ID を一覧表示するには、ListCollectionsリクエストを送信してください。

```
aws opensearchserverless list-collections
```

レスポンス例

```
{
```

```
   "collectionSummaries":[
      {
         "arn":"arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "id":"07tjusf2h91cunochc",
         "name":"my-collection",
         "status":"CREATING"
      }
   ]
}
```

検索結果を制限するには、コレクションフィルターを使用します。このリクエストは、ACTIVE 状態のコレクションへのレスポンスを次のようにフィルタリングします。

```
aws opensearchserverless list-collections --collection-filters '{ "status": "ACTIVE" }'
```

エンドポイントや OpenSearch Dashboards OpenSearch エンドポイントなど、1 つ以上のコレクションに関する詳細情報を取得するには、BatchGetCollectionリクエストを送信してください。

```
aws opensearchserverless batch-get-collection --ids ["07tjusf2h91cunochc",
 "1iu5usc4rame"]
```

> **Note**
>
> リクエストには --names または --ids を含めることができますが、両方を含めることはできません。

レスポンス例

```
{
   "collectionDetails":[
      {
         "id": "07tjusf2h91cunochc",
         "name": "my-collection",
         "status": "ACTIVE",
         "type": "SEARCH",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
```

```
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://07tjusf2h91cunochc.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/
_dashboards"
      },
      {
         "id": "178ukvtg3i82dvopdid",
         "name": "another-collection",
         "status": "ACTIVE",
         "type": "TIMESERIES",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/178ukvtg3i82dvopdid",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com/_dashboards"
      }
   ],
   "collectionErrorDetails":[]
}
```

コレクションの削除

コレクションを削除すると、コレクション内のすべてのデータとインデックスが削除されます。削除したコレクションを復元することはできません。

コンソールを使用してコレクションを削除するには

1. Amazon OpenSearch サービスコンソールのコレクションパネルから、削除するコレクションを選択します。
2. [削除] を選択して、削除を確認します。

を使用してコレクションを削除するには AWS CLI、DeleteCollection以下のリクエストを送信してください。

```
aws opensearchserverless delete-collection --id 07tjusf2h91cunochc
```

レスポンス例

```
{
   "deleteCollectionDetail":{
      "id":"07tjusf2h91cunochc",
      "name":"my-collection",
      "status":"DELETING"
   }
}
```

ベクトル検索コレクションの使用

OpenSearch サーバーレスのベクター検索コレクションタイプは、スケーラブルで高性能な類似検索機能を提供します。これにより、基盤となるベクトルデータベースインフラストラクチャを管理することなく、最新の機械学習 (ML) によって拡張された検索エクスペリエンスや生成人工知能 (AI) アプリケーションを簡単に構築できます。

ベクトル検索コレクションのユースケースには、画像検索、ドキュメント検索、音楽検索、製品のレコメンデーション、動画検索、位置ベースの検索、不正検出、異常検出などがあります。

OpenSearch サーバーレス用のベクターエンジンは、の k-最近隣 (k-NN) 検索機能を搭載しているため OpenSearch、サーバーレス環境のシンプルさで同じ機能を利用できます。このエンジンは k-NN API オペレーションをサポートします。 OpenSearch これらの操作により、全文検索、高度なフィルタリング、集計、地理空間クエリ、データの高速取得のためのネストされたクエリ、および拡張された検索結果を利用できます。

ベクトルエンジンは、ユークリッド距離、コサイン類似度、ドット積などの距離メトリクスを提供し、16,000 次元に対応できます。数値、ブール値、日付、キーワード、ジオポイントなどのさまざまなデータ型を持つフィールドを保存できます。また、保存されたベクトルにさらにコンテキストを追加するための説明情報用のテキストを含むフィールドを保存することもできます。データ型を同一場所に配置することで、複雑さが軽減され、保守性が向上するとともに、データの重複、バージョン互換性の問題、ライセンスの問題を回避できます。

ベクトル検索コレクションの開始方法

このチュートリアルでは、次のステップを実行して、ベクトル埋め込みをリアルタイムで保存、検索、取得します。

1. アクセス許可を設定する
2. コレクションを作成する

3. データをアップロードおよび検索する
4. コレクションを削除する

ステップ 1: アクセス許可を設定する

このチュートリアルを完了する (OpenSearch そして一般的にサーバーレスを使用する) には、正しい AWS Identity and Access Management (IAM) 権限が必要です。このチュートリアルでは、コレクションを作成し、データをアップロードして検索を行い、最後にそのコレクションを削除します。

ユーザーまたはロールには、以下の最低限の許可を含む ID ベースのポリシーが、アタッチされている必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

OpenSearch サーバーレス IAM 権限の詳細については、を参照してください。the section called “ID とアクセス管理”

ステップ 2: コレクションを作成する

コレクションは、 OpenSearch 特定のワークロードやユースケースをサポートするために連携するインデックスのグループです。

OpenSearch サーバーレスコレクションを作成するには

1. https://console.aws.amazon.com/aos/home にある Amazon OpenSearch サービスコンソールを開きます。
2. 左側のナビゲーションペインで [Collections] (コレクション) 、[Create collection] (コレクションを作成) を選択します。
3. コレクションに [housing] という名前を付けます。
4. コレクションタイプで、[ベクトル検索] を選択します。詳細については、「the section called "コレクションタイプを選択する"」を参照してください。
5. [デプロイのタイプ] で、[冗長性を有効化 (アクティブレプリカ)] をオフにします。これにより、開発モードまたはテストモードでコレクションが作成され、 OpenSearch コレクション内のコンピュートユニット (OCU) の数が 2 つに減ります。このチュートリアルで本番環境を作成する場合は、チェックボックスをオンのままにしてください。
6. [セキュリティ] で、[簡単作成] を選択してセキュリティ設定を合理化します。ベクトルエンジン内のすべてのデータは、転送中であっても、保管中であっても、デフォルトで暗号化されます。ベクトルエンジンは、きめ細かい (IAM) アクセス許可をサポートしているため、暗号化、ネットワーク、コレクション、インデックスを作成、更新、削除できるユーザーを定義できます。
7. [次へ] を選択します。
8. コレクションの設定を確認して、[Submit] (送信) を選択します。コレクションステータスが `Active` になるまで数分待機します。

ステップ 3: データをアップロードして検索する

インデックスとは、ベクトル埋め込みや他のフィールドを保存、検索、取得する方法を提供する共通のデータスキーマを持つドキュメントのコレクションです。 OpenSearch ダッシュボードの開発ツールコンソール、または Postman や awscurl などの HTTP ツールを使用して、 OpenSearch サーバーレスコレクションのインデックスにデータを作成してアップロードできます。このチュートリアルでは Dev Tools を使用します。

movies コレクションのデータをインデックス化して検索するには

1. 新しいコレクション用に 1 つのインデックスを作成するには、Dev Tools コンソールで次のリクエストを送信します。デフォルトでは、これにより、`nmslib` エンジンとユークリッド距離を使用してインデックスが作成されます。

```
PUT housing-index
```

```
{
   "settings": {
      "index.knn": true
   },
   "mappings": {
      "properties": {
         "housing-vector": {
            "type": "knn_vector",
            "dimension": 3
         },
         "title": {
            "type": "text"
         },
         "price": {
            "type": "long"
         },
         "location": {
            "type": "geo_point"
         }
      }
   }
}
```

2. 1 つのドキュメントを housing-index にインデックスするために、次のリクエストを送信します。

```
POST housing-index/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. インデックス内のプロパティに類似したプロパティを検索するには、次のクエリを送信します。

```
GET housing-index/_search
{
    "size": 5,
```

```
    "query": {
        "knn": {
            "housing-vector": {
                "vector": [
                    10,
                    20,
                    30
                ],
                "k": 5
            }
        }
    }
}
```

ステップ 4: コレクションを削除する

housing コレクションはテスト用のため、試用を終了したら確実に削除してください。

OpenSearch サーバーレスコレクションを削除するには

1. Amazon OpenSearch サービスコンソールに戻ります。
2. 左側のナビゲーションペインで [コレクション] を選択した後、[プロパティ] コレクションを選択します。
3. [削除] を選択し、削除を確定します。

フィルタリングされた検索

フィルターを使用して、セマンティック検索の結果を絞り込むことができます。インデックスを作成し、ドキュメントに対してフィルター検索を実行するには、前のチュートリアルの「データをアップロードおよび検索する」を次の手順に置き換えます。他のステップは同じです。フィルターの詳細については、「フィルターを使用した k-NN 検索」を参照してください。

movies コレクションのデータをインデックス化して検索するには

1. コレクションに 1 つのインデックスを作成するには、Dev Tools コンソールで次のリクエストを送信します。

```
PUT housing-index-filtered
{
  "settings": {
```

```
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "housing-vector": {
        "type": "knn_vector",
        "dimension": 3,
        "method": {
          "engine": "faiss",
          "name": "hnsw"
        }
      },
      "title": {
        "type": "text"
      },
      "price": {
        "type": "long"
      },
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

2. 1 つのドキュメントをインデックスに登録するにはhousing-index-filtered、以下のリクエストを送信します。

```
POST housing-index-filtered/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. 地理的位置から指定された距離内にある、指定された価格未満のシアトルのアパートのデータを検索するには、次のリクエストを送信します。

```
GET housing-index-filtered/_search
{
  "size": 5,
  "query": {
    "knn": {
      "housing-vector": {
        "vector": [
          0.1,
          0.2,
          0.3
        ],
        "k": 5,
        "filter": {
          "bool": {
            "must": [
              {
                "query_string": {
                  "query": "Find me 2 bedroom apartment in Seattle under $3000 ",
                  "fields": [
                    "title"
                  ]
                }
              },
              {
                "range": {
                  "price": {
                    "lte": 3000
                  }
                }
              },
              {
                "geo_distance": {
                  "distance": "100miles",
                  "location": {
                    "lat": 48,
                    "lon": 121
                  }
                }
              }
            ]
          }
        }
      }
```

```
    }
  }
}
```

数十億規模のワークロード

ベクトル検索コレクションは、数十億のベクトルを使用するワークロードをサポートします。スケーリングのためにインデックスを再作成する必要はありません。自動スケーリングが自動的にこれを実行します。次元数の多いベクトルが数百万（またはそれ以上）あり、200を超えるOCUが必要な場合は、AWS Support に連絡して、 OpenSearch アカウントの最大計算単位（OCU）を引き上げてください。

制限事項

ベクトル検索コレクションには次の制限があります。

- ベクトル検索コレクションは、Apache Lucene ANN エンジンをサポートしていません。
- ベクター検索コレクションは Faiss を含む HNSW アルゴリズムのみをサポートし、IVF と IVFQ はサポートしていません。
- ベクトル検索コレクションは、ウォームアップ、統計、モデルトレーニング API をサポートしていません。
- ベクトル検索コレクションは、インラインスクリプトまたはストアドスクリプトをサポートしていません。
- インデックス数情報は、 AWS Management Console ベクター検索コレクションにはありません。
- ベクトル検索コレクションのインデックスの更新間隔は 60 秒です。

次のステップ

これで、ベクトル検索コレクションおよびインデックスデータの作成方法がわかりました。さらに、次のいくつかの演習も有用です。

- OpenSearch Python クライアントを使用してベクター検索コレクションを操作します。に関するこのチュートリアルを参照してくださいGitHub。
- OpenSearch Java クライアントを使用してベクター検索コレクションを操作します。に関するこのチュートリアルを参照してくださいGitHub。

- LangChain OpenSearch ベクターストアとして使用するように設定します。 LangChain 言語モデルを利用したアプリケーションを開発するためのオープンソースフレームワークです。詳細については、LangChain ドキュメントを参照してください。

Amazon OpenSearch Serverless でデータライフサイクルポリシーを使用する

Amazon OpenSearch Serverless 「時系列」コレクションのデータライフサイクルポリシーは、そのコレクション内のデータの有効期間を決定します。OpenSearch Serverless は、設定した期間にわたってデータを保持します。

AWS アカウント内の「各時系列」コレクションのインデックスごとに、個別のデータライフサイクルポリシーを設定できます。OpenSearch Serverless は、少なくともポリシーで設定した保持期間の間、ドキュメントをインデックスに保持します。その後、ベストエフォート方式で、通常は 48 時間以内または保存期間の 10% 以内 (どちらか長い方) に自動的に削除します。

データライフサイクルポリシーをサポートしているのは「時系列」コレクションだけです。「検索コレクション」や「ベクトル検索」コレクションではサポートされていません。

トピック

データライフサイクルポリシー

データライフサイクルポリシーでは、一連の「ルール」を指定します。データライフサイクルポリシーでは、これらのルールに一致するインデックスまたはコレクションに関連するデータの保持期間を管理できます。これらのルールは、インデックスまたはインデックスグループ内のデータの保存期

間を定義します。各ルールは、リソースタイプ (index)、保持期間、および保持期間が適用されるリソース (インデックス) のリストで構成されています。

保持期間は、以下のいずれかの形式で定義します。

- "MinIndexRetention": "24h" — OpenSearch Serverless は、指定された期間、インデックスデータを時間単位または日単位で保持します。この期間は、24h ~ 3650d で設定できます。
- "NoMinIndexRetention": true — OpenSearch Serverless はインデックスデータを無期限に保持します。

次のサンプルポリシーでは、最初のルールがコレクション marketing 内のすべてのインデックスに対して 15 日間の保持期間を指定します。2 番目のルールは、finance コレクション内の log で始まるすべてのインデックス名には保存期間が設定されず、無期限に保持されることを指定しています。

```
{
   "lifeCyclePolicyDetail": {
      "type": "retention",
      "name": "my-policy",
      "policyVersion": "MTY4ODI0NTM2OTk1N18x",
      "policy": {
         "Rules": [
            {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/*"
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/finance/log*"
            ],
            "NoMinIndexRetention": true
         }
         ]
      },
      "createdDate": 1688245369957,
      "lastModifiedDate": 1688245369957
   }
```

```
}
```

以下のサンプルポリシールールでは、OpenSearch Serverless はアカウント内のすべてのコレクションのすべてのインデックスのデータを無期限に保持します。

```
{
   "Rules": [
      {
         "ResourceType": "index",
         "Resource": [
            "index/*/*"
         ]
      }
   ],
   "NoMinIndexRetention": true
}
```

必要な許可

OpenSearch Serverless のライフサイクルポリシーは、次の AWS Identity and Access Management (IAM) アクセス許可を使用します。IAM 条件を指定して、特定のコレクションとインデックスに関連付けられたデータライフサイクルポリシーにユーザーを制限できます。

- `aoss:CreateLifecyclePolicy` – データライフサイクルポリシーを作成します。
- `aoss:ListLifecyclePolicies` – 現在のアカウント内のすべてのデータライフサイクルポリシーを一覧表示します。
- `aoss:BatchGetLifecyclePolicy` – アカウントまたはポリシー名に関連付けられたデータライフサイクルポリシーを表示します。
- `aoss:BatchGetEffectiveLifecyclePolicy` – 特定のリソース (`index` がサポートされている唯一のリソース) のデータライフサイクルポリシーを表示します。
- `aoss:UpdateLifecyclePolicy` – 特定のデータライフサイクルポリシーを変更し、その保持設定またはリソースを変更します。
- `aoss:DeleteLifecyclePolicy` – データライフサイクルポリシーを削除します。

次の ID ベースのアクセスポリシーにより、ユーザーはすべてのデータライフサイクルポリシーを表示し、リソースパターン `collection/application-logs` を含むポリシーの更新が行えるようになります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListLifecyclePolicies",
                "aoss:BatchGetLifecyclePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

ポリシーの優先順位

ポリシー内またはポリシー間に、複数のデータライフサイクルポリシールールが同時に存在する場合があります。この場合、「両方」のルールに共通するインデックスのより一般的なリソース名またはパターンを含むルールよりもインデックスに対するリソース名またはパターンがより具体的であるルールが優先されます。

例えば、次のポリシーでは、1 つのインデックス index/sales/logstash に 2 つのルールが適用されます。この場合、index/sales/log* が index/sales/logstash に対する最長の一致なので 2 番目のルールが優先されます。そのため、OpenSearch Serverless はインデックスの保存期間を設定しません。

```
{
      "Rules":[
         {
            "ResourceType":"index",
```

```
            "Resource":[
               "index/sales/*",
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/log*",
            ],
            "NoMinIndexRetention": true
         }
      ]
   }
```

ポリシー構文

1 つ以上のルールを指定します。これらのルールは OpenSearch Serverless インデックスのデータライフサイクル設定を定義します。

各ルールには以下の要素が含まれます。MinIndexRetention または NoMinIndexRetention を指定できますが、両方を指定することはできません。

要素	説明
リソースタイプ	ルールが適用されるリソースのタイプ。データライフサイクルポリシーでサポートされる唯一のオプションは index です。
[Resource] (リソース)	リソース名やパターンのリスト。パターンはプレフィックスとワイルドカード (*) で構成され、関連付けられたアクセス許可を複数のリソースに適用することを可能にします。例えば、index/*<collection-name|pattern>* /*<index-name|pattern>* です。
MinIndexRetention	ドキュメントをインデックスに保持する最小期間: 日 (d) または時間 (h)。下限は 24h で、上限は 3650d です。

要素	説明
NoMinIndexRetention	`true` の場合、OpenSearch Serverless はドキュメントを無期限に保持します。

次に例をいくつか示します。

```
{
  "Rules": [
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/*"
      ],
      "MinIndexRetention": "20d"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/auto*/gear"
      ],
      "MinIndexRetention": "24h"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/tires"
      ],
      "NoMinIndexRetention": true
    }
  ]
}
```

データライフサイクルポリシー (AWS CLI) を作成する

OpenSearch Serverless API オペレーションを使用してデータライフサイクルポリシーを作成するには、CreateLifecyclePolicy コマンドを使用します。このコマンドは、インラインポリシーと .json ファイルの両方を受け入れます。インラインポリシーは JSON エスケープ文字列としてエンコードする必要があります。

次のリクエストはデータライフサイクルポリシーを作成します。

```
aws opensearchserverless create-lifecycle-policy \
    --name my-policy \
    --type retention \
    --policy "{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":[\"index/autoparts-inventory/*\"],\"MinIndexRetention\": \"81d\"},{\"ResourceType\":\"index\",\"Resource\":[\"index/sales/orders*\"],\"NoMinIndexRetention\":true}]}"
```

このポリシーを JSON ファイルで指定するには、`--policy file://my-policy.json` の形式を使用します。

データライフサイクルポリシーを表示する

コレクションを作成する前に、アカウント内の既存のデータライフサイクルポリシーをプレビューして、コレクション名と一致するリソースパターンがあるポリシーを確認することをお勧めします。次の ListLifecyclePolicies リクエストは、アカウント内のすべてのデータライフサイクルポリシーを一覧表示します。

```
aws opensearchserverless list-lifecycle-policies --type retention
```

リクエストは、設定されているすべてのデータライフサイクルポリシーに関する情報を返します。1 つの特定のポリシーで定義されているパターンルールを表示するには、レスポンスの `lifecyclePolicySummaries` 要素の内容でポリシー情報を探します。このポリシーの `name` と `type` を書き留め、BatchGetLifecyclePolicy リクエストでこれらのプロパティを使用して、次のポリシーの詳細を含むレスポンスを受信します。

```
{
    "lifecyclePolicySummaries": [
        {
            "type": "retention",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
    ]
}
```

結果を特定のコレクションまたはインデックスを含むポリシーに限定するには、次のようにリソースフィルターを追加します。

```
aws opensearchserverless list-lifecycle-policies --type retention --resources
 "index/autoparts-inventory/*"
```

特定のポリシーの詳細情報を表示するには、BatchGetLifecyclePolicy コマンドを使用します。

データライフサイクルポリシーを更新する

データライフサイクルポリシーを変更すると、関連するすべてのコレクションが影響を受けます。OpenSearch Serverless コンソールでデータライフサイクルポリシーを更新するには、[データライフサイクルポリシー] を展開して変更するポリシーを選択した後、[編集] を選択します。変更を行ってから、[Save (保存)] を選択します。

OpenSearch Serverless API を使用してデータライフサイクルポリシーを更新するには、UpdateLifecyclePolicy コマンドを使用します。リクエストには、ポリシーのバージョンを含める必要があります。ポリシーのバージョンは、`ListLifecyclePolicies` または `BatchGetLifecyclePolicy` コマンドを使用して取得できます。最新のポリシーバージョンを含めると、他のユーザーによる変更を意図せず上書きしてしまうことがなくなります。

次のリクエストは、データライフサイクルポリシーを新しいポリシーの JSON ドキュメントで更新します。

```
aws opensearchserverless update-lifecycle-policy \
    --name my-policy \
    --type retention \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

ポリシーを更新してから新しい保持期間が適用されるまでに数分かかる場合があります。

データライフサイクルポリシーを削除する

データライフサイクルポリシーを削除すると、一致するインデックスに適用されなくなります。OpenSearch Serverless コンソールでポリシーを削除するには、ポリシーを選択し、[Delete] (削除) を選択します。

DeleteLifecyclePolicy コマンドを使用することもできます。

```
aws opensearchserverless delete-lifecycle-policy --name my-policy --type retention
```

AWS SDK を使用した Amazon OpenSearch Serverless の操作

このセクションには、AWS SDK を使用して Amazon OpenSearch Serverless を操作する方法の例が含まれています。これらのコードサンプルには、セキュリティポリシーとコレクションの作成方法が示されています。

> **Note**
>
> コードサンプルは、現在作成中です。コードサンプル (Java、Go など) を提供したい場合は、GitHub リポジトリ内で直接プルリクエストを開きます。

トピック

- Python
- JavaScript

Python

次のサンプルスクリプトでは、Python の opensearch-py クライアントおよび AWS SDK for Python (Boto3) を使用して、暗号化、ネットワーク、データアクセスポリシーを作成します。また、一致するコレクションを作成し、いくつかのサンプルデータをインデックス化します。

必要な従属関係をインストールには、次のコマンドを実行します。

```
pip install opensearch-py
pip install boto3
pip install botocore
pip install requests-aws4auth
```

スクリプト内の `Principal` 要素を、リクエストに署名するユーザーまたはロールの Amazon リソースネーム (ARN) に置き換えます。必要に応じて、`region` を変更することもできます。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3
import botocore
import time

# Build the client using the default credential configuration.
```

```
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

client = boto3.client('opensearchserverless')
service = 'aoss'
region = 'us-east-1'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def createEncryptionPolicy(client):
    """Creates an encryption policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Encryption policy for TV collections',
            name='tv-policy',
            policy="""
                {
                    \"Rules\":[
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ]
                        }
                    ],
                    \"AWSOwnedKey\":true
                }
                """,
            type='encryption'
        )
        print('\nEncryption policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] The policy name or rules conflict with an existing
 policy.')
        else:
            raise error


def createNetworkPolicy(client):
```

```
    """Creates a network policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Network policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Description\":\"Public access for TV collection\",
                    \"Rules\":[
                        {
                            \"ResourceType\":\"dashboard\",
                            \"Resource\":[\"collection\/tv-*\"]
                        },
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[\"collection\/tv-*\"]
                        }
                    ],
                    \"AllowFromPublic\":true
                }]
                """,
            type='network'
        )
        print('\nNetwork policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A network policy with this name already exists.')
        else:
            raise error


def createAccessPolicy(client):
    """Creates a data access policy that matches all collections beginning with tv-"""
    try:
        response = client.create_access_policy(
            description='Data access policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Rules\":[
                        {
                            \"Resource\":[
```

```
                            \"index\/tv-*\/*\"
                        ],
                        \"Permission\":[
                            \"aoss:CreateIndex\",
                            \"aoss:DeleteIndex\",
                            \"aoss:UpdateIndex\",
                            \"aoss:DescribeIndex\",
                            \"aoss:ReadDocument\",
                            \"aoss:WriteDocument\"
                        ],
                        \"ResourceType\": \"index\"
                    },
                    {
                        \"Resource\":[
                            \"collection\/tv-*\"
                        ],
                        \"Permission\":[
                            \"aoss:CreateCollectionItems\"
                        ],
                        \"ResourceType\": \"collection\"
                    }
                ],
                \"Principal\":[
                    \"arn:aws:iam::123456789012:role\/Admin\"
                ]
            }]
            """,
            type='data'
        )
        print('\nAccess policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] An access policy with this name already exists.')
        else:
            raise error


def createCollection(client):
    """Creates a collection"""
    try:
        response = client.create_collection(
            name='tv-sitcoms',
```

```
            type='SEARCH'
        )
        return(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A collection with this name already exists. Try
 another name.')
        else:
            raise error


def waitForCollectionCreation(client):
    """Waits for the collection to become active"""
    response = client.batch_get_collection(
        names=['tv-sitcoms'])
    # Periodically check collection status
    while (response['collectionDetails'][0]['status']) == 'CREATING':
        print('Creating collection...')
        time.sleep(30)
        response = client.batch_get_collection(
            names=['tv-sitcoms'])
    print('\nCollection successfully created:')
    print(response["collectionDetails"])
    # Extract the collection endpoint from the response
    host = (response['collectionDetails'][0]['collectionEndpoint'])
    final_host = host.replace("https://", "")
    indexData(final_host)


def indexData(host):
    """Create an index and add some sample data"""
    # Build the OpenSearch client
    client = OpenSearch(
        hosts=[{'host': host, 'port': 443}],
        http_auth=awsauth,
        use_ssl=True,
        verify_certs=True,
        connection_class=RequestsHttpConnection,
        timeout=300
    )
    # It can take up to a minute for data access rules to be enforced
    time.sleep(45)
```

```
    # Create index
    response = client.indices.create('sitcoms-eighties')
    print('\nCreating index:')
    print(response)

    # Add a document to the index.
    response = client.index(
        index='sitcoms-eighties',
        body={
            'title': 'Seinfeld',
            'creator': 'Larry David',
            'year': 1989
        },
        id='1',
    )
    print('\nDocument added:')
    print(response)


def main():
    createEncryptionPolicy(client)
    createNetworkPolicy(client)
    createAccessPolicy(client)
    createCollection(client)
    waitForCollectionCreation(client)


if __name__ == "__main__":
    main()
```

JavaScript

次のサンプルスクリプトでは、JavaScript の opensearch-js クライアントおよび SDK for JavaScript in Node.js を使用して、暗号化、ネットワーク、データアクセスポリシーを作成します。また、一致するコレクションおよびインデックスを作成して、いくつかのサンプルデータをインデックス化します。

必要な従属関係をインストールには、次のコマンドを実行します。

```
npm i aws-sdk
npm i aws4
npm i @opensearch-project/opensearch
```

スクリプト内の Principal 要素を、リクエストに署名するユーザーまたはロールの Amazon リソースネーム (ARN) に置き換えます。必要に応じて、region を変更することもできます。

```
var AWS = require('aws-sdk');
var aws4 = require('aws4');
var {
    Client,
    Connection
} = require("@opensearch-project/opensearch");
var {
    OpenSearchServerlessClient,
    CreateSecurityPolicyCommand,
    CreateAccessPolicyCommand,
    CreateCollectionCommand,
    BatchGetCollectionCommand
} = require("@aws-sdk/client-opensearchserverless");
var client = new OpenSearchServerlessClient();

async function execute() {
    await createEncryptionPolicy(client)
    await createNetworkPolicy(client)
    await createAccessPolicy(client)
    await createCollection(client)
    await waitForCollectionCreation(client)
}

async function createEncryptionPolicy(client) {
    // Creates an encryption policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Encryption policy for TV collections',
            name: 'tv-policy',
            type: 'encryption',
            policy: " \
        { \
            \"Rules\":[ \
                { \
                    \"ResourceType\":\"collection\", \
                    \"Resource\":[ \
                        \"collection\/tv-*\" \
                    ] \
                } \
            ], \
```

```
            \"AWSOwnedKey\":true \
        }"
        });
        const response = await client.send(command);
        console.log("Encryption policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] The policy name or rules conflict with an
 existing policy.');
        } else
            console.error(error);
    };
}

async function createNetworkPolicy(client) {
    // Creates a network policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Network policy for TV collections',
            name: 'tv-policy',
            type: 'network',
            policy: " \
            [{ \
                \"Description\":\"Public access for television collection\", \
                \"Rules\":[ \
                    { \
                        \"ResourceType\":\"dashboard\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    }, \
                    { \
                        \"ResourceType\":\"collection\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    } \
                ], \
                \"AllowFromPublic\":true \
            }]"
        });
        const response = await client.send(command);
        console.log("Network policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
```

```
            console.log('[ConflictException] A network policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createAccessPolicy(client) {
    // Creates a data access policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateAccessPolicyCommand({
            description: 'Data access policy for TV collections',
            name: 'tv-policy',
            type: 'data',
            policy: " \
            [{ \
                \"Rules\":[ \
                    { \
                        \"Resource\":[ \
                            \"index\/tv-*\/*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateIndex\", \
                            \"aoss:DeleteIndex\", \
                            \"aoss:UpdateIndex\", \
                            \"aoss:DescribeIndex\", \
                            \"aoss:ReadDocument\", \
                            \"aoss:WriteDocument\" \
                        ], \
                        \"ResourceType\": \"index\" \
                    }, \
                    { \
                        \"Resource\":[ \
                            \"collection\/tv-*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateCollectionItems\" \
                        ], \
                        \"ResourceType\": \"collection\" \
                    } \
                ], \
                \"Principal\":[ \
                    \"arn:aws:iam::123456789012:role\/Admin\" \
                ] \
```

```
            }]"
        });
        const response = await client.send(command);
        console.log("Access policy created:");
        console.log(response['accessPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] An access policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createCollection(client) {
    // Creates a collection to hold TV sitcoms indexes
    try {
        var command = new CreateCollectionCommand({
            name: 'tv-sitcoms',
            type: 'SEARCH'
        });
        const response = await client.send(command);
        return (response)
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A collection with this name already
 exists. Try another name.');
        } else
            console.error(error);
    };
}

async function waitForCollectionCreation(client) {
    // Waits for the collection to become active
    try {
        var command = new BatchGetCollectionCommand({
            names: ['tv-sitcoms']
        });
        var response = await client.send(command);
        while (response.collectionDetails[0]['status'] == 'CREATING') {
            console.log('Creating collection...')
            await sleep(30000) // Wait for 30 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
```

```
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        console.log('Collection successfully created:');
        console.log(response['collectionDetails']);
        // Extract the collection endpoint from the response
        var host = (response.collectionDetails[0]['collectionEndpoint'])
        // Pass collection endpoint to index document request
        indexDocument(host)
    } catch (error) {
        console.error(error);
    };
}

async function indexDocument(host) {

    var client = new Client({
        node: host,
        Connection: class extends Connection {
            buildRequestObject(params) {
                var request = super.buildRequestObject(params)
                request.service = 'aoss';
                request.region = 'us-east-1'; // e.g. us-east-1
                var body = request.body;
                request.body = undefined;
                delete request.headers['content-length'];
                request.headers['x-amz-content-sha256'] = 'UNSIGNED-PAYLOAD';
                request = aws4.sign(request, AWS.config.credentials);
                request.body = body;

                return request
            }
        }
    });

    // Create an index
    try {
        var index_name = "sitcoms-eighties";

        var response = await client.indices.create({
            index: index_name
        });
```

```
        console.log("Creating index:");
        console.log(response.body);

        // Add a document to the index
        var document = "{ \"title\": \"Seinfeld\", \"creator\": \"Larry David\", \"year
\": \"1989\" }\n";

        var response = await client.index({
            index: index_name,
            body: document
        });

        console.log("Adding document:");
        console.log(response.body);
    } catch (error) {
        console.error(error);
    };
}

execute()
```

AWS CloudFormation を使用した Amazon OpenSearch Serverless コレクションの作成

AWS CloudFormation を使用して、コレクション、セキュリティポリシー、VPC エンドポイントなどの Amazon OpenSearch Serverless リソースを作成できます。OpenSearch Serverless CloudFormation に関する包括的なリファレンスについては、「AWS CloudFormation ユーザーガイド」の「Amazon OpenSearch Serverless」を参照してください。

次の CloudFormation サンプルテンプレートでは、シンプルなデータアクセスポリシー、ネットワークポリシー、セキュリティポリシー、および一致するコレクションが作成されます。これは、Amazon OpenSearch Serverless をすぐに立ち上げて実行し、コレクションを作成して使用するために必要な要素をプロビジョニングする良い方法です。

> **Important**
>
> この例ではパブリックネットワークアクセスを使用していますが、これは本番環境のワークロードにはお勧めしません。コレクションを保護するために VPC アクセスを使用すること

をお勧めします。詳細については、「AWS::OpenSearchServerless::VpcEndpoint」と「the section called “VPC エンドポイント”」を参照してください。

```
AWSTemplateFormatVersion: 2010-09-09
Description: 'Amazon OpenSearch Serverless template to create an IAM user, encryption
 policy, data access policy and collection'
Resources:
  IAMUSer:
    Type: 'AWS::IAM::User'
    Properties:
      UserName:  aossadmin
  DataAccessPolicy:
    Type: 'AWS::OpenSearchServerless::AccessPolicy'
    Properties:
      Name: quickstart-access-policy
      Type: data
      Description: Access policy for quickstart collection
      Policy: !Sub >-
        [{"Description":"Access for cfn user","Rules":
[{"ResourceType":"index","Resource":["index/*/*"],"Permission":["aoss:*"]},
        {"ResourceType":"collection","Resource":["collection/quickstart"],"Permission":
["aoss:*"]}],
        "Principal":["arn:aws:iam::${AWS::AccountId}:user/aossadmin"]}]
  NetworkPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-network-policy
      Type: network
      Description: Network policy for quickstart collection
      Policy: >-
        [{"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}, {"ResourceType":"dashboard","Resource":["collection/
quickstart"]}],"AllowFromPublic":true}]
  EncryptionPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-security-policy
      Type: encryption
      Description: Encryption policy for quickstart collection
      Policy: >-
        {"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}],"AWSOwnedKey":true}
```

```
  Collection:
    Type: 'AWS::OpenSearchServerless::Collection'
    Properties:
      Name: quickstart
      Type: TIMESERIES
      Description: Collection to holds timeseries data
    DependsOn: EncryptionPolicy
Outputs:
  IAMUser:
    Value: !Ref IAMUSer
  DashboardURL:
    Value: !GetAtt Collection.DashboardEndpoint
  CollectionARN:
    Value: !GetAtt Collection.Arn
```

Amazon OpenSearch Serverless の容量制限の管理

Amazon OpenSearch Serverless では、キャパシティを自分で管理する必要はありません。OpenSearch Serverless は、現在のワークロードに基づいてアカウントのコンピューティングキャパシティを自動的にスケーリングします。サーバーレスコンピューティング容量はOpenSearch コンピューティングユニット (OCUs。各 OCU は、6 GiB のメモリと対応する仮想 CPU (vCPU)、および Amazon S3 へのデータ転送を組み合わせたものです。 OpenSearch Serverless のデカップリングアーキテクチャの詳細については、「」を参照してくださいthe section called “仕組み”。

最初のコレクションを作成すると、 OpenSearch Serverless は合計 4 つの OCUs をインスタンス化します (2 つはインデックス作成用、2 つは検索用) 。これらの OCU は、インデックス作成や検索が行われていない場合でも常に存在します。後続のすべてのコレクションは、これらの OCUs を共有できます (一意の AWS KMS キーを持つコレクションを除き、4 つの OCUs)。必要に応じて、OpenSearch Serverless は自動的にスケールアウトし、インデックス作成と検索の使用が増えるにつれて OCUs を追加します。コレクションエンドポイントのトラフィックが減少すると、キャパシティはデータサイズに必要な最小限の OCU 数まで縮小されます。最大でも、インデックス作成には 1 OCU [0.5 OCU x 2]、検索には 1 OCU [0.5 OCU x 2] までスケールダウンします。

検索およびベクトル検索コレクションでは、迅速なクエリ応答時間を確保するために、すべてのデータがホットインデックスに保存されます。時系列コレクションでは、ホットストレージとウォームストレージを組み合わせて使用し、より頻繁にアクセスされるデータのクエリ応答時間を最適化するために、最新のデータがホットストレージに保存されます。詳細については、「the section called “コレクションタイプを選択する”」を参照してください。

> **Note**
>
> ベクトル検索コレクションが検索コレクションまたは時系列コレクションと同じ KMS キーを使用している場合でも、ベクトル検索コレクションは OCUs を検索コレクションおよび時系列コレクションと共有することはできません。最初のベクトルコレクション用に新しい OCUsセットが作成されます。ベクトルコレクションの OCUs は、同じ KMS キーコレクション間で共有されます。

コレクションの容量を管理し、コストを制御するには、現在のアカウントとリージョンの全体的な最大インデックス作成と検索容量を指定できます。 OpenSearch サーバーレスは、これらの仕様に基づいてコレクションリソースを自動的にスケールアウトします。

インデックス作成と検索のキャパシティは個別にスケーリングされるため、それぞれにアカウントレベルの制限を指定します。

- 最大インデックス容量 – OpenSearch サーバーレスは、この数の OCUs。
- 最大検索容量 – OpenSearch サーバーレスは、この数の OCUs。

> **Note**
>
> 現時点では、キャパシティの設定はアカウントレベルでのみ適用されます。コレクションごとにキャパシティの制限を設定することはできません。

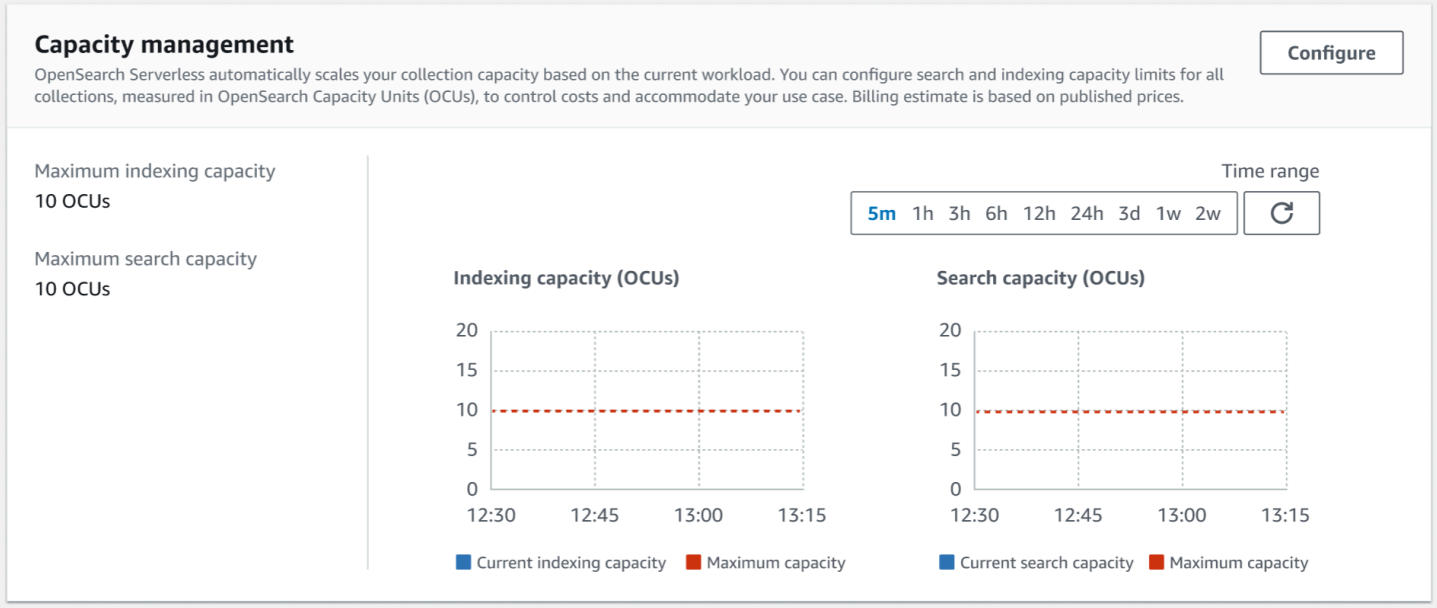
目標は、最大キャパシティがワークロードの急増を処理するために十分な量であるのを確実にすることです。設定に基づいて、 OpenSearch Serverless はコレクションの OCUs 数を自動的にスケールアウトして、インデックス作成と検索のワークロードを処理します。

トピック

- キャパシティの設定
- 最大キャパシティの制限
- キャパシティ使用量のモニタリング

キャパシティの設定

OpenSearch サーバーレスコンソールで容量設定を行うには、左側のナビゲーションペインでサーバーレスを展開し、ダッシュボード を選択します。[Capacity management] (キャパシティ管理) で、インデックス作成と検索の最大キャパシティを指定します。



を使用して容量を設定するには AWS CLI、 UpdateAccountSettingsリクエストを送信します。

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

最大キャパシティの制限

3 種類のコレクションすべてにおいて、デフォルトの最大キャパシティは、インデックス作成用に 10 OCU、検索用に 10 OCU です。アカウントの最小許容容量は、インデックス作成の場合は 1 OCU [0.5 OCU x 2]、検索の場合は 1 OCU [0.5 OCU x 2] です。すべてのコレクションで、許可される最大キャパシティは、インデックス作成用に 200 OCU、検索用に 200 OCU です。OCU カウントは、1 から最大許容容量までの任意の数に 2 の倍数で設定できます。

各 OCU には、120 GiB のインデックスデータ用に十分なホットエフェメラルストレージが含まれています。 OpenSearch Serverless は、検索コレクションとベクトル検索コレクションではインデックスごとに最大 1 TiB のデータをサポートし、時系列コレクションではインデックスごとに最大 10 TiB のホットデータをサポートします。時系列コレクションの場合は、さらに多くのデータを取り込むことができ、S3 にウォームデータとして保存できます。

すべてのクォータのリストについては、OpenSearch 「サーバーレスクォータ」を参照してください。

キャパシティ使用量のモニタリング

Search0CU およびアカウントIndexing0CUレベルの CloudWatch メトリクスをモニタリングして、コレクションのスケーリング方法を把握できます。アカウントがキャパシティに関するメトリクスのしきい値に近づいた際に通知を行うアラームを設定することをお勧めします。そうすることで、状況に応じてキャパシティの設定を調整できます。

最大キャパシティの設定が適切か、または調整が必要どうかを判断するために、これらのメトリクスを使用することもできます。これらのメトリクスを分析することで、コレクションの効率を最適化することに集中できます。 OpenSearch Serverless が に送信するメトリクスの詳細については、CloudWatch「」を参照してくださいthe section called “ OpenSearch サーバーレスのモニタリング”。

Amazon OpenSearch Serverless コレクションへのデータの取り込み

これらのセクションでは、Amazon OpenSearch Serverless コレクションへのデータインジェストでサポートされている取り込みパイプラインについて詳しく説明します。また、 OpenSearch API オペレーションを操作するために使用できるクライアントの一部についても説明します。Serverless と OpenSearch統合するには、クライアントが 2.x OpenSearch と互換性がある必要があります。

トピック

- 必要な最小限のアクセス許可
- OpenSearch 取り込み
- Fluent Bit
- Amazon Data Firehose
- Fluentd
- Go
- Java
- JavaScript
- Logstash
- Python

- Ruby
- 他のクライアントを使用した HTTP リクエストの署名

必要な最小限のアクセス許可

OpenSearch サーバーレスコレクションにデータを取り込むには、データを書き込むプリンシパルに、データアクセスポリシー で次の最小限のアクセス許可を割り当てる必要があります。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/target-collection/logs"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:WriteDocument",
               "aoss:UpdateIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

追加のインデックスに書き込む場合は、アクセス許可をより幅広く指定できます。例えば、1 つのターゲットインデックスを指定するのではなく、すべてのインデックス (index/*target-collection*/*) または一部のインデックス (index/*target-collection*/*logs**) へのアクセス許可を付与することができます。

使用可能なすべての OpenSearch API オペレーションとそれに関連するアクセス許可のリファレンスについては、「」を参照してくださいthe section called “サポートされているオペレーションとプラグイン”。

OpenSearch 取り込み

サードパーティーのクライアントを使用して OpenSearch サーバーレスコレクションに直接データを送信するのではなく、Amazon Ingestion OpenSearch を使用できます。データを Ingestion OpenSearch に送信するようにデータプロデューサーを設定すると、指定したコレクションにデータが自動的に配信されます。データを配信する前にデータを変換するように Ingestion OpenSearch を設定することもできます。詳細については、「Amazon OpenSearch Ingestion」を参照してください。

OpenSearch 取り込みパイプラインには、シンクとして設定された OpenSearch サーバーレスコレクションに書き込むためのアクセス許可が必要です。これらのアクセス権限には、コレクションを記述しそこに HTTP リクエストを送信できることが含まれます。Ingestion OpenSearch を使用してコレクションにデータを追加する手順については、「」を参照してくださいthe section called “パイプラインにコレクションへのアクセスを許可する”。

取り込みを開始するには、 OpenSearch 「」を参照してくださいthe section called “チュートリアル: コレクションにデータを取り込む”。

Fluent Bit

AWS for Fluent Bit イメージとOpenSearch 出力プラグインを使用して、 OpenSearch サーバーレスコレクションにデータを取り込むことができます。

> **Note**
>
> Serverless と統合するには、 AWS for Fluent Bit イメージのバージョン 2.30.0 OpenSearch 以降が必要です。

設定例:

設定ファイルのこのサンプル出力セクションでは、 OpenSearch サーバーレスコレクションを送信先として使用する方法を示します。重要な追加項目は `AWS_Service_Name` パラメータで、これは `aoss` です。`Host` はコレクションエンドポイントです。

```
[OUTPUT]
    Name  opensearch
    Match *
    Host  collection-endpoint.us-west-2.aoss.amazonaws.com
```

```
    Port  443
    Index  my_index
    Trace_Error On
    Trace_Output On
    AWS_Auth On
    AWS_Region <region>
    AWS_Service_Name aoss
    tls     On
    Suppress_Type_Name On
```

Amazon Data Firehose

Firehose は配信先として OpenSearch サーバーレスをサポートしています。 OpenSearch Serverless にデータを送信する手順については、「Amazon Data Firehose デベロッパーガイド」の「Kinesis Data Firehose 配信ストリームの作成」および「送信先に OpenSearch サーバーレスを選択する」を参照してください。

配信のために Firehose に提供する IAM ロールは、ターゲットコレクションに対するaoss:WriteDocument最小限のアクセス許可を持つデータアクセスポリシー内で指定する必要があります。また、データの送信先となる既存のインデックスが必要です。詳細については、「the section called “必要な最小限のアクセス許可”」を参照してください。

OpenSearch Serverless にデータを送信する前に、データに対して変換を実行する必要がある場合があります。Lambda 関数を使用してこのタスクを実行する方法については、このガイドの「Amazon Kinesis Data Firehose データ変換」を参照してください。

Fluentd

Fluentd OpenSearch プラグインを使用して、インフラストラクチャ、コンテナ、ネットワークデバイスからデータを収集し、Serverless OpenSearch コレクションに送信できます。Calyptia は、Ruby と SSL のすべてのダウンストリーム依存関係を含む Fluentd のディストリビューションを維持しています。

Fluentd を使用して OpenSearch Serverless にデータを送信するには

1. Calyptia Fluentd のバージョン 1.4.2 以降を https://www.fluentd.org/download からダウンロードします。このバージョンには、 OpenSearch サーバーレスをサポートする OpenSearch プラグインがデフォルトで含まれています。
2. パッケージをインストールします。ご使用のオペレーティングシステムに基づいて、次の Fluentd ドキュメントの指示に従ってください。

- Red Hat Enterprise Linux / CentOS / Amazon Linux
- Debian / Ubuntu
- Windows
- MacOSX

3. OpenSearch Serverless にデータを送信する設定を追加します。この設定例では、「test」というメッセージが 1 つのコレクションに送信されます。以下を実行するようにしてください。

   - にはhost、 OpenSearch サーバーレスコレクションのエンドポイントを指定します。
   - aws_service_name には、aoss を指定します。

```
<source>
@type sample
tag test
test {"hello":"world"}
</source>

<match test>
@type opensearch
host https://collection-endpoint.us-east-1.aoss.amazonaws.com
port 443
index_name fluentd
aws_service_name aoss
</match>
```

4. Calyptia Fluentd を実行して、コレクションへのデータの送信を開始します。例えば、Mac では次のコマンドを実行できます。

```
sudo launchctl load /Library/LaunchDaemons/calyptia-fluentd.plist
```

Go

次のサンプルコードでは、Go 用の opensearch-go クライアントを使用して、指定された OpenSearch Serverless コレクションへの安全な接続を確立し、単一のインデックスを作成します。region および host の値を指定する必要があります。

```
package main

import (
  "context"
  "log"
  "strings"
  "github.com/aws/aws-sdk-go-v2/aws"
  "github.com/aws/aws-sdk-go-v2/config"
  opensearch "github.com/opensearch-project/opensearch-go/v2"
  opensearchapi "github.com/opensearch-project/opensearch-go/v2/opensearchapi"
  requestsigner "github.com/opensearch-project/opensearch-go/v2/signer/awsv2"
)

const endpoint = "" // serverless collection endpoint

func main() {
 ctx := context.Background()

 awsCfg, err := config.LoadDefaultConfig(ctx,
  config.WithRegion("<AWS_REGION>"),
  config.WithCredentialsProvider(
   getCredentialProvider("<AWS_ACCESS_KEY>", "<AWS_SECRET_ACCESS_KEY>",
 "<AWS_SESSION_TOKEN>"),
  ),
 )
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an AWS request Signer and load AWS configuration using default config folder
 or env vars.
 signer, err := requestsigner.NewSignerWithService(awsCfg, "aoss") // "aoss" for Amazon
 OpenSearch Serverless
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an opensearch client and use the request-signer
 client, err := opensearch.NewClient(opensearch.Config{
  Addresses: []string{endpoint},
  Signer:    signer,
 })
 if err != nil {
```

```
  log.Fatal("client creation err", err)
 }

 indexName := "go-test-index"

  // define index mapping
 mapping := strings.NewReader(`{
  "settings": {
    "index": {
         "number_of_shards": 4
         }
       }
  }`)

 // create an index
 createIndex := opensearchapi.IndicesCreateRequest{
  Index: indexName,
    Body: mapping,
 }
 createIndexResponse, err := createIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("Error ", err.Error())
  log.Println("failed to create index ", err)
  log.Fatal("create response body read err", err)
 }
 log.Println(createIndexResponse)

 // delete the index
 deleteIndex := opensearchapi.IndicesDeleteRequest{
  Index: []string{indexName},
 }

 deleteIndexResponse, err := deleteIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("failed to delete index ", err)
  log.Fatal("delete index response body read err", err)
 }
 log.Println("deleting index", deleteIndexResponse)
}

func getCredentialProvider(accessKey, secretAccessKey, token string)
 aws.CredentialsProviderFunc {
 return func(ctx context.Context) (aws.Credentials, error) {
  c := &aws.Credentials{
```

```
    AccessKeyID:     accessKey,
    SecretAccessKey: secretAccessKey,
    SessionToken:    token,
   }
   return *c, nil
  }
}
```

Java

次のサンプルコードでは、Java 用の opensearch-java クライアントを使用して、指定された OpenSearch Serverless コレクションへの安全な接続を確立し、単一のインデックスを作成します。region および host の値を指定する必要があります。

OpenSearch サービスドメインとの重要な違いは、サービス名 (aoss ではなく) ですes。

```
// import OpenSearchClient to establish connection to OpenSearch Serverless collection
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();
// create an opensearch client and use the request-signer
OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

String index = "sample-index";

// create an index
CreateIndexRequest createIndexRequest = new
 CreateIndexRequest.Builder().index(index).build();
CreateIndexResponse createIndexResponse = client.indices().create(createIndexRequest);
System.out.println("Create index reponse: " + createIndexResponse);

// delete the index
DeleteIndexRequest deleteIndexRequest = new
 DeleteIndexRequest.Builder().index(index).build();
DeleteIndexResponse deleteIndexResponse = client.indices().delete(deleteIndexRequest);
```

```
System.out.println("Delete index reponse: " + deleteIndexResponse);

httpClient.close();
```

次のサンプルコードは、安全な接続を再度確立し、インデックスを検索します。

```
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();

OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

Response response = client.generic()
    .execute(
        Requests.builder()
            .endpoint("/" + "users" + "/_search?typed_keys=true")
            .method("GET")
            .json("{"
                + "    \"query\": {"
                + "        \"match_all\": {}"
                + "    }"
                + "}")
            .build());

httpClient.close();
```

JavaScript

次のサンプルコードでは、 の opensearch-js クライアント JavaScript を使用して、指定された OpenSearch Serverless コレクションへの安全な接続を確立し、単一のインデックスを作成し、ドキュメントを追加し、インデックスを削除します。node および `region` の値を指定する必要があります。

OpenSearch サービスドメインとの重要な違いは、サービス名 (`aoss` ではなく) ですes。

Version 3

この例では、Node.js の 用の SDK のバージョン 3 を使用しています。 JavaScript

```
const { defaultProvider } = require('@aws-sdk/credential-provider-node');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () => {
                const credentialsProvider = defaultProvider();
                return credentialsProvider();
            },
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({ index })).body);
    }

    // add a document to the index
    const document = { foo: 'bar' };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Version 2

この例では、Node.js の 用の SDK のバージョン 2 を使用しています。 JavaScript

```
const AWS = require('aws-sdk');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () =>
                new Promise((resolve, reject) => {
                    AWS.config.getCredentials((err, credentials) => {
                        if (err) {
                            reject(err);
                        } else {
                            resolve(credentials);
                        }
                    });
                }),
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index }),body) {
        console.log((await client.indices.create({
            index
        })).body);
    }

    // add a document to the index
    const document = {
        foo: 'bar'
    };
    const response = await client.index({
        id: '1',
        index: index,
```

```
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Logstash

Logstash OpenSearch プラグインを使用して、 OpenSearch Serverless コレクションにログを発行できます。

Logstash を使用して OpenSearch サーバーレスにデータを送信するには

1. Docker または Linux を使用して、プラグインのバージョン 2.0.0 以降をインストールします。logstash-output-opensearch

   Docker

   Docker は、 OpenSearch 出力プラグインがプリインストールされた Logstash OSS ソフトウェアをホストします: opensearchproject/logstash-oss-with-opensearch-output-plugin 。他のイメージと同じように次のようにイメージをプルできます。

   ```
   docker pull opensearchproject/logstash-oss-with-opensearch-output-plugin:latest
   ```

   Linux

   まだ Logstash の最新バージョンをインストールしていない場合は、インストールします。次に、出力プラグインのバージョン 2.0.0 を次のようにインストールします。

   ```
   cd logstash-8.5.0/
   bin/logstash-plugin install --version 2.0.0 logstash-output-opensearch
   ```

   プラグインが既にインストールされている場合は、次のように最新バージョンに更新します。

```
bin/logstash-plugin update logstash-output-opensearch
```

プラグインのバージョン 2.0.0 以降、 AWS SDK はバージョン 3 を使用します。8.4.0 より前のバージョンの Logstash を使用している場合は、プリインストールされている AWS プラグインをすべて削除し、logstash-integration-awsプラグインをインストールする必要があります。

```
/usr/share/logstash/bin/logstash-plugin remove logstash-input-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-input-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sns
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-cloudwatch

/usr/share/logstash/bin/logstash-plugin install --version 0.1.0.pre logstash-
integration-aws
```

2. OpenSearch 出力プラグインを OpenSearch Serverless と連携させるには、logstash.conf のopensearch出力セクションに次の変更を加える必要があります。

   - auth_type の service_name として aoss を指定します。
   - hosts のコレクションエンドポイントを指定します。
   - パラメータ default_server_major_version および legacy_template を追加します。これらのパラメータは、プラグインが OpenSearch Serverless で動作するために必要です。

```
output {
  opensearch {
    hosts => "collection-endpoint:443"
    auth_type => {
      ...
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

この設定ファイルの例は、S3 バケット内のファイルから入力を受け取り、 OpenSearch サーバーレスコレクションに送信します。

```
input {
  s3  {
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    ecs_compatibility => disabled
    hosts => "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com:443"
    index => my-index
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

3. 次に、次のように新しい設定で Logstash を実行してプラグインをテストします。

```
bin/logstash -f config/test-plugin.conf
```

Python

次のサンプルコードでは、Python 用の opensearch-py クライアントを使用して、指定された OpenSearch Serverless コレクションへの安全な接続を確立し、単一のインデックスを作成し、そのインデックスを検索します。region および host の値を指定する必要があります。

OpenSearch サービスドメインとの重要な違いは、サービス名 (aoss ではなく) ですes。

```
from opensearchpy import OpenSearch, RequestsHttpConnection, AWSV4SignerAuth
```

```
import boto3

host = ''  # serverless collection endpoint, without https://
region = ''  # e.g. us-east-1

service = 'aoss'
credentials = boto3.Session().get_credentials()
auth = AWSV4SignerAuth(credentials, region, service)

# create an opensearch client and use the request-signer
client = OpenSearch(
    hosts=[{'host': host, 'port': 443}],
    http_auth=auth,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection,
    pool_maxsize=20,
)

# create an index
index_name = 'books-index'
create_response = client.indices.create(
    index_name
)

print('\nCreating index:')
print(create_response)

# index a document
document = {
  'title': 'The Green Mile',
  'director': 'Stephen King',
  'year': '1996'
}

response = client.index(
    index = 'books-index',
    body = document,
    id = '1'
)


# delete the index
delete_response = client.indices.delete(
```

```
    index_name
)

print('\nDeleting index:')
print(delete_response)
```

Ruby

opensearch-aws-sigv4 gem は、 OpenSearch すぐにサーバーレスと OpenSearch サービスへのアクセスを提供します。これには、opensearch-ruby クライアントのすべての機能が備わっています。クライアントがこの gem の依存関係であるためです。

Sigv4 署名者をインスタンス化するときは、aoss をサービス名として指定します。

```
require 'opensearch-aws-sigv4'
require 'aws-sigv4'

signer = Aws::Sigv4::Signer.new(service: 'aoss',
                                region: 'us-west-2',
                                access_key_id: 'key_id',
                                secret_access_key: 'secret')

# create an opensearch client and use the request-signer
client = OpenSearch::Aws::Sigv4Client.new(
  { host: 'https://your.amz-opensearch-serverless.endpoint',
    log: true },
  signer)

# create an index
index = 'prime'
client.indices.create(index: index)

# insert data
client.index(index: index, id: '1', body: { name: 'Amazon Echo',
                                            msrp: '5999',
                                            year: 2011 })

# query the index
client.search(body: { query: { match: { name: 'Echo' } } })

# delete index entry
client.delete(index: index, id: '1')
```

```
# delete the index
client.indices.delete(index: index)
```

他のクライアントを使用した HTTP リクエストの署名

別のクライアントで HTTP リクエストを構築するときに、サーバーレスコレクションへのリクエストに署名する場合、次の要件が適用されます。 OpenSearch

- サービス名を aoss として指定する必要があります。
- x-amz-content-sha256 ヘッダーは、すべての AWS 署名バージョン 4 リクエストに必要です。これにより、リクエストペイロードのハッシュが指定されます。リクエストペイロードがある場合は、その値をセキュアハッシュアルゴリズム (SHA) 暗号化ハッシュ (SHA256) に設定します。リクエストペイロードがない場合は、値を e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855 に設定します。これは空の文字列のハッシュです。

トピック

- cURL を使用したインデックス作成
- Postman でのインデックス作成

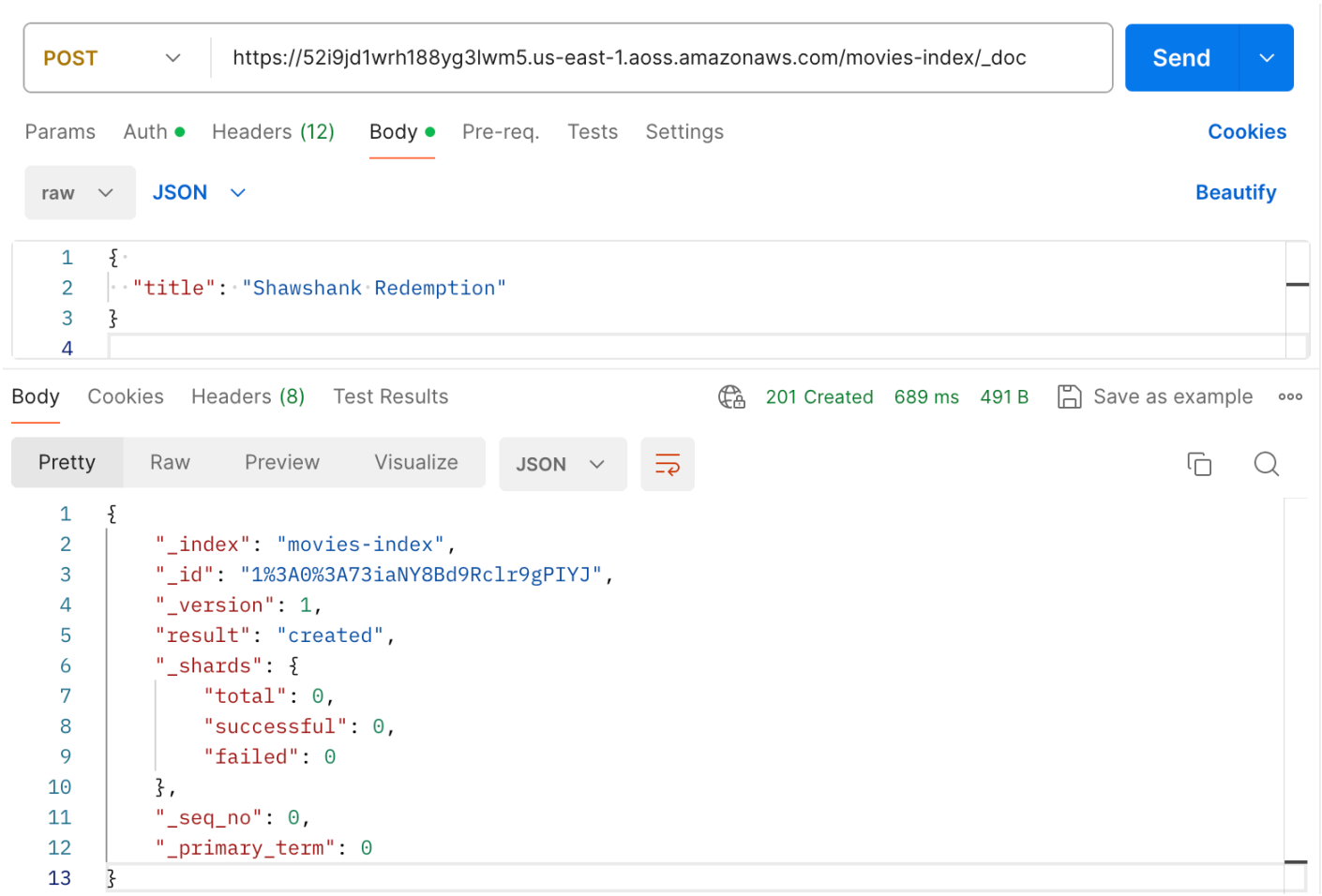
cURL を使用したインデックス作成

次のリクエスト例では、クライアント URL リクエストライブラリ (cURL) を使用して、コレクションmovies-index内の という名前のインデックスに 1 つのドキュメントを送信します。

```
curl -XPOST \
    --user "$AWS_ACCESS_KEY_ID":"$AWS_SECRET_ACCESS_KEY" \
    --aws-sigv4 "aws:amz:us-east-1:aoss" \
    --header "x-amz-content-sha256: $REQUEST_PAYLOAD_SHA_HASH" \
    --header "x-amz-security-token: $AWS_SESSION_TOKEN" \
    "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com/movies-index/_doc" \
    -H "Content-Type: application/json" -d '{"title": "Shawshank Redemption"}'
```

Postman でのインデックス作成

次の図は、Postman を使用してコレクションにリクエストを送信する方法を示しています。認証の手順については、Postman の AWS 「署名付き認証ワークフローによる認証」を参照してください。



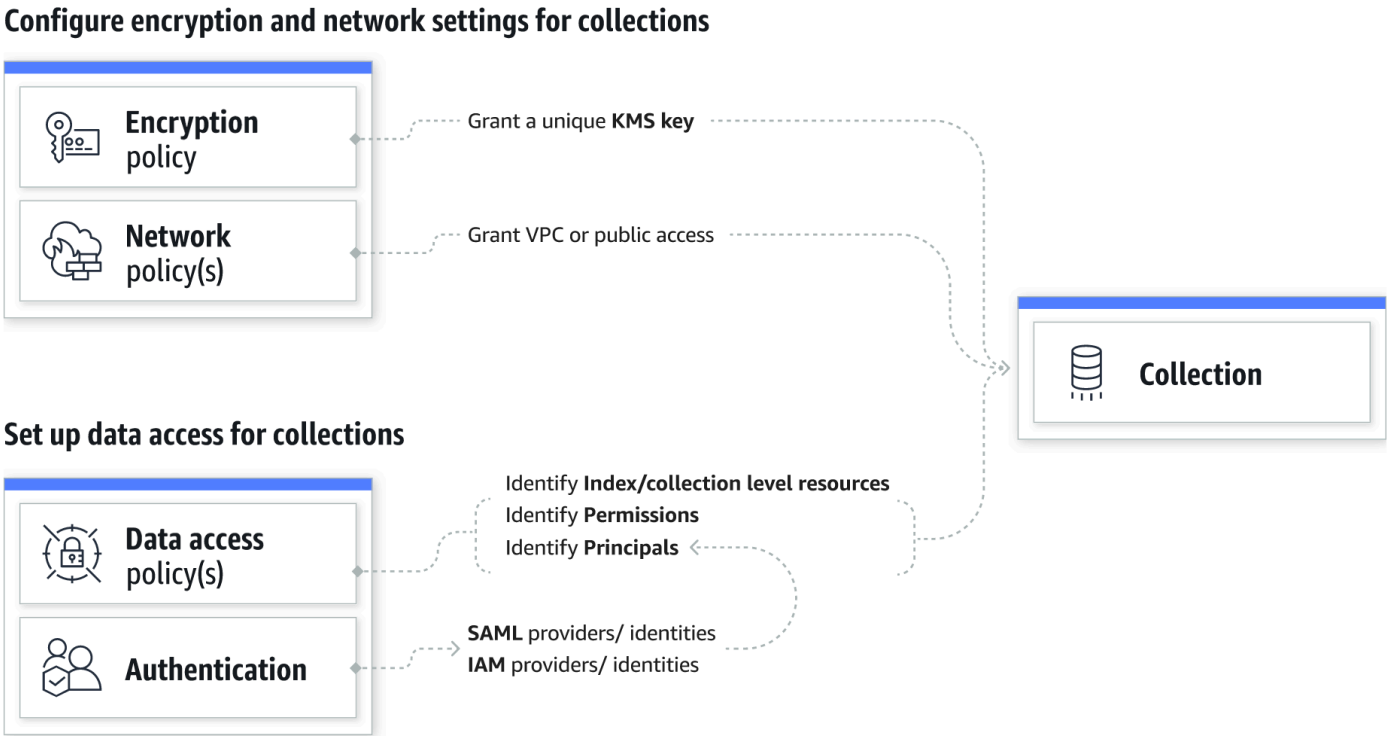
Amazon OpenSearch サーバーレスのセキュリティの概要

Amazon OpenSearch サーバーレスのセキュリティは、以下の点でAmazon OpenSearch Serviceのセキュリティと根本的に異なります。

機能	OpenSearch サービス	OpenSearch サーバーレス
データアクセスコントロール	きめ細かなアクセスコントロールと IAM ポリシーによりデータアクセスが決定します。	データアクセスポリシーによりデータアクセスが決定します。
保管時の暗号化	ドメインの保管時の暗号化はオプションです。	コレクションには保管時の暗号化が必要です。

機能	OpenSearch サービス	OpenSearch サーバーレス
セキュリティの設定および管理者	ネットワーク、暗号化、およびデータアクセスは、ドメインごとに個別に設定する必要があります。	セキュリティポリシーを使用して、複数のコレクションのセキュリティ設定を大規模に管理できます。

次の図は、機能的なコレクションを設定するセキュリティのコンポーネントを示しています。コレクションには、暗号化キー、ネットワークアクセス設定、リソースへのアクセス許可を付与するデータアクセスポリシーが割り当てられている必要があります。



トピック

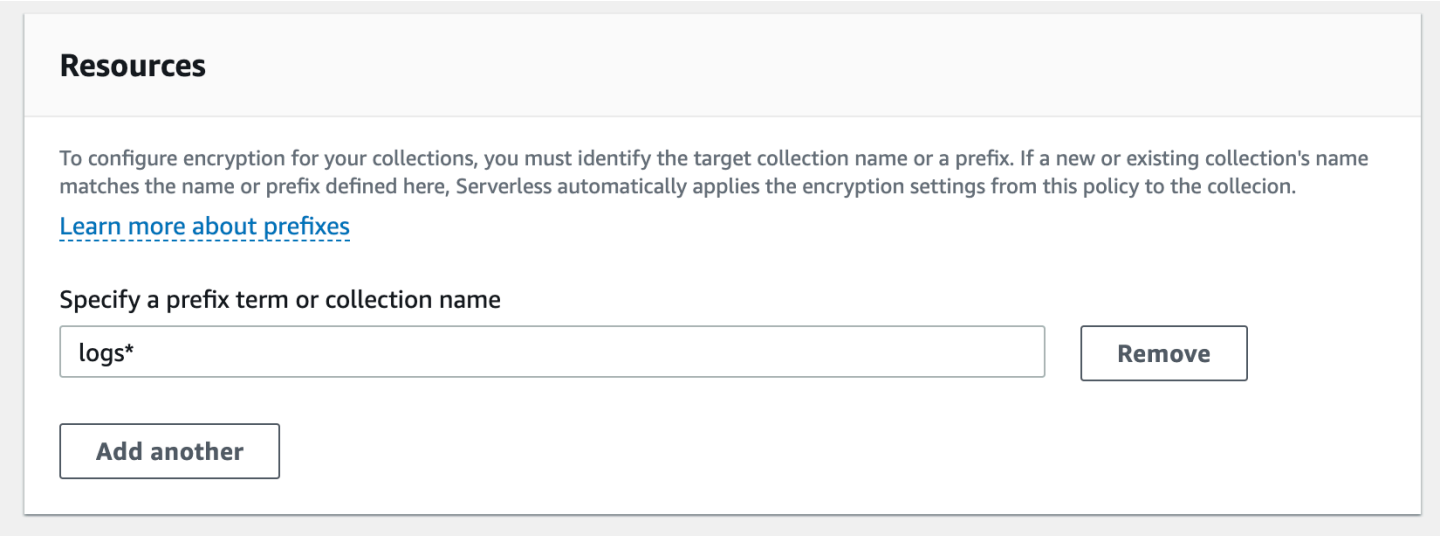
- 暗号化ポリシー
- ネットワークポリシー
- データアクセスポリシー
- IAM および SAML 認証
- インフラストラクチャセキュリティ
- Amazon OpenSearch Serverless でのセキュリティの開始方法

- Amazon OpenSearch Serverless 向けの アイデンティティとアクセス管理
- Amazon OpenSearch Serverless での暗号化
- Amazon OpenSearch Serverless のネットワークアクセス
- Amazon OpenSearch Serverless のデータアクセスコントロール
- インターフェイスエンドポイント (AWS PrivateLink) を使用して Amazon OpenSearch Serverless にアクセスする
- Amazon OpenSearch Serverless の SAML 認証
- Amazon OpenSearch Serverless のコンプライアンス検証

暗号化ポリシー

暗号化ポリシーは、 AWS 所有のキー コレクションを暗号化する鍵と顧客管理鍵のどちらで暗号化するかを定義します。暗号化ポリシーは、リソースパターンと暗号化キーの 2 つの要素で構成されます。リソースパターンは、ポリシーが適用される 1 つまたは複数のコレクションを定義します。暗号化キーは、関連するコレクションを保護する方法を決定します。

ポリシーを複数のコレクションに適用するには、ポリシーのルールにワイルドカード (*) を含めます。例えば、次のポリシーは、名前が「logs」で始まるすべてのコレクションに適用されます。

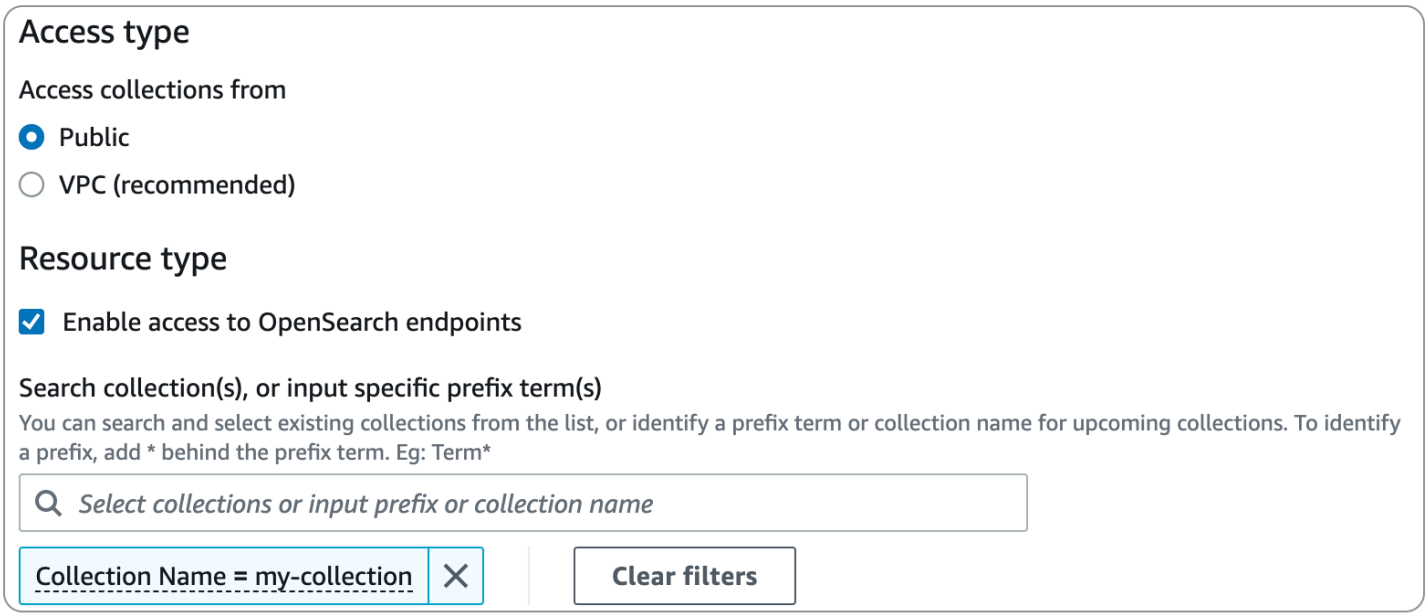


暗号化ポリシーは、特にプログラムで行う場合に、コレクションの作成と管理のプロセスを効率化します。コレクションは名前を指定するだけで作成でき、作成時に暗号化キーが自動的に割り当てられます。

ネットワークポリシー

ネットワークポリシーは、コレクションにプライベートにアクセスできるようにするか、パブリックネットワークからインターネット経由でアクセスするかを定義します。プライベートコレクションには、 OpenSearch サーバーレスで管理される VPC エンドポイントから、または Amazon Bedrock AWS のサービス などの特定のエンドポイントからプライベートアクセスを使用してアクセスできます。AWS のサービス ネットワークポリシーは、暗号化ポリシーと同様に複数のコレクションに適用できるため、多数のコレクションのネットワークアクセスを大規模に管理できます。

ネットワークポリシーは、アクセスタイプとリソースタイプの 2 つの要素で構成されます。アクセスタイプはパブリックでもプライベートでもかまいません。リソースタイプによって、選択したアクセスがコレクションエンドポイント、 OpenSearch ダッシュボードエンドポイント、またはその両方に適用されるかが決まります。



ネットワークポリシー内で VPC アクセスを設定する場合は、まず、OpenSearch サーバーレスで管理される VPC エンドポイントを 1 つ以上作成する必要があります。これらのエンドポイントを使用すると、インターネットゲートウェイ、NAT デバイス、VPN 接続、または接続を使用せずに、VPC OpenSearch 内にあるかのようにサーバーレスにアクセスできます。 AWS Direct Connect

AWS のサービス OpenSearch へのプライベートアクセスはコレクションのエンドポイントにのみ適用でき、Dashboards エンドポイントには適用されません。 OpenSearch AWS のサービス OpenSearch ダッシュボードへのアクセスは許可されません。

データアクセスポリシー

データアクセスポリシーは、ユーザーがコレクション内のデータにアクセスする方法を定義します。データアクセスポリシーは、特定のパターンに一致するコレクションとインデックスにアクセス許可を自動的に割り当てることにより、大規模なコレクションを管理するのに役立ちます。複数のポリシーを単一のリソースに適用できます。

データアクセスポリシーはルールのセットで構成され、それぞれに 3 つの構成要素 (リソースタイプ、付与されたリソース、およびアクセス許可のセット) があります。リソースタイプはコレクションでもインデックスでもかまいません。付与されたリソースは、コレクション名またはインデックス名、あるいはワイルドカード (*) 付きのパターンにすることができます。権限のリストでは、ポリシーがアクセスを許可する OpenSearch API オペレーションを指定します。さらにポリシーには、アクセスを許可する IAM ロール、ユーザー、SAML ID を指定するプリンシパルのリストが含まれています。

データアクセスポリシーの形式の詳細については、ポリシー構文を参照してください。

データアクセスポリシーを作成する前に、ポリシーでアクセスを提供するための、1 つ、または複数の IAM ロールもしくはユーザー、または SAML ID が必要です。詳細については、次のセクションを参照ください。

IAM および SAML 認証

IAM プリンシパルおよび SAML ID は、データアクセスポリシーの構成要素の 1 つです。アクセスポリシーの `principal` ステートメントには、IAM ロール、ユーザー、および SAML ID を含めることができます。その後、関連するポリシールールで指定したアクセス許可がこれらのプリンシパルに付与されます。

```
[
```

```
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie"
      ]
   }
]
```

SAML OpenSearch 認証はサーバーレス内で直接設定します。詳細については、「the section called "SAML 認証"」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

Amazon OpenSearch Serverless AWS はグローバルネットワークセキュリティによって保護されています。 AWS AWS セキュリティサービスとインフラストラクチャの保護方法については、「AWS クラウドセキュリティ」を参照してください。 AWS 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環境を設計するには、「Security Pillar AWS Well-Architected Framework におけるインフラストラクチャ保護」を参照してください。

AWS 公開されている API 呼び出しを使用して、ネットワーク経由で Amazon OpenSearch サーバーレスにアクセスします。クライアントは Transport Layer Security (TLS) をサポートしている必要があります。TLS 1.2、できれば TLS 1.3 が必要です。TLS 1.3 でサポートされる暗号のリストについては、Elastic Load Balancing ドキュメントの「TLS プロトコルと暗号」を参照してください。

さらに、IAM プリンシパルに関連付けられたアクセスキー ID とシークレットアクセスキーを使用してリクエストに署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時セキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon OpenSearch Serverless でのセキュリティの開始方法

以下のチュートリアルは、Amazon OpenSearch Serverless の使用を開始するのに役立ちます。どちらのチュートリアルでも基本的な手順は同じですが、一方はコンソールを使用し、もう一方は AWS CLI を使用します。

これらのチュートリアルのユースケースは簡素化されていることに注意してください。ネットワークおよびセキュリティポリシーは、かなりオープンです。本番環境のワークロードでは、SAML 認証、VPC アクセス、制限付きのデータアクセスポリシーなど、より強固なセキュリティ機能を設定することをお勧めします。

トピック

- チュートリアル: Amazon OpenSearch Serverless でのセキュリティの開始方法 (コンソール)
- チュートリアル: Amazon OpenSearch Serverless でのセキュリティの開始方法 (CLI)

チュートリアル: Amazon OpenSearch Serverless でのセキュリティの開始方法 (コンソール)

このチュートリアルでは、Amazon OpenSearch Serverless コンソールを使用してセキュリティポリシーを作成および管理するための基本的な手順について説明します。

このチュートリアルでは、次の手順を行います。

1. アクセス許可を設定
2. 暗号化ポリシーを作成
3. ネットワークポリシーを作成する
4. データアクセスポリシーを設定する
5. コレクションを作成
6. データをアップロードおよび検索する

このチュートリアルでは、AWS Management Console を使用してコレクションを設定する手順を説明します。AWS CLI を使用する同様の手順については、「the section called “チュートリアル: セキュリティの開始方法 (CLI)”」を参照してください。

ステップ 1: アクセス許可を設定する

> **Note**
>
> `Action":"aoss:*"` や `Action":"*"` など、より広範な ID ベースのポリシーを既に使用している場合は、この手順をスキップできます。ただし本番環境では、最小特権の原則に従い、作業を完了するのに最低限必要なアクセス許可を割り当てることをお勧めします。

このチュートリアルを完了するためには、適切な IAM のアクセス許可を持っている必要があります。ユーザーまたはロールには、以下の最低限の許可を含む ID ベースのポリシーが、アタッチされている必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:CreateCollection",
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:ListSecurityPolicies",
        "aoss:CreateAccessPolicy",
        "aoss:GetAccessPolicy",
        "aoss:ListAccessPolicies"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

OpenSearch サーバーレスアクセス許可の完全なリストについては、「」を参照してくださいthe section called “ID とアクセス管理”。

ステップ 2: 暗号化ポリシーを作成する

暗号化ポリシーは、 OpenSearch Serverless がコレクションの暗号化に使用するAWS KMSキーを指定します。AWS マネージドキー キーまたは別のキーを使用して、コレクションを暗号化できます。

このチュートリアルでは、わかりやすくするため AWS マネージドキー でコレクションを暗号化します。

暗号化ポリシーを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインで [Serverless] を展開し、[Encryption policies] (暗号化ポリシー) を選択します。
3. [Create encryption policy] (暗号化ポリシーを作成) を選択します。
4. ポリシーに「books-policy」と名前を付けます。説明には、「Encryption policy for books collection」(books コレクションの暗号化ポリシー) と入力します。
5. [Resources] (リソース) に「books」と入力します。これがコレクションの名前になります。より広範囲にしたい場合は、アスタリスク (books*) を追加します。これにより「books」という単語で始まるすべてのコレクションにポリシーを適用できます。
6. [暗号化] については、[AWS 所有キーを使用] を選択したままにしておきます。
7. [作成] を選択します。

ステップ 3: ネットワークポリシーを作成する

ネットワークポリシーは、コレクションにパブリックネットワークからインターネット経由でアクセス可能にするか、 OpenSearch サーバーレスが管理する VPC エンドポイント経由でアクセスする必要があるかを決定します。このチュートリアルでは、パブリックアクセスを設定します。

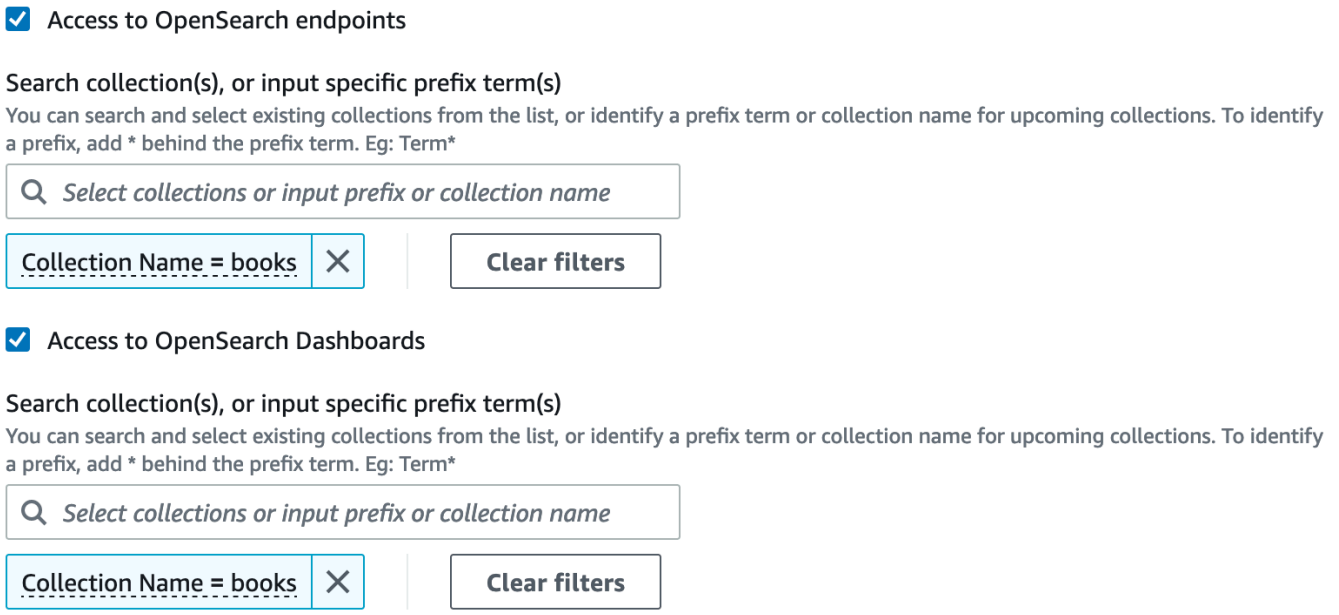
ネットワークポリシーを作成するには

1. 左のナビゲーションペインで [Network policies] (ネットワークポリシー) を選択し、[Create network policy] (ネットワークポリシーの作成) を選択します。
2. ポリシーに「books-policy」と名前を付けます。説明には、「Network policy for books collection」(books コレクションのネットワークポリシー) と入力します。
3. [Rule 1] (ルール 1) で、ルールに「Public access for books collection」(books コレクションのパブリックアクセス) という名前を付けます。
4. このチュートリアルでは、わかりやすくするため books コレクションにパブリックアクセスを設定します。アクセスタイプには、[Public] (パブリック) を選択します。

5. OpenSearch Dashboards からコレクションにアクセスします。そのためには、Dashboards と OpenSearch エンドポイントのネットワークアクセスを設定する必要があります。そうしないと、Dashboards は機能しません。

   リソースタイプで、エンドポイントへのアクセス OpenSearchと Dashboards へのアクセス OpenSearchの両方を有効にします。

6. 両方の入力ボックスに、「Collection Name = books」(コレクション名 = books) と入力します。このように設定すると、ポリシーの範囲が狭くなり 1 つのコレクション (books) にのみ適用されるようになります。ルールは次のようになります。



7. [作成] を選択します。

ステップ 4: データアクセスポリシーを作成する

コレクションのデータには、データアクセスを設定するまでアクセスできません。データアクセスポリシーは、ステップ 1 で設定した IAM ID ベースのポリシーとは別のものです。このポリシーでは、コレクション内の実際のデータにアクセスできます。

このチュートリアルでは、books コレクションへのデータのインデックス化に必要なアクセス許可を 1 人のユーザーに提供します。

データアクセスポリシーを作成するには

1. 左のナビゲーションペインで、[Data access policies] (データアクセスポリシー) を選択し、[Create access policy] (アクセスポリシーを作成) を選択します。

2. ポリシーに「books-policy」と名前を付けます。説明には、「Data access policy for books collection」(books コレクションのデータアクセスポリシー) と入力します。
3. ポリシーの定義方法として [JSON] を選択し、JSON エディターに次のポリシーを貼り付けます。

   プリンシパル ARN を、 OpenSearch Dashboards へのログインとデータのインデックス作成に使用するアカウントの ARN に置き換えます。

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/books/*"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:DescribeIndex",
               "aoss:ReadDocument",
               "aoss:WriteDocument",
               "aoss:UpdateIndex",
               "aoss:DeleteIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

   このポリシーは、books コレクション内のインデックス作成や一部のデータのインデックス化、および検索に必要な最低限のアクセス許可を 1 人のユーザーに提供します。
4. [作成] を選択します。

ステップ 5: コレクションを作成する

これで、暗号化ポリシーとネットワークポリシーを設定できました。一致するコレクションを作成すると、セキュリティ設定が自動的に適用されるようになります。

OpenSearch Serverless コレクションを作成するには

1. 左側のナビゲーションペインで [Collections] (コレクション) 、[Create collection] (コレクションを作成) を選択します。
2. コレクションに「books」と名前を付けます。
3. コレクションタイプでは、[Search] (検索) を選択します。
4. 暗号化 で、 OpenSearch Serverless はコレクション名がbooks-policy暗号化ポリシーと一致することを通知します。
5. ネットワークアクセス設定 で、 OpenSearch Serverless はコレクション名がbooks-policyネットワークポリシーと一致することを通知します。
6. [次へ] をクリックします。
7. データアクセスポリシーオプションで、 OpenSearch Serverless はコレクション名がbooks-policyデータアクセスポリシーと一致することを通知します。
8. [次へ] をクリックします。
9. コレクションの設定を確認し、[Submit] (送信) を選択します。通常、コレクションの初期化には 1 分もかかりません。

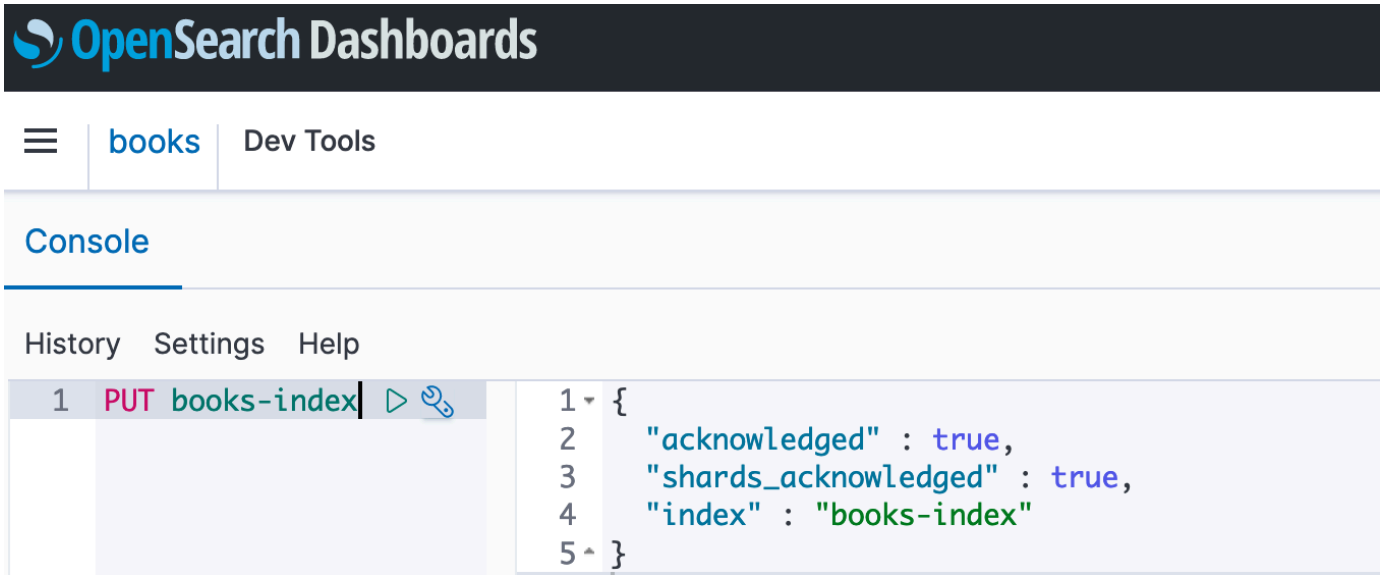
ステップ 6: データをアップロードおよび検索する

Postman または curl を使用して、 OpenSearch サーバーレスコレクションにデータをアップロードできます。簡潔にするために、これらの例では OpenSearch Dashboards コンソール内で開発ツールを使用しています。

コレクション内のデータをインデックス化して検索するには

1. 左側のナビゲーションペインで [Collections] (コレクション) を選択し、books コレクションを選択して詳細ページを開きます。
2. コレクションの OpenSearch Dashboards URL を選択します。この URL の形式は、`https://collection-id.us-east-1.aoss.amazonaws.com/_dashboards` になります。
3. データAWSアクセスポリシーで指定したプリンシパルのアクセスキーとシークレットキーを使用して OpenSearch Dashboards にサインインします。
4. OpenSearch Dashboards で、左側のナビゲーションメニューを開き、開発ツール を選択します。
5. books-index という単一のインデックスを作成するには、次のコマンドを実行します。

```
PUT books-index
```



6. 単一のドキュメントを books-index にインデックス化するには、次のコマンドを実行します。

```
PUT books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

7. OpenSearch Dashboards でデータを検索するには、少なくとも 1 つのインデックス pattern. OpenSearch uses これらのパターンを設定して、分析するインデックスを特定する必要があります。Dashboards のメインメニューを開き、[スタック管理] を選択し、[インデックスパターン] を選択してから、[インデックスパターンを作成する] を選択します。このチュートリアルでは、「books-index」と入力します。
8. [次のステップ] を選択してから、[インデックスパターンの作成] を選択します。パターンが作成されたら、`author` および `title` などのさまざまなドキュメントフィールドを表示できます。
9. データの検索を開始するには、メインメニューをもう一度開き [Discover] (検出) を選択するか、検索 API を使用します。

チュートリアル: Amazon OpenSearch Serverless でのセキュリティの開始方法 (CLI)

このチュートリアルでは、セキュリティに関するコンソール入門チュートリアルで説明されている手順を説明しますが、 は OpenSearch サービスコンソールAWS CLIではなく を使用します。

このチュートリアルでは、次の手順を実行します。

1. IAM アクセス権限ポリシーを作成する
2. IAM ポリシーを IAM ロールにアタッチする
3. 暗号化ポリシーを作成する
4. ネットワークポリシーを作成する
5. コレクションを作成
6. データアクセスポリシーを設定する
7. コレクションエンドポイントを取得する
8. 接続にデータをアップロードする
9. コレクション内のデータを検索する

このチュートリアルの目的は、非常にシンプルな暗号化、ネットワーク、およびデータアクセス設定で単一の OpenSearch サーバーレスコレクションを設定することです。例えば、暗号化用の AWS マネージドキー、パブリックネットワークアクセス、1 人のユーザーに最低限のアクセス許可を付与する簡単なデータアクセスポリシーを設定します。

本番環境では、SAML 認証、カスタム暗号化キー、VPC アクセスなどのより強固な設定を実装することを検討してください。

OpenSearch Serverless でセキュリティポリシーの使用を開始するには

1.

> **Note**
>
> `Action":"aoss:*"` や `Action":"*"` など、より広範な ID ベースのポリシーを既に使用している場合は、この手順をスキップできます。ただし本番環境では、最小特権の原則に従い、作業を完了するのに最低限必要なアクセス許可を割り当てることをお勧めします。

まず、このチュートリアルの手順を実行するために最低限必要なアクセス許可を持つ AWS Identity and Access Management ポリシーを作成します。TutorialPolicy ポリシーには次のような名前を付けます。

```
aws iam create-policy \
  --policy-name TutorialPolicy \
  --policy-document "{\"Version\": \"2012-10-17\",\"Statement\":
 [{\"Action\": [\"aoss:ListCollections\",\"aoss:BatchGetCollection\",
\"aoss:CreateCollection\",\"aoss:CreateSecurityPolicy\",\"aoss:GetSecurityPolicy\",
\"aoss:ListSecurityPolicies\",\"aoss:CreateAccessPolicy\",\"aoss:GetAccessPolicy\",
\"aoss:ListAccessPolicies\"],\"Effect\": \"Allow\",\"Resource\": \"*\"}]}"
```

レスポンス例

```
{
    "Policy": {
        "PolicyName": "TutorialPolicy",
        "PolicyId": "ANPAW6WRAECKG6QJWUV7U",
        "Arn": "arn:aws:iam::123456789012:policy/TutorialPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2022-10-16T20:57:18+00:00",
        "UpdateDate": "2022-10-16T20:57:18+00:00"
    }
}
```

2. コレクション内でデータをインデックス化して検索する IAM ロールに TutorialPolicy をアタッチします。TutorialRole ユーザーに次のような名前を付けます。

```
aws iam attach-role-policy \
  --role-name TutorialRole \
  --policy-arn arn:aws:iam::123456789012:policy/TutorialPolicy
```

3. コレクションを作成する前に、後の手順で作成する books コレクションに AWS 所有のキー を割り当てる暗号化ポリシーを作成する必要があります。

   books コレクション用の暗号化ポリシーを作成するには、次のリクエストを送信します。

```
aws opensearchserverless create-security-policy \
  --name books-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/books\"]}],\"AWSOwnedKey\":true}"
```

レスポンス例

```
{
    "securityPolicyDetail": {
        "type": "encryption",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDAwNTk5MF8x",
        "policy": {
            "Rules": [
                {
                    "Resource": [
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
            "AWSOwnedKey": true
        },
        "createdDate": 1669240005990,
        "lastModifiedDate": 1669240005990
    }
}
```

4. books コレクションにパブリックアクセスを提供するネットワークポリシーを作成します。

```
aws opensearchserverless create-security-policy --name books-policy --type network
 \
  --policy "[{\"Description\":\"Public access for books collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/books\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/books\"]}],
\"AllowFromPublic\":true}]"
```

レスポンス例

```
{
    "securityPolicyDetail": {
```

```
        "type": "network",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDI1Njk1NV8x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
                            "collection/books"
                        ],
                        "ResourceType": "dashboard"
                    },
                    {
                        "Resource": [
                            "collection/books"
                        ],
                        "ResourceType": "collection"
                    }
                ],
                "AllowFromPublic": true,
                "Description": "Public access for books collection"
            }
        ],
        "createdDate": 1669240256955,
        "lastModifiedDate": 1669240256955
    }
}
```

5. books コレクションを作成します。

```
aws opensearchserverless create-collection --name books --type SEARCH
```

レスポンス例

```
{
    "createCollectionDetail": {
        "id": "8kw362bpwg4gx9b2f6e0",
        "name": "books",
        "status": "CREATING",
        "type": "SEARCH",
        "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
        "kmsKeyArn": "auto",
```

```
        "createdDate": 1669240325037,
        "lastModifiedDate": 1669240325037
    }
}
```

6. books コレクション内のデータをインデックス化して検索するための最小限のアクセス許可を付与するデータアクセスポリシーを作成します。プリンシパル ARN をステップ 1 の TutorialRole の ARN に置き換えます。

```
aws opensearchserverless create-access-policy \
  --name books-policy \
  --type data \
  --policy "[{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":
[\"index\/books\/books-index\"],\"Permission\":[\"aoss:CreateIndex
\",\"aoss:DescribeIndex\",\"aoss:ReadDocument\",\"aoss:WriteDocument
\",\"aoss:UpdateIndex\",\"aoss:DeleteIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:role\/TutorialRole\"]}]"
```

レスポンス例

```
{
    "accessPolicyDetail": {
        "type": "data",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDM5NDY1M18x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
                            "index/books/books-index"
                        ],
                        "Permission": [
                            "aoss:CreateIndex",
                            "aoss:DescribeIndex",
                            "aoss:ReadDocument",
                            "aoss:WriteDocument",
                            "aoss:UpdateDocument",
                            "aoss:DeleteDocument"
                        ],
                        "ResourceType": "index"
                    }
```

```
                ],
                "Principal": [
                    "arn:aws:iam::123456789012:role/TutorialRole"
                ]
            }
        ],
        "createdDate": 1669240394653,
        "lastModifiedDate": 1669240394653
    }
}
```

これで、TutorialRole は books コレクション内のドキュメントをインデックス化して検索できるはずです。

7. OpenSearch API を呼び出すには、コレクションエンドポイントが必要です。次のリクエストを送信して、collectionEndpoint パラメータを取得します。

```
aws opensearchserverless batch-get-collection --names books
```

レスポンス例

```
{
    "collectionDetails": [
        {
            "id": "8kw362bpwg4gx9b2f6e0",
            "name": "books",
            "status": "ACTIVE",
            "type": "SEARCH",
            "description": "",
            "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
            "createdDate": 1665765327107,
            "collectionEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com",
            "dashboardEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com/_dashboards"
        }
    ],
    "collectionErrorDetails": []
}
```

> **Note**
>
> コレクションのステータスが ACTIVE に変わるまで、コレクションエンドポイントを表示することはできません。コレクションが正常に作成されるまで、呼び出しを複数回行いステータスを確認する必要があることがあります。

8. Postman や curl などの HTTP ツールを使用して、データを books コレクションにインデックス化します。books-index というインデックスを作成し、単一のドキュメントを追加します。

   TutorialRole の認証情報を使用して、次のリクエストを前の手順で取得したコレクションエンドポイントに送信します。

```
PUT https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

   レスポンス例

```
{
  "_index" : "books-index",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 0
}
```

9. コレクション内でデータの検索を開始するには、検索 API を使用します。次のクエリは、基本的な検索を実行します。

```
GET https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_search
```

レスポンス例


```
{
    "took": 405,
    "timed_out": false,
    "_shards": {
        "total": 6,
        "successful": 6,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 2,
            "relation": "eq"
        },
        "max_score": 1.0,
        "hits": [
            {
                "_index": "books-index:0::3xJq14MBUaOS0wL26UU9:0",
                "_id": "F_bt4oMBLle5pYmm5q4T",
                "_score": 1.0,
                "_source": {
                    "title": "The Shining",
                    "author": "Stephen King",
                    "year": 1977
                }
            }
        ]
    }
}
```


Amazon OpenSearch Serverless 向けの アイデンティティとアクセス管理

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するために役立つ AWS のサービスです。IAM 管理者は、どのユーザーを認証 (サインイン受け入れ) し、また OpenSearch Serverless リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加費用なしで使用できる AWS のサービスです。

トピック

- OpenSearch Serverless でのアイデンティティベースのポリシー
- OpenSearch Serverless でのポリシーアクション
- OpenSearch Serverless のポリシーリソース
- Amazon OpenSearch Serverless のポリシー条件キー
- ABAC と OpenSearch Serverless
- OpenSearch Serverless での一時的な認証情報の使用
- Amazon OpenSearch Serverless でのサービスにリンクされたロール
- OpenSearch Serverless での ID ベースのポリシー例

OpenSearch Serverless で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	あり

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、および条件を制御します。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、それがアタッチ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。

OpenSearch Serverless での ID ベースのポリシー例

OpenSearch Serverless での ID ベースのポリシー例については、「the section called “アイデンティティベースポリシーの例”」を参照してください。

OpenSearch Serverless で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

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

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

OpenSearch Serverless 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を使用します。

```
aoss
```

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": [
      "aoss:action1",
      "aoss:action2"
         ]
```

ワイルドカード文字 (*) を使用すると、複数のアクションを指定することができます。例えば、Describe という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "aoss:List*"
```

OpenSearch Serverless での ID ベースのポリシー例については、「OpenSearch Serverless での ID ベースのポリシー例」を参照してください。

OpenSearch Serverless 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	あり

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 Resource 要素は、アクションが適用される 1 つ以上の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、リソースレベルの権限と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

Amazon OpenSearch Serverless 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	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。

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になる条件を指定できます。Condition 要素はオプションです。equal や less than などの条件演算子を使用して条件式を作成することによって、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定するか、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を評価します。単一の条件キーに複数の値を指定すると、AWS は OR 論理演算子を使用して条件を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

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

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

OpenSearch Serverless では、属性ベースのアクセス制御 (ABAC) に加えて、以下の条件キーをサポートしています。

- aoss:collection
- aoss:CollectionId

- aoss:index

これらの条件キーは、アクセスポリシーおよびセキュリティポリシーに許可を与える場合でも使用できます。例:

```
[
   {
      "Effect":"Allow",
      "Action":[
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringLike":{
            "aoss:collection":"log"
         }
      }
   }
]
```

この例の条件は、コレクション名またはパターンに一致するルールを含むポリシーに適用されます。コンディションは、以下の動作を行います。

- StringEquals – リソース文字列として「log」が正確に配置された (つまり collection/log) ルールを使用する、ポリシーに適用されます。
- StringLike –「log」という文字列が含まれたリソース文字列で構成されたルール (例: collection/log に加え collection/logs-application または collection/applogs123) を使用する、ポリシーに対し適用されます。

> **Note**
>
> コレクション条件キーは、インデックスレベルでは適用されません。例として上記のポリシーでは、リソース文字列 index/logs-application/* を含むアクセスポリシーまたはセキュリティポリシーに対して、この条件は適用されません。

OpenSearch Serverless の条件キーのリストは、「サービス認証リファレンス」の「Condition keys for Amazon OpenSearch Serverless」(Amazon OpenSearch Serverless での条件キー) でご確認くだ

さい。条件キーの使用が可能なアクションおよびリソースついては、「Actions defined by Amazon OpenSearch Serverless」(Amazon OpenSearch Serverless で定義されるアクション) を参照してください。

ABAC と OpenSearch Serverless

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。AWS では、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)、および多数の 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操作を許可するように ABAC ポリシーを設計します。

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 Condition 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 つの条件キーのすべてをサポートする場合、そのサービスでのサポート状況の値は「はい」になります。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「属性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

OpenSearch Serverless リソースでのタグ付けの詳細については、「the section called “タグコレクション”」を参照してください。

OpenSearch Serverless での一時的な認証情報の使用

一時的な認証情報のサポート	あり

AWS のサービス には、一時的な認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報で機能する AWS のサービスなどの詳細については、「IAM ユーザーガイド」の「IAM と連携する AWS のサービス」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時的な認証情報を使用していることになります。例えば、会社の Single Sign-On (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時認証情報を作成します。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。作成後、一時的な認証情報を使用して AWS にアクセスできるようになります。AWS 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お勧めします。詳細については、「IAM の一時的なセキュリティ認証情報」を参照してください。

Amazon OpenSearch Serverless で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

サービスリンクロールは、AWS のサービス にリンクされているサービスロールの一種です。サービスがロールを引き受け、ユーザーに代わってアクションを実行できるようになります。サービスリンクロールは、AWS アカウント に表示され、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権限を表示できますが、編集することはできません。

OpenSearch Serverless でのサービスにリンクされたロールの作成または管理の詳細については、「the section called “コレクション作成ロール”」を参照してください。

OpenSearch Serverless での ID ベースのポリシー例

デフォルトでは、OpenSearch Serverless リソースを作成または変更する許可は、ユーザーおよびロールに付与されていません。また、AWS Management Console、AWS Command Line Interface (AWS CLI)、または AWS API を使用してタスクを実行することもできません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者がロールに IAM ポリシーを追加すると、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 OpenSearch Serverless が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式など) の詳細については、「サービス認証リファレンス」の「Actions, resources, and

condition keys for Amazon OpenSearch Serverless」(Amazon OpenSearch Serverless でのアクション、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
- コンソールでの OpenSearch Serverless の使用
- OpenSearch Serverless コレクションの管理
- OpenSearch Serverless コレクションの表示
- OpenSearch API オペレーションの使用

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ポリシーは非常に強力です。このポリシーは、アカウント内の OpenSearch Serverless リソースを作成、アクセス、または削除することが可能なユーザーを決定します。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:

アイデンティティベースのポリシーは、アカウント内で誰かが OpenSearch Serverless リソースの作成、アクセス、または削除を実行できるどうかを決定します。これらのアクションを実行すると、AWS アカウント に料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS マネージドポリシーを使用して開始し、最小特権の権限に移行する – ユーザーとワークロードへの権限の付与を開始するには、多くの一般的なユースケースのために権限を付与する AWS マネージドポリシーを使用します。これらは AWS アカウントで使用できます。ユースケースに応じた AWS カスタマーマネージドポリシーを定義することで、権限をさらに減らすことをお勧めします。詳細については、『IAM ユーザーガイド』の「AWS マネージドポリシー」または「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権権限とも呼ばれています。IAM を使用して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 でのポリシーと権限」を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。また、AWS CloudFormation などの特定の

AWS のサービス を介して使用する場合、条件を使用してサービスアクションへのアクセスを許可することもできます。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 - AWS アカウント内の IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。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。

コンソールでの OpenSearch Serverless の使用

OpenSearch Service コンソールで OpenSearch Serverless にアクセスするには、最小限の許可セットが必要です。このセットでは、AWS アカウントにある OpenSearch Serverless リソースについて、一覧と詳細の表示を許可する必要があります。必要最低限の許可よりも制限が厳しいアイデンティティベース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 (IAM ロールなど) に対してコンソールが意図したとおりに機能しなくなります。

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が許可されます。

以下のポリシーでは、ユーザーが OpenSearch Service コンソール内で OpenSearch Serverless にアクセスすることを許可しています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
```

```
                "aoss:ListCollections",
                "aoss:BatchGetCollection",
                "aoss:ListAccessPolicies",
                "aoss:ListSecurityConfigs",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:ListVpcEndpoints",
                "aoss:GetAccessPolicy",
                "aoss:GetAccountSettings",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy"
            ]
        }
    ]
}
```

OpenSearch Serverless コレクションの管理

このポリシーは、ユーザーに対し Amazon OpenSearch Serverless コレクションの処理および管理を許可する、「コレクション管理者」ポリシーの一例です。これによりユーザーは、コレクションを作成、表示、削除できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:aoss:region:123456789012:collection/*",
            "Action": [
                "aoss:CreateCollection",
                "aoss:DeleteCollection",
                "aoss:UpdateCollection"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "aoss:CreateAccessPolicy",
                "aoss:CreateSecurityPolicy"
            ],
            "Effect": "Allow"
```

```
        }
    ]
}
```

OpenSearch Serverless コレクションの表示

このポリシー例では、ユーザーに対し、自身のアカウント内のすべての Amazon OpenSearch Serverless コレクションについて、その詳細を表示することを許可します。このユーザーには、コレクション自体や関連するセキュリティポリシーの変更は許可されません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:ListCollections",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow"
        }
    ]
}
```

OpenSearch API オペレーションの使用

データプレーン API オペレーションは、OpenSearch Serverless でサービスからリアルタイムの値を引き出すために使用する関数で構成されています。コントロールプレーン API オペレーションは、環境のセットアップに使用する関数で構成されています。

Amazon OpenSearch Serverless データプレーン API と OpenSearch Dashboards にブラウザからアクセスするときは、コレクションリソース用に 2 つの IAM アクセス権限を追加する必要があります。これらのアクセス権限は、`aoss:APIAccessAll` と `aoss:DashboardsAccessAll` です。

> **Note**
>
> 2023 年 5 月 10 日以降、OpenSearch Serverless では、コレクションリソースにこれら 2 つの新しい IAM 許可が必要になります。`aoss:APIAccessAll` のアクセス許可はデータ

プレーンアクセスを許可し、aoss:DashboardsAccessAll アクセス許可はブラウザから OpenSearch Dashboards を許可します。2 つの新しい IAM アクセス権限を追加しなかった場合、403 エラーが表示されます。

こちらのポリシー例では、アカウント内の指定されたコレクションの、データプレーン API へのアクセスと、アカウント内のすべてのコレクションの、OpenSearch Dashboards へのアクセスをユーザーに許可しています。

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:collection/collection-id"
        },
        {
            "Effect": "Allow",
            "Action": "aoss:DashboardsAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:dashboards/default"
        }
    ]
}
```

aoss:APIAccessAll と aoss:DashboardsAccessAll はどちらも、コレクションリソースに完全な IAM アクセス権限を付与します。Dashboards アクセス権限は、OpenSearch Dashboards アクセスも提供します。各アクセス権限は独立して機能するので、aoss:APIAccessAll での明示的な拒否によって開発ツールなどのリソースへの aoss:DashboardsAccessAll アクセスがブロックされることはありません。aoss:DashboardsAccessAll での拒否についても同じことが言えます。

OpenSearch Serverless は、データプレーン呼び出し用のプリンシパルの IAM ポリシーにおける条件設定で送信元 IP アドレスのみをサポートします。

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "52.95.4.14"
    }
}
```

Amazon OpenSearch Serverless での暗号化

保管中の暗号化

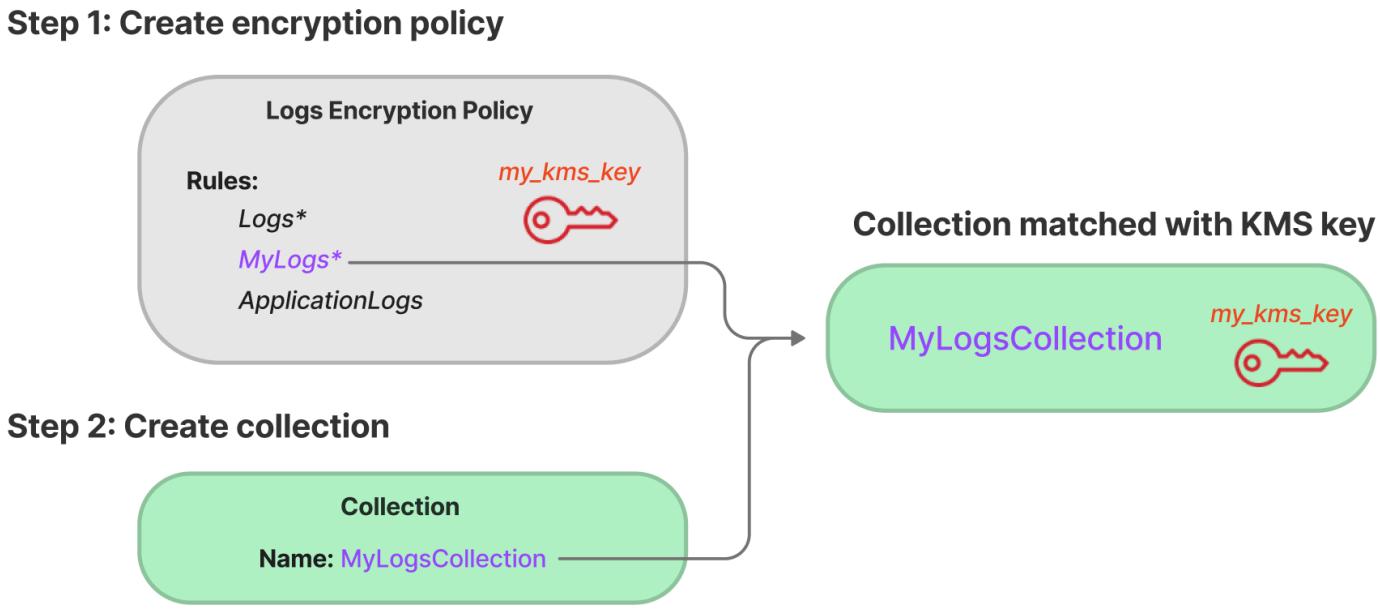
作成する各 Amazon OpenSearch Serverless コレクションは、保管中のデータの暗号化で保護されます。これは、データへの不正アクセスを防ぐのに役立つセキュリティ機能です。保管時の暗号化では、 AWS Key Management Service （ AWS KMS) を使用して暗号化キーを保存および管理します。この暗号化は、256 ビットキーを使用した Advanced Encryption Standard アルゴリズム (AES-256) を使用して実行されます。

トピック

暗号化ポリシー

暗号化ポリシーを使用すると、新しく作成され特定の名前またはパターンに一致するコレクションに対し、暗号化キーを自動的に割り当てることができ、多くのコレクションを大規模に管理できます。

暗号化ポリシーを作成する際には、プレフィックスでワイルドカードベースのマッチングルール (`MyCollection*` など) を指定するか、コレクション名を 1 つ入力します。次に、その名前またはプレフィックスパターンに一致するコレクションを作成すると、ポリシーとそれに対応する KMS キーが、そのコレクションに対し自動的に割り当てられます。



暗号化ポリシーには次の要素が含まれます。

- Rules – 1 つ以上のコレクションマッチングルール。それぞれに次のサブ要素が含まれます。
  - ResourceType – 現在、選択できるオプションは「collection」のみです。暗号化ポリシーは、コレクションリソースにのみ適用されます。
  - Resource – ポリシーが適用される 1 つ以上のコレクション名またはパターンで、形式は collection/<*collection name|pattern*> です。
- AWSOwnedKey – AWS 所有のキーを使用するかどうか。
- KmsARN – AWSOwnedKey に「false」を設定する場合は、KMS キーの Amazon リソースネーム (ARN) を指定して、関連するコレクションを暗号化します。このパラメータを含めると、OpenSearch Serverless は AWSOwnedKeyパラメータを無視します。

次のポリシー例では、autopartsinventory という名前で詳細作成されるすべてのコレクションと、「sales」という用語で始まるコレクションに対し、カスタマー管理キーを割り当てます。

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
```

```
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

ポリシーがコレクション名と一致していても、リソースパターンにワイルドカード (*) が含まれている場合には、コレクションの作成時にこの自動割り当てを上書きすることが可能です。自動キー割り当てを上書きすることを選択した場合、 OpenSearch Serverless は auto-<*collection-name >* という名前の暗号化ポリシーを作成し、コレクションにアタッチします。このポリシーは、初期状態では 1 つのコレクションにのみ適用されますが、他のコレクションを含めるように変更できます。

コレクションと一致しないようにポリシールールを変更しても、関連する KMS キーのコレクションに対する割り当ては、(自動的に) 解除されません。コレクションの暗号化には、常に初期の暗号化キーが使用されます。コレクションに対し異なる暗号化キーを使用した場合には、コレクションを再作成します。

1 つのコレクションに対し複数のポリシーのルールが一致する場合は、より詳細なルールが使用されます。例えば、あるポリシーに `collection/log*` のルールが含まれており、もう 1 つのポリシーには `collection/logSpecial` のルールが含まれている場合には、より詳細である 2 つ目のポリシーの暗号化キーが使用されます。

別のポリシーに既に存在する場合は、ポリシーで名前またはプレフィックスを使用できません。異なる暗号化ポリシーで同じリソースパターンを設定しようとすると、 OpenSearch Serverless は工ラーを表示します。

考慮事項

コレクションの暗号化を設定する際には、次の点を考慮してください。

- すべての Serverless コレクションには、保存時の暗号化が必須です。
- 使用するキーには、カスタマー管理キーと AWS 所有のキーのオプションがあります。カスタマー管理キーを使用する場合は、自動キーローテーションを有効にすることをお勧めします。
- コレクションを作成後、そのコレクションの暗号化キーを変更することはできません。コレクションを初めてセットアップするときに AWS KMS 使用する を慎重に選択します。
- コレクションに適合できるのは、単一の暗号化ポリシーのみです。

- 一意の KMS キーを持つコレクションは、 OpenSearch コンピューティングユニット (OCUs他のコレクションと共有できません。ユニークなキーを持つ各コレクションには、それぞれ独自に 4 OCU が必要です。
- 暗号化ポリシーの KMS キーを更新しても、その変更は KMS キーが既に割り当てられている、既存の一致するコレクションには影響しません。
- OpenSearch サーバーレスは、カスタマーマネージドキーに対するユーザーアクセス許可を明示的にチェックしません。データアクセスポリシーを通じてコレクションにアクセスする許可が付与されたユーザーは、関連付けられたキーで暗号化されたデータを、取り込んだりクエリできます。

必要なアクセス許可

OpenSearch Serverless の保管時の暗号化では、次の AWS Identity and Access Management (IAM) アクセス許可を使用します。IAM 条件を指定して、ユーザーを特定のコレクションに制限できます。

- `aoss:CreateSecurityPolicy` – 暗号化ポリシーを作成します。
- `aoss:ListSecurityPolicies` – アタッチされているすべての暗号化ポリシーとコレクションを一覧表示します。
- `aoss:GetSecurityPolicy` – 特定の暗号化ポリシーの詳細を表示します。
- `aoss:UpdateSecurityPolicy` – 暗号化ポリシーを変更します。
- `aoss:DeleteSecurityPolicy` – 暗号化ポリシーを削除します。

以下のアイデンティティベースのアクセスポリシー例では、ユーザーがリソースパターン `collection/application-logs` の暗号化ポリシーを管理する際に必要となる、最小限の許可を付与しています。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "aoss:CreateSecurityPolicy",
            "aoss:UpdateSecurityPolicy",
            "aoss:DeleteSecurityPolicy",
            "aoss:GetSecurityPolicy"
         ],
```

```
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"application-logs"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "aoss:ListSecurityPolicies"
         ],
         "Resource":"*"
      }
   ]
}
```

カスタマーマネージドキーのキーポリシー

コレクションを保護するためにカスタマーマネージドキーを選択すると、 OpenSearch Serverless は、選択を行うプリンシパルに代わって KMS キーを使用するアクセス許可を取得します。そのプリンシパル、ユーザー、またはロールには、 OpenSearch サーバーレスが必要とする KMS キーに対するアクセス許可が必要です。れらのアクセス許可は、キーポリシーまたは IAM ポリシーにより付与できます。

少なくとも、 OpenSearch Serverless にはカスタマーマネージドキーに対する次のアクセス許可が必要です。

- km:DescribeKey
- km:CreateGrant

例:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant"
```

```
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:ViaService": "aoss.us-east-1.amazonaws.com"
        },
        "Bool": {
          "kms:GrantIsForAWSResource": "true"
        }
      }
    }
  ]
}
```

OpenSearch サーバーレスは、kms:GenerateDataKey および kms:Decrypt アクセス許可を持つグラントを作成します。

詳細については、AWS Key Management Service デベロッパーガイドの「AWS KMSでのキーポリシーの使用」を参照してください。

OpenSearch Serverless が で許可を使用する方法 AWS KMS

OpenSearch サーバーレスでは、カスタマーマネージドキーを使用するにはグラントが必要です。

アカウントに新しいキーを使用して暗号化ポリシーを作成すると、 OpenSearch Serverless はにCreateGrantリクエストを送信して、ユーザーに代わって許可を作成します AWS KMS。の許可 AWS KMS は、顧客アカウントの KMS キーへの OpenSearch サーバーレスアクセスを許可するために使用されます。

OpenSearch Serverless では、以下の内部オペレーションでカスタマーマネージドキーを使用するには、グラントが必要です。

- にDescribeKeyリクエストを送信 AWS KMS して、指定された対称カスタマーマネージドキー ID が有効であることを確認します。
- KMS キーにGenerateDataKeyリクエストを送信して、オブジェクトを暗号化するデータキーを作成します。
- 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。

任意のタイミングで、許可に対するアクセス権を取り消したり、カスタマーマネージドキーに対するサービスからのアクセス権を削除したりできます。これを行うと、 OpenSearch Serverless

はカスタマーマネージドキーによって暗号化されたデータにアクセスできなくなります。これにより、そのデータに依存するすべてのオペレーションに影響し、非同期ワークフローでAccessDeniedException工ラーや障害が発生します。

OpenSearch サーバーレスは、特定のカスタマーマネージドキーがセキュリティポリシーまたはコレクションに関連付けられていない場合、非同期ワークフローで許可を廃止します。

暗号化ポリシーの作成 (コンソール)

暗号化ポリシーで、そのポリシーを適用する KMS キーと一連のコレクションパターンを指定します。ポリシーで定義されているパターンのいずれかに一致する新しいコレクションには、その作成時に、対応する KMS キーが割り当てられます。暗号化ポリシーは、コレクションの作成を開始する前に作成することをお勧めします。

OpenSearch サーバーレス暗号化ポリシーを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションパネルで [Serverless] (サーバーレス) を展開し、[Encryption policies] (暗号化ポリシー) を選択します。
3. [Create encryption policy] (暗号化ポリシーを作成) を選択します。
4. ポリシーの名前と説明を入力します。
5. [Resources] (リソース) に、この暗号化ポリシーに関する (1 つ以上の) リソースパターンを入力します。現在の AWS アカウント とリージョンにあり、新しく作成され、いずれかのパターンに一致しているコレクションが、自動的にこのポリシーに割り当てられます。例えば、ApplicationLogs と (ワイルドカードなしで) 入力し、後にその名前でコレクションを作成した場合、ポリシーおよび対応する KMS キーが、そのコレクションに対し割り当てられます。

   また、Logs* などのプレフィックスを指定して、名前が Logs で始まる任意の新しいコレクションに対し、ポリシーを割り当てることもできます。ワイルドカードを使用すると、複数のコレクションの暗号化設定を大規模に管理できます。
6. [Encryption] (暗号化) で、使用する KMS キーを選択します。
7. [作成] を選択します。

次のステップ: コレクションを作成する

1 つ以上の暗号化ポリシーを設定したら、それらのポリシーで定義されているルールと一致するコレクションの作成を開始できます。手順については、「the section called “コレクションの作成”」を参照してください。

コレクション作成の暗号化ステップでは、 OpenSearch Serverless は入力した名前が暗号化ポリシーで定義されたパターンと一致することを通知し、対応する KMS キーをコレクションに自動的に割り当てます。リソースパターンにワイルドカード (*) が含まれている場合は、その一致をオーバーライドして独自のキーを選択できます。

暗号化ポリシーの作成 (AWS CLI)

OpenSearch Serverless API オペレーションを使用して暗号化ポリシーを作成するには、JSON 形式でリソースパターンと暗号化キーを指定します。CreateSecurityPolicy リクエストは、インラインポリシーと .json ファイルの両方を受け入れます。

暗号化ポリシーは以下の形式になります。このサンプルファイル (my-policy.json) は、将来作成される autopartsinventory という名前のコレクションすべてと、名前が sales で始まるコレクションすべてに一致します。

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

サービス所有のキーを使用するには、AWSOwnedKey に true を設定します。

```
{
   "Rules":[
      {
```

```
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":true
}
```

次のリクエストは暗号化ポリシーを作成します。

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy file://my-policy.json
```

次に、 CreateCollection API オペレーションを使用して、リソースパターンの 1 つに一致する 1 つ以上のコレクションを作成します。

暗号化ポリシーの表示

コレクションを作成する際、コレクション名と一致するリソースパターンを持つポリシーを確認するために、アカウント内の既存の暗号化ポリシーを、プレビューする必要が生じることがあります。次のListSecurityPoliciesリクエストは、アカウント内のすべての暗号化ポリシーを一覧表示します。

```
aws opensearchserverless list-security-policies --type encryption
```

このリクエストは、すべての設定済みの暗号化ポリシーに関する情報を返します。policy 要素の内容を使用して、ポリシーで定義されているパターンルールを表示します。

```
{
   "securityPolicyDetails": [
      {
         "createdDate": 1663693217826,
         "description": "Sample encryption policy",
         "lastModifiedDate": 1663693217826,
         "name": "my-policy",
         "policy": "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"]}],\"AWSOwnedKey\":true}",
         "policyVersion": "MTY2MzY5MzIxNzgyNl8x",
```

```
        "type": "encryption"
      }
   ]
}
```

KMS キーを含む特定のポリシーに関する詳細情報を表示するには、 GetSecurityPolicy コマンドを使用します。

暗号化ポリシーの更新

暗号化ポリシー内で KMS キーを更新する場合、その変更は、設定された名前またはパターンと一致する、新規作成のコレクションだけに適用されます。KMS キーが既に割り当て済みの、既存のコレクションには影響は与えません。

同様なことが、ポリシーの一致ルールにも当てはまります。ルールを追加、変更、または削除した場合、その変更は新しく作成されたコレクションにのみ適用されます。ポリシーのルールを変更したために、コレクション名と一致しなくなったとしても、既存のコレクションに割り当てられた KMS キーは失われません。

OpenSearch サーバーレスコンソールで暗号化ポリシーを更新するには、暗号化ポリシー を選択し、変更するポリシーを選択し、編集 を選択します。変更を行ってから、[Save (保存)] を選択します。

OpenSearch Serverless API を使用して暗号化ポリシーを更新するには、 UpdateSecurityPolicyオペレーションを使用します。次のリクエストでは、新しいポリシー JSON ドキュメントを使用して、暗号化ポリシーを更新します。

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy-version 2 \
    --policy file://my-new-policy.json
```

暗号化ポリシーの削除

暗号化ポリシーを削除しても、そのポリシーで定義されている KMS キーを現在使用しているコレクションに影響はありません。 OpenSearch Serverless コンソールでポリシーを削除するには、ポリシーを選択し、「削除」を選択します。

DeleteSecurityPolicy オペレーションを使用することもできます。

```
aws opensearchserverless delete-security-policy --name my-policy --type encryption
```

転送中の暗号化

OpenSearch Serverless 内では、コレクション内のすべてのパスは、業界標準の AES-256 暗号を使用して Transport Layer Security 1.2 (TLS) を使用して転送中に暗号化されます。Opensearch のすべての API と Dashboards へのアクセスも、TLS 1.2 を経由します。TLS は、ネットワーク上でやり取りされる情報の暗号化に使用される、業界標準の暗号化プロトコルのセットです。

Amazon OpenSearch Serverless のネットワークアクセス

Amazon OpenSearch Serverless コレクションのネットワーク設定は、コレクションがパブリックネットワークからインターネット経由でアクセス可能かどうか、またはプライベートにアクセスする必要があるかどうかを決定します。

プライベートアクセスは、次のいずれかまたは両方に適用できます。

- OpenSearch サーバーレスマネージド VPC エンドポイント
- Amazon Bedrock AWS のサービス などでサポートされる

コレクションのエンドポイントとそれに対応する OpenSearch Dashboards OpenSearchエンドポイントに対して、ネットワークアクセスを個別に設定できます。

ネットワークアクセスは、さまざまなソースネットワークからのアクセスを、分離して許可するためのメカニズムです。例えば、コレクションの OpenSearch Dashboards エンドポイントがパブリックにアクセス可能であっても OpenSearch API エンドポイントがパブリックアクセス可能でない場合、ユーザーはパブリックネットワークから接続するときに Dashboards を介してのみコレクションデータにアクセスできます。パブリックネットワークから直接 OpenSearch APIsを呼び出そうとすると、ブロックされます。ソースからリソースタイプへのこのような配列には、ネットワーク設定を使用できます。Amazon OpenSearch Serverless は、IPv4 接続と IPv6 接続の両方をサポートしています。

トピック

- ネットワークポリシーの作成 (コンソール)
- ネットワークポリシーの作成 (AWS CLI)
- ネットワークポリシーの表示
- ネットワークポリシーの更新
- ネットワークポリシーの削除

ネットワークポリシー

ネットワークポリシーでは、その中でルールを定義し、そのルールに一致するコレクションに対し、ネットワークアクセス設定を自動的に割り当てることにより、多数のコレクションを大規模に管理できます。

ネットワークポリシーでは、一連のルールを指定します。これらのルールは、コレクションエンドポイントと OpenSearch Dashboards エンドポイントへのアクセス許可を定義します。各ルールは、アクセスタイプ (パブリックまたはプライベート) とリソースタイプ (コレクションおよび/または OpenSearch Dashboards エンドポイント) で構成されます。リソースタイプ (`collection` および `dashboard`) ごとに、一連のルールを指定し、ポリシーを適用する対象となるコレクションを定義します。

このサンプルポリシーでは、最初のルールは、 という用語で始まるすべてのコレクションのコレクションエンドポイントと Dashboards エンドポイントの両方への VPC エンドポイントアクセスを指定します`marketing*`。また、Amazon Bedrock アクセスも指定します。

> **Note**
>
> Amazon Bedrock AWS のサービス などの へのプライベートアクセスは、 OpenSearch Dashboards OpenSearch エンドポイントではなく、コレクションのエンドポイントにのみ適用されます。`ResourceType` が であっても`dashboard`、 OpenSearch Dashboards へのアクセスを許可 AWS のサービス することはできません。

2 番目のルールでは、`finance` コレクションに対してパブリックアクセスを指定していますが、これには、コレクションエンドポイントのみが使用されます (ダッシュボードからのアクセスは許可されません)。

```
[
   {
```

```
      "Description":"Marketing access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/marketing*"
            ]
         },
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/marketing*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   },
   {
      "Description":"Sales access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

このポリシーは、「財務」で始まるコレクションの OpenSearch Dashboards へのパブリックアクセスのみを提供します。 OpenSearch API に直接アクセスしようとすると失敗します。

```
[
  {
    "Description": "Dashboards access",
```

```
    "Rules": [
      {
        "ResourceType": "dashboard",
        "Resource": [
          "collection/finance*"
        ]
      }
    ],
    "AllowFromPublic": true
  }
]
```

ネットワークポリシーは、将来作成するコレクションだけでなく、既存のコレクションにも適用が可能です。コレクションを作成した後に、そのコレクション名と一致するルールを含むネットワークポリシーを作成できます。コレクションを作成する前に、必ずしもネットワークポリシーを用意しておく必要はありません。

考慮事項

コレクションのためにネットワークアクセスを設定する際は、以下の点を考慮してください。

- コレクションに VPC エンドポイントアクセスを設定する場合は、まず少なくとも 1 OpenSearch つのサーバーレスマネージド VPC エンドポイント を作成する必要があります。
- へのプライベートアクセスは、 OpenSearch Dashboards OpenSearchエンドポイントではなく、コレクションのエンドポイント AWS のサービス にのみ適用されます。`ResourceType` が であってもdashboard、 OpenSearch Dashboards へのアクセスを許可 AWS のサービス することはできません。
- コレクションがパブリックネットワークからアクセス可能な場合は、すべての OpenSearch Serverless マネージド VPC エンドポイントとすべての からもアクセスできます AWS のサービス。
- 単一のコレクションに対して、複数のネットワークポリシーを適用できます。詳細については、「the section called “ポリシーの優先順位”」を参照してください。

ネットワークポリシーの設定に必要なアクセス許可

OpenSearch Serverless のネットワークアクセスでは、次の AWS Identity and Access Management (IAM) アクセス許可を使用します。IAM 条件を指定しすることで、特定のコレクションに関連付けられたネットワークポリシーのみをユーザーが使用するように制限できます。

- aoss:CreateSecurityPolicy – ネットワークアクセスポリシーを作成します。
- aoss:ListSecurityPolicies – 現在のアカウントにある、すべてのネットワークポリシーを一覧表示します。
- aoss:GetSecurityPolicy – ネットワークアクセスポリシーの設定を表示します。
- aoss:UpdateSecurityPolicy – 特定のネットワークアクセスポリシーを変更し、VPC ID またはパブリックアクセスの指定を変更します。
- aoss:DeleteSecurityPolicy – ネットワークアクセスポリシーを (すべてのコレクションからデタッチした後に) 削除します。

次の ID ベースのアクセスポリシーにより、ユーザーはすべてのネットワークポリシーを表示し、リソースパターン collection/application-logs を含むポリシーの更新が行えるようになります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListSecurityPolicies",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> さらに、 OpenSearch Serverless にはコレクションリソースの `aoss:APIAccessAll`および アクセス`aoss:DashboardsAccessAll`許可が必要です。詳細については、「the section called “OpenSearch API オペレーションの使用”」を参照してください。

ポリシーの優先順位

ポリシー内またはポリシー間に、複数のネットワークポリシールールが同時に存在する場合があります。この場合、パブリックアクセスを指定するルールは、両方のルールに共通するコレクションのプライベートアクセスを指定するルールよりも優先されます。

例えば、次のポリシーでは、2 種類のルールが `finance` コレクションにネットワークアクセスを割り当てています。一方のルールでは VPC アクセスを、もう一方のルールではパブリックアクセスをそれぞれ指定しています。この場合、finance コレクションでのみ、パブリックアクセスは VPC アクセスをオーバーライドします (両方のルールに含まれるのがこのコレクションであるため)。つまり、finance コレクションにアクセスできるのは、パブリックネットワークからとなります。sales コレクションには、指定されたエンドポイントからの VPC アクセスが許可されます。

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/sales",
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   },
   {
      "Description":"Rule 2",
      "Rules":[
         {
```

```
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

異なるルールの複数の VPC エンドポイントが 1 つのコレクションに適用されている場合、これらのルールは追加的に認識され、指定されたすべてのエンドポイントから、そのコレクションへのアクセスが可能になります。`AllowFromPublic` を に設定`true`し、1 つ以上の `SourceVPCEs`または も指定した場合`SourceServices`、 OpenSearch Serverless は VPC エンドポイントとサービス識別子を無視し、関連するコレクションはパブリックアクセスを持ちます。

ネットワークポリシーの作成 (コンソール)

ネットワークポリシーは、将来作成されるポリシーだけでなく、既存のポリシーにも適用できます。コレクションの作成を開始する前に、ネットワークポリシーを作成しておくことをお勧めします。

OpenSearch サーバーレスネットワークポリシーを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションパネルで [Serverless] (サーバーレス) を展開し、[Network policie] (ネットワークポリシー) を選択します。
3. [Create network policy] (ネットワークポリシーを作成) を選択します。
4. ポリシーの名前と説明を入力します。
5. 1 つ以上のルールを指定します。これらのルールは、 OpenSearch サーバーレスコレクションとその OpenSearch Dashboards エンドポイントのアクセス許可を定義します。

   各ルールには以下の要素が含まれます。

要素	説明
ルール名	ルールの内容をわかりやすく示した名前です。例えば、「VPC access for marketing team」などとします。
アクセスタイプ	パブリックアクセスまたはプライベートアクセスを選択します。次に、次のいずれかまたは両方を選択します。 • アクセス用の VPC エンドポイント – OpenSearch サーバーレスマネージド VPC エンドポイント – マネージド VPC エンドポイントを 1 つ以上指定します。 • AWS のサービス プライベートアクセス – サポートされている を 1 つ以上選択します AWS のサービス。

要素	説明
リソースタイプ	OpenSearch エンドポイント (OpenSearch API への呼び出しを許可)、 OpenSearch Dashboards (OpenSearch プラグインの視覚化とユーザーインターフェイスへのアクセスを許可)、またはその両方へのアクセスを提供するかどうかを選択します。 **Note** AWS のサービス プライベートアクセスはコレクションの OpenSearch エンドポイントにのみ適用され、 OpenSearch Dashboards エンドポイントには適用されません。OpenSearch Dashboards を選択した場合でも、 AWS のサービス に付与できるのはエンドポイントアクセスのみです。

選択したリソースタイプごとに、ポリシー設定を適用する既存のコレクションを選択すること、および/または 1 つ以上のリソースパターンを作成することができます。リソースパターンはプレフィックスとワイルドカード (*) で構成され、ポリシー設定の適用対象となるコレクションを定義します。

例えば、`Marketing*` というパターンを含めると、名前が「Marketing」で始まる新規または既存のコレクションには、このポリシーのネットワーク設定が自動的に適用されます。ワイルドカード (*) によって、既存のコレクションや将来作成されるコレクションすべてにポリシーが適用されます。

さらに、ワイルドカードなしで将来のコレクションの名前を指定できます`Finance`。OpenSearch Serverless は、その正確な名前で新しく作成されたコレクションにポリシー設定を適用します。

6. ポリシー設定を確認し、変更点がない場合は、[Create] (作成) を選択します。

ネットワークポリシーの作成 (AWS CLI)

OpenSearch Serverless API オペレーションを使用してネットワークポリシーを作成するには、JSON 形式でルールを指定します。CreateSecurityPolicy リクエストは、インラインポリシーと .json ファイルの両方を受け入れます。すべてのコレクションとパターンは、`collection/<collection name|pattern>` の形式にする必要があります。

> **Note**
>
> リソースタイプは OpenSearch Dashboards へのアクセスのみdashboardsを許可しますが、 OpenSearch Dashboards を機能させるには、同じソースからのコレクションアクセスも許可する必要があります。例については、以下に示す 2 番目のポリシーを参照してください。

プライベートアクセスを指定するには、次の要素のいずれかまたは両方を含めます。

- `SourceVPCEs` – OpenSearch サーバーレスマネージド VPC エンドポイントを 1 つ以上指定します。
- `SourceServices` – サポートされている 1 つ以上の の識別子を指定します AWS のサービス。現在、以下のサービス識別子がサポートされています。
  - `bedrock.amazonaws.com` – Amazon Bedrock

次のサンプルネットワークポリシーは、VPC エンドポイントと Amazon Bedrock へのプライベートアクセスを、プレフィックス で始まるコレクションのコレクションエンドポイントに対してのみ提供しますlog*。認証されたユーザーは OpenSearch Dashboards にサインインできません。コレクションエンドポイントにはプログラムでのみアクセスできます。

```
[
   {
      "Description":"Private access for log collections",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/log*"
            ]
         }
      ],
```

```
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   }
]
```

次のポリシーは、 という名前の 1 つのコレクションの OpenSearch エンドポイントと OpenSearch Dashboards へのパブリックアクセスを提供しますfinance。対象のコレクションが存在しない場合、ネットワーク設定は、そのコレクションが作成された時点で適用されます。

```
[
   {
      "Description":"Public access for finance collection",
      "Rules":[
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/finance"
            ]
         },
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

次のリクエストにより、上記のネットワークポリシーが作成されます。

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type network \
    --policy "[{\"Description\":\"Public access for finance collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/finance\"]},
```

```
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/finance\"]}],
\"AllowFromPublic\":true}]"
```

このポリシーを JSON ファイルで指定するには、`--policy file://my-policy.json` の形式を使用します。

ネットワークポリシーの表示

コレクションを作成する前に、アカウント内の既存のネットワークポリシーをプレビューして、コレクション名と一致するリソースパターンが含まれているポリシーを確認しておきたい場合があります。次のListSecurityPoliciesリクエストは、アカウント内のすべてのネットワークポリシーを一覧表示します。

```
aws opensearchserverless list-security-policies --type network
```

このリクエストは、設定されているすべてのネットワークポリシーに関する情報を返します。1 つの特定のポリシーで定義されているパターンルールを表示するには、レスポンスの `securityPolicySummaries` 要素の内容でポリシー情報を探します。このポリシーtypeの nameと をメモし、これらのプロパティをGetSecurityPolicyリクエストで使用して、次のポリシーの詳細を含むレスポンスを受け取ります。

```
{
    "securityPolicyDetail": [
        {
            "type": "network",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "policy": "[{\"Description\":\"My network policy rule\",\"Rules\":
[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection/*\"]}],\"AllowFromPublic
\":true}]",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
    ]
}
```

特定のポリシーに関する詳細情報を表示するには、 GetSecurityPolicy コマンドを使用します。

ネットワークポリシーの更新

ネットワークの VPC エンドポイントまたはパブリックアクセスの指定を変更すると、関連するすべてのコレクションが影響を受けます。 OpenSearch サーバーレスコンソールでネットワークポリシーを更新するには、ネットワークポリシー を展開し、変更するポリシーを選択し、編集 を選択します。変更を行ってから、[Save (保存)] を選択します。

OpenSearch Serverless API を使用してネットワークポリシーを更新するには、UpdateSecurityPolicy コマンドを使用します。リクエストには、ポリシーのバージョンを含める必要があります。ポリシーのバージョンは、`ListSecurityPolicies` または `GetSecurityPolicy` コマンドを使用して取得できます。最新のポリシーバージョンを含めると、他のユーザーによる変更を意図せず上書きしてしまうことがなくなります。

次のリクエストは、ネットワークポリシーを新しいポリシーの JSON ドキュメントで更新します。

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type network \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

ネットワークポリシーの削除

ネットワークポリシーを削除する前に、そのポリシーをすべてのコレクションからデタッチする必要があります。 OpenSearch Serverless コンソールでポリシーを削除するには、ポリシーを選択し、「削除」を選択します。

DeleteSecurityPolicy コマンドを使用することもできます。

```
aws opensearchserverless delete-security-policy --name my-policy --type network
```

Amazon OpenSearch Serverless のデータアクセスコントロール

Amazon OpenSearch Serverless のデータアクセスコントロールを使用すると、アクセスメカニズムやネットワークソースに関係なく、ユーザーがコレクションとインデックスにアクセスできるようになります。IAM ロールと SAML アイデンティティへのアクセスを提供できます。

アクセス許可は、コレクションとインデックスリソースに適用されるデータアクセスポリシーを通じて管理します。データアクセスポリシーは、特定のパターンに一致するコレクションとインデックスにアクセス許可を自動的に割り当てることにより、大規模なコレクションを管理するのに役立ち

ます。1 つのリソースに複数のデータアクセスポリシーを適用できます。 OpenSearch Dashboards URL にアクセスするには、コレクションのデータアクセスポリシーが必要です。

トピック

データアクセスポリシーと IAM ポリシーの比較

データアクセスポリシーは、 AWS Identity and Access Management (IAM) ポリシーとは論理的に分離されます。IAM アクセス許可は、`CreateCollection` や `ListAccessPolicies` などのサーバーレス API オペレーションへのアクセスを制御します。データアクセスポリシーは、 `PUT <index>`や など、 OpenSearch サーバーレスがサポートするOpenSearch オペレーションへのアクセスを制御します`GET _cat/indices`。

`aoss:CreateAccessPolicy` や `aoss:GetAccessPolicy` (次のセクションで説明) などのデータアクセスポリシーの API オペレーションへのアクセスを制御する IAM アクセス許可は、データアクセスポリシーで指定されているアクセス許可には影響しません。

例えば、IAM ポリシーによってユーザーによる `collection-a` に対するデータアクセスポリシーの作成は拒否されているが、すべてのコレクション (*) に対するデータアクセスポリシーの作成は許可されているとします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Deny",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "collection-a"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

ユーザーがすべてのコレクション (collection/* または index/*/*) に特定のアクセス許可を付与するデータアクセスポリシーを作成した場合、ポリシーはコレクション A を含むすべてのコレクションに適用されます。

> **Important**
>
> データアクセスポリシー内でアクセス許可を付与しても、 OpenSearch サーバーレスコレクションのデータにアクセスするのに十分ではありません。関連付けられたプリンシパルにも、IAM 許可 aoss:APIAccessAll および aoss:DashboardsAccessAll に対するアクセスが付与されている必要があります。どちらのアクセス許可もコレクションリソースへのフルアクセスを許可し、Dashboards アクセス許可は OpenSearch Dashboards へのアクセスも提供します。プリンシパルがこれらの両方の IAM 許可を持っていない場合、コレクションにリクエストを送信しようとすると 403 エラーが表示されます。詳細については、「the section called “OpenSearch API オペレーションの使用”」を参照してください。

データアクセスポリシーの設定に必要な IAM アクセス許可

OpenSearch Serverless のデータアクセスコントロールは、次の IAM アクセス許可を使用します。IAM 条件を指定して、ユーザーを特定のアクセスポリシー名に制限できます。

- aoss:CreateAccessPolicy – アクセスポリシーを作成します。
- aoss:ListAccessPolicies – すべてのアクセスポリシーを一覧表示します。
- aoss:GetAccessPolicy – 特定のアクセスポリシーの詳細を表示します。
- aoss:UpdateAccessPolicy – アクセスポリシーを変更します。
- aoss:DeleteAccessPolicy – アクセスポリシーを削除します。

次の ID ベースのアクセスポリシーでは、ユーザーはすべてのアクセスポリシーを表示し、リソースパターン collection/logs を含むポリシーを更新できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:GetAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "aoss:UpdateAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "logs"
                    ]
                }
            }
        }
    ]
}
```

> **Note**
>
> さらに、 OpenSearch Serverless にはコレクションリソースの `aoss:APIAccessAll`および アクセス`aoss:DashboardsAccessAll`許可が必要です。詳細については、「the section called “OpenSearch API オペレーションの使用”」を参照してください。

ポリシー構文

データアクセスポリシーには、次の要素を持つ一連のルールが含まれています。

要素	説明
`ResourceType`	アクセス許可が適用されるリソースのタイプ (コレクションまたはインデックス)。エイリアスとテンプレートのアクセス許可はコレクションレベルで、データの作成、変更、検索のアクセス許可はインデックスレベルです。詳細については、「Supported policy permissions」(サポートされているポリシーのアクセス許可) を参照してください。
`Resource`	リソース名やパターンのリスト。パターンはプレフィックスの後にワイルドカード (*) が続くもので、これによって関連付けられたアクセス許可を複数のリソースに適用できます。 • コレクションの形式は `collection/ <name|pattern>` です。 • インデックスの形式は `index/<collection-name|pattern> /<index-name|pattern/>` です。
`Permission`	指定されたリソースに付与するアクセス許可のリスト。これによって許可されるアクセス許可とオペレーションの完全なリストについては、「the section called “サポートされている OpenSearch API オペレーションとアクセス許可”」を参照してください。
`Principal`	アクセス権が付与される 1 つ以上のプリンシパルのリスト。プリンシパルは、IAM ロール ARN または SAML アイデンティティにすることができます。これらのプリンシパルは現在の AWS アカウント内にある必要があります。データアクセスポリシーはクロスアカウントアクセスを直接サポートしていませんが、別の のユーザーがコレクション所有アカウントで引き受け AWS アカウント ることができるロールをポリシーに含めることができま

要素	説明
	す。詳細については、「the section called “クロスアカウントデータアクセス”」を参照してください。

次のポリシー例では、autopartsinventory というコレクションおよび sales* というプレフィックスで始まるすべてのコレクションにエイリアスとテンプレートのアクセス許可を付与します。また、autopartsinventory コレクション内のすべてのインデックス、および orders* というプレフィックスで始まる salesorders コレクション内のすべてのインデックスに読み取りおよび書き込みアクセス許可を付与します。

```
[
   {
      "Description": "Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/autopartsinventory",
               "collection/sales*"
            ],
            "Permission":[
               "aoss:CreateCollectionItems",
               "aoss:UpdateCollectionItems",
               "aoss:DescribeCollectionItems"
            ]
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/autopartsinventory/*",
               "index/salesorders/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
```

```
            "saml/123456789012/myprovider/user/Annie",
            "saml/123456789012/anotherprovider/group/Accounting"
        ]
    }
]
```

ポリシー内でアクセスを明示的に拒否することはできません。したがって、ポリシーのアクセス許可はすべて追加的です。例えば、あるポリシーでユーザーに aoss:ReadDocument を付与し、別のポリシーで aoss:WriteDocument を付与した場合、ユーザーには両方のアクセス許可が付与されます。3 番目のポリシーで同じユーザーに aoss:* を付与した場合、そのユーザーは関連付けられたインデックスですべてのアクションを実行できます。制限の厳しいアクセス許可が制限の緩いアクセス許可よりも優先されることはありません。

サポートされているポリシーのアクセス許可

データアクセスポリシーでは、次のアクセス許可がサポートされています。各アクセス許可で許可される API オペレーションについては OpenSearch、「」を参照してくださいthe section called “サポートされている OpenSearch API オペレーションとアクセス許可”。

コレクションアクセス許可

- aoss:CreateCollectionItems
- aoss:DeleteCollectionItems
- aoss:UpdateCollectionItems
- aoss:DescribeCollectionItems
- aoss:*

インデックスアクセス許可

- aoss:ReadDocument
- aoss:WriteDocument
- aoss:CreateIndex
- aoss:DeleteIndex
- aoss:UpdateIndex
- aoss:DescribeIndex
- aoss:*

OpenSearch Dashboards のサンプルデータセット

OpenSearch Dashboards には、独自のデータを追加する前に Dashboards を調べるのに役立つ視覚化、ダッシュボード、その他のツールを含むサンプルデータセットが用意されています。このサンプルデータからインデックスを作成するには、処理するデータセットへのアクセスを許可する、データアクセスポリシーが必要になります。次のポリシーでは、ワイルドカード (*) を使用して、3 つのサンプルデータセットすべてへのアクセスを許可します。

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/<collection-name>/opensearch_dashboards_sample_data_*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::<account-id>:user/<user>"
    ]
  }
]
```

データアクセスポリシーの作成 (コンソール)

ビジュアルエディタを使用して、または JSON 形式で、データアクセスポリシーを作成できます。ポリシーで定義されているパターンのいずれかに一致する新しいコレクションには、対応するアクセス許可がコレクションの作成時に割り当てられます。

OpenSearch サーバーレスデータアクセスポリシーを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインで、[Serverless] (サーバーレス) を展開し、[Data access control] (データアクセスコントロール) を選択します。

3. [Create access policy] (アクセスポリシーの作成) を選択します。
4. ポリシーの名前と説明を入力します。
5. ポリシーの最初のルールの名前を入力します。例えば、「Logs collection access」。
6. [Add principals] (プリンシパルの追加) を選択し、データアクセス権が付与される IAM ロール、または SAML ユーザーとグループを 1 つ、または複数選択します。

> **Note**
>
> ドロップダウンメニューからプリンシパルを選択するには、`iam:ListUsers` および `iam:ListRoles` アクセス許可 (IAM プリンシパルの場合) および `aoss:ListSecurityConfigs` アクセス許可 (SAML ID の場合) が必要です。

7. [Grant] (付与) を選択し、エイリアス、テンプレート、およびインデックスのアクセス許可を選択して、関連するプリンシパルに付与します。アクセス許可とそれによって許可されるアクセスの完全なリストについては、「the section called “サポートされている OpenSearch API オペレーションとアクセス許可”」を参照してください。
8. (オプション) ポリシーに追加のルールを設定します。
9. [作成] を選択します。ポリシーを作成してからアクセス許可が適用されるまでに 1 分程度かかる場合があります。5 分以上かかる場合は、AWS Support にお問い合わせください。

> **Important**
>
> ポリシーにインデックス許可のみが含まれていてコレクション許可が含まれていない場合でも、`Collection cannot be accessed yet. Configure data access policies so that users can access the data within this collection` といった、一致するコレクションに関するメッセージが表示されることがあります。この警告は無視できます。許可されたプリンシパルは引き続き、それぞれに割り当てられたインデックス関連の操作をコレクションで実行できます。

データアクセスポリシーの作成 (AWS CLI)

OpenSearch Serverless API を使用してデータアクセスポリシーを作成するには、`CreateAccessPolicy` コマンドを使用します。コマンドは、インラインポリシーと .json ファイルの両方を受け入れます。インラインポリシーは JSON エスケープ文字列としてエンコードする必要があります。

次のリクエストで、データアクセスポリシーが作成されます。

```
aws opensearchserverless create-access-policy \
    --name marketing \
    --type data \
    --policy "[{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"],\"Permission\":
[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",\"Resource\":
[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],\"Permission
\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]"
```

.json ファイル内でポリシーを指定するには、`--policy file://my-policy.json` の形式を使用します。

ポリシーに含まれるプリンシパルは、アクセスが付与されたOpenSearch オペレーションを使用できるようになりました。

データアクセスポリシーの表示

コレクションを作成する前に、アカウント内の既存のデータアクセスポリシーをプレビューして、コレクション名と一致するリソースパターンがあるポリシーを確認することをお勧めします。次のListAccessPoliciesリクエストは、アカウント内のすべてのデータアクセスポリシーを一覧表示します。

```
aws opensearchserverless list-access-policies --type data
```

リクエストは、設定されているすべてのデータアクセスポリシーに関する情報を返します。1 つの特定のポリシーで定義されているパターンルールを表示するには、レスポンスの accessPolicySummaries 要素の内容でポリシー情報を探します。このポリシーtypeの と をメモnameし、GetAccessPolicyリクエストでこれらのプロパティを使用して、次のポリシーの詳細を含むレスポンスを受け取ります。

```
{
    "accessPolicyDetails": [
        {
            "type": "data",
            "name": "my-policy",
            "policyVersion": "MTY2NDA1NDE4MDg1OF8x",
            "description": "My policy",
```

```
            "policy": "[{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection/autopartsinventory\",\"collection/sales*\"],
\"Permission\":[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",
\"Resource\":[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],
\"Permission\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]",
            "createdDate": 1664054180858,
            "lastModifiedDate": 1664054180858
        }
    ]
}
```

次のようにリソースフィルターを追加して、結果を特定のコレクションまたはインデックスを含むポリシーに限定できます。

```
aws opensearchserverless list-access-policies --type data --resource
 "index/autopartsinventory/*"
```

特定のポリシーの詳細を表示するには、 GetAccessPolicy コマンドを使用します。

データアクセスポリシーの更新

データアクセスポリシーを更新すると、関連するすべてのコレクションが影響を受けます。OpenSearch Serverless コンソールでデータアクセスポリシーを更新するには、データアクセスコントロール を選択し、変更するポリシーを選択し、編集 を選択します。変更を行ってから、[Save (保存)] を選択します。

OpenSearch Serverless API を使用してデータアクセスポリシーを更新するには、UpdateAccessPolicyリクエストを送信します。ポリシーバージョンを含める必要があります。ポリシーバージョンは、ListAccessPolicies または GetAccessPolicy コマンドを使用して取得できます。最新のポリシーバージョンを含めると、他のユーザーによる変更を意図せず上書きしてしまうことがなくなります。

次のUpdateAccessPolicyリクエストは、データアクセスポリシーを新しいポリシー JSON ドキュメントで更新します。

```
aws opensearchserverless update-access-policy \
    --name sales-inventory \
    --type data \
    --policy-version MTY2NDA1NDE4MDg1OF8x \
    --policy file://my-new-policy.json
```

ポリシーを更新してから新しいアクセス許可が適用されるまでに数分かかる場合があります。

データアクセスポリシーの削除

データアクセスポリシーを削除すると、関連するすべてのコレクションは、ポリシーで定義されているアクセス権を失います。ポリシーを削除する前に、IAM ユーザーと SAML ユーザーがコレクションに適切なアクセス権を持っていることを確認してください。 OpenSearch Serverless コンソールでポリシーを削除するには、ポリシーを選択し、「削除」を選択します。

DeleteAccessPolicy コマンドを使用することもできます。

```
aws opensearchserverless delete-access-policy --name my-policy --type data
```

クロスアカウントデータアクセス

クロスアカウント ID またはクロスアカウントコレクションを使用してデータアクセスポリシーを作成することはできませんが、ロールの継承オプションを使用してクロスアカウントアクセスをセットアップすることはできます。例えば、 がアクセス*account-b*が必要なコレクション*account-a*を所有している場合、 のユーザーは でロールを引き受け*account-b*ることができます*account-a*。ロールには IAM アクセス許可`aoss:APIAccessAll`と があり`aoss:DashboardsAccessAll`、 のデータアクセスポリシーに含まれている必要があります*account-a*。

インターフェイスエンドポイント (AWS PrivateLink) を使用して Amazon OpenSearch Serverless にアクセスする

を使用して AWS PrivateLink 、VPC と Amazon OpenSearch Serverless の間にプライベート接続を作成できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続を使用せずに、VPC 内にあるかのように OpenSearch Serverless にアクセスできます。VPC 内のインスタンスは、 OpenSearch サーバーレスにアクセスするためにパブリック IP アドレスを必要としません。

このプライベート接続を確立するには、 AWS PrivateLinkを利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに指定した各サブネットに、エンドポイントネットワークインターフェイスを作成します。これらは、 OpenSearch サーバーレス宛てのトラフィックのエントリポイントとして機能するリクエスタ管理のネットワークインターフェイスです。

詳細については、『AWS PrivateLink ガイド』の「AWS のサービス でアクセスする」を参照してください。

トピック

コレクションエンドポイントの DNS 解決

VPC エンドポイントを作成すると、サービスは新しい Amazon Route 53 プライベートホストゾーンを作成し、VPC にアタッチします。このプライベートホストゾーンは、 OpenSearch サーバーレスコレクション (`*.aoss.us-east-1.amazonaws.com`) のワイルドカード DNS レコードをエンドポイントに使用されるインターフェイスアドレスに解決するためのレコードで構成されます。各のすべてのコレクションとダッシュボードにアクセスするには、VPC に OpenSearch サーバーレス VPC エンドポイントが 1 つだけ必要です AWS リージョン。 OpenSearch Serverless のエンドポイントを持つすべての VPC には、独自のプライベートホストゾーンがアタッチされています。

OpenSearch Serverless は、リージョン内のすべてのコレクションに対してパブリック Route 53 ワイルドカード DNS レコードも作成します。DNS 名は Serverless OpenSearch パブリック IP アドレスに解決されます。 OpenSearch サーバーレス VPCs エンドポイントを持たない VPC 内のクライアント、またはパブリックネットワーク内のクライアントは、パブリック Route 53 リゾルバーを使用して、それらの IP アドレスを持つコレクションとダッシュボードにアクセスできます。VPC エンドポイントの IP アドレスタイプ (IPv4、IPv6、またはデュアルスタック) は、 OpenSearch Serverless のインターフェイスエンドポイントを作成するときに提供されるサブネットに基づいて決定されます。

> **Note**
>
> の update-vpc-endpoint コマンドを使用して、既存の IPv4 VPC エンドポイントをデュアルスタックに更新できます AWS CLI。

特定の VPC の DNS リゾルバーアドレスは、VPC CIDR の 2 番目の IP アドレスです。VPC 内のクライアントは、そのリゾルバーを使用して、コレクション用の VPC エンドポイントアドレスを取得

する必要があります。リゾルバーは、 OpenSearch サーバーレスによって作成されたプライベートホストゾーンを使用します。任意のアカウントのすべてのコレクション用にそのリゾルバーを使用すれば十分です。一部のコレクションエンドポイントには VPC リゾルバーを使用し、他のコレクションエンドポイントにはパブリックリゾルバーを使用することもできますが、通常は必要ありません。

VPC とネットワークアクセスポリシー

コレクションの OpenSearch APIsと Dashboards にネットワークアクセス許可を付与するには、OpenSearch サーバーレスネットワークアクセスポリシー を使用できます。このネットワークアクセスは、VPC エンドポイントまたはパブリックインターネットのいずれからでも制御できます。ネットワークポリシーはトラフィックの許可のみを制御するため、コレクション内のデータとそのインデックスに対して操作するための許可を指定するデータアクセスポリシーも設定する必要があります。 OpenSearch サーバーレス VPC エンドポイントをサービスへのアクセスポイント、ネットワークアクセスポリシーをコレクションとダッシュボードへのネットワークレベルのアクセスポイント、データアクセスポリシーをコレクション内のデータに対するあらゆるオペレーションのきめ細かなアクセスコントロールのアクセスポイントと考えてください。

ネットワークポリシーでは複数の VPC エンドポイント ID を指定できるため、コレクションにアクセスする必要がある VPC ごとに VPC エンドポイントを作成することをお勧めします。これらの VPCs、 OpenSearch サーバーレスコレクションとネットワークポリシーを所有する AWS アカウントとは異なるアカウントに属している可能性があります。あるアカウントの VPC が別のアカウントの VPC エンドポイントを使用することを目的として、2 つのアカウント間で VPC 間ピアリングまたは他のプロキシソリューションを作成することはお勧めしません。これは、各 VPC が独自のエンドポイントを持つ場合よりも安全性とコスト効率が低くなります。最初の VPC を、ネットワークポリシーで他の VPC のエンドポイントへのアクセスを設定しているその VPC の管理者が簡単に表示することはできません。

VPC とエンドポイントポリシー

Amazon OpenSearch Serverless はVPCsエンドポイントポリシーをサポートしています。エンドポイントポリシーは、VPC エンドポイントにアタッチして、エンドポイント AWS を使用して AWS サービスにアクセスできるプリンシパルを制御する IAM リソースベースのポリシーです。詳細については、「エンドポイントポリシーを使用して VPC エンドポイントへのアクセスを制御する」を参照してください。

エンドポイントポリシーを使用するには、まずインターフェイスエンドポイントを作成する必要があります。Serverless コンソールまたは OpenSearch Serverless API OpenSearch を使用してインターフェイスエンドポイントを作成できます。インターフェイスエンドポイントを作成した後、エンドポイントポリシーをエンドポイントに追加する必要があります。詳細については、「インターフェイス

エンドポイント (AWS PrivateLink) を使用して Amazon OpenSearch Serverless にアクセスする」を参照してください。

> **Note**
>
> OpenSearch サービスコンソールでエンドポイントポリシーを直接定義することはできません。

エンドポイントポリシーは、設定済みの他の ID ベースポリシー、リソースベースポリシー、ネットワークポリシー、またはデータアクセスポリシーをオーバーライドしたり、これらに置き換わったりすることはありません。エンドポイントポリシーの更新の詳細については、「エンドポイントポリシーを使用して VPC エンドポイントへのアクセスを制御する」を参照してください。

デフォルトでは、エンドポイントポリシーにより、VPC エンドポイントに対するフルアクセスが付与されます。

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

デフォルトの VPC エンドポイントポリシーにより、エンドポイントに対するフルアクセスが付与されますが、特定のロールおよびユーザーに対するアクセスを許可するように VPC エンドポイントポリシーを設定できます。これを実行するには、次の例を参照してください:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012",
```

```
                    "987654321098"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

VPC エンドポイントポリシーの条件付き要素として含める OpenSearch サーバーレスコレクションを指定できます。これを実行するには、次の例を参照してください:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CollectionName": [
                        "coll-abc"
                    ]
                }
            }
        }
    ]
}
```

VPC エンドポイントポリシーで SAML ID を使用して、VPC エンドポイントアクセスを決定できます。VPC エンドポイントポリシーのプリンシパルセクションではワイルドカード (*) を使用する必要があります。これを実行するには、次の例を参照してください:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
```

```
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
                }
            }
        }
    ]
}
```

さらに、特定の SAML プリンシパルポリシーを含めるようにエンドポイントポリシーを設定できます。これを実行するには、次を参照してください:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SamlPrincipal": [
                        "saml/123456789012/idp123/user/user1234"]
                    }
                }
            }
        ]
    }
```

Amazon OpenSearch Serverless での SAML 認証の使用の詳細については、「Amazon OpenSearch Serverless の SAML 認証」を参照してください。

また、IAM ユーザーと SAML ユーザーを同じ VPC エンドポイントポリシーに含めることもできます。これを実行するには、次の例を参照してください:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

考慮事項

OpenSearch Serverless のインターフェイスエンドポイントを設定する前に、次の点を考慮してください。

- OpenSearch Serverless は、インターフェイスエンドポイントを介した、サポートされているすべての OpenSearch API オペレーション (設定 API オペレーションではありません) の呼び出しをサポートしています。
- Serverless のインターフェイスエンドポイントを作成した後も、 OpenSearch サーバーレスコレクションにアクセスするために、ネットワークアクセスポリシーに含める必要があります。
- デフォルトでは、インターフェイスエンドポイントを介して OpenSearch サーバーレスへのフルアクセスが許可されます。セキュリティグループをエンドポイントネットワークインターフェイス

に関連付けると、インターフェイスエンドポイントを介して OpenSearch サーバーレスへのトラフィックを制御できます。

- 1 つの に最大 50 個の OpenSearch サーバーレス VPC エンドポイント AWS アカウント を設定できます。
- ネットワークポリシーでコレクションの API または Dashboards へのパブリックインターネットアクセスを有効にすると、コレクションは任意の VPC およびパブリックインターネットからアクセスできるようになります。
- オンプレミスで VPC の外部にいる場合は、 OpenSearch サーバーレス VPC エンドポイント解決に DNS リゾルバーを直接使用することはできません。VPN アクセスが必要な場合、VPC には外部クライアントが使用できる DNS プロキシリゾルバーが必要です。Route 53 は、オンプレミスネットワークまたは別の VPC から VPC への DNS クエリを解決するために使用できるインバウンドエンドポイントオプションを提供します。
- OpenSearch Serverless が作成して VPC にアタッチするプライベートホストゾーンは サービスによって管理されますが、 Amazon Route 53 リソースに表示され、アカウントに請求されます。
- その他の考慮事項については、「AWS PrivateLink ガイド」の「考慮事項」を参照してください。

必要なアクセス許可

OpenSearch Serverless の VPC アクセスでは、次の AWS Identity and Access Management (IAM) アクセス許可を使用します。IAM 条件を指定して、ユーザーを特定のコレクションに制限できます。

- `aoss:CreateVpcEndpoint` – VPC エンドポイントを作成します。
- `aoss:ListVpcEndpoints` – すべての VPC エンドポイントを一覧表示します。
- `aoss:BatchGetVpcEndpoint` – VPC エンドポイントのサブセットに関する詳細を参照してください。
- `aoss:UpdateVpcEndpoint` – VPC エンドポイントを変更します。
- `aoss:DeleteVpcEndpoint` – VPC エンドポイントを削除します。

さらに、VPC エンドポイントを作成するには、以下の Amazon EC2 許可と Route 53 許可が必要です。

- `ec2:CreateTags`
- `ec2:CreateVpcEndpoint`

- ec2:DeleteVpcEndPoints
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcEndpoints
- ec2:DescribeVpcs
- ec2:ModifyVpcEndPoint
- route53:AssociateVPCWithHostedZone
- route53:ChangeResourceRecordSets
- route53:CreateHostedZone
- route53:DeleteHostedZone
- route53:GetChange
- route53:GetHostedZone
- route53:ListHostedZonesByName
- route53:ListHostedZonesByVPC
- route53:ListResourceRecordSets

OpenSearch Serverless のインターフェイスエンドポイントを作成する

Serverless のインターフェイスエンドポイントは、 コンソールまたは OpenSearch Serverless API OpenSearch を使用して作成できます。

OpenSearch Serverless コレクションのインターフェイスエンドポイントを作成するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインで [Serverless] (サーバーレス) を展開し、[VPC endpoints] (VPC エンドポイント) を選択します。
3. [Create VPC endpoint] (VPC エンドポイントの作成) を選択します。
4. エンドポイントの名前を入力します。
5. VPC では、 OpenSearch サーバーレスにアクセスする VPC を選択します。
6. サブネット で、 OpenSearch サーバーレスにアクセスするサブネットを 1 つ選択します。

    - エンドポイントの IP アドレスと DNS タイプはサブネットタイプに基づいています

- デュアルスタック: すべてのサブネットに IPv4 と IPv6 の両方のアドレス範囲がある場合
- IPv6: すべてのサブネットが IPv6 のみのサブネットの場合
- IPv4: すべてのサブネットに IPv4 アドレス範囲がある場合

7. [Security groups] (セキュリティグループ) で、エンドポイントネットワークインターフェイスに関連付けるセキュリティグループを選択します。これは、エンドポイントに対して承認するインバウンドトラフィックのポート、プロトコル、およびソースを制限する重要なステップです。セキュリティグループルールで、VPC エンドポイントを使用するリソースが OpenSearch Serverless と通信してエンドポイントネットワークインターフェイスと通信できることを確認してください。
8. [エンドポイントの作成] を選択します。

Serverless API を使用して VPC OpenSearch エンドポイントを作成するには、`CreateVpcEndpoint` コマンドを使用します。

> **Note**
>
> エンドポイントを作成したら、その ID を書き留めます (例: `vpce-050f79086ee71ac05`)。コレクションへのエンドポイントアクセスを提供するには、この ID を 1 つまたは複数のネットワークアクセスポリシーに含める必要があります。

次のステップ: エンドポイントにコレクションへのアクセスを許可する

インターフェイスエンドポイントを作成したら、ネットワークアクセスポリシーを使用して、そのエンドポイントにコレクションへのアクセスを提供する必要があります。詳細については、「the section called “ネットワークアクセス”」を参照してください。

Amazon OpenSearch Serverless の SAML 認証

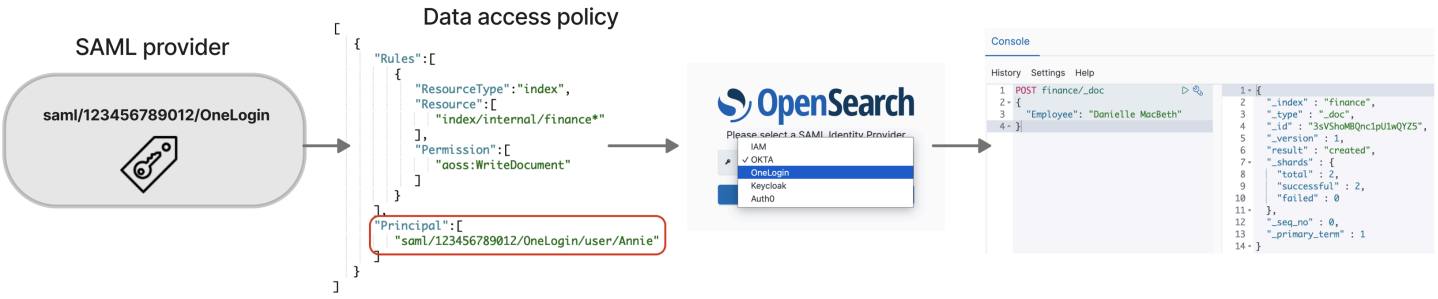
Amazon OpenSearch Serverless の SAML 認証では、既存の ID プロバイダーを使用して、サーバーレスコレクションの OpenSearch Dashboards エンドポイントにシングルサインオン (SSO) を提供できます。

SAML 認証では、サードパーティーの ID プロバイダーを使用して OpenSearch Dashboards にサインインし、データのインデックス作成と検索を行うことができます。 OpenSearch Serverless は、IAM Identity Center、Okta、Keycloak、Active Directory フェデレーションサービス (AD FS)、Auth0 など、SAML 2.0 標準を使用するプロバイダーをサポートします。Okta や Microsoft

Entra ID などの他の ID ソースのユーザーとグループを同期するように OneLoginIAM Identity Center を設定できます。IAM Identity Center でサポートされている ID ソースのリストと設定手順については、「IAM Identity Center ユーザーガイド」の「入門チュートリアル」を参照してください。

> **Note**
>
> SAML 認証は、ウェブブラウザから OpenSearch Dashboards にアクセスするためだけに使用されます。認証されたユーザーは、 OpenSearch Dashboards の開発ツールを介してのみ OpenSearch API オペレーションへのリクエストを行うことができます。SAML 認証情報では、 OpenSearch API オペレーションに直接 HTTP リクエストを行うことはできません。

SAML 認証の使用を開始するには、まず SAML の ID プロバイダー (IdP) を設定します。次に、その IdP から 1 人以上のユーザーを、データアクセスポリシーに含めます。このポリシーにより、コレクションやインデックスに対する特定のアクセス許可が付与されます。その後、ユーザーは OpenSearch Dashboards にサインインし、データアクセスポリシーで許可されているアクションを実行できます。



トピック

- 考慮事項
- 必要なアクセス許可
- SAML プロバイダーの作成 (コンソール)
- OpenSearch Dashboards へのアクセス
- SAML ID に対するコレクションデータへのアクセス権の付与
- SAML プロバイダの作成 (AWS CLI)
- SAML プロバイダーの表示
- SAML プロバイダの更新
- SAML プロバイダーの削除

考慮事項

SAML 認証を設定する際には、以下を考慮します。

- 署名済みおよび暗号化されたリクエストはサポートされていません。
- 暗号化されたアサーションはサポートされていません。
- 認証とサインアウトを IdP が開始することはサポートされていません。

必要なアクセス許可

OpenSearch Serverless の SAML 認証では、次の AWS Identity and Access Management (IAM) アクセス許可を使用します。

- `aoss:CreateSecurityConfig` – SAML プロバイダーを作成します。
- `aoss:ListSecurityConfig` – 現在のアカウントのすべての SAM Lプロバイダーを一覧表示します。
- `aoss:GetSecurityConfig` – SAML プロバイダーの情報を表示します。
- `aoss:UpdateSecurityConfig` – 特定の SAML プロバイダーの (XML メタデータを含む) 設定を変更します。
- `aoss:DeleteSecurityConfig` – SAML プロバイダーを削除します。

次の ID ベースのアクセスポリシーでは、すべての IdP 設定を管理することを、ユーザーに対し許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
```

```
    ]
}
```

Resource 要素はワイルドカードにする必要がある点に、注意してください。

SAML プロバイダーの作成 (コンソール)

以下の手順では、SAML プロバイダーを作成する方法について説明します。これにより、OpenSearch Dashboards のサービスプロバイダー (SP) による SAML 認証が可能になります。認証を IdP が開始することはサポートされていません。

OpenSearch Dashboards の SAML 認証を有効にするには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールにサインインします。
2. 左側のナビゲーションパネルで [Serverless] (サーバーレス) を展開し、[SAML authentication] (SAML 認証) を選択します。
3. [Add SAML provider] (SAML プロバイダーを追加) を選択します。
4. プロバイダーの名前と説明を入力します。

   > **Note**
   >
   > 指定した名前はパブリックアクセス可能で、ユーザーが OpenSearch Dashboards にサインインするとドロップダウンメニューに表示されます。名前は認識しやすく、また、ID プロバイダーに関する機密情報が明らかにならないものにしてください。

5. [Configure your IdP] (IDP の設定) で、アサーションコンシューマーサービス (ACS) のURL をコピーします。
6. ここでコピーした ACS URL は、ID プロバイダーを設定するために使用します。用語と手順はプロバイダーによって異なります。プロバイダーのドキュメントを参照してください。

   例えば、Okta では「SAML 2.0 ウェブアプリケーション」を作成し、ACS URL を、[Single Sign On URL] (シングルサインオン URL)、[Recipient URL] (受信者 URL)、[Destination URL] (送信先 URL) として指定します。Auth0 の場合は、この情報を [Allowed Callback URLs] (許可するコールバック URL) の中で指定します。
7. IdP にオーディエンス制限用のフィールドがある場合は、オーディエンス制限を設定します。オーディエンス制限は SAML アサーション内の値であり、これによりアサーションの対象

者を指定します。 OpenSearch サーバーレスには、 を指定しますaws:opensearch:<aws account id>。例えば aws:opensearch:*123456789012* です。

オーディエンス制限フィールドの名前は、プロバイダーによって異なります。Okta の場合は、[Audience URI (SP Entity ID)] (オーディエンス URI (SP エンティティ ID)) です。IAM ID センターでは、[Application SAML audience] (アプリケーション SAML オーディエンス) になります。

8. IAM ID センターを使用している場合は、属性マッピング (unspecified の形式を使用する Subject=${user:name}) も指定する必要があります。
9. ID プロバイダーを設定すると、IdP メタデータファイルが生成されます。この XML ファイルには、TLS 証明書、Single Sign-On エンドポイント、ID プロバイダーのエンティティ ID など、プロバイダーに関する情報が含まれています。

   IdP メタデータファイル内のテキストをコピーして、[Provide metadata from your IdP] (IdP からメタデータを提供) フィールドに貼り付けます。または、[XML ファイルからインポート] を選択し、ファイルをアップロードします。メタデータファイルは、次のように表示されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>s
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST" Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

10. ユーザーネームに対する SAML アサーションの NameID 要素を使用するために、[Custom user ID attribute] (カスタムユーザー ID 属性) フィールドは空のままにしておきます。アサーションでこの標準エレメントを使用せず、代わりにユーザーネームをカスタム属性として含める場合は、ここでその属性を指定します。属性では、大文字と小文字が区別されます。シングルユーザー属性のみがサポートされています。

    次の例では、SAML アサーション内の、NameID に対するオーバーライド属性を示しています。

    ```
    <saml2:Attribute Name="UserId" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
    format:basic">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:string">annie</saml2:AttributeValue>
    </saml2:Attribute>
    ```

11. (オプション) [Group attribute] (グループ属性) フィールドで、カスタム属性 (role または groupなど) を指定します。グループ属性は、 1 つだけがサポートされます。グループ属性には、デフォルト値はありません。これを指定しない場合、データアクセスポリシーにはユーザープリンシパルのみが含まれます。

    次に、SAML アサーションのグループ属性での例を示します。

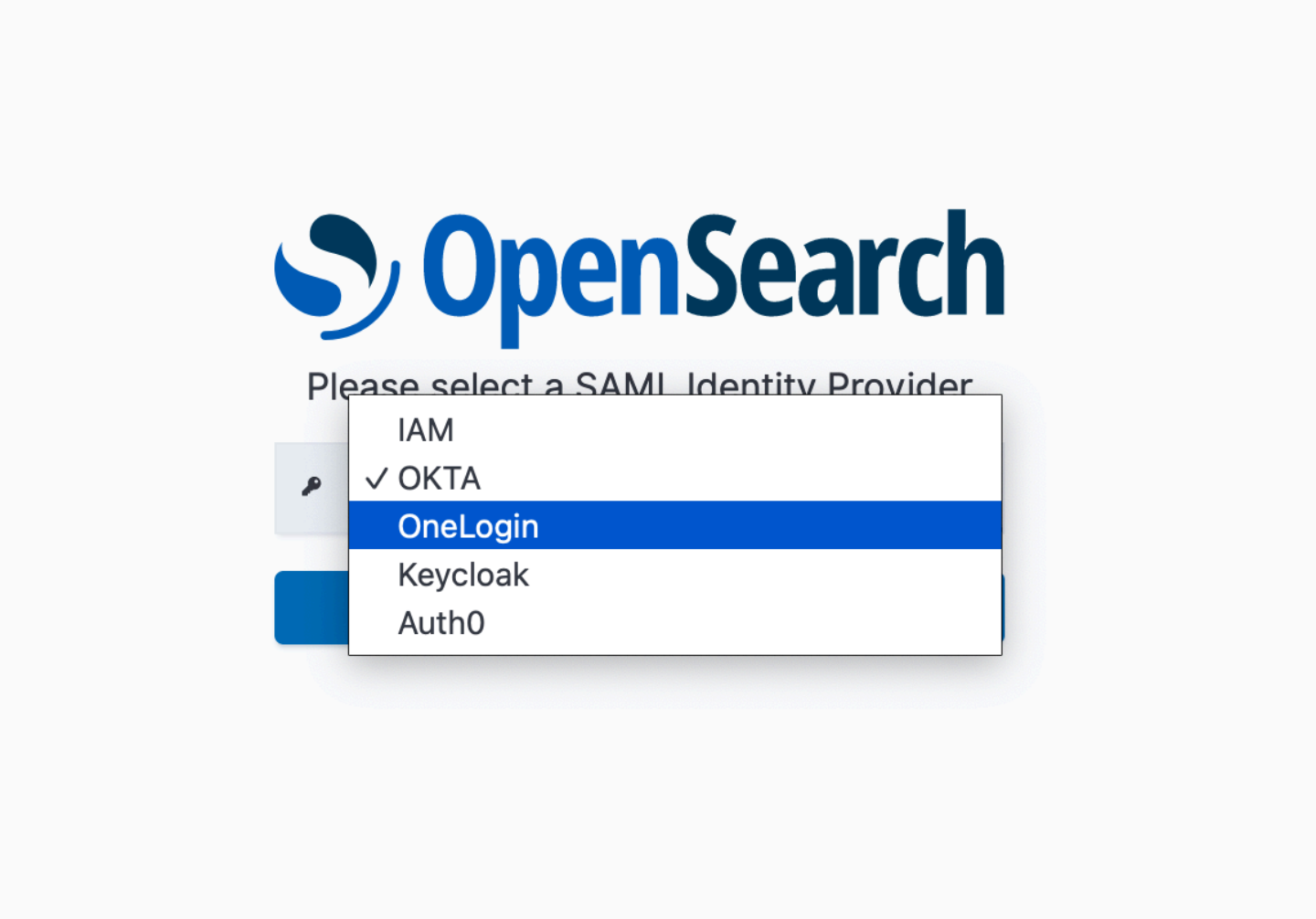
    ```
    <saml2:Attribute Name="department"
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">finance</saml2:AttributeValue>
    </saml2:Attribute>
    ```

12. デフォルトでは、 OpenSearch Dashboards は 24 時間後にユーザーをサインアウトします。OpenSearch Dashboards のタイムアウト を指定することで、この値を 1 ~ 12 時間 (15 ~ 720 分) の任意の数値に設定できます。タイムアウトを 15 分以下に設定しようとすると、セッションは 1 時間にリセットされます。
13. [Create SAML provider] (SAML プロバイダーを作成) を選択します。

OpenSearch Dashboards へのアクセス

SAML プロバイダーを設定すると、そのプロバイダーに関連付けられているすべてのユーザーとグループは Dashboards OpenSearch エンドポイントに移動できます。Dashboards URL は、すべてのコレクションで *collection-endpoint*/_dashboards/ の形式になっています。

SAML が有効になっている場合は、 のリンクを選択すると IdP 選択ページ AWS Management Console が表示され、SAML 認証情報を使用してサインインできます。まず、ドロップダウンから ID プロバイダーを選択します。



次に、自分の IdP 認証情報を使用してサインインします。

SAML が有効になっていない場合は、 のリンクを選択すると、SAML のオプションなしで IAM ユーザーまたはロールとしてログインするように AWS Management Console 指示されます。

SAML ID に対するコレクションデータへのアクセス権の付与

SAML プロバイダーを作成した後も、基盤となるユーザーとグループに対して、コレクション内のデータへのアクセス権を付与する必要があります。アクセス権は、データアクセスポリシーを介して付与します。アクセスが付与されるまで、ユーザーはコレクション内のデータの読み取り、書き込み、削除を行うことはできません。

アクセスを許可するには、データアクセスポリシーを作成し、Principal ステートメントの中で、SAML ユーザーおよび/またはグループ ID を指定します。

```
[
   {
      "Rules":[
       ...
      ],
      "Principal":[
         "saml/987654321098/myprovider/user/Shaheen",
         "saml/987654321098/myprovider/group/finance"
      ]
   }
]
```

アクセス権は、コレクション、インデックス、またはその両方に付与できます。ユーザーごとに異なる権限を持たせたい場合は、複数のルールを作成します。使用可能な許可のリストについては、「Supported policy permissions」(サポートされるポリシーの許可) を参照してください。アクセスポリシーの形式の詳細については、「Policy syntax」(ポリシーの構文) を参照してください。

SAML プロバイダの作成 (AWS CLI)

OpenSearch Serverless API を使用して SAML プロバイダーを作成するには、CreateSecurityConfigリクエストを送信します。

```
aws opensearchserverless create-security-config \
    --name myprovider \
    --type saml \
    --saml-options file://saml-auth0.json
```

.json ファイル内のキーと値のマップとして (メタデータ XML を含めながら) saml-options を指定します。メタデータ XML は、JSON エスケープ文字列としてエンコードする必要があります。

```
{
   "sessionTimeout": 70,
   "groupAttribute": "department",
   "userAttribute": "userid",
   "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>"
}
```

SAML プロバイダーの表示

次のListSecurityConfigsリクエストは、アカウント内のすべての SAML プロバイダーを一覧表示します。

```
aws opensearchserverless list-security-configs --type saml
```

このリクエストでは、既存のすべての SAML プロバイダーに関する (ID プロバイダーが生成する完全な IdP メタデータを含む) 情報を返します。

```
{
   "securityConfigDetails": [
      {
         "configVersion": "MTY2NDA1MjY4NDQ5M18x",
         "createdDate": 1664054180858,
         "description": "Example SAML provider",
         "id": "saml/123456789012/myprovider",
         "lastModifiedDate": 1664054180858,
         "samlOptions": {
            "groupAttribute": "department",
            "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>",
            "sessionTimeout": 120,
            "userAttribute": "userid"
         }
      }
   ]
}
```

将来の更新ににおける `configVersion` を含め、特定のプロバイダの詳細を表示するには、`GetSecurityConfig` リクエストを送信します。

SAML プロバイダの更新

OpenSearch Serverless コンソールを使用して SAML プロバイダーを更新するには、SAML 認証 を選択し、ID プロバイダーを選択し、 の編集 を選択します。メタデータやカスタム属性を含め、すべてのフィールドを変更可能です。

OpenSearch Serverless API を使用してプロバイダーを更新するには、 UpdateSecurityConfigリクエストを送信し、更新するポリシーの識別子を含めます。また、設定のバージョン

(ListSecurityConfigs または GetSecurityConfig コマンドで取得可能) も含める必要があります。最新バージョンを含めると、他のユーザーが行った変更を不注意に上書きしてしまうことを防げます。

次のリクエストは、プロバイダーの SAML オプションを更新します。

```
aws opensearchserverless update-security-config \
    --id saml/123456789012/myprovider \
    --type saml \
    --saml-options file://saml-auth0.json \
    --config-version MTY2NDA1MjY4NDQ5M18x
```

SAML 設定オプションは、.json ファイル内のキーと値のマップとして指定します。

> **Important**
>
> SAML オプションの更新を段階的に実行することはできません。更新時に、SAMLOptions オブジェクト内でパラメータの値を指定しない場合、既存の値は空の値で上書きされます。例えば、現在の構成で userAttribute に値が指定されていて、この値を含めずに更新を行った場合、その値は構成から削除されます。GetSecurityConfig オペレーション呼び出しにより更新を実行する前に、既存の値が何であるかを確認してください。

SAML プロバイダーの削除

SAML プロバイダーを削除した場合、データアクセスポリシー内で関連付けられたユーザーやグループへの参照は、以後、機能しなくなります。混乱を避けるため、エンドポイントを削除する前に、アクセスポリシー内で (指定している) エンドポイントへの参照を、すべて削除することをお勧めします。

OpenSearch Serverless コンソールを使用して SAML プロバイダーを削除するには、認証 を選択し、プロバイダーを選択して、削除 を選択します。

OpenSearch Serverless API を使用してプロバイダーを削除するには、 DeleteSecurityConfigリクエストを送信します。

```
aws opensearchserverless delete-security-config --id saml/123456789012/myprovider
```

Amazon OpenSearch Serverless のコンプライアンス検証

サードパーティーの監査者は、複数の コンプライアンスプログラムの一環として Amazon OpenSearch Serverless のセキュリティと AWS コンプライアンスを評価します。このプログラムには、SOC、PCI、HIPAA が含まれます。

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。 では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャ — このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべての AWS のサービス が HIPAA 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 OpenSearch Serverless コレクション

タグを使用すると、Amazon OpenSearch Serverless コレクションに任意の情報を割り当てて、その情報を分類してフィルタリングできます。タグとは、ユーザーまたは AWS が AWS リソースに割り当てるメタデータラベルです。

各タグは、キーと値から構成されます。ユーザーが割り当てるタグでは、ユーザーがキーと値を定義します。たとえば、1 つのリソースのキーを `stage` と定義し、値を `test` と定義します。

タグを使用すると、次のことを実行できます。

- AWS リソースの特定と整理。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 OpenSearch Service ドメインに割り当てる OpenSearch Serverless コレクションに同じタグを割り当てることができます。
- AWS のコストの追跡。これらのタグは、AWS Billing and Cost Management ダッシュボードで有効にします。AWS では、タグを使用してコストを分類し、毎月のコスト配分レポートを提供します。詳細については、「AWS Billing ユーザーガイド」の「Use Cost Allocation Tags」(コスト配分タグの使用) を参照してください。

OpenSearch Serverless では、プライマリリソースはコレクションです。OpenSearch Service コンソール、AWS CLI、OpenSearch Serverless API オペレーション、または AWS SDK を使用して、コレクションのタグを追加、管理、および削除できます。

必要な許可

OpenSearch Serverless は、コレクションのタグ付けに次の AWS Identity and Access Management Access Analyzer (IAM) アクセス許可を使用します。

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

タグの操作 (コンソール)

コンソールは、コレクションにタグを付けるうえで最も簡単な方法です。

タグを作成するには (コンソール)

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) にサインインします。
2. 左側のナビゲーションペインで [Serverless] (サーバーレス) を展開し、[Collections] (コレクション) を選択します。
3. タグを追加するコレクションを選択し、[Tags] (タグ) タブに移動します。
4. [管理] を選択して、[新しいタグを追加] を選択します。
5. タグキーとオプションの値を入力します。
6. [Save (保存)] を選択します。

タグを削除するには、同じ手順に従って、[タグを管理] ページで [削除] を選択します。

タグを操作するコンソールを使用する方法の詳細については、AWS マネジメントコンソール入門ガイドの「タグエディター」を参照してください。

タグの操作 (AWS CLI)

AWS CLI を使用してコレクションにタグを付けるには、TagResource リクエストを送信します。

```
aws opensearchserverless tag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tags Key=service,Value=aoss Key=source,Value=logs
```

ListTagsForResource コマンドを使用して、コレクションの既存のタグを表示します。

```
aws opensearchserverless list-tags-for-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
```

UntagResource コマンドを使用してコレクションのタグを削除します。

```
aws opensearchserverless untag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tag-keys service
```

Amazon OpenSearch Serverless でサポートされているオペレーションとプラグイン

Amazon OpenSearch Serverless は、さまざまな OpenSearch プラグインと、 で利用可能なインデックス作成、検索、メタデータ API オペレーションのサブセットをサポートしています OpenSearch。データアクセスポリシー内にある表で、左側の列にアクセス許可を含めることで、アクセス権限を特定のオペレーションに制限できます。

トピック

- サポートされている OpenSearch API オペレーションとアクセス許可
- サポートされている OpenSearch プラグイン

サポートされている OpenSearch API オペレーションとアクセス許可

次の表に、 OpenSearch Serverless がサポートする API オペレーションと、対応するデータアクセスポリシーのアクセス許可を示します。

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:CreateIndex`	PUT <index>	インデックスを作成します。詳細については、

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
		「Create index」(インデックスの作成) を参照してください。 **Note** このアクセス許可は、 OpenSearch Dashboards のサンプルデータを使用したインデックスの作成にも適用されます。
`aoss:DescribeIndex`	• GET <index> • GET <index>/_mapping • GET <index>/_mappings • GET <index>/_setting • GET <index>/_setting/<setting> • GET <index>/_settings • GET <index>/_settings/<setting> • GET _cat/indices • GET _mapping • GET _mappings • GET _resolve/index/<index> • HEAD <index>	インデックスの詳細を表示します。詳細については、以下のリソースを参照してください。 • Get index (インデックスを取得する) • Get a mapping (マッピングを取得する) • Get settings (設定を取得する) • インデックスが存在する • CAT インデックス (応答には `health`または `status`フィールドは含まれません)。

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:WriteDocument`	• DELETE <index>/_doc/<id> • POST <index>/_bulk • POST <index>/_create/<id> (検索コレクションタイプのみ) • POST <index>/_doc • POST <index>/_update/<id> (検索コレクションタイプのみ) • POST _bulk • PUT <index>/_create/<id> (検索コレクションタイプのみ) • PUT <index>/_doc/<id> (検索コレクションタイプのみ)	ドキュメントを作成および更新する。詳細については、以下のリソースを参照してください。 • 一括 • データインデックス **Note** 一部の操作は、タイプ SEARCH のコレクションに対してのみ利用可能です。詳細については、「the section called “コレクションタイプを選択する”」を参照してください。

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:ReadDocument`	• GET <index>/_analyze • GET <index>/_doc/<id> • GET <index>/_explain/<id> • GET <index>/_mget • GET <index>/_source/<id> • GET <index>/_count • GET <index>/_field_caps • GET <index>/_msearch • GET <index>/_rank_eval • GET <index>/_search • GET <index>/_validate/<query> • GET _analyze • GET _field_caps • GET _mget • GET _search • HEAD <index>/_doc/<id> • HEAD <index>/_source/<id> • POST <index>/_analyze • POST <index>/_explain/<id> • POST <index>/_count • POST <index>/_field_caps • POST <index>/_rank_eval • POST <index>/_search • POST _analyze • POST _field_caps • POST _search	ドキュメントを読み取ります。詳細については、以下のリソースを参照してください。 • Perform text analysis (テキスト分析を実行する) • Get document (ドキュメントを取得する) • Count (カウント) • Query DSL (DSL をクエリする) • Ranking evaluation (ランキングの評価) • Analyze API (API を分析する) • Explain (説明する)

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:DeleteIndex`	DELETE <target>	インデックスを削除します。詳細については、「Delete index」(インデックスを削除する) を参照してください。
`aoss:UpdateIndex`	• POST _mapping • POST <index>/_mapping/ • POST <index>/_mappings/ • POST <index>/_setting • POST <index>/_settings • POST _setting • POST _settings • PUT _mapping • PUT <index>/_mapping • PUT <index>/_mappings/ • PUT <index>/_setting • PUT <index>/_settings • PUT _setting • PUT _settings	インデックス設定を更新します。詳細については、以下のリソースを参照してください。 • Mapping (マッピング) • Update settings (設定を更新する)
`aoss:CreateCollectionItems`	POST _aliases	インデックスエイリアスを作成します。エイリアスの詳細については、「Create aliases」(エイリアスを作成する) を参照してください。

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:DescribeCollectionItems`	• GET <index>/_alias/<alias> • GET _alias • GET _alias/<alias> • GET _cat/aliases • GET _cat/templates • GET _cat/templates/<template_name> • GET _component_template • GET _component_template/<component-template> • GET _index_template • GET _index_template/<index-template> • HEAD _alias/<alias> • HEAD _component_template/<component-template> • HEAD _index_template/<name> • HEAD <index>/_alias/<alias>	エイリアスとインデックステンプレートの詳細を表示します。詳細については、以下のリソースを参照してください。 • Manage aliases (エイリアスを管理する) • Index templates (インデックステンプレート)

データアクセスポリシーでの許可	OpenSearch API オペレーション	説明と注意事項
`aoss:UpdateCollectionItems`	• POST <index>/_alias/<alias> • POST <index>/_aliases/<alias> • POST _component_template/<component-template> • POST _index_template/<index-template> • PUT <index>/_alias/<alias> • PUT <index>/_aliases/<alias> • PUT _component_template/<component-template> • PUT _index_template/<index-template>	エイリアスとインデックステンプレートを更新します。詳細については、以下のリソースを参照してください。 • Index aliases (インデックスエイリアス) • Index templates (インデックステンプレート)
`aoss:DeleteCollectionItems`	• DELETE <index>/_alias/<alias> • DELETE _component_template/<component-template> • DELETE _index_template/<index-template> • DELETE <index>/_aliases/<alias>	エイリアスとインデックステンプレートを削除します。詳細については、以下のリソースを参照してください。 • Delete aliases (エイリアスを削除する) • Delete a template (テンプレートを削除する)

サポートされている OpenSearch プラグイン

OpenSearch サーバーレスコレクションには、 コミュニティの以下のプラグインが OpenSearchあらかじめパッケージ化されています。Serverless は、自動的にプラグインのデプロイと管理を実行します。

分析プラグイン

- ICU Analysis

- Japanese (kuromoji) Analysis
- Korean (Nori) Analysis
- Phonetic Analysis
- Smart Chinese Analysis
- Stempel Polish Analysis
- Ukrainian Analysis

マッパープラグイン

- Mapper Size
- Mapper Murmur3
- Mapper Annotated Text

スクリプトプラグイン

- Painless
- Expression
- Mustache

さらに、 OpenSearch Serverless には、モジュールとして出荷されるすべてのプラグインが含まれます。

Amazon OpenSearch Serverless のモニタリング

モニタリングは、Amazon OpenSearch Serverless およびその他の AWS ソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。 は、 OpenSearch サーバーレスを監視したり、問題が発生したときに報告したり、必要に応じて自動アクションを実行したりするために、以下のモニタリングツール AWS を提供しています。

- Amazon CloudWatch は、 AWS リソースと、 AWS で実行しているアプリケーションをリアルタイムでモニタリングします。メトリクスを収集および追跡し、カスタマイズされたダッシュボードを作成し、指定されたメトリックが指定したしきい値に達したときに通知またはアクションを実行するアラームを設定できます。

例えば、 で Amazon EC2 インスタンスの CPU 使用率やその他のメトリクス CloudWatch を追跡し、必要に応じて新しいインスタンスを自動的に起動できます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

- AWS CloudTrail は、 AWS アカウントによって行われた、またはそのアカウントに代わって実行された API コールと関連イベントをキャプチャします。このツールでは、ユーザーが指定した Amazon S3 バケットにログファイルを配信します。を呼び出したユーザーとアカウント AWS、呼び出し元のソース IP アドレス、および呼び出し日時を特定できます。詳細については、『AWS CloudTrail ユーザーガイド』を参照してください。
- Amazon EventBridge は、 OpenSearch サービスドメインの変更を示すシステムイベントのストリームをほぼリアルタイムで配信します。特定のイベントを監視し、これらのイベントが発生した AWS のサービス ときに他の で自動アクションをトリガーするルールを作成できます。詳細については、「Amazon ユーザーガイド EventBridge 」を参照してください。

Amazon による OpenSearch Serverless のモニタリング CloudWatch

を使用して Amazon OpenSearch Serverless をモニタリングすることで CloudWatch、raw データを収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をより的確に把握できます。

また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

OpenSearch Serverless は、 `AWS/AOSS`名前空間で以下のメトリクスを報告します。

メトリクス	説明
`ActiveCollection`	コレクションが有効化されているかどうかを示します。値が 1 の場合は、コレクションのステータスが `ACTIVE` であることを意味します。この値は、コレクションが正しく作成された時点で出力され、そのコレクションを削除するまで 1 を保ちます。このメトリクスでは、値が 0 になることはありません。 関連する統計情報: Max

メトリクス	説明
	ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
DeletedDocuments	削除されたドキュメントの総数。 関連する統計情報: Average、Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒
IndexingOCU	コレクションデータの取り込みに使用される OpenSearch コンピューティングユニット (OCUsの数。このメトリクスは、アカウントレベルで適用されます。 関連する統計情報: Sum ディメンション: ClientId 頻度: 60 秒
IngestionDataRate	コレクションまたはインデックスに対するインデックス化レート (GiB /秒)。このメトリクスは、一括インデックス作成リクエストにのみ適用されます。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒

メトリクス	説明
IngestionDocumentErrors	コレクションまたはインデックスに対する取り込み中に発生した、ドキュメントエラーの総数。一括インデックス作成リクエストが成功すると、ライターはそのリクエストを処理し、リクエスト内で失敗したすべてのドキュメントについてエラーを出力します。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒
IngestionDocumentRate	コレクションまたはインデックスへの、ドキュメントの取り込みにおける 1 秒あたりのレート。このメトリクスは、一括インデックス作成リクエストにのみ適用されます。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒
IngestionRequestErrors	コレクションに対する一括インデックス作成リクエストエラーの合計数。 OpenSearch サーバーレスは、認証や可用性の問題など、何らかの理由で一括インデックス作成リクエストが失敗した場合に、このメトリクスを発行します。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒

メトリクス	説明
IngestionRequestLatency	コレクションへの一括書き込みオペレーションにおけるレイテンシー (秒単位)。 関連する統計情報: Minimum、Maximum、Average ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
IngestionRequestRate	コレクションが受信した、一括書き込みオペレーションの総数。 関連する統計情報: Minimum、Maximum、Average ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
IngestionRequestSuccess	コレクションに対して成功した、インデックス作成オペレーションの総数。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
SearchableDocuments	コレクションまたはインデックス内にある、検索可能なドキュメントの合計数。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒

メトリクス	説明
SearchRequestErrors	コレクションに対し発生したクエリエラーの、1 分あたりの総数。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
SearchRequestLatency	コレクションに対し、検索オペレーションを完了するために必要な平均時間 (ミリ秒)。 関連する統計情報: Minimum、Maximum、Average ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒
SearchOCU	コレクションデータの検索に使用される OpenSearch コンピューティングユニット (OCUsの数。このメトリクスは、アカウントレベルで適用されます。 関連する統計情報: Sum ディメンション: ClientId 頻度: 60 秒
SearchRequestRate	コレクションに対する検索リクエストの、1 分あたりの総数。 関連する統計情報: Average、Maximum、Sum ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒

メトリクス	説明
StorageUsedInS3	使用された Amazon S3 ストレージのバイト単位の量。OpenSearch Serverless はインデックス化されたデータを Amazon S3 に保存します。正確な値を取得するには、1 分単位で期間を選択します。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 、IndexId、IndexName 頻度: 60 秒
2xx, 3xx, 4xx, 5xx	コレクションに対するリクエストのうち、特定の HTTP レスポンスコード (2xx、3xx、4xx、5xx) で応答したものの数。 関連する統計情報: Sum ディメンション: ClientId、CollectionId 、CollectionName 頻度: 60 秒

を使用した OpenSearch Serverless API コールのログ記録 AWS CloudTrail

Amazon OpenSearch Serverless は、サーバーレスのユーザー AWS CloudTrail、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。

CloudTrail は、 OpenSearch Serverless のすべての API コールをイベントとしてキャプチャします。キャプチャされる呼び出しには、 OpenSearch サービスコンソールのサーバーレスセクションからの呼び出しと、 OpenSearch サーバーレス API オペレーションへのコード呼び出しが含まれます。

証跡を作成する場合は、 OpenSearch Serverless 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。証跡を設定しない場合でも、イベント履歴 で CloudTrail コンソールで最新の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、 OpenSearch サーバーレス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。

OpenSearch のサーバーレス情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 は で有効になります。 OpenSearch Serverless 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。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。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。

OpenSearch Serverless 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きに、証跡がすべての 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 ログで収集されたデータをより詳細に分析し、それに基づく対応を行うように他の 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
- の Amazon SNS 通知の設定 CloudTrail
- 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

すべての OpenSearch Serverless アクションは によってログに記録 CloudTrail され、OpenSearch「 Serverless API リファレンス」に記載されています。例えば、`CreateCollection`、、および `DeleteCollection`アクションを呼び出すと`ListCollections`、 CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。この ID 情報は以下のことを確認するのに役立ちます。

- 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。

OpenSearch Serverless 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要求を表します。これには、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。

次の例は、 `CreateCollection`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":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/test-user",
      "accountId":"123456789012",
      "accessKeyId":"access-key",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:role/Admin",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "webIdFederationData":{

         },
         "attributes":{
            "creationDate":"2022-04-08T14:11:34Z",
            "mfaAuthenticated":"false"
         }
      }
```

```
   },
   "eventTime":"2022-04-08T14:11:49Z",
   "eventSource":"aoss.amazonaws.com",
   "eventName":"CreateCollection",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-cli/2.1.30 Python/3.8.8 Linux/5.4.176-103.347.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/aoss.create-collection",
   "errorCode":"HttpFailureException",
   "errorMessage":"An unknown error occurred",
   "requestParameters":{
      "accountId":"123456789012",
      "name":"test-collection",
      "description":"A sample collection",
      "clientToken":"d3a227d2-a2a7-49a6-8fb2-e5c8303c0718"
   },
   "responseElements": null,
   "requestID":"12345678-1234-1234-1234-987654321098",
   "eventID":"12345678-1234-1234-1234-987654321098",
   "readOnly":false,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management",
   "tlsDetails":{
      "clientProvidedHostHeader":"user.aoss-sample.us-east-1.amazonaws.com"
   }
}
```


Amazon を使用した OpenSearch サーバーレスイベントのモニタリング EventBridge

Amazon OpenSearch Service は Amazon と統合 EventBridge され、ドメインに影響する特定のイベントを通知します。 AWS サービスからのイベントは、ほぼリアルタイムで EventBridge に配信されます。同じイベントは、Amazon の前身である Amazon CloudWatch Events にも送信されます EventBridge。ルールを記述して、注目するイベントと、イベントがルールに一致した場合に自動的に実行するアクションを指定できます。自動的にアクティブ化できるアクションの例には、以下が含まれます。

- AWS Lambda 関数の呼び出し
- Amazon EC2 Run Command の呼び出し

- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティブ化
- Amazon SNS トピックまたは Amazon SQS キューの通知

詳細については、「Amazon ユーザーガイド」の EventBridge「Amazon の開始方法 EventBridge 」を参照してください。

通知の設定

AWS ユーザー通知を使用して、 OpenSearch サーバーレスイベントが発生したときに通知を受け取ることができます。イベントは、OCU 使用量の上限に達したときなど、 OpenSearch サーバーレス環境の変化を示す指標です。 はイベント Amazon EventBridge を受け取り、通知センターと選択した配信チャネルに通知を AWS Management Console ルーティングします。指定したルールにイベントが一致すると、通知を受け取ります。

OpenSearch コンピューティングユニット (OCU) イベント

OpenSearch Serverless は、次の OCU 関連のイベントのいずれかが発生した EventBridge ときに、にイベントを送信します。

OCU の使用量が上限に近づいている

OpenSearch Serverless は、検索またはインデックス OCU の使用量が容量制限の 75% に達すると、このイベントを送信します。OCU の使用量は、設定した容量制限と現在の OCU 消費量に基づいて計算されます。

例

このタイプのイベント (検索 OCU) の例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
```

```
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
}
```

このタイプのイベント (インデックス OCU) の例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
```

OCU の使用量が上限に達した

OpenSearch Serverless は、検索またはインデックス OCU の使用量が容量制限の 100% に達すると、このイベントを送信します。OCU の使用量は、設定した容量制限と現在の OCU 消費量に基づいて計算されます。

例

このタイプのイベント (検索 OCU) の例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
```

```
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage has reached the configured maximum limit."
  }
}
```

このタイプのイベント (インデックス OCU) の例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage has reached the configured maximum limit."
  }
}
```

Amazon OpenSearch Service ドメインの作成と管理

この章では、Amazon OpenSearch Service ドメインを作成および管理する方法を説明します。ドメインは、 AWSがプロビジョニングしたオープンソース OpenSearch クラスターと同等です。ドメインを作成するときは、その設定、インスタンスタイプ、インスタンス数、ストレージ割り当てを指定します。オープンソースクラスターの詳細については、 OpenSearch ドキュメントの「クラスターの作成」を参照してください。

開始方法チュートリアルの簡単な手順とは異なり、この章ではすべてのオプションについて説明し、関連するリファレンス情報を提供します。 OpenSearch サービスコンソール、 (AWS CLI)、 AWS Command Line Interface または AWS SDKsの手順を使用して、各手順を完了できます。

OpenSearch サービスドメインの作成

このセクションでは、 OpenSearch サービスコンソールを使用するか、 AWS CLI `create-domain` コマンドで を使用して OpenSearch サービスドメインを作成する方法について説明します。

OpenSearch サービスドメインの作成 (コンソール)

コンソールを使用して OpenSearch サービスドメインを作成するには、次の手順に従います。

OpenSearch サービスドメインを作成するには (コンソール)

1. https://aws.amazon.com にアクセスし、[Sign In to the Console] (コンソールにサインイン) を選択します。
2. Analytics で、Amazon OpenSearch Service を選択します。
3. [ドメインの作成] を選択します。
4. [ドメイン名] には、ドメイン名を入力します。名前は次の基準を満たしている必要があります。
   - アカウントと に固有 AWS リージョン
   - 先頭が小文字
   - 3 ~ 28 文字
   - 小文字の a ~ z、0 ~ 9 の数字、ハイフン (-) のみ含まれる
5. ドメインの作成方法は、[標準作成] を選択します。
6. [テンプレート] で、ドメインの目的に最も一致するオプションを選択します。

- 高可用性とパフォーマンスを必要とするワークロード向けの [本番] ドメイン。これらのドメインは、可用性を高めるために、マルチ AZ (スタンバイの有無にかかわらず) と専用マスターノードを使用します。
- 開発またはテスト向けの [開発/テスト]。これらのドメインは、マルチ AZ (スタンバイの有無にかかわらず) または単一のアベイラビリティーゾーンを使用できます。

> **Important**
>
> デプロイタイプが異なると、後続のページのオプションが異なります。これらのステップにはすべてのオプションが含まれています。

7. [デプロイオプション] で、[スタンバイが有効のドメイン] を選択して 3-AZ ドメインを設定し、いずれかのゾーンのノードがスタンバイとして予約されます。このオプションでは、指定されたデータノード数、マスターノード数、インスタンスタイプ、レプリカ数、ソフトウェアアップデート設定など、いくつかのベストプラクティスが適用されます。
8. バージョン で、使用する OpenSearch またはレガシー Elasticsearch OSS のバージョンを選択します。の最新バージョンを選択することをお勧めします OpenSearch。詳細については、「the section called “サポートバージョン”」を参照してください。

   (オプション) ドメインに OpenSearch バージョンを選択した場合は、互換性モードを有効にしてバージョンを 7.10 としてレポートするように OpenSearch選択します。これにより、接続前にバージョンをチェックする特定の Elasticsearch OSS クライアントとプラグインがサービスで作業を続けることができます。
9. [インスタンスタイプ] では、データノードのインスタンスタイプを選択します。詳細については、「the section called “サポートされるインスタンスタイプ”」を参照してください。

   > **Note**
   >
   > すべてのアベイラビリティーゾーンですべてのインスタンスタイプがサポートされているわけではありません。スタンバイ付きまたはスタンバイなしのマルチ AZ を選択する場合は、R5 や I3 などの現行世代のインスタンスタイプを選択することをお勧めします。
10. [ノードの数] で、データノードの数を選択します。

    最大値については、OpenSearch 「サービスドメインとインスタンスのクォータ」を参照してください。単一ノードのクラスターは、開発とテスト用に適切ですが、本稼働のワークロードには

使用しないでください。ガイダンスについては、「the section called “ドメインのサイジング”」および「the section called “マルチ AZ ドメインの設定”」を参照してください。

11. [ストレージタイプ] で、[Amazon EBS] を選択します。一覧で利用できるボリュームタイプは、選択したインスタンスタイプに応じて異なります。特に大きなドメインを作成する際のガイダンスについては、「the section called “ペタバイトスケール”」を参照してください。
12. [EBS] ストレージでは、次の追加設定を行います。選択したボリュームのタイプによっては、一部の設定が表示されない場合があります。

設定	説明
EBS ボリュームタイプ	汎用 (SSD) - gp3 および 汎用 (SSD) - gp2、または前世代の プロビジョンド IOPS (SSD) および マグネティック (標準) から選択します。
ノードあたりの EBS ストレージサイズ	各データノードに接続する EBS ボリュームのサイズを入力します。 [EBS ボリュームサイズ] はノードあたりのサイズです。データノードの数に EBS ボリュームサイズを掛けることで、OpenSearch サービスドメインの合計クラスターサイズを計算できます。EBS ボリュームの最小サイズと最大サイズは、指定された EBS ボリュームタイプとそれがアタッチされるインスタンスタイプの両方によって異なります。詳細については、「EBS ボリュームサイズの制限」を参照してください。
プロビジョンド IOPS	Provisioned IOPS SSD ボリュームタイプを選択した場合は、ボリュームがサポートできる I/O オペレーション/秒 (IOPS) を入力します。

13. （オプション) gp3ボリュームタイプを選択した場合は、アドバンスト設定を展開し、ストレージの料金に含まれるものを超える追加の IOPS (データノードごとにプロビジョニングされる 3 TiB ボリュームサイズごとに最大 16,000 個) とスループット (データノードごとにプロビジョニングされる 3 TiB ボリュームサイズごとに最大 1,000 MiB /秒) を追加料金で指定します。 TiB 詳細については、「Amazon OpenSearch Service の料金」を参照してください。
14. （オプション) UltraWarm ストレージ を有効にするには、 UltraWarm データノードを有効にする を選択します。各インスタンスタイプには、アドレス可能なストレージの最大容量がありま

す。この量に、アドレス可能なウォームストレージの合計のウォームデータノードの数を乗算します。

15. (オプション) [コールドストレージ] を有効にするには、[コールドストレージを有効にする] を選択します。コールドストレージを有効にする UltraWarm には、 を有効にする必要があります。
16. スタンバイ付きマルチ AZ を使用する場合、3 つの専用マスターノードが既に有効になっています。必要なマスターノードのタイプを選択します。スタンバイなしのマルチ AZ のドメインを選択した場合は、[専用マスターノードを有効にする] を選択し、必要なマスターノードのタイプと数を選択します。専用マスターノードは、クラスターの安定性を高めます。また、インスタンス数が 10 を超えるドメインに必要です。本番稼働用ドメインには、3 つの専用マスターノードをお勧めします。

> **Note**
>
> 専用マスターノードおよびデータノードの異なるインスタンスタイプを選択できます。たとえば、データノードの汎用またはストレージ最適化インスタンスを選択できますが、専用マスターノードのコンピューティング最適化インスタンスは選択できません。

17. (オプション) OpenSearch または Elasticsearch 5.3 以降を実行しているドメインの場合、スナップショット設定は関係ありません。自動化されたスナップショットの詳細については、「the section called “インデックススナップショットの作成”」を参照してください。
18. 標準エンドポイントの `https://search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com` ではなく、カスタムエンドポイントを使用する場合、[カスタムエンドポイントを有効にする] を選択し、名前と証明書を入力します。詳細については、「the section called “カスタムエンドポイントの作成”」を参照してください。
19. [ネットワーク] セクションで、[VPC アクセス] または [パブリックアクセス] を選択します。[パブリックアクセス] を選択した場合は、次のステップに進みます。[VPC access] (VPC アクセス) を選択する場合は、[prerequisites] (前提条件) を満たしていることを確認してから、次の設定を行います。

設定	説明
VPC	使用する仮想プライベートクラウド (VPC) の ID を選択します。VPC とドメインは同じ にある必要があり AWS リージョン、テナンシーがデフォルト に設定されている VPC を選択する必要があります。OpenSearch サービスでは、専用テナンシーを使用する VPCsはまだサポートされていません。

設定	説明
サブネット	サブネットを選択します。マルチ AZ を有効にした場合は、2 つまたは 3 つのサブネットを選択する必要があります。 OpenSearch サービスは VPC エンドポイントと Elastic Network Interface をサブネットに配置します。 サブネット内のネットワークインターフェイス用に十分な IP アドレスを予約する必要があります。詳細については、「VPC サブネットで IP アドレスをリザーブする」を参照してください。
セキュリティグループ	必要なアプリケーションが OpenSearch ドメインによって公開されているポート (80 または 443) とプロトコル (HTTP または HTTPS) のサービスドメインに到達できるようにする 1 つ以上の VPC セキュリティグループを選択します。詳細については、「the section called “VPC サポート”」を参照してください。
IAM ロール	デフォルトのロールを保持します。 OpenSearch サービスは、この事前定義されたロール (サービスにリンクされたロール とも呼ばれます) を使用して VPC にアクセスし、VPC エンドポイントとネットワークインターフェイスを VPC のサブネットに配置します。詳細については、「VPC アクセス用のサービスにリンクされたロール」を参照してください。
IP アドレスタイプ	IP アドレスタイプとして、デュアルスタックまたは IPv4 を選択します。デュアルスタックでは、IPv4 と IPv6 のアドレスタイプ間でドメインリソースを共有できます。これが推奨オプションです。IP アドレスタイプをデュアルスタックに設定した場合、後でアドレスタイプを変更することはできません。

20. きめ細かなアクセスコントロールを有効または無効にします。

- ユーザー管理に IAM を使用する場合は、[IAM ARN をマスターユーザーとして設定] を選択し、IAM ロールの ARN を指定します。
- 内部ユーザーデータベースを使用する場合は、[Create master user] (マスターユーザーの作成) を選択し、ユーザー名とパスワードを指定します。

どのオプションを選択しても、マスターユーザーはクラスター内のすべてのインデックスとすべての OpenSearch APIsにアクセスできます。選択するオプションのガイダンスについては、「the section called “主要なコンセプト”」を参照してください。

きめ細かなアクセスコントロールを無効にしても、ドメインを VPC 内に配置するか、制限付きアクセスポリシーを適用するか、またはその両方を行うことで、ドメインへのアクセスをコントロールできます。きめ細かなアクセスコントロールを使用するには、保管時の node-to-node 暗号化と暗号化を有効にする必要があります。

> **Note**
>
> ドメイン上のデータを保護するために、きめ細かなアクセスコントロールを有効にすることを強くお勧めします。きめ細かなアクセスコントロールにより、クラスター、インデックス、ドキュメント、フィールドの各レベルでセキュリティが提供されます。

21. （オプション) OpenSearch Dashboards に SAML 認証を使用する場合は、SAML 認証を有効にするを選択し、ドメインの SAML オプションを設定します。手順については、「the section called “ OpenSearch Dashboards の SAML 認証”」を参照してください。
22. （オプション) OpenSearch Dashboards に Amazon Cognito 認証を使用する場合は、Amazon Cognito 認証を有効にする を選択します。次に、 OpenSearch Dashboards 認証に使用する Amazon Cognito ユーザープールと ID プールを選択します。これらのリソースの作成のガイダンスについては、「the section called “OpenSearch Dashboards の Amazon Cognito 認証”」を参照してください。
23. [アクセスポリシー] で、アクセスポリシーを選択するか、独自のポリシーを設定します。カスタムポリシーを作成することを選択した場合は、自分で設定することも、別のドメインからインポートすることもできます。詳細については、「the section called “Identity and Access Management”」を参照してください。

> **Note**
>
> VPC アクセスを有効にした場合、IP ベースのポリシーは使用できません。代わりに、どの IP アドレスがドメインにアクセスできるかを制御するセキュリティグループを使用できます。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

24. (オプション) ドメインへのすべてのリクエストが HTTPS 経由で到着することを要求するには、[ドメインへのすべてのトラフィックに HTTPS が必要] を選択します。 node-to-node 暗号化を有効にするには、Node-to-node 個の暗号化 を選択します。詳細については、「the section called “N ode-to-node 暗号化”」を参照してください。保管中のデータの暗号化を有効にするには、[保管時のデータの暗号化を有効にする] を選択します。スタンバイ付きマルチ AZ デプロイオプションを選択した場合、これらのオプションは既に選択されています。
25. （ オプション） AWS 所有キーを使用する を選択して、 OpenSearch サービスがユーザーに代わって AWS KMS 暗号化キーを作成する (または作成済みのキーを使用する) ようにします。それ以外の場合は、独自の KMS キーを選択します。詳細については、「the section called “保管中の暗号化”」を参照してください。
26. [オフピークウィンドウ] で、ブルー/グリーンデプロイを必要とするサービスソフトウェアの更新と自動調整の最適化をスケジュールする開始時刻を選択します。オフピークの時間帯に更新を行うことにより、トラフィックの多い時間帯にクラスターの専用マスターノードにかかる負荷を、最小限に抑えることができます。
27. Auto-Tune では、速度と安定性を向上させるために OpenSearch 、サービスがドメインにメモリ関連の設定変更を提案できるようにするかどうかを選択します。詳細については、「the section called “Auto-Tune”」を参照してください。

    (オプション) [オフピークウィンドウ] を選択すると、自動調整によってドメインが更新される定期的なウィンドウがスケジュールされます。
28. (オプション) [自動ソフトウェア更新] を選択して、自動ソフトウェア更新を有効にします。
29. (オプション) ドメインを説明するタグを追加して、その情報を分類およびフィルタリングできるようにします。詳細については、「the section called “ドメインのタグ付け”」を参照してください。
30. (オプション) クラスターの 詳細設定 を展開して設定します。これらのオプションの概要については、「the section called “高度なクラスター設定”」を参照してください。
31. [作成] を選択します。

OpenSearch サービスドメインの作成 (AWS CLI)

コンソールを使用して OpenSearch サービスドメインを作成する代わりに、 を使用できます AWS CLI。構文については、 AWS CLI コマンドリファレンスの「Amazon OpenSearch Service」を参照してください。

コマンド例

この最初の例は、次の OpenSearch サービスドメイン設定を示しています。

- OpenSearch バージョン 1.2 で mylogs という名前 OpenSearch のサービスドメインを作成します。
- r6g.large.search インスタンスタイプの 2 つのインスタンスをドメインに追加する
- 各データノードのストレージに 100 GiB 汎用 (SSD) gp3 EBS ボリュームを使用する
- 単一の IP アドレス 192.0.2.0/32 からのみ匿名アクセスを許可する

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.2 \
    --cluster-config  InstanceType=r6g.large.search,InstanceCount=2 \
    --ebs-options
 EBSEnabled=true,VolumeType=gp3,VolumeSize=100,Iops=3500,Throughput=125 \
    --access-policies '{"Version": "2012-10-17", "Statement": [{"Action": "es:*",
 "Principal":"*","Effect": "Allow", "Condition": {"IpAddress":{"aws:SourceIp":
["192.0.2.0/32"]}}}]}'
```

次の例は、次の OpenSearch サービスドメイン設定を示しています。

- Elasticsearch バージョン 7.10 で mylogs という名前 OpenSearch のサービスドメインを作成します。
- r6g.large.search インスタンスタイプの 6 つのインスタンスをドメインに追加する
- 各データノードのストレージに 100 GiB 汎用 (SSD) gp2 EBS ボリュームを使用する
- サービスへのアクセスを、ユーザーの AWS アカウント ID で識別される 1 人のユーザーに制限します: 555555555555
- 3 つのアベイラビリティーゾーンへのインスタンスを分散する

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version Elasticsearch_7.10 \
    --cluster-config
 InstanceType=r6g.large.search,InstanceCount=6,ZoneAwarenessEnabled=true,ZoneAwarenessConfig={A
 \
    --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
```

```
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
"Principal": {"AWS": "arn:aws:iam::555555555555:root" }, "Action":"es:*", "Resource":
"arn:aws:es:us-east-1:555555555555:domain/mylogs/*" } ] }'
```

次の例は、次の OpenSearch サービスドメイン設定を示しています。

- OpenSearch バージョン 1.0 で mylogs という名前 OpenSearch のサービスドメインを作成します。
- r6g.xlarge.search インスタンスタイプの 10 個のインスタンスをドメインに追加する
- 専用マスターノードとして機能する r6g.large.search インスタンスタイプの 3 つのインスタンスをドメインに追加する
- ストレージに 100 GiB プロビジョンド IOPS EBS ボリュームを使用し、各データノードに 1000 IOPS のベースラインパフォーマンスを設定する
- アクセスを単一のユーザーと単一のサブリソース _search API に制限する

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.0 \
    --cluster-config
 InstanceType=r6g.xlarge.search,InstanceCount=10,DedicatedMasterEnabled=true,DedicatedMasterTyp
 \
    --ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": { "AWS": "arn:aws:iam::555555555555:root" }, "Action": "es:*",
 "Resource": "arn:aws:es:us-east-1:555555555555:domain/mylogs/_search" } ] }'
```

> **Note**
>
> OpenSearch サービスドメインを作成しようとしたときに、同じ名前のドメインが既に存在する場合、 CLI はエラーを報告しません。その代わりに、既存ドメインの詳細が返されます。

OpenSearch サービスドメインの作成 (AWS SDKs)

AWS SDKs (Android および iOS SDKsを除く) は、 を含む Amazon OpenSearch Service API リファレンス で定義されているすべてのアクションをサポートしますCreateDomain。サンプルコー

ドについては、「the section called “AWS SDK の使用”」を参照してください。 AWS SDKsAWS「 Software Development Kits」を参照してください。

OpenSearch サービスドメインの作成 (AWS CloudFormation)

OpenSearch サービスは と統合されています。これは AWS CloudFormation、 AWS リソースとインフラストラクチャの作成と管理に費やす時間を短縮できるように、リソースのモデル化とセットアップに役立つサービスです。作成する OpenSearch ドメインを記述するテンプレートを作成し、ドメインを CloudFormation プロビジョニングして設定します。 OpenSearch ドメインの JSON テンプレートと YAML テンプレートの例を含む詳細については、「 ユーザーガイド」の「Amazon OpenSearch Service リソースタイプのリファレンスAWS CloudFormation 」を参照してください。

アクセスポリシーの設定

Amazon OpenSearch Service には、 OpenSearch サービスドメインへのアクセスを設定する方法がいくつか用意されています。詳細については、「the section called “Identity and Access Management”」および「the section called “きめ細かなアクセスコントロール”」を参照してください。

コンソールは、ドメインの特定のニーズに対応してカスタマイズできる事前定義のアクセスポリシーを提供します。他の OpenSearch サービスドメインからアクセスポリシーをインポートすることもできます。上記のアクセスポリシーが VPC アクセスを操作する方法についての詳細は、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

アクセスポリシーを設定するには (コンソール)

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。
2. Analytics で、Amazon OpenSearch Service を選択します。
3. ナビゲーションペインの [ドメイン] で、更新するドメインを選択します。
4. [アクション] から [セキュリティ設定の編集] を選択します。
5. アクセスポリシーの JSON を編集するか、事前設定済みのオプションをインポートします。
6. [変更を保存] を選択します。

高度なクラスター設定

詳細オプションを使用して、次のように設定します。

リクエストボディのインデックス

HTTP リクエストボディ内で、インデックスへの明示参照を許可するかどうかを指定します。このプロパティを `false` に設定すると、ユーザーがサブリソースのアクセスコントロールをバイパスできなくなります。デフォルトでは、値は `true` に設定されます。詳細については、「the section called “詳細オプションと API に関する考慮事項”」を参照してください。

フィールドデータのキャッシュ割り当て

フィールドデータに割り当てられる Java ヒープスペースの割合を指定します。デフォルトでは、この設定は JVM ヒープの 20% です。

> **Note**
>
> 多くのお客様が、ローテーションするインデックスのクエリを毎日実行しています。`indices.fielddata.cache.size` を使用してベンチマークテストを始めることをお勧めします。これらのほとんどのユースケースでは JVM ヒープを 40% に設定してください。非常に大きいインデックスでは、さらに大きいフィールドデータキャッシュが必要になることがあります。

句の最大数

Lucene のブールクエリで許可される句の最大数を指定します。デフォルト値は 1,024 です。クエリに含まれる句の数が最大数を超えていると、`TooManyClauses` エラーが発生します。詳細については、Lucene のドキュメントを参照してください。

Amazon OpenSearch Service での設定変更

Amazon OpenSearch Service は、ドメインを更新するときにブルー/グリーンデプロイプロセスを使用します。ブルー/グリーンデプロイでは、本番環境をコピーするドメイン更新用のアイドル環境が作成され、更新が完了した後にユーザーが新しい環境にルーティングされます。ブルー/グリーンデプロイでは、ブルー環境が現在の本稼働環境です。グリーン環境はアイドル環境です。

データはブルー環境からグリーン環境に移行されます。新しい環境の準備が整うと、 OpenSearch サービスは環境を切り替えて、グリーン環境を新しい本稼働環境に昇格させます。データ損失なしでスイッチオーバーが行われます。この方法では、新しい環境へのデプロイに失敗しても、ダウンタイムを最小限に抑えることができ、元の環境を維持することができます。

トピック

通常は blue/green デプロイの原因となる変更

次のオペレーションにより、Blue/Green デプロイが発生します。

- インスタンスタイプを変更する
- きめ細かなアクセスコントロールの有効化
- サービスソフトウェアのアップデートを行う
- 専用マスターノードを有効または無効にする
- スタンバイなしのマルチ AZ を有効または無効にする
- ストレージタイプ、ボリュームタイプ、またはボリュームサイズの変更
- 別の VPC のサブネットを選択する
- VPC セキュリティグループを追加または削除する
- OpenSearch Dashboards の Amazon Cognito 認証の有効化または無効化
- 別の Amazon Cognito ユーザープールまたは ID プールを選択する
- アドバンスト設定を変更する
- 新しい OpenSearch バージョンへのアップグレード (アップグレードの一部または全部で OpenSearch ダッシュボードが使用できない場合があります)
- 保管中のデータの暗号化または node-to-node 暗号化の有効化
- またはコールドストレージの有効化 UltraWarm または無効化
- Auto-Tune を無効にし、変更をロールバックする
- ドメインへのオプションプラグインの関連付けとドメインからのオプションプラグインの関連付けの解除
- 2 つの専用マスターノードを持つマルチ AZ ドメインの専用マスターノード数を増やす

- EBS ボリュームサイズの縮小
- 最後に行った変更が進行中であるか、6 時間以内に発生した場合は、EBS ボリュームサイズ、IOPS、またはスループットを変更する
- 監査ログの への発行を有効にします CloudWatch。

スタンバイが有効のマルチ AZ では、1 回につき 1 つの変更リクエストしか行えません。変更がすでに進行中の場合、新しいリクエストは拒否されます。現在の変更ステータスは、`DescribeDomainChangeProgress` API を使用して確認できます。

通常は blue/green デプロイが発生しない変更

ほとんどの場合、次のオペレーションにより、Blue/Green デプロイは発生しません。

- アクセスポリシーの変更
- カスタムエンドポイントの変更
- Transport Layer Security (TLS) ポリシーを変更する
- 自動スナップショットの時間を変更する
- [HTTPS が必要] を有効または無効にする
- 変更内容をロールバックせずに Auto-Tune を有効または無効にする
- ドメインに専用マスターノードがある場合は、データノードまたは UltraWarm ノード数を変更します。
- ドメインに専用マスターノードがある場合は、専用マスターインスタンスのタイプまたは数を変更します (2 つの専用マスターノードを持つマルチ AZ ドメインを除く)。
- へのエラーログまたはスローログの発行を有効または無効にする CloudWatch
- への監査ログの発行を無効にする CloudWatch
- データノードあたりのボリュームサイズを最大 3 TiB に増やし、ボリュームタイプ、IOPS、スループットを変更する
- タグの追加と削除

> **Note**
>
> サービスソフトウェアのバージョンによっては、一部の例外があります。変更によってブルー/グリーンデプロイが発生しないようにするには、このオプションが使用可能な場合は、ドメインを更新する前にドライランを実行します。一部の変更では、ドライランオプション

が提供されません。通常、ピークトラフィック時間外にクラスターを変更することをお勧めします。

変更によってブルー/グリーンデプロイが実行されるかどうかを判断する

一部のタイプの計画された設定変更をテストして、ブルー/グリーンデプロイを引き起こすかどうかを判断できます。これらの変更にコミットする必要はありません。設定の変更を開始する前に、コンソールまたは API を使用して検証チェックを実行し、ドメインが更新の対象であることを確認してください。

Console

設定の変更を検証するには

1. で Amazon OpenSearch Service コンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。
3. 設定の変更を実行するドメインを選択します。選択すると、ドメインの詳細ページが開きます。[Actions] (アクション) ドロップダウンメニューを選択してから、[Edit cluster configuration] (クラスター設定の編集) を選択します。
4. [Edit cluster configuration] (クラスター設定の編集) ページでは、インスタンスタイプ、ノードの数、およびその他の設定を変更できます。概要パネルで変更を確認したら、[Run] (実行) を選択します。
5. ドライランが完了したら、その結果がドライラン ID とともに、ページの最下部に自動表示されます。これらの結果から、変更がどのカテゴリに該当するかがわかります。

   - ブルー/グリーンデプロイが開始される
   - ブルー/グリーンデプロイは必要なし
   - 変更を保存する前に対処する必要がある検証エラーが含まれている

   各ドライランは、その直前のドライランを上書きすることに注意してください。後ほど各ドライランの詳細を調べるため、ドライラン ID を保存しておくようにしてください。各ドライランは 90 日間、または設定を更新するまで利用できます。
6. 設定の更新を続行するには、[Save changes] (変更の保存) を選択します。それ以外の場合は、[Cancel] (キャンセル) を選択します。どちらのオプションを選択しても、[Cluster

configuration] (クラスター設定) タブに戻ります。このタブで [Dry run details] (ドライランの詳細) を選択すると、最新のドライランの詳細が確認できます。このページには、ドライラン前の設定とドライラン設定 side-by-side の比較も含まれています。

API

設定 API を使用してドライラン検証を実行できます。API を使用して変更をテストするには、`DryRun` を `true`、`DryRunMode` を `Verbose` に設定します。Verbose モードは、変更によってブルー/グリーンデプロイが開始されるかどうかの判断に加えて、検証チェックも実行します。例えば、このUpdateDomainConfigリクエストは、 を有効にした結果生じるデプロイタイプをテストします UltraWarm。

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
config
{
   "ClusterConfig": {
    "WarmCount": 3,
    "WarmEnabled": true,
    "WarmType": "ultrawarm1.large.search"
   },
   "DryRun": true,
   "DryRunMode": "Verbose"
}
```

リクエストは検証チェックを実行し、変更に起因するデプロイのタイプを返しますが、実際の更新は行いません。

```
{
   "ClusterConfig": {
     ...
    },
   "DryRunResults": {
      "DeploymentType": "Blue/Green",
      "Message": "This change will require a blue/green deployment."
    }
}
```

考えられるデプロイタイプは、次のとおりです。

- `Blue/Green` – 変更によって、ブルー/グリーンデプロイが発生します。

- DynamicUpdate – 変更によって、ブルー/グリーンデプロイは発生しません。
- Undetermined – ドメインがまだ処理状態であるため、デプロイのタイプを特定できません。
- None – 設定変更はありません。

検証が失敗すると、検証失敗のリストが返されます。

```
{
   "ClusterConfig":{
      "..."
   },
   "DryRunProgressStatus":{
      "CreationDate":"2023-01-12T01:14:33.847Z",
      "DryRunId":"db00ca39-48b2-4774-bbd3-252cf094d205",
      "DryRunStatus":"failed",
      "UpdateDate":"2023-01-12T01:14:33.847Z",
      "ValidationFailures":[
         {
            "Code":"Cluster.Index.WriteBlock",
            "Message":"Cluster has index write blocks."
         }
      ]
   }
}
```

ステータスが のままの場合はpending、以降のDescribeDryRunProgress呼び出しで UpdateDomainConfig レスポンスのドライラン ID を使用して、検証のステータスを確認できます。

```
GET https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
dryRun?dryRunId=my-dry-run-id
{
    "DryRunConfig": null,
    "DryRunProgressStatus": {
        "CreationDate": "2023-01-12T01:14:42.998Z",
        "DryRunId": "db00ca39-48b2-4774-bbd3-252cf094d205",
        "DryRunStatus": "succeeded",
        "UpdateDate": "2023-01-12T01:14:49.334Z",
        "ValidationFailures": null
    },
    "DryRunResults": {
        "DeploymentType": "Blue/Green",
```

```
        "Message": "This change will require a blue/green deployment."
    }
}
```

検証チェックなしでドライラン分析を実行するには、設定 API を使用するときに DryRunMode を Basic に設定します。

Python

次の Python コードは UpdateDomainConfig API を使用してドライラン検証チェックを実行し、チェックが成功すると、ドライランなしで同じ API を呼び出して更新を開始します。チェックが失敗した場合、スクリプトはエラーを出力して停止します。

```
import time
import boto3

client = boto3.client('opensearch')

response = client.UpdateDomainConfig(
    ClusterConfig={
        'WarmCount': 3,
        'WarmEnabled': True,
        'WarmCount': 123,
    },
    DomainName='test-domain',
    DryRun=True,
    DryRunMode='Verbose'
)

dry_run_id = response.DryRunProgressStatus.DryRunId

retry_count = 0

while True:

    if retry_count == 5:
        print('An error occured')
        break

    dry_run_progress_response = client.DescribeDryRunProgress('test-domain',
 dry_run_id)
    dry_run_status = dry_run_progress_response.DryRunProgressStatus.DryRunStatus
```

```
    if dry_run_status == 'succeeded':
        client.UpdateDomainConfig(
            ClusterConfig={
            'WarmCount': 3,
            'WarmEnabled': True,
            'WarmCount': 123,
        })
        break

    elif dry_run_status == 'failed':
        validation_failures_list =
dry_run_progress_response.DryRunProgressStatus.ValidationFailures
        for item in validation_failures_list:
            print(f"Code: {item['Code']}, Message: {item['Message']}")
        break

    retry_count += 1
    time.sleep(30)
```

設定変更の開始と追跡

> **Note**
>
> 一度に 1 つの設定変更をリクエストできます。1 つのリクエストで複数の設定変更をグループ化することもできます。追加の設定変更をリクエストActiveする前に、ドメインのステータスが になるのを待ちます。

Amazon OpenSearch Service コンソールでドメイン処理ステータスフィールドと Config 変更ステータスフィールドを表示して、ドメインと設定の変更を追跡できます。API レスポンスの および ConfigChangeStatusパラメータを使用して、ドメインDomainProcessingStatusと設定の変更を追跡することもできます。詳細については、 OpenSearch 「サービス API リファレンス」のDomainStatus「 データ型」を参照してください。

ドメイン処理ステータスの可視性: コンソールのドメイン処理ステータスフィールドを確認することで、ドメインの設定ステータスを簡単に判断できます。同様に、 DomainProcessingStatus API パラメータを使用してステータスを識別できます。次の値は、ドメインの処理ステータスです。

- Active: 設定の変更は進行中ではありません。新しい設定変更リクエストを送信できます。
- Creating: ドメインを作成中です。

- `Modifying`: 新しいデータノード、EBS、gp3、IOPS プロビジョニングの追加、KMS キーの設定などの設定変更が進行中です。

> **Note**
>
> ドメインModifyingが設定変更を完了するためにシャードの移動を必要とする状況では、ステータスが と表示されることがあります。下位互換性のために、 Processingパラメータの動作は API レスポンスでは変更されず、コア設定の変更が完了するとすぐに、シャード移動の完了を待たずに false に設定されます。

- `Upgrading Engine Version`: エンジンバージョンのアップグレードが進行中です。
- `Updating Service Software`: サービスソフトウェアの更新が進行中です。
- `Deleting`: ドメインは削除中です。
- `Isolated`: ドメインは停止されています。

設定ステータスの可視性: 設定の変更は、オペレータ (新しいデータノードの追加、インスタンスタイプの変更など) またはサービス (Auto-Tune やオフピーク時の更新など) によって開始できます。最新の設定変更の詳細のステータスは、Amazon OpenSearch Service コンソールの設定変更ステータスフィールドと `ConfigChangeStatus` API レスポンスで確認できます。次の値は、ドメインの設定ステータスを示します。

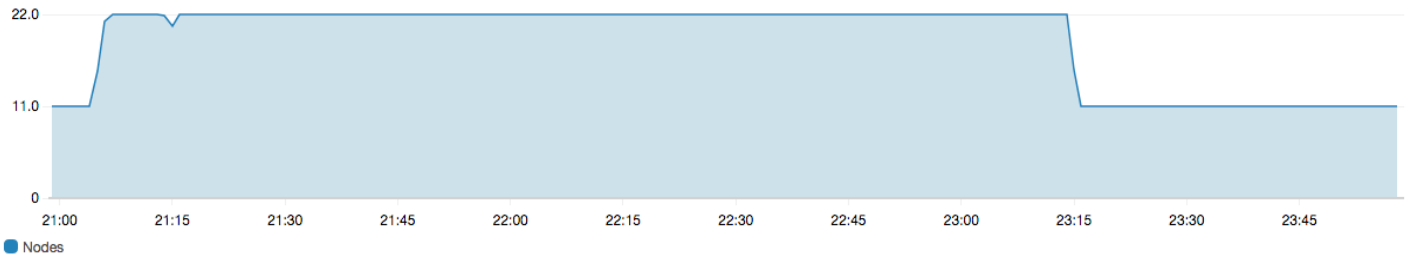
- `Pending`: 設定変更リクエストが送信されました。
- `Initializing`: サービスは設定変更リクエストを初期化しています。
- `Validating`: サービスは、要求された変更と必要なリソースを検証しています。
- `Awaiting user inputs`: オペレータがインスタンスタイプの変更など、いくつかの設定変更が続行されると予想される場合に適用されます。設定の変更を編集できます。
- `Applying changes`: サービスはリクエストされた設定変更を適用しています。
- `Cancelled`: 設定の変更がキャンセルされました。検証の失敗ステータスが表示された場合は、コンソールでキャンセルをクリックするか、 API オペレーションを呼び出すことができます。CancelDomainConfigChangeこれを行うと、適用されたすべての変更がロールバックされます。
- `Completed`: リクエストされた設定の変更は正常に完了しました。
- `Validation Failed`: リクエストされた変更は検証に失敗しました。設定の変更は適用されません。

> **Note**
>
> 検証の失敗は、ドメインに存在する赤いインデックス、選択したインスタンスタイプが使用できない、またはディスク容量不足が原因である可能性があります。検証エラーのリストについては、「」を参照してくださいthe section called “検証エラーのトラブルシューティング”。検証失敗イベント中に、設定の変更をキャンセル、再試行、または編集できます。

API 概要: `DescribeDomain`、、および `DescribeDomainConfig` API オペレーションを使用して`DescribeDomainChangeProgress`、詳細な設定更新ステータスを取得できます。さらに、`CancelDomainConfigChange`を使用して、検証に失敗した場合に更新をキャンセルできます。詳細については、OpenSearch 「Service API ドキュメント」を参照してください。

設定の変更が完了すると、ドメインの状態は に戻ります`Active`。

クラスターのヘルスと Amazon CloudWatch メトリクスを確認し、ドメインの更新中にクラスター内のノード数が一時的に増加し、多くの場合は 2 倍になります。次の図では、設定の変更中にノード数が 11 から 22 に倍増し、更新が完了すると 11 に戻っていることがわかります。



この一時的な増加により、管理対象のノード数がかなり増えるため、クラスターの専用マスターノードへの負荷が高くなります。また、 OpenSearch サービスが古いクラスターから新しいクラスターにデータをコピーするため、検索とインデックス作成のレイテンシーを増やすこともできます。クラスターでは、Blue/Green デプロイに関連するオーバーヘッドを処理するための十分な容量を維持することが重要です。

> **Important**
>
> 設定の変更およびサービスのメンテナンス中に、追加料金は発生しません。課金対象となるのは、クラスター用にリクエストしたノード数のみです。詳細については、「the section called “設定変更に関連する料金”」を参照してください。

専用マスターノードの過負荷を防ぐために、Amazon CloudWatch メトリクス を使用して使用状況をモニタリングできます。推奨される最大値については、「the section called “推奨 CloudWatch アラーム”」を参照してください。

設定変更のステージ

設定変更を開始すると、 OpenSearch Service は一連のステップを実行してドメインを更新します。設定変更の進行状況は、コンソールの設定変更ステータスで確認できます。更新が実行される正確な手順は、変更の種類によって異なります。DescribeDomainChangeProgress API オペレーションを使用して設定変更をモニタリングすることもできます。

設定変更中に更新が実行される可能性のあるステージを次に示します。

ステージ名	説明
検証	ドメインが更新の要件を満たしていることを検証し、必要に応じて検証の問題を明らかにします。
新しい環境の作成	blue/green デプロイを開始するために必要な前提条件を完了し、必要なリソー

ステージ名	説明
	スを作成します。
新しいノードのプロビジョニング	新しい環境での新しいインスタンスのセットの作成。
新しいノードでのトラフィックルーティング	新しく作成されたデータノードにトラフィックをリダイレクトします。
古いノードでのトラフィックルーティング	古いデータノードでトラフィックを無効にする。
削除するノードの準備	ノードを削除する準備をしています。このステップは、ドメインを縮小している場合にのみ発生します（例えば、8 ノードから 6 ノードへ）。

ステージ名	説明
シャードの新しいノードへのコピー	古いノードから新しいノードにシャードを移動します。
ノードの終了	シャードを削除した後、古いノードを終了して削除する。
古いリソースの削除	古い環境 (ロードバランサーなど) に関連付けられているリソースの削除。
動的更新	更新が blue/green デプロイを必要とせず、動的に適用できる場合に表示されます。
専用マスター関連の変更の適用	専用マスターインスタンスのタイプまたは数が変更されたときに表示されます。

ステージ名	説明
ボリューム関連の変更の適用	ボリュームのサイズ、タイプ、IOPS、スループットが変更されたときに表示されます。

ブルー/グリーンデプロイのパフォーマンスへの影響

ブルー/グリーンデプロイ中、Amazon OpenSearch Service クラスターは受信検索リクエストとインデックス作成リクエストに使用できます。ただし、次のようなパフォーマンスの問題が発生する可能性があります。

- クラスターで管理するノードが増えるにつれて、リーダーノードの使用量が一時的に増加します。
- Service が古いノードから新しいノードにデータ OpenSearch をコピーするにつれて、検索とインデックス作成のレイテンシーが増加しました。
- ブルー/グリーンデプロイ中にクラスターの負荷が増加するにつれて、受信リクエストの拒否が増加しました。
- レイテンシーの問題やリクエストの拒否を回避するには、クラスターが正常で、ネットワークトラフィックが少ないときにブルー/グリーンデプロイを実行する必要があります。

設定変更に関連する料金

ドメインの設定を変更すると、 OpenSearch の説明に従ってサービスが新しいクラスターを作成しますthe section called “設定変更”。古いものから新しいものへの移行中、以下の料金が発生します。

- インスタンスタイプを変更した場合、両方のクラスター (最初の 1 時間) が課金されます。最初の 1 時間の後は、新しいクラスターに対してのみ料金が発生します。EBS ボリュームはクラスターの一部であるため、2 回課金されることはありません。そのため、課金はインスタンスの課金に従います。

例: 3 つの m3.xlarge インスタンスから 4 つの m4.large インスタンスに設定を変更した場合。最初の 1 時間は、両方のクラスターに課金されます (3 * m3.xlarge + 4 * m4.large)。最初の 1 時間の後は、新しいクラスターに対してのみ料金が発生します (4 * m4.large)。

- インスタンスタイプを変更しない場合は、最大のクラスターに対してのみ課金されます (最初の 1 時間)。最初の 1 時間の後は、新しいクラスターに対してのみ料金が発生します。

  例: 6 つの m3.xlarge インスタンスから 3 つの m3.xlarge インスタンスに設定を変更した場合。最初の 1 時間は、最大のクラスターに課金されます (6 * m3.xlarge)。最初の 1 時間の後は、新しいクラスターに対してのみ料金が発生します (3 * m3.xlarge)。

検証エラーのトラブルシューティング

設定変更を開始するか、 OpenSearch または Elasticsearch バージョンアップグレードを実行すると、 OpenSearch Service はまず一連の検証チェックを実行して、ドメインが更新の対象であることを確認します。これらのチェックのいずれかが失敗した場合、ドメインを更新する前に修正する必要がある特定の問題を含む通知がコンソールで表示されます。次の表に、 OpenSearch サービスが表示できる可能性のあるドメインの問題と、それらを解決する手順を示します。

問題	エラーコード	トラブルシューティングのステップ
セキュリティグループが見つかりません	SecurityGroupNotFound	OpenSearch サービスドメインに関連付けられたセキュリティグループが存在しません。この問題を解決するには、指定された名前でセキュリティ グループを作成します。
サブネットが見つかりません	SubnetNotFound	OpenSearch サービスドメインに関連付けられたサブネットが存在しません。この問題を解決するには、VPC でサブネットを作成します。
サービスにリンクされたロールが設定されていません	SLRNotConfigured	サービスのサービスにリンクされたロールが設定されていません。OpenSearch サービスにリンクされたロールは OpenSearch 、サービスによって事前定義されており、ユーザーに代わってサービスから他の AWS のサービスを呼び出すために必要なすべてのアクセス許可が含まれています。ロールが存在しない場合は、手動で作成する必要がある場合があります。

問題	エラーコード	トラブルシューティングのステップ
十分な IP アドレスがありません	InsufficientFreeIPsForSubnets	1 つ以上の VPC サブネットに、ドメインを更新するのに十分な IP アドレスがありません。必要な IP アドレスの数を計算するには、「the section called “VPC サブネットで IP アドレスをリザーブする”」を参照してください。
Cognito ユーザープールが存在しません	CognitoUserPoolNotFound	OpenSearch サービスで Amazon Cognito ユーザープールが見つかりません。ユーザープールを作成し、正しい ID が設定されていることを確認します。ID を見つけるには、Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用できます。 `aws cognito-idp list-user-pools --max-results 60 --region us-east-1`
Cognito ID プールが存在しません	CognitoIdentityPoolNotFound	OpenSearch サービスが Cognito ID プールを見つけられません。ユーザープールを作成し、正しい ID が設定されていることを確認します。ID を見つけるには、Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用できます。 `aws cognito-identity list-identity-pools --max-results 60 --region us-east-1`
ユーザープールの Cognito ドメインが見つかりません	CognitoDomainNotFound	ユーザープールにドメイン名がありません。Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用して設定できます。 `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`

問題	エラーコード	トラブルシューティングのステップ
Cognito ロールが設定されていません	CognitoRoleNotConfigured	Amazon Cognito ユーザープールと ID プールを設定し、認証に使用するアクセス許可を OpenSearch サービスに付与する IAM ロールは設定されていません。適切な許可セットと信頼関係を使用してロールを設定します。コンソールを使用してデフォルトのCognitoAccessForAmazonOpenSearchロールを作成することも、 AWS CLI または AWS SDK を使用して手動でロールを設定することもできます。
ユーザープールを記述できません	UserPoolNotDescribable	指定された Amazon Cognito ロールには、ドメインに関連付けられたユーザープールを記述するための許可がありません。ロールの許可ポリシーで cognito-identity:DescribeUserPool アクションが許可されていることを確認してください。完全な許可ポリシーについては、「the section called “CognitoAccessForAmazonOpenSearch ロールについて”」を参照してください。
ID プールを記述できません	IdentityPoolNotDescribable	指定された Amazon Cognito ロールには、ドメインに関連付けられた ID プールを記述するための許可がありません。ロールの許可ポリシーで cognito-identity:DescribeIdentityPool アクションが許可されていることを確認してください。完全な許可ポリシーについては、「the section called “CognitoAccessForAmazonOpenSearch ロールについて”」を参照してください。
ユーザーおよび ID プールを記述できません	CognitoPoolsNotDescribable	指定された Amazon Cognito ロールには、ドメインに関連付けられたユーザーおよび ID プールを記述するための許可がありません。ロールの許可ポリシーで cognito-identity:DescribeIdentityPool および cognito-identity:DescribeUserPool アクションが許可されていることを確認してください。完全な許可ポリシーについては、「the section called “CognitoAccessForAmazonOpenSearch ロールについて”」を参照してください。

問題	エラーコード	トラブルシューティングのステップ
KMS キーが有効になっていません	KMSKeyNotEnabled	ドメインの暗号化に使用される AWS Key Management Service (AWS KMS) キーは無効になっています。すぐにキーを再度有効にします。
カスタム証明書が [ISSUED] (発行済み) の状態ではありません	InvalidCertificate	ドメインがカスタムエンドポイントを使用している場合は、 AWS Certificate Manager (ACM) で SSL 証明書を生成するか、独自の証明書をインポートして保護します。証明書のステータスは [Issued] (発行済み) である必要があります。このエラーが発生した場合は、ACM コンソールで証明書のステータスを確認してください。ステータスが [Expired] (期限切れ)、[Failed] (失敗)、[Inactive] (非アクティブ)、または [Pending validation] (検証の保留中) の場合は、ACM のトラブルシューティングドキュメントを参照して問題を解決してください。
選択したインスタンスタイプを起動するのに十分なキャパシティがありません	InsufficientInstanceCapacity	リクエストされたインスタンスタイプのキャパシティは利用できません。例えば、5 つの`i3.16xlarge.search` ノードをリクエストしたが、 OpenSearch サービスに十分な`i3.16xlarge.search` ホストが利用できないため、リクエストを満たすことができません。 OpenSearch サービスでサポートされているインスタンスタイプを確認し、別のインスタンスタイプを選択します。
クラスター内の赤いインデックス	RedCluster	クラスター内の 1 つ以上のインデックスが赤のステータスになり、全体的に赤のクラスターステータスになります。この問題のトラブルシューティングおよび修復については、「the section called “赤のクラスター状態”」を参照してください。
メモリサーキットブレーカー、リクエストが多すぎます	TooManyRequests	ドメインへの検索リクエストと書き込みリクエストが多すぎるため、 OpenSearch サービスは設定を更新できません。リクエストの数を減らしたり、インスタンスを最大 64 GiB の RAM まで垂直方向にスケールしたり、インスタンスを追加して水平方向にスケールしたりできます。

問題	エラーコード	トラブルシューティングのステップ
新しい設定はデータを保持できません (ディスク容量不足)	InsufficientStorageCapacity	設定されたストレージサイズでは、ドメインのすべてのデータを保持できません。この問題を解決するには、より大きなボリュームを選択するか、未使用のインデックスを削除するか、クラスター内のノードの数を増やしてディスク容量をすぐに解放します。

特定のノードにシャードが固定されています	ShardMovementBlocked	ドメイン内の 1 つ以上のインデックスが特定のノードにアタッチされており、再割り当てできません。これは、シャード割り当てのフィルタリングを設定したために発生する可能性が最も高いです。このフィルタリングは、特定のインデックスのシャードをホストすることを許可するノードを指定できるようにします。
		
		この問題を解決するには、影響を受けるすべてのインデックスからシャード割り当てフィルターを削除します。
		
		`PUT my-index/_settings`
		`{`
		`  "settings": {`
		`  "index.routing.allocation.require._name": null`
		`  }`
		`}`

新しい設定はすべてのシャードを保持できません (シャード数)	TooManyShards	ドメインのシャード数が多すぎるため、 OpenSearch サービスがそれらを新しい設定に移動できません。この問題を解決するには、現在のクラスターノードと同じ設定タイプのノードを追加して、ドメインを水平にスケールします。EBS ボリュームの最大サイズは、ノードのインスタンスタイプによって異なることに注意してください。 今後この問題が発生しないようにするには、「the section called “シャード数の選択”」を参照して、ユースケースに適したシャーディング戦略を定義してください。

問題	エラーコード	トラブルシューティングのステップ
ドメインに関連付けられたサブネットは IPv4 アドレスをサポートしていません	`ResultCodeIPv4BlockNotExists`	この問題を解決するには、ドメインの設定された IP アドレスタイプに応じて、VPC でサブネットを作成するか、既存のサブネットを更新します。ドメインで [IPv4 のみ] のアドレスタイプを使用している場合は、IPv4 のみのサブネットを使用します。ドメインで [デュアルスタックモード] を使用している場合は、デュアルスタックサブネットを使用します。
ドメインに関連付けられたサブネットは IPv6 アドレスをサポートしていません	`ResultCodeIPv6BlockNotExists`	この問題を解決するには、ドメインの設定された IP アドレスタイプに応じて、VPC でサブネットを作成するか、既存のサブネットを更新します。ドメインで [IPv4 のみ] のアドレスタイプを使用している場合は、IPv4 のみのサブネットを使用します。ドメインで [デュアルスタックモード] を使用している場合は、デュアルスタックサブネットを使用します。

Amazon Service での OpenSearch サービスソフトウェアの更新

> **Note**
>
> 各メジャーな (パッチでない) サービスのソフトウェア更新で行われた変更および追加に関する説明については、リリースノートを参照してください。

Amazon OpenSearch Service は、機能を追加したり、ドメインを改善したりするサービスソフトウェア更新を定期的にリリースします。コンソールの [Notifications] (通知) パネルを使用すると、更新がリリースされているかどうかの確認や、更新のステータスのチェックを簡単に行うことができます。各通知には、サービスソフトウェア更新に関する詳細が含まれます。ソフトウェア更新では、ダウンタイムを最小限に抑えるためブルー/グリーンデプロイが使用されます。

サービスソフトウェアの更新は、 OpenSearch バージョンのアップグレードとは異なります。の新しいバージョンへのアップグレードについては OpenSearch、「」を参照してくださいthe section called “ドメインのアップグレード”。

トピック

- オプションの更新と必須の更新
- パッチ更新
- 考慮事項
- サービスのソフトウェア更新を開始する
- オフピークの時間帯にソフトウェア更新をスケジュールする
- サービスソフトウェア更新のモニタリング
- ドメインが更新の対象でない場合

オプションの更新と必須の更新

OpenSearch サービスには、サービスソフトウェア更新の 2 つのカテゴリがあります。

オプションの更新

オプションのサービスソフトウェア更新には、通常、機能強化や新機能のサポートが含まれます。オプションの更新はドメインには適用されず、インストールの厳しい締切もありません。更新の有無は、メールとコンソールの通知でユーザーに通知されます。更新は、すぐに適用することもできれば、別の日時にスケジュール変更することもできます。ドメインのオフピークの時間帯にスケジュールすることも可能です。ソフトウェア更新の大半は、任意です。

更新をスケジュールするかどうかにかかわらず、ブルー/グリーンデプロイが発生するドメインに変更を加えると、 OpenSearch サービスは自動的にサービスソフトウェアを更新します。

自分のドメインで、オフピークの時間帯にオプションの更新を自動適用するように、設定することが可能です。このオプションをオンにすると、 OpenSearch サービスはオプションの更新が利用可能になってから少なくとも 13 日間待機し、72 時間 (3 日) 後に更新をスケジュールします。更新がスケジュールされると、コンソールに通知が届きます。別の日にスケジュール変更することも可能です。

自動ソフトウェア更新を有効にするには、ドメインを作成または更新する際に、[Enable automatic software update] (自動ソフトウェア更新を有効にする) を選択します。を使用して同じ設定を構成す

るには AWS CLI、ドメインを作成または更新`true`するときに `--software-update-options`をに設定します。

必須の更新

必須のサービスソフトウェア更新には、通常、ドメインの整合性と機能とを維持するための、重要なセキュリティ修正やその他必要な更新が含まれます。必須の更新の例には、Log4j の Common Vulnerabilities and Exposures (CVE) や、Instance Metadata Service Version 2 (IMDSv2) の適用などがあります。必須の更新が行われる回数は、通常、年に 3 回未満です。

OpenSearch サービスは自動的にこれらの更新をスケジュールし、スケジュールされた更新の 72 時間 (3 日) 前に E メールとコンソール通知を通じて通知します。必須の更新は、すぐに適用することもできれば、指定された期間内の別の日時にスケジュール変更することもできます。ドメインの、次回のオフピークの時間帯にスケジュールすることも可能です。必要な更新に対して何もアクションを実行せず、ブルー/グリーンデプロイの原因となるドメインの変更を行わない場合、 OpenSearch サービスは、ドメインのオフピークウィンドウ内で、指定された期限 (通常は利用可能から 14 日間) を過ぎても更新を開始できます。

更新がスケジュールされている時期に関係なく、ブルー/グリーンデプロイ の原因となるドメインに変更を加えると、 OpenSearch サービスによってドメインが自動的に更新されます。

パッチ更新

末尾が「-P」と数字のサービスソフトウェアバージョン (例: R20211203-*P4*) がパッチリリースです。パッチには、パフォーマンスの改善、マイナーなバグ修正、セキュリティ修正または体制の改善が含まれる可能性があります。パッチリリースには新機能や重大な変更は含まれず、通常は、ユーザーに直接的なまたは顕著な影響はありません。サービスソフトウェアの通知を見ると、パッチリリースがオプションか必須かがわかります。

考慮事項

ドメインを更新するかどうかを決定するときには、以下を考慮します。

- ドメインを手動で更新すると、新しい機能をより迅速に活用できます。更新を選択すると、OpenSearch サービスはリクエストをキューに入れ、時間があるときに更新を開始します。
- サービスソフトウェアの更新を開始すると、更新の開始時と完了時に OpenSearch サービスから通知が送信されます。

- ソフトウェア更新では、ダウンタイムを最小化するためにブルー/グリーンデプロイが使用されます。更新によってクラスターの専用マスターノードに一時的に負荷がかかることがあるため、関連するオーバーヘッドを処理するのに十分な容量を確保してください。
- 更新は通常、数分以内に完了しますが、システムに負荷がかかっている場合は、数時間または数日かかることがあります。ドメインを更新するときは、長時間の更新を避けるため、あらかじめ設定したオフピークの時間帯に行うことを検討します。

サービスのソフトウェア更新を開始する

サービスソフトウェアの更新は、 OpenSearch サービスコンソール、 AWS CLI、または SDKsのいずれかを使用してリクエストできます。

コンソール

サービスのソフトウェア更新をリクエストするには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [Actions] (アクション)、[Update] (更新) の順に選択し、次のオプションの中から 1 つを選択します。

   - [Apply update now] (今すぐ更新を適用) - 容量に空きがあれば、アクションを現在の時間で行うよう、すぐにスケジュールします。容量に空きがなければ、別の他の時間帯を選択できます。
   - [Schedule it in off-peak window] (オフピークウィンドウにスケジュールする) — ドメインでオフピークウィンドウが有効になっている場合のみ選択できます。ドメインで設定されたオフピークの時間帯に更新を行うようにスケジュールします。更新が、その次の時間帯に行われるという保証はありません。容量によっては翌日になる場合もあります。詳細については、「the section called “オフピークウィンドウ”」を参照してください。
   - [Schedule for specific date and time] (特定の日時にスケジュールする) — 特定の日時に更新が行われるようにスケジュールします。指定した日時が容量上の理由で利用できない場合は、別の時間帯を選択できます。

更新を後日 (ドメインのオフピークウィンドウ中かそれ以外) にスケジュールした場合は、いつでもスケジュール変更できます。手順については、「the section called “アクションのスケジュール変更”」を参照してください。

4. [確認] を選択します。

AWS CLI

サービスソフトウェアの更新を開始するstart-service-software-update AWS CLI リクエストを送信します。こちらの例では、更新がすぐにキューに追加されています。

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "NOW"
```

レスポンス：

```
{
    "ServiceSoftwareOptions": {
        "CurrentVersion": "R20220928-P1",
        "NewVersion": "R20220928-P2",
        "UpdateAvailable": true,
        "Cancellable": true,
        "UpdateStatus": "PENDING_UPDATE",
        "Description": "",
        "AutomatedUpdateDate": "1969-12-31T16:00:00-08:00",
        "OptionalDeployment": true
    }
}
```

Tip

更新をリクエストした後に、わずかな時間ですが、これをキャンセルできる時間帯があります。このPENDING_UPDATE状態の期間は大きく異なる場合があり、 AWS リージョン と、OpenSearch サービスが実行している同時更新の数によって異なります。更新をキャンセルするには、 コンソールまたは cancel-service-software-update AWS CLI コマンドを使用します。

リクエストが BaseException により失敗した場合は、指定した日時が容量上の理由で利用できないことを意味し、別の日時を指定する必要があります。 OpenSearch サービスは、レスポンスで利用可能な代替スロット候補を提供します。

AWS SDKs

このサンプル Python スクリプトは、 の describe_domain メソッドと start_service_software_update メソッド AWS SDK for Python (Boto3) を使用して、ドメインがサービスソフトウェア更新の対象であるかどうかを確認し、対象であれば更新を開始します。domain_name の値を指定する必要があります。

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

domain_name = ''  # The name of the domain to check and update

client = boto3.client('opensearch', config=my_config)


def getUpdateStatus(client):
    """Determines whether the domain is eligible for an update"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    sso = response['DomainStatus']['ServiceSoftwareOptions']
    if sso['UpdateStatus'] == 'ELIGIBLE':
        print('Domain [' + domain_name + '] is eligible for a service software update
 from version ' +
              sso['CurrentVersion'] + ' to version ' + sso['NewVersion'])
        updateDomain(client)
    else:
        print('Domain is not eligible for an update at this time.')
```

```
def updateDomain(client):
    """Starts a service software update for the eligible domain"""
    response = client.start_service_software_update(
        DomainName=domain_name
    )
    print('Updating domain [' + domain_name + '] to version ' +
          response['ServiceSoftwareOptions']['NewVersion'] + '...')
    waitForUpdate(client)


def waitForUpdate(client):
    """Waits for the domain to finish updating"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    status = response['DomainStatus']['ServiceSoftwareOptions']['UpdateStatus']
    if status == 'PENDING_UPDATE' or status == 'IN_PROGRESS':
        time.sleep(30)
        waitForUpdate(client)
    elif status == 'COMPLETED':
        print('Domain [' + domain_name +
              '] successfully updated to the latest software version')
    else:
        print('Domain is not currently being updated.')

def main():
    getUpdateStatus(client)
```

オフピークの時間帯にソフトウェア更新をスケジュールする

2023 年 2 月 16 日以降に作成された各 OpenSearch サービスドメインには、現地時間の午後 10 時から午前 8 時までの 10 時間のウィンドウがあり、オフピークウィンドウ と見なされます。OpenSearch このウィンドウを使用して、ドメインのサービスソフトウェアの更新をスケジュールします。オフピーク更新は、トラフィックの多い時間帯にクラスターの専用マスターノードにかかる負荷を最小限に抑えるのに役立ちます。 OpenSearch この 10 時間ウィンドウ外に更新を開始することはできません。

- オプションの更新の場合、 OpenSearch サービスは更新の可用性を通知するとともに、次回のオフピークウィンドウ中に更新をスケジュールするように促します。

- 必要な更新については、 OpenSearch サービスが次回のオフピークウィンドウに自動的に更新をスケジュールし、3 日前に通知します。この更新は (オフピークウィンドウ中かそれ以外に) スケジュール変更できますが、更新を完了すべき時間枠内に限られます。

ドメインごとに、デフォルトの午後 10 時の開始時刻を、カスタムの時刻で上書きすることができます。手順については、「the section called “カスタムのオフピークウィンドウの設定”」を参照してください。

コンソール

次回のオフピークウィンドウに更新をスケジュールするには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [アクション]、[更新] の順に選択します。
4. [Schedule it in off-peak window] (オフピークの時間帯に更新をスケジュール) を選択します。
5. [確認] を選択します。

スケジュール済みのアクションは [Off-peak window] (オフピークウィンドウ) タブに表示され、いつでもスケジュール変更できます。the section called “スケジュール済みのアクションを表示する” を参照してください。

CLI

を使用して次回のオフピークウィンドウに更新をスケジュールするには AWS CLI、StartServiceSoftwareUpdateリクエストを送信し、 `--schedule-at`パラメータ`OFF_PEAK_WINDOW`に を指定します。

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "OFF_PEAK_WINDOW"
```

サービスソフトウェア更新のモニタリング

OpenSearch サービスソフトウェアの更新が利用可能、必須、開始、完了、または失敗すると、サービスから通知が送信されます。これらの通知は、 OpenSearch サービスコンソールの通知パ

ネルで表示できます。通知の重要度は、それが必要な場合に更新がオプションおよび High の場合、Informational です。

OpenSearch また、サービスはサービスソフトウェアイベントを Amazon に送信します EventBridge。を使用して EventBridge 、イベントを受信したときに E メールを送信したり、特定のアクションを実行したりするルールを設定できます。チュートリアル例については、the section called “チュートリアル: 利用可能な更新に関する SNS アラートの送信”を参照してください。

Amazon に送信される各サービスソフトウェアイベントの形式を確認するには EventBridge、「」を参照してくださいthe section called “サービスソフトウェア更新イベント”。

ドメインが更新の対象でない場合

次のテーブルに示されているいずれかの状態にある場合、ドメインはサービスソフトウェア更新の対象外である可能性があります。

状態	説明
処理中のドメイン	ドメインは、設定変更中です。オペレーション完了後、更新の適格性を確認します。
赤のクラスター状態	クラスター内の 1 つ以上のインデックスが赤です。トラブルシューティングステップについては、「the section called “赤のクラスター状態”」を参照してください。
高いエラー率	リクエストを処理しようとすると、 OpenSearch クラスターは多数の 5xx エラーを返しています。この問題は通常、同時読み取りまたは書き込みリクエストが多すぎることで起こります。クラスターへのトラフィックを減らすか、ドメインのスケーリングを検討してください。
スプリットブレイン	スプリットブレインとは、 OpenSearch クラスターに複数のマスターノードがあり、それ自体が再結合することのない 2 つのクラスターに分割されていることを意味します。専用マスターノードの推奨値を使用することで、スプリットブレインを回避できます。スプレッドブレインからの復旧方法については、AWS Support にお問い合わせください。
Amazon Cognito の統合の問題	ドメインは OpenSearch Dashboards の認証を使用しますが、OpenSearch サービスは 1 つ以上の Amazon Cognito リソースを見つけることができません。この問題は通常、Amazon Cognito ユーザープー

状態	説明
	ルがない場合に発生します。この問題を解決するには、欠落しているリソースを再作成し、それを使用するように OpenSearch サービスドメインを設定します。
その他の サービスの問題	OpenSearch サービス自体に問題があると、ドメインが更新対象外として表示される可能性があります。前の条件がドメインに当てはまらず、問題が 1 日以上続く場合は、AWS Support にお問い合わせください。

Amazon OpenSearch Service のオフピークウィンドウの定義

Amazon OpenSearch Service ドメインを作成するときは、オフピーク時間と見なされる毎日の 10 時間ウィンドウを定義します。 OpenSearch このウィンドウを使用して、可能な限りトラフィック時間を比較的短くしながら、サービスソフトウェアの更新とブルー/グリーンデプロイを必要とする Auto-Tune 最適化をスケジュールします。ブルー/グリーンとは、ドメインの更新用に新しい環境を作成し、この更新の完了後にユーザーを新しい環境にルーティングするプロセスを指します。

ブルー/グリーンデプロイは無停止ですが、リソースがブルー/グリーンデプロイに使用されている間の、パフォーマンスへの潜在的影響を最小限に抑えるため、このようなデプロイは、ドメインに設定されているオフピークウィンドウ中にスケジュールすることが推奨されます。ノードの交換や、ドメインにただちにデプロイすべき更新などでは、オフピークウィンドウは使用しません。

オフピークウィンドウの開始時刻は変更が可能ですが、ウィンドウの所要時間は変更できません。

> **Note**
>
> オフピークウィンドウは 2023 年 2 月 16 日に導入されました。この日付以前に作成されたすべてのドメインでは、オフピークウィンドウはデフォルトで無効になっています。これらのドメインでは、オフピークウィンドウを手動で有効にし、設定する必要があります。この日付以後に作成されたすべてのドメインで、オフピークウィンドウはデフォルトで有効になっています。ドメインのオフピークウィンドウは、いったん有効にした後は、無効にできません。

トピック

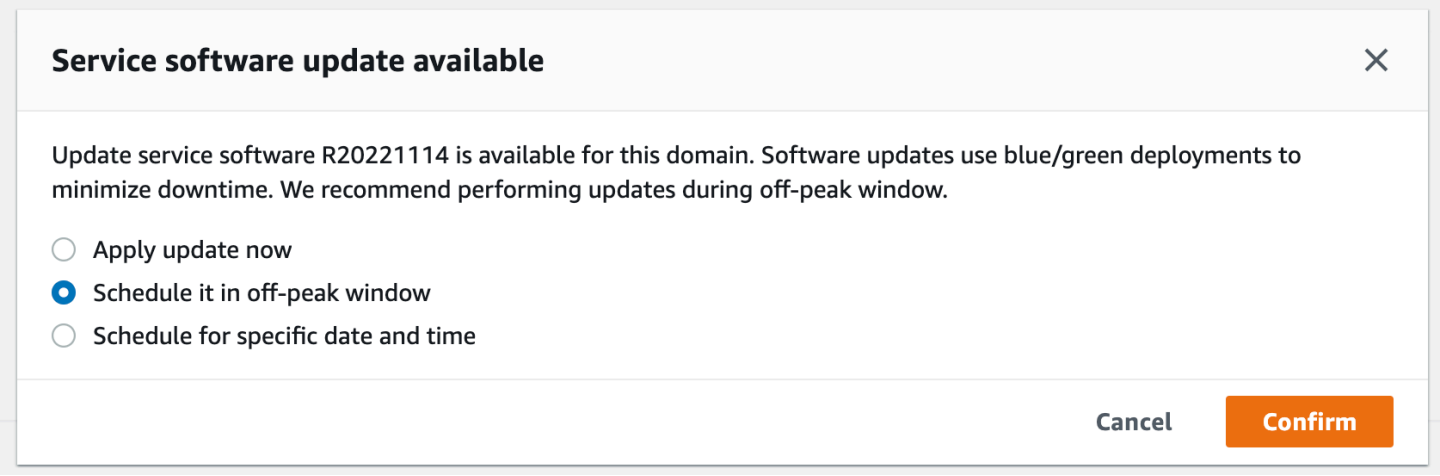
- オフピーク時のサービスソフトウェアの更新

- オフピーク時の Auto-Tune の最適化
- オフピークウィンドウの有効化
- カスタムのオフピークウィンドウの設定
- スケジュール済みのアクションを表示する
- アクションのスケジュール変更
- Auto-Tune のメンテナンスウィンドウからの移行

オフピーク時のサービスソフトウェアの更新

OpenSearch サービスには、オプションと必須の という 2 つのカテゴリのサービスソフトウェア更新があります。どちらのカテゴリにもブルー/グリーンデプロイが必要です。オプションの更新はお使いのドメインに強制されることはありませんが、必須の更新は、指定された期限 (通常は提供開始の 2 週間後) までにアクションを実行しないと自動的にインストールされます。詳細については、「the section called “オプションの更新と必須の更新”」を参照してください。

オプションの更新を実行するときは、更新をすぐに適用するか、次回のオフピークウィンドウ時にスケジュールする、またはカスタムの日時を指定して適用するかの、いずれかを選択できます。



必要な更新については、 OpenSearch サービスは自動的にオフピークの時間帯に更新を実行する日付と時刻をスケジュールします。ユーザーには、スケジュールされた更新の 3 日前に通知が届きます。必須のデプロイの期間内であれば、更新の日時を後の日時に変更することも可能です。手順については、「the section called “アクションのスケジュール変更”」を参照してください。

オフピーク時の Auto-Tune の最適化

これまで、Auto-Tune では、ブルー/グリーンデプロイが必要なスケジュールを変更する場合、メンテナンスウィンドウが使用されていました。オフピークウィンドウが導入される前に、Auto-Tune とメンテナンスウィンドウが有効になっていたドメインでは、オフピークウィンドウを使用するようにドメインを移行しない限り、更新には引き続きメンテナンスウィンドウが使用されます。

当社では、オフピークウィンドウを使用するようにドメインを移行することを推奨しています。サービスソフトウェアの更新など、ドメインの他のアクティビティのスケジュールに使用するためです。手順については、「the section called “Auto-Tune のメンテナンスウィンドウからの移行”」を参照してください。ドメインをオフピークウィンドウに移行した後は、メンテナンスウィンドウの使用に戻ることはできません。

2023 年 2 月 16 日以降に作成されたすべてのドメインでは、ブルー/グリーンデプロイのスケジュールでは、従来のメンテナンスウィンドウではなくオフピークウィンドウを使用します。ドメインのオフピークウィンドウは無効にできません。ブルー/グリーンデプロイを必要とする Auto-Tune 最適化のリストについては、「the section called “変更のタイプ”」を参照してください。

オフピークウィンドウの有効化

2023 年 2 月 16 日 (オフピークウィンドウの導入日) 以前に作成されたドメインでは、この機能はデフォルトで無効になっています。これらのドメインでは、オフピークウィンドウを手動で有効にする必要があります。ドメインのオフピークウィンドウは、有効にした後は無効にできません。

コンソール

ドメインのオフピークウィンドウを有効にするには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [オフピークウィンドウ] タブに移動し、[編集] を選択します。
4. 協定世界時 (UTC) でカスタムの開始時刻を指定します。例えば、開始時刻を米国西部 (オレゴン) リージョンの午後 11 時 30 分に設定するときは、[07:30] と指定します。
5. [変更の保存] をクリックします。

CLI

を使用してオフピークウィンドウを変更するにはAWS CLI、 UpdateDomainConfigリクエストを送信します。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'Enabled=true,
 OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

カスタムのウィンドウ開始時刻を指定しない場合、00:00 UTC がデフォルトの時刻となります。

カスタムのオフピークウィンドウの設定

ドメインのカスタムオフピークウィンドウを協定世界時 (UTC) で指定します。例えば、オフピークウィンドウを米国東部 (バージニア北部) リージョンのドメインで午後 11 時に開始するときは、[04:00 UTC] と指定します。

コンソール

ドメインのオフピークウィンドウを変更するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [オフピークウィンドウ] タブに移動します。設定済みのオフピークウィンドウと、今後予定されているドメインのアクション一覧が表示されます。
4. [編集] を選択し、新しい開始時刻を UTC で指定します。例えば、開始時刻を米国東部 (バージニア北部) リージョンの午後 9 時に設定するときは、[02:00 UCT] と指定します。
5. [変更の保存] をクリックします。

CLI

を使用してカスタムのオフピークウィンドウを設定するにはAWS CLI、 UpdateDomainConfigリクエストを送信し、24 時間形式の時と分を指定します。

例えば、次のリクエストを送信すると、ウィンドウの開始時刻が UTC の午前 2 時に変更されます。

```
aws opensearch update-domain-config \
```

```
  --domain-name my-domain \
  --off-peak-window-options 'OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

ウィンドウの開始時刻を指定しなかった場合、ドメインが作成された AWS リージョン の現地時間午後 10 時がデフォルトで開始時刻となります。

スケジュール済みのアクションを表示する

各ドメインの現在スケジュール済みのアクション、進行中のアクション、保留中のアクションは、すべて表示できます。アクションには、HIGH、MEDIUM、LOW の重要度のレベルがあります。

アクションのステータスは次の通りです。

- Pending update — アクションはキューに入って処理待ちです。
- In progress — アクションは進行中です。
- Failed - アクションの実行に失敗しました。
- Completed – アクションは正常に完了しました。
- Not eligible — サービスソフトウェアの更新のみです。クラスターが異常な状態であるため、更新を続行できません。
- Eligible — サービスソフトウェアの更新のみです。このドメインは更新の対象になっています。

コンソール

OpenSearch サービスコンソールには、ドメイン設定内のすべてのスケジュールされたアクションが、各アクションの重要度と現在のステータスとともに表示されます。

ドメインでスケジュールされているアクションを表示するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [オフピークウィンドウ] タブに移動します。
4. [スケジュールされたアクション] の下に、ドメインで現在スケジュールされているアクション、進行中のアクション、保留中のアクションがすべて表示されます。

CLI

を使用してスケジュールされたアクションを表示するにはAWS CLI、 ListScheduledActionsリクエストを送信します。

```
aws opensearch list-scheduled-actions \
  --domain-name my-domain
```

レスポンス：

```
{
    "ScheduledActions": [
        {
            "Cancellable": true,
            "Description": "The Deployment type is : BLUE_GREEN.",
            "ID": "R20220721-P13",
            "Mandatory": false,
            "Severity": "HIGH",
            "ScheduledBy": "CUSTOMER",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "SERVICE_SOFTWARE_UPDATE",
        },
        {
            "Cancellable": true,
            "Description": "Amazon Opensearch will adjust the young generation JVM
 arguments on your domain to improve performance",
            "ID": "Auto-Tune",
            "Mandatory": true,
            "Severity": "MEDIUM",
            "ScheduledBy": "SYSTEM",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "JVM_HEAP_SIZE_TUNING",
        }
    ]
}
```

アクションのスケジュール変更

OpenSearch サービスは、スケジュールされたサービスソフトウェアの更新と Auto-Tune 最適化を通知します。スケジュールの変更は、すぐに適用することもできれば、別の日時に変えることもできます。

> **Note**
>
> OpenSearch サービスは、選択した時刻から 1 時間以内にアクションをスケジュールできます。例えば、更新を午後 5 時に適用する場合、午後 5 時から 6 時の間に適用できます。

コンソール

アクションのスケジュールを変更するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ドメイン名を選択し、設定を開きます。
3. [オフピークウィンドウ] タブに移動します。
4. [スケジュールされたアクション] の下で、アクションを選択し、[スケジュール変更] を選択します。
5. 以下のオプションのいずれかを選択します。
   - [今すぐ更新を適用] - 容量に空きがあれば、アクションを現在の時間で行うよう、すぐにスケジュールします。容量に空きがなければ、別の他の時間帯を選択できます。
   - [オフピークウィンドウでのスケジュール設定] - 次回のオフピークウィンドウ中にピックアップするようにアクションにマークします。変更が、次回のウィンドウ中に適用されるという保証はありません。容量によっては翌日になる場合もあります。
   - [この更新のスケジュールを変更する] - 変更を適用する日付と時刻をカスタムで指定できます。指定した日時が容量上の理由で利用できない場合は、別の時間帯を選択できます。
   - [スケジュールされた更新のキャンセル] - 更新をキャンセルします。このオプションを利用できるのは、オプションのサービスソフトウェア更新のみです。Auto-Tune アクション、または必須のソフトウェア更新には利用できません。
6. [変更の保存] をクリックします。

CLI

を使用してアクションのスケジュールを変更するにはAWS CLI、 UpdateScheduledActionリクエストを送信します。アクション ID を取得するには、ListScheduledActions リクエストを送信します。

次のリクエストを送信すると、サービスソフトウェアの更新が特定の日時にスケジュール変更されます。

```
aws opensearch update-scheduled-action \
  --domain-name my-domain \
  --action-id R20220721-P13 \
  --action-type "SERVICE_SOFTWARE_UPDATE" \
  --desired-start-time 1677348395000 \
  --schedule-at TIMESTAMP
```

レスポンス：

```
{
   "ScheduledAction": {
      "Cancellable": true,
      "Description": "Cluster status is updated.",
      "Id": "R20220721-P13",
      "Mandatory": false,
      "ScheduledBy": "CUSTOMER",
      "ScheduledTime": 1677348395000,
      "Severity": "HIGH",
      "Status": "PENDING_UPDATE",
      "Type": "SERVICE_SOFTWARE_UPDATE"
   }
}
```

リクエストが SlotNotAvailableException により失敗した場合は、指定した日時が容量上の理由で利用できないことを意味し、別の日時を指定する必要があります。 OpenSearch サービスは、レスポンスで利用可能な代替スロット候補を提供します。

Auto-Tune のメンテナンスウィンドウからの移行

ドメインが 2023 年 2 月 16 日以前に作成された場合、ブルー/グリーンデプロイを必要とする Auto-Tune 最適化のスケジュールに、メンテナンスウィンドウを使用できます。既存の Auto-Tune ドメインを移行して、代わりにオフピークウィンドウを使用します。

> **Note**
>
> オフピークウィンドウを使用するようにドメインを移行した後は、メンテナンスウィンドウの使用に戻ることはできません。

コンソール

オフピークウィンドウを使用するようにドメインを移行するには

1. Amazon OpenSearch Service コンソールで、ドメインの名前を選択して設定を開きます。
2. [Auto-Tune] タブに進み、[編集] を選択します。
3. [オフピークウィンドウへ移行] を選択します。
4. [開始時刻 (UTC)] に、オフピークウィンドウの開始時刻を協定世界時 (UTC) で入力します。
5. [変更の保存] をクリックします。

CLI

を使用して Auto-Tune メンテナンスウィンドウからオフピークウィンドウに移行するにはAWS CLI、 UpdateDomainConfigリクエストを送信します。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=[]
```

ドメインを Auto-Tune メンテナンスウィンドウからオフピークウィンドウへ移行するには、オフピークウィンドウをオンにします。オフピークウィンドウは、個別のリクエストで、または同じリクエストで有効にすることができます。手順については、「the section called “オフピークウィンドウの有効化”」を参照してください。

Amazon OpenSearch Service での通知

Amazon OpenSearch Service の通知には、ドメインのパフォーマンスと正常性に関する重要な情報が含まれています。 OpenSearch サービスは、サービスソフトウェアの更新、Auto-Tune の強化、クラスターヘルスイベント、およびドメインエラーについて通知します。通知は、 OpenSearch と Elasticsearch OSS のすべてのバージョンで利用できます。

通知は、 OpenSearch サービスコンソールの通知パネルで表示できます。 OpenSearch サービスのすべての通知は Amazon EventBridgeにも表示されます。通知とサンプルイベントの完全なリストについては、「the section called “イベントのモニタリング”」を参照してください。

トピック

- 通知の使用開始
- 通知重要度
- サンプル EventBridge イベント

通知の使用開始

通知は、ドメインの作成時に自動的に有効になります。 OpenSearch サービスコンソールの通知パネルに移動して、通知をモニタリングして確認します。各通知には、通知が投稿された時刻、関連するドメイン、重要度とステータスレベル、簡単な説明などの情報が含まれます。コンソールでは、最大 90 日間の履歴通知を表示できます。

[通知] パネルにアクセスした後または通知を確認した後に、`es:ListNotifications` または `es:UpdateNotificationStatus` を実行するアクセス許可がないことを示すエラーメッセージが表示されることがあります。この問題を解決するには、IAM でユーザーまたはロールに次のアクセス許可を付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "es:UpdateNotificationStatus",
      "es:ListNotifications"
    ],
    "Resource": "arn:aws:es:*:123456789012:domain/*"
  }]
}
```

IAM コンソールは、エラー (「IAM does not recognize one or more actions.」) をスローしますが、このエラーは安全に無視できます。また、特定のドメインに対する `es:UpdateNotificationStatus` アクションを制限することもできます。詳細については、「the section called “ポリシーエレメントのリファレンス”」を参照してください。

通知重要度

OpenSearch Service の通知は、既に実行したアクションやドメインのオペレーションに関連する情報 、または必須のセキュリティパッチの適用などの特定のアクションを実行する必要があるアクションの実行可能な です。各通知には、重要度 (Informational、Low、Medium、High、または Critical) が関連付けられています。次の表に重大度の一覧を示します。

緊急度	説明	例
Informational	ドメインのオペレーションに関連する情報。	• サービスソフトウェア更新が利用可能 • Auto-Tune が開始されました
Low	推奨されるアクションですが、アクションを行わない場合、ドメインの可用性やパフォーマンスに悪影響を及ぼしません。	• Auto-Tune がキャンセルされました • シャードカウントが高い警告
Medium	推奨されるアクションが実行されないが、アクションが実行されるまでの時間枠が長くなった場合、影響が生じる可能性があります。	• サービスソフトウェアの更新に失敗しました • シャード数の制限を超えました
High	悪影響を避けるためには、緊急のアクションが必要です。	• サービスソフトウェア更新が必要 • KMS キーにアクセスできない
Critical	悪影響を避けるために、またはそこから回復するために、即時のアクションが必要です。	現在利用可能なものはありません

サンプル EventBridge イベント

次の例は、Amazon に送信される OpenSearch サービス通知イベントを示しています EventBridge。更新はオプションでなので、通知の重要度は Informational です。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] available."
  }
}
```

Amazon OpenSearch サービスのマルチ AZ ドメインの設定

データ損失を防ぎ、サービスが中断した際の Amazon OpenSearch Service クラスターのダウンタイムを最小限に抑えるため、同じリージョンの 2 つまたは 3 つのアベイラビリティーゾーンにノードを分散できます。これはマルチ AZ と呼ばれる構成です。アベイラビリティーゾーンは各リージョン内の独立した場所です。 AWS

プロダクションワークロードを実行するドメインでは、Multi-AZ with Standby デプロイのオプションが推奨されます。このオプションが次の構成を作成します。

- ドメインが 3 つのゾーン間にデプロイされました。
- 専用マスターノードおよびデータノードに、現行世代のインスタンスタイプを選択します。
- 3 つの専用マスターノードと 3 つ (または 3 の倍数) のデータノード。
- ドメイン内の各インデックスに 2 つ以上のレプリカ、または 3 の倍数のデータコピー (プライマリノードとレプリカの両方を含む)。

このセクションの残りの部分では、これらの構成とそのコンテキストについて説明します。

Multi-AZ with Standby

Multi-AZ with Standby は、99.99% の可用性、本番環境のワークロードに対する一貫したパフォーマンス、およびドメインの設定と管理の簡素化を実現する Amazon OpenSearch Service ドメインのデ

プロイメントオプションです。Multi-AZ with Standby を使用したドメインは、インフラストラクチャの障害に対して回復力があり、パフォーマンスや可用性に一切影響を与えません。このデプロイオプションは、指定されたデータノード数、マスターノード数、インスタンスタイプ、レプリカ数、ソフトウェア更新設定、Auto-Tune の有効化など、数多くのベストプラクティスを義務付けることでこの標準を達成しています。

マルチ AZ をスタンバイで使用すると、 OpenSearch Service は 3 つのアベイラビリティーゾーンにまたがるドメインを作成します。各ゾーンにはデータの完全なコピーが含まれ、データは各ゾーンに均等に分散されます。ドメインは、これらのゾーンのいずれかの 1 つのノードをスタンバイとして予約します。このノードは、検索リクエストの処理を行いません。 OpenSearch Service は、基盤となるインフラストラクチャの障害を検出すると、1 分もかからずにスタンバイノードを自動的にアクティブ化します。ドメインは、引き続きインデックス作成と検索リクエストを処理し、その影響は、フェイルオーバーの実行に必要な時間に限られます。データやリソースの再配布は行われないため、クラスターのパフォーマンスには影響せず、可用性が下がるリスクもありません。Multi-AZ with Standby は、追加コストなしでご利用いただけます。

AWS Management Consoleでスタンバイ付きのドメインを作成するには、2 つの方法があります。まず、Easy create 作成方法でドメインを作成すると、 OpenSearch Service は以下を含むあらかじめ決められた設定を自動的に使用します。

- 3 つのアベイラビリティーゾーン、1 つはスタンバイとして機能
- 3 つの専用マスターノードとデータノード
- Auto-Tune がドメインで有効になっている
- データノード用の GP3 ストレージ

また、[標準作成] の作成方法を選択し、デプロイオプションとして [スタンバイ状態のドメイン] を選択することもできます。これにより、ドメインをカスタマイズしながら、引き続き 3 つのゾーンと 3 つのマスターノードなどスタンバイの主な機能を要求することができます。データノードの数は 3 の倍数 (アベイラビリティーゾーンの数) を選ぶことが推奨されます。

ドメインを作成したら、ドメインの詳細ページに移動し、[クラスター設定] タブで、アベイラビリティーゾーンに「スタンバイ状態の 3 つの AZ」と表示されていることを確認します。

既存のドメインを Multi-AZ with Standby へ移行する際に問題が生じた場合は、トラブルシューティングガイドの「Multi-AZ with Standby に移行する際のエラー」を参照してください。

制限事項

Multi-AZ with Standby のドメインを設定するときは、次の制限を考慮します。

- ノード上のシャードの合計数は 1000 を超えることはできず、クラスター上のシャードの合計数は 75000 を超えることはできず、1 つのシャードの容量は 65 GB を超えることはできません。
- Multi-AZ with Standby は、m5、c5、r5、r6g、c6g、m6g、r6gd、i3 インスタンスタイプのみで機能します。サポートされているインスタンスタイプの詳細については、「サポートされるインスタンスタイプ」を参照してください。
- スタンバイではプロビジョンド IOPS SSD、汎用 SSD (GP3)、または instance-backed ストレージのみを使用できます。
- Multi-AZ with Standby UltraWarmドメインで有効にする場合、ウォームノードの数は、使用するアベイラビリティーゾーンの数の倍数でなければなりません。

Multi-AZ without Standby

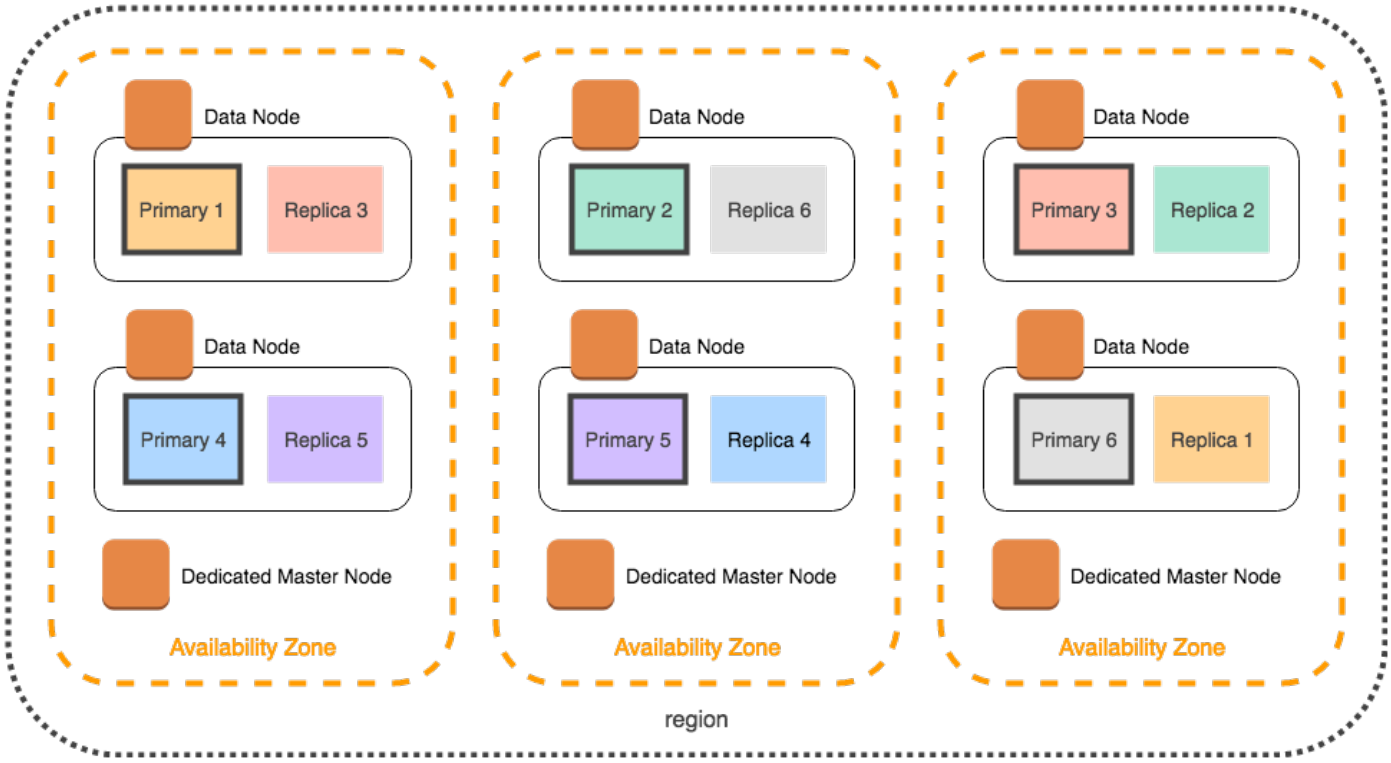
OpenSearch サービスはスタンバイなしでもマルチ AZ をサポートし、99.9% の可用性を実現します。ノードはアベイラビリティーゾーン全体に分散しており、可用性は、アベイラビリティーゾーンとデータコピーの数に依存します。スタンバイ付きの場合はドメインをベストプラクティスで設定する必要がありますが、スタンバイが付いていない場合は、アベイラビリティゾーン、ノード、レプリカの数を自分で選択できます。この方法は、スタンバイ付きのドメインを作成した場合に既存のワークフローが中断する可能性がない限り、推奨されません。

この方法を選択した場合でも、ノード、ディスク、シングル AZ での障害に対する回復力を維持するため、3 つのアベイラビリティーゾーンを選択することが引き続き推奨されます。障害が発生すると、クラスターは、可用性と冗長性を保つため他のリソースにデータを再分散します。このデータ移動によりクラスターでのリソースの使用量が増加し、パフォーマンスに影響する可能性があります。クラスターの容量が適切でないと可用性は低下します。これでは、マルチ AZ の目的が大きく損なわれかねません。

でスタンバイせずにドメインを設定する唯一の方法は、標準作成方法を選択し、デプロイオプションとして [スタンバイなしのドメイン] を選択することです。 AWS Management Console

シャード分散

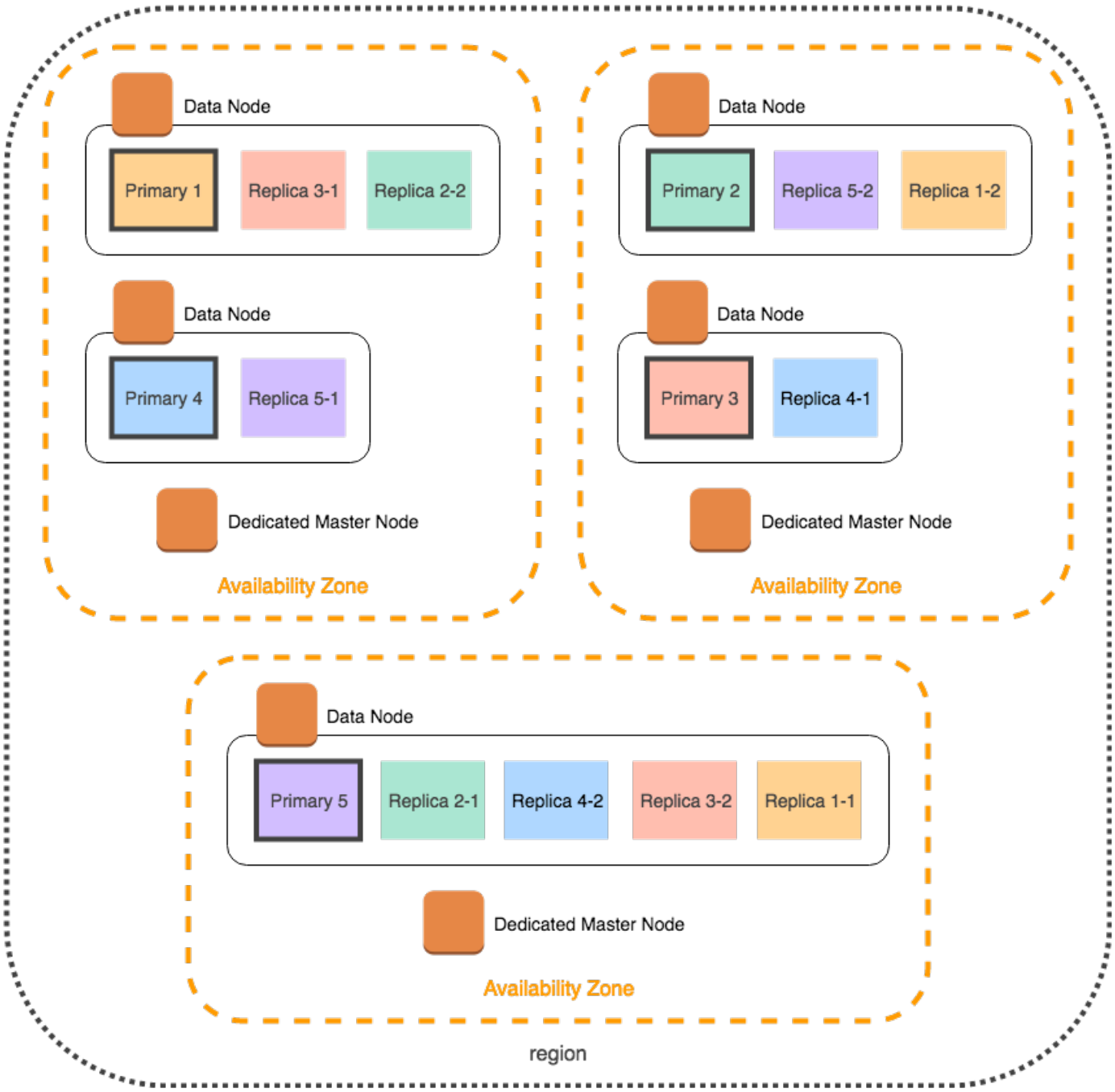
スタンバイの付いていないマルチ AZ を有効にする場合は、クラスターの各インデックスに 1 つ以上のレプリカを作成します。レプリカがないと、 OpenSearch Service はデータのコピーを他のアベイラビリティーゾーンに配布できません。さいわい、すべてのインデックスのデフォルト設定で、レプリカの数は 1 つです。次の図が示すように、 OpenSearch Service はプライマリシャードとそれに対応するレプリカシャードをさまざまなゾーンに分散するように最善を尽くします。



OpenSearch Service はアベイラビリティーゾーンごとにシャードを配布するだけでなく、ノードごとにシャードを配布します。それでも、特定のドメイン設定ではシャード数が不均等にある場合があります。次のドメインを考えます。

- データノード x 5
- プライマリシャード x 5
- レプリカ x 2
- アベイラビリティーゾーン x 3

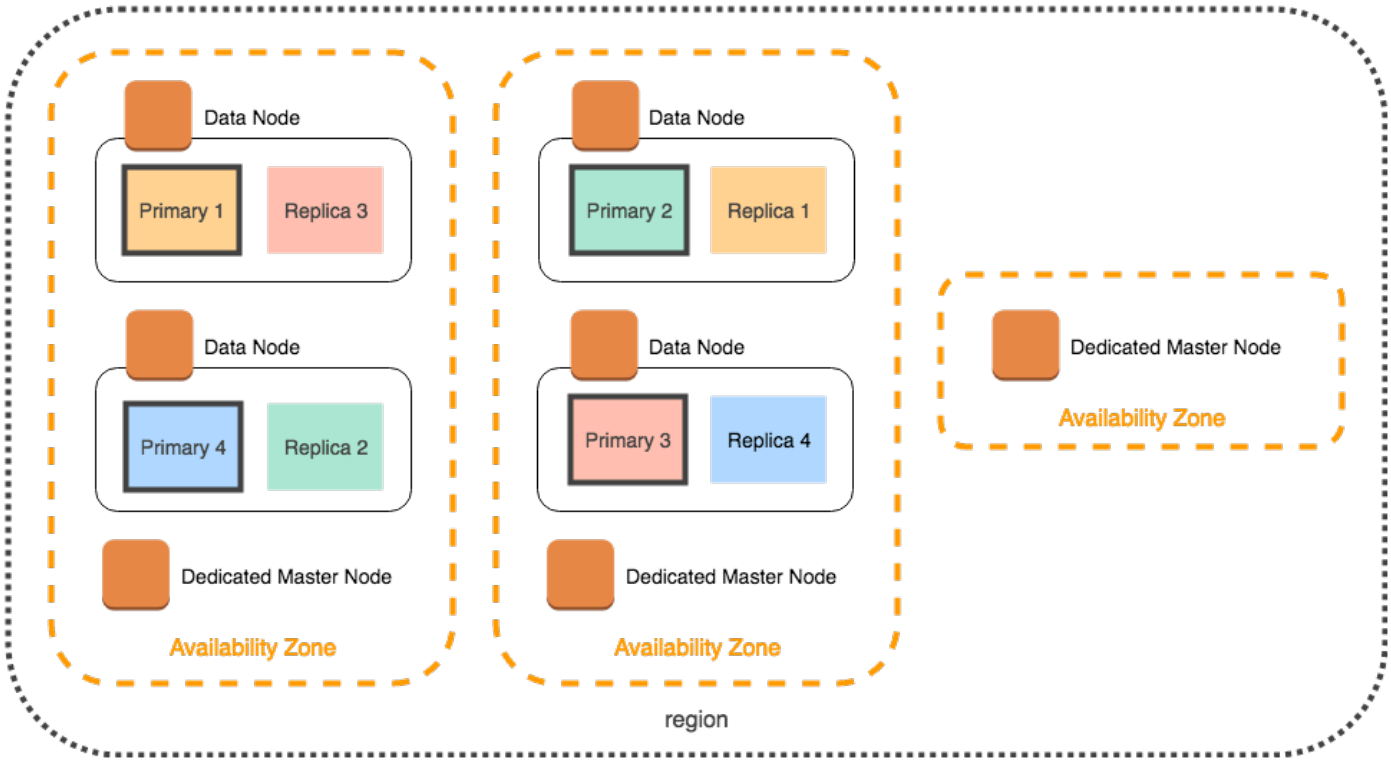
この状況では、次の図に示すように、 OpenSearch Service はプライマリシャードとレプリカシャードをゾーン全体に分散させるために 1 つのノードに過負荷をかける必要があります。



こうした、個々のノードに負荷がかかってパフォーマンスが低下する状況を避けるため、Multi-AZ with Standby を選択するか、インデックスごとのレプリカ数を 2 つ以上にする場合は インスタンスの数を 3 の倍数にすることが推奨されます。

専用マスターノード分散

ドメインの設定時に 2 つのアベイラビリティーゾーンを選択しても、 OpenSearch Service は専用マスターノードを 3 つのアベイラビリティーゾーンに自動的に分散します。この分散により、ゾーンでサービス中断が発生した場合にクラスターのダウンタイムを防ぐことができます。推奨される 3 つの専用マスターノードを使用していて、1 つのアベイラビリティーゾーンがダウンした場合でも、クラスターには専用マスターノードのクォーラム (2) が残り、新しいマスターを選択できます。この設定は以下の図のようになります。



3 つのアベイラビリティーゾーンで使用できない旧世代のインスタンスタイプを選択した場合、以下のシナリオが適用されます。

- ドメインに 3 つのアベイラビリティーゾーンを選択した場合、 OpenSearch Service はエラーを返します。異なるインスタンスタイプを選択し、もう一度試します。
- ドメインに 2 つのアベイラビリティーゾーンを選択した場合、 OpenSearch Service は専用マスターノードを 2 つのゾーンに分散します。

アベイラビリティーゾーンの中断

アベイラビリティーゾーンの中断が発生することはほとんどありませんが、起こり得ます。以下の表に、各種マルチ AZ 設定と中断時の動作を示します。表の最後の行は Multi-AZ with Standby に適用され、それ以外のすべての行は Multi-AZ without Standby にのみ適用される構成になっています。

リージョン内のアベイラビリティーゾーン数	選択したアベイラビリティーゾーンの数	専用マスターノードの数	1 つのアベイラビリティーゾーンで中断が発生した場合の動作
2 以上	2	0	ダウンタイム。クラスターのデータノードの半分が失われます。マスターを選択する前に残りのアベイラビリティーゾーンで少なくとも 1 つを置き換える必要があります。
2	2	3	ダウンタイムの可能性は 50/50。 OpenSearch サービスは 2 つの専用マスターノードを 1 つのアベイラビリティーゾーンに、もう 1 つをもう 1 つのアベイラビリティーゾーンに分散します。 • 専用マスターノードが 1 つのアベイラビリティーゾーンで中断が発生した場合、残りのアベイラビリティーゾーン内の 2 つの専用マスターノードがマスターを選択できます。 • 専用マスターノードが 2 つのアベイラビリティーゾーンで中断が発生した場合、残りのアベイラビリティーゾーンが回復するまでクラスターは使用できません。
3 以上	2	3	ダウンタイムは発生しません。 OpenSearch Service は専用マスターノードを 3 つのアベイラビリティーゾーンに自動的に分散するため、残りの 2 つの専用マスターノードがマスターを選択できます。

リージョン内のアベイラビリティーゾーン数	選択したアベイラビリティーゾーンの数	専用マスターノードの数	1 つのアベイラビリティーゾーンで中断が発生した場合の動作
3 以上	3	0	ダウンタイムはありません。データノードの約 3 分の 2 が引き続きマスターを選択できます。
3 以上	3	3	ダウンタイムはありません。残りの 2 つの専用マスターノードがマスターを選択できます。

どの構成でも、原因にかかわらず、ノード障害によってクラスターの残りのデータノードの負荷が高まる可能性があります。その間、 OpenSearch Service は、現在欠落しているノードを置き換える新しいノードを自動的に構成します。

たとえば、3 ゾーン設定でアベイラビリティーゾーンが中断した場合、3 分の 2 に相当するデータノードがクラスターへのリクエストを処理する必要があります。これらのリクエストを処理するとき、残りのノードはオンラインになる新しいノードにもシャードをレプリケートするため、さらにパフォーマンスに影響が及びます。ワークロードにとって可用性が重要な場合、クラスターにリソースを追加してこの問題を緩和することを検討してください。

> **Note**
>
> OpenSearch Service はマルチ AZ ドメインを透過的に管理するため、アベイラビリティーゾーンの中断を手動でシミュレートすることはできません。

VPC 内で Amazon OpenSearch Service ドメインを起動する

Virtual Private Cloud (VPC) では、Amazon OpenSearch Service ドメインなどの AWS リソースを起動できます。VPC は、 専用の仮想ネットワークです AWS アカウント。VPC は、 AWS クラウドの他の仮想ネットワークから論理的に切り離されています。 OpenSearch サービスドメインを VPC に配置すると、インターネットゲートウェイ、NAT デバイス、VPN 接続を必要とせずに、OpenSearch VPC 内のサービスと他のサービス間の安全な通信が可能になります。すべてのトラフィックは AWS クラウド内で安全に保持されます。

> **Note**
>
> OpenSearch サービスドメインを VPC に配置する場合、コンピュータは VPC に接続できる必要があります。多くの場合、この接続では、VPN、Transit Gateway、マネージド型のネットワーク、またはプロキシサーバーを使用します。VPC の外部からドメインに直接アクセスすることはできません。

トピック

VPC 対パブリックドメイン

以下に、VPC ドメインとパブリックドメインの違いを示します。それぞれの違いについては、後半で説明します。

- 論理的な隔離により、VPC 内に存在するドメインには、パブリックエンドポイントを使用するドメインに比較して、より拡張されたセキュリティレイヤーがあります。
- パブリックドメインはインターネットに接続されたあらゆるデバイスからアクセスできますが、VPC ドメインには何らかの形式の VPN またはプロキシが必要です。
- パブリックドメインと比較すると、コンソールに表示される VPC ドメインの情報は少なくなります。具体的には、[クラスターヘルス] タブにはシャード情報が含まれておらず、[インデックス] タブは存在しません。
- ドメインエンドポイントは、異なる形式 (`https://search-domain-name` 対 `https://vpc-domain-name`) を取ります。
- セキュリティグループには IP ベースのアクセス権限ポリシーですでに強化されているため、VPC 内に存在するドメインに IP ベースのアクセス権限ポリシーを適用することはできません。

制限事項

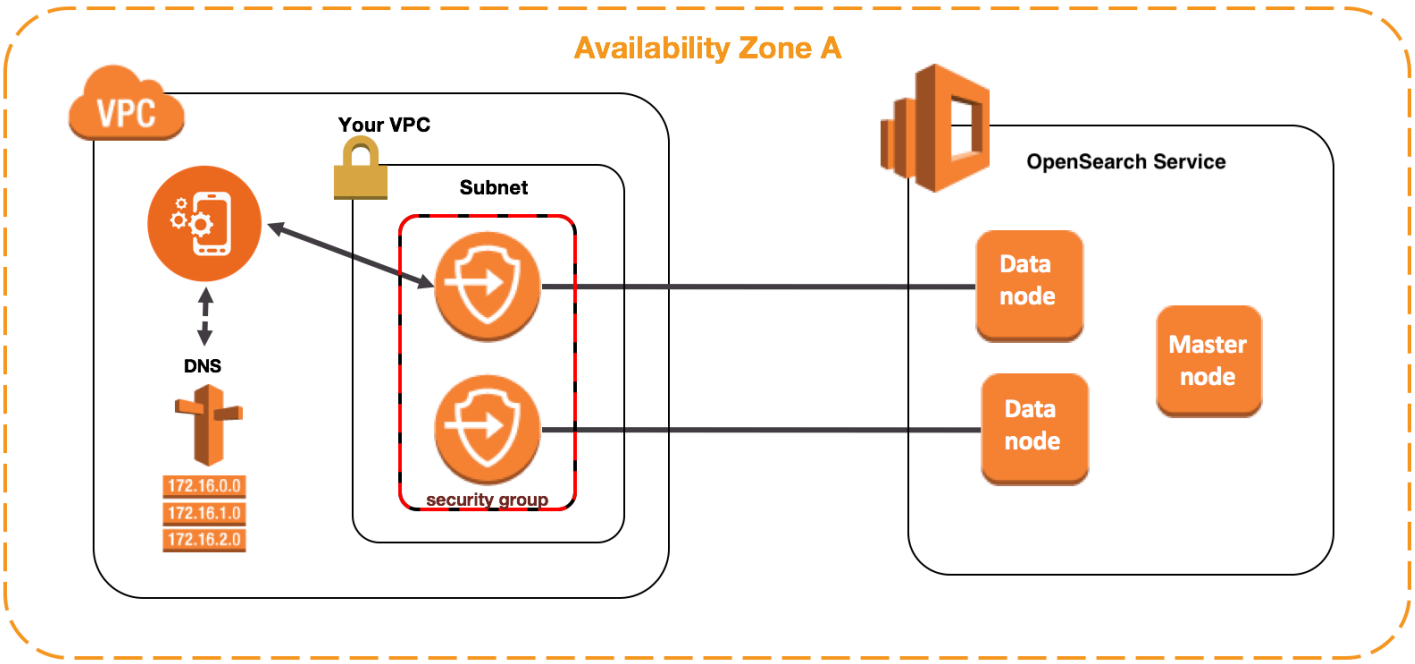
VPC 内の OpenSearch サービスドメインの運用には、次の制限があります。

- VPC 内に新規ドメインを起動する場合、後でパブリックエンドポイントの使用に切り替えることはできません。その逆も同じ結果となります。パブリックエンドポイントでドメインを作成する場合、後で VPC に配置することはできません。代わりに、新規のドメインを作成して、データを移行する必要があります。
- VPC 内でドメインを起動すること、あるいはパブリックエンドポイントを使用することができますが、両方を実行することはできません。ドメイン作成時にどちらかを選択する必要があります。
- 専有テナントを使用している VPC 内でドメインを起動することはできません。テナントを [デフォルト] に設定した VPC を使用する必要があります。
- VPC 内にドメインをセットした後で、そのドメインを別の VPC に移動することはできませんが、サブネットとセキュリティグループの設定は変更可能です。
- VPC 内に存在するドメインの OpenSearch Dashboards のデフォルトインストールにアクセスするには、ユーザーが VPC にアクセスできる必要があります。このプロセスはネットワーク構成によって異なりますが、VPN への接続、ネットワークの管理あるいはプロキシサーバーまたは Transit Gateway の使用が必要となる場合がほとんどです。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」、「Amazon VPC ユーザーガイド」および「the section called “ OpenSearch Dashboards へのアクセスの制御”」を参照してください。

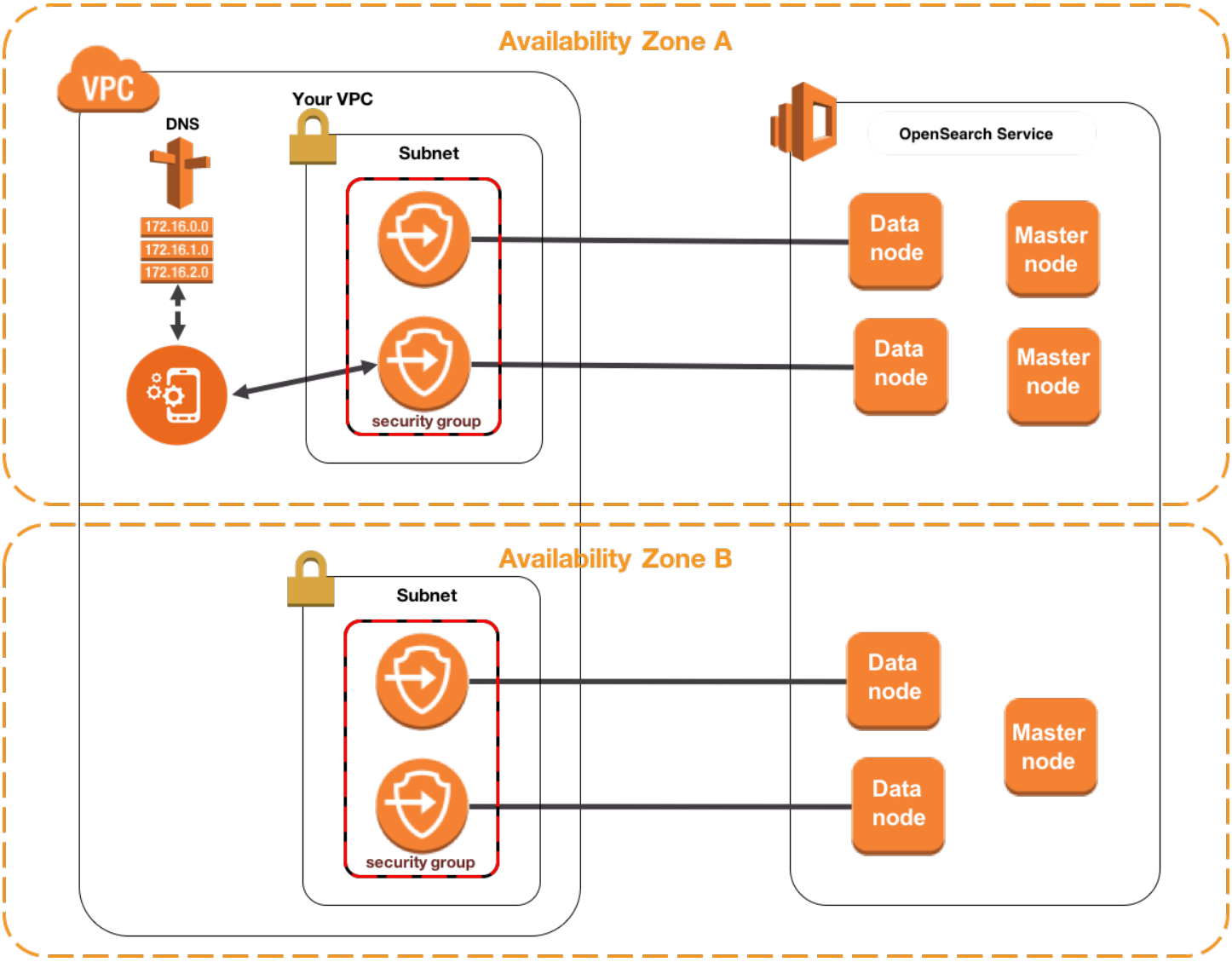
アーキテクチャ

VPCs、 OpenSearch サービスは VPC の 1 つ、2 つ、または 3 つのサブネットにエンドポイントを配置します。ドメインに対して複数のアベイラビリティーゾーンを有効にする場合、各サブネットは同じリージョンの異なるアベイラビリティーゾーンに存在する必要があります。1 つのアベイラビリティーゾーンのみを使用する場合、 OpenSearch サービスはエンドポイントを 1 つのサブネットにのみ配置します。

次の図は、1 つのアベイラビリティーゾーンの VPC アーキテクチャを示しています。



次の図は、2 つのアベイラビリティーゾーンの VPC アーキテクチャを示しています。



OpenSearch また、サービスは各データノードの VPC に Elastic Network Interface (ENI) を配置します。 OpenSearch サービスは、サブネットの IPv4 アドレス範囲から各 ENI にプライベート IP アドレスを割り当てます。また、このサービスは、IP アドレスにパブリック DNS ホスト名 (これは、ドメインエンドポイントです) も割り当てます。データノード用に適確な IP アドレスでエンドポイント (DNS ホスト名) を解決するために、パブリック DNS サービスを使用する必要があります。

- `enableDnsSupport` オプションを `true` (デフォルト値) に設定して、VPC が Amazon が提供する DNS サーバーを使用している場合、 OpenSearch サービスエンドポイントの解決は成功します。
- VPC がプライベート DNS サーバーを使用しており、サーバーがパブリック権限の DNS サーバーに到達して DNS ホスト名を解決できる場合、 OpenSearch サービスエンドポイントの解決も成功します。

IP アドレスは変更する可能性があるため、ドメインのエンドポイントを定期的に解決して常に正しいデータノードにアクセスできるようにすることが重要です。DNS 解決間隔を 1 分間に設定することが推奨されます。クライアントを使用している場合には、クライアント側の DNS キャッシュもクリアしていることも確認する必要があります。

パブリックアクセスから VPC アクセスに移行する

ドメインを作成するときは、このドメインがパブリックエンドポイントにあるか、あるいは VPC 内に存在するかを指定します。一度作成すると、選択を別のものに切り替えることはできません。代わりに、新規のドメインを作成して、手動で再度インデックスするか、またはデータを移行する必要があります。スナップショットは、データの移行に便利な手段です。スナップショットの作成と復元の詳細については、「the section called “インデックススナップショットの作成”」を参照してください。

VPC ドメインのアクセスポリシーについて

OpenSearch サービスドメインを VPC に配置すると、本質的に強力なセキュリティレイヤーが提供されます。パブリックアクセスのドメインを作成する場合、エンドポイントは次の形式になります。

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

「パブリック」ラベルが示すように、このエンドポイントはすべてのインターネット接続デバイスからアクセスできますが、このアクセスを制御でき、制御する必要があります。ウェブブラウザでエンドポイントにアクセスするときに Not Authorized メッセージを受信することがありますが、リクエストはドメインに届きます。

VPC アクセスを使用してドメインを作成する場合、このエンドポイントはパブリックエンドポイントに類似しています。

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

ウェブブラウザでエンドポイントにアクセスしようとすると、リクエストがタイムアウトする場合があります。基本的な GET リクエストを実行するときでも、コンピューターが VPC に接続していることが必要です。多くの場合、この接続では、VPN、Transit Gateway、マネージド型のネットワーク、またはプロキシサーバーを使用します。使用できるさまざまな形式の詳細については、Amazon VPC ユーザーガイドの「VPC の例」を参照してください 。開発に焦点を置いた例については、「the section called “VPC ドメインをテストする”」を参照してください。

この接続要件に加えて、VPC はセキュリティグループを使用したドメインへのアクセス管理を可能にします。多くのユースケースでは、このセキュリティ機能の組み合わせで十分となり、ドメインにオープンなアクセスポリシーを安心して適用できます。

オープンアクセスポリシーを使用して操作しても、インターネット上の誰でも OpenSearch サービスドメインにアクセスできるわけではありません。むしろ、リクエストが OpenSearch サービスドメインに到達し、関連付けられたセキュリティグループが許可する場合、ドメインはリクエストを受け入れることを意味します。唯一の例外は、きめ細かなアクセスコントロール、または IAM ロールを指定するアクセスポリシーを使用している場合です。これらの状況では、ドメインがリクエストを受信するには、セキュリティグループがそのリクエストを許可し、そしてこのリクエストが有効な認証情報で署名されていることが必要です。

> **Note**
>
> セキュリティグループは既に IP ベースのアクセスポリシーを適用しているため、VPC 内に存在する OpenSearch サービスドメインに IP ベースのアクセスポリシーを適用することはできません。パブリックエンドポイントを使用する場合、IP ベースのポリシーを引き続き利用できます。

開始する前に: VPC アクセスの前提条件

VPC と新しい OpenSearch サービスドメイン間の接続を有効にする前に、以下を実行する必要があります。

- 「VPC を作成する」

  VPC を作成するには、Amazon VPC コンソール、 AWS CLI、またはいずれかの AWS SDKsを使用できます。詳細については、Amazon VPC ユーザーガイドの「VPC の使用」を参照してください。VPC が既にある場合、このステップは省略できます。

- IP アドレスのリザーブ

  OpenSearch サービスでは、ネットワークインターフェイスを VPC のサブネットに配置することで、VPC をドメインに接続できます。各ネットワークインターフェースは 1 つの IP アドレスに関連付けられます。ネットワークインターフェースのためにサブネットで十分な数の IP アドレスをリザーブする必要があります。詳細については、「VPC サブネットで IP アドレスをリザーブする」を参照してください。

VPC ドメインをテストする

VPC のセキュリティを強化することで、ドメインへの接続や、基本的なテスト実行を行うことができます。すでに OpenSearch サービス VPC ドメインがあり、VPN サーバーを作成しない場合は、次のプロセスを試してください。

1. ドメインのアクセスポリシーに [きめ細かなアクセスコントロールのみを使用] を選択します。この設定は、テスト完了後にいつでも更新できます。
2. OpenSearch サービスドメインと同じ VPC、サブネット、およびセキュリティグループに Amazon Linux Amazon EC2 インスタンスを作成します。

   このインスタンスはテストを目的としており、必要な作業はわずかであるため、安価なインスタンスタイプ (例: `t2.micro`) を選択します。インスタンスにパブリック IP アドレスを割り当てたら、新しいキーペアを作成するか、既存のキーペアを選択します。新しいキーを作成する場合は、`~/.ssh` ディレクトリにダウンロードします。

   インスタンス作成の詳細については、「Amazon EC2 Linux インスタンスの開始方法」を参照してください。
3. インターネットゲートウェイを VPC に追加します。
4. VPC のルートテーブルで、新しいルートを追加します。[送信先] で、コンピュータのパブリック IP アドレスを含む CIDR ブロックを指定します。[ターゲット] で、先ほど作成したインターネットゲートウェイを指定します。

   たとえば、コンピュータが 1 つのみの場合は `123.123.123.123/32`、複数の場合には `123.123.123.0/24` を指定します。
5. セキュリティグループで、2 つのインバウンドルールを指定します。

タイプ	プロトコル	ポート範囲	ソース
SSH (22)	TCP (6)	22	*your-cidr-block*
HTTPS (443)	TCP (6)	443	*your-security-group-id*

   最初のルールでは、EC2 インスタンスに SSH 接続できます。2 つ目は、EC2 インスタンスが HTTPS 経由で OpenSearch サービスドメインと通信できるようにします。
6. ターミナルから、次のコマンドを実行します。

```
ssh -i ~/.ssh/your-key.pem ec2-user@your-ec2-instance-public-ip -N -L
 9200:vpc-domain-name.region.es.amazonaws.com:443
```

このコマンドは、EC2 インスタンスを介して https://localhost:9200 にリクエストを OpenSearch サービスドメインに転送する SSH トンネルを作成します。コマンドでポート 9200 を指定すると、ローカル OpenSearch インストールがシミュレートされますが、任意のポートが使用されます。 OpenSearch サービスはポート 80 (HTTP) または 443 (HTTPS) 経由の接続のみを受け入れます。

このコマンドではフィードバックは返らず、無限に実行されます。停止するには、`Ctrl + C` を押します。

7. ウェブブラウザで https://localhost:9200/_dashboards/ に移動します。セキュリティ例外の承認が必要な場合があります。

   https://localhost:9200curl、Postman、またはお気に入りのプログラミング言語を使用して、 にリクエストを送信することもできます。

> **Tip**
>
> 証明書の不一致が原因で curl のエラーが発生した場合は、`--insecure` フラグを試してください。

VPC サブネットで IP アドレスをリザーブする

OpenSearch サービスは、ネットワークインターフェイスを VPC のサブネット (複数のアベイラビリティーゾーンを有効にする場合は VPC の複数のサブネット) に配置することで、ドメインを VPC に接続します。各ネットワークインターフェースは 1 つの IP アドレスに関連付けられます。OpenSearch サービスドメインを作成する前に、ネットワークインターフェイスに対応するのに十分な数の IP アドレスが各サブネットで使用可能になっている必要があります。

基本的な計算式は次のとおりです。各サブネットで OpenSearch サービスが予約する IP アドレスの数は、データノードの数の 3 倍をアベイラビリティーゾーンの数で割ったものです。

例

- 3 つのアベイラビリティーゾーンで 1 つのドメインに 9 つのデータノードがある場合、サブネットあたりの IP アドレスの数は、9 x 3 ÷ 3 = 9 になります。

- 2 つのアベイラビリティーゾーンで 1 つのドメインに 8 つのデータノードがある場合、サブネットあたりの IP アドレスの数は、8 x 3 ÷ 2 = 12 になります。
- 1 つのアベイラビリティーゾーンで 6 つのドメインに 1 つのデータノードがある場合、サブネットあたりの IP アドレスの数は、6 x 3 ÷ 1 = 18 になります。

ドメインを作成すると、 OpenSearch Service は IP アドレスを予約し、ドメイン用にいくつか を使用し、残りは Blue/Green デプロイ用に予約します。Amazon EC2 コンソール の [ネットワークインターフェース] セクションからネットワークインターフェースと関連する IP アドレスを表示できます。説明列には、ネットワークインターフェイスが関連付けられている OpenSearch サービスドメインが表示されます。

> **Tip**
>
> OpenSearch サービスの予約済み IP アドレス専用のサブネットを作成することをお勧めします。専用サブネットを使用することで、他のアプリケーションやサービスとの重複を回避でき、将来的にクラスターをスケールする必要が生じた場合に追加の IP アドレスをリザーブできることを確保します。詳細については、「VPC でサブネットを作成する」を参照してください。

VPC アクセス用の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、ユーザーに代わってリソースを作成および管理できるように、サービスにアクセス許可を委任する一意のタイプの IAM ロールです。 OpenSearch サービスには、VPC にアクセスし、ドメインエンドポイントを作成し、VPC のサブネットにネットワークインターフェイスを配置するためのサービスにリンクされたロールが必要です。

OpenSearch サービスコンソールを使用して VPC 内にドメインを作成すると、 OpenSearch サービスによってロールが自動的に作成されます。この自動作成を成功させるには、`iam:CreateServiceLinkedRole` アクションへのアクセス許可が必要です。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許可」を参照してください。

OpenSearch サービスがロールを作成したら、IAM コンソールを使用してロール (`AWSServiceRoleForAmazonOpenSearchService`) を表示できます。

このロールのアクセス権限およびその削除方法の詳細については、「the section called “サービスリンクロールの使用”」を参照してください。

Amazon OpenSearch Service でのインデックススナップショットの作成

Amazon OpenSearch Service のスナップショットは、クラスターのインデックスと状態のバックアップです。状態には、クラスター設定、ノード情報、インデックス設定、シャードの割り当てなどが含まれます。

OpenSearch サービススナップショットには次の形式があります。

- 自動スナップショットは、クラスターの復元専用です。クラスターのステータスが赤くなった場合や、データが失われた場合に、ドメインを復元するために使用できます。詳細については、以下の「スナップショットの復元」を参照してください。 OpenSearch サービスは、自動スナップショットを事前設定された Amazon S3 バケットに追加料金なしで保存します。
- 手動スナップショットは、クラスターの復元、またはクラスター間でのデータの移行に使用します。手動スナップショットを開始する必要があります。これらのスナップショットは独自の Amazon S3 バケットに保存され、標準の S3 料金が適用されます。セルフマネージド OpenSearch クラスターのスナップショットがある場合は、そのスナップショットを使用して OpenSearch サービスドメインに移行できます。詳細については、「Amazon OpenSearch Service への移行」を参照してください。

すべての OpenSearch サービスドメインは自動スナップショットを作成しますが、頻度は次の点で異なります。

- OpenSearch または Elasticsearch 5.3 以降を実行しているドメインの場合、 OpenSearch Service は 1 時間ごとに自動スナップショットを取得し、そのうち最大 336 個を 14 日間保持します。時間単位のスナップショットは増分的な性質があるため、中断が少なくなります。また、ドメインの問題が発生した場合に、より最近のリカバリポイントを提供します。
- Elasticsearch 5.1 以前を実行しているドメインの場合、 OpenSearch Service は指定した時間内に毎日自動スナップショットを取得し、そのうち最大 14 個を保持し、30 日以上スナップショットデータを保持しません。

クラスターのステータスが赤になると、クラスターのステータスが維持されている間、すべての自動スナップショットが失敗します。2 週間以内にこの問題を解決しない場合、クラスターのデータは永遠に失われます。トラブルシューティングステップについては、「the section called “赤のクラスター状態”」を参照してください。

トピック

前提条件

スナップショットを手動で作成するには、IAM および Amazon S3 を使用して作業する必要があります。スナップショットの取得を試す前に、次の前提条件を満たしていることを確認します。

前提条件	説明
S3 バケット	OpenSearch サービスドメインの手動スナップショットを保存する S3 バケットを作成します。手順については、Amazon Simple Storage Service ユーザーガイドの「最初のバケットを作成する」を参照してください。 次の場所で使用するには、バケットの名前を覚えておいてください。 • IAM ロールにアタッチされた IAM ポリシーの `Resource` ステートメント • スナップショットリポジトリの登録に使用する Python クライアント (この方法を使用する場合) **Important** S3 Glacier ライフサイクルルールをこのバケットに適用しないでください。手動スナップショットは、S3 Glacier ストレージクラスをサポートしていません。
IAM ロール	サービスにアクセス許可を委任する IAM OpenSearch ロールを作成します。手順については、IAM ユーザーガイドの「IAM ロールの作成 (コンソール)」を参照し

前提条件	説明
	てください。この章の残りの部分では、このロールを TheSnapshotRole と呼びます。 IAM ポリシーのアタッチ 次のポリシーを TheSnapshotRole にアタッチして、S3 バケットへのアクセスを許可します。 `{ "Version": "2012-10-17", "Statement": [{ "Action": [ "s3:ListBucket" ], "Effect": "Allow", "Resource": [ "arn:aws:s3::: s3-bucket-name " ] }, { "Action": [ "s3:GetObject", "s3:PutObject", "s3:DeleteObject" ], "Effect": "Allow", "Resource": [ "arn:aws:s3::: s3-bucket-name /*" ] } ] }` ポリシーをロールにアタッチする方法については、IAM ユーザーガイドの「IAM ID 許可の追加」を参照してください。 信頼関係を編集する 次の例に示すように、 の信頼関係を編集TheSnapshotRole して Principal ステートメントで OpenSearch サービスを指定します。

前提条件	説明

	<pre>{
	  "Version": "2012-10-17",
	  "Statement": [{
	    "Sid": "",
	    "Effect": "Allow",
	    "Principal": {
	      "Service": "es.amazonaws.com"
	    },
	    "Action": "sts:AssumeRole"
	  }]
	
	}</pre>
	信頼関係を編集する手順については、IAM ユーザーガイドの「ロールの信頼ポリシーの変更」を参照してください。

前提条件	説明

アクセス許可	スナップショットリポジトリを登録するには、 TheSnapshotRole を OpenSearch サービスに渡す必要があります。さらに、es:ESHttpPut アクションへのアクセスも必要です。これらの両方の許可を付与するには、リクエストの署名に認証情報が使用されている IAM ロールまたはユーザーに次のポリシーをアタッチします。
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "iam:PassRole", "Resource": "arn:aws:iam:: 123456789012 :role/TheSnapshotRole " }, { "Effect": "Allow", "Action": "es:ESHttpPut", "Resource": "arn:aws:es: region:123456789012 :domain/domain-name /*" } ] }`
	
	ユーザーまたはロールが TheSnapshotRole を渡すための iam:PassRole 許可を持っていない場合、次のステップでリポジトリを登録しようとすると、次の一般的なエラーが発生することがあります。
	
	`$ python register-repo.py {"Message":"User: arn:aws:iam:: 123456789012 :user/MyUserAccount is not authorized to perform: iam:PassRole on resource: arn:aws:iam:: 123456789012 :role/TheSnapshotRole "}`

手動スナップショットレポジトリの登録

手動インデックススナップショットを作成する前に、スナップショットリポジトリを OpenSearch Service に登録する必要があります。この 1 回限りのオペレーションでは、「」で説明されているよ

うにTheSnapshotRole、 へのアクセスが許可されている認証情報を使用して AWS リクエストに署名する必要がありますthe section called “前提条件”。

ステップ 1: OpenSearch Dashboards でスナップショットロールをマッピングする (きめ細かなアクセスコントロールを使用している場合)

きめ細かなアクセスコントロールにより、リポジトリの登録時に追加のステップが導入されます。HTTP 基本認証を他のすべての目的で使用する場合でも、TheSnapshotRoleを渡すための `iam:PassRole` 許可を持っている IAM ロールまたはユーザーに `manage_snapshots` ロールをマップする必要があります。

1. OpenSearch サービスドメインの OpenSearch Dashboards プラグインに移動します。Dashboards エンドポイントは、 OpenSearch サービスコンソールのドメインダッシュボードにあります。
2. メインメニューから [セキュリティ]、[ロール] を選択し、[manage_snapshots] ロールを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。
4. TheSnapshotRole を渡すための許可を持っているロールの ARN を追加します。[Backend roles] (バックエンドロール) の下にロール ARN を配置します

   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. [マップ] を選択し、ユーザーまたはロールが [マッピングされたユーザー] の下に表示されていることを確認します。

ステップ 2: リポジトリを登録する

次の [Snapshots] (スナップショット) のタブには、スナップショットディレクトリの登録方法が記されています。手動スナップショットの暗号化に関するオプションと、新しいドメインに移行した後にスナップショットを登録する際のオプションについては、関連するタブを参照してください。

Snapshots

スナップショットリポジトリを登録するには、 OpenSearch サービスドメインエンドポイントに PUT リクエストを送信します。curl、サンプル Python クライアント、Postman、またはその他の方法を使用して署名付きリクエストを送信し、スナップショットリポジトリを登録できます。OpenSearch Dashboards コンソールで PUT リクエストを使用してリポジトリを登録することはできません。

リクエストは以下のような形式です。

```
PUT domain-endpoint/_snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "base_path": "my/snapshot/directory",
    "region": "region",
    "role_arn": "arn:aws:iam::123456789012:role/TheSnapshotRole"
  }
}
```

> **Note**
>
> リポジトリ名は「cs-」で始めることはできません。さらに、複数のドメインから同じリポジトリに書き込まない必要があります。リポジトリへの書き込みアクセス権を持つドメインは、1 つだけにする必要があります。

ドメインが Virtual Private Cloud (VPC) に存在する場合は、リクエストが正常にスナップショットレポジトリに登録するようにお使いのコンピュータが VPC に接続されていることが必要です。VPC へのアクセスはネットワーク構成によって異なりますが、VPN あるいは社内ネットワークへの接続を含む場合がよくあります。 OpenSearch サービスドメインにアクセスできることを確認するには、ウェブブラウザ`https://your-vpc-domain.region.es.amazonaws.com`で に移動し、デフォルトの JSON レスポンスを受け取ることを確認します。

Amazon S3 バケットがドメイン AWS リージョン 以外の にある場合は OpenSearch、リクエスト`"endpoint": "s3.amazonaws.com"`に パラメータを追加します。

Encrypted snapshots

現在、 AWS Key Management Service (KMS) キーを使用して手動スナップショットを暗号化することはできませんが、サーバー側の暗号化 (SSE) を使用して保護することはできます。

スナップショットリポジトリとして使用しているバケットの S3 管理のキーを使って SSE を有効にするには、PUT リクエストの `"settings"` ブロックに `"server_side_encryption": true` を追加します。詳細については、Amazon Simple Storage Service ユーザーガイドの「Amazon S3 管理の暗号化キー によるサーバー側の暗号化を使用したデータの保護」を参照してください。

または、スナップショットリポジトリとして使用する S3 バケットのサーバー側の暗号化に AWS KMS キーを使用することもできます。この方法を使用する場合は、S3 バケットの暗号化に使用される AWS KMS キーに必ずアクセスTheSnapshotRole許可を付与してください。詳細については、「AWS KMSのキーポリシー」を参照してください。

Domain migration

スナップショットリポジトリの登録は 1 回限りのオペレーションです。ただし、1 つのドメインから別のドメインに移行するには、古いドメインと新しいドメインで同じスナップショットレポジトリを登録する必要があります。リポジトリ名は任意です。

新しいドメインに移行する場合、または同じリポジトリを複数のドメインに登録する場合は、次のガイドラインを考慮してください。

- 新しいドメインにリポジトリを登録する場合は、`"readonly": true` を PUT リクエストの `"settings"` ブロックに追加します。この設定により、古いドメインのデータが誤って上書きされるのを防ぐことができます。リポジトリへの書き込みアクセス権を持つドメインは、1 つだけにする必要があります。
- データを別の AWS リージョンへ移行する場合 (us-east-2 にある古いドメインとバケットから us-west-2 にある新しいドメインへ等) は、`"region": "region"` を PUT ステートメントの `"endpoint": "s3.amazonaws.com"` に置き換えて、リクエストを再送信します。

サンプル Python クライアントの使用

Python クライアントは、シンプルな HTTP リクエストよりも自動化が容易で、再利用性が向上します。この方法を使用してスナップショットリポジトリを登録する場合は、次のサンプル Python コードを `register-repo.py` などの Python ファイルとして保存します。クライアントでは、AWS SDK for Python (Boto3)、リクエストおよび requests-aws4auth パッケージが必要になります。クライアントには、他のスナップショットオペレーションのコメントアウトされた例が含まれています。

サンプルコードで、次の変数を更新します: `host`、`region`、`path`、および `payload`。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = '' # domain endpoint
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository

path = '/_snapshot/my-snapshot-repo-name' # the OpenSearch API endpoint
url = host + path

payload = {
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "base_path": "my/snapshot/directory",
    "region": "us-west-1",
    "role_arn": "arn:aws:iam::123456789012:role/snapshot-role"
  }
}

headers = {"Content-Type": "application/json"}

r = requests.put(url, auth=awsauth, json=payload, headers=headers)

print(r.status_code)
print(r.text)

# # Take snapshot
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot'
# url = host + path
#
# r = requests.put(url, auth=awsauth)
#
# print(r.text)
#
# # Delete index
#
# path = 'my-index'
# url = host + path
#
# r = requests.delete(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshot (all indexes except Dashboards and fine-grained access control)
```

```
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {
#   "indices": "-.kibana*,-.opendistro_security,-.opendistro-*",
#   "include_global_state": False
# }
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
#
# # Restore snapshot (one index)
#
# path = '/_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {"indices": "my-index"}
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
```

手動スナップショットの作成

スナップショットは瞬時に取得されません。完了するまでに時間がかかり、クラスターの完全な point-in-time ビューを表すものではありません。スナップショットが進行中の間も、ドキュメントのインデックス作成や、クラスターへの他のリクエストを行うことはできますが、新しいドキュメントおよび既存のドキュメントの更新は一般的にスナップショットに含まれません。スナップショットには、スナップショット OpenSearch の開始時に存在していたプライマリシャードが含まれます。スナップショットのスレッドプールのサイズによっては、わずかな時間の違いで、スナップショットにさまざまなシャードが含まれることがあります。スナップショットのベストプラクティスについては、「the section called “スナップショットパフォーマンスの向上”」を参照してください。

スナップショットのストレージとパフォーマンス

OpenSearch スナップショットは増分です。つまり、最後に成功したスナップショット以降に変更されたデータのみを保存します。増分のみ保存されるため、スナップショットの取得頻度が高い場合でも低い場合でも、ディスク使用量を最小限に抑えることができます。つまり、スナップショットを 1 時間ごとに 1 週間 (合計 168 個) 取得すると、週末に 1 つのスナップショットを取得する場合よりも、使用するディスク容量は少なくなる場合があります。また、スナップショットの取得頻度が高くなるほど、完了までにかかる時間は短くなります。例えば、日次スナップショットの完了には 20 ~ 30 分かかる場合がありますが、時間単位のスナップショットは数分以内に完了することもあります。一部の OpenSearch ユーザーは、30 分ごとにスナップショットを作成します。

スナップショットを取得する

スナップショットを作成するときは、以下の情報を指定します。

- スナップショットリポジトリの名前
- スナップショットの名前

この章の例では、わかりやすく簡潔にするために、一般的な HTTP クライアントである curl を使用します。curl リクエストにユーザー名とパスワードを渡す方法については、「開始方法のチュートリアル」を参照してください。

アクセスポリシーでユーザーまたはロールを指定する場合は、スナップショットリクエストに署名する必要があります。curl の場合、`--aws-sigv4` のオプション (バージョン 7.75.0 以降) を使用できます。また、サンプル Python クライアントのコメントアウトされた例を使って、curl コマンドが使用しているのと同じエンドポイントに、署名された HTTP リクエストを行うこともできます

手動スナップショットを作成するには、次のステップを実施します。

1. 現在進行中のスナップショットは取得できません。確認するには、以下のコマンドを実行します。

   ```
   curl -XGET 'domain-endpoint/_snapshot/_status'
   ```

2. 次のコマンドを実行して、手動でスナップショットを取得します。

   ```
   curl -XPUT 'domain-endpoint/_snapshot/repository-name/snapshot-name'
   ```

特定のインデックスを包含または除外したり、他の設定を指定したりするには、リクエスト本文を追加します。リクエスト構造については、 OpenSearch ドキュメントの「スナップショットを作成する」を参照してください。

> **Note**
>
> スナップショットの作成に必要な時間は、 OpenSearch サービスドメインのサイズに応じて増加します。長時間実行しているスナップショット操作の場合は、504 GATEWAY_TIMEOUT エラーが発生する場合があります。通常、このエラーは無視して、オペレーションが正常に完了するのを待ってかまいません。次のコマンドを実行して、ドメインのすべてのスナップショットの状態を確認します。
>
> ```
> curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
> ```

スナップショットの復元

スナップショットを復元する前に、宛先ドメインがスタンバイ付きマルチ AZ を使用していないことを確認してください。スタンバイが有効になっている場合、復元オペレーションは失敗します。

> **Warning**
>
> インデックスのエイリアスを使用する場合は、インデックスを削除する前に、エイリアスへの書き込みリクエストを中止するか、エイリアスを別のインデックスに切り替える必要があります。書き込みリクエストを中止すると、次のシナリオの回避に有効です。
>
> 1. インデックスを削除すると、そのエイリアスも削除される。
> 2. 削除したばかりのエイリアスに対する障害のある書き込みリクエストにより、エイリアスと同じ名前で新しいインデックスが作成される。
> 3. 新しいインデックスとの命名の競合により、エイリアスを使用できなくなる。エイリアスを別のインデックスに切り替えた場合は、スナップショットから復元するときに `"include_aliases": false` を指定します。

スナップショットを復元するには

1. 復元するスナップショットを特定します。カスタムアナライザーパッケージや割り当て要件設定など、このインデックスのすべての設定がドメインと互換性があることを確認してください。すべてのスナップショットレポジトリを表示するには、次のコマンドを実行します。

```
curl -XGET 'domain-endpoint/_snapshot?pretty'
```

リポジトリを識別した後、次のコマンドを実行してすべてのスナップショットを表示します。

```
curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
```

> **Note**
>
> ほとんどの自動スナップショットは、cs-automated リポジトリに保存されます。ドメインで暗号化された保管中のデータは cs-automated-enc リポジトリに保存されます。検索する手動スナップショットレポジトリが表示されない場合は、ドメインにその登録をしたことを確認します。

2. (オプション) クラスターのインデックスとスナップショットのインデックスの間に名前の競合がある場合は、 OpenSearch サービスドメインの 1 つ以上のインデックスを削除または名前を変更します。インデックスのスナップショットを、同じ名前のインデックスがすでに含まれている OpenSearch クラスターに復元することはできません。

   インデックスの付けた名前の競合がある場合は、次のオプションがあります。

   - 既存の OpenSearch サービスドメインのインデックスを削除し、スナップショットを復元します。
   - スナップショットから復元する際、インデックスの名前を変更し、その後インデックスを再作成します。インデックスの名前を変更する方法については、 OpenSearch ドキュメントのこのリクエスト例を参照してください。
   - スナップショットを別の OpenSearch サービスドメインに復元します (手動スナップショットでのみ可能)。

   次のコマンドは、ドメイン内の既存のインデックスをすべて削除します。

```
curl -XDELETE 'domain-endpoint/_all'
```

ただし、すべてのインデックスを復元する予定がない場合は、インデックスを 1 つだけ削除できます。

```
curl -XDELETE 'domain-endpoint/index-name'
```

3. スナップショットを復元するには、次のコマンドを実行します。

```
curl -XPOST 'domain-endpoint/_snapshot/repository-name/snapshot-name/_restore'
```

OpenSearch Dashboards に対する特別なアクセス許可ときめ細かなアクセスコントロールインデックスにより、特に自動スナップショットから復元しようとすると、すべてのインデックスの復元が失敗する可能性があります。次の例では、1 つのインデックスである `my-index` を `2020-snapshot` から `cs-automated` スナップショットレポジトリで復元します。

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "my-index"}' \
-H 'Content-Type: application/json'
```

または、Dashboards ときめ細かなアクセスコントロールインデックスを除くすべてのインデックスを復元することもできます。

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "-.kibana*,-.opendistro*"}' \
-H 'Content-Type: application/json'
```

`rename_pattern` と `rename_replacement` のパラメータを使用すると、データを削除することなくスナップショットを復元することができます。これらのパラメータの詳細については、OpenSearch ドキュメントの「スナップショットの復元 API リクエストフィールド」と「リクエストの例」を参照してください。

> **Note**
>
> 関連するインデックスに対してすべてのプライマリシャードを使用できなかった場合、スナップショットが `PARTIAL` の `state` になっている可能性があります。この値は、1 つ以上のシャードからのデータが正しく保存されていないことを示します。部分スナップショット

からの復元も可能ですが、不足しているインデックスの復元に古いスナップショットの使用が必要になる場合もあります。

手動スナップショットの削除

次のコマンドを実行して、手動スナップショットを削除します。

```
DELETE _snapshot/repository-name/snapshot-name
```

Snapshot Management を用いたスナップショットの自動化

OpenSearch Dashboards でスナップショット管理 (SM) ポリシーを設定して、スナップショットの定期的な作成と削除を自動化できます。SM は、インデックスのグループのスナップショットを作成できますが、Index State Management は、インデックスごとに 1 つのスナップショットしか作成できません。Service で SM を使用するには OpenSearch 、独自の Amazon S3 リポジトリを登録する必要があります。リポジトリの登録手順については、「手動スナップショットレポジトリの登録」を参照してください。

SM 以前は、 OpenSearch Service は、デフォルトでオンになっている無料の自動スナップショット機能を提供していました。この機能は、サービスが管理する cs-* リポジトリにスナップショットを送信します。この機能を無効にする場合は、 AWS Supportまでお問い合わせください。

SM 機能の詳細については、 OpenSearch ドキュメントの「スナップショット管理」を参照してください。

SM は、現在は、複数のインデックスタイプでのスナップショット作成をサポートしていません。例えば、* を使って複数のインデックスでスナップショットを作成する際に、一部のインデックスがウォーム層にある場合、スナップショットの作成は失敗します。スナップショットに複数のインデックスタイプを含める必要がある場合は、SM でこのオプションがサポートされるまでは、ISM スナップショットアクションを使用します。

のアクセス許可を設定します。

以前の OpenSearch サービスドメインバージョンから 2.5 にアップグレードする場合、スナップショット管理のセキュリティ許可がドメインで定義されていない可能性があります。きめ細かいアクセスコントロールを使用して、ドメインでスナップショット管理を使用するには、管理者以外のユーザーがこのロールにマッピングされている必要があります。スナップショット管理ロールを手動で作成するときは、次の手順を実行します。

1. OpenSearch ダッシュボードで、セキュリティに移動し、アクセス許可 を選択します。
2. [アクショングループの作成] を選択し、以下のグループを設定します。

グループ名	許可
snapshot_management_full_access	• cluster:admin/opensearch/snapshot_management/* • cluster:admin/opensearch/notifications/feature/publish • cluster:admin/repository/* • cluster:admin/snapshot/*
snapshot_management_read_access	• cluster:admin/opensearch/snapshot_management/policy/get • cluster:admin/opensearch/snapshot_management/policy/search • cluster:admin/opensearch/snapshot_management/policy/explain • cluster:admin/repository/get • cluster:admin/snapshot/get

3. [ロール]、[ロールの作成] の順に選択します。
4. ロールに snapshot_management_role という名前を付けます。
5. [Cluster permissions] (クラスターのアクセス権限) で、snapshot_management_full_access または snapshot_management_read_access を選択します。
6. [作成] を選択します。
7. ロールを作成したら、任意のユーザー、またはスナップショットを管理するバックエンドロールにそれをマッピングします。

考慮事項

Snapshot Management を設定するときは、次の点を考慮します。

- リポジトリごとに使用できるポリシーは 1 つです。
- 1 つのポリシーで最大 400 のスナップショットを作成できます。

- この機能は、ドメインのステータスが赤色の場合、JVM の負荷が高い（85% 以上）場合、スナップショット機能が停止している場合は、実行されません。クラスターの全体的なインデックス作成と検索のパフォーマンスが影響を受けていると、SM も影響を受けている可能性があります。
- スナップショットのオペレーションは、前のオペレーションが完了した後に始まるため、1 つのポリシーでスナップショットのオペレーションが並行して実行されることはありません。
- 同じスケジュールのポリシーが複数存在すると、リソースが急増する可能性があります。ポリシーのスナップショット化したインデックスが重複している場合、シャードレベルのスナップショットオペレーションは順番にしか実行されないため、パフォーマンスの問題が連鎖的に発生する可能性があります。ポリシーがリポジトリを共有すると、そのリポジトリへの書き込みオペレーションが急増します。
- 特別なユースケースでない限り、スナップショットオペレーションの自動化は 1 時間に 1 回以下でスケジュールすることが推奨されます。

インデックスステート管理を用いたスナップショットの自動化

Index State Management (ISM) `snapshot` オペレーションを使用して、経過時間、サイズ、ドキュメント数の変化に基づいてインデックスのスナップショットを自動的にトリガーできます。ISM は、インデックスごとに 1 つのスナップショットが必要な場合に最適です。インデックスグループのスナップショットが必要な場合は、「Snapshot Management を用いたスナップショットの自動化」を参照してください。

OpenSearch サービスで SM を使用するには、独自の Amazon S3 リポジトリを登録する必要があります。`snapshot` オペレーションを使用した ISM ポリシーの例については、「サンプルポリシー」を参照してください。

スナップショットの Curator の使用

ISM がインデックスとスナップショットの管理に機能しない場合は、代わりに Curator を使用できます。複雑なクラスターでの管理タスクの単純化に役立つ、アドバンストフィルタリング機能が提供されています。pip を使用して Curator をインストールします。

```
pip install elasticsearch-curator
```

Curator はコマンドラインインターフェース (CLI) または Python API として使用できます。Python API を使用する場合は、従来の elasticsearch-py クライアントのバージョン 7.13.4 以前を使用する必要があります。opensearch-py クライアントは、サポートされていません。

CLI を使用する場合は、コマンドラインで認証情報をエクスポートして curator.yml を次のように設定します。

```
client:
  hosts: search-my-domain.us-west-1.es.amazonaws.com
  port: 443
  use_ssl: True
  aws_region: us-west-1
  aws_sign_request: True
  ssl_no_validate: False
  timeout: 60

logging:
  loglevel: INFO
```

Amazon OpenSearch Service ドメインのアップグレード

> **Note**
>
> OpenSearch および Elasticsearch バージョンのアップグレードは、サービスソフトウェアの更新とは異なります。サービスドメインのサービスソフトウェアの更新については、OpenSearch 「」を参照してくださいthe section called “サービスソフトウェア更新”。

Amazon OpenSearch Service は、 OpenSearch 1.0 以降、または Elasticsearch 5.1 以降を実行するドメインのインプレースアップグレードを提供します。Amazon Data Firehose や Amazon CloudWatch Logs などのサービスを使用してデータを OpenSearch サービスにストリーミングする場合は、移行 OpenSearch 前にこれらのサービスが新しいバージョンの をサポートしていることを確認してください。

トピック

- サポートされているアップグレードパス
- アップグレードの開始 (コンソール)
- アップグレードの開始 (CLI)
- アップグレードの開始 (SDK)
- 検証障害のトラブルシューティング
- アップグレードのトラブルシューティング

- スナップショットを使用してデータを移行する

サポートされているアップグレードパス

現在、 OpenSearch サービスは以下のアップグレードパスをサポートしています。

元のバージョン	目的のバージョン
OpenSearch 1.3 または 2.x	OpenSearch 2.x バージョン 2.3 には、次のような重要な変更点があります。 • バージョン 2.0 では、 `type`パラメータがすべての OpenSearch API エンドポイントから削除されました。詳細については、「breaking changes」(重要な変更点) を参照してください。 • ドメインに Elasticsearch 6.8 で最初に作成されたインデックス (ホット UltraWarm、またはコールド) が含まれている場合、それらのインデックスは OpenSearch 2.3 と互換性がありません。 バージョン 2.3 にアップグレードする前に、互換性のないインデックスを再インデックスする必要があります。互換性のないインデックス UltraWarm またはコールドインデックスの場合は、ホットストレージに移行し、データのインデックスを再作成してから、ウォームストレージまたはコールドストレージに戻します。または、インデックスが不要になった場合は削除することもできます。 最初にこれらの手順を実行することなく、誤ってドメインをバージョン 2.3 にアップグレードした場合、互換性のないインデックスを現在のストレージ階層から移行することができなくなります。唯一の選択肢はインデックスを削除することです。
OpenSearch 1.x	OpenSearch 1.x
Elasticsearch 7.x	Elasticsearch 7.x または OpenSearch 1.x

元のバージョン	目的のバージョン
	Important OpenSearch 1.x では、多数の重大な変更が導入されています。詳細については、「Amazon OpenSearch Service 名称変更」を参照してください。
Elasticsearch 6.8	Elasticsearch 7.x または OpenSearch 1.x **Important** Elasticsearch 7.0 および OpenSearch 1.0 には、多くの重大な変更が含まれています。インプレースアップグレードを開始する前に、6.x ドメインの手動スナップショットを作成し、テスト 7.x または OpenSearch 1.x ドメインで復元し、そのテストドメインを使用してアップグレードの潜在的な問題を特定することをお勧めします。 OpenSearch 1.0 の重大な変更については、「」を参照してくださいAmazon OpenSearch Service 名称変更。 Elasticsearch 6.x と同様に、インデックスに含めることができるマッピングタイプは1つだけですが、そのタイプには _doc という名前を付ける必要があります。その結果、特定の API では、リクエスト本文にマッピングタイプ (_bulk API など) が不要になりました。 新しいインデックスの場合、セルフホスト型 Elasticsearch 7.x および OpenSearch 1.x のデフォルトシャード数は 1.x 以降の Elasticsearch 7.x 上の OpenSearch サービスドメインでは、以前のデフォルトである 5 つが保持されます。
Elasticsearch 6.x	Elasticsearch 6.x

元のバージョン	目的のバージョン
Elasticsearch 5.6	Elasticsearch 6.x **Important** バージョン 6.x で作成されたインデックスは、複数のマッピングタイプをサポートしなくなりました。バージョン 5.x で作成されたインデックスは、6.x クラスターに復元された場合でも、複数のマッピングタイプをサポートします。クライアントコードによって作成されるマッピングタイプがインデックスごとに 1 種類のみであることを確認します。 Elasticsearch 5.6 から 6.x へのアップグレード中のダウンタイムを最小限に抑えるために、 OpenSearch サービスは `.kibana`インデックスを に再インデックスし `.kibana-6` 、 を削除し `.kibana`、 という名前のエイリアスを作成し `.kibana`、新しいインデックスを新しいエイリアスにマッピングします。
Elasticsearch 5.x	Elasticsearch 5.x

アップグレードプロセスには、3 つの手順が含まれます。

1. アップグレード前のチェック — OpenSearch サービスは、アップグレードをブロックし、これらのチェックが成功しない限り次のステップに進みません。
2. スナップショット – OpenSearch サービスは OpenSearch または Elasticsearch クラスターのスナップショットを取得し、スナップショットが成功しない限り次のステップに進みません。アップグレードが失敗した場合、 OpenSearch サービスはこのスナップショットを使用してクラスターを元の状態に復元します。詳細については、the section called “アップグレード後にダウングレードできない”を参照してください。
3. アップグレード – OpenSearch サービスによってアップグレードが開始されます。このアップグレードが完了するまでに 15 分から数時間かかる場合があります。アップグレードの一部またはすべて中に OpenSearch ダッシュボードが使用できなくなる場合があります。

アップグレードの開始 (コンソール)

アップグレードプロセスは元に戻せず、一時停止またはキャンセルすることはできません。アップグレード中、ドメインの設定を変更することはできません。アップグレードを開始する前に、続行するかどうかを再度確認します。これらの同じ手順を使用して、アップグレードを実際に開始することなくアップグレード前の確認を実行することができます。

クラスターに専用マスターノードがある場合、 OpenSearch アップグレードはダウンタイムなしで完了します。ない場合、クラスターがマスターノードを選択している間、アップグレード後数秒間応答しなくなることがあります。

ドメインを OpenSearch または Elasticsearch の新しいバージョンにアップグレードするには

1. ドメインの手動スナップショットを作成します。このスナップショットは、以前の OpenSearch バージョンの使用に戻る場合、新しいドメインで復元できるバックアップとして機能します。
2. https://aws.amazon.com にアクセスし、[Sign In to the Console] (コンソールにサインイン) を選択します。
3. 分析 で、Amazon OpenSearch サービス を選択します。
4. ナビゲーションペインの [Domains] (ドメイン) で、アップグレードするドメインを選択します。
5. [アクション] から [更新] を選択します。
6. アップグレードするバージョンを選択します。 OpenSearch バージョンにアップグレードする場合は、「互換モードを有効にする」オプションが表示されます。この設定を有効にすると、はバージョンを 7.10 として OpenSearch レポートし、Logstash などの Elasticsearch OSS クライアントとプラグインが Amazon OpenSearch Service で引き続き作業できるようにします。この設定は後で無効にできます
7. [アップグレード] を選択します。
8. ドメインダッシュボードのステータスをチェックし、アップグレードのステータスをモニタリングします。

アップグレードの開始 (CLI)

次のオペレーションを使用して、ドメインの正しいバージョンの OpenSearch または Elasticsearch を特定し、インプレースアップグレードを開始し、アップグレード前のチェックを実行し、進行状況を表示できます。

- `get-compatible-versions` (`GetCompatibleVersions`)

- `upgrade-domain` (`UpgradeDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

詳細については、 AWS CLI コマンドリファレンスおよび Amazon OpenSearch Service API リファレンスを参照してください。

アップグレードの開始 (SDK)

このサンプルでは、 のOpenSearchService低レベル Python クライアント AWS SDK for Python (Boto) を使用して、ドメインが特定のバージョンへのアップグレードの対象であるかどうかを確認し、アップグレードして、アップグレードステータスを継続的にチェックします。

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default Region.

DOMAIN_NAME = ''  # The name of the domain to upgrade
TARGET_VERSION = ''  # The version you want to upgrade the domain to. For example,
 OpenSearch_1.1

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)
client = boto3.client('opensearch', config=my_config)


def check_versions():
    """Determine whether domain is eligible for upgrade"""
    response = client.get_compatible_versions(
        DomainName=DOMAIN_NAME
    )
    compatible_versions = response['CompatibleVersions']
    for i in range(len(compatible_versions)):
        if TARGET_VERSION in compatible_versions[i]["TargetVersions"]:
            print('Domain is eligible for upgrade to ' + TARGET_VERSION)
```

```
            upgrade_domain()
            print(response)
        else:
            print('Domain not eligible for upgrade to ' + TARGET_VERSION)


def upgrade_domain():
    """Upgrades the domain"""
    response = client.upgrade_domain(
        DomainName=DOMAIN_NAME,
        TargetVersion=TARGET_VERSION
    )
    print('Upgrading domain to ' + TARGET_VERSION + '...' + response)
    time.sleep(5)
    wait_for_upgrade()


def wait_for_upgrade():
    """Get the status of the upgrade"""
    response = client.get_upgrade_status(
        DomainName=DOMAIN_NAME
    )
    if (response['UpgradeStep']) == 'UPGRADE' and (response['StepStatus']) ==
 'SUCCEEDED':
        print('Domain successfully upgraded to ' + TARGET_VERSION)
    elif (response['StepStatus']) == 'FAILED':
        print('Upgrade failed. Please try again.')
    elif (response['StepStatus']) == 'SUCCEEDED_WITH_ISSUES':
        print('Upgrade succeeded with issues')
    elif (response['StepStatus']) == 'IN_PROGRESS':
        time.sleep(30)
        wait_for_upgrade()


def main():
    check_versions()


if __name__ == "__main__":
    main()
```

検証障害のトラブルシューティング

OpenSearch または Elasticsearch バージョンアップグレードを開始すると、 OpenSearch Service は最初に一連の検証チェックを実行して、ドメインがアップグレードの対象であるかどうかを確認します。これらのチェックのいずれかが失敗した場合、ドメインを更新する前に修正する必要がある特定の問題を含む通知が表示されます。発生する可能性のある問題とその解決手順の一覧については、the section called “検証エラーのトラブルシューティング” を参照してください。

アップグレードのトラブルシューティング

インプレース アップグレードには、正常なドメインが必要です。ドメインがアップグレードの対象とならなかったり、さまざまな理由でアップグレードに失敗する場合があります。次の表は、最も一般的な問題を示しています。

問題	説明
オプションのプラグインはサポートされていません	オプションのプラグインを使用してドメインをアップグレードすると、OpenSearch サービスによってプラグインも自動的にアップグレードされます。そのため、ドメインのターゲットバージョンは、これらのオプションプラグインもサポートしている必要があります。ターゲットバージョンでは使用できないオプションのプラグインがドメインにインストールされている場合、アップグレードリクエストは失敗します。
ノードあたりのシャードが多すぎます	OpenSearch、および Elasticsearch の 7.x バージョンのデフォルト設定では、ノードあたり 1,000 シャード以下です。現在のクラスター内のノードがこの設定を超えると、 OpenSearch サービスではアップグレードできません。トラブルシューティングのオプションについては、「the section called “シャードの最大制限を超えました”」を参照してください。
処理中のドメイン	ドメインは、設定変更中です。オペレーション完了後、アップグレードの適格性を確認します。
赤のクラスター状態	クラスター内の 1 つ以上のインデックスが赤です。トラブルシューティングステップについては、「the section called “赤のクラスター状態”」を参照してください。
高いエラー率	クラスターはリクエストの処理を試みましたが、大量の 5xx エラーを返しています。この問題は通常、同時読み取りまたは書き込みリクエス

問題	説明
	トが多すぎることで起こります。クラスターへのトラフィックを減らすか、ドメインのスケーリングを検討してください。
スプリットブレイン	スプリットブレインとは、 クラスターに 1 つ以上のマスターノードがあり、決して独自に再結合することのない 2 つのクラスターに分割されているという意味です。専用マスターノードの推奨値を使用することで、スプリットブレインを回避できます。スプレッドブレインからの復旧方法については、AWS Support にお問い合わせください。
マスターノードが見つからない	OpenSearch サービスがクラスターのマスターノードを見つけられません。ドメインでマルチ AZ を使用している場合、アベイラビリティーゾーンの障害によってクラスターがクォーラムを失い、新しいマスターノードを選択できなくなる可能性があります。問題が自動的に解決されない場合、AWS Support にお問い合わせください。
保留中のタスクが多すぎる	マスターノードに高い負荷がかかっており、たくさんの保留中のタスクがあります。クラスターへのトラフィックを減らすか、ドメインのスケーリングを検討してください。
障害が発生したストレージボリューム	1 つ以上のノードのディスクボリュームが正常に機能していません。この問題は、高いエラー率や保留中のタスクが多すぎるなど、他の問題とともに頻繁に発生します。問題が単独で発生し、自動的に解決されない場合は、AWS Support にお問い合わせください。
KMS キーの問題	ドメインの暗号化に使用する KMS キーがアクセス不可能であるか、または存在しません。詳細については、「the section called “保管中のデータを暗号化するドメインのモニタリング”」を参照してください。
スナップショットを作成中	ドメインは現在、スナップショットを作成しています。スナップショットの完了後に、アップグレードの適格性を確認します。手動スナップショットのリポジトリが一覧表示できること、それらのリポジトリ内のスナップショットが一覧表示できること、および手動スナップショットが作成できることも確認してください。 OpenSearch サービスがスナップショットが進行中かどうかを確認できない場合、アップグレードは失敗する可能性があります。

問題	説明
スナップショットのタイムアウトまたは障害	アップグレード前のスナップショットの完了まで時間がかかりすぎたか、失敗しました。クラスター状態を確認して、再度お試しください。問題が解決しない場合は、AWS Support までお問い合わせください。
互換性のないインデックス	1 つ以上のインデックスがターゲットバージョンと互換性がありません。この問題は、古いバージョンの OpenSearch または Elasticsearch からインデックスを移行した場合に発生する可能性があります。インデックスを再作成して、再試行してください。
高いディスク使用率	クラスターのディスク使用率が 90% を超えています。データを削除またはドメインをスケーリングして、再度お試しください。
高い JVM 使用率	JVM メモリプレッシャーが 75% を超えています。クラスターへのトラフィックを削減するか、ドメインをスケーリングして、再度お試しください。
OpenSearch Dashboards エイリアスの問題	`.dashboards` はエイリアスとして既に設定されており、互換性のないインデックスにマッピングされています。おそらく以前のバージョンの OpenSearch Dashboards のインデックスです。インデックスを再作成して、もう一度試してください。
赤の Dashboards ステータス	OpenSearch Dashboards のステータスは赤です。アップグレードが完了したら、Dashboards を使用してみてください。赤いステータスが続く場合は、手動で解決してから再度お試しください。
クラスター間の互換性	アップグレードできるのは、アップグレード後にソースドメインとデスティネーションドメインの間で、クラスター間の互換性が維持されている場合のみです。アップグレードプロセス中に、互換性のない接続があれば特定されます。続行するには、リモートドメインをアップグレードするか、互換性のない接続を削除します。ドメインでレプリケーションがアクティブな場合は、接続を削除した後でレプリケーションを再開できないことに注意してください。

問題	説明
その他の OpenSearch サービスの問題	OpenSearch サービス自体に問題があると、ドメインがアップグレード対象外として表示される可能性があります。前述の条件がドメインに当てはまらず、問題が 1 日以上続く場合は、AWS Support にお問い合わせください。

スナップショットを使用してデータを移行する

インプレースアップグレードは、ドメインを新しいバージョン OpenSearch または Elasticsearch バージョンにアップグレードする、より簡単、迅速、信頼性の高い方法です。スナップショットは、Elasticsearch の 5.1 より前のバージョンから移行する必要がある場合や、まったく新しいクラスターに移行する場合に適しています。

次の表は、スナップショットを使用して、別の OpenSearch または Elasticsearch バージョンを使用するドメインにデータを移行する方法を示しています。スナップショットの作成と復元の詳細については、「the section called “インデックススナップショットの作成”」を参照してください。

元のバージョン	目的のバージョン	移行プロセス

OpenSearch 1.3 または 2.x	OpenSearch 2.x	1. OpenSearch 2.3 の重大な変更を確認して、インデックスまたはアプリケーションを調整する必要があるかどうかを確認します。
		2. 1.3 または 2.x ドメインの手動スナップショットを作成します。
		3. 元の 1.3 または 2.x ドメインよりも上位のバージョンである 2.x ドメインを作成します。
		4. 元のドメインのスナップショットを 2.x ドメインに復元します。オペレーション中に、新しい名前で `.opensearch` インデックスを復元することが必要になる場合があります。
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`

元のバージョン	目的のバージョン	移行プロセス

		<pre>"ignore_unavailable": true,
		"rename_pattern": ".opensearch",
		"rename_replacement": ".backup-opensearch"
		}</pre>
		次に、新しいドメインで .backup-opensearch のインデックスを再作成し、エイリアス .opensearch を付けることができます。_restore のデフォルトは false であるため、_restore REST 呼び出しには include_global_state は含まれません。その結果、テストドメインにはインデックステンプレートが含まれず、バックアップの完全な状態も含まれません。
		5. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス

OpenSearch 1.x	OpenSearch 1.x	1. 1.x ドメインの手動スナップショットを作成します。
		2. 元の 1.x ドメインよりも上位のバージョンの 1.x ドメインを作成します。
		3. 元のドメインのスナップショットを新しい 1.x ドメインに復元します。オペレーション中に、新しい名前で .opensearch インデックスを復元することが必要になる場合があります。
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".opensearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		次に、新しいドメインで .backup-opensearch のインデックスを再作成し、エイリアス .opensearch を付けることができます。_restore のデフォルトは false であるため、_restore REST 呼び出しには include_global_state は含まれません。その結果、テストドメインにはインデックステンプレートが含まれず、バックアップの完全な状態も含まれません。
		4. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス

Elasticsearch 6.x または 7.x	OpenSearch 1.x	1. OpenSearch 1.0 の重大な変更を確認して、インデックスまたはアプリケーションを調整する必要があるかどうかを確認します。
		2. Elasticsearch 7.x または 6.x ドメインの手動スナップショットを作成します。
		3. OpenSearch 1.x ドメインを作成します。
		4. Elasticsearch ドメインから OpenSearch ドメインにスナップショットを復元します。オペレーション中に、新しい名前で `.elasticsearch` インデックスを復元することが必要になる場合があります。
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		次に、新しいドメインで `.backup-opensearch` のインデックスを再作成し、エイリアス `.elasticsearch` を付けることができます。`_restore` のデフォルトは false であるため、`_restore` REST 呼び出しには `include_global_state` は含まれません。その結果、テストドメインにはインデックステンプレートが含まれず、バックアップの完全な状態も含まれません。
		5. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス

Elasticsearch 6.x	Elasticsearch 7.x	1. 7.0 の重大な変更を確認して、インデックスまたはアプリケーションを調整する必要があるかどうかを確認します。
		2. 6.x ドメインの手動スナップショットを作成します。
		3. 7.x ドメインを作成します。
		4. 元のドメインのスナップショットを 7.x ドメインに復元します。オペレーション中に、新しい名前で `.opensearch` インデックスを復元することが必要になる場合あります。
		
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-elasticsearch"`
		`}`
		
		次に、新しいドメインで `.backup-elasticsearch` のインデックスを再作成し、エイリアス `.elasticsearch` を付けることができます。`_restore` のデフォルトは false であるため、`_restore` REST 呼び出しには `include_global_state` は含まれません。その結果、テストドメインにはインデックステンプレートが含まれず、バックアップの完全な状態も含まれません。
		5. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス
Elasticsearch 6.x	Elasticsearch 6.8	1. 6.x ドメインの手動スナップショットを作成します。 2. 6.8 ドメインを作成します。 3. 元のドメインのスナップショットを 6.8 ドメインに復元します。 4. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。
Elasticsearch 5.x	Elasticsearch 6.x	1. インデックスやアプリケーションを調整する必要があるかどうかを確認するには、「6.0 の重要な変更」を参照します。 2. 5.x ドメインの手動スナップショットを作成します。 3. 6.x ドメインを作成します。 4. 元のドメインのスナップショットを 6.x ドメインに復元します。 5. 5.x ドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。
Elasticsearch 5.x	Elasticsearch 5.6	1. 5.x ドメインの手動スナップショットを作成します。 2. 5.6 ドメインを作成します。 3. 元のドメインのスナップショットを 5.6 ドメインに復元します。 4. 元のドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス
Elasticsearch 2.3	Elasticsearch 6.x	Elasticsearch 2.3 のスナップショットは 6.x とは互換性がありません。データを 2.3 から 6.x に直接移行するには、新しいドメインでインデックスを手動で再作成する必要があります。 または、この表の 2.3 から 5.x のステップに従い、新しい 5.x ドメインで `_reindex` オペレーションを実行して、2.3インデックスを5.x インデックスに変換してから、5.x から 6.x のステップに従うこともできます。
Elasticsearch 2.3	Elasticsearch 5.x	1. 5.0 の重大な変更を確認して、インデックスまたはアプリケーションを調整する必要があるかどうかを確認します。 2. 2.3 ドメインの手動スナップショットを作成します。 3. 5.x ドメインを作成します。 4. 2.3 ドメインのスナップショットを 5.x ドメインに復元します。 5. 2.3 ドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

元のバージョン	目的のバージョン	移行プロセス
Elasticsearch 1.5	Elasticsearch 5.x	Elasticsearch 1.5 のスナップショットは 5.x とは互換性がありません。データを 1.5 から 5.x に移行するには、新しいドメインでインデックスを手動で再作成する必要があります。 **Important** 1.5 スナップショットは 2.3 と互換性がありますが、 OpenSearch サービス 2.3 ドメインは `_reindex`オペレーションをサポートしていません。それらのインデックスを再作成できないため、1.5 ドメインで作成されたインデックスは、2.3 スナップショットから 5.x ドメインへの復元に失敗します。
Elasticsearch 1.5	Elasticsearch 2.3	1. 移行プラグインを使用して、バージョン 2.3 に直接アップグレードできるかどうかを確認します。移行前にデータの変更が必要になる場合もあります。 a. ウェブブラウザで、`http://`*domain-endpoint*`/_plugin/migration/` を開きます。 b. [今すぐ確認を実行] を選択します。 c. 結果を確認し、必要に応じて、手順に従いデータを変更します。 2. 1.5 ドメインの手動スナップショットを作成します。 3. 2.3 ドメインを作成します。 4. 1.5 ドメインのスナップショットを 2.3 ドメインに復元します。 5. 1.5 ドメインが不要であれば、削除します。削除しなかった場合は、ドメインの利用料金が引き続き発生します。

Amazon OpenSearch サービスのカスタムエンドポイントの作成

Amazon OpenSearch Service ドメインのカスタムエンドポイントを作成すると、 OpenSearch OpenSearch およびダッシュボード URL を参照しやすくなります。会社のブランドを含めることも、 easier-to-remember 標準のエンドポイントよりも短いエンドポイントを使用することもできます。

新しいドメインに切り替える必要がある場合は、新しい URL を指すように DNS を更新し、以前と同じエンドポイントを引き続き使用します。

カスタムエンドポイントを保護するには、 AWS Certificate Manager (ACM) で証明書を生成するか、独自の証明書をインポートします。

新しいドメインのカスタムエンドポイント

サービスコンソールまたは設定 API を使用して、 OpenSearch OpenSearch 新しいサービスドメインのカスタムエンドポイントを有効にできます。 AWS CLI

エンドポイントをカスタマイズするには (コンソール)

1. OpenSearch サービスコンソールから [ドメインの作成] を選択し、ドメインの名前を指定します。
2. [カスタムエンドポイント] の下で、[カスタムエンドポイントを有効にする] を選択します。
3. [カスタムホスト名] では、希望するカスタムエンドポイントのホスト名を入力します。ホスト名は、www.yourdomain.com や example.yourdomain.com のような完全修飾ドメイン名 (FQDN) である必要があります。

   > **Note**
   >
   > 「ワイルドカード証明書」がない場合は、カスタムドメインのサブドメイン用に新しい証明書を取得する必要があります。

4. [AWS 証明書] では、ドメインに使用する SSL 証明書を選択します。利用可能な証明書がない場合は、証明書を ACM にインポートするか、ACM を使用して証明書をプロビジョニングできます。詳細については、AWS Certificate Manager ユーザーガイドの「証明書の発行と管理」を参照してください。

> **Note**
>
> 証明書にはカスタムエンドポイント名があり、 OpenSearch サービスドメインと同じアカウントにある必要があります。証明書のステータスは「ISSUED」である必要があります。

- 残りのステップに従って、ドメインを作成し、[作成] を選択します。
- 処理が完了したら、ドメインを選択して、カスタムエンドポイントを表示します。

CLI または設定 API を使用するには、`CreateDomain` および `UpdateDomainConfig` オペレーションを使用します。詳細については、「AWS CLI コマンドリファレンス」と「Amazon OpenSearch サービス API リファレンス」を参照してください。

既存のドメインのカスタムエンドポイント

OpenSearch 既存のサービスドメインにカスタムエンドポイントを追加するには、[編集] を選択し、上記のステップ 2 ~ 4 を実行します。

次のステップ

OpenSearch サービスドメインのカスタムエンドポイントを有効にしたら、Amazon Route 53 (またはお好みの DNS サービスプロバイダー) で CNAME マッピングを作成できます。CNAME マッピングを作成すると、カスタムエンドポイントとそのサブドメインにトラフィックをルーティングできるようになります。このマッピングがないと、トラフィックをカスタムエンドポイントにルーティングできません。Route 53 でこのマッピングを作成する手順については、「新しいドメインの DNS ルーティングの設定」と「サブドメイン用の新しいホストゾーンの作成」を参照してください。他のプロバイダについては、そのドキュメントを参照してください。

カスタムエンドポイントが自動生成されたドメインエンドポイントを指す CNAME レコードを作成します。ドメインがデュアルスタックの場合は、サービスによって生成された 2 つのエンドポイントのいずれかに CNAME レコードを指定できます。カスタムエンドポイントのデュアルスタック機能は、CNAME レコードを指定するサービス生成エンドポイントによって異なります。カスタムエンドポイントホスト名は CNAME レコードの名前であり、ドメインエンドポイントホスト名は CNAME レコードの値です。

OpenSearchダッシュボードに SAML 認証を使用する場合は、新しい SSO URL で IdP を更新する必要があります。

Amazon Route 53 を使用してエイリアスレコードタイプを作成し、ドメインのカスタムエンドポイントをデュアルスタック検索エンドポイントに向けることができます。エイリアスレコードタイプを作成するには、デュアルスタック IP アドレスタイプを使用するようにドメインを設定する必要があります。これは Route 53 API を使用して行うことができます。

Route 53 API を使用してエイリアスレコードタイプを作成するには、ドメインのエイリアスターゲットを指定します。ドメインのエイリアスターゲットは、 OpenSearch サービスコンソールのカスタムエンドポイントセクションの Hosted Zone (デュアルスタック) フィールドで確認するか、`DescribeDomain` API を使用しての値をコピーすることで確認できます`DomainEndpointV2HostedZoneId`。

Amazon OpenSearch Service の Auto-Tune

Amazon OpenSearch Service の Auto-Tune では、OpenSearch クラスターのパフォーマンスと使用状況のメトリクスを使用して、メモリ関連の設定の変更を提案します。これには、キューとキャッシュのサイズ、ノードの Java 仮想マシン (JVM) 設定が含まれます。これらのオプションの変更により、クラスターの速度と安定性が向上します。

変更は、すぐにデプロイされるものもあれば、ドメインのオフピークウィンドウにスケジュールされるものもあります。いつでもデフォルトの OpenSearch Service 設定に戻すことができます。Auto-Tune がドメインのパフォーマンスメトリクスの収集および分析を行うと、通知ページの OpenSearch Service コンソールでレコメンデーションを確認できます。

Auto-Tune は、サポートされているインスタンスタイプを用いて、OpenSearch バージョン、または Elasticsearch 6.7 以降を実行しているドメインの商用AWS リージョンで利用可能です。

トピック

- 変更のタイプ
- Auto-Tune を有効または無効にする
- Auto-Tune の機能強化のスケジューリング
- オートチューンの変更の監視

変更のタイプ

Auto-Tune には、大きく分けて次の 2 つのカテゴリがあります。

- クラスターの実行時に適用される、無停止での変更。
- ブルー/グリーンデプロイを必要とする変更。これは、ドメインのオフピークウィンドウに適用されます。

Auto-Tune では、ドメインのパフォーマンスメトリクスに基づいて、次の設定の調整を提案できます。

タイプの変更	カテゴリ	説明
JVM ヒープサイズ	Blue/Green	デフォルトでは、OpenSearch Service は JVM ヒープ (最大 32 GiB のヒープサイズ) にインスタンスの RAM の 50% を使用します。 この割合を増やすと、OpenSearch のメモリは増えますが、オペレーティングシステムやその他のプロセスでは減ります。値を大きくすると、ガベージコレクションの一時停止の数は減りますが、一時停止の長さが増えます。
JVM 新世代設定	Blue/Green	JVM「新世代」設定は、マイナーガベージコレクションの頻度に影響します。マイナーコレクションを頻繁に行うと、メジャーコレクションの数が減り、一時停止することがあります。
キューサイズ	無停止	デフォルトでは、検索キューのサイズは `1000` であり、書き込みキューのサイズは `10000` です。リクエストを処理するために追加のヒープが利用可能な場合、Auto-Tune は検索キューと書き込みキューを自動的にスケーリングします。
キャッシュサイズ	無停止	フィールドキャッシュは、ヒープ上のデータ構造をモニタリングするので、キャッシュの使用をモニタリングすることが重要です。Auto-Tune は、フィールドデータキャッシュサイズをスケーリングして、メモリ不足や回路ブレーカーの問題を回避します。 シャードリクエストキャッシュはノードレベルで管理され、デフォルトの最大サイズはヒープの 1% です。Auto-Tune は、シャードリクエストキャッシュのサイズをスケーリングして、設定されたクラスターが処理できるよりも多くの検索およびインデックスリクエストを受け入れます。

タイプの変更	カテゴリ	説明
リクエストサイズ	無停止	デフォルトでは、実行中のリクエストの集計サイズが合計 JVM の 10% を超えている場合 (t2 インスタンスタイプでは 2%、t3.small では 1%)、既存のリクエストが完了するまで、OpenSearch はすべての新しい _search および _bulk をスロットルします。 Auto-Tune は、システム上で現在占有されている JVM の量に基づいて、このしきい値 (通常は 5 ~ 15%) を自動的に調整します。例えば、JVM のメモリ負荷が高い場合、Auto-Tune はしきい値を 5% に減らすことがあり、その時点で、クラスターが安定してしきい値が増加するまで、拒否が多くなる可能性があります。

Auto-Tune を有効または無効にする

OpenSearch Service では、新しいドメインで、デフォルトで Auto-Tune が有効になります。既存のドメインで Auto-Tune を有効または無効にするには、コンソールを使用することをお勧めします。これにより、プロセスが簡素化されます。Blue/Green デプロイを必要とする変更

現在、AWS CloudFormation を使用して Auto-Tune を有効または無効にすることはできません。

コンソール

既存のドメインで自動調整を有効にするには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) を開きます。
2. ナビゲーションペインの [ドメイン] で、ドメイン名を選択して [クラスター設定] を開きます。
3. Auto-Tune がまだ有効になっていない場合は、[オンにする] をクリックします。
4. 必要に応じて [オフピークウィンドウ] を選択し、ドメインに設定されたオフピークウィンドウにブルー/グリーンデプロイを要求する最適化をスケジュールします。詳細については、「the section called “Auto-Tune の機能強化のスケジューリング”」を参照してください。
5. [Save changes] (変更の保存) をクリックします。

CLI

AWS CLI を使用して Auto-Tune を有効にするには、UpdateDomainConfig リクエストを送信します。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options DesiredState=ENABLED
```

Auto-Tune の機能強化のスケジューリング

2023 年 2 月 16 日以前、Auto-Tune はメンテナンスウィンドウを使用して、ブルー/グリーンデプロイを必要とする変更をスケジュールしていました。現在ではメンテナンスウィンドウが廃止され、オフピークウィンドウが採用されています。これは、1 日の中でドメインのトラフィックが一般的に少なくなる 10 時間の時間ブロックです。オフピークウィンドウのデフォルトの開始時間は変更できますが、時間の長さを変更することはできません。

2023 年 2 月 16 日にオフピークウィンドウが導入される前に Auto-Tune メンテナンスウィンドウを有効にしていたドメインは、従来のメンテナンスウィンドウを中断することなく引き続き使用できます。ですが、既存のドメインを移行して、代わりにドメインメンテナンスのオフピークウィンドウを使用することをお勧めします。手順については、「the section called “Auto-Tune のメンテナンスウィンドウからの移行”」を参照してください。

コンソール

オフピークウィンドウに Auto-Tune アクションをスケジュールするには

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home) を開きます。
2. ナビゲーションペインの [ドメイン] で、ドメイン名を選択して [クラスター設定] を開きます。
3. [Auto-Tune] タブに進み、[編集] を選択します。
4. Auto-Tune がまだ有効になっていない場合は、[オンにする] をクリックします。
5. [オフピークウィンドウ中に最適化をスケジュールする] で、[オフピークウィンドウ] をクリックします。
6. [Save changes] (変更の保存) をクリックします。

CLI

設定されたオフピークウィンドウ中に Auto-Tune アクションをスケジュールするようにドメインを設定するには、UpdateDomainConfig リクエストに `UseOffPeakWindow` を含めてください。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=null
```

オートチューンの変更の監視

Amazon CloudWatch でオートチューンの統計情報を監視できます。メトリクスの一覧については、「the section called “Auto-Tune メトリクス”」を参照してください。

OpenSearch Service はオートチューンイベントを Amazon EventBridge に送信します。EventBridge を使用して、イベントの受信時に E メールを送信したり、特定のアクションを実行したりするルールを設定できます。EventBridge に送信される各オートチューンイベントのフォーマットを見るには、「the section called “Auto-Tune イベント”」を参照してください。

Amazon OpenSearch Service ドメインのタグ付け

タグを使用すると、Amazon OpenSearch Service ドメインに任意の情報を割り当てて、その情報を分類してフィルタリングできます。タグは、 OpenSearch サービスドメインを定義して関連付けるキーと値のペアです。これらのタグを使用して、類似のタグ付けされたリソースの費用をグループ化することでコストを追跡できます。 AWS タグに意味論的意味を適用しません。タグは単なる文字列として解釈されます。タグには、次の要素があります。

タグ要素	説明	必須
タグキー	タグキーは、タグ名です。キーは、アタッチされている OpenSearch サービスドメインで一意である必要があります。タグキーと値の基本的な制限のリストについては、「ユーザー定義タグの制限」を参照してください。	はい
タグ値	タグ値は、タグの文字列値です。タグ値は `null` を指定できます。また、タグセット内で一意である必要はありません。例えば、project/Trinity と cost-center/Trinity のタグセット内に 1 つのキーと値のペアを	いいえ

タグ要素	説明	必須
	使用できます。タグキーと値の基本的な制限のリストについては、「ユーザー定義タグの制限」を参照してください。	

各 OpenSearch サービスドメインにはタグセットがあり、その OpenSearch サービスドメインに割り当てられたすべてのタグが含まれています。 AWS は OpenSearch サービスドメインにタグを自動的に割り当てません。タグセットには、0～50 個のタグを含めることができます。ドメインに追加したタグのキーが既存のタグのキーと同じ場合、既存の値は新しい値によって上書きされます。

タグ付けの例

キーを使用してカテゴリを定義し、タグ値をそのカテゴリの項目にすることができます。例えば、 のタグキー`project`と のタグ値を定義して`Salix`、 OpenSearch サービスドメインが Salix プロジェクトに割り当てられていることを示すことができます。タグを使用して、`environment=test`や などのキーを使用して、 OpenSearch サービスドメインがテスト用または本番稼働用として指定することもできます`environment=production`。 OpenSearch サービスドメインに関連付けられているメタデータの追跡が簡単になるように、一貫した一連のタグキーを使用してください。

タグを使用して、自分のコスト構造を反映するように AWS 請求書を整理することもできます。これを行うには、サインアップして、タグキー値が含まれた AWS アカウント 請求書を取得します。その後、同じタグキー値を持つリソースに従って請求情報を整理し、結合したリソースのコストを確認します。例えば、複数の OpenSearch サービスドメインにキーと値のペアをタグ付けし、請求情報を整理して、複数のサービスにわたる各ドメインの合計コストを確認できます。詳細については、https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/cost-alloc-tags.html 請求情報とコスト管理の「AWS コスト配分タグの使用」を参照してください。

> **Note**
>
> タグは承認用にキャッシュに格納されます。このため、 OpenSearch サービスドメインのタグの追加と更新には数分かかることがあります。

タグの操作 (コンソール)

コンソールは、ドメインにタグを付ける最も簡単な方法です。

タグを作成するには (コンソール)

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。
2. 分析 で、Amazon OpenSearch Service を選択します。
3. タグを追加するドメインを選択し、[タグ] タブに移動します。
4. [管理] を選択して、[新しいタグを追加] を選択します。
5. タグキーとオプションの値を入力します。
6. [保存] を選択します。

タグを削除するには、同じ手順に従って、[タグを管理] ページで [削除] を選択します。

タグを操作するコンソールを使用する方法の詳細については、AWS マネジメントコンソール入門ガイドの「タグエディター」を参照してください。

タグの操作 (AWS CLI)

--add-tags コマンド AWS CLI で を使用してリソースタグを作成できます。

[Syntax] (構文)

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

パラメータ	説明
`--arn`	タグがアタッチされている OpenSearch サービスドメインの Amazon リソース名。
`--tag-list`	スペースで区切られたキーと値のペアの以下の形式のセット: `Key=<key>,Value=<value>`

例

次の例では、[ログ] ドメイン用に 2 つのタグを作成します。

```
aws opensearch add-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-list
 Key=service,Value=OpenSearch Key=instances,Value=m3.2xlarge
```

--remove-tags コマンドを使用して、 OpenSearch サービスドメインからタグを削除できます。

構文

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

パラメータ	説明
`--arn`	タグがアタッチされている OpenSearch サービスドメインの Amazon リソースネーム (ARN)。
`--tag-keys`	OpenSearch サービスドメインから削除するスペース区切りのキーと値のペアのセット。

例

次の例は、[ログ] ドメインから前述の例で作成した 2 つのタグを削除します。

```
aws opensearch remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-keys service instances
```

--list-tags コマンドを使用して、 OpenSearch サービスドメインの既存のタグを表示できます。

[Syntax] (構文)

```
list-tags --arn=<domain_arn>
```

パラメータ	説明
`--arn`	タグがアタッチされている OpenSearch サービスドメインの Amazon リソースネーム (ARN)。

例

次の例は、[ログ] ドメインのすべてのリソースタグをリスト表示します。

```
aws opensearch list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

タグの操作 (AWS SDKs

AWS SDKs (Android および iOS SDKs) は、Amazon OpenSearch Service API リファレンス で定義されているすべてのアクションをサポートします。これには、AddTags、ListTags、および RemoveTagsオペレーションが含まれます。 AWS SDKs「 AWS Software Development Kits」を参照してください。

Python

この例では、AWS SDK for Python (Boto) のOpenSearchService低レベル Python クライアントを使用して、ドメインにタグを追加し、ドメインにアタッチされたタグを一覧表示し、ドメインからタグを削除します。DOMAIN_ARN、TAG_KEY および TAG_VALUE の値を指定する必要があります。

```
import boto3
from botocore.config import Config  # import configuration

DOMAIN_ARN = ''  # ARN for the domain. i.e "arn:aws:es:us-east-1:123456789012:domain/
my-domain
TAG_KEY = ''  # The name of the tag key. i.e 'Smileyface'
TAG_VALUE = ''  # The value assigned to the tag. i.e 'Practicetag'

# defines the configurations parameters such as region

my_config = Config(region_name='us-east-1')
client = boto3.client('opensearch', config=my_config)


# defines the client variable

def addTags():
    """Adds tags to the domain"""

    response = client.add_tags(ARN=DOMAIN_ARN,
                               TagList=[{'Key': TAG_KEY,
                                         'Value': TAG_VALUE}])

    print(response)


def listTags():
    """List tags that have been added to the domain"""
```

```
    response = client.list_tags(ARN=DOMAIN_ARN)
    print(response)


def removeTags():
    """Remove tags that have been added to the domain"""

    response = client.remove_tags(ARN=DOMAIN_ARN, TagKeys=[TAG_KEY])

    print('Tag removed')
    return response
```

Amazon OpenSearch Service ドメインに対する管理アクションを実行する

Amazon OpenSearch Service には、ドメインに関する問題のトラブルシューティングが必要な場合にきめ細かな制御を提供する管理オプションがいくつか用意されています。これらのオプションには、データノードで OpenSearch プロセスを再起動する機能や、データノードを再起動する機能が含まれます。

OpenSearch サービスはノードのヘルスパラメータをモニタリングし、異常が発生した場合は、ドメインを安定させるために是正措置を講じます。ノードでプロセスを再起動 OpenSearchし、ノード自体を再起動する管理オプションを使用すると、これらの緩和アクションの一部を制御できます。

これらのアクションを実行するには AWS Management Console、 AWS CLI、、または AWS SDK を使用できます。以下のセクションでは、コンソールでこれらのアクションを実行する方法について説明します。

ノードで OpenSearch プロセスを再起動する

ノードで OpenSearch プロセスを再起動するには

1. OpenSearch のサービスコンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。作業するドメインの名前を選択します。
3. ドメインの詳細ページが開いたら、[インスタンスの正常性] タブに移動します。
4. [データノード] で、プロセスを再起動するノードの横にあるボタンを選択します。
5. Actions ドロップダウンを選択し、Restart OpenSearch/Elasticsearch process を選択します。

6. モーダルで [確認] を選択します。
7. 開始したアクションのステータスを確認するには、ノードの名前を選択します。ノードの詳細ページが開いたら、ノード名の下にある [イベント] タブを選択して、そのノードに関連するイベントのリストが表示します。

データノードを再起動する

データノードを再起動するには

1. OpenSearch のサービスコンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。作業するドメインの名前を選択します。
3. ドメインの詳細ページが開いたら、[インスタンスの正常性] タブに移動します。
4. [データノード] で、プロセスを再起動するノードの横にあるボタンを選択します。
5. [アクション] ドロップダウンを選択し、[ノードの再起動] を選択します。
6. モーダルで [確認] を選択します。
7. 開始したアクションのステータスを確認するには、ノードの名前を選択します。ノードの詳細ページが開いたら、ノード名の下にある [イベント] タブを選択して、そのノードに関連するイベントのリストが表示します。

ノード上の Dashboard または Kibana プロセスを再起動する

ノード上の Dashboard または Kibana プロセスを再起動するには

1. OpenSearch のサービスコンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。作業するドメインの名前を選択します。
3. ドメインの詳細ページが開いたら、[インスタンスの正常性] タブに移動します。
4. [データノード] で、プロセスを再起動するノードの横にあるボタンを選択します。
5. [アクション] ドロップダウンを選択し、[Dashboard/Kibana プロセスを再起動] を選択します。
6. モーダルで [確認] を選択します。
7. 開始したアクションのステータスを確認するには、ノードの名前を選択します。ノードの詳細ページが開いたら、ノード名の下にある [イベント] タブを選択して、そのノードに関連するイベントのリストが表示します。

制限事項

管理オプションには以下の制限があります。

- 管理オプションは Elasticsearch バージョン 7.x 以降でサポートされています。
- 管理オプションは、スタンバイ付きマルチ AZ のドメインをサポートしていません。
- OpenSearch および Elasticsearch プロセスの再起動とデータノードの再起動は、3 つ以上のデータノードを持つドメインでサポートされています。
- Dashboards と Kibana のプロセスサポートは、2 つ以上のデータノードがあるドメインでサポートされます。
- ノードで OpenSearch プロセスを再起動したり、ノードを再起動したりするには、ドメインが赤色の状態ではなく、すべてのインデックスにレプリカが設定されている必要があります。

Amazon S3 での Amazon OpenSearch Service ダイレクトクエリの使用 Amazon S3

Amazon OpenSearch Service の直接クエリを使用して、Amazon S3 内のデータをクエリできます。Amazon OpenSearch Service は、サービスを切り替えることなく、Amazon S3 の運用ログと Amazon S3 に基づくデータレイクを分析する方法としてAmazon S3 との直接クエリ統合を提供します。クラウドオブジェクトストア内のデータを分析できるようになりました。同時に、 OpenSearch サービスの運用分析と視覚化を使用できるようになりました。

Amazon S3 を使用した直接クエリを使用すると、複雑な ETL パイプラインを構築したり、OpenSearch サービスと Amazon S3 ストレージの両方でデータを複製したりするコストが発生する必要がなくなります。また、事前定義済みのダッシュボードを含む一般的なログタイプテンプレートの統合をインストールし、そのログタイプに合わせてデータアクセラレーションを設定することもできます。テンプレートには、VPC フローログ、AWS CloudTrail ログ、Amazon S3 ログが含まれます。アクセラレーションには、インデックスのスキップ、マテリアライズドビュー、およびカバーされたインデックスが含まれます。

トピック

料金

直接クエリの作成と処理に使用される既存の OpenSearch サービスと Amazon S3 リソースに対して料金が発生します。Amazon S3 に送信されるクエリは、請求可能なコンピューティングを使用し、OpenSearch 1 時間あたりのコンピューティングユニット (OCUsとして表示されます。

Amazon S3 を使用した直接クエリには、インタラクティブクエリとアクセラレーションの 2 つのタイプがあります。インタラクティブクエリは、Amazon S3 内のデータの分析を実行します。新しいクエリを実行すると、 OpenSearch サービスは少なくとも 3 分間続く新しいセッションを開始します。 OpenSearch サービスはセッションをアクティブに保ち、後続のクエリが迅速に実行されるようにします。アクセラレーションクエリは、コンピューティングを使用して OpenSearch Service のインデックスを維持します。これらのクエリは通常、インタラクティブクエリの実行を高速化するために、さまざまな量のデータを OpenSearch サービスに取り込むため、時間がかかります。

詳細については、「Amazon OpenSearch Service の料金」を参照してください。

制限事項

Amazon S3 を使用した OpenSearch サービスダイレクトクエリには、次の制限が適用されます。

- OpenSearch サービスダイレクトクエリをサポートするには、 OpenSearch ドメインがバージョン 2.13 以降である必要があります。
- OpenSearch Serverless では使用できません。
- OpenSearch ドメイン と は同じ にある AWS Glue Data Catalog 必要があります AWS アカウント。Amazon S3 バケットは別の アカウント (IAM ポリシーに条件を追加する必要があります) にあることができますが、ドメイン AWS リージョン と同じ にある必要があります。
- 一部のデータ型はサポートされていません。サポートされるデータ型は、Parquet、CSV、および JSON に限定されます。
- OpenSearch Amazon S3 を使用したサービスダイレクトクエリは、Query Workbench から生成された Spark テーブルのみをサポートします。 AWS Glue Data Catalog または Athena 内で生成されたテーブルは、アクセラレーションを維持し、インデックスを最新の状態に保つために必要な Spark ストリーミングではサポートされていません。
- クエリの前にデータをフラット化するか、SQL in OpenSearch Service を使用してネストされた列を専用列に変更する必要があります。
- 列が欠落している場合は、`COALESCE` SQL 関数を使用して結果を返す必要がある場合があります。
- データ構造が変更された場合は、テーブルの更新と既存のアクセラレーションが必要です AWS Glue 。
- OpenSearch インスタンスタイプには、インスタンスタイプ (10 v. 100) に応じてネットワークペイロードの制限があります。
- AWS CloudFormation テンプレートはまだサポートされていません。

レコメンデーション

ダイレクトクエリを使用する場合は、次の操作を行うことをお勧めします。

- 年、月、日、時間のパーティション形式を使用してデータを Amazon S3 に取り込み、クエリを高速化します。
- クエリの制限を使用して、データが過度にプルバックされないようにします。
- Index State Management (該当する場合) を使用して、マテリアライズドビューとカバーインデックスのストレージを維持します。
- 不要になったアクセラレーションジョブとインデックスを削除します。
- スキップインデックスを構築するときは、ブルームフィルターを使用して高基数を実現し、範囲が大きい場合は最小/最大を使用します。高基数フィールドで設定された値を使用することをお勧めします。
- リファレンスガイドを使用して Amazon S3 にデータをエクスポートします。、CloudFront、CloudTrailおよび Elastic Load Balancing などの AWS ログを使用できます。

クォータ

アカウントには、Amazon S3 での OpenSearch Service direct query に関連する以下のクォータがあります。 Amazon S3 クエリを開始するたびに、 OpenSearch サービスはセッションを開き、少なくとも 10 分間存続します。これにより、後続のクエリにおけるセッション開始時間がなくなり、クエリのレイテンシーが低減されます。

説明	最大	オーバーライド可能
ドメインあたりの接続数	10	[Yes (はい)]
ドメインあたりのデータソース	20	あり
ドメインあたりのインデックス	5	あり
データソースあたりの同時実行セッション	10	[Yes (はい)]
クエリあたりの最大 OCU	60	あり
最大クエリ実行時間 (分)	30	あり

説明	最大	オーバーライド可能
アクセラレーションあたりの最大 OCUs	20	あり
最大エフェメラルストレージ	20	あり

サポートされるリージョン

Amazon S3 で OpenSearch サービスダイレクトクエリが利用できるリージョンは、アジアパシフィック (香港) 、アジアパシフィック (ムンバイ) 、アジアパシフィック (ソウル) 、アジアパシフィック (シンガポール) 、アジアパシフィック (シドニー) 、アジアパシフィック (東京) 、カナダ (中部) 、欧州 (フランクフルト) 、欧州 (アイルランド) 、欧州 (ストックホルム) 、米国東部 (バージニア北部) 、米国東部 (オハイオ) 、米国西部 (オレゴン) です。

Amazon S3 との Amazon OpenSearch Service データソース統合の作成 Amazon S3

AWS Management Console または API を使用して、 OpenSearch サービス用の新しい Amazon S3 ダイレクトクエリデータソースを作成できます。新しいデータソースはそれぞれ AWS Glue Data Catalog 、 を使用して Amazon S3 バケットを表すテーブルを管理します。

トピック

前提条件

データソースを作成する前に、 OpenSearch バージョン 2.13 以降のドメインが必要です。この設定手順については、「」を参照してくださいthe section called “ OpenSearch サービスドメインの作成”。

新しいダイレクトクエリデータソースを設定する

AWS Management Console または OpenSearch Service API を使用して、ドメインでダイレクトクエリデータソースを設定できます。

AWS Management Console

1. で Amazon OpenSearch Service コンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。
3. 新しいデータソースを設定するドメインを選択します。選択すると、ドメインの詳細ページが開きます。一般的なドメインの詳細の下にある [接続] タブを選択し、[ダイレクトクエリ] セクションを見つけます。
4. [作成] を選択します。
5. データソースの作成ページで、新しいデータソースの名前を入力します。[データソースのタイプ] で、[Amazon S3] を選択します。 AWS Glue Data Catalog および Amazon S3 でアクセスできるものに制限がある既存の IAM ロールを選択します。
6. [作成] を選択します。これにより、Dashboards URL OpenSearch を含むデータソースの詳細画面が開きます。この URL に移動して、次のステップを完了できます。

OpenSearch サービス API

AddDataSource API オペレーションを使用して、ドメインに新しいデータソースを作成します。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource

{
   "DataSourceType": {
        "s3GlueDataCatalog": {
            "RoleArn": "arn:aws:iam::account-id:role/Admin"
        }
    }
   "Description": "data-source-description",
   "Name": "my-data-source"
}
```

次のサンプルポリシーは、データソースの作成と管理に必要な最小特権の許可を示しています。s3:* や AdminstratorAccessポリシーなど、より広範なアクセス許可がある場合、これらのアクセス許可にはサンプルポリシーの最小特権のアクセス許可が含まれます。

統合には、Amazon S3 と に書き込むためのアクセスが必要です AWS Glue Data Catalog。Amazon S3 では、アクセラレーションを構築するときにチェックポイントの場所を維持するために書き込みアクセスが必要です。の場合 AWS Glue Data Catalog、 OpenSearch サービス内からの統合に必要なデータベース、テーブル、パーティションを管理するための書き込みアクセスが必要です。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"HttpActionsForOpenSearchDomain",
         "Effect":"Allow",
         "Action":"es:ESHttp*",
         "Resource":"arn:aws:es:<region>:<account>:domain/<domain_name>/*"
      },
      {
         "Sid":"AmazonOpenSearchS3GlueDirectQueryReadAllS3Buckets",
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:ListBucket"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"<account>"
            }
         },
         "Resource":"*"
      },
      {
         "Sid":"AmazonOpenSearchDirectQueryGlueCreateAccess",
         "Effect":"Allow",
         "Action":[
            "glue:CreateDatabase",
            "glue:CreatePartition",
            "glue:CreateTable",
            "glue:BatchCreatePartition"
         ],
         "Resource":"*"
```

```
      },
      {
         "Sid":"AmazonOpenSearchS3GlueDirectQueryModifyAllGlueResources",
         "Effect":"Allow",
         "Action":[
            "glue:DeleteDatabase",
            "glue:DeletePartition",
            "glue:DeleteTable",
            "glue:GetDatabase",
            "glue:GetDatabases",
            "glue:GetPartition",
            "glue:GetPartitions",
            "glue:GetTable",
            "glue:GetTableVersions",
            "glue:GetTables",
            "glue:UpdateDatabase",
            "glue:UpdatePartition",
            "glue:UpdateTable",
            "glue:BatchGetPartition",
            "glue:BatchDeletePartition",
            "glue:BatchDeleteTable"
         ],
         "Resource":[
            "arn:aws:glue:us-east-1:<account>:table/*",
            "arn:aws:glue:us-east-1:<account>:database/*",
            "arn:aws:glue:us-east-1:<account>catalog"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"<account>"
            }
         }
      },
      {
         "Sid":"ReadAndWriteActionsForS3CheckpointBucket",
         "Effect":"Allow",
         "Action":[
            "s3:ListMultipartUploadParts",
            "s3:DeleteObject",
            "s3:GetObject",
            "s3:PutObject",
            "s3:GetBucketLocation",
            "s3:ListBucket"
         ],
```

```
         "Condition":{
            "StringEquals":{
               "aws:ResourceAccount":"<account>"
            }
         },
         "Resource":[
            "arn:aws:s3:::<checkpoint_bucket_name>",
            "arn:aws:s3:::<checkpoint_bucket_name>/*"
         ]
      }
   ]
}
```

異なるアカウントで Amazon S3 バケットをサポートするには、Amazon S3 ポリシーに条件を含めて、適切なアカウントを追加する必要があります。

```
"Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "{{accountId}}"
                }
```

また、ロールには、ターゲット ID を指定する次の信頼ポリシーが必要です。

```
{
    "Version":"2012-10-17",
    "Statement":[
       {
          "Effect":"Allow",
          "Principal":{
             "Service": "directquery.opensearchservice.amazonaws.com"
          },
          "Action":"sts:AssumeRole"
       }
     ]
}
```

ロールを作成する手順については、「カスタム信頼ポリシーを使用したロールの作成」を参照してください。

OpenSearch Service できめ細かなアクセスコントロールを有効にしている場合、データソース用に新しい OpenSearchきめ細かなアクセスコントロールロールが自動的に作成されます。新しいきめ細

かなアクセスコントロールロールの名前は になりますAWS OpenSearchDirectQuery *<name of data source>*。

デフォルトでは、ロールは直接クエリデータソースインデックスにのみアクセスできます。データソースへのアクセスを制限または許可するようにロールを設定できますが、このロールのアクセスは調整しないことをお勧めします。データソースを削除すると、このロールは削除されます。これにより、他のユーザーがロールにマッピングされている場合、そのユーザーのアクセスが削除されます。

AWS Glue Data Catalog ロールをマッピングする (データソースの作成後にきめ細かなアクセスコントロールが有効になっている場合)

データソースの作成後にきめ細かなアクセスコントロールを有効にしている場合は、直接クエリを実行するために、管理者以外のユーザーを AWS Glue Data Catalog アクセス権を持つ IAM ロールにマッピングする必要があります。IAM ロールにマッピングできるバックエンド glue_access ロールを手動で作成するには、次のステップを実行します。

> **Note**
>
> インデックスは、データソースに対するクエリに使用されます。特定のデータソースのリクエストインデックスに対する読み取りアクセスを持つユーザーは、そのデータソースに対するすべてのクエリを読み取ることができます。結果インデックスに対する読み取りアクセスを持つユーザーは、そのデータソースに対するすべてのクエリの結果を読み取ることができます。

1. OpenSearch Dashboards のメインメニューから、セキュリティ 、ロール 、ロールの作成 を選択します。
2. ロールに glue_access という名前を付けます。
3. [クラスターの許可] で、indices:data/write/bulk*、indices:data/read/scroll、または indices:data/read/scroll/clear を選択します。
4. [インデックス] で、ロールアクセスをユーザーに付与する次のインデックスを入力します:
   - .query_execution_request_*<name of data source>*
   - query_execution_result_*<name of data source>*
   - flint_*
5. [インデックスの許可] で、indices_all を選択します。
6. [作成] を選択します。

7. [マッピングされたユーザー]、[マッピングの管理] を選択します。
8. [バックエンドロール] で、ドメインを呼び出す許可を必要とする AWS Glue ロールの ARN を追加します。

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

9. マップを選択し、ロールがマッピングされたユーザーに表示されていることを確認します。

ロールのマッピングの詳細については、「the section called “ユーザーへのロールのマッピング”」を参照してください。

次のステップ

データソースを作成すると、 OpenSearch サービスによって Dashboards URL OpenSearch が提供されます。これを使用して、アクセスコントロールの設定、テーブルの定義、一般的なログタイプ用のログタイプベースのダッシュボードの設定、およびデータのクエリを実行できます。

OpenSearch Dashboards でのデータソースの設定

データソースを作成したので、セキュリティ設定の構成、Amazon S3 テーブルの定義、高速データインデックスの設定を行うことができます。このセクションでは、データをクエリする前に、OpenSearch Dashboards のデータソースに関するさまざまなユースケースについて説明します。

以下のセクションを設定するには、まず OpenSearch Dashboards でデータソースに移動する必要があります。左側のナビゲーションの [管理] で、[データソース] を選択します。[データソースを管理] で、コンソールで作成したデータソースの名前を選択します。

アクセス制御を設定する

データソースの詳細ページで、「アクセスコントロール」セクションを見つけ、「編集」を選択します。セキュリティプラグインがインストールされている場合は、[制限対象] を選択し、新しいデータソースに対するアクセスを提供するロールベースのグループを選択します。管理者のみがデータソースにアクセスできるようにする場合は、[管理者のみ] を選択することもできます。

> **Important**
>
> インデックスは、データソースに対するクエリに使用されます。特定のデータソースのリクエストインデックスに対する読み取りアクセスを持つユーザーは、そのデータソースに対す

るすべてのクエリを読み取ることができます。結果インデックスに対する読み取りアクセスを持つユーザーは、そのデータソースに対するすべてのクエリの結果を読み取ることができます。

一般的な AWS ログタイプの統合を設定する

OpenSearch Dashboards を使用すると、Parquet 形式でサポートされている Amazon VPC フローログを除き、raw ログを使用して Amazon S3 に保存されている一般的なログタイプを簡単に使用開始できます。 OpenSearch Dashboards は、 AWS Glue Data Catalog テーブル、保存されたクエリ、ダッシュボードなどのアセットへのアクセスをインストールする統合を提供します。これらのアセットはアクセラ OpenSearch レーションを搭載しており、インストール後に自動的に更新されます。データソースの詳細ページまたは左側のナビゲーションから統合を設定できます。これを実行するには:

1. インストールするログタイプを選択します。インストールするログタイプに Amazon S3 タグがあることを確認します。
2. まだ選択されていない場合は、接続タイプを Amazon S3 接続として選択します。
3. 統合をインストールするデータソース名、データの Amazon S3 の場所、アクセラレーションインデックスステータスを維持するために使用するチェックポイント、ユースケースに基づいて目的のアセットを選択します。

> **Note**
>
> IAM ロールを作成するときに、チェックポイントの場所に対する書き込みアクション許可を持つチェックポイントの Amazon S3 リソースを指定しました。チェックポイントの場所の書き込みアクセス権を持つ Amazon S3 バケットの場所を参照する必要があります。そうしないと、統合がインストールするアクセラレーションは失敗します。

> **Note**
>
> Amazon VPC フローログ統合では、 OpenSearch Dashboards を使用してパッチをインストールする必要があります。インストールしたダッシュボードへの入力には数分かかる場合があります。

Amazon S3 にデータをエクスポートするためのリファレンスガイド

以下のリファレンスガイドを使用して、Amazon S3 にデータをエクスポートできます。

ソース:

- Apache アクセス
- CloudFront
- CloudTrail

- Elastic Load Balancing
- Amazon S3
- AWS WAF
- Amazon VPC フロー
- NGINX

Query Workbench を使用して Spark テーブルを作成する

OpenSearch サービスから Amazon S3 への直接クエリは、 内の Spark テーブルを使用します AWS Glue Data Catalog。 OpenSearch Dashboards を離れることなく、Query Workbench 内からテーブルを作成できます。

データソース内の既存のデータベースとテーブルを管理する場合、または直接クエリを使用する新しいテーブルを作成するには、左側のナビゲーションから Query Workbench を選択し、データソースドロップダウンから Amazon S3 データソースを選択します。

S3 に保存されている VPC フローログのテーブルを Parquet 形式で設定するには、次のクエリを実行します。

```
CREATE TABLE
datasourcename.gluedatabasename.vpclogstable (version INT, account_id STRING,
 interface_id STRING,
srcaddr STRING, dstaddr STRING, srcport INT, dstport INT, protocol INT, packets
 BIGINT,
bytes BIGINT, start BIGINT, end BIGINT, action STRING, log_status STRING,
`aws-account-id` STRING, `aws-service` STRING, `aws-region` STRING, year STRING,
month STRING, day STRING, hour STRING)
```

```
USING parquet PARTITIONED BY (aws-account-id, aws-service, aws-region, year, month,
day, hour)

LOCATION "s3://accountnum-vpcflow/AWSLogs"
```

テーブルを作成した後、次のクエリを実行して、テーブルがダイレクトクエリと互換性があることを確認します:

```
MSCK REPAIR TABLE  datasourcename.databasename.vpclogstable
```

高速クエリ

データソースの詳細ページで、[パフォーマンスを高速化] オプションを選択します。Amazon S3 でデータを迅速に操作できるように、データのインデックスを OpenSearch Service に作成するように設定できる 3 種類のアクセラレーションがあります。インデックスのスキップ、マテリアライズドビュー、インデックスのカバーです。

スキップインデックス

スキップインデックスを使用すると、Amazon S3 に保存されているデータのメタデータのみにインデックスを付けることができます。スキップインデックスを使用してテーブルをクエリすると、クエリプランナーはすべてのパーティションとファイルをスキャンするのではなく、インデックスを参照してクエリを書き換え、効率的にデータを見つけます。これにより、スキップインデックスを使用すると、保存されたデータの特定の場所を迅速に絞り込むことができます。

データソースの詳細ページから、高速化するデータベースとテーブルを選択して開始できる Accelerate Performance を選択します。または、スキップインデックスを自動生成することもできます。高速化するフィールドを手動で追加する場合は、フィールドの追加ボタンを選択します。フィールドを追加するときに、追加するスキップインデックスのタイプが尋ねられます。次のいずれかを選択する必要があります。

- パーティション: データパーティションの詳細を使用してデータを検索します (年、月、日、時間などのパーティションベースの列に最適です)
- MinMax: インデックス付き列の下限と上限を使用してデータを検索します (数値列に最適)
- ValueSet: 一意の値セットを使用してデータを検索します (低中程度のカーディナリティで完全一致が必要な列に最適です)。
- BloomFilter: ブルームフィルターを使用してデータを検索します (カーディナリティが高く、完全一致を必要としない列に最適です)

Query Workbench を使用して、テーブルにスキップインデックスを手動で作成することもできます。データソースドロップダウンから S3 データソースを選択し、次のクエリを追加するだけです。

```
CREATE SKIPPING INDEX
ON datasourcename.gluedatabasename.vpclogstable(
 `srcaddr` BLOOM_FILTER,
 `dstaddr` BLOOM_FILTER,
 `day` PARTITION,
 `account_id`BLOOM_FILTER
 ) WITH (
index_settings = '{"number_of_shards":5,"number_of_replicas":1}',
auto_refresh = true,
checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint'
)
```

マテリアライズドビュー

マテリアライズドビューを使用すると、集計などの複雑なクエリを使用して、Dashboard のビジュアライゼーションを強化できます。マテリアライズドビューは、クエリに応じて少量のデータを OpenSearch サービスストレージに取り込みます。 OpenSearch 次に、サービスは、視覚化に使用できる取り込まれたデータからインデックスを作成します。マテリアライズドビューインデックスはthe section called “インデックスステート管理”、他の OpenSearch インデックスと同様に、 で管理できます。

ターゲットインデックスを指定するため、インデックスに名前を付け、データの入力と処理の遅延を定義するウォーターマーク遅延を追加するように求められます。

次のクエリを使用して、 で作成した VPC フローログテーブルの新しいマテリアライズドビューを作成しますthe section called “Query Workbench を使用して Spark テーブルを作成する”。

```
CREATE MATERIALIZED VIEW {table_name}__week_live_mview AS
  SELECT
    cloud.account_uid AS `aws.vpc.cloud_account_uid`,
    cloud.region AS `aws.vpc.cloud_region`,
    cloud.zone AS `aws.vpc.cloud_zone`,
    cloud.provider AS `aws.vpc.cloud_provider`,

    CAST(IFNULL(src_endpoint.port, 0) AS LONG) AS `aws.vpc.srcport`,
    CAST(IFNULL(src_endpoint.svc_name, 'Unknown') AS STRING)  AS `aws.vpc.pkt-src-aws-
service`,
    CAST(IFNULL(src_endpoint.ip, '0.0.0.0') AS STRING)  AS `aws.vpc.srcaddr`,
```

```
    CAST(IFNULL(src_endpoint.interface_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
interface_uid`,
    CAST(IFNULL(src_endpoint.vpc_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-vpc_uid`,
    CAST(IFNULL(src_endpoint.instance_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
instance_uid`,
    CAST(IFNULL(src_endpoint.subnet_uid, 'Unknown') AS STRING)  AS `aws.vpc.src-
subnet_uid`,

    CAST(IFNULL(dst_endpoint.port, 0) AS LONG) AS `aws.vpc.dstport`,
    CAST(IFNULL(dst_endpoint.svc_name, 'Unknown') AS STRING) AS `aws.vpc.pkt-dst-aws-
service`,
    CAST(IFNULL(dst_endpoint.ip, '0.0.0.0') AS STRING)  AS `aws.vpc.dstaddr`,
    CAST(IFNULL(dst_endpoint.interface_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
interface_uid`,
    CAST(IFNULL(dst_endpoint.vpc_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-vpc_uid`,
    CAST(IFNULL(dst_endpoint.instance_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
instance_uid`,
    CAST(IFNULL(dst_endpoint.subnet_uid, 'Unknown') AS STRING)  AS `aws.vpc.dst-
subnet_uid`,
    CASE
      WHEN regexp(dst_endpoint.ip, '(10\\..*)|(192\\.168\\..*)|(172\\.1[6-9]\\..*)|
(172\\.2[0-9]\\..*)|(172\\.3[0-1]\\.*)')
        THEN 'ingress'
      ELSE 'egress'
      END AS `aws.vpc.flow-direction`,

    CAST(IFNULL(connection_info['protocol_num'], 0) AS INT) AS
 `aws.vpc.connection.protocol_num`,
    CAST(IFNULL(connection_info['tcp_flags'], '0') AS STRING)  AS
 `aws.vpc.connection.tcp_flags`,
    CAST(IFNULL(connection_info['protocol_ver'], '0') AS STRING)  AS
 `aws.vpc.connection.protocol_ver`,
    CAST(IFNULL(connection_info['boundary'], 'Unknown') AS STRING)  AS
 `aws.vpc.connection.boundary`,
    CAST(IFNULL(connection_info['direction'], 'Unknown') AS STRING)  AS
 `aws.vpc.connection.direction`,

    CAST(IFNULL(traffic.packets, 0) AS LONG) AS `aws.vpc.packets`,
    CAST(IFNULL(traffic.bytes, 0) AS LONG) AS `aws.vpc.bytes`,

    CAST(FROM_UNIXTIME(time / 1000) AS TIMESTAMP) AS `@timestamp`,
    CAST(FROM_UNIXTIME(start_time / 1000) AS TIMESTAMP) AS `start_time`,
    CAST(FROM_UNIXTIME(start_time / 1000) AS TIMESTAMP) AS `interval_start_time`,
    CAST(FROM_UNIXTIME(end_time / 1000) AS TIMESTAMP) AS `end_time`,
```

```
    status_code AS `aws.vpc.status_code`,

    severity AS `aws.vpc.severity`,
    class_name AS `aws.vpc.class_name`,
    category_name AS `aws.vpc.category_name`,
    activity_name AS `aws.vpc.activity_name`,
    disposition AS `aws.vpc.disposition`,
    type_name AS `aws.vpc.type_name`,

    region AS `aws.vpc.region`,
    accountid AS `aws.vpc.account-id`
  FROM
  datasourcename.gluedatabasename.vpclogstable
WITH (
  auto_refresh = true,
  refresh_interval = '15 Minute',
  checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint',
  watermark_delay = '1 Minute',
)
```

カバーインデックス

カバーインデックスを使用すると、テーブル内の指定した列からデータを取り込むことができます。これは、3 つのインデックス作成タイプの中で最もパフォーマンスが優れています。 OpenSearch サービスは目的の列からすべてのデータを取り込むため、パフォーマンスが向上し、高度な分析を実行できます。

マテリアライズドビューと同様に、 OpenSearch Service はカバーインデックスデータから新しいインデックスを作成します。この新しいインデックスは、ダッシュボードの視覚化や、異常検出や地理空間機能などのその他の OpenSearch サービス機能に使用できます。他のインデックスと同様にthe section called “インデックスステート管理”、 でカバービュー OpenSearch インデックスを管理できます。

次のクエリを使用して、 で作成した VPC フローログテーブルの新しいカバーインデックスを作成しますthe section called “Query Workbench を使用して Spark テーブルを作成する”。

```
CREATE INDEX vpc_covering_index
ON datasourcename.gluedatabasename.vpclogstable (version, account_id, interface_id,
srcaddr, dstaddr, srcport, dstport, protocol, packets,
bytes, start, action, log_status STRING,
`aws-account-id`, `aws-service`, `aws-region`, year,
```

```
month, day, hour )
WITH (
  auto_refresh = true,
  refresh_interval = '15 minute',
  checkpoint_location = 's3://accountnum-vpcflow/AWSLogs/checkpoint'
)
```

OpenSearch Dashboards でのデータのクエリ

テーブルを設定し、必要なオプションのクエリアクセラレーションを設定したら、データの分析の実行を開始できます。データにクエリを実行するには、 OpenSearch Dashboards の Discover ページまたは Observability ページのドロップダウンメニューからデータソースを選択します。

スキップインデックスを使用している場合、またはインデックスを作成していない場合は、SQL または Piped Processing Language (PPL) を使用してデータをクエリできます。マテリアライズドビューまたはカバーインデックスを設定している場合は、インデックスが既に存在するため、Dashboard 全体で Dashboards Query Language (DQL) を使用できます。オブザーバビリティプラグインで PPL を使用したり、Query Workbench プラグインで SQL を使用したりすることもできます。現在、PPL と SQL をサポートしているのは、オブザーバビリティプラグインと Query Workbench プラグインのみです。 OpenSearch Service API を使用したデータのクエリについては、「非同期 API ドキュメント」を参照してください。

SQL

次のクエリを使用して、 で作成した VPC フローログテーブルのサンプル SQL クエリを実行しますthe section called “Query Workbench を使用して Spark テーブルを作成する”。

```
SELECT srcaddr, SUM (CAST(bytes AS LONG)) as total_bytes
 FROM datasourcename.gluedatabasename.vpclogstable GROUP BY srcaddrORDER BY total_bytes
 DESCLIMIT 10;
```

PPL

次のクエリを使用して、 で作成した VPC ログテーブルのサンプル PPL クエリを実行しますthe section called “Query Workbench を使用して Spark テーブルを作成する”。

```
source = datasourcename.gluedatabasename.vpclogstable | fields account_id, srcaddr,
 dstaddr, action | head 10
```

レコメンデーション

結果が期待どおりに返されない場合があります。問題が発生した場合は、次のアクションを実行することをお勧めします。

- SELECT * ステートメントは結果を返しません - テーブルをチェックして、ネストされたストラク列を分解する必要があるかどうかを確認します。
- 複数のテーブルを選択する場合は、 SQL UNIONステートメントを使用して複数のテーブルを参照します。
- アクセラレーションは、特定の数のワーカーを使用してクエリを実行するように設定されます。クエリの応答が遅い場合は、パフォーマンスを向上させるクエリを実行するワーカーを手動で割り当てることができます。
- スキップインデックスを構築する場合、高基数にブルームフィルターを使用し、広い範囲に最小/最大を使用してドメインのスペースを節約します。完全一致を実行する必要がある場合は、中程度のカーディナリティフィールドに値を設定することをお勧めします。
- 一般的な SQL クエリの詳細については、AWS 「サービスログ」を参照してください。

データソースの管理

データソースの管理は、ダイレクトクエリデータソースやその他の ソリューションの信頼性、可用性、パフォーマンスを維持する上で AWS 重要な部分です。 AWS には、モニタリング、問題発生時の報告、必要に応じて自動アクションを実行するための以下のツールが用意されています。

トピック

- CloudWatch メトリクスデータソースによるモニタリング
- データソースの有効化と無効化
- AWS Budget によるモニタリング
- Amazon S3 を使用した Amazon OpenSearch Service データソースの削除 Amazon S3

CloudWatch メトリクスデータソースによるモニタリング

を使用して直接クエリをモニタリングできます CloudWatch。 CloudWatch は raw データを収集し、読み取り可能なほぼリアルタイムのメトリクスに処理します。これらの統計は 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をより的確に把握できます。

また、特定のしきい値をモニタリングし、しきい値に達したときに通知を送信したりアクションを実行したりするようにアラームを設定することもできます。詳細については、「Amazon とは」を参照してください CloudWatch。

ダイレクトクエリは、次のメトリクスを報告します。

メトリクス	説明
AsyncQueryCreateAPI	非同期クエリを作成するために API に対して行われたリクエストの合計数。 関連する統計： 平均、最大、合計 ディメンション： `ClientId`、`DomainName` 頻度: 60 秒
AsyncQueryGetApiRequestCount	非同期クエリ結果を取得するために API に対して行われたリクエストの合計数。 関連する統計： 平均、最大、合計 ディメンション： `ClientId`、 `DomainName` 頻度: 60 秒
AsyncQueryCancelApiRequestCount	非同期クエリをキャンセルするために API に対して行われたリクエストの合計数。 関連する統計： 平均、最大、合計 ディメンション: `ClientId`、 `DomainName` 頻度: 60 秒

メトリクス	説明
AsyncQueryGetApiFailedRequestCusErrCount	顧客関連のエラー (無効なクエリ ID など) により非同期クエリ結果を取得するときに失敗したリクエストの数。 関連する統計： 平均、最大、合計 ディメンション: `ClientId`、 `DomainName` 頻度: 60 秒
AsyncQueryCancelApiFailedRequestCusErrCount	顧客関連のエラー (無効なクエリ ID など) により非同期クエリ結果を取得するときに失敗したリクエストの数。 関連する統計： Average、Maximum、Sum ディメンション: `ClientId`、`DomainName` 頻度: 60 秒
AsyncQueryCancelApiFailedRequestSysErrCount	顧客関連のエラーによる非同期クエリの作成時に失敗したリクエストの数。 関連する統計情報: Average、Maximum、Sum ディメンション: `ClientId`、 `DomainName` 頻度: 60 秒
AsyncQueryGetApiFailedRequestSysErrCount	システム関連のエラーが原因で非同期クエリ結果を取得したときに失敗したリクエストの数。 関連する統計情報: Average、Maximum、Sum ディメンション： `ClientId`、 `DomainName` 頻度: 60 秒

データソースの有効化と無効化

データソースの直接クエリの使用を停止する場合は、データソースを無効にすることができます。データソースを無効にすると、既存のクエリの実行が終了し、ユーザーによる新しいクエリの実行がすべて停止します。

データソースが無効になると、インデックスのスキップ、マテリアライズドビュー、インデックスのカバーなどのクエリパフォーマンスを向上させるためのアクセラレーション設定が手動に設定されます。無効化後にデータソースがアクティブに設定されると、ユーザークエリは想定どおりに実行されます。以前にセットアップされ、手動に設定されているアクセラレーションは、スケジュールに従って再度実行するように手動で設定する必要があります。

AWS Budget によるモニタリング

Amazon OpenSearch Service は、アカウントレベルで OCU 使用状況データを請求情報とコスト管理の Cost Explorer に入力しています。お客様は、アカウントレベルで OCU の使用状況を考慮し、しきい値を超えたときにしきい値とアラートを設定できます。

Cost Explorer でフィルタリングする使用タイプの形式は、 RegionCode-DirectQueryOCU (OCU 時間) のようになります。 DirectQueryOCU (OCU-Hours) の使用量がしきい値に達したときに通知を受け取りたいお客様は、 AWS Budgets アカウントを作成し、設定したしきい値に基づいてアラートを設定できます。オプションで、お客様は Amazon SNS トピックを設定することを選択できます。これにより、しきい値の基準を満たした場合にデータソースがオフになります。

> **Note**
>
> AWS Budgets の使用状況データはリアルタイムではなく、最大 8 時間遅れる可能性があります。

Amazon S3 を使用した Amazon OpenSearch Service データソースの削除 Amazon S3

データソースを削除すると、Amazon OpenSearch Service によってドメインから削除されます。また、Amazon OpenSearch Service はデータソースに関連付けられたインデックスも削除します。トランザクションデータは Amazon S3 から削除されませんが、Amazon S3 は新しいデータを OpenSearch サービスに送信しません。

データソース統合は、 AWS Management Console または OpenSearch Service API を使用して削除できます。

AWS Management Console

データソースを削除するには

1. で Amazon OpenSearch Service コンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインから [ドメイン] を選択します。
3. データソースを削除するドメインを選択します。選択すると、ドメインの詳細ページが開きます。全般情報の下にある [接続] タブを選択し、[ダイレクトクエリ] セクションを見つけます。
4. 削除するデータソースを選択し、[削除] を選択して、削除を確認します。

OpenSearch サービス API

DeleteDataSource API オペレーションを使用して、ドメイン内の既存のデータソースを削除します。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

Amazon OpenSearch Service ドメインのモニタリング

モニタリングは、Amazon OpenSearch Service およびその他の AWS ソリューションの信頼性、可用性、およびパフォーマンスを維持する上で重要な部分です。AWS には、OpenSearch Service リソースをモニタリングしたり、問題が発生したときに報告したり、必要に応じて自動アクションを実行したりするために以下のツールが用意されています。

Amazon CloudWatch

Amazon CloudWatch は OpenSearch Service リソースをリアルタイムでモニタリングします。メトリクスの収集と追跡、カスタマイズしたダッシュボードの作成、およびメトリクスが特定のしきい値に達したときに通知したりアクションを実行したりするアラームの設定を行うことができます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

Amazon CloudWatch Logs

Amazon CloudWatch Logs を使用すると、OpenSearch のログファイルをモニタリング、保存、およびアクセスすることができます。CloudWatch Logs は、ログファイル内の情報をモニタリングし、特定のしきい値が満たされたときに通知できます。詳細については、Amazon CloudWatch Logs ユーザーガイドを参照してください。

Amazon EventBridge

Amazon EventBridge は、OpenSearch Service ドメインの変更を記述したシステムイベントのストリームをほぼリアルタイムに配信します。特定のイベントを監視し、これらのイベントが発生したときに他の AWS サービスで自動アクションをトリガーするルールを作成できます。詳細については、「Amazon EventBridge ユーザーガイド」を参照してください。

AWS CloudTrail

AWS CloudTrail は、OpenSearch Service に対して行われた設定 API 呼び出しをイベントとしてキャプチャします。指定した Amazon S3 バケットにこれらのイベントが渡されます。この情報を使用して、リクエストを行ったユーザーとアカウント、リクエスト元のソース IP アドレス、およびリクエストの発生日時を特定できます。詳細については、AWS CloudTrail ユーザーガイドを参照してください。

トピック

- Amazon による OpenSearch クラスターメトリクスのモニタリング CloudWatch
- Amazon CloudWatch Logs による OpenSearch ログのモニタリング

- Amazon OpenSearch サービスの監査ログの監視
- Amazon OpenSearch によるサービスイベントのモニタリング EventBridge
- AWS CloudTrail での Amazon OpenSearch Service API 呼び出しのモニタリング

Amazon による OpenSearch クラスターメトリクスのモニタリング CloudWatch

Amazon OpenSearch Service は、ドメインから Amazon . CloudWatch lets にデータを発行します CloudWatch。ユーザーは、これらのデータポイントに関する統計を、メトリクス と呼ばれる時系列データの順序付けられたセットとして取得します。 OpenSearch サービスは、ほとんどのメトリクスを CloudWatch 60 秒間隔で に送信します。汎用または磁気 EBS ボリュームを使用する場合、EBS ボリュームのメトリクスは 5 分ごとのみに更新されます。すべての累積メトリクス (例: `ThreadpoolWriteRejeceted`、`ThreadpoolSearchRejected`) はメモリ内であり、状態が失われます。メトリクスは、ノードドロップ、ノードバウンス、ノード交換、ブルー/グリーンデプロイ中にリセットされます。Amazon の詳細については CloudWatch、「Amazon ユーザーガイド CloudWatch 」を参照してください。

OpenSearch サービスコンソールには、 からの raw データに基づいて一連のグラフが表示されます CloudWatch。必要に応じて、コンソールのグラフ CloudWatch の代わりに でクラスターデータを表示することもできます。サービスは、メトリクスを 2 週間アーカイブし、その後破棄します。メトリクスは追加料金なしで提供されますが、ダッシュボードとアラームの作成には引き続き CloudWatch 料金がかかります。詳細については、「Amazon CloudWatch の料金」を参照してください。

OpenSearch サービスは、次のメトリクスを に発行します CloudWatch。

- the section called “クラスターメトリクス”
- the section called “専用マスターノードメトリクス”
- the section called “EBS ボリュームメトリクス”
- the section called “インスタンスメトリクス”
- the section called “UltraWarm メトリクス”
- the section called “コールドストレージのメトリクス”
- the section called “アラートメトリクス”
- the section called “異常検出のメトリクス”

- the section called “非同期検索メトリクス”
- the section called “SQL メトリクス”
- the section called “k-NN メトリクス”
- the section called “クラスター間検索のメトリクス”
- the section called “クラスター間レプリケーションメトリクス”
- the section called “Learning to Rank のメトリクス”
- the section called “Piped Processing Language のメトリクス”

でのメトリクスの表示 CloudWatch

CloudWatch メトリクスは、最初にサービス名前空間によってグループ化され、次に各名前空間内のさまざまなディメンションの組み合わせによってグループ化されます。

CloudWatch コンソールを使用してメトリクスを表示するには

1. https://console.aws.amazon.com/cloudwatch/ で CloudWatch コンソールを開きます。
2. 左のナビゲーションペインで、[Metrics] (メトリクス) を見つけ、[All metrics] (すべてのメトリクス) を選択します。ES/OpenSearchService 名前空間を選択します。
3. ディメンションを選択して、対応するメトリクスを表示します。個別のノードのメトリクスは、`ClientId, DomainName, NodeId` ディメンションにあります。クラスターメトリクスは、`Per-Domain, Per-Client Metrics` ディメンションにあります。一部のノードメトリクスは、クラスターレベルで集計されるため、両方のディメンションに含まれます。シャードメトリクスは、`ClientId, DomainName, NodeId, ShardRole` ディメンションにあります。

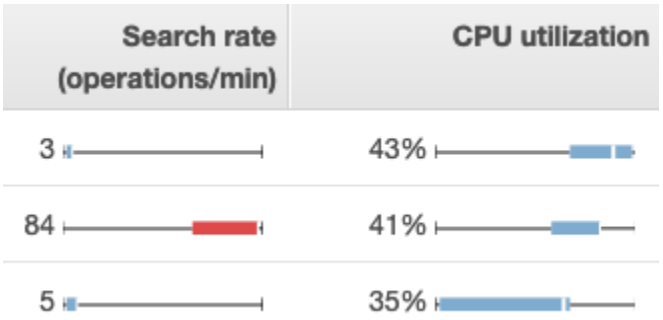
を使用してメトリクスのリストを表示するには AWS CLI

次のコマンドを実行します。

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

Service でのヘルスチャートの解釈 OpenSearch

OpenSearch サービスでメトリクスを表示するには、クラスターヘルスタブとインスタンスヘルスタブを使用します。インスタンスのヘルスタブでは、ボックスチャートを使用して各 OpenSearchノードのヘルスを at-a-glance 可視化します。



- それぞれの色付きの箱は、指定した期間におけるノードの値範囲を示しています。
- 青色の箱は他のノードと同じ値を示します。赤色の箱は異常値を示します。
- 各箱の白線は、ノードの現在値を示します。
- 各拍の両脇にある「ひげ」は、期間におけるすべてのノードの最大値と最小値を示します。

ドメインの設定を変更すると、[クラスターヘルス] タブと [インスタンスヘルス] タブの各インスタンスのリストが正しい数に戻る前に、しばらくの間 2 倍のサイズになる場合があります。この動作の説明については、「the section called “設定変更”」を参照してください。

クラスターメトリクス

Amazon OpenSearch Service は、クラスターに対して以下のメトリクスを提供します。

メトリクス	説明
`ClusterStatus.green`	値 1 は、すべてのインデックスシャードがクラスターのノードに割り当てられることを示します。 関連する統計: Maximum
`ClusterStatus.yellow`	値 1 は、すべてのインデックスのプライマリシャードがクラスターのノードに割り当てられていることを示しますが、1 つ以上のインデックスのレプリカシャードが割り当てられていません。詳細については、「the section called “黄色のクラスター状態”」を参照してください。 関連する統計: Maximum

メトリクス	説明
ClusterStatus.red	値 1 は、少なくとも 1 つのインデックスのプライマリとレプリカの両方のシャードが、クラスターのノードに割り当てられないことを示します。詳細については、「the section called “赤のクラスター状態”」を参照してください。 関連する統計: Maximum
Shards.active	アクティブなプライマリとレプリカの両方のシャードの合計数。 関連する統計: Maximum、Sum
Shards.unassigned	クラスターのノードに割り当てられていないシャードの数。 関連する統計: Maximum、Sum
Shards.delayedUnassigned	タイムアウト設定によってノード割り当てが遅れたシャードの数。 関連する統計: Maximum、Sum
Shards.activePrimary	アクティブなプライマリシャードの数。 関連する統計: Maximum、Sum
Shards.initializing	初期化中のシャードの数。 関連する統計情報: Sum
Shards.relocating	再配置中のシャードの数。 関連する統計情報: Sum
Nodes	専用マスターノードとノードを含む、 OpenSearch サービスクラスター内の UltraWarm ノードの数。詳細については、「the section called “設定変更”」を参照してください。 関連する統計: Maximum

メトリクス	説明
SearchableDocuments	クラスター内のすべてのデータノードで検索可能なドキュメントの合計数。 関連する統計情報: Minimum、Maximum、Average
DeletedDocuments	クラスター内のすべてのデータノードで削除対象としてマークされたドキュメントの合計数。これらのドキュメントは検索結果に表示されなくなり、セグメントのマージ中に削除されたドキュメント OpenSearch のみをディスクから削除します。このメトリクスは、削除リクエスト後に増加し、セグメントマージ後に減少します。 関連する統計情報: Minimum、Maximum、Average
CPUUtilization	クラスター内のデータノードの CPU 使用率の割合。Maximum は、CPU 使用率が最も高いノードを示します。Average は、クラスター内のすべてのノードを表します。このメトリクスは、個別のノードでも利用できます。 関連する統計: Maximum、Average

メトリクス	説明
FreeStorageSpace	クラスター内のデータノードの空き領域。Sum はクラスターの合計空き容量を示しますが、正確な値を得るには期間を 1 分にする必要があります。Minimum と Maximum は、それぞれ空き領域が最も少ないノードと最も多いノードを示します。このメトリクスは個々のノードでも使用できます。このメトリクスClusterBlockException が に達すると、 OpenSearch サービスは をスローします0。復旧するには、インデックスを削除する、より大きなインスタンスを追加する、既存のインスタンスに EBS ベースのストレージを追加する、のいずれかを実行する必要があります。詳細については、「the section called “使用可能なストレージ領域の不足”」を参照してください。 OpenSearch サービスコンソールには、この値が GiB で表示されます。Amazon CloudWatch コンソールには、MiB で表示されます。 **Note** FreeStorageSpace は、 および _cat/allocation APIsが提供する値 OpenSearch _cluster/stats よりも常に低くなります。 OpenSearch サービスは、内部オペレーションのために各インスタンスのストレージ領域の割合を予約します。詳細については、「ストレージ要件の計算」を参照してください。 関連する統計: Minimum、Maximum、Average、Sum
ClusterUsedSpace	クラスターの合計使用領域。正確な値を取得するには、期間を 1 分のままにしておく必要があります。 OpenSearch サービスコンソールには、この値が GiB で表示されます。Amazon CloudWatch コンソールには、MiB で表示されます。 関連する統計: Minimum、Maximum

メトリクス	説明
ClusterIndexWritesBlocked	クラスターで、着信する書き込みリクエストを受け入れるか、ブロックするかを指定します。値 0 では、クラスターでリクエストを受け入れます。値 1 ではリクエストをブロックします。 代表的なものとしては、FreeStorageSpace が少なすぎる、JVMMemoryPressure が高すぎるなどがあります。この問題を軽減するには、ディスク容量の追加やクラスターのスケーリングを検討します。 関連する統計: Maximum
JVMMemoryPressure	クラスター内のすべてのデータノードに使用される Java ヒープの最大パーセンテージ。 OpenSearch サービスは、Java ヒープにインスタンスの RAM の半分を使用し、ヒープサイズは 32 GiB までです。インスタンスは最大 64 GiB の RAM まで垂直スケーリングでき、それ以上はインスタンスを追加することで水平方向にスケーリングできます。「the section called “推奨 CloudWatch アラーム”」を参照してください。 関連する統計: Maximum **Note** このメトリクスのロジックは、サービスソフトウェア R20220323 で変更されました。詳細については、「リリースノート」を参照してください。
OldGenJVMMemoryPressure	クラスター内のすべてのデータノードにおける「旧世代」で使用する Java ヒープの最大パーセンテージ。このメトリクスは、ノードレベルでも使用できます。 関連する統計: Maximum

メトリクス	説明
AutomatedSnapshotFailure	クラスターの失敗した自動スナップショットの数。1 の値は、自動スナップショットが過去 36 時間、ドメイン用に取られなかったことを示します。 関連する統計: Minimum、Maximum
CPUCreditBalance	クラスター内の、データノードに使用できる残りの CPU クレジット。CPU クレジットは、フル CPU パフォーマンスを 1 分間実現します。詳細については、Amazon EC2 デベロッパーガイドの「CPU クレジット」を参照してください。このメトリクスは、T2 のインスタンスタイプでのみ使用できます。 関連する統計: Minimum
OpenSearchDashboardsHealthyNodes	OpenSearch Dashboards のヘルスチェック。最小、最大、および平均がすべて 1 に等しい場合、Dashboards は正常に動作しています。最大が 1、最小が 0、平均が 0.7 の 10 個のノードがある場合、これは 7 個のノード (70%) が正常であり、3 個のノード (30%) が正常でないことを意味します。 関連する統計情報: Minimum、Maximum、Average
OpensearchDashboardsReportingFailedRequestSysErrCount	サーバーの問題または機能の制限により失敗した OpenSearch Dashboards レポートを生成するリクエストの数。 関連する統計情報: Sum
OpensearchDashboardsReportingFailedRequestUserErrCount	クライアントの問題により失敗した OpenSearch Dashboards レポートを生成するリクエストの数。 関連する統計情報: Sum
OpensearchDashboardsReportingRequestCount	OpenSearch Dashboards レポートを生成するリクエストの合計数。 関連する統計情報: Sum

メトリクス	説明
OpensearchDashboardsReportingSuccessCount	OpenSearch Dashboards レポートを生成するために成功したリクエストの数。 関連する統計情報: Sum
KMSKeyError	値 1 は、保管中のデータの暗号化に使用される AWS KMS キーが無効になっていることを示します。通常のオペレーションにドメインを復元するには、キーを再度有効にします。コンソールでは、保管時のデータを暗号化するドメインに対してのみこのメトリクスが表示されます。 関連する統計: Minimum、Maximum
KMSKeyInaccessible	値 1 は、保管中のデータの暗号化に使用される AWS KMS キーが削除されたか、 OpenSearch サービスへの許可が取り消されたことを示します。この状態にあるドメインを復元することはできません。ただし、手動のスナップショットがある場合は、それを使用してドメインのデータを新しいドメインに移行できます。コンソールでは、保管時のデータを暗号化するドメインに対してのみこのメトリクスが表示されます。 関連する統計: Minimum、Maximum
InvalidHostHeaderRequests	無効な (または欠落している) ホストヘッダーを含む OpenSearch クラスターに対して行われた HTTP リクエストの数。有効なリクエストには、ホストヘッダー値としてドメインホスト名が含まれます。 OpenSearch サービスでは、制限付きアクセスポリシーを持たないパブリックアクセスドメインに対する無効なリクエストが拒否されます。すべてのドメインに制限付きアクセスポリシーを適用することをお勧めします。 このメトリクスの値が大きい場合は、 OpenSearch クライアントがリクエストにドメインホスト名 (IP アドレスなどではない) を含めていることを確認します。 関連する統計情報: Sum

メトリクス	説明
OpenSearchRequests (previously ElasticsearchRequests)	OpenSearch クラスターに対して行われたリクエストの数。 関連する統計情報: Sum
2xx, 3xx, 4xx, 5xx	特定の HTTP レスポンスコード (2xx、3xx、4xx、5xx) の発生につながった、ドメインへのリクエストの数。 関連する統計情報: Sum
ThroughputThrottle	ディスクがスロットリングされたかどうかが示されます。スロットリングは、ReadThroughputMicroBursting と WriteThroughputMicroBursting の合計スループットが最大スループット MaxProvisionedThroughput よりも高い場合に発生します。MaxProvisionedThroughput は、インスタンススループットまたはプロビジョニングされたボリュームスループットのうち、低い方の値です。値 1 は、ディスクがスロットリングされていることを示しています。値 0 は正常な動作を示します。 インスタンスのスループットについては、「Amazon EBS 最適化インスタンスを使用する」を参照してください。ボリュームスループットの詳細については、「Amazon EBS ボリュームの種類」を参照してください。 関連する統計: Minimum、Maximum

メトリクス	説明
IopsThrottle	ドメインの 1 秒あたりの入出力オペレーション数 (IOPS) がスロットリングされているかどうかを示します。スロットリングは、データノードの IOPS が、データノードの EBS ボリュームまたは EC2 インスタンスの最大許容制限を超えた場合に発生します。 インスタンス IOPS の詳細については、「Amazon EBS 最適化インスタンス」を参照してください。ボリューム IOPS の詳細については、「Amazon EBS ボリュームタイプ」を参照してください。 関連する統計: Minimum、Maximum

専用マスターノードメトリクス

Amazon OpenSearch Service は、専用マスターノード に対して以下のメトリクスを提供します。

メトリクス	説明
MasterCPUUtilization	専用マスターノードが使用する CPU リソースの最大パーセンテージ。このメトリクスが 60 パーセントに達する場合、インスタンスタイプのサイズを増やすことをお勧めします。 関連する統計: Maximum
MasterFreeStorageSpace	このメトリクスは関係ないため無視できます。このサービスはデータノードとしてマスターノードを使用しません。
MasterJVMMemoryPressure	クラスター内のすべての専用マスターノードで使用する Java ヒープの最大パーセンテージ。このメトリクスが 85 パーセントに達する場合、より大規模なインスタンスタイプに移行することをお勧めします。 関連する統計: Maximum

メトリクス	説明
	Note このメトリクスのロジックは、サービスソフトウェア R20220323 で変更されました。詳細については、「リリースノート」を参照してください。
MasterOldGenJVMMemoryPressure	マスターノードごとの「旧世代」で使用される Java ヒープの最大パーセンテージ。 関連する統計: Maximum
MasterCPUCreditBalance	クラスター内の専用マスターノードで使用できる、残りの CPU クレジット。CPU クレジットは、フル CPU パフォーマンスを 1 分間実現します。詳細については、Amazon EC2 デベロッパーガイドの「CPU クレジット」を参照してください。このメトリクスは、T2 のインスタンスタイプでのみ使用できます。 関連する統計: Minimum
MasterReachableFromNode	MasterNotDiscovered 例外のヘルスチェック。値 1 は正常な動作を示します。値 0 は、/_cluster/health/ の動作が正常ではないことを示します。 障害が発生すると、ソースノードからマスターノードにアクセスすることができなくなります。通常、ネットワーク接続の問題または AWS 依存関係の問題が原因です。 関連する統計: Maximum
MasterSysMemoryUtilization	使用中のマスターノードのメモリの割合。 関連する統計: Maximum

EBS ボリュームメトリクス

Amazon OpenSearch Service では、EBS ボリュームの以下のメトリクスが提供されます。

メトリクス	説明
ReadLatency	EBS ボリュームでの読み取り操作のレイテンシー (秒単位)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
WriteLatency	EBS ボリュームでの書き込み操作のレイテンシー (秒単位)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
ReadThroughput	EBS ボリュームでの読み取り操作のスループット (バイト/秒単位)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
ReadThroughputMicroBursting	マイクロバーストを考慮に入れたときの、EBS ボリュームにおける読み取りオペレーションのスループット (バイト/秒)。このメトリクスは、個別のノードでも利用できます。マイクロバーストは、EBS ボリュームがきわめて短い時間 (1 分未満) に高い IOPS またはスループットをバーストするときに発生します。 関連する統計情報: Minimum、Maximum、Average
WriteThroughput	EBS ボリュームでの書き込み操作のスループット (バイト/秒単位)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
WriteThroughputMicroBursting	マイクロバーストを考慮に入れたときの、EBS ボリュームにおける書き込みオペレーションのスループット (バイト/秒)。このメトリクスは、個別のノードでも利用できます。マイクロバーストは、EBS ボリュームがきわめて短い時間 (1 分未満) に高い IOPS またはスループットをバーストするときに発生します。 関連する統計情報: Minimum、Maximum、Average
DiskQueueDepth	EBS ボリュームに対する保留中の入出力 (I/O) リクエストの数。

メトリクス	説明
	関連する統計情報: Minimum、Maximum、Average
ReadIOPS	EBS ボリュームでの読み取り操作の入出力 (I/O) 操作数 (1 秒あたり)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
ReadIOPSMicroBursting	マイクロバーストを考慮に入れたときの、EBS ボリュームでの読み取り操作の入出力 (I/O) 操作数 (1 秒あたり)。このメトリクスは、個別のノードでも利用できます。マイクロバーストは、EBS ボリュームがきわめて短い時間 (1 分未満) に高い IOPS またはスループットをバーストするときに発生します。 関連する統計情報: Minimum、Maximum、Average
WriteIOPS	EBS ボリュームでの書き込み操作の入出力 (I/O) 操作数 (1 秒あたり)。このメトリクスは、個別のノードでも利用できます。 関連する統計情報: Minimum、Maximum、Average
WriteIOPS MicroBursting	マイクロバーストを考慮に入れたときの、EBS ボリュームでの書き込み操作の入出力 (I/O) 操作数 (1 秒あたり)。このメトリクスは、個別のノードでも利用できます。マイクロバーストは、EBS ボリュームがきわめて短い時間 (1 分未満) に高い IOPS またはスループットをバーストするときに発生します。 関連する統計情報: Minimum、Maximum、Average
BurstBalance	EBS ボリュームの、バーストバケットに残っている入出力 (I/O) クレジットの割合。値 100 は、ボリュームが最大クレジット数を累積したことを意味します。このパーセンテージが 70% を下回る場合は、「the section called “低 EBS バーストバランス”」を参照してください。gp3 ボリュームタイプを使用するドメインと、ボリュームサイズが 1,000 GiB を超える gp2 ボリュームを使用するドメインのバーストバランスは 0 のままになります。 関連する統計情報: Minimum、Maximum、Average

インスタンスメトリクス

Amazon OpenSearch Service は、ドメイン内のインスタンスごとに次のメトリクスを提供します。OpenSearch また、これらのインスタンスメトリクスを集約して、クラスター全体の状態に関するインサイトを提供します。この動作を確認するには、コンソールで [サンプル数] 統計を使用します。以下のテーブルの各メトリクスには、ノードとクラスターに関連する統計を含みます。

> **Important**
>
> Elasticsearch のバージョンが異なる場合、_index API への呼び出しの処理にも異なるスレッドプールが使用されます。Elasticsearch 1.5 および 2.3 は、インデックス作成スレッドプールを使用します。Elasticsearch 5.x 、6.0、および 6.2 はバルクスレッドプールを使用し、Elasticsearch OpenSearch 6.3 以降は書き込みスレッドプールを使用します。現在、OpenSearch サービスコンソールにはバルクスレッドプールのグラフは含まれていません。`GET _cluster/settings?include_defaults=true` を使用して、クラスターのスレッドプールとキューサイズを確認します。

メトリクス	説明
`ConcurrentSearchRate`	データノード上のすべてのシャードに対して 1 分あたりの同時セグメント検索を使用した検索リクエストの合計数。_search API への 1 回の呼び出しに対して、さまざまなシャードから結果が返される可能性があります。これらのシャードのうちの 5 つが 1 つのノードにある場合、クライアントが 1 つのリクエストしか行っていない場合でも、ノードはこのメトリクスについて 5 を報告します。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum、Sum
`ConcurrentSearchLatency`	ノードで N 分から 1 分 (N-1) までの同時セグメント検索を使用したすべての検索で取得された合計時間のミリ秒単位の差。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum

メトリクス	説明
IndexingLatency	ノード内のすべてのインデックス作成オペレーションにかかった合計時間 (ミリ秒) の、N 分と (N-1) 分の差。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum
IndexingRate	1 分あたりのインデックス作成オペレーションの数。2 つのドキュメントを追加し、2 つのカウントを 4 つのオペレーションとして更新する _bulk API への 1 回の呼び出し。これは 1 つ以上のノードに分散する可能性があります。そのインデックスに 1 つ以上のレプリカがあり、 OpenSearch最適化されたインスタンス がないドメインにある場合、クラスター内の他のノードも合計 4 つのインデックス作成オペレーションを記録します。最適化されたインスタンスを持つ OpenSearch ドメインの場合、レプリカを持つ他のノードはオペレーションを記録しません。ドキュメントの削除はこのメトリクスに対してカウントされません。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum、Sum
SearchLatency	ノード内のすべての検索にかかった合計時間 (ミリ秒) の、N 分と (N-1) 分の差。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum

メトリクス	説明
SearchRate	データノードのすべてのシャードに対する 1 分あたりの検索リクエストの総数。_search API への 1 回の呼び出しに対して、さまざまなシャードから結果が返される可能性があります。これらのシャードのうちの 5 つが 1 つのノードにある場合、クライアントが 1 つのリクエストしか行っていない場合でも、ノードはこのメトリクスについて 5 を報告します。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum、Sum
SegmentCount	データノードでのセグメントの数。セグメントが多いほど、各検索にかかる時間が長くなります。 OpenSearch 場合によっては、小さいセグメントを大きいセグメントにマージします。 関連するノードの統計: Maximum、Average 関連するクラスターの統計: Sum、Maximum、Average
SysMemoryUtilization	インスタンスが使用中のメモリの割合。このメトリクスの高い値は正常であり、通常はクラスターに問題はありません。潜在的なパフォーマンスおよび安定性の問題の指標については、「JVMMemoryPressure メトリクス」を参照してください。 関連するノードの統計: Minimum、Maximum、Average 関連するクラスターの統計: Minimum、Maximum、Average
JVMGCYoungCollectionCount	「新世代」ガベージコレクションが実行された回数。実行数が大量になり、かつ増え続けることは、通常のクラスター操作の一部です。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average

メトリクス	説明
JVMGCYoungCollectionTime	クラスターで「新世代」ガベージコレクションの実行にかかった時間 (ミリ秒)。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
JVMGCOldCollectionCount	「旧世代」ガベージコレクションが実行された回数。十分なリソースがあるクラスターでは、この回数は少ないままですが、まれに増加します。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
JVMGCOldCollectionTime	クラスターで「旧世代」ガベージコレクションの実行にかかった時間 (ミリ秒)。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
OpenSearchDashboardsConcurrentConnections	Dashboards への OpenSearchアクティブな同時接続の数。この数が一貫して増加する場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
OpenSearchDashboardsHealthyNode	個々の OpenSearch Dashboards ノードのヘルスチェック。値 1 は正常な動作を示します。値 0 は Dashboards がアクセス不可であることを示します。 関連するノードの統計: Minimum 関連するクラスターの統計: Minimum、Maximum、Average

メトリクス	説明
OpenSearchDashboardsHeapTotal	OpenSearch Dashboards に割り当てられたヒープメモリの容量を MiB 単位で表します。EC2 インスタンスタイプが異なると、正確なメモリ割り当てに影響する可能性があります。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
OpenSearchDashboardsHeapUsed	OpenSearch Dashboards が使用するヒープメモリの絶対量を MiB で表したものです。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
OpenSearchDashboardsHeapUtilization	Dashboards が OpenSearch使用する使用可能なヒープメモリの最大パーセンテージ。この値が 80% を超えて増加する場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Minimum、Maximum、Average
OpenSearchDashboardsOS1MinuteLoad	OpenSearch Dashboards の 1 分間の CPU 負荷平均。CPU ロードは、理想的には 1.00 未満にとどまるはずです。一時的なスパイクは問題ありませんが、このメトリクスが一貫して 1.00 を超える場合は、インスタンスタイプのサイズを増やすことをお勧めします。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum

メトリクス	説明
OpenSearchDashboardsRequestTotal	OpenSearch Dashboards に対して行われた HTTP リクエストの合計数。システムの速度が遅い、または Dashboards リクエストの数が多い場合は、インスタンスタイプのサイズを増やすことを検討してください。 関連するノードの統計: Sum 関連するクラスターの統計: Sum
OpenSearchDashboardsResponseTimesMaxInMillis	OpenSearch Dashboards がリクエストに応答するのにかかる最大時間をミリ秒単位で表します。リクエストで結果が返ってくるために一貫して時間がかかる場合は、インスタンスタイプのサイズを増やすことを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Maximum、Average
SearchTaskCancelled	コーディネーターノードのキャンセル数。 関連するノードの統計: Sum 関連するクラスターの統計: Sum
SearchShardTaskCancelled	データノードのキャンセル数。 関連するノードの統計: Sum 関連するクラスターの統計: Sum、
ThreadpoolForce_mergeQueue	強制マージスレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average

メトリクス	説明
ThreadpoolForce_mergeRejected	強制マージスレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum
ThreadpoolForce_mergeThreads	強制マージスレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolIndexQueue	インデックス作成スレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。インデックスキューの最大サイズは 200 です。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
ThreadpoolIndexRejected	インデックス作成スレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum
ThreadpoolIndexThreads	インデックス作成スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum

メトリクス	説明
ThreadpoolSearchQueue	検索スレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。検索キューの最大サイズは 1,000 です。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
ThreadpoolSearchRejected	検索スレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum
ThreadpoolSearchThreads	検索スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
Threadpoolsql-workerQueue	SQL 検索スレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
Threadpoolsql-workerRejected	SQL 検索スレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum

メトリクス	説明
Threadpoolsql-workerThreads	SQL 検索スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolBulkQueue	バルクスレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
ThreadpoolBulkRejected	バルクスレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum
ThreadpoolBulkThreads	バルクスレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolIndexSearcherQueue	インデックス検索スレッドプール内のキューに入れられたタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
ThreadpoolIndexSearcherRejected	インデックス検索スレッドプールで拒否されたタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum

メトリクス	説明
ThreadpoolIndexSearcherThreads	インデックス検索スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolWriteThreads	書き込みスレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolWriteQueue	書き込みスレッドプールでキューに入っているタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
ThreadpoolWriteRejected	書き込みスレッドプールで拒否されたタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum **Note** バージョン 7.1 では、デフォルトの書き込みキューサイズが 200 から 10000 に引き上げられたため、このメトリクスは OpenSearch サービスからの拒否を示す唯一の指標ではなくなりました。CoordinatingWriteRejected 、PrimaryWriteRejected 、および ReplicaWriteRejected メトリクスを使用して、バージョン 7.1 以降での拒否をモニタリングします。

メトリクス	説明
CoordinatingWriteRejected	前回の OpenSearch サービスプロセスの起動以降にインデックス作成の負荷により調整ノードで発生した拒否の合計数。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum このメトリクスは、バージョン 7.1 以降で使用できます。
PrimaryWriteRejected	前回の OpenSearch サービスプロセスの起動以降にインデックス作成のプレッシャーが原因でプライマリシャードで発生した拒否の合計数。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum このメトリクスは、バージョン 7.1 以降で使用できます。
ReplicaWriteRejected	前回の OpenSearch サービスプロセスの起動以降にインデックス作成のプレッシャーが原因でレプリカシャードで発生した拒否の合計数。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum このメトリクスは、バージョン 7.1 以降で使用できます。

UltraWarm メトリクス

Amazon OpenSearch Service では、UltraWarmノードについて以下のメトリクスを提供しています。

メトリクス	説明
WarmCPUUtilization	クラスター内の UltraWarm ノードの CPU 使用率。Maximum は、CPU 使用率が最も高いノードを示します。Average は、クラスター内の

メトリクス	説明
	すべての UltraWarm ノードを表します。このメトリクスは、個々の UltraWarm ノードでも使用できます。 関連する統計: Maximum、Average
WarmFreeStorageSpace	ウォームストレージの空き容量 (MiB)。はアタッチされたディスクではなく Amazon S3 を使用するため UltraWarm、関連する統計Sumは のみです。正確な値を取得するには、期間を 1 分のままにしておく必要があります。 関連する統計情報: Sum
WarmSearchableDocuments	クラスター内のすべてのウォームインデックスで検索可能なドキュメントの合計数。正確な値を取得するには、期間を 1 分のままにしておく必要があります。 関連する統計情報: Sum
WarmSearchLatency	N 分から 1 分 (N-1) UltraWarm の間の でのすべての検索で取得された合計時間のミリ秒単位の差。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum
WarmSearchRate	UltraWarm ノード上のすべてのシャードに対する 1 分あたりの検索リクエストの合計数。_search API への 1 回の呼び出しに対して、さまざまなシャードから結果が返される可能性があります。これらのシャードのうちの 5 つが 1 つのノードにある場合、クライアントが 1 つのリクエストしか行っていない場合でも、ノードはこのメトリクスについて 5 を報告します。 関連するノードの統計: Average 関連するクラスターの統計: Average、Maximum、Sum

メトリクス	説明
WarmStorageSpaceUtilization	クラスターで使用中のウォームストレージスペースの合計容量 (MiB)。 関連する統計: Maximum
HotStorageSpaceUtilization	クラスターで使用しているホットストレージの合計容量。 関連する統計: Maximum
WarmSysMemoryUtilization	使用中のウォームノードのメモリの割合。 関連する統計: Maximum
HotToWarmMigrationQueueSize	現在、ホットストレージからウォームトストレージへの移行を待機しているインデックスの数。 関連する統計: Maximum
WarmToHotMigrationQueueSize	現在、ウォームストレージからホットストレージへの移行を待機しているインデックスの数。 関連する統計: Maximum
HotToWarmMigrationFailureCount	失敗したホットからウォームへの移行の合計数。 関連する統計情報: Sum
HotToWarmMigrationForceMergeLatency	移行プロセスの強制マージステージの平均レイテンシー。この段階が一貫して時間がかかりすぎる場合は、index.ultrawarm.migration.force_merge.max_num_segments を増やすことを検討してください。 関連する統計: Average

メトリクス	説明
HotToWarmMigrationSnapshotLatency	移行プロセスのスナップショットステージの平均レイテンシー。この段階が一貫して時間がかかりすぎる場合は、シャードが適切にサイズ設定され、クラスター全体に分散されていることを確認します。 関連する統計: Average
HotToWarmMigrationProcessingLatency	ホットからウォームへの移行が成功した場合の平均レイテンシーで、キューに費やした時間を含まない。この値は、移行プロセスの強制マージ、スナップショット、およびシャード再配置ステージを完了するのにかかる時間の合計です。 関連する統計: Average
HotToWarmMigrationSuccessCount	ホットからウォームへの移行に成功した合計数。 関連する統計情報: Sum
HotToWarmMigrationSuccessLatency	ホットからウォームへの移行が成功した場合の平均レイテンシーで、キューに費やされた時間を含む。 関連する統計: Average
WarmThreadpoolSearchThreads	UltraWarm 検索スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスターの統計: Average、Sum
WarmThreadpoolSearchRejected	UltraWarm 検索スレッドプールで拒否されたタスクの数。この数が継続的に増加する場合は、ノード UltraWarmの追加を検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum

メトリクス	説明
WarmThreadpoolSearchQueue	UltraWarm 検索スレッドプール内のキューに入れられたタスクの数。キューサイズが一貫して大きい場合は、ノードの追加 UltraWarmを検討してください。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
WarmJVMMemoryPressure	ノードに使用される UltraWarm Java ヒープの最大パーセンテージ。 関連する統計: Maximum Note このメトリクスのロジックは、サービスソフトウェア R20220323 で変更されました。詳細については、「リリースノート」を参照してください。
WarmOldGenJVMMemoryPressure	UltraWarm ノードあたりの「旧世代」に使用される Java ヒープの最大パーセンテージ。 関連する統計: Maximum
WarmJVMGCYoungCollectionCount	「未世代」ガベージコレクションが UltraWarm ノードで実行された回数。実行数が大量になり、かつ増え続けることは、通常のクラスター操作の一部です。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
WarmJVMGCYoungCollectionTime	クラスターがノードで「小さな世代」ガベージコレクションの実行に UltraWarm費やしたミリ秒単位の時間。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average

メトリクス	説明
WarmJVMGCOldCollectionCount	「旧世代」ガベージコレクションが UltraWarm ノードで実行された回数。十分なリソースがあるクラスターでは、この回数は少ないままですが、まれに増加します。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
WarmConcurrentSearchRate	UltraWarm ノード上のすべてのシャードに対して 1 分あたりの同時セグメント検索を使用した検索リクエストの合計数。_search API への 1 回の呼び出しに対して、さまざまなシャードから結果が返される可能性があります。これらのシャードのうちの 5 つが 1 つのノードにある場合、クライアントが 1 つのリクエストしか行っていない場合でも、ノードはこのメトリクスについて 5 を報告します。 関連するノードの統計: Average 関連するクラスターの統計: Sum、Maximum、Average
WarmConcurrentSearchLatency	UltraWarm ノードで N 分から 1 分 (N-1) の間の同時セグメント検索を使用したすべての検索で取得された合計時間のミリ秒単位の差。 関連するノードの統計: Average 関連するクラスターの統計: Maximum、Average
WarmThreadpoolIndexSearcherQueue	UltraWarm インデックス検索スレッドプール内のキューに入れられたタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum、Maximum、Average
WarmThreadpoolIndexSearcherRejected	UltraWarm インデックス検索スレッドプールで拒否されたタスクの数。 関連するノードの統計: Maximum 関連するクラスターの統計: Sum

メトリクス	説明
WarmThreadpoolIndexSearcherThreads	UltraWarm インデックス検索スレッドプールのサイズ。 関連するノードの統計: Maximum 関連するクラスター統計: Sum、Average

コールドストレージのメトリクス

Amazon OpenSearch Service は、コールドストレージ について以下のメトリクスを提供します。

メトリクス	説明
ColdStorageSpaceUtilization	クラスターで使用しているコールドストレージの合計容量 (MiB)。 関連する統計情報: Max
ColdToWarmMigrationFailureCount	コールドからウォームへの移行に失敗した移行の合計数。 関連する統計情報: Sum
ColdToWarmMigrationLatency	コールドからウォームへの移行が正常に完了するまでの時間。 関連する統計: Average
ColdToWarmMigrationQueueSize	現在、コールドストレージからウォームストレージへの移行を待機しているインデックスの数。 関連する統計: Maximum
ColdToWarmMigrationSuccessCount	コールドからウォームへの移行に成功した合計数。 関連する統計情報: Sum
WarmToColdMigrationFailureCount	ウォームからコールドへの移行に失敗した合計数。 関連する統計情報: Sum

メトリクス	説明
WarmToColdMigrationLatency	ウォームからコールドへの移行が正常に完了するまでの時間。 関連する統計: Average
WarmToColdMigrationQueueSize	現在、ウォームストレージからコールドストレージへの移行を待機しているインデックスの数。 関連する統計: Maximum
WarmToColdMigrationSuccessCount	ウォームからコールドへの移行に成功した合計数。 関連する統計情報: Sum

OR1 メトリクス

Amazon OpenSearch Service は、OR1 インスタンス に対して次のメトリクスを提供します。

メトリクス	説明
RemoteStorageUsedSpace	クラスターで使用中の Amazon S3 スペースの合計容量 (MiB)。 関連する統計情報: Sum
RemoteStorageWriteRejected	リモートストレージとレプリケーションの負荷により、プライマリシャードで拒否されたリクエストの総数。これは、前回の OpenSearch サービスプロセスのスタートアップから計算されます。 関連する統計情報: Sum

アラートメトリクス

Amazon OpenSearch Service では、 にアラートを送信するために以下のメトリクスが用意されています。

メトリクス	説明
AlertingDegraded	値 1 は、アラートインデックスが赤であるか、1 つ以上のノードがスケジュールどおりでないことを意味します。値 0 は正常な動作を示します。 関連する統計: Maximum
AlertingIndexExists	値 1 は、.opensearch-alerting-config インデックスが存在することを意味します。値 0 は、そのインデックスが存在しないことを意味します。アラート機能を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum
AlertingIndexStatus.green	インデックスのヘルス。値 1 は、緑を意味します。値 0 は、インデックスが存在しないか、緑ではないことを意味します。 関連する統計: Maximum
AlertingIndexStatus.red	インデックスのヘルス。値 1 は、赤を意味します。値 0 は、インデックスが存在しないか、赤でないことを意味します。 関連する統計: Maximum
AlertingIndexStatus.yellow	インデックスのヘルス。値 1 は、黄色を意味します。値 0 は、インデックスが存在しないか、黄色でないことを意味します。 関連する統計: Maximum
AlertingNodesNotOnSchedule	値 1 は、一部のジョブがスケジュールどおりに実行されていないことを意味します。値 0 は、すべてのアラートジョブがスケジュールどおりに実行されていることを意味します (またはアラートジョブが存在しないことを意味します)。 OpenSearch サービスコンソールを確認するか、_nodes/stats リクエストを実行して、リソース使用率が高いノードがあるかどうかを確認します。 関連する統計: Maximum

メトリクス	説明
AlertingNodesOnSchedule	値 1 は、すべてのアラートジョブがスケジュールどおりに実行されていることを意味します (またはアラートジョブが存在しないことを意味します)。値 0 は、一部のジョブがスケジュールどおりに実行されていないことを意味します。 関連する統計: Maximum
AlertingScheduledJobEnabled	値 1 は、opensearch.scheduled_jobs.enabled クラスター設定が true であることを意味します。値 0 は、その設定が false であり、スケジュールされたジョブが無効であることを意味します。 関連する統計: Maximum

異常検出のメトリクス

Amazon OpenSearch Service は、異常検出 について以下のメトリクスを提供します。

メトリクス	説明
ADPluginUnhealthy	値 1 は、異常検出プラグインが正しく動作していないことを意味します。これは、障害の数が多いか、使用しているインデックスの 1 つが赤の状態であるためです。値 0 は、プラグインが想定どおりに動作していることを示します。 関連する統計: Maximum
ADExecuteRequestCount	異常検出のリクエストの数。 関連する統計情報: Sum
ADExecuteFailureCount	異常検出に失敗したリクエストの数。 関連する統計情報: Sum
ADHCExecuteFailureCount	高基数ディテクターの異常検出に失敗したリクエストの数。 関連する統計情報: Sum

メトリクス	説明
ADHCExecuteRequestCount	高基数ディテクターの異常検出のリクエストの数。 関連する統計情報: Sum
ADAnomalyResultsIndexStatusIndexExists	値 1 は、.opensearch-anomaly-results エイリアスが指すインデックスが存在することを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum
ADAnomalyResultsIndexStatus.red	値 1 は、 .opensearch-anomaly-results エイリアスが指すインデックスが赤の状態であることを意味します。値 0 は、そうでないことを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum
ADAnomalyDetectorsIndexStatusIndexExists	値 1 は、.opensearch-anomaly-detectors インデックスが存在することを意味します。値 0 は、そのインデックスが存在しないことを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum
ADAnomalyDetectorsIndexStatus.red	値 1 は、.opensearch-anomaly-detectors インデックスが赤の状態であることを意味します。値 0 は、そうでないことを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum
ADModelsCheckpointIndexStatusIndexExists	値 1 は、.opensearch-anomaly-checkpoints インデックスが存在することを意味します。値 0 は、そのインデックスが存在しないことを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum

メトリクス	説明
ADModelsCheckpointIndexStatus.red	値 1 は、.opensearch-anomaly-checkpoints インデックスが赤の状態であることを意味します。値 0 は、そうでないことを意味します。異常検出を初めて使用するまで、この値は 0 のままです。 関連する統計: Maximum

非同期検索メトリクス

Amazon OpenSearch Service は、非同期検索 に対して以下のメトリクスを提供します。

非同期検索コーディネーターノードの統計 (コーディネーターノードあたり)

メトリクス	説明
AsynchronousSearchSubmissionRate	過去 1 分間に送信された非同期検索の数。
AsynchronousSearchInitializedRate	過去 1 分間に初期化された非同期検索の数。
AsynchronousSearchRunningCurrent	現在実行中の非同期検索の数。
AsynchronousSearchCompletionRate	過去 1 分間に正常に完了した非同期検索の数。
AsynchronousSearchFailureRate	過去 1 分間に完了し、失敗した非同期検索の数。

メトリクス	説明
AsynchronousSearchPersistRate	過去 1 分間に持続した非同期検索の数。
AsynchronousSearchPersistFailedRate	過去 1 分間に持続できなかった非同期検索の数。
AsynchronousSearchRejected	ノードの稼働時間以降に拒否された非同期検索の合計数。
AsynchronousSearchCancelled	ノードの稼働時間以降にキャンセルされた非同期検索の合計数。
AsynchronousSearchMaxRunningTime	過去 1 分間にノードで実行されている非同期検索の最長時間。

非同期検索クラスター統計

メトリクス	説明
AsynchronousSearchStoreHealth	過去 1 分間に持続したインデックス (赤/赤以外) 内のストアのヘルス。
AsynchronousSearchStoreSize	過去 1 分間のすべてのシャードのシステムインデックスのサイズ。
AsynchronousSearch	過去 1 分間にシステムインデックスに保存されたレスポンスの数。

メトリクス	説明
StoredResponseCount	

Auto-Tune メトリクス

Amazon OpenSearch Service は、Auto-Tune の以下のメトリクスを提供します。

メトリクス	説明
AutoTuneChangesHistoryHeapSize	ヒープサイズチューニング値の MiB 単位の変更履歴。
AutoTuneChangesHistoryJVMYoungGenArgs	JVM YongGen 引数の変更履歴。
AutoTuneFailed	Auto-Tune の変更が失敗したかどうかを示すブール値。
AutoTuneSucceeded	Auto-Tune の変更が成功したかどうかを示すブール値。
AutoTuneValue	中断を伴わない変更のキュー変更履歴 (カウント) とキャッシュチューニング変更履歴 (MiB 単位)。

Multi-AZ with Standby メトリクス

Amazon OpenSearch Service は、スタンバイ のマルチ AZ に対して次のメトリクスを提供します。

アクティブなアベイラビリティーゾーンにおけるデータノードのノードレベルメトリクス

メトリクス	説明
CPUUtilization	クラスター内のデータノードの CPU 使用率の割合。Maximum は、CPU 使用率が最も高いノードを示します。Average は、クラスター内のすべ

メトリクス	説明
	てのノードを表します。このメトリクスは、個別のノードでも利用できます。
FreeStorageSpace	クラスター内のデータノードの空き領域。Sum はクラスターの合計空き容量を示しますが、正確な値を得るには期間を 1 分にする必要があります。Minimum と Maximum は、それぞれ空き領域が最も少ないノードと最も多いノードを示します。このメトリクスは個々のノードでも使用できます。このメトリクスClusterBlockException が に達すると、OpenSearch サービスは をスローします0。復旧するには、インデックスを削除する、より大きなインスタンスを追加する、既存のインスタンスに EBS ベースのストレージを追加する、のいずれかを実行する必要があります。詳細については、「the section called “使用可能なストレージ領域の不足”」を参照してください。 OpenSearch サービスコンソールには、この値が GiB で表示されます。Amazon CloudWatch コンソールには、MiB で表示されます。
JVMMemoryPressure	クラスター内のすべてのデータノードに使用される Java ヒープの最大パーセンテージ。 OpenSearch サービスは、Java ヒープにインスタンスの RAM の半分を使用し、ヒープサイズは 32 GiB までです。インスタンスは最大 64 GiB の RAM まで垂直スケーリングでき、それ以上はインスタンスを追加することで水平方向にスケーリングできます。「the section called “推奨 CloudWatch アラーム”」を参照してください。
SysMemoryUtilization	インスタンスが使用中のメモリの割合。このメトリクスの高い値は正常であり、通常はクラスターに問題はありません。潜在的なパフォーマンスおよび安定性の問題の指標については、「JVMMemoryPressure メトリクス」を参照してください。
IndexingLatency	ノード内のすべてのインデックス作成オペレーションにかかった合計時間 (ミリ秒) の、N 分と (N-1) 分の差。
IndexingRate	1 分あたりのインデックス作成オペレーションの数。
SearchLatency	ノード内のすべての検索にかかった合計時間 (ミリ秒) の、N 分と (N-1) 分の差。

メトリクス	説明
SearchRate	データノードのすべてのシャードに対する 1 分あたりの検索リクエストの総数。
ThreadpoolSearchQueue	検索スレッドプールでキューに入っているタスクの数。キューのサイズが一貫して大きい場合は、クラスターのスケーリングを検討してください。検索キューの最大サイズは 1,000 です。
ThreadpoolWriteQueue	書き込みスレッドプールでキューに入っているタスクの数。
ThreadpoolSearchRejected	検索スレッドプールで拒否されたタスクの数。この数が増え続ける場合は、クラスターのスケーリングを検討してください。
ThreadpoolWriteRejected	書き込みスレッドプールで拒否されたタスクの数。

アクティブなアベイラビリティーゾーンにおけるクラスターのクラスターレベルメトリクス

メトリクス	説明
DataNodes	アクティブシャードとスタンバイシャードの合計数。
DataNodes Shards.active	アクティブなプライマリとレプリカの両方のシャードの合計数。
DataNodes Shards.unassigned	クラスターのノードに割り当てられていないシャードの数。
DataNodes Shards.initializing	初期化中のシャードの数。

メトリクス	説明
DataNodes Shards.relocating	再配置中のシャードの数。

アベイラビリティーゾーンのローテーションメトリクス

ActiveReads.*Availability-Zone* = 1 の場合、ゾーンはアクティブです。ActiveReads.*Availability-Zone* = 0 の場合、ゾーンはスタンバイ状態です。

ポイントインタイムメトリクス

Amazon OpenSearch Service は、ポイントインタイム (PIT) 検索について以下のメトリクスを提供します。

PIT コーディネーターノードの統計 (コーディネーターノードあたり)

メトリクス	説明
CurrentPointInTime	ノード内のアクティブな PIT 検索コンテキストの数。
TotalPointInTime	ノードのアップタイム以降の、期限切れの PIT 検索コンテキストの数。
AvgPointInTimeAliveTime	ノードのアップタイム以降の、PIT 検索コンテキストの平均キープアライブ数。
HasActivePointInTime	値が 1 の場合、アクティブな PIT コンテキストがノードのアップタイム以降に存在していることを示します。値が 0 の場合は存在していません。
HasUsedPointInTime	値が 1 の場合、期限切れの PIT コンテキストがノードのアップタイム以降に存在していることを示します。値が 0 の場合は存在していません。

SQL メトリクス

Amazon OpenSearch Service では、SQL サポート に関する以下のメトリクスを提供しています。

メトリクス	説明
SQLFailedRequestCountByCusErr	クライアントの問題により失敗した _sql API へのリクエストの数。例えば、IndexNotFoundException により、リクエストが HTTP ステータスコード 400 を返す場合があります。 関連する統計情報: Sum
SQLFailedRequestCountBySysErr	サーバーの問題または機能の制限により失敗した _sql API へのリクエストの数。例えば、VerificationException により、リクエストが HTTP ステータスコード 503 を返す場合があります。 関連する統計情報: Sum
SQLRequestCount	_sql API へのリクエストの数。 関連する統計情報: Sum
SQLDefaultCursorRequestCount	SQLRequestCount に似ていますが、測定対象はページネーションのリクエストのみです。 関連する統計情報: Sum
SQLUnhealthy	値 1 は、特定のリクエストに応答して、SQL プラグインが 5xx レスポンスコードを返すか、無効なクエリ DSL を に渡すことを示します OpenSearch。他のリクエストは引き続き成功します。値 0 は、最近の障害がないことを示します。値 1 が持続して表示される場合、クライアントがプラグインに対して行っているリクエストのトラブルシューティングを行います。 関連する統計: Maximum

k-NN メトリクス

Amazon OpenSearch Service には、k 最近傍 (k-NN) プラグインの以下のメトリクスが含まれています。

メトリクス	説明
KNNCacheCapacityReached	キャッシュ容量に達したかどうかのノード単位のメトリクス。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計: Maximum
KNNCircuitBreakerTriggered	サーキットブレーカーがトリガーされるかどうかのクラスタ単位のメトリクス。KNNCacheCapacityReached についていずれかのノードが 1 の値を返す場合、この値も 1 を返します。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計: Maximum
KNNEvictionCount	メモリ制約またはアイドル時間のためにキャッシュから削除されたグラフ数のノード単位のメトリクス。インデックスの削除のために発生した明示的な削除はカウントされません。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum
KNNGraphIndexErrors	ドキュメントの knn_vector フィールドを、エラーを生成したグラフに追加するリクエストの数のノードごとのメトリクス。 関連する統計情報: Sum
KNNGraphIndexRequests	ドキュメントの knn_vector フィールドを、グラフに追加するリクエストの数のノードごとのメトリクス。 関連する統計情報: Sum
KNNGraphMemoryUsage	現在のキャッシュサイズ (メモリー内のすべてのグラフの合計サイズ) のノードごとのメトリクス (KB)。このメトリクスは、おおよその k-NN 検索にのみ関係します。

メトリクス	説明
	関連する統計: Average
KNNGraphQueryErrors	エラーを生成したグラフクエリの数のノード単位のメトリクス。 関連する統計情報: Sum
KNNGraphQueryRequests	グラフクエリの数のノード単位のメトリクス。 関連する統計情報: Sum
KNNHitCount	キャッシュヒットの数のノード単位のメトリクス。ユーザーがすでにメモリにロードされているグラフのクエリを行ったときに、キャッシュヒットが発生します。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum
KNNLoadExceptionCount	グラフをキャッシュにロードしようとしたときに例外が発生した回数のノード単位のメトリクス。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum
KNNLoadSuccessCount	プラグインがグラフをキャッシュに正常にロードした回数のノード単位のメトリクス。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum
KNNMissCount	キャッシュミスの数のノード単位のメトリクス。ユーザーがまだメモリにロードされていないグラフのクエリを行ったときに、キャッシュミスが発生します。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum

メトリクス	説明
KNNQueryRequests	k-NN プラグインが受信したクエリリクエストの数のノード単位のメトリクス。 関連する統計情報: Sum
KNNScriptCompilationErrors	スクリプトのコンパイル中のエラーの数のノード単位のメトリクス。この統計は、k-NN スコアスクリプト検索にのみ関係します。 関連する統計情報: Sum
KNNScriptCompilations	k-NN スクリプトがコンパイルされた回数のノードごとのメトリクス。通常、この値は 1 または 0 であるはずですが、コンパイルされたスクリプトを含むキャッシュがいっぱいになると、k-NN スクリプトが再コンパイルされる可能性があります。この統計は、k-NN スコアスクリプト検索にのみ関係します。 関連する統計情報: Sum
KNNScriptQueryErrors	スクリプトクエリ中のエラーの数のノード単位のメトリクス。この統計は、k-NN スコアスクリプト検索にのみ関係します。 関連する統計情報: Sum
KNNScriptQueryRequests	スクリプトクエリの合計数のノード単位のメトリクス。この統計は、k-NN スコアスクリプト検索にのみ関係します。 関連する統計情報: Sum
KNNTotalLoadTime	k-NN がグラフをキャッシュにロードするのにかかった時間 (ナノ秒)。このメトリクスは、おおよその k-NN 検索にのみ関係します。 関連する統計情報: Sum

クラスター間検索のメトリクス

Amazon OpenSearch Service は、クラスター間検索 について以下のメトリクスを提供します。

ソースドメインのメトリクス

メトリクス	ディメンション	説明
CrossClusterOutboundConnections	ConnectionId	接続されたノードの数。スキップされたドメインが 1 つ以上レスポンスに含まれている場合は、このメトリクスを使用して異常な接続を追跡します。この数が 0 になった場合、その接続は正常ではありません。
CrossClusterOutboundRequests	ConnectionId	ターゲットドメインに送信された検索リクエストの数。ドメインでクラスター間検索リクエストが過負荷になっているかどうかを確認し、このメトリクスのスパイクと JVM/CPU スパイクの関連性を探るために使用します。

ターゲットドメインのメトリクス

メトリクス	ディメンション	説明
CrossClusterInboundRequests	ConnectionId	ソースドメインから受信した着信接続リクエストの数。

予期せず接続が失われた場合は、 CloudWatch アラームを追加します。アラームを作成する手順については、「静的しきい値 に基づいてアラームを作成する」を参照してください CloudWatch。

クラスター間レプリケーションメトリクス

Amazon OpenSearch Service は、クラスター間レプリケーション について以下のメトリクスを提供します。

メトリクス	説明
ReplicationRate	1 秒あたりのレプリケーションオペレーションの平均率。このメトリクスは IndexingRate メトリクスに似ています。
LeaderCheckPoint	特定の接続については、すべてのレプリケートインデックスにおけるリーダーチェックポイントの合計値。このメトリクスを使用して、レプリケーションのレイテンシーを測定できます。
FollowerCheckPoint	特定の接続については、すべてのレプリケートインデックスにおけるフォロワーチェックポイントの合計値。このメトリクスを使用して、レプリケーションのレイテンシーを測定できます。
ReplicationNumSyncingIndices	SYNCING のレプリケーションステータスを持つインデックスの数。
ReplicationNumBootstrappingIndices	BOOTSTRAPPING のレプリケーションステータスを持つインデックスの数。
ReplicationNumPausedIndices	PAUSED のレプリケーションステータスを持つインデックスの数。
ReplicationNumFailedIndices	FAILED のレプリケーションステータスを持つインデックスの数。
CrossClusterOutboundReplicationRequests	フォロワードメイン上のレプリケーショントランスポートリクエストの数。トランスポートリクエストは内部的なものであり、レプリケーション API オペレーションが呼び出されるたびに発生します。これらは、フォロワードメインがリーダードメインからの変更をポーリングする際にも発生します。

メトリクス	説明
CrossClusterInboundReplicationRequests	リーダードメイン上のレプリケーショントランスポートリクエストの数。トランスポートリクエストは内部的なものであり、レプリケーション API オペレーションが呼び出されるたびに発生します。
AutoFollowNumSuccessStartReplication	特定の接続のレプリケーションルールによって正常に作成されたフォロワーインデックスの数。
AutoFollowNumFailedStartReplication	一致するパターンがあったときにレプリケーションルールによって作成されなかったフォロワーインデックスの数。この問題は、リモートクラスターのネットワーク上の問題、またはセキュリティ上の問題 (すなわち、関連付けられているロールにレプリケーションを開始する許可がない) が原因となって発生する可能性があります。
AutoFollowLeaderCallFailure	新しいデータをプルするための、フォロワーインデックスからリーダーインデックスへのクエリが失敗したかどうか。1 の値は、直前の 1 分間に 1 つ以上の失敗した呼び出しがあったことを意味します。

Learning to Rank のメトリクス

Amazon OpenSearch Service では、 Learning to Rank に関する以下のメトリクスを提供しています。

メトリクス	説明
LTRRequestTotalCount	ランク付けリクエストの合計数。
LTRRequestErrorCount	失敗したリクエストの合計数。
LTRStatus.red	プラグインの実行に必要なインデックスの 1 つが赤であるかどうかを追跡します。

メトリクス	説明
LTRMemoryUsage	プラグインで使用されるメモリの合計。
LTRFeaturMemoryUsageInBytes	Learning to Rank 機能フィールドで使用されるメモリの量 (バイト単位)。
LTRFeaturesetMemoryUsageInBytes	すべての Learning to Rank 機能セットで使用されるメモリの量 (バイト単位)。
LTRModelMemoryUsageInBytes	すべての Learning to Rank モデルで使用されるメモリの量 (バイト単位)。

Piped Processing Language のメトリクス

Amazon OpenSearch Service は、パイプ処理言語 について以下のメトリクスを提供します。

メトリクス	説明
PPLFailedRequestCountByCusErr	クライアントの問題により失敗した _ppl API へのリクエストの数。例えば、IndexNotFoundException により、リクエストが HTTP ステータスコード 400 を返す場合があります。
PPLFailedRequestCountBySysErr	サーバーの問題または機能の制限により失敗した _ppl API へのリクエストの数。例えば、VerificationException により、リクエストが HTTP ステータスコード 503 を返す場合があります。
PPLRequestCount	_ppl API へのリクエストの数。

Amazon CloudWatch Logs による OpenSearch ログのモニタリング

Amazon OpenSearch Service は、Amazon CloudWatch Logs を通じて次の OpenSearch ログを公開します。

- エラーログ
- 検索リクエストのスローログ
- シャードスローログ
- 監査ログ

検索シャードスローログ、インデックス作成シャードスローログ、エラーログは、パフォーマンスと安定性の問題のトラブルシューティングに役立ちます。監査ログは、コンプライアンスの目的でユーザーのアクティビティを追跡します。すべてのログはデフォルトで無効になっています。有効にすると、標準 CloudWatch 料金が適用されます。

> **Note**
>
> エラーログは、 OpenSearch および Elasticsearch バージョン 5.1 以降でのみ使用できます。スローログは、すべての OpenSearch および Elasticsearch バージョンで使用できます。

ログには、Apache Log4j 2 とその組み込みログレベル (最も重要度が低いものから最も高いものまで) である `TRACE`、`DEBUG`、`INFO`、`WARN`、`ERROR`、 OpenSearch を使用します`FATAL`。

エラーログを有効にする`ERROR`と、 OpenSearch サービスは `WARN`、、および の ログ行`FATAL`を に発行します CloudWatch。また、 OpenSearch `DEBUG`レベルから次のようないくつかの例外も発行します。

- `org.opensearch.index.mapper.MapperParsingException`
- `org.opensearch.index.query.QueryShardException`
- `org.opensearch.action.search.SearchPhaseExecutionException`
- `org.opensearch.common.util.concurrent.OpenSearchRejectedExecutionException`
- `java.lang.IllegalArgumentException`

以下のように、エラーログは多くの状況でトラブルシューティングに役立ちます。

- Painless スクリプトのコンパイルの問題
- 無効なクエリ
- インデックス作成の問題
- スナップショットの失敗
- Index State Management の移行の失敗

トピック

ログ発行を有効にする (コンソール)

OpenSearch サービスコンソールは、 へのログの発行を有効にする最も簡単な方法です CloudWatch。

へのログ発行を有効にするには CloudWatch （ コンソール ）

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。
2. Analytics で、Amazon OpenSearch Service を選択します。
3. 更新するドメインを選択します。
4. [ログ] タブで、ログタイプを選択し、[有効化] を選択します。
5. 新しい CloudWatch ロググループを作成するか、既存のロググループを選択します。

> **Note**
>
> 複数のログを有効にする場合は、各ログを個別にロググループに発行することをお勧めします。分離することで、ログをスキャンしやすくなります。

6. 適切なアクセス権限を含むアクセスポリシーを選択するか、コンソールに用意された JSON を使用してポリシーを作成します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn:*"
    }
  ]
}
```

Confused Deputy Problem (混乱した代理の問題) から自分を守るために、aws:SourceAccount および aws:SourceArn の条件キーをポリシーに追加することをお勧めします。ソースアカウントはドメインの所有者であり、ソース ARN はドメインの ARN です。これらの条件キーを追加するには、ドメインがサービスソフトウェア R20211203 以降にある必要があります。

例えば、次の条件ブロックをポリシーに追加できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
```

```
}
```

> **Important**
>
> CloudWatch ログは、リージョン ごとに 10 個のリソースポリシーをサポートします。複数の OpenSearch サービスドメインのログを有効にする場合は、この制限に達しないように、複数のロググループを含むより広範なポリシーを作成して再利用する必要があります。ポリシーを更新する手順については、「the section called “ログ発行の有効化 (AWS CLI)”」を参照してください。

7. [有効] を選択します。

   ドメインのステータスが [アクティブ] から [処理中] に変わります。ステータスは、ログの発行が有効になる前に [アクティブ] に戻る必要があります。この変更には通常 30 分かかりますが、ドメイン設定によってはさらに時間がかかる場合があります。

シャードスローログのいずれかを有効にした場合は、「」を参照してくださいthe section called “シャードスローログしきい値の設定”。監査ログを有効にした場合は、「the section called “ステップ 2: OpenSearch ダッシュボードで監査ログを有効にする”」を参照してください。エラーログのみ有効にしている場合は、追加のステップを行う必要はありません。

ログ発行の有効化 (AWS CLI)

ログの発行を有効にする前に、 CloudWatch ロググループが必要です。まだロググループがない場合は、次のコマンドを使用して作成できます。

```
aws logs create-log-group --log-group-name my-log-group
```

次のコマンドを入力してロググループの ARN を検索し、それを書き留めます。

```
aws logs describe-log-groups --log-group-name my-log-group
```

これで、ロググループに書き込むアクセス許可を OpenSearch サービスに付与できます。ロググループの末尾に近い位置にロググループの ARN を指定する必要があります。

```
aws logs put-resource-policy \
  --policy-name my-policy \
```

```
  --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Sid": "",
 "Effect": "Allow", "Principal": { "Service": "es.amazonaws.com"}, "Action":
[ "logs:PutLogEvents","logs:CreateLogStream"],"Resource": "cw_log_group_arn:*"}]}'
```

> **Important**
>
> CloudWatch ログは、リージョン ごとに 10 個のリソースポリシーをサポートします。複数の OpenSearch サービスドメインでシャードスローログを有効にする場合は、この制限に達しないように、複数のロググループを含むより広範なポリシーを作成して再利用する必要があります。

後でこのポリシーを確認する必要がある場合は、`aws logs describe-resource-policies` コマンドを使用します。ポリシーを更新するには、新しいポリシードキュメントで同じ `aws logs put-resource-policy` コマンドを実行します。

最後に、`--log-publishing-options` オプションを使用して発行を有効化できます。オプションの構文は、`create-domain` コマンドと `update-domain-config` コマンドのどちらでも同じです。

パラメータ	有効な値
`--log-publishing-options`	`SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`ES_APPLICATION_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`AUDIT_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`

> **Note**
>
> 複数のログを有効にする場合は、各ログを個別にロググループに発行することをお勧めします。分離することで、ログをスキャンしやすくなります。

例

次の例では、指定されたドメインの検索およびインデックス作成シャードスローログの発行を有効にします。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --log-publishing-options
 "SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group,Enabled=true},INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-other-log-group,Enabled=true}"
```

への発行を無効にするには CloudWatch、 で同じコマンドを実行しますEnabled=false。

シャードスローログのいずれかを有効にした場合は、「」を参照してくださいthe section called “シャードスローログしきい値の設定”。監査ログを有効にした場合は、「the section called “ステップ 2: OpenSearch ダッシュボードで監査ログを有効にする”」を参照してください。エラーログのみ有効にしている場合は、追加のステップを行う必要はありません。

ログ発行の有効化 (AWS SDK)

ログ発行を有効にする前に、まず CloudWatch ロググループを作成し、その ARN を取得し、そのグループに書き込むためのアクセス許可を OpenSearch サービスに付与する必要があります。関連するオペレーションについては、「Amazon CloudWatch Logs API リファレンス」を参照してください。

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

これらのオペレーションには、AWS SDK を使用してアクセスできます。

AWS SDKs (Android および iOS SDKs) は、 `CreateDomain`および `--log-publishing-options`のオプションを含め、Amazon OpenSearch Service API リファレンス で定義されているすべてのオペレーションをサポートします`UpdateDomainConfig`。

シャードスローログのいずれかを有効にした場合は、「」を参照してくださいthe section called "シャードスローログしきい値の設定"。エラーログのみ有効にしている場合は、追加のステップを行う必要はありません。

ログ発行の有効化 (CloudFormation)

この例では、 CloudFormation を使用して というロググループを作成し`opensearch-logs`、適切なアクセス許可を割り当て、アプリケーションログ、シャードスローログの検索、スローログのインデックス作成のためにログの発行を有効にしたドメインを作成します。

ログ発行を有効にする前に、 CloudWatch ロググループを作成する必要があります。

```
Resources:
  OpenSearchLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: opensearch-logs
Outputs:
  Arn:
    Value:
      'Fn::GetAtt':
        - OpenSearchLogGroup
        - Arn
```

テンプレートは、ロググループの ARN を出力します。この場合、ARNは `arn:aws:logs:us-east-1:123456789012:log-group:opensearch-logs` です。

ARN を使用して、ロググループに書き込むアクセス許可を OpenSearch サービスに付与するリソースポリシーを作成します。

```
Resources:
 OpenSearchLogPolicy:
   Type: AWS::Logs::ResourcePolicy
   Properties:
     PolicyName: my-policy
     PolicyDocument: "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"\",
 \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"es.amazonaws.com\"}, \"Action
```

```
\":[ \"logs:PutLogEvents\",\"logs:CreateLogStream\"],\"Resource\": \"arn:aws:logs:us-
east-1:123456789012:log-group:opensearch-logs:*\"}]}"
```

最後に、次の CloudFormation スタックを作成します。これにより、ログの発行を含む OpenSearch サービスドメインが生成されます。アクセスポリシーは、 のユーザーがドメイン AWS アカウント へのすべての HTTP リクエストを行うことを許可します。

```
Resources:
  OpenSearchServiceDomain:
    Type: "AWS::OpenSearchService::Domain"
    Properties:
      DomainName: my-domain
      EngineVersion: "OpenSearch_1.0"
      ClusterConfig:
        InstanceCount: 2
        InstanceType: "r6g.xlarge.search"
        DedicatedMasterEnabled: true
        DedicatedMasterCount: 3
        DedicatedMasterType: "r6g.xlarge.search"
      EBSOptions:
        EBSEnabled: true
        VolumeSize: 10
        VolumeType: "gp2"
      AccessPolicies:
        Version: "2012-10-17"
        Statement:
            Effect: "Allow"
            Principal:
                AWS: "arn:aws:iam::123456789012:user/es-user"
            Action: "es:*"
            Resource: "arn:aws:es:us-east-1:123456789012:domain/my-domain/*"
      LogPublishingOptions:
        ES_APPLICATION_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        SEARCH_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        INDEX_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
```

```
          Enabled: true
```

構文の詳細については、AWS CloudFormation ユーザーガイドの「ログの発行オプション」を参照してください。

検索リクエストのスローログしきい値の設定

検索リクエストのスローログは、バージョン 2.13 以降で実行されている OpenSearch サービスドメインでの検索に使用できます。検索リクエストのスローログのしきい値は、リクエストにかかった合計時間に対して設定されます。これは、個々のシャードに時間がかかったように設定されたシャードリクエストのスローログとは異なります。

検索リクエストのスローログは、クラスター設定で指定できます。これは、インデックス設定で有効にするシャードスローログとは異なります。例えば、 OpenSearch REST API を使用して次の設定を指定できます。

```
PUT domain-endpoint/_cluster/settings
{
  "transient": {
    "cluster.search.request.slowlog.threshold.warn": "5s",
    "cluster.search.request.slowlog.threshold.info": "2s"
  }
}
```

シャードスローログしきい値の設定

OpenSearch はデフォルトでシャードスローログを無効にします。へのシャードスローログの発行を有効にした後も CloudWatch、 OpenSearch インデックスごとにログ記録しきい値を指定する必要があります。これらのしきい値は、ログに記録する内容とログレベルを正確に定義します。

例えば、 OpenSearch REST API を使用してこれらの設定を指定できます。

```
PUT domain-endpoint/index/_settings
{
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

スローログのテスト

検索リクエストとシャードスローログの両方が正常に発行されていることをテストするには、非常に低い値から始めてログが に表示されることを確認し CloudWatch、しきい値をより有用なレベルに引き上げることを検討してください。

ログが表示されない場合は、以下を確認してください。

- CloudWatch ロググループが存在していますか？ CloudWatch コンソールを確認します。
- OpenSearch サービスにはロググループに書き込むアクセス許可がありますか？ OpenSearch サービスコンソールを確認します。
- OpenSearch サービスドメインはロググループに発行するように設定されていますか？ OpenSearch サービスコンソールを確認するか、 AWS CLI `describe-domain-config`オプションを使用するか、 SDK のいずれか`DescribeDomainConfig`を使用して を呼び出します。 SDKs
- OpenSearch ログ記録のしきい値は、リクエストがしきい値を超えているほど低くなっていますか？

  ドメインの検索リクエストのスローログしきい値を確認するには、次のコマンドを使用します。

  ```
  GET domain-endpoint/_cluster/settings?flat_settings
  ```

  インデックスのシャードスローログしきい値を確認するには、次のコマンドを使用します。

  ```
  GET domain-endpoint/index/_settings?pretty
  ```

インデックスのスローログを無効にする場合は、変更したしきい値をデフォルト値の -1 に戻します。

OpenSearch サービスコンソールまたは CloudWatch を使用して への発行を無効にしても、ログの生成 OpenSearch は停止 AWS CLI されません。これらのログの発行のみが停止します。シャードスローログが不要になった場合はインデックス設定を、検索リクエストスローログが不要になった場合はドメイン設定を必ず確認してください。

ログの表示

アプリケーションの表示と でのスローログの表示 CloudWatch は、他の CloudWatch ログの表示と同じです。詳細については、「Amazon Logs ユーザーガイド」の「ログデータの表示」を参照してください。 CloudWatch

ログを表示する際の考慮事項は以下のとおりです。

- OpenSearch サービスは、各行の最初の 255,000 文字のみを に発行します CloudWatch。残りのコンテンツは切り捨てられます。監査ログの場合、それはメッセージあたり 10,000 文字です。
- では CloudWatch、ログストリーム名に 、`-index-slow-logs`、`-search-slow-logs`、およびのサフィックス`-audit-logs`があり`-application-logs`、その内容を識別しやすくなっています。

Amazon OpenSearch サービスの監査ログの監視

Amazon OpenSearch Service ドメインがきめ細かいアクセスコントロールを使用している場合は、データの監査ログを有効にできます。監査ログは高度にカスタマイズ可能で、認証の成功と失敗、リクエスト、インデックスの変更、受信した検索クエリなど OpenSearch、 OpenSearchクラスター上のユーザーアクティビティを追跡できます。デフォルトの設定では、一般的な一連のユーザーアクションが追跡されますが、正確なニーズに合わせて設定を調整することをお勧めします。

OpenSearch アプリケーションログやスローログと同様に、 OpenSearch Service CloudWatch は監査ログを Logs に公開します。有効にすると、 CloudWatch 標準料金が適用されます。

> **Note**
>
> 監査ログを有効にするには、`security_manager`ユーザーロールをそのロールにマッピングして OpenSearch `plugins/_security` REST API にアクセスできるようにする必要があります。詳細については、「the section called “マスターユーザーの変更”」を参照してください。

トピック

制限事項

監査ログには以下の制限事項があります。

- 監査ログには、送信先のドメインアクセスポリシーによって拒否されたクロスクラスター検索リクエストは含まれません。
- 各監査ログメッセージの最大サイズは 10,000 文字です。この制限を超えると、監査ログメッセージが切り捨てられます。

監査ログ記録の有効化

監査ログの有効化は、2 段階のプロセスです。まず、監査ログを Logs に公開するようにドメインを設定します。 CloudWatch 次に、 OpenSearch ダッシュボードで監査ログを有効にし、ニーズに合わせて構成します。

> **Important**
>
> これらの手順の実行中にエラーが発生した場合、トラブルシューティング情報については、the section called “監査ログを有効にできない” を参照してください。

ステップ 1: 監査ログを有効にし、アクセスポリシーを設定する

次の手順では、コンソールを使用して監査ログを有効にする方法について説明します。また、 AWS CLI、OpenSearch またはサービス API を使用して有効化することもできます。

OpenSearch サービスドメイン (コンソール) の監査ログを有効にするには

1. ドメインを選択して設定を開き、[ログ] タブに移動します。
2. [監査ログ] を選択してから、[有効化] を選択します。
3. CloudWatch ロググループを作成するか、既存のロググループを選択します。
4. 適切なアクセス権限を含むアクセスポリシーを選択するか、コンソールに用意された JSON を使用してポリシーを作成します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn"
    }
  ]
}
```

Confused Deputy Problem (混乱した代理の問題) から自分を守るため
に、aws:SourceAccount および aws:SourceArn の条件キーをポリシーに追加することをお勧めします。ソースアカウントはドメインの所有者であり、ソース ARN はドメインの ARN です。これらの条件キーを追加するには、ドメインがサービスソフトウェア R20211203 以降にある必要があります。

例えば、次の条件ブロックをポリシーに追加できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

5. [Enable (有効化)] を選択します。

ステップ 2: OpenSearch ダッシュボードで監査ログを有効にする

OpenSearch サービスコンソールで監査ログを有効にしたら、 OpenSearch ダッシュボードでも監査ログを有効にし、ニーズに合わせて設定する必要があります。

1. [OpenSearch ダッシュボード] を開き、左側のメニューから [セキュリティ] を選択します。
2. [監査ログ] を選択します。
3. [監査ログ作成を有効にする] を選択します。

Dashboards UI には、[全般設定] および [コンプライアンス設定] の下で、監査ログの設定の完全なコントロールが用意されています。すべての設定オプションの説明については、「監査ログの設定」を参照してください。

を使用して監査ログを有効にします。 AWS CLI

AWS CLI 以下のコマンドは、既存のドメインの監査ログを有効にします。

```
aws opensearch update-domain-config --domain-name my-domain --log-publishing-options
 "AUDIT_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-
group:my-log-group,Enabled=true}"
```

ドメインを作成するときに、監査ログを有効にすることもできます。詳細については、「AWS CLI コマンドのリファレンス」を参照してください。

設定 API を使用して監査ログを有効にする

次の設定 API へのリクエストにより、既存のドメインで監査ログが有効になります。

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "LogPublishingOptions": {
    "AUDIT_LOGS": {
      "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group1:sample-domain",
      "Enabled":true
    }
  }
}
```

詳細については、Amazon OpenSearch サービス API リファレンスをご覧ください。

監査ログのレイヤーとカテゴリ

クラスターコミュニケーションは、2 つの別々のレイヤーを介して行われます: RESTレイヤーとトランスポートレイヤー。

- REST レイヤーは、curl、Logstash、 OpenSearch ダッシュボード、Java 高レベル REST クライアント、Python リクエストライブラリなどの HTTP クライアント、つまりクラスターに到達するすべての HTTP リクエストとの通信をカバーします。

- トランスポートレイヤーは、ノード間のコミュニケーションを対象としています。例えば、検索リクエストが（REST レイヤーを介して）クラスターに到着した後、リクエストを処理する調整ノードは、クエリを他のノードに送信し、そのレスポンスを受け取り、必要なドキュメントを収集し、それらを最終的なレスポンスと照合します。シャード割り当てや再分散などのオペレーションは、トランスポートレイヤーでも実行されます。

レイヤー全体の監査ログと、レイヤーの個々の監査カテゴリを有効または無効にできます。次の表に、監査カテゴリの概要と、監査カテゴリが使用可能なレイヤーを示します。

カテゴリ	説明	REST で利用可能	トランスポートで利用可能
FAILED_LOGIN	リクエストに無効な資格情報が含まれ、認証に失敗しました。	はい	はい
MISSING_PRIVILEGES	ユーザーには、リクエストを行う権限がありませんでした。	はい	はい
GRANTED_PRIVILEGES	ユーザーには、リクエストを行う権限がありました。	はい	はい
OPENSEARCH_SECURITY_INDEX_ATTEMPT	リクエストが `.opendistro_security` インデックスを修正しようとしました。	いいえ	はい
AUTHENTICATED	リクエストに有効な認証情報が含まれ、認証に成功しました。	はい	はい
INDEX_EVENT	リクエストは、インデックスの作成、エイリアスの設定、強制マージの実行など、	いいえ	はい

カテゴリ	説明	REST で利用可能	トランスポートで利用可能
	インデックスの管理オペレーションを実行しました。indices:admin/ このカテゴリに含まれるアクションの全リストは、ドキュメントに記載されています。OpenSearch		

これらの標準カテゴリに加えて、きめ細かなアクセスコントロールには、データコンプライアンス要件を満たすように設計されたいくつかの追加カテゴリが用意されています。

カテゴリ	説明
COMPLIANCE_DOC_READ	リクエストは、インデックス内のドキュメントに対して読み取りイベントを実行しました。
COMPLIANCE_DOC_WRITE	リクエストは、インデックス内のドキュメントに対して書き込みイベントを実行しました。
COMPLIANCE_INTERNAL_CONFIG_READ	リクエストは、.opendistro_security インデックスで読み取りイベントを実行しました。
COMPLIANCE_INTERNAL_CONFIG_WRITE	リクエストは、.opendistro_security インデックスで書き込みイベントを実行しました。

カテゴリおよびメッセージ属性の任意の組み合わせを指定できます。例えば、REST リクエストを送信してドキュメントのインデックスを作成すると、監査ログに次の行が表示されることがあります。

- REST レイヤーで AUTHENTICATED (認証)
- トランスポートレイヤーで GRANTED_PREVIEWE (認可)

- COMPLIANCE_DOC_WRITE (インデックスに書き込まれたドキュメント)

監査ログの設定

監査ログには、多数の設定オプションがあります。

全般設定

全般設定では、個々のカテゴリまたはレイヤー全体を有効または無効にできます。除外カテゴリとして GRANTED_Privileges および Authenticated を残すことを強くお勧めします。それ以外の場合、これらのカテゴリは、クラスターへの有効なリクエストごとにログされます。

名前	バックエンドの設定	説明
REST レイヤー	enable_rest	REST レイヤーで発生するイベントを有効または無効にします。
REST 無効なカテゴリ	disabled_rest_categories	REST レイヤーで無視する監査カテゴリを指定します。これらのカテゴリを変更すると、監査ログのサイズが大幅に増加する可能性があります。
トランスポートレイヤー	enable_transport	トランスポートレイヤーで発生するイベントを有効または無効にします。
トランスポート無効なカテゴリ	disabled_transport_categories	トランスポートレイヤーで無視する必要がある監査カテゴリを指定します。これらのカテゴリを変更すると、監査ログのサイズが大幅に増加する可能性があります。

属性設定では、各ログ行の詳細量をカスタマイズできます。

名前	バックエンドの設定	説明
一括リクエスト	resolve_bulk_requests	この設定を有効にすると、一括要求でドキュメントごとにログが生成されます。これにより、監査ログのサイズが大幅に増加する可能性があります。

名前	バックエンドの設定	説明
リクエスト本文	log_request_body	リクエストのリクエストボディを含めます。
インデックスの解決	resolve_indices	インデックスに対してエイリアスを解決します。

無視設定を使用して、一連のユーザーまたは API パスを除外します。

名前	バックエンドの設定	説明
無視されたユーザー	ignore_users	除外するユーザーを指定します。
無視されたリクエスト	ignore_requests	除外するリクエストパターンを指定します。

コンプライアンス設定

コンプライアンス設定では、インデックス、ドキュメント、またはフィールドレベルのアクセスを調整できます。

名前	バックエンドの設定	説明
コンプライアンスのロギング	enable_compliance	コンプライアンスのロギングを有効または無効にします。

読み取りおよび書き込みイベントのロギングには以下の設定を指定できます。

名前	バックエンドの設定	説明
内部設定の口ギング	internal_config	`.opendistro_security` インデックスのイベントの口ギングを有効または無効にします。

読み取りイベントには以下の設定を指定できます。

名前	バックエンドの設定	説明
メタデータの読み取り	read_metadata_only	読み取りイベントのメタデータのみを含めます。ドキュメントフィールドを含めないでください。
無視されたユーザー	read_ignore_users	読み取りイベントには特定のユーザーを含めないでください。
監視するフィールド	read_watched_fields	読み取りイベントを監視するインデックスとフィールドを指定します。監視するフィールドを追加すると、ドキュメントアクセスごとに 1 つの口グが生成されます。これにより、監査口グのサイズが大幅に増加する可能性があります。監視するフィールドは、インデックスパターンとフィールドパターンをサポートします。 `{ "index-name-pattern": [ "field-name-pattern" ], "logs*": [ "message" ], "twitter": [ "id", "user*" ] }`

書き込みイベントには以下の設定を指定できます。

名前	バックエンドの設定	説明
メタデータの書き込み	write_metadata_only	書き込みイベントのメタデータのみを含めます。ドキュメントフィールドを含めないでください。
差分のログ	write_log_diffs	write_metadata_only が false の場合は、書き込みイベント間の違いのみを含めます。
無視されたユーザー	write_ignore_users	書き込みイベントには特定のユーザーを含めないでください。
インデックスの監視	write_watched_indices	書き込みイベントを監視するインデックスまたはインデックスパターンを指定します。監視するフィールドを追加すると、ドキュメントアクセスごとに 1 つのログが生成されます。これにより、監査ログのサイズが大幅に増加する可能性があります。

監査ログの例

このセクションには、設定例、検索リクエスト、およびインデックスのすべての読み取りおよび書き込みイベントの結果となる監査ログが含まれます。

ステップ 1: 監査ログを設定する

ロググループへの監査ログの公開を有効にしたら、 CloudWatch OpenSearch ダッシュボードの監査ログページに移動して [監査ログを有効にする] を選択します。

1. [全般設定] で、[設定] を選択し、[REST レイヤー] が有効であることを確認します。
2. [コンプライアンス設定] で、[設定] を選択します。
3. [書き込み] の下で、[監視フィールド] に、追加このインデックスへのすべての書き込みイベントのために `accounts` を追加します。
4. [読み取り] の下で、[監視フィールド] に、`accounts` インデックスの `ssn` および `id-` フィールドを追加します。

```
{
```

```
  "accounts-": [
    "ssn",
    "id-"
  ]
}
```

ステップ 2: 読み取りイベントと書き込みイベントを実行する

1. [OpenSearch ダッシュボード] に移動し、[開発ツール] を選択して、サンプルドキュメントにインデックスを付けます。

```
PUT accounts/_doc/0
{
  "ssn": "123",
  "id-": "456"
}
```

2. 読み取りイベントをテストするには、次のリクエストを送信します。

```
GET accounts/_search
{
  "query": {
    "match_all": {}
  }
}
```

ステップ 3: ログを確認する

1. https://console.aws.amazon.com/cloudwatch/ CloudWatch でコンソールを開きます。
2. ナビゲーションペインで、[ロググループ] を選択します。
3. 監査ログを有効にするときに指定したロググループを選択します。ロググループ内では、OpenSearch Service はドメイン内のノードごとにログストリームを作成します。
4. [ログストリーム] で、[すべて検索] を選択します。
5. 読み取りおよび書き込みイベントについては、対応するログを参照してください。ログが表示される前に、5 秒の遅延が予想されます。

   監査ログの書き込みの例

```
{
  "audit_compliance_operation": "CREATE",
  "audit_cluster_name": "824471164578:audit-test",
  "audit_node_name": "be217225a0b77c2bd76147d3ed3ff83c",
  "audit_category": "COMPLIANCE_DOC_WRITE",
  "audit_request_origin": "REST",
  "audit_compliance_doc_version": 1,
  "audit_node_id": "3xNJhm4XS_yTzEgDWcGRjA",
  "@timestamp": "2020-08-23T05:28:02.285+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "3.236.145.227",
  "audit_trace_doc_id": "lxnJGXQBqZSlDB91r_uZ",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 8,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
}
```


監査ログの読み取りの例


```
{
  "audit_cluster_name": "824471164578:audit-docs",
  "audit_node_name": "806f6050cb45437e2401b07534a1452f",
  "audit_category": "COMPLIANCE_DOC_READ",
  "audit_request_origin": "REST",
  "audit_node_id": "saSevm9ASte0-pjAtYi2UA",
  "@timestamp": "2020-08-31T17:57:05.015+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "54.240.197.228",
  "audit_trace_doc_id": "config:7.7.0",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 0,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
```

```
}
```

リクエスト本文を含めるには、 OpenSearch ダッシュボードの [コンプライアンス設定] に戻り、[メタデータの書き込み] を無効にします。特定のユーザーによるイベントを除外するには、そのユーザーを [無視されたユーザー] に追加します。

各監査ログフィールドの説明については、「監査ログフィールドのリファレンス]を参照してください。監査ログデータの検索と分析については、Amazon Logs ユーザーガイドの「 CloudWatch CloudWatch ログインサイトによるログデータの分析」を参照してください。

REST API を使用した監査ログの設定

OpenSearch 監査ログの設定にはダッシュボードを使用することをお勧めしますが、きめ細かいアクセス制御 REST API を使用することもできます。このセクションには、リクエストの例が含まれています。REST API に関する詳細なドキュメントは、このドキュメントにあります。OpenSearch

```
PUT _opendistro/_security/api/audit/config
{
  "enabled": true,
  "audit": {
    "enable_rest": true,
    "disabled_rest_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "enable_transport": true,
    "disabled_transport_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "resolve_bulk_requests": true,
    "log_request_body": true,
    "resolve_indices": true,
    "exclude_sensitive_headers": true,
    "ignore_users": [
      "kibanaserver"
    ],
    "ignore_requests": [
      "SearchRequest",
      "indices:data/read/*",
      "/_cluster/health"
```

```
    ]
  },
  "compliance": {
    "enabled": true,
    "internal_config": true,
    "external_config": false,
    "read_metadata_only": true,
    "read_watched_fields": {
      "read-index-1": [
        "field-1",
        "field-2"
      ],
      "read-index-2": [
        "field-3"
      ]
    },
    "read_ignore_users": [
      "read-ignore-1"
    ],
    "write_metadata_only": true,
    "write_log_diffs": false,
    "write_watched_indices": [
      "write-index-1",
      "write-index-2",
      "log-*",
      "*"
    ],
    "write_ignore_users": [
      "write-ignore-1"
    ]
  }
}
```

Amazon OpenSearch によるサービスイベントのモニタリング EventBridge

Amazon OpenSearch Service は Amazon EventBridge と統合して、ドメインに影響する特定のイベントを通知します。 AWS サービスからのイベントは、 EventBridge ほぼリアルタイムでに配信されます。同じイベントが、 CloudWatch Amazonの前身であるAmazonイベントにも送信されます。EventBridge簡単なルールを記述して、注目するイベントと、イベントがルールに一致した場合に自

動的に実行するアクションを指定できます。自動的にトリガーできるオペレーションには、以下が含まれます。

- 関数を呼び出す AWS Lambda
- Amazon EC2 Run Command の呼び出し
- Amazon Kinesis Data Streams へのイベントの中継
- AWS Step Functions ステートマシンのアクティベーション
- Amazon SNS トピックまたは Amazon SQS キューの通知

詳細については、Amazon EventBridge EventBridge ユーザーガイドの「Amazon を使い始める」を参照してください。

トピック

サービスソフトウェア更新イベント

OpenSearch EventBridge 以下のサービスソフトウェアの更新イベントのいずれかが発生すると、サービスがイベントを送信します。

サービスソフトウェア更新が利用可能

OpenSearch サービスソフトウェアの更新が可能になると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update R20220928 available. Service Software
 Deployment Mechanism:
                   Blue/Green. For more information on deployment configuration,
 please
                   see: https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

サービスソフトウェア更新をスケジュールしました

OpenSearch サービスソフトウェアの更新がスケジュールされると、サービスからこのイベントが送信されます。オプションの更新の場合、予定日に通知が届きます。再スケジュールはいつでも可能です。必要な更新の場合、予定日の 3 日前に通知が届きます。再スケジュールは、決められた期間内に行えます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
```

```
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Scheduled",
    "severity": "High",
    "description": "A new service software update [R20200330-p1] has been scheduled at
 [21st May 2023 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

サービスソフトウェア更新を再スケジュールしました

OpenSearch オプションのサービスソフトウェアアップデートが再スケジュールされたときに、サービスからこのイベントが送信されます。詳細については、「the section called “オプションの更新と必須の更新”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Rescheduled",
    "severity": "High",
    "description": "The service software update [R20200330-p1], which was originally
 scheduled for
                    [21st May 2023 12:40 GMT], has been rescheduled to [23rd May 2023
 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
```

```
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

サービスソフトウェア更新が開始しました

OpenSearch サービスソフトウェアの更新が開始されると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Started",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] started.
  }
}
```

サービスソフトウェアの更新が完了しました

OpenSearch サービスソフトウェアの更新が完了すると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
```

```
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Completed",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] completed."
  }
}
```

サービスソフトウェア更新をキャンセルしました

OpenSearch サービスソフトウェアの更新がキャンセルされると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled as a
                    newer update is available. Please schedule the latest update."
  }
}
```

スケジュールされたサービスソフトウェア更新をキャンセルしました

OpenSearch 以前にドメインに対して予定されていたサービスソフトウェアの更新がキャンセルされたときに、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled."
  }
}
```

サービスソフトウェア更新が実行されませんでした

OpenSearch サービスは、サービスソフトウェアの更新を開始できない場合にこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
```

```
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Unexecuted",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] cannot be
 started. Reason: [reason]"
  }
}
```

サービスソフトウェアの更新に失敗しました

OpenSearch サービスソフトウェアの更新が失敗すると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Failed",
    "severity": "High",
    "description": "Installation of service software update [R20200330-p1] failed.
 [reason].
  }
}
```

サービスソフトウェア更新が必要

OpenSearch サービスソフトウェアの更新が必要な場合、サービスはこのイベントを送信します。詳細については、「the section called “オプションの更新と必須の更新”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Required",
    "severity": "High",
    "description": "Service software update [R20200330-p1] available. Update
                    will be automatically installed after [21st May 2023] if no
                    action is taken. Service Software Deployment Mechanism: Blue/Green.
                    For more information on deployment configuration, please see:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

Auto-Tune イベント

OpenSearch 以下のAuto-Tune EventBridge イベントのいずれかが発生すると、サービスがイベントを送信します。

Auto-Tune 保留中

OpenSearch Auto-Tune がクラスタのパフォーマンスと可用性を向上させるためのチューニングの推奨事項を特定すると、サービスはこのイベントを送信します。このイベントは、Auto-Tune が無効になっているドメインに対してのみ表示されます。

例

このタイプのイベントの例を以下に示します。

```
{
```

```
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Pending",
    "description": "Auto-Tune recommends the following new settings for your
 domain: { JVM Heap size : 60%}. Enable Auto-Tune to improve cluster stability and
 performance.",
    "scheduleTime": "{iso8601-timestamp}"
  }
}
```

Auto-Tune が開始されました

OpenSearch Auto-Tune がドメインに新しい設定を適用し始めると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Started",
    "scheduleTime": "{iso8601-timestamp}",
    "startTime": "{iso8601-timestamp}",
```

```
    "description" : "Auto-Tune is applying the following settings to your domain: { JVM
 Heap size : 60%}."
  }
}
```

Auto-Tune には、スケジュールされた Blue/Green のデプロイが必要です。

OpenSearch Auto-Tune がブルー/グリーンデプロイのスケジュールを必要とするチューニングの推奨事項を特定すると、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Pending",
    "startTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has identified the following settings for your domain
 that require a blue/green deployment: { JVM Heap size : 60%}.
                    You can schedule the deployment for your preferred time."
  }
}
```

Auto-Tune がキャンセルされました

OpenSearch 保留中のチューニングに関する推奨事項がないため、Auto-Tune スケジュールがキャンセルされたときに、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Cancelled",
    "scheduleTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has cancelled the upcoming blue/green deployment."
  }
}
```

自動チューニングが完了しました

OpenSearch Auto-Tune が Blue/Green デプロイメントを完了し、クラスタが新しい JVM 設定で稼働状態になったときに、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "completionTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has completed the blue/green deployment and successfully
 applied the following settings: { JVM Heap size : 60%}."
```

```
  }
}
```

Auto-Tune が無効になり、変更が元に戻されました

OpenSearch Auto-Tune が無効化され、適用された変更がロールバックされると、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:es:us-east-1:123456789012:domain/test-domain" ],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "description": "Auto-Tune is now disabled. All settings have been reverted. Auto-
Tune will continue to evaluate
                    cluster performance and provide recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
}
```

Auto-Tune が無効になり、変更が保持されました

OpenSearch Auto-Tune が無効化され、適用された変更が保持されている場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
```

```
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "description": "Auto-Tune is now disabled. The most-recent settings by Auto-Tune
 have been retained.
                   Auto-Tune will continue to evaluate cluster performance and provide
 recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
}
```

クラスターヘルスイベント

OpenSearch EventBridge クラスターの状態が悪化したときに、サービスが特定のイベントを送信します。

赤いクラスターの復旧が開始しました

OpenSearch クラスターのステータスが 1 時間以上赤くなり続けると、サービスからこのイベントが送信されます。クラスタステータスを修正するために、スナップショットから赤いインデックスの自動復元が試行されます。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
```

```
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Started",
      "severity":"High",
      "description":"Your cluster status is red. We have started automatic snapshot
 restore for the red indices.
                     No action is needed from your side. Red indices [red-index-0, red-
index-1]"
   }
}
```

赤いクラスタの復旧が部分的に完了しました

OpenSearch 赤色のクラスターの状態を修正しようとしたときに、スナップショットから赤色のインデックスのサブセットしか復元できなかった場合に、サービスがこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Partially Restored",
      "severity":"High",
      "description":"Your cluster status is red. We were able to restore the following
 Red indices from
                    snapshot: [red-index-0]. Indices not restored: [red-index-1].
 Please refer https://docs.aws.amazon.com/opensearch-service/latest/developerguide/
handling-errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
```

```
}
```

赤いクラスターの復旧に失敗しました

OpenSearch 赤色のクラスターの状態を修正しようとしてインデックスの復元に失敗すると、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Failed",
      "severity":"High",
      "description":"Your cluster status is red. We were unable to restore the Red
 indices automatically.
                    Indices not restored: [red-index-0, red-index-1]. Please refer
 https://docs.aws.amazon.com/opensearch-service/latest/developerguide/handling-
errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
}
```

削除するシャード

OpenSearch 赤色のクラスターの状態が 14 日間赤色の状態が続いた後、自動的に修正しようとしたが、1 つ以上のインデックスが赤色のままである場合に、サービスからこのイベントが送信されます。さらに 7 日後 (合計 21 日連続赤色)、 OpenSearch Service はすべての赤色のインデックスから割り当てられていないシャードの削除を続行します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:36:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"Medium",
      "description":"Your cluster status is red. Please fix the red indices as soon as
 possible.
                    If not fixed by 2022-04-12 01:51:47+00:00, we will delete all
 unassigned shards,
                    the unit of storage and compute, for these red indices to recover
 your domain and make it green.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.
                    test_data, test_data1",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) to be deleted"
   }
}
```

シャードが削除されました

OpenSearch クラスターのステータスが 21 日間連続して赤になった後、サービスからこのイベントが送信されます。すべての赤いインデックスの未割り当てシャード (ストレージとコンピューティング) が削除されます。詳細については、「the section called “赤いクラスターの自動修復”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
```

```
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:54:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"High",
      "description":"We have deleted unassinged shards, the unit of storage and
 compute, in
                    red indices: index-1, index-2 because these indices were red for
 more than
                    21 days and could not be restored with the automated restore
 process.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) deleted"
   }
}
```

シャードカウントが高い警告

OpenSearch ホットデータノード全体の平均シャード数が、推奨デフォルトの上限である 1,000 個の 90% を超えたときに、サービスからこのイベントが送信されます。Elasticsearch の新しいバージョンでは、 OpenSearch ノードあたりの最大シャード数の制限を設定できるようになっていますが、ノードあたりのシャード数は 1,000 以下にすることをお勧めします。シャード数の選択を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
```

```
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Low",
     "description":"One or more data nodes have close to 1000 shards. To ensure optimum
 performance and stability of your
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

シャード数の制限を超えました

OpenSearch ホットデータノード全体の平均シャード数が推奨デフォルトの上限である 1,000 を超えたときに、サービスからこのイベントが送信されます。Elasticsearch の新しいバージョンでは、OpenSearch ノードあたりの最大シャード数の制限を設定できるようになっていますが、ノードあたりのシャード数は 1,000 以下にすることをお勧めします。シャード数の選択を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes have more than 1000 shards. To ensure
 optimum performance and stability of your
```

```
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

ディスク容量の不足

OpenSearch クラスター内の 1 つ以上のノードの使用可能なストレージ容量が 25% 未満、または 25 GB 未満になると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Space",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes in your cluster has less than 25% of storage
 space or less than 25GB.
                  Your cluster will be blocked for writes at 20% or 20GB. Please refer
 to the documentation for more information - https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/handling-errors.html#troubleshooting-cluster-block"
  }
}
```

低ディスクウォーターマーク超過

OpenSearch クラスター内のすべてのノードの使用可能なストレージ容量が 10% 未満、または 10 GB 未満になると、サービスからこのイベントが送信されます。すべてのノードが低ディスクウォーターマークを超過すると、新しいインデックスは黄色のクラスターになり、すべてのノードが高ディスクウォーターマークを下回ると、赤色のクラスターになります。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Watermark Breach",
     "status":"Warning",
     "severity":"Medium",
     "description":"Low Disk Watermark threshold is about to be breached. Once the
 threshold is breached, new index creation will be blocked on all
                    nodes to prevent the cluster status from turning red. Please
 increase disk size to suit your storage needs. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#troubleshooting-cluster-block".
  }
}
```

EBS バーストバランスが 70% 未満

OpenSearch 1 つ以上のデータノードの EBS バースト残高が 70% を下回ると、サービスはこのイベントを送信します。EBS バーストバランスが枯渇すると、クラスターが広範囲にわたって使用できなくなり、I/O リクエストがスロットリングされ、インデックス作成や検索リクエストのレイテンシーが高くなり、タイムアウトが発生する可能性があります。この問題を修正するステップについては、「the section called “低 EBS バーストバランス”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
```

```
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"Medium",
     "description":"EBS burst balance on one or more data nodes is below 70%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue."
  }
}
```

EBS バーストバランスが 20% 未満

OpenSearch 1 つ以上のデータノードの EBS バースト残高が 20% を下回ると、サービスはこのイベントを送信します。EBS バーストバランスが枯渇すると、クラスターが広範囲にわたって使用できなくなり、I/O リクエストがスロットリングされ、インデックス作成や検索リクエストのレイテンシーが高くなり、タイムアウトが発生する可能性があります。この問題を修正するステップについては、「the section called “低 EBS バーストバランス”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"High",
     "description":"EBS burst balance on one or more data nodes is below 20%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
```

```
                    to fix this issue.
  }
}
```

ディスクスループットのスロットリング

OpenSearch EBS ボリュームまたは EC2 インスタンスのスループット制限により、ドメインへの読み取り/書き込みリクエストが制限されている場合に、サービスからこのイベントが送信されます。この通知を受け取った場合は、 AWS 推奨されるベストプラクティスに従ってボリュームまたはインスタンスをスケールアップすることを検討してください。ボリュームタイプが gp2 の場合は、ボリュームサイズを大きくします。ボリュームタイプが gp3 の場合は、より多くのスループットをプロビジョニングします。また、インスタンスベースと最大 EBS スループットがプロビジョニングされたボリュームスループット以上であることを確認し、それに応じてスケールアップすることもできます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Disk Throughput Throttle",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is experiencing throttling due to instance or volume
 throughput limitations.
                    Please consider scaling your domain to suit your throughput needs.
 In July 2023, we improved
                    the accuracy of throughput throttle calculation by replacing ‘Max
 volume throughput’ with
                    ‘Provisioned volume throughput’. Please refer to the documentation
 for more information."
  }
}
```

大きなシャードサイズ

OpenSearch クラスター内の 1 つ以上のシャードが 50 GiB または 65 GiB を超えたときに、サービスからこのイベントが送信されます。クラスターのパフォーマンスと安定性を最適化するには、シャードサイズを小さくしてください。

詳細については、シャーディングのベストプラクティスをご覧ください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Large Shard Size",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more shards are larger than 65GiB. To ensure optimum cluster
 performance and stability, reduce shard sizes.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-large-shard-
size."
  }
}
```

高い JVM 使用率

OpenSearch `JVMMemoryPressure`ドメインのメトリックスが 80% を超えると、サービスからこのイベントが送信されます。30 分間で 92% を超えると、クラスターに対するすべての書き込みオペレーションがブロックされます。クラスターの最適な安定性を確保するには、クラスターへのトラフィックを減らすか、またはドメインをスケールしてワークロードに十分なメモリを提供します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High JVM Usage",
     "status":"Warning",
     "severity":"High",
     "description":"JVM memory pressure has exceeded 80%. If it exceeds 92% for 30
 minutes, all write operations to your cluster
                    will be blocked. To ensure optimum cluster stability, reduce
 traffic to the cluster or use larger instance types.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-high-jvm."
  }
}
```

GC が不足している

OpenSearch JVM の最大値が 70% を超え、最大値と最小値の差が 30% 未満の場合、サービスはこのイベントを送信します。これは、JVM がワークロード用のガベージコレクションサイクル中に十分なメモリを再利用できないことを示唆している可能性があります。これにより、レスポンスがますます遅くなり、レイテンシーが高くなる可能性があります。また場合によっては、ヘルスチェックのタイムアウトによりノードがドロップすることさえあります。クラスターの最適な安定性を確保するには、クラスターへのトラフィックを減らすか、またはドメインをスケールしてワークロードに十分なメモリを提供します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
```

```
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Insufficient GC",
     "status":"Warning",
     "severity":"Medium",
     "description":"Maximum JVM is above 70% and JVM range is less than 30%. This may
 indicate insufficient garbage collection for your workload.
                   For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-insufficient-
gc."
  }
}
```

カスタムインデックスルーティングの警告

OpenSearch ドメインが処理中であり、カスタム index.routing.allocation 設定のインデックスが含まれていると、サービスがこのイベントを送信します。これにより、Blue-Green デプロイメントが停止する可能性があります。設定が適切に適用されていることを確認します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Custom Index Routing Warning",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is in processing state and contains indice(s) with
 custom index.routing.allocation
```

```
                    settings which can cause blue-green deployments to get stuck.
 Verify settings are applied properly.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-index-routing."
  }
}
```

シャードロックの失敗

OpenSearch とのシャードが割り当てられていないためにドメインが正常でない場合、サービスはこのイベントを送信します。[ShardLockObtainFailedException]詳細については、「Amazon OpenSearch Service のメモリ内シャードロック例外を解決するにはどうすればよいですか?」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Failed Shard Lock",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is unhealthy due to unassigned shards with
 [ShardLockObtainFailedException]. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html#monitoring-events-failed-shard-lock."
}
```

VPC エンドポイントイベント

OpenSearch EventBridge AWS PrivateLink サービスは特定のイベントを関連するインターフェイスエンドポイントに送信します。

VPC エンドポイントが作成できない

OpenSearch リクエストされた VPC エンドポイントを作成できない場合、サービスはこのイベントを送信します。このエラーは、1 つのリージョンで許可される VPC エンドポイントの数の制限に到達したために発生することがあります。このエラーは、指定されたサブネットまたはセキュリティグループが存在しない場合にも表示されます。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Create Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to create VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: You've reached the limit on the
                     number of VPC endpoints that you can create in the AWS Region."
   }
}
```

VPC エンドポイントが更新できない

OpenSearch リクエストされた VPC エンドポイントを削除できない場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to update VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: <failure message>."
    }
}
```

VPC エンドポイントが削除できない

OpenSearch リクエストされた VPC エンドポイントを削除できない場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Delete Validation",
```

```
      "status":"Failed",
      "severity":"High",
      "description":"Unable to delete VPC endpoint aos-0d4c74c0342343 for domain
                    arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: Specified subnet doesn't exist."
   }
}
```

ノードの廃止イベント

OpenSearch サービスは、 EventBridge 以下のノードリタイアイベントのいずれかが発生したときにイベントを送信します。

ノードの廃止をスケジュールしました

OpenSearch ノードのリタイアが予定されている場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Scheduled",
    "severity": "Medium",
    "description": "An automated action to retire and replace a node has been scheduled
 on your domain.
                    The node will be replaced in the next off-peak window. For more
 information, see
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html."
  }
}
```

ノードの廃止が完了しました

OpenSearch ノードのリタイアが完了すると、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Completed",
    "severity": "Medium",
    "description": "The node has been retired and replaced with a new node."
  }
}
```

ノードの廃止に失敗しました

OpenSearch ノードのリタイアが失敗すると、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
```

```
    "event": "Node Retirement Notification",
    "status": "Failed",
    "severity": "Medium",
    "description": "Node retirement failed. No actions are required from your end. We
 will automatically
                   retry replacing the node."
  }
}
```

デグレードノードのリタイアイベント

OpenSearch ノード上のハードウェアの劣化によりノードの交換が必要になったときに、サービスがこれらのイベントを送信します。

デグレードノードのリタイア通知

OpenSearch デグレードしたノードをリタイアして交換する自動アクションがドメインでスケジュールされている場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"db233454-aad1-7676-3b15-10a84b052baa",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T08:16:06Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"An automated action to retire and replace a node has
 been scheduled on your domain. For more information, please see https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/monitoring-events.html.",
      "event":"Degraded Node Retirement Notification",
      "status":"Scheduled"
   }
```

```
}
```

デグレードノードのリタイアが完了しました。

OpenSearch デグレードノードがリタイアされ、新しいノードに置き換えられたときに、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"7444215c-90f9-a52d-bcda-e85973a9a762",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2024-01-11T10:20:30Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"The node has been retired and replaced with a new node.",
      "event":"Degraded Node Retirement Notification",
      "status":"Completed"
   }
}
```

デグレードノードのリタイアは失敗しました。

OpenSearch デグレードノードのリタイアに失敗した場合、サービスはこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"c328e9bb-93b9-c0b2-b17a-df527fdf96b6",
   "detail-type":"Amazon OpenSearch Service Notification",
   "source":"aws.es",
```

```
   "account":"123456789012",
   "time":"2024-01-11T08:31:38Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-node-replacement"
   ],
   "detail":{
      "severity":"Medium",
      "description":"Node retirement failed. No actions are required from your end. We
 will automatically re-try replacing the node.",
      "event":"Degraded Node Retirement Notification",
      "status":"Failed"
   }
}
```

ドメインエラーイベント

OpenSearch EventBridge 以下のドメインエラーのいずれかが発生すると、サービスがイベントを送信します。

ドメイン更新の検証エラー

OpenSearch ドメインの設定を更新または変更しようとしたときに 1 つ以上の検証エラーが発生すると、サービスはこのイベントを送信します。これらの障害を解決するためのステップについては、「the section called “検証エラーのトラブルシューティング”」を参照してください。

例

このタイプのイベントの例を以下に示します。

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Domain Update Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
```

```
      "event":"Domain Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to perform updates to your domain due to the following
 validation failures: <failures>
                    Please see the documentation for more information https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/managedomains-
configuration-changes.html#validation"
   }
}
```

KMS キーにアクセスできない

OpenSearch AWS KMS サービスはキーにアクセスできない場合にこのイベントを送信します。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Domain Error Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"KMS Key Inaccessible",
     "status":"Error",
     "severity":"High",
     "description":"The KMS key associated with this domain is inaccessible. You are at
 risk of losing access to your domain.
                    For more information, please refer to https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

ドメインの分離

OpenSearch ドメインが隔離され、ネットワークからアクセスできないためにリクエストを受信、読み取り、または書き込みできなくなったときに、サービスからこのイベントが送信されます。

例

このタイプのイベントの例を以下に示します。

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2023-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Domain Isolation Notification",
     "status":"Error",
     "severity":"High",
     "description":"Your OpenSearch Service domain has been isolated. An isolated
 domain is unreachable by network and cannot receive, read, or write requests. For more
 information and assistance, please contact AWS Support at https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

チュートリアル:Amazon OpenSearch EventBridge サービスイベントのリスニング

このチュートリアルでは、Amazon OpenSearch Service AWS Lambda CloudWatch イベントを受信してログログストリームに書き込む簡単な関数を設定します。

前提条件

このチュートリアルでは、既存の OpenSearch Service ドメインがあることを前提としています。ドメインをまだ作成していない場合は、1 つ作成するために「ドメインの作成と管理」の手順に従います。

ステップ 1: Lambda 関数を作成する

この手順では、 OpenSearch サービスイベントメッセージのターゲットとして機能する単純な Lambda 関数を作成します。

ターゲットの Lambda 関数を作成するには

1. https://console.aws.amazon.com/lambda/ AWS Lambda でコンソールを開きます。
2. [Lambda 関数の作成]、[一から作成] の順に選択します。
3. [関数名] では、event-handler と入力します。
4. [ランタイム] では、[Python 3.8] を選択します。
5. [Create function] を選択します。
6. [Function code] (関数コード) セクションで、以下の例に一致するようにサンプルコードを編集します。

```
import json

def lambda_handler(event, context):
    if event["source"] != "aws.es":
        raise ValueError("Function only supports input from events with a source
 type of: aws.es")

    print(json.dumps(event))
```

これは、 OpenSearch サービスによって送信されたイベントを出力する単純な Python 3.8 関数です。すべてが正しく設定されていれば、このチュートリアルの最後に、この Lambda CloudWatch 関数に関連付けられているログストリームにイベントの詳細が表示されます。

7. [デプロイ] を選択します。

ステップ 2: イベントルールを登録する

このステップでは、 EventBridge サービスドメインからイベントをキャプチャするルールを作成します。 OpenSearch このルールでは、それが定義されているアカウント内のすべてのイベントがキャプチャされます。イベントメッセージ自体に、イベントソースに関する情報 (イベントソースの送信元ドメインの情報など) が含まれています。この情報を使用して、プログラムでイベントをフィルタリングおよびソートできます。

ルールを作成するには EventBridge

1. https://console.aws.amazon.com/events/ EventBridge でコンソールを開きます。
2. [ルールの作成] を選択します。
3. ルールに event-rule と名前を付けます。
4. [次へ] をクリックします。

5. イベントパターンには、[AWS サービス]、[Amazon OpenSearch サービス]、[すべてのイベント] を選択します。このパターンは、 OpenSearch OpenSearch すべてのサービスドメインとすべてのサービスイベントに適用されます。または、より具体的なパターンを作成し、一部の結果をフィルターで除外することもできます。
6. [次へ] を選択します。
7. ターゲットに [Lambda 関数] を選択します。機能ドロップダウンで [イベントハンドラ] を選択します。
8. [次へ] を選択します。
9. タグをスキップして [次へ] をクリックします。
10. 設定を確認して、[ルールの作成] を選択します。

ステップ 3: 設定をテストする

OpenSearch 次にサービスコンソールの Notifications セクションで通知を受け取ったときに、すべてが正しく設定されていれば、Lambda 関数がトリガーされ、 CloudWatch その関数の口グログストリームにイベントデータが書き込まれます。

設定をテストするには

1. https://console.aws.amazon.com/cloudwatch/ CloudWatch でコンソールを開きます。
2. ナビゲーションペインで、[口グ] を選択して Lambda 関数 (/aws/lambda/event-handler など) のロググループを選択します。
3. イベントデータを表示するログストリームを選択します。

チュートリアル: 利用可能なソフトウェア更新に関する Amazon SNS アラートの送信

このチュートリアルでは、Amazon Service で利用可能なサービスソフトウェアアップデートの通知をキャプチャし、 Amazon Simple Notification Service (Amazon OpenSearch SNS) 経由でメール通知を送信する Amazon EventBridge イベントルールを設定します。

前提条件

このチュートリアルでは、 OpenSearch 既存のサービスドメインがあることを前提としています。ドメインをまだ作成していない場合は、1 つ作成するために「ドメインの作成と管理」の手順に従います。

ステップ 1: Amazon SNS トピックを作成してサブスクライブする

新しいイベントルールのイベントターゲットとして使用する Amazon SNS トピックを設定します。

Amazon SNS ターゲットを作成するには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. [トピック] および [トピックの作成] を選択します。
3. ジョブタイプでは、[標準] を選択し、ジョブに software-update と名前を付けます。
4. [Create topic] (トピックの作成) を選択します。
5. トピックが作成されたら、[サブスクリプションの作成] を選択します。
6. [プロトコル] で [E メール] を選択します。[Endpoint (エンドポイント)] では、現在アクセスできるメールアドレスを入力し、[Create subscription (サブスクリプションの作成)] を選択します。
7. メールアカウントを確認し、サブスクリプションの確認メールメッセージが届くのを待ちます。確認メールが届いたら、[サブスクリプションの確認] を選択します。

ステップ 2: イベントルールを登録する

次に、サービスソフトウェア更新イベントのみをキャプチャするイベントルールを登録します。

イベントルールを作成するには

1. https://console.aws.amazon.com/events/ EventBridge でコンソールを開きます。
2. [ルールの作成] を選択します。
3. ルールに softwareupdate-rule と名前を付けます。
4. [次へ] をクリックします。
5. イベントパターンには、[サービス]、[Amazon AWS OpenSearch サービス]、[Amazon OpenSearch サービスソフトウェア更新通知] を選択します。このパターンは、Service OpenSearch からのすべてのサービスソフトウェア更新イベントと一致します。イベントパターンの詳細については、Amazon EventBridgeユーザーガイドの「Amazon EventBridge イベントパターン」を参照してください。
6. 特定の重大度のみをフィルターできます (オプション)。各イベントの重大度については、「the section called “サービスソフトウェア更新イベント”」を参照してください。
7. [次へ] をクリックします。
8. ターゲットに [SNS トピック] を選択し、[software-update] (ソフトウェア更新) を選択します。
9. [次へ] をクリックします。

10. タグをスキップして [次へ] を選択します。
11. ルールの設定を確認して、[ルールの作成] を選択します。

次回 OpenSearch Service から利用可能なサービスソフトウェアの更新に関する通知を受け取ったときに、すべてが正しく設定されていれば、Amazon SNS から更新に関するメールアラートが送信されます。

AWS CloudTrail での Amazon OpenSearch Service API 呼び出しのモニタリング

Amazon OpenSearch Service は、ユーザー、ロール、または OpenSearch Service の AWS サービスによって取られたアクションのレコードを提供するサービスである AWS CloudTrail と統合されます。CloudTrail は、OpenSearch Service のすべての設定 API 呼び出しをイベントとしてキャプチャします。

> **Note**
>
> CloudTrail は、`CreateDomain` や `GetUpgradeStatus` などの、Configuration API への呼び出しのみをキャプチャします。CloudTrail は、`_search` や `_bulk` などの、OpenSearch APIへの呼び出しをキャプチャしません。これらの呼び出しについては、「the section called "監査ログのモニタリング"」を参照してください。

キャプチャされる呼び出しには、OpenSearch Service コンソール、AWS CLI、または AWS SDK からの呼び出しが含まれます。証跡を作成する場合は、OpenSearch Service のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信を有効にすることができます 証跡を設定しない場合でも、CloudTrail コンソールの [Event history (イベント履歴)] で最新のイベントを表示できます。CloudTrail で収集された情報を使用して、OpenSearch Service 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。

CloudTrail の詳細については、AWS CloudTrail ユーザーガイドを参照してください。

CloudTrail での Amazon OpenSearch Service 情報

CloudTrail は、アカウント作成時に AWS アカウント で有効になります。OpenSearch Service でアクティビティが発生すると、そのアクティビティは [Event history] (イベント履歴) で AWS のその他のサービスのイベントとともに CloudTrail イベントに記録されます。AWS アカウント 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、「CloudTrail Event 履歴でのイベントの表示」を参照してください。

OpenSearch Service のイベントなど、AWS アカウント 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。追跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての 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 ログで収集し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ためにその他の AWS のサービスを設定できます。詳細については、以下を参照してください。

- AWS アカウント の追跡の作成
- AWS サービスと CloudTrail ログの統合
- CloudTrail の Amazon SNS 通知の設定
- CloudTrail ログファイルを複数のリージョンから受け取る、複数のアカウントから CloudTrail ログファイルを受け取る

すべての OpenSearch Service 設定 API アクションは CloudTrail によってログに記録され、「Amazon OpenSearch Service API リファレンス」に記載されています。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。アイデンティティ情報から以下を判断することができます。

- リクエストが、ルートと AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか
- リクエストの送信に使用された一時的なセキュリティ認証情報に、ロールとフェデレーティッドユーザーのどちらが使用されたか
- リクエストが、別の AWS のサービスによって送信されたかどうか

詳細については、CloudTrail userIdentity エレメントを参照してください。

Amazon OpenSearch Service のログファイルエントリを理解する

[トレイル] は、指定した Simple Storage Service (Amazon S3) バケットにイベントをログファイルとして配信するように構成できます。CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。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。

`CreateDomain` オペレーションを示す CloudTrail ログエントリの例は、次のとおりです。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/test-user",
    "accountId": "123456789012",
    "accessKeyId": "access-key",
    "userName": "test-user",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T21:59:11Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-08-21T22:00:05Z",
  "eventSource": "es.amazonaws.com",
  "eventName": "CreateDomain",
  "awsRegion": "us-west-1",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "engineVersion": "OpenSearch_1.0",
    "clusterConfig": {
      "instanceType": "m4.large.search",
      "instanceCount": 1
    },
    "snapshotOptions": {
      "automatedSnapshotStartHour": 0
    },
    "domainName": "test-domain",
    "encryptionAtRestOptions": {},
    "eBSOptions": {
      "eBSEnabled": true,
      "volumeSize": 10,
      "volumeType": "gp2"
    },
```

```
    "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}",
    "advancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    }
  },
  "responseElements": {
    "domainStatus": {
      "created": true,
      "clusterConfig": {
        "zoneAwarenessEnabled": false,
        "instanceType": "m4.large.search",
        "dedicatedMasterEnabled": false,
        "instanceCount": 1
      },
      "cognitoOptions": {
        "enabled": false
      },
      "encryptionAtRestOptions": {
        "enabled": false
      },
      "advancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
      },
      "upgradeProcessing": false,
      "snapshotOptions": {
        "automatedSnapshotStartHour": 0
      },
      "eBSOptions": {
        "eBSEnabled": true,
        "volumeSize": 10,
        "volumeType": "gp2"
      },
      "engineVersion": "OpenSearch_1.0",
      "processing": true,
      "aRN": "arn:aws:es:us-west-1:123456789012:domain/test-domain",
      "domainId": "123456789012/test-domain",
      "deleted": false,
      "domainName": "test-domain",
      "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",
\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}"
    }
```

```
  },
  "requestID": "12345678-1234-1234-1234-987654321098",
  "eventID": "87654321-4321-4321-4321-987654321098",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Amazon OpenSearch Service のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ — クラウドで AWS AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。 AWS また、 では、安全に使用できるサービスも提供しています。AWS コンプライアンスプログラムの一環として、サードパーティーの監査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ています。Amazon OpenSearch Service 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AWS プログラムによる対象範囲内のサービス」を参照してください。
- クラウドのセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、 OpenSearch サービスを使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するように OpenSearch サービスを設定する方法を示します。また、サービスリソースのモニタリングや保護に役立つ他の OpenSearch 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 OpenSearch Service でのデータ保護
- Amazon OpenSearch Service での Identity and Access Management
- サービス間での不分別な代理処理の防止
- Amazon サービスのきめ細かいアクセス制御 OpenSearch
- Amazon OpenSearch Service のコンプライアンス検証
- Amazon OpenSearch の耐障害性
- Amazon OpenSearch Service の JWT 認証と認可
- Amazon OpenSearch Service のインフラストラクチャセキュリティ
- OpenSearch Dashboards の SAML 認証

- OpenSearch Dashboards の Amazon Cognito 認証の設定
- Amazon OpenSearch Service のサービスにリンクされたロールの使用

Amazon OpenSearch Service でのデータ保護

責任 AWS 共有モデル、Amazon OpenSearch Service 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AWS 責任共有モデルおよび GDPRのブログ記事を参照してください。

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

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 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 OpenSearch サービスまたは他の 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。

Amazon OpenSearch Service の保管中のデータの暗号化

OpenSearch サービスドメインは、データへの不正アクセスを防ぐセキュリティ機能である保管中のデータの暗号化を提供します。この機能は AWS Key Management Service （ AWS KMS) を使用して暗号化キーを保存および管理し、256 ビットキー (AES-256) を使用して暗号化を実行する Advanced Encryption Standard アルゴリズムを使用します。有効にすると、この機能によりドメインの次の要素が暗号化されます。

- すべてのインデックス (UltraWarm ストレージ内のインデックスを含む)
- OpenSearch ログ
- スワップファイル
- アプリケーションディレクトリのその他すべてのデータ
- 自動スナップショット

以下は、保管中のデータの暗号化を有効にするときに暗号化されませんが、追加のステップを行って保護することができます。

- 手動スナップショット: 現在、 AWS KMS キーを使用して手動スナップショットを暗号化することはできません。ただし、S3 で管理されたキーまたは KMS キーによるサーバー側の暗号化を使用してスナップショットリポジトリとして使用しているバケットを暗号化できます。手順については、「the section called “手動スナップショットレポジトリの登録”」を参照してください。
- スローログとエラーログ: ログを発行して暗号化する場合は、 OpenSearch サービスドメインと同じ AWS KMS キーを使用して CloudWatch ログロググループを暗号化できます。詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「 を使用してログのログデータを暗号化 AWS KMSする」を参照してください。 CloudWatch

> **Note**
>
> ドメインで UltraWarm または コールドストレージが有効になっている場合、既存のドメインで保管時の暗号化を有効にすることはできません。まず、 UltraWarm またはコールドストレージを無効にし、保管時の暗号化を有効にしてから、 UltraWarm またはコールドストレージを再度有効にする必要があります。インデックスを UltraWarm またはコールドストレージに保持する場合は、 UltraWarm またはコールドストレージを無効にする前に、インデックスをホットストレージに移動する必要があります。

OpenSearch サービスは、非対称暗号化 KMS キーではなく、対称暗号化 KMS キーのみをサポートします。KMS マスターキーを作成する方法については、AWS Key Management Service デベロッパーガイドの「キーの作成」を参照してください。

保管時の暗号化が有効になっているかどうかにかかわらず、すべてのドメインは AES-256 と OpenSearch サービスマネージドキーを使用してカスタムパッケージを自動的に暗号化します。

アクセス許可

OpenSearch サービスコンソールを使用して保管中のデータの暗号化を設定するには、次のアイデンティティベースのポリシーなど AWS KMS、 への読み取りアクセス許可が必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS 所有キー以外のキーを使用する場合は、キーの許可を作成するアクセス許可も必要です。通常、これらの許可は、キーの作成時に指定するリソースベースのポリシーの形式になります。

キーを OpenSearch Service に限定する場合は、kms:ViaService 条件をそのキーポリシーに追加できます。

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

詳細については、「 AWS Key Management Service デベロッパーガイド」のAWS 「KMS でのキーポリシーの使用」を参照してください。

保管中のデータの暗号化の有効化

新しいドメインで保管中のデータを暗号化するには、 OpenSearch または Elasticsearch 5.1 以降が必要です。既存のドメインで有効にするには、 OpenSearch または Elasticsearch 6.7 以降が必要です。

保管中のデータの暗号化を有効にするには (コンソール)

1. AWS コンソールでドメインを開き、「アクション」と「セキュリティ設定の編集」を選択します。
2. 暗号化した状態で、保管中のデータの暗号化を有効にするを選択します。
3. 使用する AWS KMS キーを選択し、変更の保存 を選択します。

設定 API を使って、暗号化を有効にすることもできます。次のリクエストは、既存ドメインで保存中のデータの暗号化を有効にします。

```
{
   "ClusterConfig":{
      "EncryptionAtRestOptions":{
         "Enabled": true,
         "KmsKeyId":"arn:aws:kms:us-east-1:123456789012:alias/my-key"
      }
   }
}
```

KMS キーを無効化または削除

ドメインの暗号化に使用したキーを無効化または削除すると、ドメインにアクセスできなくなります。 OpenSearch サービスは、KMS キーにアクセスできないことを通知する通知を送信します。すぐにキーを再度有効にして、ドメインにアクセスします。

キーが削除された場合、 OpenSearch サービスチームはデータの復旧をサポートできません。 は、少なくとも 7 日間の待機期間後にのみキー AWS KMS を削除します。キーの削除が保留中の場合は、削除をキャンセルするか、手動スナップショットを取って、ドメインのデータの損失を防ぎます。

保管中のデータの暗号化の無効化

保管中のデータを暗号化するようにドメインを設定した後、設定を無効にすることはできません。代わりに、既存のドメインの手動スナップショットを作成し、別のドメインを作成してからデータを移行して、古いドメインを削除することができます。

保管中のデータを暗号化するドメインのモニタリング

保管中のデータを暗号化するドメインには、`KMSKeyError` と `KMSKeyInaccessible` の 2 つの追加のメトリクスがあります。これらのメトリクスは、ドメインの暗号化キーに問題が発生した場合にのみ表示されます。これらのメトリクスの詳細については、「the section called “クラスターメトリクス”」を参照してください。これらは、 OpenSearch サービスコンソールまたは Amazon CloudWatch コンソールを使用して表示できます。

> **Tip**
>
> 各メトリクスはドメインの重要な問題であるため、両方の CloudWatch アラームを作成することをお勧めします。詳細については、「the section called “推奨 CloudWatch アラーム”」を参照してください。

その他の考慮事項

- 自動キーローテーションでは AWS KMS キーのプロパティが保持されるため、ローテーションは OpenSearch データにアクセスする機能には影響しません。暗号化された OpenSearch サービスドメインは、手動キーローテーションをサポートしていません。これには、新しいキーの作成と古いキーへの参照の更新が含まれます。詳細については、AWS Key Management Service デベロッパーガイドの「キーローテーション」を参照してください。
- 特定のインスタンスタイプは、保管中のデータの暗号化をサポートしていません。詳細については、「the section called “サポートされるインスタンスタイプ”」を参照してください。
- 保管中のデータを暗号化するドメインでは、自動スナップショット用に別のリポジトリ名を使用します。詳細については、「the section called “スナップショットの復元”」を参照してください。
- 静止時に暗号化を有効にすることを強くお勧めしますが、CPU オーバーヘッドが増加し、数ミリ秒の遅延が発生する可能性があります。ただし、ほとんどのユースケースはこれらの違いに敏感ではなく、影響の大きさは、クラスター、クライアント、および使用プロファイルの構成によって異なります。

Amazon OpenSearch Service の N ode-to-node 暗号化

N ode-to-node 暗号化は、Amazon OpenSearch Service のデフォルト機能に加えてセキュリティレイヤーを追加します。

各 OpenSearch サービスドメインは、ドメインが VPC アクセスを使用するかどうかにかかわらず、独自の専用 VPC 内にあります。このアーキテクチャにより、潜在的な攻撃者が OpenSearch ノード間のトラフィックを傍受するのを防ぎ、クラスターを安全に保つことができます。ただし、デフォルトでは、VPC 内のトラフィックは暗号化されません。N ode-to-node 暗号化は、VPC 内のすべての通信に対して TLS 1.2 暗号化を有効にします。

HTTPS 経由で OpenSearch Service にデータを送信する場合、 node-to-node 暗号化は、データがクラスター全体に OpenSearch 配信 (および再配布) されるときに、データが暗号化されたままになるようにします。データが HTTP 経由で暗号化されずに到着した場合、 OpenSearch サービスはクラスターに到達した後にデータを暗号化します。ドメインへのすべてのトラフィックは、コンソール AWS CLI、または設定 API を使用して HTTPS 経由で到着するように要求できます。

きめ細かなアクセスコントロール を有効にする場合は、N 暗号化ode-to-node が必要です。

node-to-node 暗号化の有効化

新しいドメインでの N ode-to-node 暗号化には OpenSearch、 の任意のバージョン、または Elasticsearch 6.0 以降が必要です。既存のドメインで node-to-node 暗号化を有効にするには OpenSearch、 の任意のバージョン、または Elasticsearch 6.7 以降が必要です。 AWS コンソールで既存のドメインを選択し、[アクション] から [セキュリティ設定の編集] を選択します。

または、 AWS CLI または 設定 API を使用することもできます。詳細については、AWS CLI 「 コマンドリファレンス」およびOpenSearch 「サービス API リファレンス」を参照してください。

node-to-node 暗号化の無効化

node-to-node 暗号化を使用するようにドメインを設定した後は、設定を無効にすることはできません。代わりに、暗号化されたドメインの手動スナップショットを作成し、別のドメインを作成してからデータを移行して、古いドメインを削除することができます。

Amazon OpenSearch Service での Identity and Access Management

Amazon OpenSearch Service には、ドメインへのアクセスを制御する方法がいくつか用意されています。このトピックでは、さまざまなポリシータイプ、それぞれがやり取りする方法、および独自のカスタムポリシーを作成する方法を説明します。

> **Important**
>
> VPC サポートでは、 OpenSearch サービスアクセスコントロールに関する追加の考慮事項がいくつか導入されています。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

ポリシーのタイプ

OpenSearch サービスは、次の 3 種類のアクセスポリシーをサポートしています。

- the section called “リソースベースのポリシー”
- the section called “アイデンティティベースのポリシー”
- the section called “IP ベースのポリシー”

リソースベースのポリシー

ドメインを作成するときに、ドメインアクセスポリシーと呼ばれる場合があるリソースベースのポリシーを追加します。これらのポリシーは、ドメインのサブリソースでプリンシパルが実行するアクションを指定します (cross-cluster 検索を除く)。サブリソースには、 OpenSearch インデックスと APIs。Principal 要素は、アクセスを許可するアカウント、ユーザー、またはロールを指定します。Resource 要素は、これらのプリンシパルがアクセスできるサブリソースを指定します。

例えば、次のリソースベースのポリシーでは、`test-domain` のサブリソースへのフルアクセス (`es:*`) が `test-user` に付与されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:*"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

このポリシーには、2 つの重要な考慮事項が適用されます。

- これらの権限はこのドメインのみに適用されます。他のドメインで同様のポリシーを作成しない限り、test-user は test-domain にしかアクセスできません。
- Resource 要素の末尾の /* は重要であり、リソースベースのポリシーがドメイン自体ではなく、ドメインのサブリソースにのみ適用されることを示します。リソースベースのポリシーでは、es:* アクションは es:ESHttp* と同等です。

  例えば、test-user はインデックス (GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index) に対してリクエストを行うことができますが、ドメインの設定を更新できません (POST https://es.us-west-1.amazonaws.com/2021-01-01/opensearch/domain/test-domain/config)。2 つのエンドポイント間の違いに注目してください。設定 API にアクセスするには ID ベースのポリシーが必要です。

ワイルドカードを追加することで、インデックス名の一部を指定できます。次の例では、commerce で始まるすべてのインデックスを特定しています。

```
arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce*
```

この場合、ワイルドカードの意味は、commerce で始まる名前の test-domain のインデックスに test-user がリクエストを送信できるということです。

test-user をさらに制限するには、次のポリシーを適用することができます。

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-data/
_search"
    }
  ]
}
```

これにより、test-user は commerce-data インデックスに対する検索で、1 つのオペレーションのみを実行できます。ドメインのその他すべてのインデックスはアクセス不可となり、、test-user は、es:ESHttpPut または es:ESHttpPost アクションを使用する許可なしにドキュメントを追加あるいは変更できなくなります。

次に、パワーユーザーのロールを設定することができます。このポリシーは、power-user-role にインデックスのすべての URI に対する HTTP GET メソッドおよび PUT メソッドへのアクセスを付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-data/
*"
    }
```

```
  ]
}
```

ドメインが VPC 内にあるか、きめ細かなアクセスコントロールを使用している場合は、オープンドメインアクセスポリシーを使用できます。それ以外の場合、ドメインアクセスポリシーには、プリンシパルまたは IP アドレスによる制限が含まれている必要があります。

使用できるすべてのアクションの詳細については、「the section called “ポリシーエレメントのリファレンス”」を参照してください。データをより細かく制御するには、きめ細かなアクセスコントロールとともにオープンドメインアクセスポリシーを使用します。

アイデンティティベースのポリシー

各 OpenSearch サービスドメインの一部であるリソースベースのポリシーとは異なり、 AWS Identity and Access Management (IAM) サービスを使用して、アイデンティティベースのポリシーをユーザーまたはロールにアタッチします。リソースベースのポリシーと同様に、アイデンティティベースのポリシーは、サービスに誰がアクセスできるか、どのアクションキーを実行できるか、そして該当する場合には、これらのアクションを実行できるリソースを指定します。

これらを実行する実際の義務はないため、アイデンティティベースのポリシーはより一般的になる傾向があります。これにより、多くの場合、ユーザーが実行できる設定 API アクションのみ管理されます。これらのポリシーを設定したら、 OpenSearch サービスでリソースベースのポリシー (またはきめ細かなアクセスコントロール) を使用して、 OpenSearch インデックスと APIsへのアクセス権をユーザーに付与できます。

> **Note**
>
> AWS 管理`AmazonOpenSearchServiceReadOnlyAccess`ポリシーを持つユーザーは、コンソールでクラスターのヘルスステータスを表示できません。クラスターのヘルスステータス (およびその他の OpenSearch データ) を表示できるようにするには、アクセスポリシーに`es:ESHttpGet`アクションを追加し、アカウントまたはロールにアタッチします。

アイデンティティベースのポリシーはユーザーあるいはロール (プリンシパル) にアタッチされるため、JSON はプリンシパルを指定しません。次のポリシーは、`Describe` および `List` で始まるアクションへのアクセスを付与します。このアクションの組み合わせはドメインの設定への読み取り専用のアクセスを提供しますが、ドメイン自体に保存されたデータへのアクセスは提供しません。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:Describe*",
        "es:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

管理者は、 OpenSearch サービスおよびすべてのドメインに保存されているすべてのデータへのフルアクセスを持つ場合があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

ID ベースのポリシーでは、タグを使用して設定 API へのアクセスを制御できます。例えば、次のポリシーでは、ドメインに team:devops タグがある場合、アタッチされたプリンシパルがドメインの設定を表示および更新できるようにします。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:UpdateDomainConfig",
      "es:DescribeDomain",
      "es:DescribeDomainConfig"
    ],
    "Effect": "Allow",
```

```
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/team": [
          "devops"
        ]
      }
    }
  }]
}
```

タグを使用して OpenSearch API へのアクセスを制御することもできます。 OpenSearch API のタグベースのポリシーは、HTTP メソッドにのみ適用されます。例えば、次のポリシーでは、ドメインに `environment:production` タグがある場合、アタッチされたプリンシパルが GET および PUT リクエストを OpenSearch API に送信できます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:ESHttpGet",
      "es:ESHttpPut"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

OpenSearch API をよりきめ細かく制御するには、きめ細かなアクセスコントロール の使用を検討してください。

> **Note**
>
> タグベースのポリシーに 1 つ以上の OpenSearch APIs を追加した後、ドメインに変更を有効にするには、1 つのタグオペレーション (タグの追加、削除、変更など) を実行する必要が

あります。タグベースのポリシーに OpenSearch API オペレーションを含めるには、サービスソフトウェア R20211203 以降を使用している必要があります。

OpenSearch サービスは、 API ではなく、設定 API の RequestTagおよび TagKeys グローバル条件キーをサポートします OpenSearch 。これらの条件は、リクエスト内にタグを含む API 呼び出し (CreateDomain、AddTags、および RemoveTags など) にのみ適用されます。次のポリシーでは、アタッチされたプリンシパルがドメインを作成できるようにしますが、それは、team:it タグをリクエストに含めた場合のみです。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "es:CreateDomain",
      "es:AddTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/team": [
          "it"
        ]
      }
    }
  }
}
```

アクセス制御でタグを使用する方法とリソースベースのポリシーとアイデンティティベースのポリシーの違いについての詳細は、「IAM ユーザーガイド」を参照してください。

IP ベースのポリシー

IP ベースのポリシーは、1 つ以上の IP アドレスあるいは CIDR ブロックにドメインへのアクセスを制限します。技術的には、IP ベースのポリシーは異なるタイプのポリシーではありません。代わりに、これらのポリシーは、匿名のプリンシパルを指定し、特別な Condition 要素を含む、リソースベースのポリシーです。

IP ベースのポリシーの主な特徴は、 OpenSearch サービスドメインへの署名なしリクエストを許可することです。これにより、curl や OpenSearch Dashboards などのクライアントを使用したり、プ

ロキシサーバー経由でドメインにアクセスしたりできます。詳細については、「the section called “プロキシを使用して OpenSearch Dashboards から OpenSearch サービスにアクセスする”」を参照してください。

> **Note**
>
> ドメインで VPC アクセスを有効にすると、IP ベースのポリシーを設定することはできません。代わりに、どの IP アドレスがドメインにアクセスできるかを制御するセキュリティグループを使用できます。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

以下のポリシーは、指定された IP 範囲からのすべての HTTP リクエストが `test-domain` にアクセスする許可を付与します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

ドメインにパブリックエンドポイントがあり、きめ細かなアクセスコントロールを使用しない場合は、IAM プリンシパルと IP アドレスを組み合わせることをお勧めします。このポリシーでは、指定した IP 範囲からのリクエストの場合にのみ、`test-user` HTTP アクセスを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::987654321098:user/test-user"
      ]
    },
    "Action": [
      "es:ESHttp*"
    ],
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24"
        ]
      }
    },
    "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
  }]
}
```

OpenSearch サービスリクエストの作成と署名

完全にオープンなリソースベースのアクセスポリシーを設定しても、 OpenSearch サービス設定 API へのすべてのリクエストに署名する必要があります。ポリシーで IAM ロールまたはユーザーを指定する場合、 OpenSearch APIsへのリクエストも Signature Version AWS 4 を使用して署名する必要があります。署名メソッドは API によって異なります。

- OpenSearch サービス設定 API を呼び出すには、いずれかの AWS SDKs を使用することをお勧めします。SDK を使用するほうが、独自のリクエストを作成し署名するよりも、プロセスが簡素化し、大幅な時間の節約ができます。API エンドポイント設定には次の形式を使用します。

  ```
  es.region.amazonaws.com/2021-01-01/
  ```

  例えば、次のリクエストは、movies ドメインの設定を変更しますが、自分で署名する必要があります (お勧めしません)。

  ```
  POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/movies/config
  ```

```
{
  "ClusterConfig": {
    "InstanceType": "c5.xlarge.search"
  }
}
```

Boto 3 などのいずれかの SDK を使用する場合、SDK はリクエスト署名を自動的に処理します。

```
import boto3

client = boto3.client(es)
response = client.update_domain_config(
  DomainName='movies',
  ClusterConfig={
    'InstanceType': 'c5.xlarge.search'
  }
)
```

Java コードの例については、「the section called “AWS SDK の使用”」を参照してください。

- OpenSearch APIs を呼び出すには、独自のリクエストに署名する必要があります。 OpenSearch APIs形式を使用します。

```
domain-id.region.es.amazonaws.com
```

例えば、次のリクエストは、thor の movies インデックスを検索します。

```
GET https://my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
```

> **Note**
>
> このサービスは、署名バージョン 4 で署名された HTTP POST リクエストの URL で渡されるパラメータを無視します。

複数のポリシーが衝突する場合

複数のポリシーが同意しない、あるいはユーザーを明示的に指定しない場合、困難が生じます。IAM ユーザーガイドの「IAM の詳細を理解する」では、ポリシーの評価ロジックの適切な概要が説明されています。

- デフォルトでは、すべてのリクエストが拒否されます。
- 明示的な許可はこのデフォルトに優先します。
- 明示的な拒否はすべての許可に上書きされます。

例えば、リソースベースのポリシーによってドメインサブリソース (OpenSearch インデックスまたは API) へのアクセスが許可されているが、アイデンティティベースのポリシーによってアクセスが拒否された場合、アクセスは拒否されます。アイデンティティベースのポリシーとリソースベースのポリシーによってユーザーがアクセス許可を持つべきかを指定いない場合、アクセスは許可されます。ドメインサブリソースのすべての結果の概要における交差するポリシーの図を以下で参照してください。

	リソースベースのポリシーで許可	リソースベースのポリシーで拒否	リソースベースのポリシーで許可あるいは拒否されない
Allowed in identity-based policy	許可	拒否	許可
Denied in identity-based policy	拒否	拒否	拒否
Neither allowed nor denied in identity-based policy	許可	拒否	拒否

ポリシーエレメントのリファレンス

OpenSearch サービスは、IAM ポリシー要素リファレンス のほとんどのポリシー要素をサポートしますが、 は除きますNotPrincipal。次の表は、最も一般的な工レメントを示しています。

JSON ポリシーエレメント	[概要]
Version	ポリシー言語の最新バージョンは 2012-10-17 です。すべてのアクセスポリシーでこの値を指定する必要があります。
Effect	このエレメントは、指定されたアクションへのアクセスをステートメントが許可するか拒否するかを指定します。有効な値は Allow または Deny です。
Principal	この要素は、リソースへのアクセスを許可または拒否する AWS アカウント または IAM ロールまたはユーザーを指定し、いくつかの形式を取ることができます。 • AWS アカウント: "Principal":{"AWS": ["123456789012"]} または "Principal":{"AWS": ["arn:aws:iam::123456789012:root"]} • IAM ユーザー: "Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]} • IAM ロール: "Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]} **Important** * ワイルドカードを指定すると、ドメインへの匿名アクセスが有効になります。これは IP ベースの条件を追加する、VPC サポートを使用する、またはきめ細かなアクセスコントロールを有効にしない限り、お勧めしません。さらに、以下のポリシーを注意深く調べて、広範なアクセスを許可していないことを確認します。 • 関連する AWS プリンシパル (IAM ロールなど) にアタッチされているアイデンティティベースのポリシー

JSON ポリシーエレメント	[概要]
	• 関連付けられた AWS リソースにアタッチされたリソースベースのポリシー (AWS Key Management Service KMS キーなど)

JSON ポリシーエレメント	[概要]

Action	OpenSearch サービスは OpenSearch HTTP メソッドのESHttp*アクションを使用します。残りのアクションは設定 API に適用されます。
	
	特定の es: アクションは、リソースレベルの許可をサポートします。例えば、すべてのドメインを削除する許可を付与せずに、1 つの特定のドメインのみ削除する許可をユーザーに付与することができます。その他のアクションはサービス自体に適用されます。例えば、es:ListDomainNames は単一ドメインのコンテキストでは意味を成しませんが、それでもワイルドカードを必要とします。
	
	使用可能なすべてのアクションのリストと、ドメインサブリソース (test-domain/*) 、ドメイン設定 ()、またはサービス (test-domain) にのみ適用されるかどうかについては、「サービス認証リファレンス」の「Amazon OpenSearch Service 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。
	リソースベースのポリシーは、リソースレベルのアクセス権限とは異なっています。リソースベースのポリシー は、ドメインにアタッチする完全な JSON ポリシーです。リソースレベルのアクセス許可は、特定のドメインあるいはサブリソースにアクションを制限します。実際には、リソースレベルのアクセス許可はリソース、あるいはアイデンティティベースのポリシーのオプションの一部として捉えることができます。
	
	es:CreateDomain へのリソースレベルのアクセス権限は直観的ではないように見えることがあります。(つまり、すでに存在するドメインを作成するアクセス権限をなぜユーザーに付与するのでしょうか？) ワイルドカードの使用でドメインに簡単な命名スキームを適用できます ("Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*" など)。
	
	もちろん、次のように、制限がより緩和されたリソース要素のアクションを含めることもできます。
	
	`{`
	`  "Version": "2012-10-17",`
	`  "Statement": [`

JSON ポリシーエレメント	[概要]

	<pre>{
	  "Effect": "Allow",
	  "Action": [
	  "es:ESHttpGet",
	  "es:DescribeDomain"
	  ],
	  "Resource": "*"
	  }
	  ]
	}</pre>
	アクションとリソースのペアリングについての詳細は、テーブルの Resource 要素を参照してください。

Condition	OpenSearch サービスは、IAM ユーザーガイド AWS のグローバル条件コンテキストキーで説明されているほとんどの条件をサポートします。注目すべき例外には、 OpenSearch サービスがサポートしていない aws:PrincipalTag キーが含まれます。
	
	IP ベースのポリシーを設定する場合、次に示すように、IP アドレスまたは CIDR ブロックを条件として指定します。
	
	<pre>"Condition": {
	  "IpAddress": {
	  "aws:SourceIp": [
	  "192.0.2.0/32"
	  ]
	  }
	}</pre>
	で説明したようにthe section called “アイデンティティベースのポリシー”、、aws:RequestTag 、および aws:ResourceTag aws:TagKeys 条件キーは、設定 API と OpenSearch APIsに適用されます。

JSON ポリシーエレメント	[概要]

Resource	OpenSearch サービスは、次の 3 つの基本的な方法でResource要素を使用します。
	
	• などの OpenSearch サービス自体に適用されるアクションes:ListDomainNames や、フルアクセスを許可するアクションには、次の構文を使用します。
	
	`"Resource": "*"`
	
	• es:DescribeDomain のように、ドメインの設定に関連するアクションには、次の構文を使用します。
	
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name "`
	
	• es:ESHttpGet のように、ドメインのサブリソースを適用するアクションには、次の構文を使用します。
	
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /*"`
	
	ワイルドカードを使用する必要はありません。 OpenSearch サービスでは、 OpenSearch インデックスまたは API ごとに異なるアクセスポリシーを定義できます。例えば、test-index インデックスへのユーザーのアクセス許可を制限できます。
	
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index"`
	
	test-index へのフルアクセスの代わりに、検索 API のみにポリシーを制限することもできます。
	
	`"Resource": "arn:aws:es: region:aws-account-id:domain/domain-name /test-index/_search"`

JSON ポリシーエレメント	[概要]
	個々のドキュメントへのアクセスを制御することもできます。 `"Resource": "arn:aws:es:  region:aws-account-id:domain/domain-name /test-index/test-type/1"` 基本的に、 がサブリソースを URI として OpenSearch 表現する場合、アクセスポリシーを使用してサブリソースへのアクセスを制御できます。どのリソースにユーザーがアクセスできるかをさらに細かく制御するには、「the section called “きめ細かなアクセスコントロール”」を参照してください。 どのアクションがリソースレベルのアクセス権限をサポートするかの詳細については、このテーブルの Action 要素を参照してください。

詳細オプションと API に関する考慮事項

OpenSearch サービスにはいくつかの高度なオプションがあり、そのうちの 1 つはアクセスコントロールに影響しますrest.action.multi.allow_explicit_index。デフォルト設定の true では、特定の状況下でユーザーがサブリソースへのアクセス権限を回避することが許可されます。

例えば、次のリソースベースのポリシーを考えます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
```

```
      "Resource": [
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/*",
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/_bulk"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-
index/*"
    }
  ]
}
```

このポリシーは、 test-indexと OpenSearch 一括 API へのtest-userフルアクセスを許可します。また、GET への restricted-index リクエストを許可します。

次のインデックスリクエストは、見てわかるように、アクセス権限エラーによって失敗します。

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

インデックス API とは異なり、バルク API では、単一の呼び出しで多くのドキュメントの作成、更新、削除を実行できます。多くの場合、これらのオペレーションはリクエスト URL ではなく、リクエストの本文で指定します。 OpenSearch サービスは URLs を使用してドメインサブリソースへのアクセスを制御するため、 は実際に一括 API を使用して に変更を加えるtest-userことができますrestricted-index。ユーザーにはインデックスで POST アクセス権限が欠如していますが、次のリクエストは成功します。

```
POST https://search-test-domain.us-west-1.es.amazonaws.com/_bulk
```

```
{ "index" : { "_index": "restricted-index", "_type" : "movie", "_id" : "1" } }
{ "title": "Your Name", "director": "Makoto Shinkai", "year": "2016" }
```

このような状況では、アクセスポリシーはその目的達成に失敗します。ユーザーがこれらの制限を回避することを防ぐためには、rest.action.multi.allow_explicit_index を false に変更できます。この値が false の場合、リクエストボディでインデックス名を指定するバルク、mget、および msearch API へのすべての呼び出し動作は停止します。つまり、_bulk への呼び出しは機能しなくなりますが、test-index/_bulk への呼び出しは動作します。この 2 番目のエンドポイントにはインデックス名が含まれるため、リクエストボディにそれを指定する必要はありません。

OpenSearch ダッシュボードはmgetとmsearchに大きく依存しているため、この変更後に正しく動作する可能性はほとんどありません。部分的な解決策としては、rest.action.multi.allow_explicit_index を true のまま残して、特定のユーザーが 1 つ以上の API にアクセスすることを拒否します。

この設定の変更については、「the section called “高度なクラスター設定”」を参照してください。

同様に、以下のリソースベースのポリシーには 2 つの小さな問題があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    },
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-
index/*"
    }
  ]
}
```

- 明示的な拒否にも関わらず、test-user が GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search や GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search などの呼び出しを実行して、restricted-index のドキュメントにアクセスできることです。
- Resource エレメントレファレンス restricted-index/* により、test-user はインデックスのドキュメントに直接アクセスする権限がありません。しかし、ユーザーにはインデックス全体を削除する権限があります。アクセスと削除を防ぐには、このポリシーで restricted-index* を指定する必要があります。

広範囲な許可と一部の拒否を混合するよりは、最小限の権限の原則を守り、タスクを実行するために必要なアクセス権限のみを付与することが最も安全なアプローチです。個々のインデックスまたは OpenSearchオペレーションへのアクセスの制御の詳細については、「」を参照してくださいthe section called “きめ細かなアクセスコントロール”。

> **Important**
>
> * ワイルドカードを指定すると、ドメインへの匿名アクセスが可能になります。ワイルドカードを使用することはお勧めしません。さらに、以下のポリシーを注意深く調べて、広範なアクセスを許可していないことを確認してください。
>
> - 関連付けられた AWS プリンシパルにアタッチされたアイデンティティベースのポリシー (IAM ロールなど)
> - 関連付けられたリソースにアタッチされた AWS リソースベースのポリシー (AWS Key Management Service KMS キーなど)

アクセスポリシーの設定

- Service でリソースベースおよび IP ベースのポリシーを作成または変更する手順については OpenSearch 、「」を参照してくださいthe section called “アクセスポリシーの設定”。
- IAM でアイデンティティベースのポリシーを作成または変更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 。

追加のサンプルポリシー

この章には多くのサンプルポリシーが含まれていますが、 AWS アクセスコントロールは複雑なテーマであり、例として理解するのが最善です。詳細については、IAM ユーザーガイドの「IAM 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 OpenSearch Service API アクセス許可リファレンス

アクセスコントロールをセットアップするときに、IAM ID (アイデンティティベースのポリシー) にアタッチできるアクセス許可ポリシーを記述します。詳細については「サービス認証リファレンスガイド」で以下のトピックを参照してください。

- OpenSearch サービス のアクション、リソース、および条件キー。
- Ingestion のアクション、リソース OpenSearch 、および条件キー。

このリファレンスには、IAM ポリシーで使用できる API オペレーションに関する情報が含まれています。また、アクセス許可を付与できる AWS リソースと、きめ細かなアクセスコントロールのために含めることができる条件キーも含まれています。

ポリシーの `Action` フィールドにアクションを、ポリシーの `Resource` フィールドにリソース値を、ポリシーの `Condition` フィールドに条件を指定します。 OpenSearch サービスのアクションを指定するには、`es:`プレフィックス の後に API オペレーション名 (例:) を使用しますes:CreateDomain。Ingestion のアクションを指定するには、 `osis:` プレフィックスの後に API OpenSearch オペレーション (などosis:CreatePipeline) を使用します。

AWS Amazon OpenSearch Service 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。 AWS のサービス は、新しい 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 管理ポリシー」を参照してください。

AmazonOpenSearchDirectQueryGlueCreateAccess

Amazon OpenSearch Service Direct Query Service に `CreateDatabase`、`CreatePartitionCreateTable`、および へのアクセスを許可しますBatchCreatePartition AWS Glue API。

AmazonOpenSearchDirectQueryGlueCreateAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchServiceFullAccess

OpenSearch のサービス設定 API オペレーションと のリソースへのフルアクセスを許可します AWS アカウント。

AmazonOpenSearchServiceFullAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchServiceReadOnlyAccess

のすべての OpenSearch サービスリソースへの読み取り専用アクセスを許可します AWS アカウント。

AmazonOpenSearchServiceReadOnlyAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchServiceRolePolicy

IAM エンティティに `AmazonOpenSearchServiceRolePolicy` をアタッチすることはできません。このポリシーは、サービスがアカウントリソースにアクセスできるようにする OpenSearch サービスにリンクされたロールにアタッチされます。詳細については、「the section called “アクセス許可”」を参照してください。

AmazonOpenSearchServiceRolePolicy ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchServiceCognitoAccess

Cognito 認証を有効にするために必要な最小限の Amazon Cognito の許可を提供します。

AmazonOpenSearchServiceCognitoAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchIngestionServiceRolePolicy

IAM エンティティに `AmazonOpenSearchIngestionServiceRolePolicy` をアタッチすることはできません。このポリシーは、 OpenSearch 取り込みパイプラインの VPC アクセスを有効にし、タグを作成し、取り込み関連の CloudWatch メトリクスをアカウントに発行できるようにするサービスにリンクされたロールにアタッチされます。詳細については、「the section called “サービスリンクロールの使用”」を参照してください。

AmazonOpenSearchIngestionServiceRolePolicy ポリシーは IAM コンソールで確認できます。

OpenSearchIngestionSelfManagedVpcePolicy

IAM エンティティに `OpenSearchIngestionSelfManagedVpcePolicy` をアタッチすることはできません。このポリシーは、 OpenSearch 取り込みパイプラインの自己管理型 VPC アクセスを有効にし、タグを作成し、取り込み関連の CloudWatch メトリクスをアカウントに発行できるようにするサービスにリンクされたロールにアタッチされます。詳細については、「the section called “サービスリンクロールの使用”」を参照してください。

OpenSearchIngestionSelfManagedVpcePolicy ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchIngestionFullAccess

の Ingestion API OpenSearch オペレーションとリソースへのフルアクセスを許可します AWS アカウント。

AmazonOpenSearchIngestionFullAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchIngestionReadOnlyAccess

のすべての OpenSearch 取り込みリソースへの読み取り専用アクセスを許可します AWS アカウント。

AmazonOpenSearchIngestionReadOnlyAccess ポリシーは IAM コンソールで確認できます。

AmazonOpenSearchServerlessServiceRolePolicy

サーバー OpenSearch レスメトリクスデータを に送信するために必要な最小限の Amazon CloudWatch アクセス許可を提供します CloudWatch。

AmazonOpenSearchServerlessServiceRolePolicy ポリシーは IAM コンソールで確認できます。

OpenSearch AWS マネージドポリシーのサービスの更新

この OpenSearch サービスが変更の追跡を開始してからの、 サービスの AWS マネージドポリシーの更新に関する詳細を表示します。

変更	説明	日付
「OpenSearchIngestionSelfManagedVpcePolicy 」を追加	OpenSearch 取り込みパイプラインの自己管理型 VPC アクセスを有効にし、タグを作成し、取り込み関連の CloudWatch メトリクスをアカウントに発行できるようにする新しいポリシー。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2024 年 6 月 12 日
追加済みAmazonOpenSearchDirectQueryGlueCreateAccess	Amazon OpenSearch Service Direct Query Service に CreateDatabase 、CreatePartition CreateTable 、および へのアクセスを許可しますBatchCreatePartition AWS Glue API。	2024 年 5 月 6 日
「AmazonOpenSearchServiceRolePolicy 」および「AmazonElasticsearchServiceRolePolicy 」を更新しました。	サービスにリンクされたロールが IPv6 アドレスの割り当てと割り当て解除を行うために必要なアクセス許可が追加されました。 廃止された Elasticsearch ポリシー も下位互換性を確保するために更新されました。	2023 年 10 月 18 日

変更	説明	日付
「AmazonOpenSearchIngestionServiceRolePolicy 」を追加	OpenSearch 取り込みパイプラインの VPC アクセスを有効にし、タグを作成し、取り込み関連の CloudWatch メトリクスをアカウントに発行できるようにする新しいポリシー。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2023 年 4 月 26 日
「AmazonOpenSearchIngestionFullAccess 」を追加	の Ingestion API OpenSearch オペレーションとリソースへのフルアクセスを許可する新しいポリシー AWS アカウント。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2023 年 4 月 26 日
「AmazonOpenSearchIngestionReadOnlyAccess 」を追加	のすべての Ingestion OpenSearch リソースへの読み取り専用アクセスを許可する新しいポリシー AWS アカウント。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2023 年 4 月 26 日

変更	説明	日付
「AmazonOpenSearchServerlessServiceRolePolicy 」を追加	OpenSearch Serverless メトリクスデータを に送信するために必要な最小限のアクセス許可を提供する新しいポリシー Amazon CloudWatch。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2022 年 11 月 29 日
「AmazonOpenSearchServiceRolePolicy 」および「AmazonElasticsearchServiceRolePolicy 」を更新しました。	サービスにリンクされたロールがOpenSearch サービスマネージド VPC エンドポイント を作成するために必要なアクセス許可を追加しました。アクションには、リクエストにタグ OpenSearchManaged=true が含まれていないと実行できないものがあります。 廃止された Elasticsearch ポリシー も下位互換性を確保するために更新されました。	2022 年 11 月 7 日

変更	説明	日付
「AmazonOpenSearchServiceRolePolicy 」および「AmazonElasticsearchServiceRolePolicy 」を更新しました。	OpenSearch クラスターメトリクスを Amazon に発行するために必要な PutMetricData アクションのサポートが追加されました CloudWatch。 廃止された Elasticsearch ポリシー も下位互換性を確保するために更新されました。 ポリシーの JSON については、「IAM コンソール」を参照してください。	2022 年 9 月 12 日
「AmazonOpenSearchServiceRolePolicy 」および「AmazonElasticsearchServiceRolePolicy 」を更新しました。	acm リソースタイプのサポートを追加しました。このポリシーは、カスタムエンドポイントが有効なドメインを作成および更新するために、サービスにリンクされたロールが ACM リソースを検証および検証するために必要な最小限の AWS Certificate Manager (ACM) 読み取り専用アクセス許可を提供します。 非推奨の Elasticsearch ポリシーも、下位互換性を確保するために更新されました。	2022 年 7 月 28 日

変更	説明	日付
「AmazonOpenSearchServiceCognitoAccess 」および「AmazonESCognitoAccess 」を更新しました。	UpdateUserPoolClient アクションのサポートが追加されました。これは、Elasticsearch からへのアップグレード中に Cognito ユーザープール設定を設定するために必要です OpenSearch。 SetIdentityPoolRoles すべてのリソースへのアクセスを許可するアクションの権限を修正しました。 非推奨の Elasticsearch ポリシーも、下位互換性を確保するために更新されました。	2021 年 12 月 20 日
「AmazonOpenSearchServiceRolePolicy 」を更新	security-group リソースタイプのサポートを追加しました。このポリシーは、VPC アクセスを有効にするために、サービスにリンクされたロールで必要な最小限の Amazon EC2 および Elastic Load Balancing の許可を提供します。	2021 年 9 月 9 日

変更	説明	日付
• 「AmazonOpenSearchServiceFullAccess 」を追加 • 「AmazonESFullAccess 」を廃止	この新しいポリシーは、古いポリシーを置き換えるためのものです。どちらのポリシーも、 OpenSearch サービス設定 API および OpenSearch APIs。きめ細かなアクセスコントロールおよびリソースベースのポリシーは引き続きアクセスを制限できます。	2021 年 9 月 7 日
• 「AmazonOpenSearchServiceReadOnlyAccess 」を追加 • 「AmazonESReadOnlyAccess 」を廃止	この新しいポリシーは、古いポリシーを置き換えるためのものです。どちらのポリシーも、 OpenSearch サービス設定 API (es:Describe* 、es:List*、および es:Get*) への読み取り専用アクセスを提供し、OpenSearch APIs の HTTP メソッドにはアクセスしません。	2021 年 9 月 7 日
• 「AmazonOpenSearchServiceCognitoAccess 」を追加 • 「AmazonESCognitoAccess 」を廃止	この新しいポリシーは、古いポリシーを置き換えるためのものです。どちらのポリシーも、Cognito 認証を有効にするために必要な最小限の Amazon Cognito の許可を提供します。	2021 年 9 月 7 日

変更	説明	日付
• 「AmazonOpenSearchServiceRolePolicy」を追加 • 「AmazonElasticsearchServiceRolePolicy」を廃止	この新しいポリシーは、古いポリシーを置き換えるためのものです。どちらのポリシーも、VPC アクセスを有効にするために、サービスにリンクされたロールで必要な最小限の Amazon EC2 および Elastic Load Balancing の許可を提供します。	2021 年 9 月 7 日
変更の追跡を開始しました	Amazon OpenSearch Service は、 AWSマネージドポリシーの変更を追跡するようになりました。	2021 年 9 月 7 日

サービス間での不分別な代理処理の防止

混乱した代理問題とは、あるアクションを実行する許可を持たないエンティティが、より多くの特権を持つエンティティにアクションの実行を強制できることで生じるセキュリティ上の問題です。AWS では、サービス間でのなりすましが、混乱した代理問題を生じさせ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 (呼び出し先サービス) を呼び出す場合に発生します。呼び出し元サービスが操作され、それ自身のアクセス許可を使用して、本来アクセス許可が付与されるべきではない方法で別の顧客のリソースに対して働きかけることがあります。これを防ぐために AWS では、お客様の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を提供しています。これには、アカウントのリソースへのアクセス権が付与されたサービスプリンシパルを使用します。

リソースポリシー内では aws:SourceArn および aws:SourceAccount のグローバル条件コンテキストキーを使用して、Amazon OpenSearch Service が別のサービスに付与する、リソースへのアクセス許可を制限することをお勧めします。aws:SourceArn の値に Amazon S3 バケット ARN などのアカウント ID が含まれていない場合は、両方のグローバル条件コンテキストキーを使用して、アクセス許可を制限する必要があります。同じポリシーステートメントでこれらのグローバル条件コンテキストキーの両方を使用し、アカウント ID にaws:SourceArn の値が含まれていない場合、aws:SourceAccount 値と aws:SourceArn 値の中のアカウントには、同じアカウント ID

を使用する必要があります。クロスサービスのアクセスにリソースを 1 つだけ関連付けたい場合は、aws:SourceArn を使用します。クロスサービスが使用できるように、アカウント内の任意のリソースを関連づけたい場合は、aws:SourceAccount を使用します。

aws:SourceArn の値は OpenSearch Service ドメインの ARN でなければなりません。

不分別な代理処理の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 を指定しながら、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には、グローバルコンテキスト条件キー aws:SourceArn で、ARN の未知部分を示すためにワイルドカード (*) を使用します。例えば、arn:aws:*es*:*:*123456789012*:* です。

次の例では、OpenSearch Service で aws:SourceArn および aws:SourceAccount グローバル条件コンテキストキーを使用して、混乱した代理問題を回避する方法を示します。

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Sid":"ConfusedDeputyPreventionExamplePolicy",
      "Effect":"Allow",
      "Principal":{
         "Service":"es.amazonaws.com"
      },
      "Action":"sts:AssumeRole",
      "Condition":{
         "StringEquals":{
            "aws:SourceAccount":"123456789012"
         },
         "ArnLike":{
            "aws:SourceArn":"arn:aws:es:region:123456789012:domain/my-domain"
         }
      }
   }
}
```

Amazon サービスのきめ細かいアクセス制御 OpenSearch

きめ細かなアクセス制御により、Amazon Service 上のデータへのアクセスを制御する新たな方法が提供されています。 OpenSearch 例えば、リクエスト者によっては、検索で 1 つのインデックスのみから結果が返されるようにしたい場合があります。ドキュメント内の特定のフィールドを非表示にしたい場合や、特定のドキュメントをまとめて除外したい場合もあります。

きめ細かなアクセスコントロールには、以下の利点があります。

- ロールベースアクセスコントロール
- インデックスレベル、ドキュメントレベル、フィールドレベルのセキュリティ
- OpenSearch ダッシュボード、マルチテナンシー
- およびダッシュボードの HTTP 基本認証 OpenSearch OpenSearch

トピック

全体像:きめ細かいアクセス制御とサービスセキュリティ OpenSearch

Amazon OpenSearch サービスのセキュリティには主に 3 つのレイヤーがあります。

ネットワーク

1 つ目のセキュリティレイヤーはネットワークで、 OpenSearch リクエストがサービスドメインに到達するかどうかを決定します。ドメインを作成するときに [パブリックアクセス] を選択した場合、インターネットに接続されたクライアントからのリクエストがドメインエンドポイントに

到達できます。[VPC アクセス] を選択した場合、クライアントがエンドポイントに到達するためには、クライアントが VPC に接続する必要があります (かつ、関連するセキュリティグループがその接続を許可する必要があります)。詳細については、「the section called “VPC サポート”」を参照してください。

ドメインアクセスポリシー

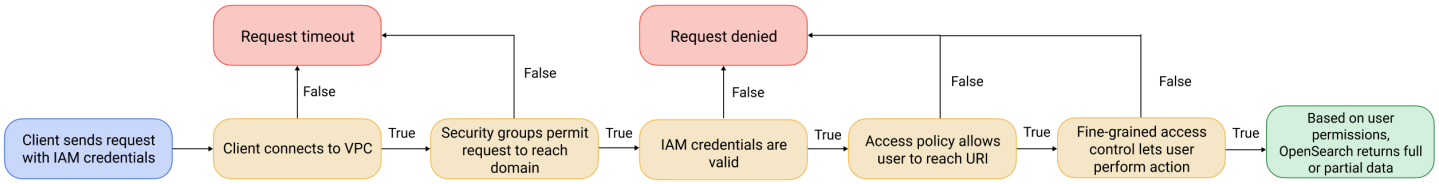
2 番目のセキュリティレイヤーは、ドメインアクセスポリシーです。リクエストがドメインエンドポイントに到達すると、リソースベースのアクセスポリシーにより、指定された URI へのアクセスリクエストが許可または拒否されます。アクセスポリシーは、 OpenSearchリクエストが自身に届く前に、ドメインの「エッジ」でリクエストを受け付けるか拒否します。

きめ細かなアクセスコントロール

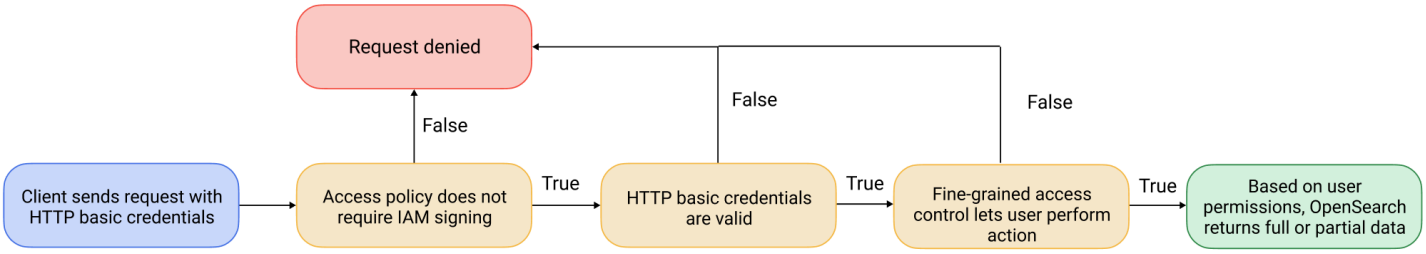
最後の 3 番目のセキュリティレイヤーは、きめ細かなアクセスコントロールです。リソースベースのアクセスポリシーにより、リクエストがドメインエンドポイントに到達することが許可された後、きめ細かなアクセスコントロールにより、ユーザー認証情報が評価されて、ユーザーが認証されるか、リクエストが拒否されます。きめ細かなアクセスコントロールによりユーザーが認証された場合、そのユーザーにマッピングされているすべてのロールが取得され、付与されるすべての許可を使用してリクエストの処理方法が決定されます。

> **Note**
>
> リソースベースのアクセスポリシーに IAM ロールまたはユーザーが含まれている場合、クライアントは署名バージョン 4 AWS を使用して署名付きリクエストを送信する必要があります。そのため、アクセスポリシーは、特に内部ユーザーデータベースと HTTP 基本認証を使用する場合、きめ細かなアクセスコントロールと競合することがあります。ユーザー名とパスワードで、かつ IAM 認証情報で、リクエストに署名することはできません。一般に、きめ細かなアクセスコントロールを有効にする場合は、署名付きリクエストを必須としないドメインアクセスポリシーを使用することをお勧めします。

以下の図表は、きめ細かなアクセスコントロールが有効な VPC アクセスドメイン、IAM ベースのアクセスポリシー、IAM マスターユーザーという、一般的な構成を示しています。



以下の図表は、きめ細かなアクセスコントロールが有効なパブリックアクセスドメイン、IAM プリンシパルを使用しないアクセスポリシー、内部ユーザーデータベース内のマスターユーザーという、別の一般的な構成を示しています。



例

movies/_search?q=thor への GET リクエストを考えてみます。ユーザーに movies インデックスを検索する許可があることを確認します。そうであれば、ユーザーにその内部のすべてのドキュメントを表示するアクセス許可があることを確認します。レスポンスで、フィールドを省略または匿名化するかを選択します。マスターユーザーの場合、レスポンスは以下のようになります。

```
{
  "hits": {
    "total": 7,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "directors": [
            "Kenneth Branagh",
            "Joss Whedon"
          ],
          "release_date": "2011-04-21T00:00:00Z",
          "genres": [
            "Action",
            "Adventure",
```

```
            "Fantasy"
          ],
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor",
          "actors": [
            "Chris Hemsworth",
            "Anthony Hopkins",
            "Natalie Portman"
          ],
          "year": 2011
        }
      },
      ...
    ]
  }
}
```

許可がより制限されたユーザーが上記と同じリクエストを発行した場合、レスポンスは以下のようになります。

```
{
  "hits": {
    "total": 2,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "year": 2011,
          "release_date":
 "3812a72c6dd23eef3c750c2d99e205cbd260389461e19d610406847397ecb357",
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor"
        }
      },
      ...
    ]
```

```
  }
}
```

レスポンスでは、ヒット数、およびヒットあたりのフィールド数が少なくなっています。また、release_date フィールドは匿名化されています。許可のないユーザーが上記と同じリクエストを発行した場合、クラスターによってエラーが返されます。

```
{
  "error": {
    "root_cause": [{
      "type": "security_exception",
      "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
    }],
    "type": "security_exception",
    "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
  },
  "status": 403
}
```

ユーザーが無効な認証情報を提供した場合、クラスターによって Unauthorized 例外が返されます。

主要なコンセプト

きめ細かなアクセス制御を開始する際には、以下の概念を検討してください。

- 役割 — きめ細かなアクセス制御を使用する中核的な方法です。この場合、ロールは IAM ロールとは異なります。ロールには、クラスター全体、インデックス固有、ドキュメントレベル、フィールドレベルの許可の任意の組み合わせが含まれます。
- マッピング — ロールを設定したら、そのロールを 1 人以上のユーザーにマッピングします。例えば、3 つのロールを 1 人のユーザーにマッピングできます。1 つは Dashboards へのアクセスを許可し、1 つは index1 への読み取り専用アクセスを許可し、もう 1 つは index2 への書き込みアクセスを許可します。または、これらのすべての許可を 1 つのロールに含めることもできます。
- ユーザー — OpenSearch クラスターにリクエストを行うユーザーまたはアプリケーション。ユーザーは、リクエストを行うときに指定する認証情報 (IAM アクセスキーまたはユーザー名とパスワードのいずれか) を持っています。

マスターユーザーについて

OpenSearch Service のマスターユーザーは、ユーザー名とパスワードの組み合わせ、 OpenSearch または基盤となるクラスターへの完全な権限を持つ IAM プリンシパルのいずれかです。 OpenSearch クラスターへのすべてのアクセス権を持ち、 OpenSearch ダッシュボード内で内部ユーザー、ロール、ロールマッピングを作成できるユーザーは、マスターユーザーと見なされます。

OpenSearch サービスコンソールまたは CLI で作成されたマスターユーザーは、あらかじめ定義された 2 つのロールに自動的にマップされます。

- `all_access`— クラスター全体のすべての操作へのフルアクセス権、すべてのクラスターインデックスへの書き込み権限、およびすべてのテナントへの書き込み権限を付与します。
- `security_manager`— Security プラグインへのアクセスと、ユーザーと権限の管理を可能にします。

この 2 つのロールを持つユーザーは、 OpenSearch ダッシュボードの [セキュリティ] タブにアクセスして、ユーザーと権限を管理できます。別の内部ユーザーを作成し、`all_access`そのユーザーをそのロールにのみマップした場合、そのユーザーは [セキュリティ] タブにアクセスできません。`all_access``security_manager`とロールの両方に明示的にマッピングすることで、追加のマスターユーザーを作成できます。手順については、「the section called “追加のマスターユーザー”」を参照してください。

ドメインのマスターユーザーを作成する場合、既存の IAM プリンシパルを指定するか、内部ユーザーデータベース内にマスターユーザーを作成できます。どちらを使用するかを決める際には、以下の点を考慮してください。

- IAM プリンシパル — マスターユーザー用に IAM プリンシパルを選択する場合、クラスターへのすべてのリクエストは署名バージョン 4 AWS を使用して署名する必要があります。

  OpenSearch サービスは IAM プリンシパルの権限を一切考慮しません。IAM ユーザーまたはロールは認証のみを目的としています。そのユーザーまたはロールのポリシーは、マスターユーザーの承認には影響しません。承認は OpenSearch Security プラグインのさまざまな権限によって処理されます。

  たとえば、IAM プリンシパルには IAM 権限をまったく割り当てず、マシンまたはユーザーがそのユーザーまたはロールに対して認証できる限り、Service のマスターユーザーの権限を使用できます。 OpenSearch

複数のクラスターで同じユーザーを使用したい場合、Amazon Cognito を使用してダッシュボードにアクセスする場合、または署名バージョン 4 OpenSearch の署名をサポートするクライアントがある場合は、IAM をお勧めします。

- 内部ユーザーデータベース — 内部ユーザーデータベースに (ユーザー名とパスワードの組み合わせで) マスターを作成すると、HTTP 基本認証 (および IAM 認証情報) を使用してクラスターにリクエストを送信できます。curl を含め、ほとんどのクライアントは基本認証をサポートしています。curl は --aws-sigv4 AWS オプション付きの署名バージョン 4 もサポートします。OpenSearch 内部ユーザーデータベースはインデックスに保存されるため、他のクラスターと共有することはできません。

  複数のクラスター間でユーザーを再利用する必要がない場合、Dashboards へのアクセスに Amazon Cognito ではなく HTTP 基本認証を使用する場合、または基本認証のみをサポートするクライアントがある場合は、内部ユーザーデータベースをお勧めします。内部ユーザーデータベースは OpenSearch Service を使い始める最も簡単な方法です。

きめ細かなアクセスコントロールの有効化

コンソール、または設定 API を使用して AWS CLI、きめ細かなアクセス制御を有効にします。この手順については、「ドメインの作成と管理」を参照してください。

きめ細かいアクセス制御には Elasticsearch 6.7 以降が必要です OpenSearch 。また、ドメインへのすべてのトラフィックには HTTPS、保存データの暗号化、暗号化も必要です。node-to-node きめ細かなアクセス制御の、高度な機能の設定方法によっては、さらに多くのリクエストを処理するために、個々のデータノードにコンピューティングリソースとメモリリソースが必要になる場合があります。きめ細かなアクセスコントロールを有効にした後、無効にすることはできません。

既存のドメインでのきめ細かなアクセスコントロールの有効化

Elasticsearch 6.7 OpenSearch 以降を実行している既存のドメインでは、きめ細かいアクセス制御を有効にできます。

既存のドメインに対してきめ細かなアクセスコントロールを有効にするには (コンソール)

1. ドメインを選択し、[アクション] および [セキュリティ設定の編集] を選択します。
2. [きめ細かなアクセスコントロールを有効にする] を選択します。
3. マスターユーザーの作成方法を選択します。

- ユーザー管理に IAM を使用する場合は、[IAM ARN をマスターユーザーとして設定] を選択し、IAM ロールの ARN を指定します。
- 内部ユーザーデータベースを使用する場合は、[Create master user] (マスターユーザーの作成) を選択し、ユーザー名とパスワードを指定します。

4. (オプション) [Open/IP ベースのアクセスポリシーの移行期間を有効にする] を選択します。この設定により、30 日間の移行期間が有効になります。この期間中、既存のユーザーが中断することなくドメインにアクセスできるようになり、IP ベースのアクセスポリシーは引き続きドメインで動作します。この移行期間中に、管理者が必要なロールを作成し、それらをドメインのユーザーにマップすることをお勧めします。オープンアクセスポリシーまたは IP ベースのアクセスポリシーの代わりに ID ベースのポリシーを使用している場合は、この設定を無効にすることができます。

   また、移行期間中にきめ細かなアクセス制御を使用するようにクライアントを更新する必要があります。たとえば、きめ細かいアクセス制御で IAM ロールをマッピングする場合、署名バージョン 4 でリクエストへの署名を開始するようにクライアントを更新する必要があります。AWS きめ細かなアクセス制御を使用して HTTP 基本認証を構成する場合、リクエストに応じた基本認証資格情報を提供するようにクライアントを更新する必要があります。

   移行期間中、 OpenSearch ドメインのダッシュボードエンドポイントにアクセスするユーザーは、ログインページではなくディスカバーページに直接移動します。管理者とマスターユーザーは、Login を選択して管理者の認証情報を使用してログインし、ロールのマッピングを設定できます。

   > **Important**
   >
   > OpenSearch 30 日が経過すると、サービスは自動的に移行期間を無効にします。必要なロールを作成してユーザーにマップしたら、すぐに終了することをお勧めします。移行期間が終了した後、再度有効化することはできません。

5. [変更を保存] を選択します。

この変更により、青/緑のデプロイがトリガーされ、その間にクラスターの状態が赤になりますが、すべてのクラスターオペレーションは影響を受けません。

既存のドメインに対してきめ細かなアクセスコントロールを有効にするには (CLI)

きめ細かなアクセスコントロールを使用して移行期間を有効にするには、AnonymousAuthEnabled を true に設定します。

```
aws opensearch update-domain-config --domain-name test-domain --region us-east-1 \
      --advanced-security-options '{ "Enabled": true,
 "InternalUserDatabaseEnabled":true, "MasterUserOptions": {"MasterUserName":"master-
username","MasterUserPassword":"master-password"},"AnonymousAuthEnabled": true}'
```

default_role について

きめ細かなアクセスコントロールには、ロールマッピングが必要です。ドメインで ID ベースのアクセスポリシーを使用している場合、 OpenSearch Service は既存のユーザーを適切に移行できるように、ユーザーを default_role という新しいロールに自動的にマッピングします。この一時的なマッピングにより、独自のロールマッピングを作成するまで、ユーザーは IAM 署名付き GET および PUT リクエストを正常に送信できます。

このロールによってサービスドメインにセキュリティの脆弱性や欠陥が追加されることはありません。 OpenSearch 独自のロールを設定したら、すぐにデフォルトのロールを削除して、それに応じてマッピングすることをお勧めします。

移行シナリオ

次の表に、既存のドメインできめ細かなアクセス制御を有効にする前と後の各認証方式の動作と、管理者がユーザーをロールに適切にマッピングするために必要な手順について説明します。

認証方法	きめ細かなアクセスコントロールの有効化の前	きめ細かなアクセスコントロールの有効化の後	管理タスク
アイデンティティベースのポリシー	IAM ポリシーを満たすすべてのユーザーが、ドメインにアクセスできます。	移行期間を有効にする必要はありません。 OpenSearch サービスは IAM ポリシーを満たすすべてのユーザーを default_role に自動的にマッピングし、ユーザーが引き続きド	1. ドメインでカスタムロールマッピングを作成します。 2. default_role を削除します。

認証方法	きめ細かなアクセスコントロールの有効化の前	きめ細かなアクセスコントロールの有効化の後	管理タスク
		メインにアクセスできるようにします。	
IP ベースのポリシー	許可された IP アドレスまたは CIDR ブロックのすべてのユーザーがドメインにアクセスできます。	30 日間の移行期間中は、許可された IP アドレスまたは CIDR ブロックのすべてのユーザーが引き続きドメインにアクセスできます。	1. ドメインでカスタムロールマッピングを作成します。 2. ロールマッピングの設定に応じて、基本認証情報または IAM 認証情報を提供するようにクライアントを更新します。 3. 移行期間を無効にします。許可された IP アドレスまたは CIDR ブロックからのユーザーは、基本認証または IAM 認証情報なしでリクエストを送信すると、ドメインへのアクセスが失われます。
オープンアクセスポリシー	インターネット上のすべてのユーザーがドメインにアクセスできます。	30 日間の移行期間中は、インターネット上のすべてのユーザーが引き続きドメインにアクセスできます。	1. ドメインでロールマッピングを作成します。 2. ロールマッピングの設定に応じて、基本認証情報または IAM 認証情報を提供するようにクライアントを更新します。 3. 移行期間を無効にします。基本認証または IAM 認証情報なしでリクエストを送信するユーザーは、ドメインにアクセスできなくなります。

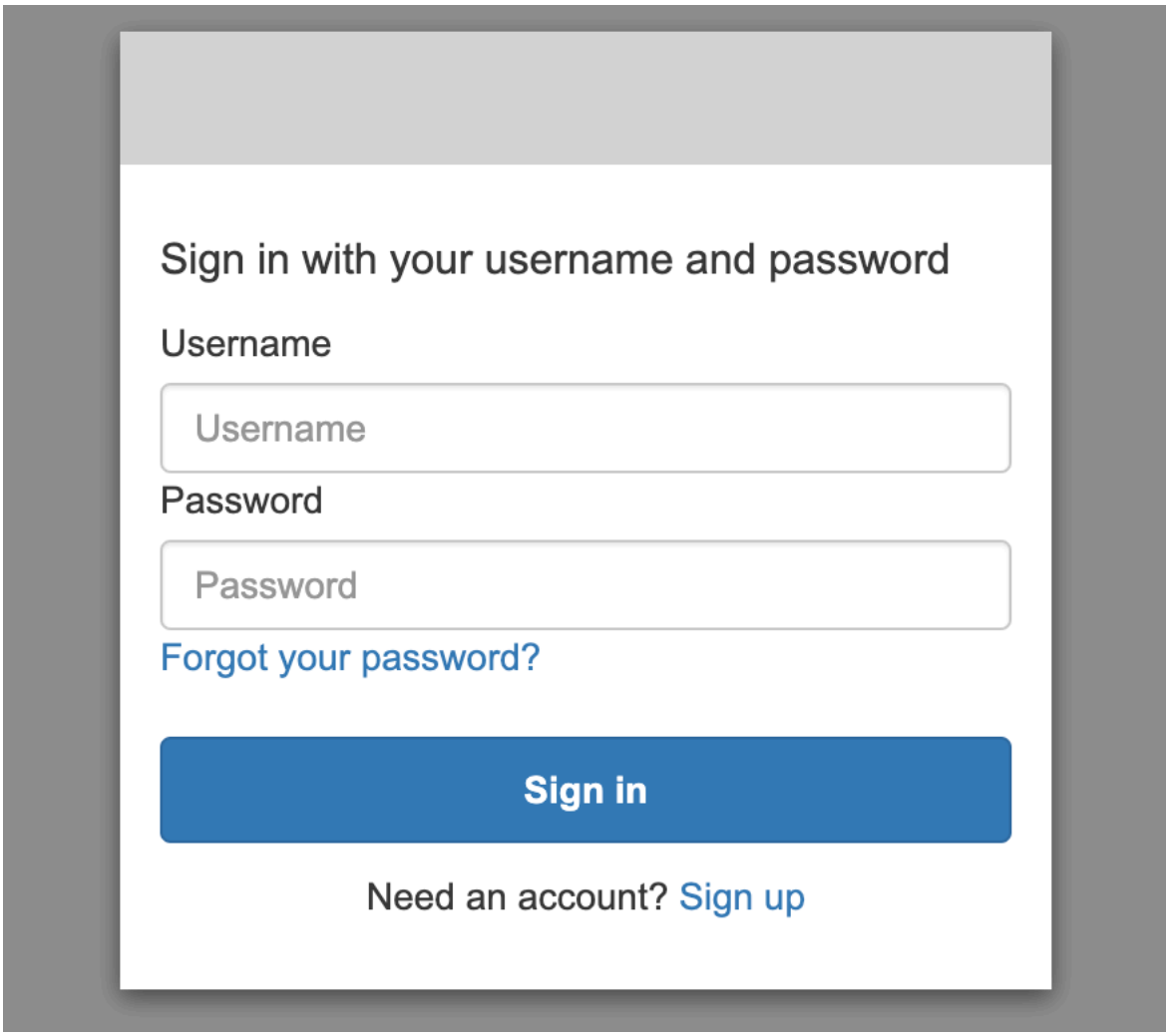
マスターユーザーとしてダッシュボードにアクセスする OpenSearch 。

きめ細かいアクセス制御には、 OpenSearch 管理タスクを簡素化するダッシュボードプラグインがあります。Dashboards を使用して、ユーザー、ロール、マッピング、アクショングループ、テナン

トを管理できます。ただし、 OpenSearch Dashboards のサインインページと基礎となる認証方法は、ユーザーの管理方法やドメインの設定方法によって異なります。

- ユーザー管理に IAM を使用する場合は、the section called “OpenSearch Dashboards の Amazon Cognito 認証” を使用して、Dashboards にアクセスします。それ以外の方法では、Dashboards の「機能しない」サインインページが表示されます。「the section called “制限事項”」を参照してください。

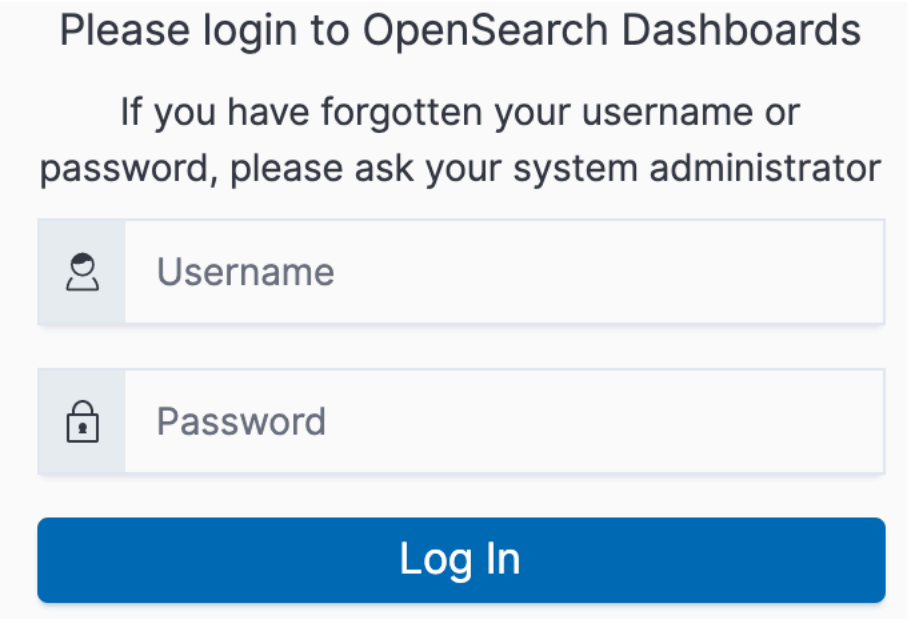
  Amazon Cognito 認証では、ID プールから引き受けたいずれかのロールが、マスターユーザーに指定した IAM ロールと一致する必要があります。この設定の詳細については、「the section called “(オプション) きめ細かなアクセスの設定”」および「the section called “チュートリアル: Cognito 認証によるきめ細かなアクセスコントロール”」を参照してください。



- 内部ユーザーデータベースを使用することを選択した場合は、マスターユーザー名とパスワードを使用して Dashboards にサインインできます。HTTPS 経由で Dashboards にアクセスする必要が

あります。このログイン画面は、Dashboards の Amazon Cognito 認証と SAML 認証の両方に置き換えられます。

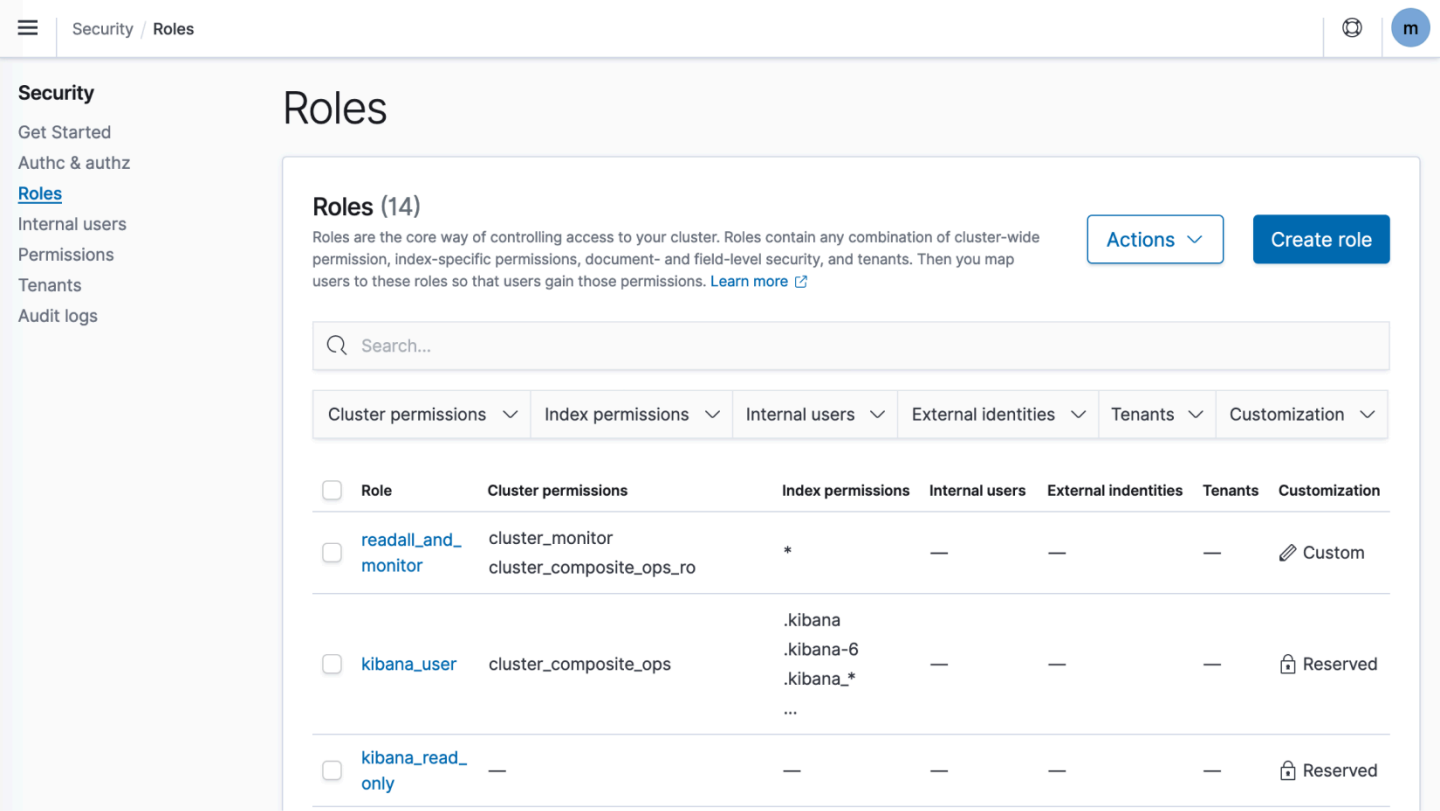
この設定の詳細については、「the section called “チュートリアル: 内部ユーザーデータベースと基本認証”」を参照してください。



- SAML 認証を使用することを選択した場合は、外部 ID プロバイダーからの認証情報を使用してサインインできます。詳細については、「the section called “ OpenSearch Dashboards の SAML 認証”」を参照してください。

許可の管理

「the section called “主要なコンセプト”」で説明しているように、ロール、ユーザー、マッピングを使用してきめ細かなアクセスコントロールの許可を管理します。このセクションでは、これらのリソースを作成して適用する方法について説明します。これらの操作を実行するには、マスターユーザーとして Dashboards にサインインすることをお勧めします。



> **Note**
>
> ユーザーに付与するために選択する権限は、ユースケースによって大きく異なります。このドキュメントですべてのシナリオを実行可能にカバーすることはできません。ユーザーに付与する権限を決定する際には、 OpenSearch 以下のセクションで説明するクラスター権限とインデックス権限を必ず参照し、常に最小権限の原則に従ってください。

ロールの作成

OpenSearch ダッシュボードまたは REST API `_plugins/_security` の操作を使用して、きめ細かなアクセス制御を行う新しいロールを作成できます。詳細については、「ロールの作成」を参照してください。

きめ細かなアクセスコントロールには、多くの事前定義されたロールも含まれます。 OpenSearch Dashboards や Logstash などのクライアントからさまざまなリクエストが寄せられるため OpenSearch、最小限の権限でロールを手動で作成するのは難しい場合があります。例えば、`opensearch_dashboards_user` ロールには、ユーザーがインデックスパターン、可視化、ダッシュボード、およびテナントを操作するのに必要な許可が含まれます。このロールは、他のイ

ンデックスへのアクセスを許可する追加のロールと共に、Dashboards にアクセスするすべてのユーザーまたはバックエンドロールにマッピングすることをお勧めします。

Amazon OpenSearch OpenSearch サービスには以下の役割はありません。

- observability_full_access
- observability_read_access
- reports_read_access
- reports_full_access

Amazon OpenSearch Service には、 OpenSearch以下では利用できないロールがいくつかあります。

- ultrawarm_manager
- ml_full_access
- cold_manager
- notifications_full_access
- notifications_read_access

クラスターレベルのセキュリティ

クラスターレベルの許可により、_mget、_msearch、_bulk などの広範なリクエストの実施、ヘルスのモニタリング、スナップショットの取得などを可能にするかどうかを制御できます。ロールを作成するときに、[クラスター許可] セクションを使用してこれらの許可を管理します。クラスタレベルの権限の完全なリストについては、「クラスタ権限」を参照してください。

多くの場合、個別の権限ではなく、デフォルトのアクショングループの組み合わせを使用して、目的のセキュリティ体制を実現できます。クラスターレベルのアクショングループのリストについては、「クラスターレベル」を参照してください。

インデックスレベルのセキュリティ

インデックスレベルの許可により、新しいインデックスの作成、インデックスの検索、ドキュメントの読み取りと書き込み、ドキュメントの削除、エイリアスの管理などを可能にするかどうかを制御できます。ロールを作成するときに、[インデックス許可] セクションを使用してこれらのアクセス許可を管理します。インデックスレベルの権限の完全なリストについては、「インデックス権限」を参照してください。

多くの場合、個別の権限ではなく、デフォルトのアクショングループの組み合わせを使用して、目的のセキュリティ体制を実現できます。インデックスレベルのアクショングループのリストについては、「インデックスレベル」を参照してください。

ドキュメントレベルのセキュリティ

ドキュメントレベルのセキュリティにより、インデックス内のどのドキュメントをユーザーが表示できるかを制限できます。ロールを作成するときは、 OpenSearch インデックスパターンとクエリを指定します。そのロールにマッピングされたすべてのユーザーは、クエリに一致するドキュメントのみを表示できます。ドキュメントレベルのセキュリティは、検索時に受け取るヒットの数に影響します。

詳細については、「ドキュメントレベルのセキュリティ」を参照してください。

フィールドレベルのセキュリティ

フィールドレベルのセキュリティにより、どのドキュメントフィールドをユーザーが表示できるかを制御できます。ロールを作成するときに、含めるフィールドと除外するフィールドのリストを追加します。フィールドを含めると、そのロールにマッピングされたユーザーはそれらのフィールドのみを表示できます。フィールドを除外すると、除外されたフィールドを除くすべてのフィールドを表示できます。フィールドレベルのセキュリティは、検索時にヒットに含まれるフィールドの数に影響します。

詳細については、「フィールドレベルのセキュリティ」を参照してください。

フィールドのマスキング

フィールドのマスキングは、フィールドレベルのセキュリティに代わるもので、フィールド内のデータを完全に削除するのではなく、匿名化できます。ロールを作成するときに、マスクするフィールドのリストを追加します。フィールドのマスキングは、検索時にフィールドの内容を表示できるかどうかに影響します。

> **Tip**
>
> 標準マスクをフィールドに適用すると、 OpenSearch Service は安全なランダムハッシュを使用するため、集計結果が不正確になる可能性があります。マスキングされたフィールドで集計を実施するには、代わりにパターンベースのマスキングを使用します。

ユーザーの作成

内部ユーザーデータベースを有効にした場合は、 OpenSearchダッシュボードまたは REST API `_plugins/_security` の操作を使用してユーザーを作成できます。詳細については、「ユーザーの作成」を参照してください。

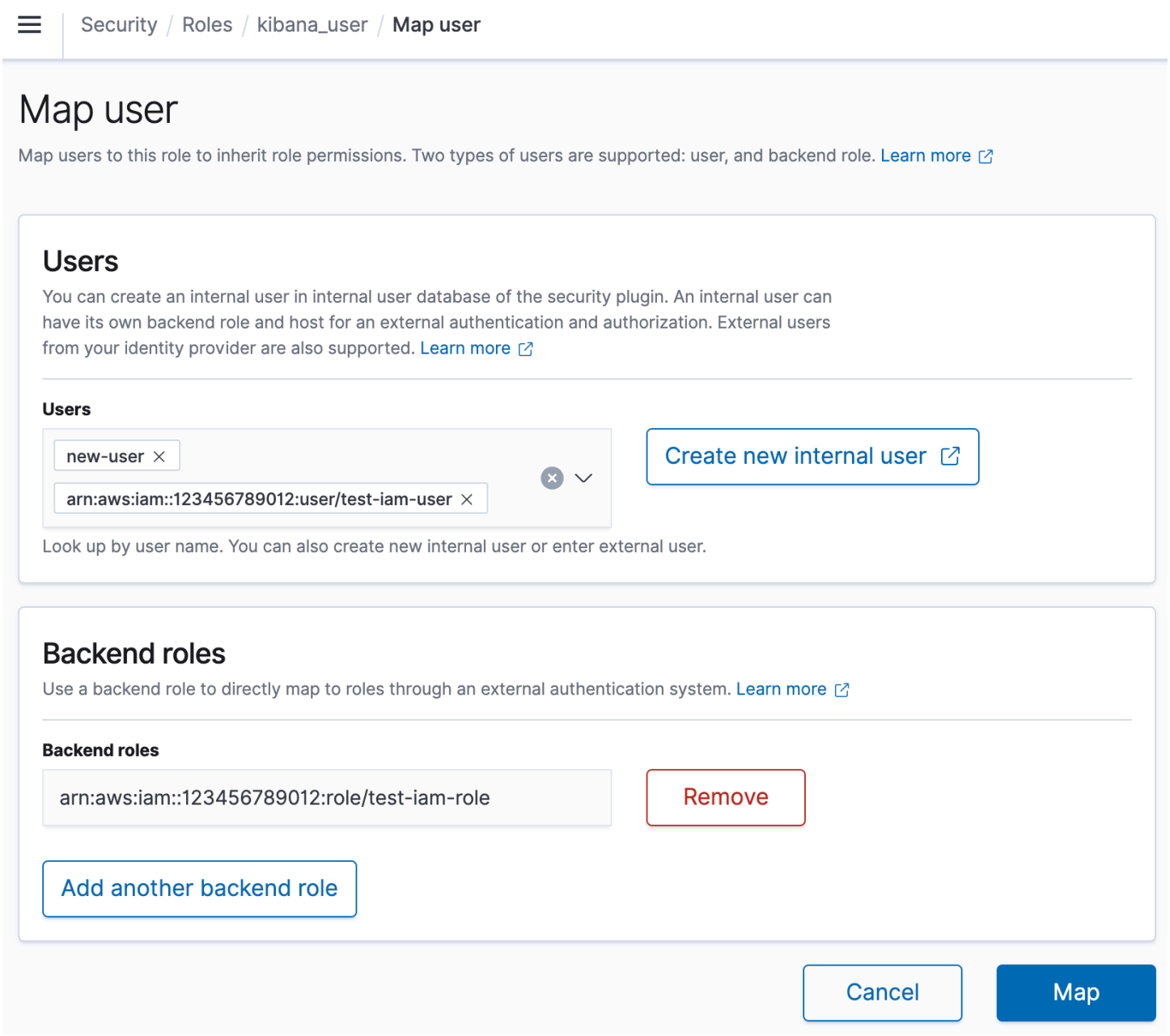
マスターユーザーに IAM を選択した場合は、Dashboards のこの部分は無視してください。その代わりに IAM ロールを作成します。詳細については、IAM ユーザーガイドを参照してください。

ユーザーへのロールのマッピング

ロールのマッピングは、きめ細かなアクセスコントロールの最も重要な側面です。きめ細かなアクセスコントロールには、使用の開始に役立ついくつかの事前定義されたロールがありますが、ロールをユーザーにマッピングしない限り、クラスターに対するすべてのリクエストは許可エラーという結果になります。

バックエンドロールを使用すると、ロールのマッピングプロセスを簡素化できます。このロールを使えば、同じロールを 100 人のユーザーに割り当てるのではなく、100 人のユーザーが共有している 1 つのバックエンドロールにロールをマッピングできます。バックエンドロールは、IAM ロールまたは任意の文字列にすることができます。

- [Users] (ユーザー) セクションで、ユーザー、ユーザー ARN、および Amazon Cognito ユーザー文字列を指定します。Cognito ユーザー文字列の形式は `Cognito/user-pool-id/username` です。
- [バックエンドロール] セクションで、バックエンドロールと IAM ロール ARN を指定します。



OpenSearch ダッシュボードまたは REST API `_plugins/_security` の操作を使用して、ロールをユーザーにマッピングできます。詳細については、「ユーザーをロールにマッピングする」を参照してください。

アクショングループの作成

アクショングループは、さまざまなリソースで再利用できる許可のセットです。 OpenSearch ダッシュボードまたは REST API `_plugins/_security` の操作を使用して新しいアクショングループを作成できますが、ほとんどのユースケースではデフォルトのアクショングループで十分です。デ

フォルトのアクショングループの詳細については、「デフォルトのアクショングループ」を参照してください。

OpenSearch ダッシュボードのマルチテナンシー

テナントは、インデックスパターン、可視化、ダッシュボード、その他の Dashboards オブジェクトを保存するための領域です。Dashboards マルチテナンシーを使用すると、他の Dashboards ユーザーと作業を安全に共有できます (プライベートに保持できます)。また、テナントを動的に設定できます。どのロールがテナントにアクセスできるか、それらのロールに読み取りアクセスがあるか書き込みアクセスがあるかを制御できます。グローバルテナントがデフォルトです。詳細については、「ダッシュボードのマルチテナンシー」を参照してくださいOpenSearch 。

現在のテナントを表示したりテナントを変更したりするには

1. [OpenSearch ダッシュボード] に移動してサインインします。
2. 右上隅にあるユーザーアイコンを選択し、[テナントの切り替え] を選択します。
3. 可視化またはダッシュボードを作成する前に、テナントを確認します。他のすべての Dashboards ユーザーと作業を共有する場合は、[グローバル] を選択します。一部の Dashboards ユーザーと作業を共有するには、別の共有テナントを選択します。それ以外の場合は、[プライベート] を選択します。

> **Note**
>
> OpenSearch Dashboards はテナントごとに個別のインデックスを管理し、というインデックステンプレートを作成します。`tenant_template`インデックスを削除または変更しないでください。`tenant_template`テナントインデックスマッピングの設定を誤ると、OpenSearch ダッシュボードが誤動作するおそれがあります。

推奨される設定

きめ細かなアクセスコントロールは他のセキュリティ機能と連携するため、ここでは、ほとんどのユースケースで適切に機能するきめ細かなアクセスコントロールの推奨される設定を示します。

説明	マスターユーザー	ドメインアクセスポリシー

OpenSearch API の呼び出しには IAM 認証情報を使用し、ダッシュボードへのアクセスには SAML 認証を使用してください。Dashboards または REST API を使用して、きめ細かなアクセスコントロールのロールを管理します。	IAM ロールまたはユーザー	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  }
		  ]
		}</pre>

API の呼び出しには IAM 認証情報または基本認証を使用してください。OpenSearch Dashboards または REST API を使用して、きめ細かなアクセスコントロールのロールを管理します。
 	
この構成は、 OpenSearch特に基本認証のみをサポートするクライアントがある場合に、柔軟性に優れています。
 	
既存の ID プロバイダーがある場合は、SAML 認証を使用して、Dashboards にアクセスします。それ以外の場合は、内部ユーザーデータベー
	ユーザー名とパスワード	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  }
		  ]
		}</pre>

説明	マスターユーザー	ドメインアクセスポリシー
スで Dashboards ユーザーを管理します。		

OpenSearch API の呼び出しには IAM 認証情報を使用し、ダッシュボードへのアクセスには Amazon Cognito を使用します。Dashboards または REST API を使用して、きめ細かなアクセスコントロールのロールを管理します。	IAM ロールまたはユーザー	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  }
		  ]
		}</pre>

OpenSearch API の呼び出しには IAM 認証情報を使用し、ダッシュボードへのほとんどのアクセスをブロックします。REST API を使用して、きめ細かなアクセスコントロールのロールを管理します。	IAM ロールまたはユーザー	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  },
		  {
		  "Effect": "Deny",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /_dashboards*"
		  }
		  ]
		}</pre>

制限事項

きめ細かなアクセスコントロールには、いくつかの重要な制限があります。

- hosts のロールマッピングの側面は、ロールをホスト名または IP アドレスにマッピングすることですが、ドメインが VPC 内にある場合は機能しません。ただし、ロールをユーザーとバックエンドロールにマッピングすることは可能です。
- マスターユーザーに IAM を選択し、Amazon Cognito または SAML 認証を有効にしない場合、Dashboards の「機能しない」サインインページが表示されます。
- マスターユーザーに IAM を選択しても、内部ユーザーデータベースにユーザーを作成することはできます。ただし、この設定では HTTP 基本認証が有効になっていないため、これらのユーザー認証情報で署名されたリクエストは拒否されます。
- SQL を使用して、アクセスできないインデックスをクエリすると、「許可なし」のエラーが表示されます。インデックスが存在しない場合、「インデックスなし」のエラーが表示されます。このエラーメッセージの違いは、インデックスの名前を推測できれば、そのインデックスが存在していることがわかる点です。

  この問題を最小限に抑えるために、インデックス名に機密情報を含めないでください。SQL へのすべてのアクセスを拒否するには、ドメインアクセスポリシーに以下の要素を追加します。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": [
      "*"
    ]
  },
  "Action": [
    "es:*"
  ],
  "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/_plugins/_sql"
}
```

- ドメインバージョンが 2.3 以上で、きめ細かいアクセス制御が有効になっている場合、max_clause_count を 1 に設定するとドメインで問題が発生します。このアカウントにはもっと高い数値を設定することをおすすめします。
- ダイレクトクエリ用に作成されたデータソースについて、きめ細かいアクセス制御が設定されていないドメインで詳細なアクセス制御を有効にする場合は、詳細なアクセス制御ロールを自分で設

定する必要があります。きめ細かいアクセスロールを設定する方法の詳細については、「Amazon S3 との Amazon OpenSearch Service データソース統合の作成」を参照してください。

マスターユーザーの変更

マスターユーザーの詳細を忘れた場合は、コンソール、 AWS CLI、または configuration API を使用して再設定できます。

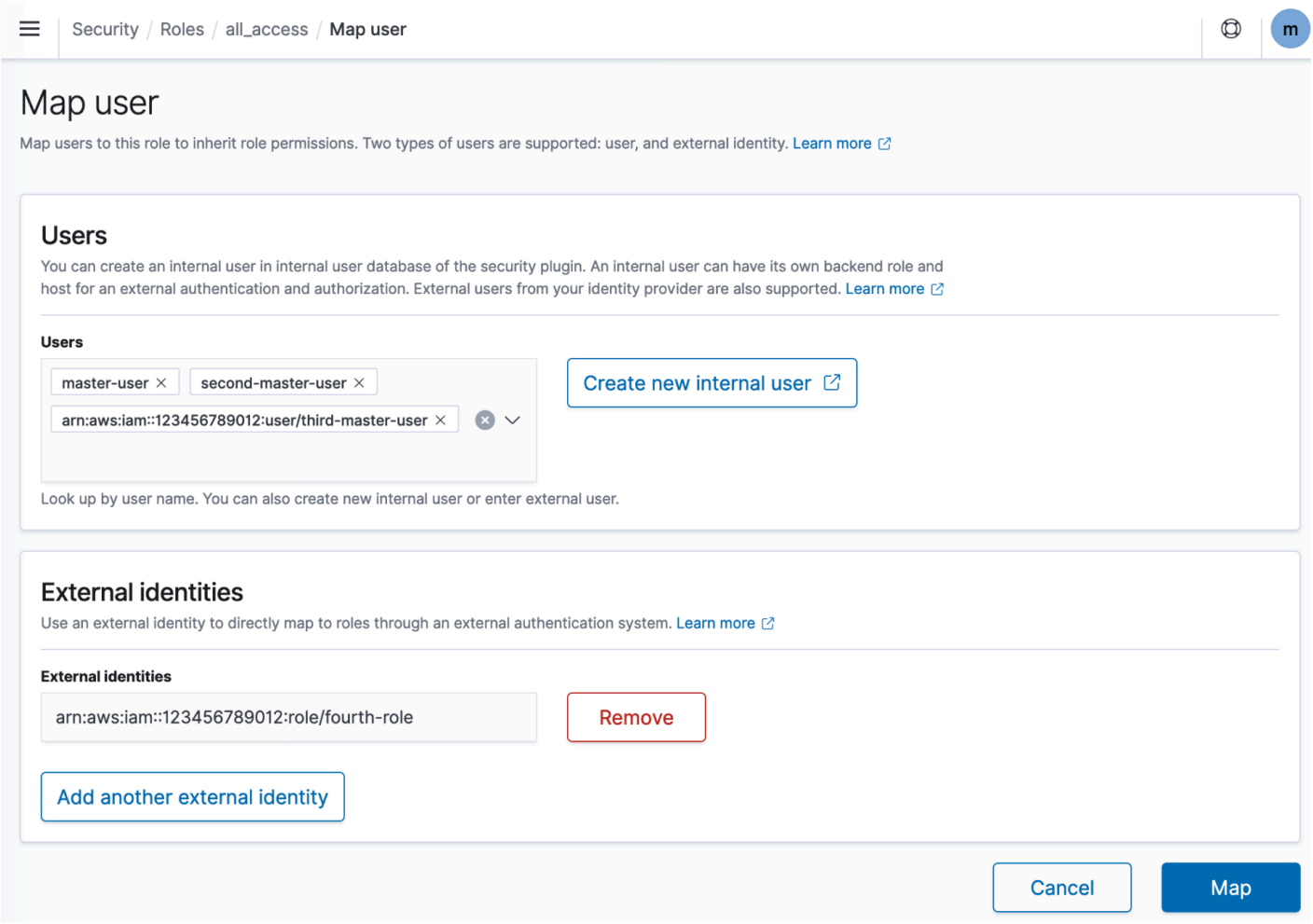
マスターユーザーを変更するには (コンソール)

1. https://console.aws.amazon.com/aos/home/ にある Amazon OpenSearch サービスコンソールに移動します。
2. ドメインを選択し、[Actions] (アクション)、[Edit security configuration] (セキュリティ設定の編集) を選択します。
3. [マスターユーザーとして IAM ARN を設定] または [マスターユーザーを作成] を選択します。
   - 以前に IAM マスターユーザーを使用していた場合、きめ細かなアクセスコントロールにより、指定した新しい IAM ARN に `all_access` ロールが再マッピングされます。
   - 以前に内部ユーザーデータベースを使用していた場合、きめ細かなアクセスコントロールにより、新しいマスターユーザーが作成されます。新しいマスターユーザーを使用して、古いマスターユーザーを削除できます。
   - 内部ユーザーデータベースから IAM マスターユーザーへの切り替えでは、内部ユーザーデータベースからユーザーを削除しません。代わりに、HTTP 基本認証を無効にするだけです。内部ユーザーデータベースからユーザーを手動で削除するか、HTTP 基本認証を再度有効にする必要がある場合に備えて、ユーザーを保持します。
4. [変更を保存] を選択します。

追加のマスターユーザー

ドメインの作成時にマスターユーザーを指定しますが、必要に応じて、このマスターユーザーを使用して追加のマスターユーザーを作成できます。 OpenSearch ダッシュボードと REST API の 2 つのオプションがあります。

- Dashboards で [セキュリティ]、[ロール] の順に選択し、新しいマスターユーザーを `all_access` および `security_manager` ロールにマッピングします。



- REST API を使用するには、以下のリクエストを送信します。

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

```
PUT _plugins/_security/api/rolesmapping/security_manager
{
```

```
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

これらのリクエストは現在のロールマッピングを置き換えるため、PUT リクエストに現在のすべてのロールを含めることができるように、最初に GET リクエストを実行します。REST API が特に役立つのは、Dashboards にアクセスできず、IAM ロールを Amazon Cognito から `all_access` ロールにマッピングする場合です。

手動スナップショット

きめ細かなアクセスコントロールにより、手動スナップショットの取得がいくらか複雑になります。スナップショットリポジトリを登録するには、HTTP 基本認証を他のすべての目的で使用する場合でも、the section called “前提条件” で定義されるように、`TheSnapshotRole` を引き受けるための `iam:PassRole` 許可がある IAM ロールに `manage_snapshots` ロールをマッピングする必要があります。

次に、その IAM ロールを使用して、「the section called “手動スナップショットレポジトリの登録”」に概説されているように、署名付きリクエストをドメインに送信します。

統合

OpenSearch Service AWS で他のサービスを使用する場合は、それらのサービスの IAM ロールに適切な権限を与える必要があります。たとえば、Firehose の配信ストリームでは、という名前の IAM ロールがよく使用されます。`firehose_delivery_role`Dashboards で、きめ細かなアクセスコントロール用のロールを作成し、そのロールに IAM ロールをマッピングします。この場合、新しいロールには以下の許可が必要です。

```
{
  "cluster_permissions": [
    "cluster_composite_ops",
    "cluster_monitor"
```

```
  ],
  "index_permissions": [{
    "index_patterns": [
      "firehose-index*"
    ],
    "allowed_actions": [
      "create_index",
      "manage",
      "crud"
    ]
  }]
}
```

許可は、各サービスが実行するアクションによって異なります。 AWS IoT AWS Lambda データにインデックスを付けるルールや関数には Firehose と同様の権限が必要ですが、検索のみを行う Lambda 関数ではより制限されたセットを使用できます。

REST API の相違点

きめ細かいアクセスコントロール REST API は、/Elasticsearch のバージョンによって若干異なります。 OpenSearchPUT リクエストを行う前に、GET リクエストを行って、想定されるリクエスト本文を確認します。例えば、GET への _plugins/_security/api/user リクエストはすべてのユーザーを返すため、それらの結果を変更して使用して、有効な PUT リクエストを行うことができます。

Elasticsearch 6.x では、ユーザーの作成リクエストは以下のようになります。

```
PUT _opendistro/_security/api/user/new-user
{
  "password": "some-password",
  "roles": ["new-backend-role"]
}
```

Elasticsearch 7.x では、リクエストは次のようになります (Elasticsearch OpenSearch を使用している場合はに変更してください)。_plugins _opendistro

```
PUT _plugins/_security/api/user/new-user
{
  "password": "some-password",
  "backend_roles": ["new-backend-role"]
```

```
}
```

さらに、Elasticsearch 6.x では、テナントはロールのプロパティです。

```
GET _opendistro/_security/api/roles/all_access

{
  "all_access": {
    "cluster": ["UNLIMITED"],
    "tenants": {
      "admin_tenant": "RW"
    },
    "indices": {
      "*": {
        "*": ["UNLIMITED"]
      }
    },
    "readonly": "true"
  }
}
```

Elasticsearch 7.x では OpenSearch 、リクエストは独自の URI を持つオブジェクトです (Elasticsearch を使用している場合はに変更)。`_plugins` `_opendistro`

```
GET _plugins/_security/api/tenants

{
  "global_tenant": {
    "reserved": true,
    "hidden": false,
    "description": "Global tenant",
    "static": false
  }
}
```

OpenSearch REST API のドキュメントについては、セキュリティプラグイン API リファレンスをご覧ください。

> **Tip**
>
> 内部ユーザーデータベースを使用する場合は、curl を使用してリクエストを行い、ドメインをテストできます。次のサンプルコマンドを試してください。

```
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_search'
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_plugins/
_security/api/user'
```

チュートリアル: IAM マスターユーザーと Amazon Cognito 認証を使用してドメインを設定する

このチュートリアルでは、きめ細かなアクセスコントロールを実現する Amazon OpenSearch Service の一般的なユースケース、つまり、ダッシュボード用の Amazon Cognito 認証を使用する IAM マスターユーザーについて説明します。 OpenSearch

マスター IAM ロールと制限付き IAM ロールを設定し、それらを Amazon Cognito のユーザーに関連付けます。その後、 OpenSearch マスターユーザーはダッシュボードにサインインし、制限付きユーザーをロールにマッピングし、きめ細かいアクセスコントロールを使用してユーザーの権限を制限できます。

これらの手順は、認証に Amazon Cognito ユーザープールを使用しますが、この同じ基本プロセスは、Cognito 認証プロバイダに対して機能するため、異なる IAM ロールを異なるユーザーに割り当てることができます。

このチュートリアルでは、次の手順を実行します。

1. マスター IAM ロールと制限付き IAM ロールを作成する
2. Cognito 認証を使用してドメインを作成する
3. Cognito ユーザープールとアイデンティティプールの設定
4. ダッシュボードでのロールのマッピング OpenSearch
5. アクセス許可をテストする

ステップ 1: マスター IAM ロールと制限付き IAM ロールを作成する

AWS Identity and Access Management (IAM) コンソールに移動し、2 つのロールを個別に作成します。

- `MasterUserRole` – マスターユーザー。クラスターに対するフルアクセスの許可を持ち、ロールとロールマッピングを管理します。
- `LimitedUserRole` – マスターユーザーよりも制限されたロール。マスターユーザーから制限付きアクセスが許可されます。

ロールを作成する手順については、「カスタム信頼ポリシーを使用したロールの作成」を参照してください。

両方のロールに次の信頼ポリシーがある必要があります。これにより、Cognito ID プールがロールを引き受けることができます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "{identity-pool-id}"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }]
}
```

> **Note**
>
> `identity-pool-id` を Amazon Cognito ID プールの一意の識別子に置き換えます。例えば `us-east-1:0c6cdba7-3c3c-443b-a958-fb9feb207aa6` です。

ステップ 2: Cognito 認証を使用してドメインを作成する

https://console.aws.amazon.com/aos/home/ にある Amazon OpenSearch サービスコンソールに移動し、以下の設定でドメインを作成します。

- OpenSearch 1.0 以降、または Elasticsearch 7.8 以降
- パブリックアクセス
- マスターユーザー (前のステップで作成) として `MasterUserRole` できめ細かなアクセスコントロールが可能
- Amazon Cognito OpenSearch 認証がダッシュボードに対して有効になっています。Cognito 認証を有効にし、ユーザーと ID プールを選択する手順については、「the section called “Amazon Cognito 認証を使用するためのドメインの設定”」を参照してください。
- 次のドメインアクセスポリシー:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{account-id}:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- ドメインへのすべてのトラフィックに HTTPS を必須とする
- 暗号化は行われません。ode-to-node
- 保管中のデータの暗号化

ステップ 3: Cognito ユーザーの設定

ドメインの作成中に、Amazon Cognito 開発者ガイドの「ユーザープールの作成」に従って、Amazon Cognito 内でマスターユーザーと、制限付きユーザーを設定します。最後に、

「Amazon Cognito でアイデンティティプールを作成する」の手順に従ってアイデンティティプールを設定します。ユーザープールと ID プールは、同じ AWS リージョンに存在している必要があります。

ステップ 4: OpenSearch ダッシュボードでのロールのマッピング

ユーザーを設定したら、 OpenSearch マスターユーザーとしてダッシュボードにサインインし、ユーザーをロールにマップできます。

1. OpenSearch サービスコンソールに戻り、 OpenSearch 作成したドメインのダッシュボード URL に移動します。URL はこの形式に従います: *domain-endpoint*/_dashboards/。
2. master-user 認証情報を使用してサインインします。
3. [Add sample data] (サンプルデータを追加) を選択し、サンプルフライトデータを追加します。
4. 左側のナビゲーションペインで [Security] (セキュリティ)、[Roles] (ロール)、[Create role] (ロールを作成) の順に選択します。
5. ロールに new-role という名前を付けます。
6. [Index] (インデックス) には、opensearch_dashboards_sample_data_fli* (Elasticsearch ドメインの kibana_sample_data_fli*) を指定します。
7. [Index permissions] (インデックスアクセス許可) には、[read] (読み取り) を選択します。
8. [ドキュメントレベルのセキュリティ] で、以下のクエリを指定します。

```
{
  "match": {
    "FlightDelay": true
  }
}
```

9. フィールドレベルのセキュリティでは、[除外] を選択し、FlightNum を指定します。
10. [匿名化] では、Dest を指定します。
11. [作成] を選択します。
12. [マッピングされたユーザー]、[マッピングの管理] を選択します。外部 ID として LimitedUserRole の Amazon リソースネーム (ARN) を追加し、[Map] (マッピング) を選択します。
13. ロールのリストに戻り、[opensearch_dashboards_user] を選択します。[マッピングされたユーザー]、[マッピングの管理] を選択します。バックエンドロールとして LimitedUserRole の ARN を追加し、[マップ] を選択します。

ステップ 5: アクセス許可をテストする

ロールが正しくマッピングされると、制限付きユーザーとしてサインインし、アクセス許可をテストできます。

1. 新しいプライベートブラウザウィンドウで、 OpenSearch ドメインのダッシュボード URL に移動し、`limited-user`認証情報を使用してサインインし、[Explore on my own] を選択します。
2. [開発ツール] に進み、デフォルトの検索を実行します。

   ```
   GET _search
   {
     "query": {
       "match_all": {}
     }
   }
   ```

   許可エラーに注意してください。`limited-user` には、クラスター全体の検索を実行する許可がありません。
3. 別の検索を実行します。

   ```
   GET opensearch_dashboards_sample_data_flights/_search
   {
     "query": {
       "match_all": {}
     }
   }
   ```

   一致するすべてのドキュメントでは、[FlightDelay] フィールドが `true` であり、`Dest` フィールドが匿名化されて、`FlightNum` フィールドはありません。
4. 元のブラウザウィンドウで、`master-user` としてサインインし、[開発ツール] を選択して、同じ検索を実行します。許可、ヒット数、一致するドキュメント、含まれるフィールドが異なっています。

チュートリアル: 内部ユーザーデータベースと HTTP 基本認証でドメインを設定する

このチュートリアルでは、もう 1 つの一般的なきめ細かなアクセス制御のユースケース、内部ユーザーデータベースのマスターユーザーと Dashboards の HTTP 基本認証について説明します。

OpenSearch その後、 OpenSearch マスターユーザーはダッシュボードにサインインし、内部ユーザーを作成し、ユーザーをロールにマップし、きめ細かいアクセス制御を使用してユーザーの権限を制限できます。

このチュートリアルでは、次の手順を実行します。

1. マスターユーザーでドメインを作成する
2. ダッシュボードで内部ユーザーを設定します。 OpenSearch
3. OpenSearch ダッシュボードでの役割のマッピング
4. アクセス許可をテストする

ステップ 1: ドメインを作成する

https://console.aws.amazon.com/aos/home/ にある Amazon OpenSearch サービスコンソールに移動し、以下の設定でドメインを作成します。

- OpenSearch 1.0 以降、または Elasticsearch 7.9 以降
- パブリックアクセス
- 内部ユーザーデータベース内のマスターユーザーによるきめ細かなアクセスコントロール (このチュートリアルの残りの部分では `TheMasterUser`)
- Dashboards の Amazon Cognito 認証が無効にされました
- 以下のアクセスポリシー:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{account-id}:root"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- ドメインへのすべてのトラフィックに HTTPS を必須とする
- 暗号化なし ode-to-node
- 保管中のデータの暗号化

ステップ 2: OpenSearch ダッシュボードで内部ユーザーを作成する

ドメインができたので、 OpenSearch ダッシュボードにサインインして内部ユーザーを作成できます。

1. OpenSearch サービスコンソールに戻り、 OpenSearch 作成したドメインのダッシュボード URL に移動します。URL はこの形式に従います: *domain-endpoint*/_dashboards/。
2. TheMasterUser でサインインします。
3. [Add sample data] (サンプルデータを追加) を選択し、サンプルフライトデータを追加します。
4. 左側のナビゲーションペインで、[セキュリティ]、[内部ユーザー]、[内部ユーザーの作成] を選択します。
5. ユーザーに new-user という名前を付け、パスワードを指定します。次に [作成] を選択します。

ステップ 3: OpenSearch ダッシュボードにロールをマッピングする

ユーザーが設定されたので、ユーザーをロールにマップできます。

1. [OpenSearch ダッシュボード] の [セキュリティ] セクションにとどまり、[ロール]、[ロールの作成] を選択します。
2. ロールに new-role という名前を付けます。
3. [インデックス] には、インデックスパターンに opensearch_dashboards_sample_data_fli* (Elasticsearch ドメインの kibana_sample_data_fli*) を指定します。
4. アクショングループで、[読み取り] を選択します。
5. [ドキュメントレベルのセキュリティ] で、以下のクエリを指定します。

```
{
  "match": {
    "FlightDelay": true
  }
```

```
}
```

6. フィールドレベルのセキュリティでは、[除外] を選択し、FlightNum を指定します。
7. [匿名化] では、Dest を指定します。
8. [作成] を選択します。
9. [マッピングされたユーザー]、[マッピングの管理] を選択します。次に、new-user を [ユーザー] に追加し、[マップ] を選択します。
10. ロールのリストに戻り、[opensearch_dashboards_user] を選択します。[マッピングされたユーザー]、[マッピングの管理] を選択します。次に、new-user を [ユーザー] に追加し、[マップ] を選択します。

ステップ 4: アクセス許可をテストします

ロールが正しくマッピングされると、制限付きユーザーとしてサインインし、アクセス許可をテストできます。

1. 新しいプライベートブラウザウィンドウで、 OpenSearch ドメインのダッシュボード URL に移動し、new-user認証情報を使用してサインインし、[Explore on my own] を選択します。
2. [開発ツール] に進み、デフォルトの検索を実行します。

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

許可エラーに注意してください。new-user には、クラスター全体の検索を実行する許可がありません。

3. 別の検索を実行します。

```
GET dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

一致するすべてのドキュメントでは、[FlightDelay] フィールドが true であり、Dest フィールドが匿名化されて、FlightNum フィールドはありません。

4. 元のブラウザウィンドウで、TheMasterUser としてサインインし、[開発ツール] を選択して、同じ検索を実行します。許可、ヒット数、一致するドキュメント、含まれるフィールドが異なっています。

Amazon OpenSearch Service のコンプライアンス検証

サードパーティーの監査者は、複数の コンプライアンスプログラムの一環として Amazon OpenSearch Service のセキュリティと AWS コンプライアンスを評価します。このプログラムには、SOC、PCI、HIPAA が含まれます。

コンプライアンス要件がある場合は、 OpenSearch または Elasticsearch 6.0 以降の任意のバージョンの使用を検討してください。以前のバージョンの Elasticsearch では、保管中のデータの暗号化とnode-to-node 暗号化の組み合わせは提供されておらず、ニーズを満たす可能性はほとんどありません。ユースケースにとってきめ細かなアクセスコントロールが重要な場合は、 OpenSearch または Elasticsearch 6.7 以降の任意のバージョンの使用を検討することもできます。 ???いずれにしても、ドメインの作成時に特定のバージョン OpenSearch または Elasticsearch バージョンを選択しても、コンプライアンスは保証されません。

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。 では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。

- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャ — このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

  > **Note**
  >
  > すべての AWS のサービス が HIPAA 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に対応するのに役立ちます。
- 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 OpenSearch の耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。AWS リージョン 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・隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、アベイラビリティーゾーン間で中断せずに、自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョン とアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

OpenSearch Service では、AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズに対応できるように複数の機能を提供しています。

- マルチ AZ ドメインとレプリカシャード
- 自動スナップショットと手動スナップショット

Amazon OpenSearch Service の JWT 認証と認可

Amazon OpenSearch Service では、認証と認可に JSON ウェブトークン (JWTsを使用できるようになりました。JWTsは、シングルサインオン (SSO) アクセスを許可するために使用される JSON ベースのアクセストークンです。Service の JWTs を使用して OpenSearch 、 OpenSearch サービスドメインへのリクエストを検証するシングルサインオントークンを作成できます。JWTs を使用するには、きめ細かなアクセスコントロールを有効にし、有効な RSA または ECDSA PEM 形式のパブリックキーを指定する必要があります。きめ細かなアクセスコントロールの詳細については、Amazon OpenSearch Service の「きめ細かなアクセスコントロール」を参照してください。

JSON ウェブトークンは、 OpenSearch サービスコンソール、 AWS Command Line Interface （ AWS CLI ）、または AWS SDKsを使用して設定できます。

考慮事項

Amazon OpenSearch Service で JWTsを使用する前に、次の点を考慮する必要があります。

- PEM 形式の RSA パブリックキーのサイズのため、 AWS コンソールを使用して JWT 認証と認可を設定することをお勧めします。
- JWTs のサブジェクトとロールフィールドを指定するときは、有効なユーザーとロールを指定する必要があります。指定しないと、リクエストは拒否されます。

ドメインアクセスポリシーの変更

JWT 認証と認可を使用するようにドメインを設定する前に、JWT ユーザーがドメインにアクセスできるようにドメインアクセスポリシーを更新する必要があります。それ以外の場合、受信する JWT 認証リクエストはすべて拒否されます。サブリソース (/*) へのフルアクセスを提供するために推奨されるドメインアクセスポリシーは次のとおりです。

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

JWT 認証と認可の設定

JWT 認証と承認は、ドメインの作成プロセス中、または既存のドメインを更新することで有効にできます。セットアップ手順は、選択したオプションによって若干異なります。

次の手順では、 OpenSearch サービスコンソールで JWT 認証と認可用に既存のドメインを設定する方法について説明します。

1. ドメイン設定 で、 の JWT 認証と認可に移動し、JWT OpenSearch認証と認可を有効にする を選択します。
2. ドメインに使用するパブリックキーを設定します。これを行うには、パブリックキーを含む PEM ファイルをアップロードするか、手動で入力します。

   > **Note**
   >
   > アップロードされたキーまたは入力されたキーが有効でない場合、問題を示す警告がテキストボックスの上に表示されます。

3. (オプション) 追加設定で、次のオプションフィールドを設定できます。

- サブジェクトキー — このフィールドを空のままにして、JWTsのデフォルトsubキーを使用できます。
- ロールキー — このフィールドを空のままにして、JWTsのデフォルトrolesキーを使用できます。

変更を加えたら、ドメインを保存します。

JWT を使用してテストリクエストを送信する

指定されたサブジェクトとロールのペアを使用して新しい JWT を作成したら、テストリクエストを送信できます。これを行うには、プライベートキーを使用して、JWT を作成したツールを使用してリクエストに署名します。 OpenSearch サービスは、この署名を検証することで受信リクエストを検証できます。

> **Note**
>
> JWT にカスタムサブジェクトキーまたはロールキーを指定した場合は、JWT に正しいクレーム名を使用する必要があります。

以下は、JWT トークンを使用してドメインの検索エンドポイントを介して OpenSearch Service にアクセスする方法の例です。

```
curl -XGET "$search_endpoint" -H "Authorization: Bearer <JWT>"
```

JWT 認証と認可の設定 (AWS CLI)

次の AWS CLI コマンドは、ドメインが存在する OpenSearch ことを条件として、 の JWT 認証と認可を有効にします。

```
aws opensearch update-domain-config --domain-name <your_domain_name> --advanced-
security-options '{"JWTOptions":{"Enabled":true, "PublicKey": "<your_public_key>",
 "SubjectKey": "<your_subject_key>", "RolesKey": "<your_roles_key>"}}'
```

JWT 認証と認可の設定 (API による設定)

設定 API への次のリクエストは、既存のドメイン OpenSearch で の JWT 認証と承認を有効にします。

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "AdvancedSecurityOptions": {
    "JWTOptions": {
      "Enabled": true,
      "PublicKey": "public-key",
      "RolesKey": "optional-roles-key",
      "SubjectKey": "optional-subject-key"
   }
  }
}
```

キーペアの生成

OpenSearch ドメインに JWTs を設定するには、プライバシー強化メール (PEM) 形式でパブリックキーを指定する必要があります。Amazon OpenSearch Service は現在JWTs を使用する場合、RSA と ECDSA の 2 つの非対称暗号化アルゴリズムをサポートしています。

共通 openssl ライブラリを使用して RSA キーペアを作成するには、次の手順に従います。

1. `openssl genrsa -out privatekey.pem 2048`
2. `openssl rsa -in privatekey.pem -pubout -out publickey.pem`

この例では、 `publickey.pem` ファイルには Amazon OpenSearch Service で使用するパブリックキーが含まれ、 `privatekey.pem`にはサービスに送信された JWTsに署名するためのプライベートが含まれています。さらに、JWTsを生成する必要がある場合は、プライベートキーを一般的に使用されるpkcs8形式に変換することもできます。

アップロードボタンを使用して PEM ファイルをコンソールに直接追加する場合、ファイルには`.pem`拡張子、 などの他のファイル拡張子`.crt`.`cert`、または `.key`が必要です。現時点ではサポートされていません。

Amazon OpenSearch Service 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 OpenSearch Service は グローバル AWS ネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 AWS Well-Architected Framework」の「インフラストラクチャ保護」を参照してください。

が AWS 公開した API コールを使用して、ネットワーク経由で OpenSearch サービスにアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。TLS 1.2、できれば TLS 1.3 が必要です。
- 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。

が AWS 公開した API コールを使用して、ネットワーク経由で OpenSearch サービス設定 API にアクセスします。許容する最低限必要な TLS バージョンを設定するには、ドメインエンドポイントオプションで `TLSSecurityPolicy` 値を次のように指定します。

```
aws opensearch update-domain-config --domain-name my-domain --domain-endpoint-options
 '{"TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"}'
```

詳細については、「AWS CLI コマンドリファレンス」を参照してください。

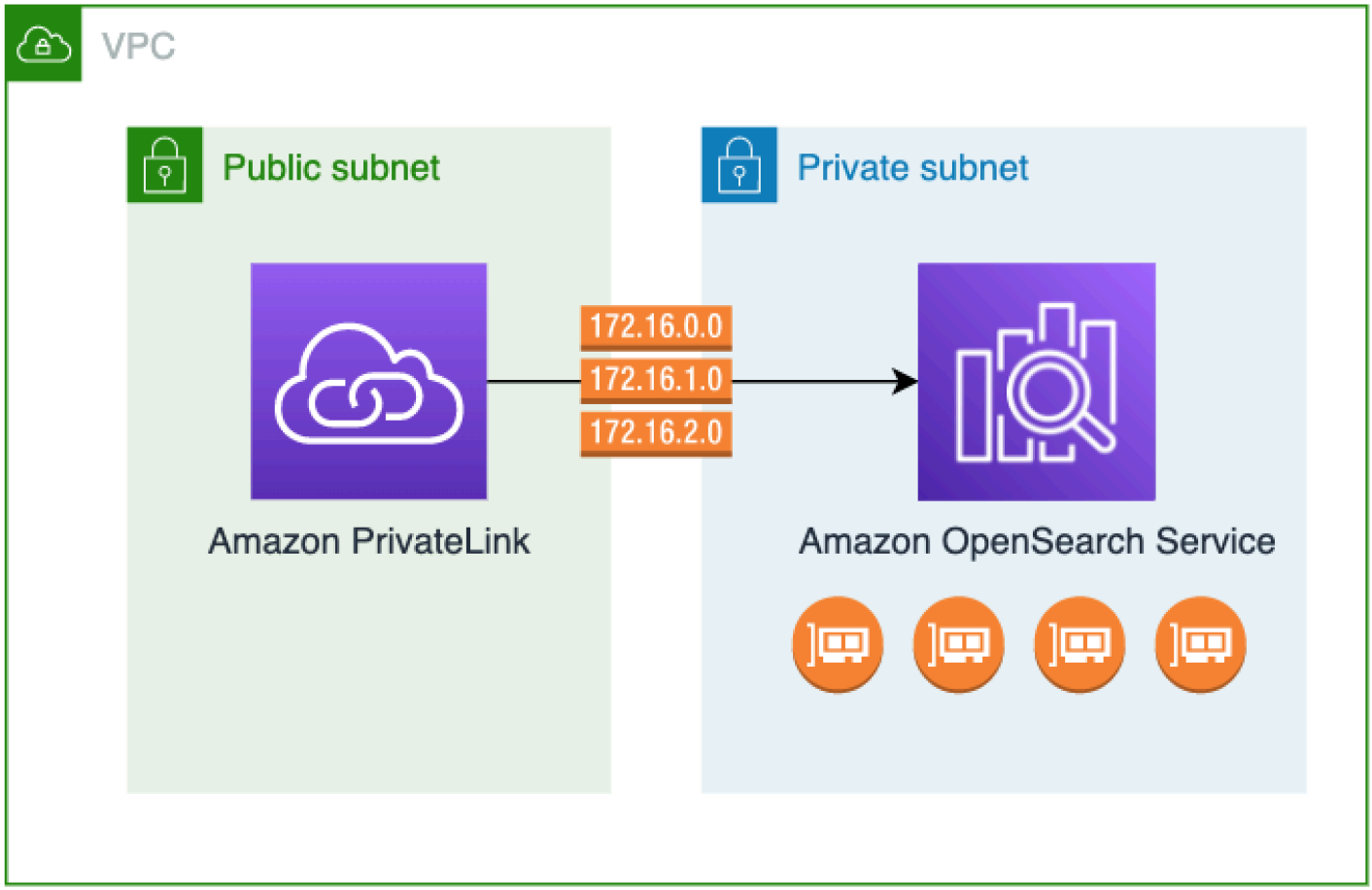
ドメイン設定に応じて、 OpenSearch API へのリクエストに署名する必要があることもあります。詳細については、「the section called “ OpenSearch サービスリクエストの作成と署名”」を参照してください。

OpenSearch サービスは、インターネットに接続されたデバイスからリクエストを受信できるパブリックアクセスドメインと、パブリックインターネットから分離された VPC アクセスドメインをサポートしています。

OpenSearch サービスマネージド VPC エンドポイント (AWS PrivateLink) を使用して Amazon OpenSearch サービスにアクセスする

OpenSearch サービスマネージド VPC エンドポイント (を使用 AWS PrivateLink) を設定することで、Amazon OpenSearch Service ドメインにアクセスできます。これらのエンドポイントは、VPC と Amazon OpenSearch Service の間にプライベート接続を作成します。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続を使用せずに、VPC 内にあるかのように OpenSearch サービス VPC ドメインにアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくても OpenSearch サービスにアクセスできます。

同じ VPC、異なる VPC、または異なる 内のパブリックサブネットまたはプライベートサブネットで実行されている追加のエンドポイントを公開するように OpenSearch サービスドメインを設定できます AWS アカウント。これにより、セキュリティレイヤーを追加して、どこで実行されているかにかかわらずドメインにアクセスできます。インフラストラクチャを管理する必要はありません。次の図は、同じ VPC 内の OpenSearch サービスマネージド VPC エンドポイントを示しています。



このプライベート接続を確立するには、 を使用する OpenSearch サービスマネージドインターフェイス VPC エンドポイント を作成します AWS PrivateLink。インターフェイス VPC エンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスが作成されます。これらは、サービス宛てのトラフィックのエントリポイントとして機能する OpenSearch サービスマネージド型のネットワークインターフェイスです。で請求される OpenSearch サービスマネージド VPC エンドポイントには、標準のAWS PrivateLink インターフェイスエンドポイント料金が適用されます AWS PrivateLink。

OpenSearch および従来の Elasticsearch のすべてのバージョンを実行しているドメインの VPC エンドポイントを作成できます。詳細については、『AWS PrivateLink ガイド』の「AWS PrivateLinkによるアクセス」を参照してください。

OpenSearch サービスの考慮事項と制限事項

OpenSearch サービスのインターフェイス VPC エンドポイントを設定する前に、AWS PrivateLink「 ガイド」の「考慮事項」を確認してください。

OpenSearch サービスマネージド VPC エンドポイントを使用する場合は、次の点を考慮してください。

- VPC ドメインへの接続には、インターフェイス VPC エンドポイントのみを使用できます。パブリックドメインはサポートされません。
- VPC エンドポイントは、同じ AWS リージョン内のドメインにのみ接続できます。
- VPC エンドポイントでサポートされているプロトコルは HTTPS のみです。HTTP は許可されていません。
- OpenSearch サービスは、インターフェイス VPC エンドポイントを介した、サポートされているすべての OpenSearch API オペレーションの呼び出しをサポートしています。
- アカウントごとに最大 50 個のエンドポイント、ドメインごとに最大 10 個のエンドポイントを設定できます。1 つのドメインに含めることができる認証済みプリンシパルの数は、最大 10 です。
- 現在、 AWS CloudFormation を使用してインターフェイス VPC エンドポイントを作成することはできません。
- インターフェイス VPC エンドポイントは、 OpenSearch サービスコンソールまたはOpenSearch サービス API を使用してのみ作成できます。Amazon VPC コンソールを使用して OpenSearch サービスのインターフェイス VPC エンドポイントを作成することはできません。
- OpenSearch サービスマネージド VPC エンドポイントには、インターネットからアクセスできません。 OpenSearch サービスマネージド VPC エンドポイントは、ルートテーブルとセキュリティ

グループで許可されているように、エンドポイントがプロビジョニングされている VPCs 内、またはエンドポイントがプロビジョニングされている VPC とピアリングされている VPC 内でのみアクセスできます。

- VPC エンドポイントポリシーは、 OpenSearch サービスではサポートされていません。セキュリティグループをエンドポイントのネットワークインターフェイスに関連付けて、インターフェイス VPC エンドポイントを介した OpenSearch サービスへのトラフィックを制御できます。
- サービスにリンクされたロールは、VPC エンドポイントの作成に使用するのと同じ AWS アカウントに存在する必要があります。
- OpenSearch サービス VPC エンドポイントを作成、更新、削除するには、Amazon サービスのアクセス許可に加えて、次の Amazon EC2 アクセス許可が必要です。 OpenSearch
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`
  - `ec2:CreateTags`
  - `ec2:DescribeTags`
  - `ec2:DescribeSubnets`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeVpcs`

> **Note**
>
> 現在、VPC エンドポイントの作成を OpenSearch サービスに制限することはできません。今後のアップデートでこれを可能にするよう取り組んでいます。

ドメインへのアクセスを提供する

ドメインにアクセスする VPC が別の にある場合は AWS アカウント、インターフェイス VPC エンドポイントを作成する前に、所有者のアカウントから承認する必要があります。

別の の VPC AWS アカウント にドメインへのアクセスを許可するには

1. https://console.aws.amazon.com/aos/home/ で Amazon OpenSearch Service コンソールを開きます。

2. ナビゲーションペインで、[Domains] (ドメイン) を選択し、アクセス権を付与するドメインを開きます。
3. [VPC endpoints] (VPC エンドポイント) タブに移動すると、ドメインにアクセスできるアカウントと対応する VPC が表示されます。
4. [Authorize principal] (プリンシパルを承認) を選択します。
5. ドメインにアクセスするアカウントの AWS アカウント ID を入力します。このステップでは、指定されたアカウントがドメインに対して VPC エンドポイントを作成することを承認します。
6. [承認] を選択します。

VPC ドメインのインターフェイス VPC エンドポイントを作成する

OpenSearch サービス用のインターフェイス VPC エンドポイントは、 OpenSearch サービスコンソールまたは AWS Command Line Interface () を使用して作成できますAWS CLI。

OpenSearch サービスドメインのインターフェイス VPC エンドポイントを作成するには

1. https://console.aws.amazon.com/aos/home/ で Amazon OpenSearch Service コンソールを開きます。
2. 左のナビゲーションペインで [VPC endpoints] (VPC エンドポイント) を選択します。
3. [エンドポイントの作成] を選択します。
4. 現在の のドメインに接続するか、別の のドメインに接続する AWS アカウント かを選択しますAWS アカウント。
5. このエンドポイントで接続するドメインを選択します。ドメインが現在の にある場合は AWS アカウント、ドロップダウンを使用してドメインを選択します。ドメインが別のアカウントにある場合は、接続するドメインの Amazon リソースネーム (ARN) を入力します。別のアカウントのドメインを選択するには、所有者にドメインへのアクセス権を付与してもらう必要があります。
6. VPC の場合、 OpenSearch サービスにアクセスする VPC を選択します。
7. サブネット で、 OpenSearch サービスにアクセスするサブネットを 1 つ以上選択します。
8. [Security groups] (セキュリティグループ) で、エンドポイントネットワークインターフェイスに関連付けるセキュリティグループを選択します。これは、エンドポイントに対して承認するインバウンドトラフィックのポート、プロトコル、およびソースを制限する重要なステップです。セキュリティグループのルールでは、VPC エンドポイントを使用して OpenSearch サービスと通信し、エンドポイントネットワークインターフェイスと通信するリソースを許可する必要があります。

9. [エンドポイントの作成] を選択します。エンドポイントは 2～5 分でアクティブになるはずです。

設定 API を使用した OpenSearch サービスマネージド VPC エンドポイントの操作

次の API オペレーションを使用して、 OpenSearch サービスマネージド VPC エンドポイントを作成および管理します。

- CreateVpcEndpoint
- ListVpcEndpoints
- UpdateVpcEndpoint
- DeleteVpcEndpoint

VPC ドメインへのエンドポイントアクセスを管理するには、次の API オペレーションを使用します。

- AuthorizeVpcEndpointAccess
- ListVpcEndpointAccess
- ListVpcEndpointsForDomain
- RevokeVpcEndpointAccess

OpenSearch Dashboards の SAML 認証

OpenSearch Dashboards の SAML 認証を使用すると、既存の ID プロバイダーを使用して、OpenSearch または Elasticsearch 6.7 以降を実行している Amazon OpenSearch Service ドメイン上の Dashboards のシングルサインオン (SSO) を提供できます。SAML 認証を使用するには、きめ細かなアクセスコントロールを有効にする必要があります。

OpenSearch Dashboards の SAML 認証では、Amazon Cognito または内部ユーザーデータベース を介して認証するのではなく、サードパーティーの ID プロバイダーを使用して Dashboards にログインし、きめ細かなアクセスコントロールを管理し、データを検索して可視化を構築できます。 OpenSearch サービスは、Okta、Keycloak、Active Directory フェデレーションサービス (ADFS)、Auth0、 などの SAML 2.0 標準を使用するプロバイダーをサポートします AWS IAM Identity Center。

Dashboards の SAML 認証は、ウェブブラウザから OpenSearch Dashboards にアクセスするためだけのものです。SAML 認証情報では、 OpenSearch または APIs に直接 HTTP リクエストを行うことはできません。

SAML 設定の概要

このドキュメントは、ユーザーに既存の ID プロバイダーがあり、そのプロバイダーについてある程度の知識があることを前提としています。正確なプロバイダーに対して詳細な設定手順を提供することはできません。これは、 OpenSearch サービスドメインに対してのみ行います。

OpenSearch Dashboards ログインフローは、次の 2 つの形式のいずれかになります。

- サービスプロバイダー (SP) が開始されている: Dashboards に移動します (例えば、`https://my-domain.us-east-1.es.amazonaws.com/_dashboards`)。これにより、ログイン画面にリダイレクトされます。ログイン後、ID プロバイダーはお客様を Dashboards にリダイレクトします。
- ID プロバイダー (IdP) が開始： ID プロバイダーに移動し、ログインして、アプリケーションディレクトリから OpenSearch Dashboards を選択します。

OpenSearch サービスは、SP 開始と IdP 開始の 2 つのシングルサインオン URLs を提供しますが、必要な OpenSearch Dashboards ログインフローに一致するもののみが必要です。

どの認証タイプを使用するかにかかわらず、目的は ID プロバイダーを介してログインし、ユーザーネーム (必須) とバックエンドロール (オプションですが、推奨されます) を含む SAML アサーションを受け取ることです。この情報により、きめ細かなアクセスコントロールが、SAML ユーザーに許可を割り当てることができます。外部 ID プロバイダーでは、バックエンドロールは通常「ロール」または「グループ」と呼ばれます。

考慮事項

SAML 認証を設定するときは、以下を考慮してください。

- IdP メタデータファイルのサイズにより、SAML 認証の設定には AWS コンソールを使用することを強くお勧めします。
- ドメインは、一度に 1 つの Dashboards 認証方法のみをサポートします。OpenSearch Dashboards の Amazon Cognito 認証が有効になっている場合は、SAML 認証を有効にする前に無効にする必要があります。

- SAML で Network Load Balancer を使用する場合は、最初にカスタムエンドポイントを作成します。詳細については、「???」を参照してください。

VPC ドメインの SAML 認証

SAML は、アイデンティティプロバイダーとサービスプロバイダー間の直接的な通信を必要としません。したがって、 OpenSearch ドメインがプライベート VPC 内でホストされていても、ブラウザが OpenSearch クラスターと ID プロバイダーの両方と通信できる限り、SAML を使用できます。ブラウザは、基本的にアイデンティティプロバイダーとサービスプロバイダーの仲介として機能します。SAML 認証フローを説明する便利な図表については、Okta のドキュメントを参照してください。

ドメインアクセスポリシーの変更

SAML 認証を設定する前に、ドメインアクセスポリシーを更新して SAML ユーザーがドメインにアクセスできるようにする必要があります。これを実行しない場合は、アクセス拒否エラーが表示されます。

次のドメインアクセスポリシーをお勧めします。これにより、ドメイン上のサブリソース (/*) にフルアクセスが付与されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

ポリシーをより制限するには、ポリシーに IP アドレス条件を追加します。この条件は、指定された IP アドレス範囲またはサブネットのみへのアクセスを制限します。例えば、次のポリシーでは、192.0.2.0/24 サブネットからのアクセスのみを許可します。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "domain-arn/*"
    }
  ]
}
```

> **Note**
>
> オープンドメインアクセスポリシーでは、ドメインできめ細かなアクセスコントロールを有効にする必要があります。有効にしないと、次のエラーが表示されます。
> `To protect domains with public access, a restrictive policy or fine-grained access control is required.`
> マスターユーザーまたは内部ユーザーが堅牢なパスワードで設定されている場合、きめ細かなアクセスコントロールの使用中にポリシーを開いたままにしておくことは、セキュリティの観点からも許容できる場合があります。詳細については、「???」を参照してください。

SP 開始認証または IdP 開始認証の設定

これらのステップでは、 OpenSearch Dashboards で SP 開始認証または IdP 開始認証を使用して SAML 認証を有効にする方法について説明します。両方を有効にするために必要な追加のステップについては、「SP 開始認証と IdP 開始認証の両方を有効にする」を参照してください。

ステップ 1: SAML 認証を有効にする

SAML 認証は、ドメインの作成中に有効化、または既存のドメインで [Actions] (アクション)、[Edit security configuration] (セキュリティ設定の編集) の順に選択することで有効化できます。以下のステップは、どちらを選択するかに応じて若干異なります。

ドメイン設定内で、 OpenSearch Dashboards/Kibana の SAML 認証 で、SAML 認証を有効にするを選択します。

ステップ 2: アイデンティティプロバイダーを設定する

SAML 認証をいつ設定しているかに応じて、以下のステップを実行します。

新しいドメインを作成している場合

新しいドメインを作成中の場合、 OpenSearch サービスはまだサービスプロバイダーエンティティ ID または SSO URLs を生成できません。アイデンティティプロバイダーは、SAML 認証を適切に有効化するためにこれらの値を必要としますが、これらの値はドメインの作成後しか生成できません。ドメインの作成中にこの相互依存性を回避するには、IdP 設定に一時的な値を指定して必要なメタデータを生成し、ドメインがアクティブになった後でそれらを更新することができます。

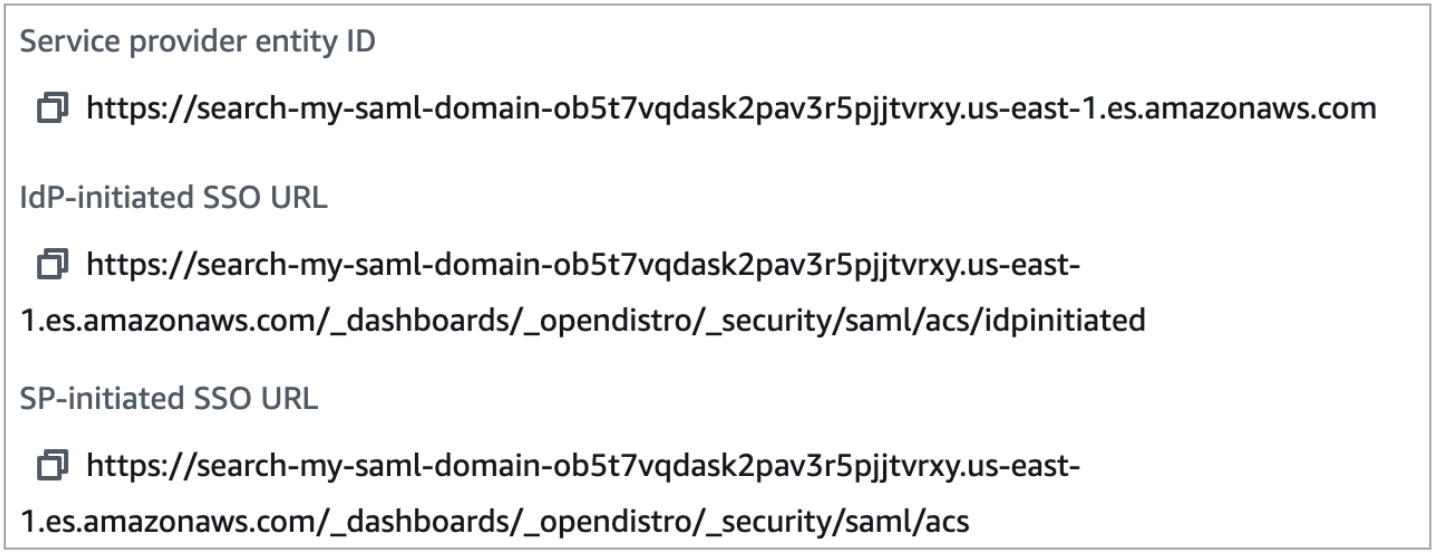
カスタムエンドポイントを使用している場合は、URL がどうなるかを推測できます。例えば、カスタムエンドポイントが `www.custom-endpoint.com` の場合、サービスプロバイダーのエンティティ ID は `www.custom-endpoint.com`、IdP 開始の SSO URL は `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated`、SP 開始の SSO URL は `www.custom-endpoint.com/_dashboards/_opendistro/_security/saml/acs` になります。ドメインを作成される前に、これらの値を使用して ID プロバイダーを設定できます。例については、次のセクションを参照ください。

カスタムエンドポイントを使用していない場合は、IdP に一時的な値を入力して必要なメタデータを生成し、ドメインがアクティブになった後でそれらを更新することができます。

例えば、Okta では、[Single sign on URL] (シングルサインオン URL) と [Audience URI (SP Entity ID)] (オーディエンス URI (SP エンティティ ID) フィールドに `https://temp-endpoint.amazonaws.com` を入力できます。そうすることで、メタデータの生成が可能になります。その後、ドメインがアクティブになったら、 OpenSearch サービスから正しい値を取得し、Okta で更新できます。手順については、「the section called “ステップ 6: IdP URL を更新する”」を参照してください。

既存のドメインを編集している場合

既存のドメインで SAML 認証を有効化している場合は、サービスプロバイダーのエンティティ ID と SSO URL の 1 つをコピーします。使用する URL のガイダンスについては、「the section called "SAML 設定の概要"」を参照してください。



これらの値を使用して、アイデンティティプロバイダーを設定します。これはプロセスの最も複雑な部分ですが、残念ながら、用語とステップはプロバイダーによって大きく異なります。プロバイダーのドキュメントを参照してください。

例えば Okta では、SAML 2.0 ウェブアプリケーションを作成します。[Single sign on URL] (シングルサインオン URL) には、SSO URL を指定します。Audience URI (SP エンティティ ID) では、SP エンティティ ID を指定します。

Okta には、ユーザーおよびバックエンドロールではなく、ユーザーとグループがあります。[Group Attribute Statements] (グループ属性ステートメント) では、`role` を [Name] (名前) フィールドに、正規表現 `.+` を [Filter] (フィルター) フィールドに追加することをお勧めします。このステートメントは、Okta の ID プロバイダーに、ユーザーの認証後に SAML アサーションの `role` フィールドの下に、すべてのユーザーグループを含めるよう指示します。

IAM アイデンティティセンターで、SP エンティティ ID をアプリケーション SAML オーディエンスとして指定します。また、次の属性マッピングも指定する必要があります: `Subject=${user:subject}:format=unspecified` と `Role=${user:groups}:format=uri`

Auth0 では、通常のウェブアプリケーションを作成し、SAML 2.0 アドオンを有効にします。Keycloak では、クライアントを作成します。

ステップ 3: IdP メタデータをインポートする

ID プロバイダーを設定すると、IdP メタデータファイルが生成されます。この XML ファイルには、TLS 証明書、Single Sign-On エンドポイント、ID プロバイダーのエンティティ ID など、プロバイダーに関する情報が含まれています。

IdP メタデータファイルの内容をコピーし、 OpenSearch サービスコンソールの IdP フィールドからメタデータに貼り付けます。または、[XML ファイルからインポート] を選択し、ファイルをアップロードします。メタデータファイルは、次のように表示されます。

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

ステップ 4: SAML フィールドを設定する

IdP メタデータを入力したら、 OpenSearch サービスコンソールで次の追加フィールドを設定します。

- [IdP entity ID] (IdP エンティティ ID) – メタデータファイルから `entityID` プロパティの値をコピーし、このフィールドにそれを貼り付けます。多くの ID プロバイダーは、この値も設定後の概要の一部として表示します。一部のプロバイダーは、それを「発行者」と呼んでいます。

- SAML マスターユーザー名および SAML マスターバックエンドロール – 指定したユーザーおよび/またはバックエンドロールは、新しいマスターユーザー に相当するクラスターへの完全なアクセス許可を受け取りますが、これらのアクセス許可は OpenSearch Dashboards 内でのみ使用できます。

  例えば、Okta では、グループ admins に属しているユーザー jdoe がある可能性があります。jdoe を [SAML マスターユーザーネーム] フィールドに追加した場合、そのユーザーのみが完全な許可を受け取ります。admins を SAML マスターバックエンドロールフィールドに追加する場合は、admins グループに属するすべてのユーザーに完全な許可が付与されます。

  > **Note**
  >
  > SAML アサーションの内容は、SAML マスターユーザーネームおよび SAML マスターロールに使用する文字列と正確に一致する必要があります。一部の ID プロバイダーはユーザー名の前にプレフィックスを追加するため、 hard-to-diagnose 一致しない可能性があります。ID プロバイダーのユーザーインターフェイスで、jdoe を確認できる場合がありますが、SAML アサーションには auth0|jdoe が含まれている可能性があります。常に SAML アサーションからの文字列を使用します。

多くの ID プロバイダーでは、設定プロセス中にサンプルのアサーションを表示できます。また、SAML トレーサーは、実際のアサーションの内容を調べてトラブルシューティングするのに役立ちます。アサーションは次のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion ID="id67229299299259351343340162"
 IssueInstant="2020-09-22T22:03:08.633Z" Version="2.0"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">idp-issuer</
saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">username</saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml2:SubjectConfirmationData NotOnOrAfter="2020-09-22T22:08:08.816Z"
 Recipient="domain-endpoint/_dashboards/_opendistro/_security/saml/acs"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-09-22T21:58:08.816Z"
 NotOnOrAfter="2020-09-22T22:08:08.816Z">
```

```
    <saml2:AudienceRestriction>
      <saml2:Audience>domain-endpoint</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2020-09-22T19:54:37.274Z">
    <saml2:AuthnContext>

 <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport<
saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
      <saml2:AttributeValue
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xs:string">GroupName Match Matches regex ".+" (case-sensitive)
      </saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
```

ステップ 5: (オプション) 追加設定を実行する

[Additional settings] (その他の設定) で、次のオプションフィールドを設定します。

- [Subject key] (件名キー) – このフィールドを空のままにしておいて、ユーザーネームの SAML アサーションの NameID エレメントを使用することができます。アサーションでこの標準エレメントを使用せず、代わりにユーザーネームをカスタム属性として含める場合は、ここでその属性を指定します。
- [Roles key] (ロールキー) – バックエンドロール (推奨) を使用する場合は、このフィールドでアサーションからの属性 (role または group など) を指定します。これは、SAML トレーサーなど、どのツールが役立つかという別の状況です。
- セッション有効期限 – デフォルトでは、 OpenSearch Dashboards は 24 時間後にユーザーをログアウトします。この値は、新しい値を指定することで、60 から 1,440 (24 時間) までの任意の数値に設定できます。

設定に問題がなければ、ドメインを保存します。

ステップ 6: IdP URL を更新する

ドメインの作成中に SAML 認証を有効化した場合は、XML メタデータファイルを生成するために IdP 内で一時的な URL を指定する必要がありました。ドメインステータスが `Active` に変わったら、正しい URL を取得して IdP を変更できます。

URL を取得するには、ドメインを選択してから、[Actions] (アクション)、[Edit security configuration] (セキュリティ設定の編集) の順に選択します。 OpenSearch Dashboards/Kibana の SAML 認証では、正しいサービスプロバイダーエンティティ ID と SSO URLsを見つけることができます。値をコピーし、それらを使用してアイデンティティプロバイダーを設定することで、ステップ 2 で指定した一時的な URL を置き換えます。

ステップ 7: SAML ユーザーをロールにマップする

ドメインのステータスがアクティブで、IdP が正しく設定されると、 OpenSearch Dashboards に移動します。

- SP 開始の URL を選択した場合は、*`domain-endpoint`*`/_dashboards` に移動します。特定のテナントに直接ログインするには、URL に `?security_tenant=`*`tenant-name`* を追加できます。
- IdP 開始の URL を選択した場合は、ID プロバイダーのアプリケーションディレクトリに移動します。

どちらの場合も、SAML マスターユーザーまたは SAML マスターバックエンドロールに属しているユーザーとしてログインします。ステップ 7 からの例を続けるには、`jdoe`、または `admins` グループのメンバーとしてログインします。

OpenSearch Dashboards がロードされたら、セキュリティ、ロール を選択します。次に、ロールをマッピングして、他のユーザーが OpenSearch Dashboards にアクセスできるようにします。

例えば、信頼できる同僚 `jroe` を `all_access` および `security_manager` ロールにマッピングします。また、バックエンドロール `analysts` を `readall` および `opensearch_dashboards_user` ロールにマッピングすることもできます。

OpenSearch Dashboards ではなく API を使用する場合は、次のサンプルリクエストを参照してください。

```
PATCH _plugins/_security/api/rolesmapping
[
```

```
  {
    "op": "add", "path": "/security_manager", "value": { "users": ["master-user",
 "jdoe", "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/all_access", "value": { "users": ["master-user", "jdoe",
 "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/readall", "value": { "backend_roles": ["analysts"] }
  },
  {
    "op": "add", "path": "/opensearch_dashboards_user", "value": { "backend_roles":
 ["analysts"] }
  }
]
```

SP 開始認証と IdP 開始認証両方の設定

SP 開始認証と IdP 開始認証の両方を設定する場合は、ID プロバイダーを通じて設定する必要があります。例えば、Okta では、次のステップを実行できます。

1. SAML アプリケーション内で、[General] (全般)、[SAML settings] (SAML 設定) に移動します。
2. [Single sign on URL] (シングルサインオン URL) で、IdP 開始 SSO URL を指定します。例えば `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs/idpinitiated` です。
3. [Allow this app to request other SSO URLs] (このアプリが他の SSO URL をリクエストすることを許可) を有効にします。
4. [Requestable SSO URLs] (リクエスト可能な SSO URL) で、1 つ以上の SP 開始 SSO URL を追加します。例えば `https://search-domain-hash/_dashboards/_opendistro/_security/saml/acs` です。

SAML 認証の設定 (AWS CLI)

次の AWS CLI コマンドは、既存のドメインで OpenSearch Dashboards の SAML 認証を有効にします。

```
aws opensearch update-domain-config \
  --domain-name my-domain \
```

```
  --advanced-security-options '{"SAMLOptions":{"Enabled":true,"MasterUserName":"my-
idp-user","MasterBackendRole":"my-idp-group-or-role","Idp":{"EntityId":"entity-
id","MetadataContent":"metadata-content-with-quotes-escaped"},"RolesKey":"optional-
roles-key","SessionTimeoutMinutes":180,"SubjectKey":"optional-subject-key"}}'
```

メタデータ XML では、すべての引用符と改行文字をエスケープする必要があります。例えば、<KeyDescriptor use="signing"> および改行の代わりに <KeyDescriptor use=\"signing\">\n を使用します。の使用の詳細については AWS CLI、AWS CLI 「コマンドリファレンス」を参照してください。

SAML 認証の設定 (設定 API)

設定 API に対する次のリクエストは、既存のドメインで OpenSearch Dashboards の SAML 認証を有効にします。

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "AdvancedSecurityOptions": {
    "SAMLOptions": {
      "Enabled": true,
      "MasterUserName": "my-idp-user",
      "MasterBackendRole": "my-idp-group-or-role",
      "Idp": {
        "EntityId": "entity-id",
        "MetadataContent": "metadata-content-with-quotes-escaped"
      },
      "RolesKey": "optional-roles-key",
      "SessionTimeoutMinutes": 180,
      "SubjectKey": "optional-subject-key"
    }
  }
}
```

メタデータ XML では、すべての引用符と改行文字をエスケープする必要があります。例えば、<KeyDescriptor use="signing"> および改行の代わりに <KeyDescriptor use=\"signing\">\n を使用します。設定 API の使用の詳細については、OpenSearch 「サービス API リファレンス」を参照してください。

SAML のトラブルシューティング

エラー	詳細
リクエスト: '*/some/path* ' は許可されていません。	正しい SSO URL (ステップ 3) を ID プロバイダーに提供したことを確認します。
SAML を有効にするには、有効な ID プロバイダーメタデータドキュメントを指定してください。	IdP メタデータファイルは SAML 2.0 標準に準拠していません。検証ツールを使用してエラーをチェックします。
SAML 設定オプションは、コンソールでは表示されません。	最新のサービスソフトウェアに更新します。
SAML 設定エラー: SAML 設定を取得中に問題が発生しました。設定を確認してください。	この一般的なエラーは、さまざまな原因で発生することがあります。 • ID プロバイダーに正しい SP エンティティ ID と SSO URL が指定されていることを確認します。 • IdP メタデータファイルを再生成し、IdP エンティティ ID を確認します。更新されたメタデータを AWS コンソールに追加します。 • ドメインアクセスポリシーが OpenSearch Dashboards と へのアクセスを許可していることを確認します`_plugins/_security/*` 。一般的に、きめ細かなアクセスコントロールを使用するドメインではオープンアクセスポリシーを使用することをお勧めします。 • SAML の設定のステップについては、ID プロバイダーのドキュメントを参照してください。
ロールがありません: このユーザーには使用できるロールがありません。システム管理者に連絡してください。	認証に成功しましたが、ユーザーネームおよび SAML アサーションからのバックエンドロールはどのロールにもマッピングされていないため、許可がありません。これらのマッピングでは、大文字と小文字が区別されます。

エラー	詳細
	システム管理者は、SAML トレーサー などのツールを使用して SAML アサーションの内容を確認し、次のリクエストを使用してロールマッピングを確認できます。 `GET _plugins/_security/api/rolesmapping`
OpenSearch Dashboards にアクセスしようとすると、ブラウザは HTTP 500 エラーを継続的にリダイレクトまたは受信します。	このエラーは、SAML アサーションに合計約 1,500 文字の多数のロールが含まれている場合に発生します。例えば、平均の長さが 20 文字である 80 ロールを渡すと、ウェブブラウザの Cookie のサイズ制限を超える可能性があります。 OpenSearch バージョン 2.7 以降、SAML アサーションは最大 5000 文字のロールをサポートします。
ADFS からログアウトすることはできません。	ADFS では、 OpenSearch サービスがサポートしていないすべてのログアウトリクエストに署名する必要があります。IdP メタデータファイル<SingleLogoutService /> から を削除して、 OpenSearch サービスが独自の内部ログアウトメカニズムを使用するように強制します。
`Could not find entity descriptor for __PATH__.`	メタデータ XML で OpenSearch サービスに提供される IdP のエンティティ ID は、SAML レスポンスのエンティティ ID とは異なります。これを修正するには、それらが一致していることを確認してください。ドメインで [CW アプリケーションエラーログ] を有効にして、SAML 統合の問題をデバッグするためのエラーメッセージを見つけます。

エラー	詳細
`Signature validation failed. SAML response rejected.`	OpenSearch サービスは、メタデータ XML で提供される IdP の証明書を使用して SAML レスポンスの署名を検証できません。これは手動エラーであるか、または IdP が証明書をローテーションした可能性があります。を介して OpenSearch Service に提供されるメタデータ XML で IdP から最新の証明書を更新します AWS Management Console。
`__PATH__ is not a valid audience for this response.`	SAML レスポンスのオーディエンスフィールドがドメインエンドポイントと一致しません。このエラーを修正するには、ドメインエンドポイントと一致するように SP オーディエンスフィールドを更新します。カスタムエンドポイントを有効にしている場合は、オーディエンスフィールドがカスタムエンドポイントと一致する必要があります。ドメインで [CW アプリケーションエラーログ] を有効にして、SAML 統合の問題をデバッグするためのエラーメッセージを見つけます。
ブラウザは、レスポンスで `Invalid Request Id` ともに HTTP 400 エラーを受け取ります。	このエラーは通常、IdP 開始 URL を形式 *<DashboardsURL>* `/_opendistro/_security/saml/acs` で設定した場合に発生します。代わりに、URL を形式 *<DashboardsURL>* `/_opendistro/_security/saml/acs/idpinitiated` で設定します。

エラー	詳細
レスポンスは __PATH__ の代わりに __PATH__ で受信されました。	SAML レスポンスの宛先フィールドが次の URL 形式のいずれかと一致しません。 • *<DashboardsURL>* /_opendistro/_security/saml/acs • *<DashboardsURL>* /_opendistro/_security/saml/acs/idpinitiated . 使用するログインフロー (SP 開始または IdP 開始) に応じて、 OpenSearch URLs のいずれかに一致する送信先フィールドに を入力します。
レスポンスには InResponseTo 属性がありますが、InResponseTo 属性は想定されていません。	SP 開始ログインフローで IdP 開始 URL を使用しています。代わりに、SP 開始 URL を使用してください。

SAML 認証の無効化

OpenSearch Dashboards の SAML 認証を無効にするには (コンソール)

1. ドメインを選択し、[アクション] から [セキュリティ設定の編集] を選択します。
2. [SAML 認証を有効にする] のチェックを外します。
3. [変更の保存] を選択します。
4. ドメインの処理が終了したら、次のリクエストを用いてきめ細かなアクセスコントロールのロールマッピングを確認します。

```
GET _plugins/_security/api/rolesmapping
```

Dashboards の SAML 認証を無効にしても、SAML マスターユーザーネームおよび/または SAML マスターバックエンドロールのマッピングを削除しません。これらのマッピングを削除する場合は、内部ユーザーデータベース (有効な場合) を使用して Dashboards にログインするか、API を使用してそれらを削除します。

```
PUT _plugins/_security/api/rolesmapping/all_access
```

```
{
  "users": [
    "master-user"
  ]
}
```

OpenSearch Dashboards の Amazon Cognito 認証の設定

Amazon Cognito を使用して、OpenSearch Dashboards の Amazon OpenSearch Service のデフォルトインストールを認証し、保護することができます。Amazon Cognito 認証はオプションであり、OpenSearch または Elasticsearch 5.1 以降を使用しているドメインでのみ使用できます。Amazon Cognito 認証を設定しない場合でも、IP ベースのアクセスポリシーとプロキシサーバー、HTTP 基本認証、または SAML を使用して、Dashboards を保護することができます。

認証プロセスの多くは Amazon Cognito で発生しますが、このセクションでは、Amazon Cognito リソースを OpenSearch Service ドメインで使用するように設定するためのガイドラインと要件を示します。すべての Amazon Cognito リソースに標準の料金が適用されます。

> **Tip**
>
> OpenSearch Dashboards の Amazon Cognito 認証を使用するようにドメインを初めて設定するときは、コンソールを使用することをお勧めします。Amazon Cognito リソースは高度にカスタマイズ可能であり、コンソールにより、該当する機能を識別して理解しやすくなります。

トピック

- 一般的な設定の問題
- OpenSearch Dashboards の Amazon Cognito 認証を無効にする
- OpenSearch Dashboards の Amazon Cognito 認証を使用するドメインを削除する

前提条件

OpenSearch Dashboards の Amazon Cognito 認証を設定する前に、いくつかの前提条件を満たす必要があります。OpenSearch Service コンソールは、これらのリソースの作成の効率化を支援しますが、各リソースの目的を理解すると、設定およびトラブルシューティングに役立ちます。Dashboards の Amazon Cognito 認証では、次のリソースが必要になります。

- Amazon Cognito ユーザープール
- Amazon Cognito ID プール
- `AmazonOpenSearchServiceCognitoAccess` ポリシーがアタッチされた IAM ロール (`CognitoAccessForAmazonOpenSearch`)

> **Note**
>
> ユーザープールと ID プールは、同じ AWS リージョン に存在している必要があります。同じユーザープール、ID プール、および IAM ロールを使用して、Dashboards の Amazon Cognito 認証を複数の OpenSearch Service ドメインに追加できます。詳細については、「the section called “クォータ”」を参照してください。

ユーザープールについて

ユーザープールには、次の 2 つの主要な機能があります。それらは、ユーザーのディレクトリを作成および管理する機能と、ユーザーによるサインアップとログインを許可する機能です。ユーザープールを作成する手順については、Amazon Cognito デベロッパーガイドの「ユーザープールを作成する」を参照してください。

OpenSearch Service で使用するユーザープールを作成するときは、以下の点を考慮します。

- Amazon Cognito ユーザープールにはドメイン名が必要です。OpenSearch Service はこのドメイン名を使用して、Dashboards にアクセスするためのログインページにユーザーをリダイレクトします。ユーザープールには、ドメイン名以外に、デフォルト以外の設定は必要ありません。

- プールの必須の標準属性 (名前、誕生日、E メールアドレス、電話番号など) を指定する必要があります。ユーザープールの作成後にこれらの属性を変更することはできないため、この時点で関連するものを選択します。
- ユーザープールを作成する際に、ユーザーが自分のアカウントを作成できるかどうか、アカウントのパスワードの最低強度、多要素認証を有効にするかどうかを選択します。外部 ID プロバイダを使用する予定がある場合、これらの設定は重要ではありません。技術的には、ID プロバイダとしてユーザープールを有効にし、さらに外部 ID プロバイダを有効にすることができますが、ほとんどのユーザーはどちらか一方を希望します。

ユーザープール ID の形式は *region_ID* です。AWS CLI または AWS SDK を使用して OpenSearch Service を設定する計画の場合は、ID をメモしておきます。

ID プールについて

ID プールにより、ユーザーのログイン後に、権限の制限された一時的なロールをユーザーに割り当てることができます。ID プールを作成する手順については、Amazon Cognito デベロッパーガイドの「ID プール」を参照してください。OpenSearch Service で使用する ID プールを作成するときは、以下の点を考慮します。

- Amazon Cognito コンソールを使用する場合は、[認証されていない ID に対してアクセスを有効にする] チェックボックスをオンにして、ID プールを作成する必要があります。ID プールを作成し、OpenSearch Service ドメインを設定すると、Amazon Cognito によりこの設定は無効になります。
- ID プールに外部 ID プロバイダを追加する必要はありません。Amazon Cognito 認証を使用するように OpenSearch Service を設定すると、先ほど作成したユーザープールを使用するように ID プールが設定されます。
- ID プールを作成した後、認証されていない IAM ロールと認証された IAM ロールを選択する必要があります。これらのロールにより、ログイン前およびログイン後にユーザーに与えられるアクセスポリシーが指定されます。Amazon Cognito コンソールを使用する場合は、これらのロールを自動的に作成できます。認証されたロールを作成した後に、ARN を書き留めます。この形式は `arn:aws:iam::`*`123456789012`*`:role/Cognito_`*`identitypoolname`*`Auth_Role` です。

ID プール ID の形式は *region*`:`*ID*`-`*ID*`-`*ID*`-`*ID*`-`*ID* です。AWS CLI または AWS SDK を使用して OpenSearch Service を設定する計画の場合は、ID をメモしておきます。

CognitoAccessForAmazonOpenSearch ロールについて

OpenSearch Service には、Amazon Cognito ユーザーおよび ID プールを設定し、認証のためにそれらを使用するための許可が必要です。この目的のために、AWS 管理ポリシーの 1 つである AmazonOpenSearchServiceCognitoAccess を使用できます。AmazonESCognitoAccess は、サービスが Amazon OpenSearch Service に改名された際、AmazonOpenSearchServiceCognitoAccess に置き換えられた従来のポリシーです。どちらのポリシーも、Cognito 認証を有効にするために必要な最小限の Amazon Cognito の許可を提供します。ポリシーの JSON については、「IAM コンソール」を参照してください。

コンソールを使用して OpenSearch Service ドメインを作成または設定する場合、IAM ロールが作成され、その AmazonOpenSearchServiceCognitoAccess ポリシー (Elasticsearch ドメインの場合 AmazonESCognitoAccess ポリシー) がロールに添付されます。このロールのデフォルト名は CognitoAccessForAmazonOpenSearch です。

ロールの許可ポリシー AmazonOpenSearchServiceCognitoAccess および AmazonESCognitoAccess のいずれも、OpenSearch Service が、すべてのアイデンティティおよびユーザープールに対して以下のアクションを実行することを許可します。

- アクション: cognito-idp:DescribeUserPool
- アクション: cognito-idp:CreateUserPoolClient
- アクション: cognito-idp:DeleteUserPoolClient
- アクション: cognito-idp:UpdateUserPoolClient
- アクション: cognito-idp:DescribeUserPoolClient
- アクション: cognito-idp:AdminInitiateAuth
- アクション: cognito-idp:AdminUserGlobalSignOut
- アクション: cognito-idp:ListUserPoolClients
- アクション: cognito-identity:DescribeIdentityPool
- アクション: cognito-identity:SetIdentityPoolRoles
- アクション: cognito-identity:GetIdentityPoolRoles

AWS CLI またはいずれかの AWS SDK を使用する場合は、独自のロールを作成し、ポリシーをアタッチして、OpenSearch Service ドメインを設定するときに、このロールの ARN を指定する必要があります。ロールには、次の信頼関係が必要です。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "opensearchservice.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

手順については、IAM ユーザーガイドの「AWS のサービスに許可を委任するロールの作成」および「IAM ポリシーのアタッチおよびデタッチ」を参照してください。

Amazon Cognito 認証を使用するためのドメインの設定

前提条件を完了したら、Dashboards の Amazon Cognito を使用するように OpenSearch Service ドメインを設定できます。

> **Note**
>
> Amazon Cognito はすべての AWS リージョン で利用可能なわけではありません。サポートされているリージョンのリストについては、AWS リージョン および エンドポイント を参照してください。OpenSearch Service で使用する Amazon Cognito 用に同じリージョンを使用する必要はありません。

Amazon Cognito 認証の設定 (コンソール)

コンソールによって CognitoAccessForAmazonOpenSearch ロールが作成されるため、コンソールの設定エクスペリエンスが最もシンプルです。標準の OpenSearch Service のアクセス許可に加えて、コンソールで OpenSearch Dashboards の Amazon Cognito 認証を使用するドメインを作成するには、次のアクセス許可のセットが必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

アイデンティティ (ユーザー、ユーザーグループ、またはロール) にアクセス許可を追加するには、「IAM ID アクセス許可の追加 (コンソール)」を参照してください。

CognitoAccessForAmazonOpenSearch がすでに存在する場合、必要なアクセス許可は少なくなります。

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
```

```
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

Dashboards の Amazon Cognito 認証を設定するには (コンソール)

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home/) を開きます。
2. [ドメイン] の下で、設定するドメインを選択します。
3. [アクション] から [セキュリティ設定の編集] を選択します。
4. [Amazon Cognito 認証を有効にする] を選択します。
5. [Region] (リージョン) で、Amazon Cognito ユーザープールと ID プールが含まれている AWS リージョン を選択します。
6. [Cognito ユーザープール] で、ユーザープールを選択または作成します。ガイダンスについては、「the section called “ユーザープールについて”」を参照してください。
7. [Cognito ID プール] で、ID プールを選択または作成します。ガイダンスについては、「the section called “ID プールについて”」を参照してください。

   > **Note**
   >
   > [ユーザープールの作成] および [ID プールの作成] リンクから Amazon Cognito コンソールに移動したら、これらのリソースを手動で作成します。このプロセスは自動的には行われません。詳細については、「the section called “前提条件”」を参照してください。

8. [IAM ロール名] で、CognitoAccessForAmazonOpenSearch のデフォルト値を使用するか (推奨)、新しい名前を入力します。このロールの目的の詳細については、「the section called “CognitoAccessForAmazonOpenSearch ロールについて”」を参照してください。
9. [Save changes] (変更の保存) をクリックします。

ドメインによる処理が終了したら、追加の設定ステップについて「the section called “認証されたロールの許可”」および「the section called “ID プロバイダの設定”」を参照してください。

Amazon Cognito 認証の設定 (AWS CLI)

--cognito-options パラメータを使用して、OpenSearch Service ドメインを設定します。create-domain コマンドと update-domain-config コマンドの両方で次の構文が使用されます。

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

例

次の例では、CognitoAccessForAmazonOpenSearch ロールを使用して、Dashboards の Amazon Cognito 認証を有効にするドメインを us-east-1 リージョンで作成し、Cognito_Auth_Role へのドメインアクセスを提供します。

```
aws opensearch create-domain --domain-name my-domain --region us-east-1 --access-policies '{ "Version":"2012-10-17", "Statement":[{"Effect":"Allow","Principal":{"AWS":
 ["arn:aws:iam::123456789012:role/
Cognito_Auth_Role"]},"Action":"es:ESHttp*","Resource":"arn:aws:es:us-east-1:123456789012:domain/*" }]}' --engine-version "OpenSearch_1.0"
 --cluster-config InstanceType=m4.xlarge.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeSize=10 --cognito-options
 Enabled=true,UserPoolId="us-east-1_123456789",IdentityPoolId="us-east-1:12345678-1234-1234-1234-123456789012",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

ドメインによる処理が終了したら、追加の設定ステップについて「the section called “認証されたロールの許可”」および「the section called “ID プロバイダの設定”」を参照してください。

Amazon Cognito 認証の設定 (AWS SDK)

AWS SDK (Android および iOS SDK を除く) は、CreateDomain および UpdateDomainConfig オペレーションの CognitoOptions パラメータを含む、「Amazon OpenSearch Service API リファレンス」で定義されているすべてのオペレーションをサポートします。AWS SDK のインストールと使用の詳細については、「AWS Software Development Kits」を参照してください。

ドメインによる処理が終了したら、追加の設定ステップについて「the section called “認証されたロールの許可”」および「the section called “ID プロバイダの設定”」を参照してください。

認証されたロールの許可

デフォルトでは、「the section called “ID プールについて”」のガイドラインに従って設定された、認証された IAM ロールには、OpenSearch Dashboards にアクセスするために必要な権限がありません。追加のアクセス権限を持つロールを提供する必要があります。

> **Note**
>
> きめ細かなアクセスコントロールを設定しており、オープンまたは IP ベースのアクセスポリシーを使用している場合は、この手順をスキップできます。

これらのアクセス許可をアイデンティティベースのポリシーに含めることはできますが、認証されたユーザーがすべての OpenSearch Service ドメインにアクセスできるようにする場合を除いて、1 つのドメインにアタッチされたリソースベースのポリシーの方が適切です。

`Principal` には、the section called “ID プールについて” のガイドラインに従って設定した Cognito 認証ロールの ARN を指定します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":[
               "arn:aws:iam::123456789012:role/Cognito_identitypoolnameAuth_Role"
            ]
         },
         "Action":[
            "es:ESHttp*"
         ],
         "Resource":"arn:aws:es:region:123456789012:domain/domain-name/*"
      }
   ]
}
```

リソースベースのポリシーを OpenSearch Service ドメインに追加する手順については、「the section called “アクセスポリシーの設定”」を参照してください。

ID プロバイダの設定

Dashboards の Amazon Cognito 認証を使用するようにドメインを設定すると、OpenSearch Service はユーザープールにアプリクライアントを追加し、認証されたプロバイダとして ID プールにユーザープールを追加します。

> **Warning**
>
> アプリクライアントを名前変更または削除しないでください。

ユーザープールの設定方法に応じて、ユーザーアカウントを手動で作成する必要がある場合や、ユーザーが自分のアカウントを作成できる場合があります。これらの設定で問題ない場合は、それ以上アクションを実行する必要はありません。ただし、多くのユーザーが、外部 ID プロバイダの使用を希望します。

SAML 2.0 ID プロバイダを有効にするには、SAML メタデータドキュメントを提供する必要があります。Login with Amazon、Facebook、Google などのソーシャル ID プロバイダを有効にするには、これらのプロバイダからのアプリ ID とアプリシークレットが必要です。ID プロバイダの任意の組み合わせを有効にすることができます。

ユーザープールを設定する最も簡単な方法は、Amazon Cognito コンソールを使用することです。手順については、Amazon Cognito デベロッパーガイドの「ユーザープールのフェデレーションを使用する」および「ユーザープールアプリの ID プロバイダ設定を指定する」を参照してください。

(オプション) きめ細かなアクセスの設定

デフォルトの ID プール設定により、同じ IAM ロール (`Cognito_identitypoolAuth_Role`) が、ログインするすべてのユーザーが割り当てられることにお気づきかもしれません。これは、すべてのユーザーが同じ AWS リソースにアクセスできることを意味します。Amazon Cognito とともにきめ細かなアクセスコントロールを使用する場合、例えば、組織のアナリストが複数のインデックスに対して読み取り専用アクセスを持つようにする一方で、デベロッパーはすべてのインデックスに対して書き込みアクセスを持つようにする場合、次の 2 つのオプションがあります。

- ユーザーグループを作成し、ユーザーの認証トークンに基づいて IAM ロールを選択するように ID プロバイダを設定する (推奨)。
- 1 つ以上のルールに基づいて IAM ロールを選択するように ID プロバイダを設定する。

きめ細かなアクセスコントロールを含むチュートリアルについては、「the section called “チュートリアル: Cognito 認証によるきめ細かなアクセスコントロール”」を参照してください。

> **Important**
>
> デフォルトロールと同様に、Amazon Cognito は追加の各ロールの信頼関係に含まれている必要があります。詳細については、Amazon Cognito デベロッパーガイドの「ロールマッピング用のロールの作成」を参照してください。

ユーザーグループとトークン

ユーザーグループを作成するときは、グループのメンバー用の IAM ロールを選択します。グループの作成の詳細については、Amazon Cognito デベロッパーガイドの「ユーザーグループ」を参照してください。

1 つ以上のユーザーグループを作成した後、ID プールのデフォルトロールではなく、グループのロールをユーザーに割り当てるように認証プロバイダを設定できます。[トークンから選択する] を選択してから、[デフォルトの認証されたロールを使用する] または [拒否] を選択して、ID プールでのグループの一部ではないユーザーの処理方法を指定します。

ルール

基本的に、ルールは Amazon Cognito が順番に評価する一連の `if` ステートメントです。例えば、ユーザーの E メールアドレスに `@corporate` が含まれる場合、Amazon Cognito はそのユーザーに `Role_A` を割り当てます。ユーザーの E メールアドレスに `@subsidiary` が含まれる場合、そのユーザーに `Role_B` が割り当てられます。それ以外の場合は、デフォルトの認証されたロールがユーザーに割り当てられます。

詳細については、Amazon Cognito デベロッパーガイドの「ルールベースのマッピングを使用してユーザーにロールを割り当てる」を参照してください。

(オプション) サインインページのカスタマイズ

Amazon Cognito コンソールを使用して、カスタムロゴをアップロードし、サインインページで CSS の変更を行うことができます。手順および CSS プロパティの一覧については、Amazon Cognito デベロッパーガイドの「ユーザープールのアプリ UI カスタマイズ設定の指定」を参照してください。

(オプション) アドバンストセキュリティを設定する

Amazon Cognito ユーザープールでは、多要素認証、侵害された認証情報の確認、およびアダプティブ認証などのアドバンストセキュリティ機能がサポートされています。詳細については、Amazon Cognito デベロッパーガイドの「セキュリティの管理」を参照してください。

テスト

設定に問題がなければ、ユーザーエクスペリエンスが期待どおりであることを確認します。

OpenSearch Dashboards にアクセスするには

1. ウェブブラウザで `https://opensearch-domain/_dashboards` にアクセスします。特定のテナントに直接ログインするには、URL に `?security_tenant=tenant-name` を追加します。
2. 任意の認証情報を使用してサインインします。
3. OpenSearch Dashboards がロードされたら、少なくとも 1 つのインデックスパターンを設定します。Dashboards は、そのパターンを使用して、どのインデックスを分析するかを特定します。「*」と入力し、[次のステップ]、[Create index pattern (インデックスパターンの作成)] の順に選択します。
4. データを検索または調査するには、[検出] を選択します。

このプロセスのいずれかのステップが失敗した場合は、トラブルシューティング情報について、「the section called “一般的な設定の問題”」を参照してください。

クォータ

Amazon Cognito には、そのリソースの多くでソフト制限があります。多数の OpenSearch Service ドメインに対して Dashboards 認証を有効にする場合は、必要に応じて Amazon Cognito におけるクォータと制限の引き上げのリクエストについて確認してください。

各 OpenSearch Service ドメインでは、アプリクライアントがユーザープールに追加されます。これにより、認証プロバイダが ID プールに追加されます。10 を超えるドメインに対して OpenSearch Dashboards 認証を有効にすると、「ID プールあたりの Amazon Cognito の最大ユーザープールプロバイダ数」の制限が適用されることがあります。制限を超えた場合、Dashboards の Amazon Cognito 認証を使用する設定を試みている OpenSearch Service ドメインが、処理中の設定状態でスタックする可能性があります。

一般的な設定の問題

一般的な設定の問題と解決策を、以下の表に示します。

OpenSearch Service の設定

問題	解決策
`OpenSearch Service can't create the role` (コンソール)	適切な IAM アクセス許可がありません。「the section called “Amazon Cognito 認証の設定 (コンソール)”」で指定されたアクセス許可を追加します。
`User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonOpenSearch` (コンソール)	CognitoAccessforAmazonOpenSearch ロールの `iam:PassRole` 許可がありません。次のポリシーをアカウントにアタッチします: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:PassRole" ], "Resource": "arn:aws:iam:: 123456789012:role/service-role/CognitoAccessForAmazonOpenSearch " } ] }` または、`IAMFullAccess` ポリシーをアタッチできます。
`User is not authorized to perform: cognito-identity:ListIdentityPools on resource`	Amazon Cognito の読み取り許可がありません。`AmazonCognitoReadOnly` ポリシーをアカウントにアタッチします。
`An error occurred (ValidationException) when calling`	OpenSearch Service が `CognitoAccessForAmazonOpenSearch` ロールの信頼関係で指定されて

問題	解決策
`the CreateDomain operation : OpenSearch Service must be allowed to use the passed role`	いません。ロールで、「the section called “CognitoAccessForAmazonOpenSearch ロールについて”」で指定された信頼関係が使用されていることを確認します。または、コンソールを使用して Amazon Cognito 認証を設定します。コンソールによってロールが作成されます。
`An error occurred (ValidationException) when calling the CreateDomain operation : User is not authorized to perform: cognito-idp: action on resource: user pool`	`--cognito-options` で指定されたロールに、Amazon Cognito にアクセスする許可がありません。このロールに、AWS が管理する `AmazonOpenSearchServiceCognitoAccess` ポリシーがアタッチされていることを確認します。または、コンソールを使用して Amazon Cognito 認証を設定します。コンソールによってロールが作成されます。
`An error occurred (ValidationException) when calling the CreateDomain operation : User pool does not exist`	OpenSearch Service はユーザープールを見つけることができません。ユーザープールを作成し、正しい ID が設定されていることを確認します。ID を見つけるには、Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用できます。 `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateDomain operation : IdentityPool not found`	OpenSearch Service は ID プールを見つけることができません。ユーザープールを作成し、正しい ID が設定されていることを確認します。ID を見つけるには、Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用できます。 `aws cognito-identity list-identity-pools --max-results 60 --region region`

問題	解決策
`An error occurred (ValidationException) when calling the CreateDomain operation: Domain needs to be specified for user pool`	ユーザープールにドメイン名がありません。Amazon Cognito コンソールまたは次の AWS CLI コマンドを使用して、ドメイン名を設定できます。 `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

OpenSearch Dashboards へのアクセス

問題	解決策
目的の ID プロバイダがログインページに表示されない。	「the section called “ID プロバイダの設定”」で指定したように、OpenSearch Service アプリクライアントで ID プロバイダが有効になっていることを確認します。
ログインページが、組織に関連付けられているように見えない。	「the section called “(オプション) サインインページのカスタマイズ”」を参照してください。
ログイン認証情報が受け入れられない。	「the section called “ID プロバイダの設定”」で指定したように ID プロバイダを設定したことを確認します。 ID プロバイダとしてユーザープールを使用している場合は、アカウントが Amazon Cognito コンソールに存在することを確認します。
OpenSearch Dashboards がまったくロードされないか、正しく動作しない。	Amazon Cognito の認証されたロールでは、ドメイン (/*) が Dashboards にアクセスして使用するためには、`es:ESHttp*` の許可が必要です。「the section called “認証されたロールの許可”」で指定したように、アクセスポリシーを追加したことを確認します。
あるタブで OpenSearch Dashboards からサインアウトすると、残りのタブには更新トークンが取り消された	Amazon Cognito 認証使用時に OpenSearch Dashboards セッションからサインアウトすると、OpenSearch Service で AdminUserGlobalSignOut オペレーション

問題	解決策
ことを示すメッセージが表示されます。	が実行されて、すべてのアクティブな OpenSearch Dashboards セッションからサインアウトさせられます。
`Invalid identity pool configuration. Check assigned IAM roles for this pool.`	Amazon Cognito には、認証されたユーザーに代わって IAM ロールを引き受ける許可がありません。ロールの信頼関係を変更して、以下を含めます。 `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Federated": "cognito-identity.amazonaws.com" }, "Action": "sts:AssumeRoleWithWebIdentity", "Condition": { "StringEquals": { "cognito-identity.amazonaws.com:aud": " identity-pool-id " }, "ForAnyValue:StringLike": { "cognito-identity.amazonaws.com:amr": "authenticated" } } }] }`
`Token is not from a supported provider of this identity pool.`	ユーザープールからアプリクライアントを削除すると、この一般的でないエラーが発生する可能性があります。新しいブラウザセッションで Dashboards を開いてみます。

OpenSearch Dashboards の Amazon Cognito 認証を無効にする

Dashboards の Amazon Cognito 認証を無効にするには、次の手順を使用します。

Dashboards の Amazon Cognito 認証を無効にするには (コンソール)

1. Amazon OpenSearch Service コンソール (https://console.aws.amazon.com/aos/home/) を開きます。
2. [Domains] (ドメイン) で、設定するドメインを選択します。
3. [アクション] から [セキュリティ設定の編集] を選択します。
4. [Amazon Cognito 認証を有効にする] を選択解除します。
5. [Save changes] (変更の保存) をクリックします。

> **Important**
>
> Amazon Cognito ユーザープールと ID プールが不要になった場合は、それらを削除します。削除しなかった場合は、利用料金が引き続き発生します。

OpenSearch Dashboards の Amazon Cognito 認証を使用するドメインを削除する

Dashboards の Amazon Cognito 認証を使用するドメインが [Processing] (処理中) の設定状態で止まらないようにするには、関連する Amazon Cognito ユーザープールと ID プールを削除する前に OpenSearch Service ドメインを削除します。

Amazon OpenSearch Service のサービスにリンクされたロールの使用

Amazon OpenSearch Service は AWS Identity and Access Management 、 (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、 OpenSearch サービスに直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは OpenSearch 、サービスによって事前定義されており、ユーザーに代わってサービスから他の AWS のサービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、 OpenSearch サービスの設定が簡単になります。 OpenSearch サービスはサービスにリンクされたロールのアクセス許可を定義し、特に定義されている場合を除き、 OpenSearch サービスのみがそのロールを引き受けることができます。定義したアクセス許可には、信頼ポリシーと許可ポリ

シーが含まれます。この許可ポリシーを他の IAM エンティティにアタッチすることはできません。サービスにリンクされたロールとアクセス許可ポリシーの更新については、「Amazon OpenSearch Service のドキュメント履歴」を参照してください。

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。

トピック

- サービスにリンクされたロールを使用して VPC ドメインを作成する
- サービスにリンクされたロールを使用した OpenSearch サーバーレスコレクションの作成
- サービスにリンクされたロールを使用した OpenSearch 取り込みパイプラインの作成

サービスにリンクされたロールを使用して VPC ドメインを作成する

Amazon OpenSearch Service は AWS Identity and Access Management 、 (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、 OpenSearch サービスに直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは OpenSearch 、サービスによって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

OpenSearch サービスは、 という名前のサービスにリンクされたロールを使用します。これによりAWSServiceRoleForAmazonOpenSearchService、ロールがドメインの VPC アクセスを有効にするために必要な最小限の Amazon EC2 および Elastic Load Balancing アクセス許可が提供されます。

従来の Elasticsearch ロール

Amazon OpenSearch Service は、 というサービスにリンクされたロールを使用しますAWSServiceRoleForAmazonOpenSearchService。アカウントには、AWSServiceRoleForAmazonElasticsearchService と呼ばれるサービスにリンクされたロールも含まれることがあります。これは、非推奨の Elasticsearch API エンドポイントと連携します。

レガシー Elasticsearch ロールがアカウントに存在しない場合、 OpenSearch サービスが OpenSearch ドメインを初めて作成するときに、新しい OpenSearch サービスにリンクされたロールを自動的に作成します。そうでない場合、アカウントでは Elasticsearch ロールが引き続き使用さ

れます。この自動作成を成功させるためには、iam:CreateServiceLinkedRole アクションへのアクセス許可が必要です。

アクセス許可

AWSServiceRoleForAmazonOpenSearchService サービスにリンクされたロールは、ロールの引き受けについて以下のサービスを信頼します。

- opensearchservice.amazonaws.com

という名前のロール許可ポリシーAmazonOpenSearchServiceRolePolicyにより、 OpenSearch サービスは指定されたリソースに対して次のアクションを実行できます。

- アクション: * 上で acm:DescribeCertificate
- アクション: * 上で cloudwatch:PutMetricData
- アクション: * 上で ec2:CreateNetworkInterface
- アクション: * 上で ec2:DeleteNetworkInterface
- アクション: * 上で ec2:DescribeNetworkInterfaces
- アクション: * 上で ec2:ModifyNetworkInterfaceAttribute
- アクション: * 上で ec2:DescribeSecurityGroups
- アクション: * 上で ec2:DescribeSubnets
- アクション: * 上で ec2:DescribeVpcs
- アクション: すべてのネットワークインターフェースと VPC エンドポイントで ec2:CreateTags
- アクション: * 上で ec2:DescribeTags
- アクション: すべての VPC、セキュリティグループ、サブネット、およびルートテーブル、ならびにリクエストにタグ OpenSearchManaged=true が含まれる場合、すべての VPC エンドポイントで ec2:CreateVpcEndpoint
- アクション: すべての VPC、セキュリティグループ、サブネット、およびルートテーブル、ならびにリクエストにタグ OpenSearchManaged=true が含まれる場合、すべての VPC エンドポイントで ec2:ModifyVpcEndpoint
- アクション: リクエストにタグ OpenSearchManaged=true が含まれる場合、すべてのエンドポイントで ec2:DeleteVpcEndpoints
- アクション: * 上で ec2:AssignIpv6Addresses
- アクション: * 上で ec2:UnAssignIpv6Addresses

- アクション: * 上で elasticloadbalancing:AddListenerCertificates
- アクション: elasticloadbalancing:RemoveListenerCertificates 上で *

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、権限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「Service-linked role permissions」を参照してください。

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。を使用して VPC 対応ドメインを作成すると AWS Management Console、 OpenSearch サービスにリンクされたロールが自動的に作成されます。この自動作成を成功させるためには、iam:CreateServiceLinkedRole アクションへのアクセス許可が必要です。

サービスにリンクされたロールは、IAM コンソール、IAM CLI、または IAM API を使用して手動で作成することもできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの作成」を参照してください。

サービスにリンクされたロールを編集する

OpenSearch サービスでは、AWSServiceRoleForAmazonOpenSearchService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すると、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 を使用したロール記述の編集はできます。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。

サービスリンクロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、モニタリングや保守が積極的に行われていない未使用のエンティティを排除できます。ただし、手動で削除する前に、サービスリンクロールをクリーンアップする必要があります。

サービスにリンクされたロールのクリーンアップ

IAM を使用してサービスにリンクされたロールを削除するには、まずそのロールにアクティブなセッションがないことを確認し、そのロールで使用されているリソースをすべて削除する必要があります。

サービスにリンクされたロールにアクティブなセッションがあるかどうかを、IAM コンソールで確認するには

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/iam/ で IAM コンソールを開きます。
2. IAM コンソールのナビゲーションペインで [ロール] を選択します。次に、AWSServiceRoleForAmazonOpenSearchService ロールの名前 (チェックボックスではありません) を選択します。
3. 選択したロールの [概要] ページで、[アクセスアドバイザー] タブを選択します。
4. アクセスアドバイザー タブで、サービスにリンクされたロールの最新のアクティビティを確認します。

> **Note**
>
> OpenSearch サービスがAWSServiceRoleForAmazonOpenSearchServiceロールを使用しているかどうか不明な場合は、ロールを削除できます。サービスでロールが使用されている場合、削除は失敗し、ロールが使用されているリソースを表示できます。ロールが使用されている場合は、ロールを削除する前、またはロールを使用しているリソースを削除する前にセッションが終了するのを待つ必要があります。サービスにリンクされたロールのセッションを取り消すことはできません。

サービスにリンクされたロールの手動削除

IAM コンソール、API、または AWS CLI からサービスにリンクされたロールを削除します。手順については、IAM ユーザーガイド の「サービスにリンクされたロールの削除」を参照してください。

サービスにリンクされたロールを使用した OpenSearch サーバーレスコレクションの作成

OpenSearch サーバーレスは AWS Identity and Access Management (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、 OpenSearch サービスに直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは OpenSearch 、サービスによって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

OpenSearch Serverless は、 という名前のサービスにリンクされたロールを使用します。このロールはAWSServiceRoleForAmazonOpenSearchServerless、サーバーレス関連の CloudWatch

メトリクスをアカウントに公開するために必要なアクセス許可を提供します。に関連付けられたロールのアクセス許可ポリシーの名前 AWSServiceRoleForAmazonOpenSearchServerless は ですAmazonOpenSearchServerlessServiceRolePolicy。ポリシーの詳細については、「 マネージドポリシーリファレンスガイドAmazonOpenSearchServerlessServiceRolePolicy」の「」を参照してください。 AWS

OpenSearch Serverless のサービスにリンクされたロールのアクセス許可

OpenSearch Serverless は、 という名前のサービスにリンクされたロールを使用します。これにより AWSServiceRoleForAmazonOpenSearchServerless、 OpenSearch サーバーレスはユーザーに代わって AWS のサービスを呼び出すことができます。

AWSServiceRoleForAmazonOpenSearchServerless 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。

- `observability.aoss.amazonaws.com`

という名前のロールアクセス許可ポリシーAmazonOpenSearchServerlessServiceRolePolicyにより、 OpenSearch サーバーレスは指定されたリソースに対して次のアクションを実行できます。

- アクション: すべての AWS リソースcloudwatch:PutMetricDataで

> **Note**
>
> ポリシーには条件キー が含まれています。つまり{"StringEquals": {"cloudwatch:namespace": "AWS/AOSS"}}、サービスにリンクされたロールは、メトリクスデータを AWS/AOSS CloudWatch名前空間にのみ送信できます。

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「Service-linked role permissions」を参照してください。

OpenSearch Serverless のサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。 AWS Management Console、、AWS CLIまたは AWS API で OpenSearch Serverless コレクションを作成すると、 OpenSearch Serverless 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。

> **Note**
>
> コレクションを初めて作成するときは、ID ベースのポリシーで `iam:CreateServiceLinkedRole` を割り当てる必要があります。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。 OpenSearch サーバーレスコレクションを作成すると、 OpenSearch サーバーレスはサービスにリンクされたロールを再度作成します。

IAM コンソールを使用して、Amazon OpenSearch Serverless ユースケースでサービスにリンクされたロールを作成することもできます。 AWS CLI または AWS API で、サービス名を使用して`observability.aoss.amazonaws.com`サービスにリンクされたロールを作成します。

```
aws iam create-service-linked-role --aws-service-name
 "observability.aoss.amazonaws.com"
```

詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

OpenSearch Serverless のサービスにリンクされたロールの編集

OpenSearch Serverless では、 AWSServiceRoleForAmazonOpenSearchServerless 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すると、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 を使用したロール記述の編集はできます。詳細については、IAM ユーザーガイドの「サービスリンクロールの編集」を参照してください。

OpenSearch Serverless のサービスにリンクされたロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、積極的にモニタリングまたは保守されていない未使用のエンティティを保持しないようにできます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

を削除するには AWSServiceRoleForAmazonOpenSearchServerless、まず 内のすべての OpenSearch Serverless コレクションを削除する必要があります AWS アカウント。

> **Note**
>
> リソースを削除しようとしたときに OpenSearch Serverless がロールを使用している場合、削除が失敗する可能性があります。その場合は、数分待ってからオペレーションを再試行してください。

IAM を使用してサービスリンクロールを手動で削除するには

IAM コンソール、、または AWS API を使用して AWS CLI、サービスにリンクされたロールを削除します AWSServiceRoleForAmazonOpenSearchServerless。詳細については、IAM ユーザーガイドのサービスにリンクされたロールの削除を参照してください。

OpenSearch Serverless サービスにリンクされたロールでサポートされているリージョン

OpenSearch Serverless は、 OpenSearch Serverless が利用可能なすべてのリージョンで、AWSServiceRoleForAmazonOpenSearchServerless サービスにリンクされたロールの使用をサポートしています。サポートされているリージョンのリストについては、「」の「Amazon OpenSearch Serverless エンドポイントとクォータ」を参照してくださいAWS 全般のリファレンス。

サービスにリンクされたロールを使用した OpenSearch 取り込みパイプラインの作成

Amazon Ingestion OpenSearch は AWS Identity and Access Management 、 (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、取り込みに直接リンクされた一意のタイプの IAM OpenSearch ロールです。サービスにリンクされたロールは Ingestion OpenSearch によって事前定義されており、サービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

OpenSearch 取り込みでは、 という名前のサービスにリンクされたロールを使用します。ただしAWSServiceRoleForAmazonOpenSearchIngestionService、セルフマネージド VPC を使用する場合は、 という名前のサービスにリンクされたロールを使用しますAWSServiceRoleForOpensearchIngestionSelfManagedVpce。アタッチされたポリシーは、ロールがアカウントと Ingestion の間に Virtual Private Cloud (VPC) OpenSearch を作成し、 CloudWatchアカウントにメトリクスを発行するために必要なアクセス許可を提供します。

アクセス許可

AWSServiceRoleForAmazonOpenSearchIngestionService サービスにリンクされたロールは、ロールの引き受けについて以下のサービスを信頼します。

- osis.amazon.com

という名前のロールアクセス許可ポリシーAmazonOpenSearchIngestionServiceRolePolicy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することを Ingestion OpenSearch に許可します。

- アクション: * 上で ec2:DescribeSubnets
- アクション: * 上で ec2:DescribeSecurityGroups
- アクション: * 上で ec2:DeleteVpcEndpoints
- アクション: * 上で ec2:CreateVpcEndpoint
- アクション: * 上で ec2:DescribeVpcEndpoints
- アクション: arn:aws:ec2:*:*:network-interface/* 上で ec2:CreateTags
- アクション: cloudwatch:namespace": "AWS/OSIS" 上で cloudwatch:PutMetricData

AWSServiceRoleForOpensearchIngestionSelfManagedVpce サービスにリンクされたロールは、ロールの引き受けについて以下のサービスを信頼します。

- self-managed-vpce.osis.amazon.com

という名前のロールアクセス許可ポリシーOpenSearchIngestionSelfManagedVpcePolicy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することを Ingestion OpenSearch に許可します。

- アクション: * 上で ec2:DescribeSubnets
- アクション: * 上で ec2:DescribeSecurityGroups
- アクション: * 上で ec2:DescribeVpcEndpoints
- アクション: cloudwatch:PutMetricData 上で cloudwatch:namespace": "AWS/OSIS"

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、権限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「Service-linked role permissions」を参照してください。

OpenSearch 取り込み用のサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。 AWS Management Console、、 AWS CLIまたは AWS API OpenSearch で Ingestion パイプラインを作成すると、Ingestion OpenSearch 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。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ方法でアカウントにロールを再作成できます。Ingestion OpenSearch パイプラインを作成すると、Ingestion OpenSearch によってサービスにリンクされたロールが再度作成されます。

OpenSearch Ingestion のサービスにリンクされたロールの編集

OpenSearch 取り込みでは、`AWSServiceRoleForAmazonOpenSearchIngestionService`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「サービスリンクロールの編集」を参照してください。

OpenSearch Ingestion のサービスにリンクされたロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

サービスリンクロールのクリーンアップ

IAM を使用してサービスリンクロールを削除するには、最初に、そのロールで使用されているリソースをすべて削除する必要があります。

> **Note**
>
> リソースを削除しようとしたときに Ingestion OpenSearch がロールを使用している場合、削除が失敗する可能性があります。失敗した場合は、数分待ってから操作を再試行してください。

AWSServiceRoleForAmazonOpenSearchIngestionService または **AWSServiceRoleForOpensearchIngestionSelfManagedVpce**ロールが使用する OpenSearch 取り込みリソースを削除するには

1. Amazon OpenSearch Service コンソールに移動し、取り込み を選択します。
2. すべてのパイプラインを削除します。手順については、「the section called “パイプラインの削除”」を参照してください。

OpenSearch Ingestion のサービスにリンクされたロールを削除する

OpenSearch 取り込みコンソールを使用して、サービスにリンクされたロールを削除できます。

サービスにリンクされたロールを削除するには (コンソール)

1. [IAM console] (IAM コンソール) に入ります。
2. ロール を選択し、 AWSServiceRoleForAmazonOpenSearchIngestionServiceまたは AWSServiceRoleForOpensearchIngestionSelfManagedVpceロールを検索します。
3. ロールを選択し、[Delete] (削除) をクリックします。

Amazon OpenSearch Service のサンプルコード

この章には、以下の Amazon OpenSearch Service で使用するための一般的なサンプルコードが含まれます: さまざまなプログラミング言語での HTTP リクエストの署名、HTTP リクエストの本文の圧縮、ドメインを作成するための AWS SDK の使用。

トピック

Elasticech クライアントの互換性

最新バージョンの Elasticsearch クライアントには、人為的に互換性を損なうライセンスチェックやバージョンチェックが含まれる場合があります。次のテーブルに、OpenSearch Service との最大限の互換性で使用するための、これらのクライアントのバージョンに関する推奨事項を示します。

> **Important**
>
> これらのクライアントバージョンは古くて、Log4j を含む最新の依存関係では更新されません。可能であれば、クライアントの OpenSearch バージョンを使用することを強くお勧めします。

クライアント	推奨バージョン
Java 低レベル REST クライアント	7.13.4
Java 高レベル REST クライアント	7.13.4
Python Elasticsear	7.13.4
Ruby Elasticsearch クライアント	7.13.3
Node.js Elasticsearch クライアント	7.13.0

Amazon OpenSearch サービスでの HTTP リクエストの圧縮

Amazon OpenSearch サービスドメイン内の HTTP リクエストとレスポンスは、gzip 圧縮を使用して圧縮できます。gzip 圧縮を使用すると、ドキュメントのサイズを縮小し、帯域幅の使用率とレイテンシーを低減できるため、転送速度が向上します。

Gzip 圧縮は、Elasticsearch 6.0 OpenSearch 以降を実行しているすべてのドメインでサポートされています。 OpenSearch 一部のクライアントには gzip 圧縮のサポートが組み込まれており、多くのプログラミング言語にはプロセスを簡略化するライブラリがあります。

gzip 圧縮を有効にする

OpenSearch 同様の設定と混同しないでください。 OpenSearch Service `http_compression.enabled` 固有のもので、ドメインで gzip 圧縮を有効または無効にします。Elasticsearch 7 OpenSearch を実行しているドメイン。 x では gzip 圧縮がデフォルトで有効になっているのに対し、Elasticsearch 6 を実行しているドメインでは gzip 圧縮が有効になっています。 x ではデフォルトで無効になっています。

gzip 圧縮を有効にするには、次のリクエストを送信します。

```
PUT _cluster/settings
{
  "persistent" : {
    "http_compression.enabled": true
  }
}
```

`_cluster/settings` へのリクエストは圧縮解除する必要があるので、クラスター設定を更新するために別のクライアントまたは標準の HTTP 要求を使用する必要がある場合があります。

gzip 圧縮が正常に有効になったことを確認するには、以下のリクエストを送信してください。

```
GET _cluster/settings?include_defaults=true
```

レスポンスに次の設定が表示されることを確認してください。

```
...
"http_compression": {
```

```
  "enabled": "true"
}
...
```

必要なヘッダー

gzip 圧縮されたリクエストボディを含める場合は、標準の Content-Type: application/json ヘッダーを保持し、Content-Encoding: gzip ヘッダーを追加します。gzip 圧縮レスポンスを受け入れるには、Accept-Encoding: gzip ヘッダーも同様に追加します。 OpenSearch クライアントが gzip 圧縮をサポートしている場合、これらのヘッダーは自動的に含まれる可能性があります。

サンプルコード (Python 3)

次のサンプルでは、opensearch-py を使用して圧縮を実行し、リクエストを送信します。このコードは、IAM 認証情報を使用してリクエストに署名します。

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = '' # e.g. my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Create the client.
search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    http_compress = True, # enables gzip compression for request bodies
    connection_class = RequestsHttpConnection
)

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
```

```
  "year": "2011"
}

# Send the request.
print(search.index(index='movies', id='1', body=document, refresh=True))

# print(search.index(index='movies', doc_type='_doc', id='1', body=document,
 refresh=True))
```

あるいは、適切なヘッダーを指定し、リクエストボディを自身で圧縮し、リクエストのような標準 HTTP ライブラリを使用します。このコードは、HTTP 基本認証情報を使用してリクエストに署名します。これは、きめ細かなアクセスコントロール使用する場合に、ドメインがサポートする場合があります。

```
import requests
import gzip
import json

base_url = '' # The domain with https:// and a trailing slash. For example, https://my-
test-domain.us-east-1.es.amazonaws.com/
auth = ('master-user', 'master-user-password') # For testing only. Don't store
 credentials in code.

headers = {'Accept-Encoding': 'gzip', 'Content-Type': 'application/json',
           'Content-Encoding': 'gzip'}

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Compress the document.
compressed_document = gzip.compress(json.dumps(document).encode())

# Send the request.
path = 'movies/_doc?refresh=true'
url = base_url + path
response = requests.post(url, auth=auth, headers=headers, data=compressed_document)
print(response.status_code)
print(response.text)
```

Amazon OpenSearch Service を操作するための AWS SDKの使用

このセクションには、AWS SDK を使用して Amazon OpenSearch Service 設定 API を操作する方法の例が含まれています。これらのコードサンプルは、OpenSearch Service ドメインの作成、更新、削除方法を示しています。

Java

このセクションには、AWS SDK for Java のバージョン 1 および 2 の例が含まれています。

Version 2

この例では、OpenSearchClientBuilder コンストラクタを使用して、AWS SDK for Java のバージョン 2 から OpenSearch ドメインの作成、その設定の更新、および削除を行います。`waitForDomainProcessing` への呼び出しのコメント (および `deleteDomain` への呼び出しのコメント) を削除し、ドメインをオンラインにして使用できるようにします。

```
package com.example.samples;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.opensearch.OpenSearchClient;
import software.amazon.awssdk.services.opensearch.model.ClusterConfig;
import software.amazon.awssdk.services.opensearch.model.EBSOptions;
import software.amazon.awssdk.services.opensearch.model.CognitoOptions;
import software.amazon.awssdk.services.opensearch.model.NodeToNodeEncryptionOptions;
import software.amazon.awssdk.services.opensearch.model.CreateDomainRequest;
import software.amazon.awssdk.services.opensearch.model.CreateDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainRequest;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainResponse;
import software.amazon.awssdk.services.opensearch.model.OpenSearchException;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */
```

```
public class OpenSearchSample {

    public static void main(String[] args) {

     String domainName = "my-test-domain";

     // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.

        OpenSearchClient client = OpenSearchClient.builder()
          // Unnecessary, but lets you use a region different than your default.
          .region(Region.US_EAST_1)
          // Unnecessary, but if desired, you can use a different provider chain.
          .credentialsProvider(DefaultCredentialsProvider.create())
                    .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
 Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */

    public static void createDomain(OpenSearchClient client, String domainName) {

     // Create the request and set the desired configuration options

        try {
```

```
            ClusterConfig clusterConfig = ClusterConfig.builder()
                    .dedicatedMasterEnabled(true)
                    .dedicatedMasterCount(3)
                    // Small, inexpensive instance types for testing. Not
 recommended for production.
                    .dedicatedMasterType("t2.small.search")
                    .instanceType("t2.small.search")
                    .instanceCount(5)
                    .build();

            // Many instance types require EBS storage.
            EBSOptions ebsOptions = EBSOptions.builder()
                    .ebsEnabled(true)
                    .volumeSize(10)
                    .volumeType("gp2")
                    .build();

            NodeToNodeEncryptionOptions encryptionOptions =
 NodeToNodeEncryptionOptions.builder()
                    .enabled(true)
                    .build();

            CreateDomainRequest createRequest = CreateDomainRequest.builder()
                    .domainName(domainName)
                    .engineVersion("OpenSearch_1.0")
                    .clusterConfig(clusterConfig)
                    .ebsOptions(ebsOptions)
                    .nodeToNodeEncryptionOptions(encryptionOptions)
                    // You can uncomment this line and add your account ID, a
 username, and the
                    // domain name to add an access policy.
                    // .accessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")
                    .build();

            // Make the request.
            System.out.println("Sending domain creation request...");
            CreateDomainResponse createResponse =
 client.createDomain(createRequest);
            System.out.println("Domain status:
 "+createResponse.domainStatus().toString());
```

```
            System.out.println("Domain ID:
"+createResponse.domainStatus().domainId());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
     * specified options. Some options require other Amazon Web Services resources,
such as an
     * Amazon Cognito user pool and identity pool, whereas others require just an
     * instance type or instance count.
     *
     * @param client
     *             The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *             The name of the domain to update
     */

    public static void updateDomain(OpenSearchClient client, String domainName) {

     // Updates the domain to use three data instances instead of five.
        // You can uncomment the Cognito line and fill in the strings to enable
Cognito
        // authentication for OpenSearch Dashboards.

        try {

            ClusterConfig clusterConfig = ClusterConfig.builder()
                    .instanceCount(5)
                    .build();

            CognitoOptions cognitoOptions = CognitoOptions.builder()
                    .enabled(true)
                    .userPoolId("user-pool-id")
                    .identityPoolId("identity-pool-id")
                    .roleArn("role-arn")
                    .build();
```

```
            UpdateDomainConfigRequest updateRequest =
UpdateDomainConfigRequest.builder()
                    .domainName(domainName)
                    .clusterConfig(clusterConfig)
                    //.cognitoOptions(cognitoOptions)
                    .build();

            System.out.println("Sending domain update request...");
            UpdateDomainConfigResponse updateResponse =
client.updateDomainConfig(updateRequest);
            System.out.println("Domain config:
"+updateResponse.domainConfig().toString());


        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *             The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *             The name of the domain that you want to delete
     */

    public static void deleteDomain(OpenSearchClient client, String domainName) {

        try {

            DeleteDomainRequest deleteRequest = DeleteDomainRequest.builder()
                    .domainName(domainName)
                    .build();

            System.out.println("Sending domain deletion request...");
            DeleteDomainResponse deleteResponse =
client.deleteDomain(deleteRequest);
            System.out.println("Domain status: "+deleteResponse.toString());
```

```
        } catch (OpenSearchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    /**
     * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
     * take a similar amount of time. This method checks every 15 seconds and
finishes only when
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */

    public static void waitForDomainProcessing(OpenSearchClient client, String
domainName) {
        // Create a new request to check the domain status.
        DescribeDomainRequest describeRequest = DescribeDomainRequest.builder()
                .domainName(domainName)
                .build();

        // Every 15 seconds, check whether the domain is processing.
        DescribeDomainResponse describeResponse =
client.describeDomain(describeRequest);
        while (describeResponse.domainStatus().processing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse = client.describeDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
changes for your domain.");
```

```
        System.out.println("Domain description: "+describeResponse.toString());
    }
}
```

Version 1

この例では AWSElasticsearchClientBuilder を使用して、AWS SDK for Java のバージョン 1 からレガシーm p Elasticsearch ドメインの作成、その設定の更新、および削除を行います。waitForDomainProcessing への呼び出しのコメント (および deleteDomain への呼び出しのコメント) を削除し、ドメインをオンラインにして使用できるようにします。

```
package com.amazonaws.samples;

import java.util.concurrent.TimeUnit;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticsearch.AWSElasticsearch;
import com.amazonaws.services.elasticsearch.AWSElasticsearchClientBuilder;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainResult;
import
 com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.EBSOptions;
import com.amazonaws.services.elasticsearch.model.ElasticsearchClusterConfig;
import com.amazonaws.services.elasticsearch.model.ResourceNotFoundException;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigRequest;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigResult;
import com.amazonaws.services.elasticsearch.model.VolumeType;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {
```

```
        final String domainName = "my-test-domain";

        // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.
        final AWSElasticsearch client = AWSElasticsearchClientBuilder
                .standard()
                // Unnecessary, but lets you use a region different than your
default.
                .withRegion(Regions.US_WEST_2)
                // Unnecessary, but if desired, you can use a different provider
chain.
                .withCredentials(new DefaultAWSCredentialsProviderChain())
                .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */
    private static void createDomain(final AWSElasticsearch client, final String
domainName) {

        // Create the request and set the desired configuration options
        CreateElasticsearchDomainRequest createRequest = new
CreateElasticsearchDomainRequest()
                .withDomainName(domainName)
                .withElasticsearchVersion("7.10")
```

```
                .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withDedicatedMasterEnabled(true)
                        .withDedicatedMasterCount(3)
                        // Small, inexpensive instance types for testing. Not
 recommended for production
                        // domains.
                        .withDedicatedMasterType("t2.small.elasticsearch")
                        .withInstanceType("t2.small.elasticsearch")
                        .withInstanceCount(5))
                // Many instance types require EBS storage.
                .withEBSOptions(new EBSOptions()
                        .withEBSEnabled(true)
                        .withVolumeSize(10)
                        .withVolumeType(VolumeType.Gp2));
                // You can uncomment this line and add your account ID, a username,
 and the
                // domain name to add an access policy.
                // .withAccessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")


        // Make the request.
        System.out.println("Sending domain creation request...");
        CreateElasticsearchDomainResult createResponse =
 client.createElasticsearchDomain(createRequest);
        System.out.println("Domain creation response from Amazon OpenSearch
 Service:");
        System.out.println(createResponse.getDomainStatus().toString());
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
     * specified options. Some options require other Amazon Web Services resources,
 such as an
     * Amazon Cognito user pool and identity pool, whereas others require just an
     * instance type or instance count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain to update
     */
```

```
   private static void updateDomain(final AWSElasticsearch client, final String
domainName) {
       try {
           // Updates the domain to use three data instances instead of five.
           // You can uncomment the Cognito lines and fill in the strings to enable
Cognito
           // authentication for OpenSearch Dashboards.
           final UpdateElasticsearchDomainConfigRequest updateRequest = new
UpdateElasticsearchDomainConfigRequest()
                   .withDomainName(domainName)
                   // .withCognitoOptions(new CognitoOptions()
                           // .withEnabled(true)
                           // .withUserPoolId("user-pool-id")
                           // .withIdentityPoolId("identity-pool-id")
                           // .withRoleArn("role-arn")
                   .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                           .withInstanceCount(3));

           System.out.println("Sending domain update request...");
           final UpdateElasticsearchDomainConfigResult updateResponse = client
                   .updateElasticsearchDomainConfig(updateRequest);
           System.out.println("Domain update response from Amazon OpenSearch
Service:");
           System.out.println(updateResponse.toString());
       } catch (ResourceNotFoundException e) {
           System.out.println("Domain not found. Please check the domain name.");
       }
   }

   /**
    * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
    * several minutes.
    *
    * @param client
    *             The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *             The name of the domain that you want to delete
    */
   private static void deleteDomain(final AWSElasticsearch client, final String
domainName) {
       try {
           final DeleteElasticsearchDomainRequest deleteRequest = new
DeleteElasticsearchDomainRequest()
                   .withDomainName(domainName);
```

```
            System.out.println("Sending domain deletion request...");
            final DeleteElasticsearchDomainResult deleteResponse =
client.deleteElasticsearchDomain(deleteRequest);
            System.out.println("Domain deletion response from Amazon OpenSearch
Service:");
            System.out.println(deleteResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
     * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
     * take a similar amount of time. This method checks every 15 seconds and
finishes only when
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */
    private static void waitForDomainProcessing(final AWSElasticsearch client, final
String domainName) {
        // Create a new request to check the domain status.
        final DescribeElasticsearchDomainRequest describeRequest = new
DescribeElasticsearchDomainRequest()
                .withDomainName(domainName);

        // Every 15 seconds, check whether the domain is processing.
        DescribeElasticsearchDomainResult describeResponse =
client.describeElasticsearchDomain(describeRequest);
        while (describeResponse.getDomainStatus().isProcessing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse =
client.describeElasticsearchDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
```

```
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
        System.out.println("Domain description response from Amazon OpenSearch
 Service:");
        System.out.println(describeResponse.toString());
    }
}
```

Python

この例では、OpenSearchService低レベル Python クライアントを使用して、AWS SDK for Python (Boto)からドメインの作成、その設定の更新、および削除を行います。

```
import boto3
import botocore
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-west-2'
)

client = boto3.client('opensearch', config=my_config)

domainName = 'my-test-domain'  # The name of the domain


def createDomain(client, domainName):
    """Creates an Amazon OpenSearch Service domain with the specified options."""
    response = client.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_1.0',
        ClusterConfig={
```

```
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-west-2:123456789012:domain/my-test-domain/*
\"}]}",
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    print("Creating domain...")
    print(response)


def updateDomain(client, domainName):
    """Updates the domain to use three data nodes instead of five."""
    try:
        response = client.update_domain_config(
            DomainName=domainName,
            ClusterConfig={
                'InstanceCount': 3
            }
        )
        print('Sending domain update request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def deleteDomain(client, domainName):
```

```
    """Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes."""
    try:
        response = client.delete_domain(
            DomainName=domainName
        )
        print('Sending domain deletion request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def waitForDomainProcessing(client, domainName):
    """Waits for the domain to finish processing changes."""
    try:
        response = client.describe_domain(
            DomainName=domainName
        )
        # Every 15 seconds, check whether the domain is processing.
        while response["DomainStatus"]["Processing"] == True:
            print('Domain still processing...')
            time.sleep(15)
            response = client.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is available.
        print('Amazon OpenSearch Service has finished processing changes for your
 domain.')
        print('Domain description:')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def main():
    """Create a new domain, update its configuration, and delete it."""
```

```
    createDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    updateDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    deleteDomain(client, domainName)
```

ノード

この例では、SDK for JavaScript in Node.js のバージョン 3 OpenSearch クライアントを使用して、ドメインの作成、その設定の更新、および削除を行います。

```
var {
    OpenSearchClient,
    CreateDomainCommand,
    DescribeDomainCommand,
    UpdateDomainConfigCommand,
    DeleteDomainCommand
} = require("@aws-sdk/client-opensearch");
var sleep = require('sleep');

var client = new OpenSearchClient();

var domainName = 'my-test-domain'

// Create a new domain, update its configuration, and delete it.
createDomain(client, domainName)
waitForDomainProcessing(client, domainName)
updateDomain(client, domainName)
waitForDomainProcessing(client, domainName)
deleteDomain(client, domainName)

async function createDomain(client, domainName) {
    // Creates an Amazon OpenSearch Service domain with the specified options.
    var command = new CreateDomainCommand({
        DomainName: domainName,
        EngineVersion: 'OpenSearch_1.0',
        ClusterConfig: {
        'InstanceType': 't2.small.search',
        'InstanceCount': 5,
        'DedicatedMasterEnabled': 'True',
        'DedicatedMasterType': 't2.small.search',
        'DedicatedMasterCount': 3
        },
```

```
        EBSOptions:{
            'EBSEnabled': 'True',
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies: "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/my-test-domain/*
\"}]}",
         NodeToNodeEncryptionOptions:{
            'Enabled': 'True'
        }
    });
    const response = await client.send(command);
    console.log("Creating domain...");
    console.log(response);
}

async function updateDomain(client, domainName) {
    // Updates the domain to use three data nodes instead of five.
    var command = new UpdateDomainConfigCommand({
        DomainName: domainName,
        ClusterConfig: {
        'InstanceCount': 3
        }
    });
    const response = await client.send(command);
    console.log('Sending domain update request...');
    console.log(response);
}

async function deleteDomain(client, domainName) {
    // Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes.
    var command = new DeleteDomainCommand({
        DomainName: domainName
    });
    const response = await client.send(command);
    console.log('Sending domain deletion request...');
    console.log(response);
}

async function waitForDomainProcessing(client, domainName) {
    // Waits for the domain to finish processing changes.
```

```
    try {
        var command = new DescribeDomainCommand({
            DomainName: domainName
        });
        var response = await client.send(command);

        while (response.DomainStatus.Processing == true) {
            console.log('Domain still processing...')
            await sleep(15000) // Wait for 15 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        // Once we exit the loop, the domain is available.
        console.log('Amazon OpenSearch Service has finished processing changes for your
 domain.');
        console.log('Domain description:');
        console.log(response);

    } catch (error) {
        if (error.name === 'ResourceNotFoundException') {
            console.log('Domain not found. Please check the domain name.');
            }
    };
}
```

Amazon OpenSearch Service でのデータのインデックス作成

Amazon OpenSearch Service は REST API を使用するため、ドキュメントのインデックス作成には多数のメソッドがあります。curl などの標準クライアントを使用することも、HTTP リクエストの送信が可能な任意のプログラミング言語を使用することもできます。操作プロセスをさらに簡素化するために、 OpenSearch Service には多くのプログラミング言語用のクライアントがあります。上級ユーザーは、直接「the section called “ストリーミングデータを OpenSearch サービスにロードする”」に進むことができます。

Amazon OpenSearch Ingestion を使用して、 OpenSearch サービス内に構築されたフルマネージドデータコレクターであるデータを取り込むことを強くお勧めします。詳細については、「Amazon OpenSearch Ingestion」を参照してください。

インデックス作成の概要については、 OpenSearch ドキュメントを参照してください。

インデックスの命名制限

OpenSearch サービスインデックスには、次の命名制限があります。

- すべての文字を小文字にする必要があります。
- インデックスの名前を _ または - から始めることはできません。
- インデックス名にスペース、カンマ、:、"、*、+、/、\、|、?、#、>、< を含めることはできません。

インデックス、タイプ、またはドキュメント ID 名に機密情報を含めないでください。 OpenSearch サービスはこれらの名前を Uniform Resource Identifiers (URIs) で使用します。サーバーおよびアプリケーションは、HTTP リクエストを頻繁に記録します。このため、URI に機密情報が含まれている場合、不要なデータ漏えいにつながる可能性があります。

```
2018-10-03T23:39:43 198.51.100.14 200 "GET https://opensearch-domain/dr-jane-doe/flu-patients-2018/202-555-0100/ HTTP/1.1"
```

関連する JSON ドキュメントを表示する許可がない場合でも、Dr. Doe の患者 (電話番号は 202-555-0100) が 2018 年にインフルエンザにかかったことを、この架空の口グ行から推測することが可能です。

OpenSearch サービスがインデックス名 (などmy-index-12.34.56.78.91) で実際の IP アドレスまたは認識された IP アドレスを検出すると、その IP アドレスはマスクされます。_cat/indices を呼び出すと、以下のレスポンスが生成されます。

```
green open my-index-x.x.x.x.91    soY19tBERoKo71WcEScidw 5 1 0 0   2kb  1kb
```

不必要な混乱を避けるため、インデックス名には IP アドレスを含めないでください。

レスポンスサイズの削減

_index および _bulk API のレスポンスには、多くの情報が含まれています。この情報は、リクエストのトラブルシューティングや、再試行ロジックの実装には役立ちますが、帯域幅を広く使用する場合があります。この例では、32 バイトのドキュメントのインデックスを作成すると、レスポンスは 339 バイト (ヘッダーを含む) になります。

```
PUT opensearch-domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

レスポンス

```
{
  "_index": "more-movies",
  "_type": "_doc",
  "_id": "1",
  "_version": 4,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 3,
  "_primary_term": 1
}
```

このレスポンスサイズは一見小さく見えますが、1 日あたり 1,000,000 ドキュメント (1 秒あたり約 11.5 ドキュメントのインデックス) を作成した場合、339 バイト/1 レスポンスは、1 か月あたり 10.17 GB のダウンロードトラフィックになります。

データ転送コストが懸念される場合は、 filter_pathパラメータを使用して OpenSearch サービスレスポンスのサイズを小さくしますが、失敗したリクエストを特定または再試行するために必要なフィールドを除外しないように注意してください。このフィールドはクライアントによって異なります。filter_path パラメータはすべての OpenSearch Service REST APIs、 _indexや APIs など、頻繁に呼び出す _bulk APIsでは特に便利です。

```
PUT opensearch-domain/more-movies/_doc/1?filter_path=result,_shards.total
{"title": "Back to the Future"}
```

レスポンス

```
{
  "result": "updated",
  "_shards": {
    "total": 2
  }
}
```

フィールドを含める代わりに、- プリフィックスを使用してフィールドを除外することができます。filter_path は、ワイルドカードもサポートしています。

```
POST opensearch-domain/_bulk?filter_path=-took,-items.index._*
{ "index": { "_index": "more-movies", "_id": "1" } }
{"title": "Back to the Future"}
{ "index": { "_index": "more-movies", "_id": "2" } }
{"title": "Spirited Away"}
```

レスポンス

```
{
  "errors": false,
  "items": [
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    },
    {
      "index": {
```

```
        "result": "updated",
        "status": 200
      }
    }
  ]
}
```

インデックスコーデック

インデックスコーデックは、インデックスに格納されているフィールドを圧縮してディスクに保存する方法を決定します。インデックスコーデックは、圧縮アルゴリズムを指定する静的 `index.codec` 設定によって制御されます。この設定は、インデックスのシャードサイズと操作のパフォーマンスに影響します。

サポートされているコーデックとそのパフォーマンス特性のリストについては、 OpenSearch ドキュメントの「サポートされているコーデック」を参照してください。

インデックスコーデックを選択するときは、次の点を考慮してください。

- 既存のインデックスのコーデック設定を変更する手間を省くために、新しいコーデック設定を使用する前に、本番環境以外の環境で代表的なワークロードをテストしてください。詳細については、「インデックスコーデックを変更する」を参照してください。
- k-NN または Security Analytics インデックスに Zstandard 圧縮コーデック (`"index.codec": "zstd"` または `"index.codec": "zstd_no_dict"`) を使用することはできません。

Amazon OpenSearch Service へのストリーミングデータのロード

OpenSearch Ingestion を使用すると、サードパーティーのソリューションを使用しなくても、ストリーミングデータを Amazon OpenSearch Service ドメインに直接ロードできます。 OpenSearch Ingestion にデータを送信するには、データプロデューサーを設定し、指定したドメインまたはコレクションにデータを自動的に配信します。取り込みを開始するには、 OpenSearch 「」を参照してくださいthe section called “チュートリアル: コレクションにデータを取り込む”。

OpenSearch サービスのサポートが組み込まれている Amazon Data Firehose や Amazon CloudWatch Logs など、他のソースを使用してストリーミングデータをロードすることもできます。Amazon S3、Amazon Kinesis Data Streams、Amazon DynamoDB などの他のものでは、イベントハンドラとして AWS Lambda 関数を使用します。Lambda 関数は、データを処理してドメインにストリーミングすることで、新しいデータに応答します。

> **Note**
>
> Lambda では、いくつかの一般的なプログラミング言語がサポートされ、ほとんどの AWS リージョンで利用できます。詳細については、「AWS Lambda デベロッパーガイド」の「Lambda の開始方法」、および「AWS 全般のリファレンス」の「AWS サービスエンドポイント」を参照してください。

トピック

取り込みからストリーミングデータをロード OpenSearch する

Amazon OpenSearch Ingestion を使用して、 OpenSearch サービスドメインにデータをロードできます。 OpenSearch Ingestion にデータを送信するようにデータプロデューサーを設定すると、指定したコレクションにデータが自動的に配信されます。データを配信する前にデータを変換するように OpenSearch Ingestion を設定することもできます。詳細については、「Amazon OpenSearch Ingestion」を参照してください。

Amazon S3 からストリーミングデータをロードする

Lambda を使用して、Amazon S3 から OpenSearch サービスドメインにデータを送信できます。S3 バケットに到着する新しいデータにより、Lambda へのイベント通知がトリガーされた後、インデックス作成を実行するカスタムコードが実行されます。

このデータのストリーミング方法には非常に柔軟性があります。オブジェクトメタデータのインデックスを作成したり、オブジェクトがプレーンテキストの場合は、オブジェクト本文の要素を解析してインデックス作成したりすることができます。このセクションでは、正規表現を使用してログファイルを解析し、一致をインデックス作成するシンプルな Python サンプルコードがあります。

前提条件

続行する前に、以下のリソースが必要です。

前提条件	説明
Amazon S3 バケット	詳細については、Amazon Simple Storage Service ユーザーガイドの「最初の S3 バケットを作成する」を参照してください。バケットは、OpenSearch サービスドメインと同じリージョンに存在する必要があります。
OpenSearch サービスドメイン	Lambda 関数により処理された後のデータのターゲット。詳細については、「the section called “ OpenSearch サービスドメインの作成”」を参照してください。

Lambda デプロイパッケージを作成する

デプロイパッケージは、コードとその依存関係を含む ZIP または JAR ファイルです。このセクションには、Python サンプルコードがあります。他のプログラミング言語については、AWS Lambda デベロッパーガイドの「Lambda デプロイパッケージ」を参照してください。

1. ディレクトリを作成します。このサンプルでは、名前 `s3-to-opensearch` を使用します。
2. `sample.py` という名前のディレクトリ内にファイルを作成します。

```
import boto3
import re
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-s3-index'
datatype = '_doc'
```

```
url = host + '/' + index + '/' + datatype

headers = { "Content-Type": "application/json" }

s3 = boto3.client('s3')

# Regular expressions used to parse some simple log lines
ip_pattern = re.compile('(\d+\.\d+\.\d+\.\d+)')
time_pattern = re.compile('\[(\d+\/\w\w\w\/\d\d\d\d:\d\d:\d\d:\d\d\s-\d\d\d\d)\]')
message_pattern = re.compile('\"(.+)\"')

# Lambda execution starts here
def handler(event, context):
    for record in event['Records']:

        # Get the bucket name and key for the new file
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Get, read, and split the file into lines
        obj = s3.get_object(Bucket=bucket, Key=key)
        body = obj['Body'].read()
        lines = body.splitlines()

        # Match the regular expressions to each line and index the JSON
        for line in lines:
            line = line.decode("utf-8")
            ip = ip_pattern.search(line).group(1)
            timestamp = time_pattern.search(line).group(1)
            message = message_pattern.search(line).group(1)

            document = { "ip": ip, "timestamp": timestamp, "message": message }
            r = requests.post(url, auth=awsauth, json=document, headers=headers)
```

region と host の変数を編集します。

3. まだ持っていない場合、pip をインストールしてから、依存関係を新しい package ディレクトリにインストールします。

```
cd s3-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

すべての Lambda 実行環境に Boto3 がインストールされているため、デプロイパッケージに含める必要はありません。

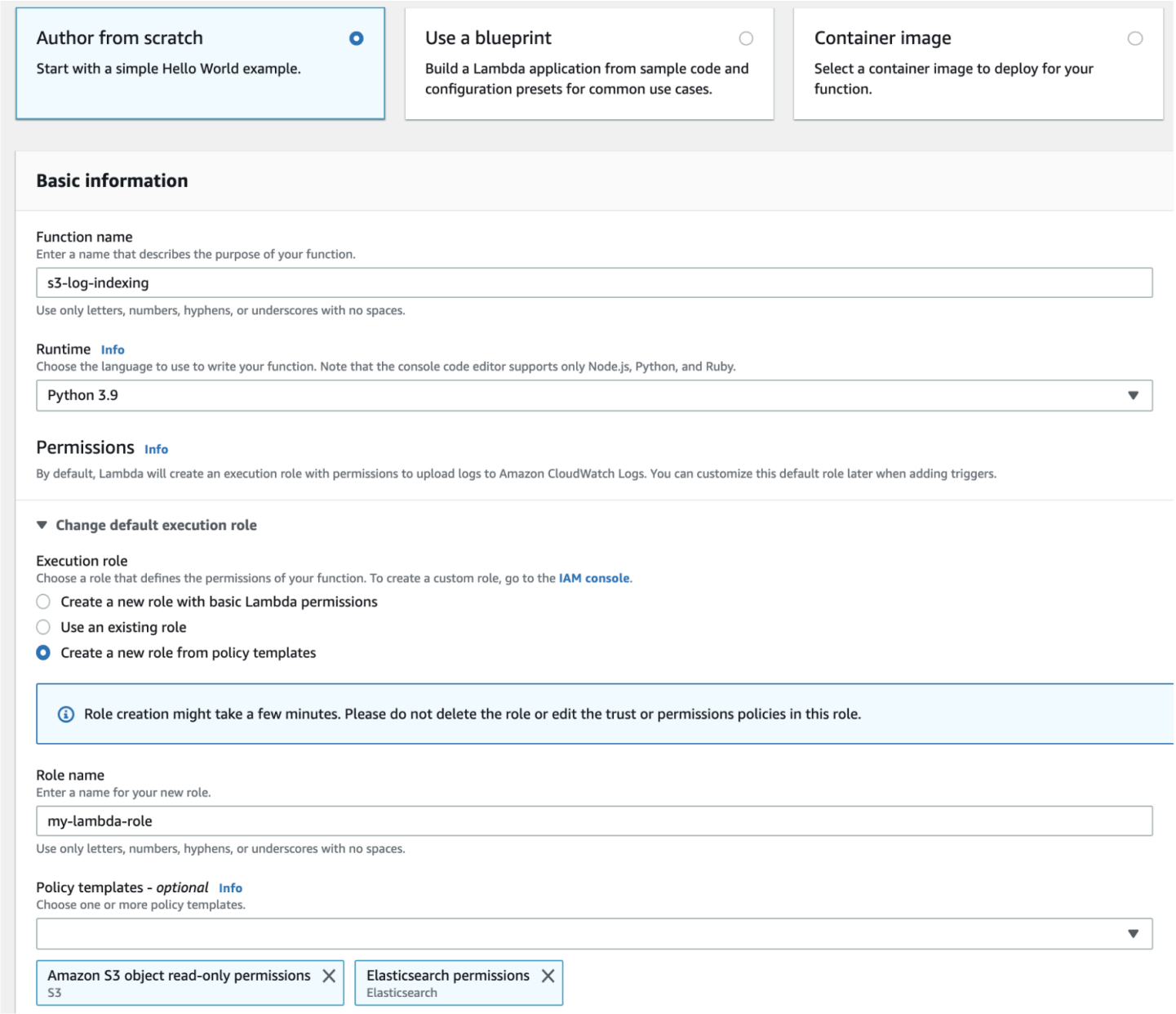
4. アプリケーションコードや相互依存性をパッケージ化します。

```
cd package
zip -r ../lambda.zip .

cd ..
zip -g lambda.zip sample.py
```

Lambda 関数を作成する

デプロイパッケージを作成すると、Lambda 関数を作成できます。関数を作成するとき、名前、ランタイム (たとえば、Python 3.8)、IAM ロールを選択します。IAM ロールにより、関数の許可が定義されます。詳細な手順については、AWS Lambda デベロッパーガイドの「コンソールで Lambda 関数を作成する」を参照してください。

この例では、コンソールを使用していることを前提としています。次のスクリーンショットに示すように、Python 3.9 と、S3 読み取りアクセス許可と OpenSearch サービス書き込みアクセス許可を持つロールを選択します。



関数を作成した後、トリガーを追加する必要があります。この例では、S3 バケットにログファイルが到着するたびにコードを実行します。

1. [トリガーの追加] を選択し、[S3] を選択します。
2. バケットを選択します。
3. [イベントタイプ] で、[PUT] を選択します。
4. [プレフィックス] に `logs/` と入力します。
5. [サフィックス] では、`.log` と入力します。
6. 再帰呼び出しの警告を確認し、[追加] を選択します。

最後に、デプロイパッケージをアップロードすることができます。

1. [~ からアップロード] および [.zip ファイルをアップロード] を選択してから、デプロイパッケージをアップロードするプロンプトに従います。
2. アップロードが終了したら、[ランタイム設定] を編集し、[ハンドラ] を `sample.handler` に変更します。この設定により、トリガー後に実行するファイル (`sample.py`) およびメソッド (`handler`) が Lambda に通知されます。

この時点で、ログファイル用のバケット、ログファイルがバケットに追加されるたびに実行される関数、解析とインデックス作成を実行するコード、検索と可視化のための OpenSearch サービスドメインなどのリソースの完全なセットがあります。

Lambda 関数をテストする

関数を作成した後、Amazon S3 バケットにファイルをアップロードしてテストできます。次のサンプルログの行を使用して、`sample.log` という名前のファイルを作成します。

```
12.345.678.90 - [10/Oct/2000:13:55:36 -0700] "PUT /some-file.jpg"
12.345.678.91 - [10/Oct/2000:14:56:14 -0700] "GET /some-file.jpg"
```

S3 バケットの `logs` フォルダにファイルをアップロードします。手順については、Amazon Simple Storage Service ユーザーガイドの「バケットにオブジェクトをアップロードする」を参照してください。

次に、 OpenSearch サービスコンソールまたは OpenSearch Dashboards を使用して、`lambda-s3-index`インデックスに 2 つのドキュメントが含まれていることを確認します。標準検索リクエストを行うこともできます。

```
GET https://domain-name/lambda-s3-index/_search?pretty
{
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vTYXaWIBJWV_TTkEuSDg",
        "_score" : 1.0,
```

```
        "_source" : {
          "ip" : "12.345.678.91",
          "message" : "GET /some-file.jpg",
          "timestamp" : "10/Oct/2000:14:56:14 -0700"
        }
      },
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vjYmaWIBJWV_TTkEuCAB",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.90",
          "message" : "PUT /some-file.jpg",
          "timestamp" : "10/Oct/2000:13:55:36 -0700"
        }
      }
    ]
  }
}
```

Amazon Kinesis Data Streams からストリーミングデータをロードする

Kinesis Data Streams から OpenSearch サービスにストリーミングデータをロードできます。データストリームに到達する新しいデータによって、Lambda へのイベント通知がトリガーされ、インデックス作成を実行するカスタムコードが実行されます。このセクションには、いくつかの簡単な Python サンプルコードがあります。

前提条件

続行する前に、以下のリソースが必要です。

前提条件	説明
Amazon Kinesis Data Stream	Lambda 関数のイベントソース。詳細については、「Kinesis Data Streams」を参照してください。
OpenSearch サービスドメイン	Lambda 関数により処理された後のデータのターゲット。詳細については、「the section called “ OpenSearch サービスドメインの作成”」を参照してください。

前提条件	説明

IAM ロール	このロールには、次のような基本的な OpenSearch サービス、Kinesis、および Lambda アクセス許可が必要です。
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "es:ESHttpPost", "es:ESHttpPut", "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents", "kinesis:GetShardIterator", "kinesis:GetRecords", "kinesis:DescribeStream", "kinesis:ListStreams" ], "Resource": "*" } ] }`
	
	ロールには、次の信頼関係が必要です。
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }`

前提条件	説明
	詳細については、IAM ユーザーガイドの「IAM ロールの作成」を参照してください。

Lambda 関数を作成する

「the section called “Lambda デプロイパッケージを作成する”」の手順に従いますが、kinesis-to-opensearch という名前のディレクトリを作成し、sample.py に次のコードを使用します。

```
import base64
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-kine-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        id = record['eventID']
        timestamp = record['kinesis']['approximateArrivalTimestamp']

        # Kinesis data is base64-encoded, so decode here
        message = base64.b64decode(record['kinesis']['data'])

        # Create the JSON document
        document = { "id": id, "timestamp": timestamp, "message": message }
        # Index the document
```

```
        r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return 'Processed ' + str(count) + ' items.'
```

region と host の変数を編集します。

まだ持っていない場合、pip をインストールしてから、次のコマンドを使用して、依存関係をインストールします。

```
cd kinesis-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

次に、「the section called “Lambda 関数を作成する”」の手順に従いますが、「the section called “前提条件”」の IAM ロールと、トリガーの以下の設定を指定します。

- [Kinesis ストリーム]: Kinesis ストリーム
- [バッチサイズ]: 100
- [開始位置]: 水平トリム

詳細については、Amazon Kinesis Data Streams デベロッパーガイドの「Amazon Kinesis Data Streams とは」を参照してください。

この時点で、Kinesis データストリーム、ストリームが新しいデータを受信してそのデータのインデックスを作成した後に実行される関数、検索と視覚化のための OpenSearch サービスドメインなどのリソースの完全なセットがあります。

Lambda 関数をテストする

関数を作成した後、 AWS CLIを使用してデータストリームに新しいレコードを追加することでテストできます。

```
aws kinesis put-record --stream-name test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

次に、 OpenSearch サービスコンソールまたは OpenSearch Dashboards を使用して、 にドキュメントlambda-kine-indexが含まれていることを確認します。以下のリクエストを使用することもできます。

```
GET https://domain-name/lambda-kine-index/_search
{
  "hits" : [
    {
      "_index": "lambda-kine-index",
      "_type": "_doc",
      "_id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42",
      "_score": 1,
      "_source": {
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42"
      }
    }
  ]
}
```

Amazon DynamoDB テーブルからストリーミングデータをロードする

を使用して AWS Lambda 、Amazon DynamoDB から OpenSearch サービスドメインにデータを送信できます。データテーブルに到着する新しいデータにより、Lambda へのイベント通知がトリガーされた後、インデックス作成を実行するカスタムコードが実行されます。

前提条件

続行する前に、以下のリソースが必要です。

前提条件	説明
DynamoDB テーブル	このテーブルにはソースデータが含まれています。詳細については、Amazon DynamoDB デベロッパーガイドの「DynamoDB テーブルの基本的なオペレーション」を参照してください。 テーブルは OpenSearch サービスドメインと同じリージョンに存在し、ストリームが新しいイメージ に設定されている必要があります。詳細については、「ストリームの有効化」を参照してください。

前提条件	説明
OpenSearch サービスドメイン	Lambda 関数により処理された後のデータのターゲット。詳細については、「the section called “ OpenSearch サービスドメインの作成”」を参照してください。

前提条件	説明
IAM ロール	このロールには、次のような基本的な OpenSearch サービス、DynamoDB、および Lambda 実行アクセス許可が必要です。 `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["es:ESHttpPost", "es:ESHttpPut", "dynamodb:DescribeStream", "dynamodb:GetRecords", "dynamodb:GetShardIterator", "dynamodb:ListStreams", "logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents"], "Resource": "*" }] }`  ロールには、次の信頼関係が必要です。  `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" }] }`

前提条件	説明
	詳細については、IAM ユーザーガイドの「IAM ロールの作成」を参照してください。

Lambda 関数を作成する

「the section called “Lambda デプロイパッケージを作成する”」の手順に従いますが、ddb-to-opensearch という名前のディレクトリを作成し、sample.py に次のコードを使用します。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-east-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        # Get the primary key for use as the OpenSearch ID
        id = record['dynamodb']['Keys']['id']['S']

        if record['eventName'] == 'REMOVE':
            r = requests.delete(url + id, auth=awsauth)
        else:
            document = record['dynamodb']['NewImage']
            r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return str(count) + ' records processed.'
```

region と host の変数を編集します。

まだ持っていない場合、pip をインストールしてから、次のコマンドを使用して、依存関係をインストールします。

```
cd ddb-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

次に、「the section called “Lambda 関数を作成する”」の手順に従いますが、「the section called “前提条件”」の IAM ロールと、トリガーの以下の設定を指定します。

- [テーブル]: DynamoDB テーブル
- [バッチサイズ]: 100
- [開始位置]: 水平トリム

詳細については、Amazon DynamoDB デベロッパーガイドの「DynamoDB Streams と Lambda を用いた新しい項目の処理」を参照してください。

この時点で、ソースデータの DynamoDB テーブル、テーブルに対する変更の DynamoDB ストリーム、ソースデータの変更後に実行される関数、それらの変更のインデックス作成、検索と可視化のための OpenSearch サービスドメインなどのリソースの完全なセットがあります。

Lambda 関数をテストする

関数を作成した後、 AWS CLIを使用して DynamoDB テーブルに新しい項目を追加することでテストできます。

```
aws dynamodb put-item --table-name test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

次に、 OpenSearch サービスコンソールまたは OpenSearch Dashboards を使用して、 にドキュメントlambda-indexが含まれていることを確認します。以下のリクエストを使用することもできます。

```
GET https://domain-name/lambda-index/_doc/00001
{
```

```
    "_index": "lambda-index",
    "_type": "_doc",
    "_id": "00001",
    "_version": 1,
    "found": true,
    "_source": {
        "director": {
            "S": "Kevin Costner"
        },
        "id": {
            "S": "00001"
        },
        "title": {
            "S": "The Postman"
        }
    }
}
```

Amazon Data Firehose からストリーミングデータをロードする

Firehose は、配信先として OpenSearch サービスをサポートします。ストリーミングデータを OpenSearch サービスにロードする方法については、「Amazon Data Firehose デベロッパーガイド」の「Kinesis Data Firehose 配信ストリームの作成」および「送信先に OpenSearch サービスを選択する」を参照してください。

Service にデータをロードする前に OpenSearch 、データの変換を実行する必要がある場合があります。Lambda 関数を使用してこのタスクを実行する方法については、このガイドの「Amazon Kinesis Data Firehose データ変換」を参照してください。

配信ストリームを設定すると、Firehose には「ワンクリック」の IAM ロールがあり、Lambda を使用して OpenSearch サービスへのデータの送信、Amazon S3 上のデータのバックアップ、およびデータの変換に必要なリソースアクセスが許可されます。このようなロールを手動で作成する作業は複雑になるため、用意されているロールの使用をお勧めします。

Amazon からストリーミングデータをロードする CloudWatch

CloudWatch Logs サブスクリプションを使用して、 CloudWatch ログから OpenSearch サービスドメインにストリーミングデータをロードできます。Amazon CloudWatch サブスクリプションの詳細については、「サブスクリプションによるログデータのリアルタイム処理」を参照してください。設定情報については、「Amazon デベロッパーガイド」の「Amazon OpenSearch Service への CloudWatch ログデータのストリーミング」を参照してください。 CloudWatch

AWS IoTからストリーミングデータをロードする

ルール AWS IoT を使用して、 からデータを送信できます。 https://docs.aws.amazon.com/iot/latest/developerguide/iot-rules.html詳細については、「 AWS IoT デベロッパーガイド」の「 OpenSearchアクション」を参照してください。

Logstash を用いて Amazon OpenSearch Service へデータをロードする

Logstash のオープンソース版 (Logstash OSS) には、一括 API を使用して Amazon OpenSearch Service ドメインにデータをアップロードする便利な方法があります。このサービスは、Amazon S3 入力プラグインを含む、すべての標準 Logstash 入力プラグインをサポートします。OpenSearch Service は、基本認証と IAM 認証情報の両方をサポートする logstash-output-opensearch 出力プラグインをサポートします。プラグインはバージョン 8.1 以降の Logstash OSS で動作します。

構成

Logstash の設定は、ドメインが使用する認証のタイプによって異なります。

どの認証方法を使用していても、設定ファイルの出力セクションで `ecs_compatibility` を `disabled` に設定する必要があります。Logstash 8.0 では、デフォルトで ECS 互換モードですべてのプラグインが実行される画期的な変更が導入されました。レガシー動作を維持するには、デフォルト値をオーバーライドする必要があります。

きめ細かなアクセス制御の設定

OpenSearch Service ドメインで HTTP Basic 認証によるきめ細かなアクセスコントロールを使用する場合、設定は他の OpenSearch クラスターと同様です。この設定ファイルの例では、Filebeat のオープンソース版 (Filebeat OSS) から入力を受け取ります。

```
input {
  beats  {
    port => 5044
  }
}

output {
  opensearch {
    hosts       => "https://domain-endpoint:443"
```

```
    user         => "my-username"
    password     => "my-password"
    index        => "logstash-logs-%{+YYYY.MM.dd}"
    ecs_compatibility => disabled
    ssl_certificate_verification => false
  }
}
```

設定は Beats アプリケーションとユースケースによって異なりますが、Filebeat OSS の設定は次のようになります。

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /path/to/logs/dir/*.log
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.ilm.enabled: false
setup.ilm.check_exists: false
setup.template.settings:
  index.number_of_shards: 1
output.logstash:
  hosts: ["logstash-host:5044"]
```

IAM ロールの設定

ドメインで IAM ベースのドメインアクセスポリシーを使用する場合や IAM マスターユーザーによるきめ細かなアクセスコントロールを使用する場合、OpenSearch Service へのすべてのリクエストには IAM 認証情報を使用して署名する必要があります。次の ID ベースのポリシーは、ドメインのサブリソースへの、すべての HTTP リクエストを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/*"
```

```
    }
  ]
}
```

Logstash を設定するには、プラグインをその出力に使用するように、設定ファイルを変更します。この設定ファイルの例では、S3 バケット内のファイルから入力を受け取ります。

```
input {
  s3 {
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    hosts => ["domain-endpoint:443"]
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      }
      index  => "logstash-logs-%{+YYYY.MM.dd}"
      ecs_compatibility => disabled
  }
}
```

設定ファイル内で IAM 認証情報を提供したくない場合は、それらをエクスポート (または `aws configure` を実行) できます。

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN="your-session-token"
```

OpenSearch Service ドメインが VPC にある場合、Logstash OSS マシンは VPC に接続できる必要があり、また VPC セキュリティグループを介してドメインにアクセスできる必要があります。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

Amazon OpenSearch Service でのデータの検索

URI 検索やリクエストボディ検索など、Amazon OpenSearch Service でドキュメントを検索する一般的な方法がいくつかあります。 OpenSearch サービスは、カスタムパッケージ、SQL サポート、非同期検索など、検索エクスペリエンスを向上させる追加機能を提供します。包括的な OpenSearch 検索 API リファレンスについては、「」のOpenSearch ドキュメントを参照してください。

> **Note**
>
> 次のサンプルリクエストは OpenSearch APIs。一部のリクエストでは古い Elasticsearch バージョンを使用できない可能性があります。

トピック

URI 検索

Universal Resource Identifier (URI) 検索は、最もシンプルな検索方法です。URI 検索では、HTTP リクエストパラメータとしてクエリを指定します。

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

以下にレスポンスの例を示します。


```
{
  "took": 25,
  "timed_out": false,
  "_shards": {
    "total": 10,
    "successful": 10,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 85,
      "relation": "eq",
    },
    "max_score": 6.6137657,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "tt0077975",
        "_score": 6.6137657,
        "_source": {
          "directors": [
            "John Landis"
          ],
          "release_date": "1978-07-27T00:00:00Z",
          "rating": 7.5,
          "genres": [
            "Comedy",
            "Romance"
          ],
          "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTY2OTQxNTc1OF5BMl5BanBnXkFtZTYwNjA3NjI5._V1_SX400_.jpg",
          "plot": "At a 1962 College, Dean Vernon Wormer is determined to expel the
 entire Delta Tau Chi Fraternity, but those troublemakers have other plans for him.",
          "title": "Animal House",
          "rank": 527,
          "running_time_secs": 6540,
          "actors": [
```

```
            "John Belushi",
            "Karen Allen",
            "Tom Hulce"
          ],
          "year": 1978,
          "id": "tt0077975"
        }
      },
      ...
    ]
  }
}
```

デフォルトでは、このクエリは家という語句をすべてのインデックスのすべてのフィールドで検索します。検索を絞り込むには URI でインデックス (movies) およびドキュメントフィールド (title) を指定します。

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

リクエストに追加のパラメータを含めることはできますが、サポートされているパラメータは OpenSearch 検索オプションの小さなサブセットのみを提供します。次のリクエストは 20 の結果 (デフォルトでは 10) を返し、年ごとに (_score ではなく) ソートされます。

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

リクエストボディ検索

より複雑な検索を実行するには、クエリに HTTP リクエストボディと OpenSearchドメイン固有の言語 (DSL) を使用します。クエリ DSL では、検索 OpenSearch オプションの全範囲を指定できます。

> **Note**
>
> テキストフィールドの値に Unicode 特殊文字を含めることはできません。値は特殊文字で区切られた複数の値として解析されます。このような誤った解析により、意図せずに文書がフィルタリングされ、アクセス制御が損なわれる可能性があります。詳細については、 ドキュメントの「テキストフィールドの Unicode 特殊文字に関する注意事項 OpenSearch」を参照してください。

次の match クエリでは、最終的な URI 検索の例と似ています。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "sort": {
    "year": {
      "order": "desc"
    }
  },
  "query": {
    "query_string": {
      "default_field": "title",
      "query": "house"
    }
  }
}
```

> **Note**
>
> _search API はリクエストボディ検索のための HTTP GET および POST を受け入れます。ただし、すべての HTTP クライアントが GET リクエストにリクエストボディを追加することをサポートしているわけではありません。POST は普遍的な選択です。

多くの場合、いくつかのフィールドを検索する必要がある場合がありますが、すべてのフィールドを検索する必要はありません。multi_match クエリを使用します。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title", "plot", "actors", "directors"]
    }
  }
}
```

ブーストフィールド

特定のフィールドをブーストすることで、検索の関連性を向上させることができます。ブーストは乗数で、あるフィールドの一致を他のフィールドの一致より重く重み付けします。次の例では、`title` フィールドでの John の一致は `plot` フィールドの一致の 2 倍で `_score` に影響を及ぼし、また `actors` または `directors` フィールドの一致の 4 倍で影響を及ぼします。その結果、John Wick や John Carter のような映画が検索結果の最上部にあり、John Travolta 主演の映画は最下部にあります。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "john",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

検索結果のハイライト

`highlight` オプションは、クエリが 1 つ以上のフィールドと一致した場合、`hits`配列内の追加のオブジェクトを返す OpenSearch ように に指示します。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    }
  }
}
```

クエリが plot フィールドのコンテンツに一致した場合、ヒットは次のようになります。

```
{
  "_index": "movies",
  "_type": "movie",
  "_id": "tt0091541",
  "_score": 11.276199,
  "_source": {
    "directors": [
      "Richard Benjamin"
    ],
    "release_date": "1986-03-26T00:00:00Z",
    "rating": 6,
    "genres": [
      "Comedy",
      "Music"
    ],
    "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTIzODEzODE2OF5BMl5BanBnXkFtZTcwNjQ3ODcyMQ@@._V1_SX400_.jpg",
    "plot": "A young couple struggles to repair a hopelessly dilapidated house.",
    "title": "The Money Pit",
    "rank": 4095,
    "running_time_secs": 5460,
    "actors": [
      "Tom Hanks",
      "Shelley Long",
      "Alexander Godunov"
    ],
    "year": 1986,
    "id": "tt0091541"
  },
  "highlight": {
    "plot": [
      "A young couple struggles to repair a hopelessly dilapidated <em>house</em>."
    ]
  }
}
```

デフォルトでは、 は一致する文字列をタグで OpenSearch ラップし、マッチの周りに最大 100 文字のコンテキストを提供し、句読点、スペース、タブ、改行を識別してコンテンツを文に分割します。これらの設定はすべてカスタマイズ可能です。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
```

```
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    },
    "pre_tags": "<strong>",
    "post_tags": "</strong>",
    "fragment_size": 200,
    "boundary_chars": ".,!? "
  }
}
```

Count API

ドキュメントの内容に関心がなく、一致の数だけを知りたい場合は、_search API の代わりに _count API を使用できます。次のリクエストでは、query_string クエリを使用してロマンチックコメディを識別します。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

以下にレスポンスの例を示します。

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
```

```
    "skipped": 0,
    "failed": 0
  }
}
```

検索結果のページ分割

大量の検索結果を表示する必要がある場合は、さまざまな方法を使用してページネーションを実装できます。

ポイントインタイム

ポイントインタイム (PIT) 機能とは、時間が固定されているデータセットにさまざまなクエリを実行できる、検索の一種です。これは、特にディープページ分割の場合 OpenSearchに、 で推奨されるページ分割方法です。PIT は、 OpenSearch サービスバージョン 2.5 以降で使用できます。PIT の詳細については「???」を参照してください。

`from` と `size` のパラメーター

ページ分割の最も簡単な方法は、`from` と `size` パラメータを使用した方法です。次のリクエストは、検索結果のゼロから始まるインデックスリストのうち、20～39 までの結果を返します。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "from": 20,
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

検索ページ分割の詳細については、 OpenSearch ドキュメントの「結果のページ分割」を参照してください。

Dashboards Query Language

Dashboards クエリ言語 (DQL) を使用して、 OpenSearch Dashboards でデータとビジュアライゼーションを検索できます。DQL は、用語、ブール値、日付と範囲、およびネストされたフィールドの 4 つの主要なクエリタイプを使用します。

用語のクエリ

用語クエリでは、検索する用語を指定する必要があります。

用語クエリを実行するには、次のように入力します。

```
host:www.example.com
```

ブール値クエリ

ブール演算子 AND、or、および not を使用して、複数のクエリを組み合わせることができます。

ブール値クエリを実行するには、次を貼り付けます。

```
host.keyword:www.example.com and response.keyword:200
```

日付と範囲のクエリ

日付と範囲のクエリを使用して、クエリの前後の日付を検索できます。

- > は、指定した日付より後の日付を検索することを示します。
- < は、指定した日付より前の日付を検索することを示します。

```
@timestamp > "2020-12-14T09:35:33"
```

ネストされたフィールドのクエリ

ネストされたフィールドを持つドキュメントがある場合は、ドキュメントのどの部分を取得するかを指定する必要があります。ネストされたフィールドを含むサンプルドキュメントを次に示します。

```
{"NBA players":[
    {"player-name": "Lebron James",
      "player-position": "Power forward",
      "points-per-game": "30.3"
    },
```

```
    {"player-name": "Kevin Durant",
      "player-position": "Power forward",
      "points-per-game": "27.1"
    },
    {"player-name": "Anthony Davis",
      "player-position": "Power forward",
      "points-per-game": "23.2"
    },
    {"player-name": "Giannis Antetokounmpo",
      "player-position": "Power forward",
      "points-per-game":"29.9"
    }
  ]
}
```

DQL を使用して特定のフィールドを取得するには、次を貼り付けます。

```
NBA players: {player-name: Lebron James}
```

ネストされたドキュメントから複数のオブジェクトを取得するには、次を貼り付けます。

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo}
```

範囲内を検索するには、次を貼り付けます。

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo and < 30}
```

別のオブジェクト内にネストされたオブジェクトがドキュメントにある場合でも、すべてのレベルを指定することでデータを取得できます。これを実行するには、次を貼り付けます。

```
Top-Power-forwards.NBA players: {player-name:Lebron James}
```

Amazon OpenSearch サービスのカスタムパッケージ

Amazon OpenSearch Service では、ストップワードやシノニムなどのカスタム辞書ファイルをアップロードできます。また、ドメインに関連付けることができるパッケージ済みのオプションプラグインもいくつか用意されています。これら両方のタイプのファイルの総称は、パッケージです。

辞書ファイルを使用すると、頻度の高い特定の単語を無視したり、「フローズンカスタード」、「ジェラート」、「アイスクリーム」 OpenSearch などの用語を同等のものとして扱うように指示したりして、検索結果を改善できます。また、日本語 (kuromoji) 分析プラグインなどのステミングを改善することもできます。

オプションプラグインはドメインに追加機能を提供できます。例えば、Amazon Personalize プラグインを使用して、パーソナライズされた検索結果を得ることができます。オプションプラグインは ZIP-PLUGIN パッケージタイプを使用します。オプションのプラグインについての詳細は、「the section called “エンジンバージョンに応じたプラグイン”」を参照してください。

トピック

- パッケージの許可要件
- Amazon S3 へのパッケージのアップロード
- パッケージのインポートと関連付け
- パッケージとの併用 OpenSearch
- パッケージの更新
- 辞書の手動インデックス更新
- パッケージの関連付け解除と削除

パッケージの許可要件

管理者権限のないユーザーがパッケージを管理するには、特定の AWS Identity and Access Management (IAM) アクションが必要です。

- `es:CreatePackage`- OpenSearch サービスリージョンにパッケージを作成
- `es:DeletePackage`- OpenSearch サービスリージョンからパッケージを削除
- `es:AssociatePackage` - パッケージをドメインに関連付ける
- `es:DissociatePackage` - ドメインからパッケージの関連付けを解除する

カスタムパッケージが存在する Amazon S3 バケットパスまたはオブジェクトに対する許可も必要です。

ドメインアクセスポリシー内ではなく、IAM 内のすべての許可を付与します。詳細については、「the section called “Identity and Access Management”」を参照してください。

Amazon S3 へのパッケージのアップロード

このセクションでは、オプションのプラグインパッケージが既にプリインストールされているため、カスタム辞書パッケージをアップアップロードする方法について説明します。カスタム辞書をドメインに関連付ける前に、Amazon S3 バケットにアップロードする必要があります。手順については、Amazon Simple Storage Service ユーザーガイドの「オブジェクトのアップロード」を参照してください。サポートされているプラグインをアップロードする必要はありません。

辞書に機密情報が含まれている場合は、アップロード時に S3 が管理する鍵によるサーバー側の暗号化を指定してください。 OpenSearch キーを使用して保護されている S3 上のファイルにサービスはアクセスできません。 AWS KMS

ファイルをアップロードしたら、その S3 パスを書き留めます。パスの形式は `s3://bucket-name/file-path/file-name` です。

テスト目的で、次のシノニムファイルを使用できます。`synonyms.txt` と名前を付けて保存します。

```
danish, croissant, pastry
ice cream, gelato, frozen custard
sneaker, tennis shoe, running shoe
basketball shoe, hightop
```

Hunspell 辞書など一部の辞書では、複数のファイルが使用され、ファイルシステム上に独自のディレクトリが必要になります。現時点では、 OpenSearch Service は単一ファイルのディクショナリのみをサポートしています。

パッケージのインポートと関連付け

コンソールは、カスタムディクショナリを Service にインポートする最も簡単な方法です。 OpenSearch Amazon S3 からディクショナリをインポートすると、 OpenSearch Service はパッケージの独自のコピーを保存し、そのコピーをサービスマネージドキーで AES-256 OpenSearch を使用して自動的に暗号化します。

オプションのプラグインは OpenSearch Service にあらかじめインストールされているため、自分でアップロードする必要はありませんが、プラグインをドメインに関連付ける必要があります。使用可能なプラグインは、コンソールの [パッケージ] 画面に一覧表示されます。

パッケージをインポートし、ドメインに関連付けます。 AWS Management Console

1. Amazon OpenSearch サービスコンソールで、[パッケージ] を選択します。

2. [パッケージのインポート] を選択します。
3. カスタム辞書にわかりやすい名前を付けます。
4. ファイルへの S3 パスを指定し、[送信] を選択します。
5. [パッケージ] 画面に戻ります。
6. パッケージのステータスが [使用可能] の場合は、パッケージを選択します。オプションのプラグインは自動的に [使用可能] になります。
7. [ドメインへの関連付け] を選択します。
8. ドメインを選択し、[関連付け] を選択します。
9. ナビゲーションペインで、ドメインを選択し、[パッケージ] タブに進みます。
10. パッケージがカスタム辞書の場合、パッケージが [使用可能] になったら ID を書き留めておきます。`analyzers/id` OpenSearchへのリクエストのファイルパスとして使用します。

または、SDK AWS CLI、または設定 API を使用してパッケージをインポートし、関連付けることもできます。詳細については、「AWS CLI コマンドリファレンス」と「Amazon OpenSearch サービス API リファレンス」を参照してください。

パッケージとの併用 OpenSearch

このセクションでは、カスタム辞書とオプションプラグインの両タイプのパッケージの使用方法について説明します。

カスタム辞書の使用

ファイルをドメインに関連付けた後は、トークナイザやトークンフィルターの作成時に、`synonyms_path`、`stopwords_path`、`user_dictionary` などのパラメータで使用できます。正確なパラメータは、オブジェクトによって異なります。`synonyms_path` と `stopwords_path` はいくつかのオブジェクトでサポートされますが、`user_dictionary` は kuromoji プラグイン専用です。

IK (中国語) 分析プラグインの場合、カスタム辞書ファイルをカスタムパッケージとしてアップロードしてドメインに関連付けることができます。アップロードしたカスタム辞書ファイルはプラグインによって自動的にピックアップされます。`user_dictionary` パラメータは必要ありません。ファイルがシノニムファイルの場合は、`synonyms_path` パラメータを使用します。

次の例では、シノニムファイルを新しいインデックスに追加します。

```
PUT my-index
```

```
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["my_filter"]
          }
        },
        "filter": {
          "my_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F111111111",
            "updateable": true
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "standard",
        "search_analyzer": "my_analyzer"
      }
    }
  }
}
```

このリクエストは、標準トークナイザとシノニムトークンフィルターを使用するインデックスのカスタムアナライザーを作成します。

- トークナイザは、いくつかのルールのセットに基づいて、文字のストリームをトークン (通常は単語) に分割します。最も単純な例は、空白文字に遭遇するたびに、先行する文字をトークンに分割する空白トークナイザです。より複雑な例は標準トークナイザです。これは、一連の文法ベースのルールを使用して多くの言語で動作します。
- トークンフィルターは、トークンの追加、変更、削除を行います。たとえば、シノニムトークンフィルターは、シノニムリストで単語が見つかったときにトークンを追加します。ストップトークンフィルターは、ストップワードリスト内の単語を検出すると、トークンを削除します。

また、このリクエストはマッピングにテキストフィールド (description) を追加し、 OpenSearch 新しいアナライザーを検索アナライザーとして使用するように指示します。インデックスアナライザーとして、まだ標準アナライザーが使用されていることがわかります。

最後に、トークンフィルターの行 "updateable": true をメモします。このフィールドは検索アナライザーにのみ適用され、インデックスアナライザーには適用されず、後で検索アナライザーを更新したい場合に不可欠です。

テスト目的のために、インデックスにいくつかのドキュメントを追加します。

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "description": "ice cream" }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "description": "croissant" }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "description": "tennis shoe" }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "description": "hightop" }
```

次に、シノニムを使用して検索します。

```
GET my-index/_search
{
  "query": {
    "match": {
      "description": "gelato"
    }
  }
}
```

この場合、 OpenSearch 次の応答が返されます。

```
{
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 0.99463606,
    "hits": [{
```

```
      "_index": "my-index",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.99463606,
      "_source": {
        "description": "ice cream"
      }
    }]
  }
}
```

> **Tip**
>
> 辞書ファイルは、そのサイズに比例して Java ヒープ領域を使用します。たとえば、2 GiB の辞書ファイルは、ノード上で 2 GiB のヒープ領域を消費する可能性があります。大きなファイルを使用する場合は、収容に十分なヒープ領域がノードにあることを確認してください。JVMMemoryPressure メトリクスをモニタリングし、必要に応じてクラスターをスケーリングします。

オプションプラグインの使用

OpenSearch このサービスでは、 OpenSearch あらかじめインストールされているオプションのプラグインをドメインに関連付けて使用することができます。 OpenSearch オプションのプラグインパッケージは特定のバージョンと互換性があり、そのバージョンのドメインにのみ関連付けることができます。ドメインで利用可能なパッケージのリストには、そのドメインバージョンと互換性のある、サポートされているすべてのプラグインが含まれています。プラグインをドメインに関連付けると、ドメインへのインストールプロセスが開始されます。そうすれば、 OpenSearch Service にリクエストを送信する際にプラグインを参照して使用できるようになります。

プラグインの関連付けと解除には、ブルー/グリーンデプロイが必要です。詳細については、「the section called “通常は blue/green デプロイの原因となる変更”」を参照してください。

オプションのプラグインには、言語アナライザーやカスタマイズされた検索結果が含まれます。例えば、Amazon Personalize Search Ranking プラグインは、機械学習を使用してお客様の検索結果をパーソナライズします。このプラグインについて詳しくは、「 OpenSearchからの検索結果のカスタマイズ」を参照してください。サポートされているすべてのプラグインのリストについては、「the section called “エンジンバージョンに応じたプラグイン”」を参照してください。

Sudachi プラグイン

Sudachi プラグインの場合、ディクショナリファイルを再関連付けしても、すぐにドメインには反映されません。設定変更やその他の更新の一環として、次のブルー/グリーンデプロイがドメインで実行されると、辞書が更新されます。あるいは、更新されたデータを含む新しいパッケージを作成し、この新しいパッケージを使用して新しいインデックスを作成し、既存のインデックスを新しいインデックスに再インデックスしてから、古いインデックスを削除することもできます。インデックスの再作成方法を使用する場合は、トラフィックが中断されないようにインデックスエイリアスを使用してください。

さらに、Sudachi プラグインは API オペレーションでアップロードできるバイナリ Sudachi 辞書のみをサポートします。CreatePackageビルド済みのシステムディクショナリの詳細およびユーザーディクショナリのコンパイルプロセスについては、Sudachi のドキュメントを参照してください。

次の例は、Sudachi トークナイザでシステムディクショナリとユーザーディクショナリを使用する方法を示しています。これらのディクショナリは、タイプ TXT-DICTIONARY のカスタムパッケージとしてアップロードし、追加設定でパッケージ ID を指定する必要があります。

```
PUT sudachi_sample
{
  "settings": {
    "index": {
      "analysis": {
        "tokenizer": {
          "sudachi_tokenizer": {
            "type": "sudachi_tokenizer",
            "additional_settings": "{\"systemDict\": \"<system-dictionary-package-
id>\",\"userDict\": [\"<user-dictionary-package-id>\"]}"
        }
        },
        "analyzer": {
          "sudachi_analyzer": {
            "filter": ["my_searchfilter"],
            "tokenizer": "sudachi_tokenizer",
            "type": "custom"
          }
        },
        "filter":{
          "my_searchfilter": {
            "type": "sudachi_split",
            "mode": "search"
          }
```

```
        }
      }
    }
  }
}
```

パッケージの更新

オプションのプラグインパッケージはすでに更新されているため、このセクションではカスタム辞書パッケージの更新方法についてのみ説明します。Amazon S3 に新しいバージョンの辞書をアップロードしても、Amazon OpenSearch Service のパッケージは自動的に更新されません。OpenSearch サービスはファイルの独自のコピーを保存するため、新しいバージョンを S3 にアップロードする場合は、手動で更新する必要があります。

関連付けられた各ドメインはそのファイルの独自のコピーも保存します。検索動作を予測可能に保つために、ドメインは明示的に更新されるまで、現在のパッケージバージョンを引き続き使用します。カスタムパッケージを更新するには、Service でファイルを変更し Amazon S3 Control、OpenSearch Service でパッケージを更新してから、更新を適用します。

パッケージを次のように更新します。 AWS Management Console

1. OpenSearch サービスコンソールで [パッケージ] を選択します。
2. パッケージを選択し、[更新] を選択します。
3. ファイルへの S3 パスを指定し、[パッケージの更新] を選択します。
4. [パッケージ] 画面に戻ります。
5. パッケージのステータスが [使用可能] に変わったら、それを選択します。次に、関連付けられたドメインを 1 つ以上選択し、[更新の適用] を選択し、確定します。関連付けステータスが [アクティブ] に変わるまで待ちます。
6. 次の手順は、インデックスの設定方法によって異なります。

   - ドメインが Elasticsearch 7.8 OpenSearch 以降を実行していて、updateable フィールドが true に設定された検索アナライザーのみを使用している場合は、それ以上の操作は必要ありません。 OpenSearch サービスは _plugins/_refresh_search_analyzers API を使用してインデックスを自動的に更新します。
   - ドメインで Elasticsearch 7.7 以前を実行している場合、インデックスアナライザーを使用している場合、またはフィールドを使用していない場合は、を参照してください。updateable the section called “辞書の手動インデックス更新”

コンソールは最も単純な方法ですが、、SDK AWS CLI、または設定 API を使用してサービスパッケージを更新することもできます。 OpenSearch 詳細については、「AWS CLI コマンドリファレンス」と「Amazon OpenSearch サービス API リファレンス」を参照してください。

AWS SDK を使用してパッケージを更新します。

コンソールでパッケージを手動で更新する代わりに、SDK を使用して更新プロセスを自動化できます。次のサンプル Python スクリプトは、新しいパッケージファイルを Amazon S3 にアップロードし、 OpenSearch Service 内のパッケージを更新し、新しいパッケージを指定されたドメインに適用します。更新が成功したことを確認したら、 OpenSearch新しいシノニムが適用されたことを示すサンプル呼び出しを行います。

host、region、file_name、bucket_name、s3_key、package_id、domain_name、および query の値を指定する必要があります。

```
from requests_aws4auth import AWS4Auth
import boto3
import requests
import time
import json
import sys

host = ''  # The OpenSearch domain endpoint with https:// and a trailing slash. For
 example, https://my-test-domain.us-east-1.es.amazonaws.com/
region = ''  # For example, us-east-1
file_name = ''  # The path to the file to upload
bucket_name = ''  # The name of the S3 bucket to upload to
s3_key = ''  # The name of the S3 key (file name) to upload to
package_id = ''  # The unique identifier of the OpenSearch package to update
domain_name = ''  # The domain to associate the package with
query = ''  # A test query to confirm the package has been successfully updated

service = 'es'
credentials = boto3.Session().get_credentials()
client = boto3.client('opensearch')
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def upload_to_s3(file_name, bucket_name, s3_key):
    """Uploads file to S3"""
    s3 = boto3.client('s3')
    try:
```

```
        s3.upload_file(file_name, bucket_name, s3_key)
        print('Upload successful')
        return True
    except FileNotFoundError:
        sys.exit('File not found. Make sure you specified the correct file path.')


def update_package(package_id, bucket_name, s3_key):
    """Updates the package in OpenSearch Service"""
    print(package_id, bucket_name, s3_key)
    response = client.update_package(
        PackageID=package_id,
        PackageSource={
            'S3BucketName': bucket_name,
            'S3Key': s3_key
        }
    )
    print(response)


def associate_package(package_id, domain_name):
    """Associates the package to the domain"""
    response = client.associate_package(
        PackageID=package_id, DomainName=domain_name)
    print(response)
    print('Associating...')


def wait_for_update(domain_name, package_id):
    """Waits for the package to be updated"""
    response = client.list_packages_for_domain(DomainName=domain_name)
    package_details = response['DomainPackageDetailsList']
    for package in package_details:
        if package['PackageID'] == package_id:
            status = package['DomainPackageStatus']
            if status == 'ACTIVE':
                print('Association successful.')
                return
            elif status == 'ASSOCIATION_FAILED':
                sys.exit('Association failed. Please try again.')
            else:
                time.sleep(10)  # Wait 10 seconds before rechecking the status
                wait_for_update(domain_name, package_id)
```

```
def sample_search(query):
    """Makes a sample search call to OpenSearch"""
    path = '_search'
    params = {'q': query}
    url = host + path
    response = requests.get(url, params=params, auth=awsauth)
    print('Searching for ' + '"' + query + '"')
    print(response.text)
```

> **Note**
>
> を使用してスクリプトを実行したときに「package not found」エラーが表示される場合は AWS CLI、Boto3 が ~/.aws/config で指定されているリージョン (S3 バケットがあるリージョンではない) を使用している可能性があります。`aws configure` を実行して正しいリージョンを指定するか、明示的にクライアントにそのリージョンを追加します。
>
> ```
> client = boto3.client('opensearch', region_name='us-east-1')
> ```

辞書の手動インデックス更新

手動によるインデックス更新はカスタム辞書にのみ適用され、オプションのプラグインには適用されません。更新された辞書を使用するには、次の条件のいずれかを満たしている場合、インデックスを手動で更新する必要があります。

- ドメインは Elasticsearch 7.7 以前を実行しています。
- カスタムパッケージをインデックスアナライザーとして使用しています。
- 検索アナライザーとしてカスタムパッケージを使用していますが、[更新可能] フィールドは含まれていません。

新しいパッケージファイルでアナライザーを更新するには、次の 2 つのオプションがあります。

- 更新するインデックスを閉じて開きます。

```
POST my-index/_close
POST my-index/_open
```

- インデックスを再作成します。まず、更新されたシノニムファイル (または完全に新しいファイル) を使用するインデックスを作成します。UTF-8 のみがサポートされていることに注意してください。

```
PUT my-new-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "synonym_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["synonym_filter"]
          }
        },
        "filter": {
          "synonym_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F222222222"
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "synonym_analyzer"
      }
    }
  }
}
```

次に、古いインデックスをその新しいインデックスに再作成します。

```
POST _reindex
{
  "source": {
    "index": "my-index"
  },
```

```
  "dest": {
    "index": "my-new-index"
  }
}
```

インデックスアナライザーを頻繁に更新する場合は、インデックスエイリアスを使用して、最新のインデックスへの一貫したパスを維持します。

```
POST _aliases
{
  "actions": [
    {
      "remove": {
        "index": "my-index",
        "alias": "latest-index"
      }
    },
    {
      "add": {
        "index": "my-new-index",
        "alias": "latest-index"
      }
    }
  ]
}
```

古いインデックスが必要ない場合は、削除します。

```
DELETE my-index
```

パッケージの関連付け解除と削除

カスタム辞書であれオプションのプラグインであれ、ドメインからパッケージを切り離すと、新しいインデックスを作成するときにそのパッケージを使用できなくなります。パッケージの関連付けが解除されると、そのパッケージを使用していた既存のインデックスはそのパッケージを使用できなくなります。関連付けを解除する前に、どのインデックスからもパッケージを削除する必要があります。削除しないと、関連付けの解除は失敗します。

コンソールは、パッケージをドメインから切り離し、サービスから削除する最も簡単な方法です。OpenSearch OpenSearch サービスからパッケージを削除しても、Amazon S3 の元の場所からパッケージは削除されません。

AWS Management Consoleでドメインからパッケージの関連付けを解除します

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。
2. [分析] で [Amazon OpenSearch サービス] を選択します。
3. ナビゲーションペインで、ドメインを選択し、[パッケージ] タブを選択します。
4. パッケージを選択して、[アクション] を選択し、[関連付け解除] を選択します。選択内容を確認します。
5. パッケージがリストから消えるのを待ちます。ブラウザを更新することが必要な場合があります。
6. パッケージを他のドメインで使用する場合は、ここで停止します。パッケージの削除を続行するには (カスタム辞書の場合)、ナビゲーションペインで [パッケージ] を選択します。
7. パッケージを選択し、[削除] を選択します。

または、SDK AWS CLI、または設定 API を使用してパッケージの関連付けを解除したり削除したりします。詳細については、「AWS CLI コマンドリファレンス」と「Amazon OpenSearch サービス API リファレンス」を参照してください。

SQL を使用した Amazon OpenSearch Service データのクエリ

JSON ベースのクエリ DSL を使用するのではなく OpenSearch 、SQL を使用して Amazon Service をクエリできます。 OpenSearch SQL を用いたクエリの実施は、その言語にすでに精通している場合や、それを使用するアプリケーションにドメインを統合する場合に役立ちます。SQL サポートは、 OpenSearch または Elasticsearch 6.5 以降を実行しているドメインで使用できます。

> **Note**
>
> このドキュメントでは、 OpenSearch サービスと SQL プラグインのさまざまなバージョン、および JDBC および ODBC ドライバーのバージョン互換性について説明します。基本および複雑なクエリ、関数、メタデータクエリ、集計関数の構文については、オープンソースのOpenSearchドキュメントを参照してください。

次の表を使用して、各 OpenSearch および Elasticsearch バージョンでサポートされている SQL プラグインのバージョンを確認します。

OpenSearch

OpenSearch バージョン	SQL プラグインバージョン	注目すべき機能
2.13.0	2.13.0.0	
2.11.0	2.11.0.0	PPL 言語とクエリのサポートを追加
2.9.0	2.9.0.0	Spark コネクターを追加し、テーブル関数と PromQL 関数をサポートします
2.7.0	2.7.0.0	`datasource` API を追加
2.5.0	2.5.0.0	
2.3.0	2.3.0.0	`maketime` および `makedate` 日時関数を追加
1.3.0	1.3.0.0	デフォルトのクエリ制限サイズと、値リスト内から選択する IN 句をサポート
1.2.0	1.2.0.0	可視化応答形式の新しいプロトコルを追加
1.1.0	1.1.0.0	SQL と PPL のフィルターとしてマッチ機能をサポートする
1.0.0	1.0.0.0	データストリームのクエリをサポートする

Open Distro for Elasticsearch

Elasticsearch バージョン	SQL プラグインバージョン	注目すべき機能
7.10	1.13.0	Window 関数の NULL FIRST および LAST、CAST() 関数、SHOW、および DESCRIBE コマンド
7.9	1.11.0	追加の日付/時刻関数、ORDER BY キーワードの追加

Elasticsearch バージョン	SQL プラグインバージョン	注目すべき機能
7.8	1.9.0	
7.7	1.8.0	
7.3	1.3.0	複数の文字列および数値演算子
7.1	1.1.0	

サンプル呼び出し

SQL を用いてデータのクエリを行うには、次の形式を使用して _sql に HTTP リクエストを送信します。

```
POST domain-endpoint/_plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

> **Note**
>
> ドメインが ではなく Elasticsearch を実行している場合 OpenSearch、形式は です_opendistro/_sql。

注意と相違点

_plugins/_sql の呼び出しではリクエストボディにインデックス名が含まれるため、バルク、mget、および msearch オペレーションと同じアクセスポリシーの考慮事項が適用されます。これまでどおり、API オペレーションにアクセス許可を付与するときは、最小権限の原則に従う必要があります。

きめ細かなアクセスコントロールでの SQL の使用に関連するセキュリティ上の考慮事項については、「the section called “きめ細かなアクセスコントロール”」を参照してください。

OpenSearch SQL プラグインには、調整可能な設定が多数含まれています。 OpenSearch サービスでは、プラグイン設定_cluster/settingsパス () ではなく パスを使用します_plugins/_query/settings。

```
PUT _cluster/settings
{
  "transient" : {
    "plugins.sql.enabled" : true
  }
}
```

従来の Elasticsearch ドメインの場合、plugins を opendistro に置き換えます。

```
PUT _cluster/settings
{
  "transient" : {
    "opendistro.sql.enabled" : true
  }
}
```

SQL Workbench

SQL Workbench は OpenSearch Dashboards ユーザーインターフェイスで、オンデマンド SQL クエリの実行、SQL を REST に相当するものへの変換、結果をテキスト、JSON、JDBC、CSV として表示および保存できます。詳細については、「クエリワークベンチ」を参照してください。

SQL CLI

SQL CLI は、opensearchsql コマンドで起動できるスタンドアロンの Python アプリケーションです。インストール、構成、および使用のステップについては、「SQL CLI」を参照してください。

JDBC ドライバー

Java Database Connectivity (JDBC) ドライバーを使用すると、 OpenSearch サービスドメインをお気に入りのビジネスインテリジェンス (BI) アプリケーションと統合できます。ドライバーをダウンロードするには、こちらをクリックしてください。詳細については、GitHub「」リポジトリを参照してください。

ドライバー用のバージョン互換性の概要を、次のテーブルに示します。

OpenSearch

OpenSearch バージョン	JDBC ドライバーバージョン
2.13	1.1.0.1
2.11	1.1.0.1
2.9	1.1.0.1
2.7	1.1.0.1
2.5	1.1.0.1
2.3	1.1.0.1
1.3	1.1.0.1
1.2	1.1.0.1
1.1	1.1.0.1
1.0	1.1.0.1

Open Distro for Elasticsearch

Elasticsearch バージョン	JDBC ドライバーバージョン
7.10	1.13.0
7.9	1.11.0
7.8	1.9.0
7.7	1.8.0
7.4	1.4.0
7.1	1.0.0
6.8	0.9.0

Elasticsearch バージョン	JDBC ドライバーバージョン
6.7	0.9.0
6.5	0.9.0

ODBC ドライバー

Open Database Connectivity (ODBC) ドライバーは、Windows と macOS 用の、読み取り専用の ODBC ドライバーです。これを使用すれば、Microsoft Excel のようなビジネスインテリジェンスやデータ可視化のアプリケーションを SQL プラグインに接続できます。

アー OpenSearch ティファクトページ で作業ドライバーファイルの例をダウンロードできます。ドライバーのインストールについては、「」の「SQL リポジトリ GitHub」を参照してください。

Amazon OpenSearch Service での k-Narest Neighbor (k-NN) 検索

関連付けられた K 最近傍アルゴリズムの略で、Amazon OpenSearch Service の k-NN を使用すると、ベクトル空間内のポイントを検索し、ユークリッド距離またはコサイン類似度でそれらのポイントの「最近傍」を見つけることができます。ユースケースには、推奨 (音楽アプリケーションの「おすすめの曲」機能など)、画像認識、不正行為の検出などがあります。

> **Note**
>
> このドキュメントでは、 OpenSearch サービスと k-NN プラグインのさまざまなバージョン間のバージョン互換性、および マネージド OpenSearch サービスでプラグインを使用する場合の制限について説明します。シンプルな例と複雑な例、パラメータリファレンス、プラグインの完全な API リファレンスなど、k-NN OpenSearch プラグインの包括的なドキュメントについては、オープンソースドキュメント を参照してください。オープンソースのドキュメントでは、パフォーマンスのチューニングと k-NN 固有のクラスター設定についても説明します。

次の表を使用して、Amazon OpenSearch Service ドメインで実行されている k-NN プラグインのバージョンを確認します。各 k-NN プラグインバージョンは、 OpenSearchまたは Elasticsearch バージョンに対応します。

OpenSearch

OpenSearch バージョン	k-NN プラグインバージョン	注目すべき機能
2.13	2.13.0.0	
2.11	2.11.0.0	k-NN クエリ内の `ignore_unmapped` のサポートが追加されました。
2.9	2.9.0.0	Faiss エンジンによる k-NN バイトベクトルと効率的なフィルタリングが実装されました。
2.7	2.7.0.0	
2.5	2.5.0.0	k-NN モデルシステムインデックス SystemIndexPlugin 用に拡張、コア HybridFS に Lucene 固有のファイル拡張を追加
2.3	2.3.0.0	
1.3	1.3.0.0	
1.2	1.2.0.0	Faiss ライブラリのサポートを追加
1.1	1.1.0.0	
1.0	1.0.0.0	下位互換性をサポートしながらREST APIの名前が変更され、`opendistro` から `opensearch` に名前空間の名前が変更されます

Elasticsearch

Elasticsearch バージョン	k-NN プラグインバージョン	注目すべき機能
7.1	1.3.0.0	ユークリッド距離
7.4	1.4.0.0	

Elasticsearch バージョン	k-NN プラグインバージョン	注目すべき機能
7.7	1.8.0.0	コサイン類似度
7.8	1.9.0.0	
7.9	1.11.0.0	ウォームアップ API、カスタムスコアリング
7.10	1.13.0.0	ハミング距離、L1 ノルム距離、ペインレススクリプティング

k-NN を用いた開始方法

k-NN を使用するには、index.knn 設定でインデックスを作成し、knn_vector データ型の 1 つ以上のフィールドを追加する必要があります。

```
PUT my-index
{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "my_vector1": {
        "type": "knn_vector",
        "dimension": 2
      },
      "my_vector2": {
        "type": "knn_vector",
        "dimension": 4
      }
    }
  }
}
```

knn_vector データ型は、必要な dimension パラメータによって定義された浮動小数点数とともに、10,000 個までの浮動小数点数の単一のリストをサポートします。インデックスを作成したら、そのインデックスにデータを追加します。

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "my_vector1": [1.5, 2.5], "price": 12.2 }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "my_vector1": [2.5, 3.5], "price": 7.1 }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "my_vector1": [3.5, 4.5], "price": 12.9 }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "my_vector1": [5.5, 6.5], "price": 1.2 }
{ "index": { "_index": "my-index", "_id": "5" } }
{ "my_vector1": [4.5, 5.5], "price": 3.7 }
{ "index": { "_index": "my-index", "_id": "6" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 10.3 }
{ "index": { "_index": "my-index", "_id": "7" } }
{ "my_vector2": [2.5, 3.5, 5.6, 6.7], "price": 5.5 }
{ "index": { "_index": "my-index", "_id": "8" } }
{ "my_vector2": [4.5, 5.5, 6.7, 3.7], "price": 4.4 }
{ "index": { "_index": "my-index", "_id": "9" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 8.9 }
```

次に、knn クエリタイプを使用してデータを検索できます。

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  }
}
```

この場合、k はクエリが返す最近傍の数ですが、size オプションも含める必要があります。それ以外の場合は、クエリ全体の k 結果ではなく、各シャード (および各セグメント) の k 結果が返されます。k-NN は、最大 10,000 の k 値をサポートします。

knn クエリを他の句と混在させると、返される結果は k の結果よりも少なくなる場合があります。この例では、post_filter 句を使用することにより、結果の数が 2 から 1 に減ります。

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  },
  "post_filter": {
    "range": {
      "price": {
        "gte": 6,
        "lte": 10
      }
    }
  }
}
```

最適なパフォーマンスを維持しながら大量のクエリを処理する必要がある場合は、_msearch API を使用して JSON を使用して一括検索を作成し、1 つのリクエストを送信して複数の検索を実行できます。

```
GET _msearch
{ "index": "my-index"}
{ "query": { "knn": {"my_vector2":{"vector": [2, 3, 5, 6],"k":2 }} } }
{ "index": "my-index", "search_type": "dfs_query_then_fetch"}
{ "query": { "knn": {"my_vector1":{"vector": [2, 3],"k":2 }} } }
```

次の動画は、K-NN クエリの一括ベクトル検索を設定する方法を示しています。

k-NN の違い、チューニング、および制限

OpenSearch では、 _cluster/settings API を使用してすべての k-NN 設定を変更できます。OpenSearch サービスでは、 knn.memory.circuit_breaker.enabledと を除くすべての設定を変更できますknn.circuit_breaker.triggered。k-NN 統計は Amazon CloudWatch メトリクスとして含まれます。

特に、各データノードの KNNGraphMemoryUsageメトリクスを knn.memory.circuit_breaker.limit 統計とインスタンスタイプの使用可能な RAM と照らし合わせて確認します。 OpenSearch サービスは Java ヒープにインスタンスの RAM の半分を使用します (ヒープサイズは 32 GiB まで)。デフォルトでは、k-NN は残りの半分の 50% まで使用するため、32 GiB の RAM を持つインスタンスタイプは 8 GiB のグラフに対応できます (32 * 0.5 * 0.5)。グラフのメモリ使用量がこの値を超えると、パフォーマンスが低下する可能性があります。

インデックスがおおよその k-NN () を使用している場合、k-NN インデックスを UltraWarmまたはコールドストレージに移行することはできません"index.knn": true。index.knn が false (k-NN) に設定されている場合、インデックスを他のストレージ階層に移動できます。

Amazon OpenSearch Service でのクラスター間検索

Amazon OpenSearch Service でのクラスター間検索では、接続された複数のドメインでクエリと集計を実行できます。さまざまなタイプのワークロードを実行している場合には特に、単一の大きなドメインではなく、複数の小さなドメインを使用する方が合理的です。

ワークロード固有のドメインを使用すると、次のタスクを実行できます。

- 特定のワークロードのインスタンスタイプを選択することにより、各ドメインを最適化する。
- ワークロード間で障害分離の境界を確立する。これは、ワークロードの 1 つに障害が発生しても、その障害はその特定のドメイン内で食い止められ、他のワークロードに影響しないことを意味します。
- ドメイン間でより簡単にスケーリングできる。

クラスター間検索は OpenSearch Dashboards をサポートしているため、すべてのドメインにわたって視覚化とダッシュボードを作成できます。ドメイン間で転送される検索結果には、標準の AWS データ転送料金がかかります。

> **Note**
>
> オープンソースには、クラスター間検索に関するドキュメント OpenSearch もあります。オープンソースクラスターのセットアップは、マネージド Amazon OpenSearch Service ドメインとは大きく異なります。特に、 OpenSearch サービスでは、cURL AWS Management Console ではなく を使用してクラスター間接続を設定します。さらに、 マネージドサービスは、きめ細かなアクセスコントロールに加えて、クラスター間認証に AWS Identity and Access Management (IAM) を使用します。したがって、オープンソースのドキュメントでは

なくこの OpenSearch ドキュメントを使用して、ドメインのクラスター間検索を設定することをお勧めします。

トピック

制限事項

クラスター間検索には、以下に示す重要な制限事項があります。

- Elasticsearch ドメインを OpenSearch ドメインに接続することはできません。
- セルフマネージド OpenSearch/Elasticsearch クラスターに接続することはできません。
- リージョン間でドメインを接続するには、両方のドメインが Elasticsearch 7.10 以降または 上にある必要があります OpenSearch。
- ドメインでは、最大 20 の送信接続が可能です。同様に、ドメインでは最大 20 の着信接続が可能です。つまり、1 つのドメインは、最大 20 の他のドメインに接続できます。
- ソースドメインは、宛先ドメインと同じかそれ以降のバージョンである必要があります。2 つのドメイン間で双方向接続を設定し、その一方または両方をアップグレードする場合は、まず接続の 1 つを削除する必要があります。
- クラスター間検索で、カスタム辞書や SQL を使用することはできません。
- を使用してドメイン AWS CloudFormation を接続することはできません。
- M3 インスタンスとバースト可能 (T2 および T3) インスタンスではクラスター間検索は使用できません。

クラスター間検索の前提条件

クラスター間検索をセットアップする前に、ドメインが次の要件を満たしていることを確認してください。

- 2 つの OpenSearch ドメイン、またはバージョン 6.7 以降の Elasticsearch ドメイン
- Fine-grained access controlが有効
- N ode-to-node 暗号化が有効

クラスター間検索の料金

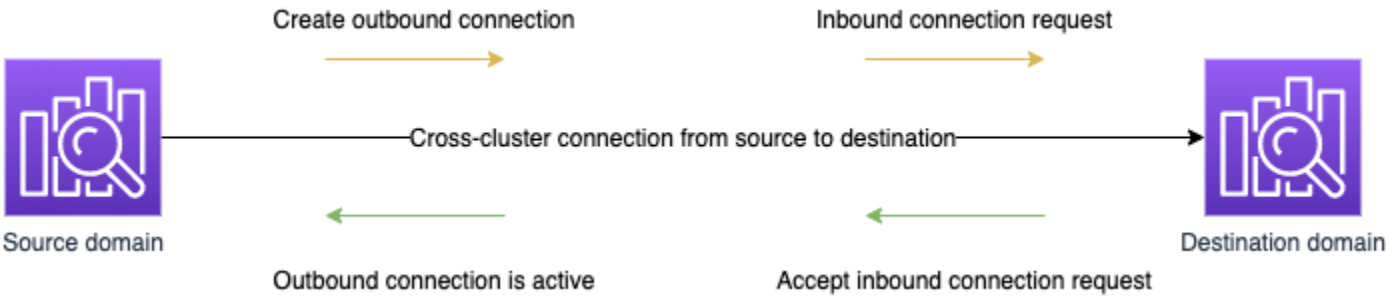
ドメインをまたがって検索する場合も、追加料金はかかりません。

接続のセットアップ

"ソース" ドメインは、クラスター間検索リクエストが発生したドメインを指します。つまり、ソースドメインは、ユーザーからの最初の検索リクエストの送信先です。

"ターゲット" ドメインは、ソースドメインから照会するドメインです。

クラスター間の接続は、ソースからターゲットドメインへの単方向のみです。つまり、ターゲットドメインからソースドメインを照会することはできません。ただし、反対方向で別の接続をセットアップすることは可能です。



ソースドメインは、ターゲットドメインへの "アウトバウンド" 接続を作成します。ターゲットドメインは、ソースドメインから "インバウンド" 接続リクエストを受信します。

接続をセットアップするには

1. ドメインダッシュボードで、ドメインを選択し、[接続] タブに進みます。
2. [アウトバウンド接続] セクションで、[リクエスト] を選択します。

3. [接続のエイリアス] に接続の名前を入力します。
4. AWS アカウント とリージョンのドメインに接続するか、別のアカウントまたはリージョンに接続するかを選択します。

   - AWS アカウント とリージョンのクラスターに接続するには、ドロップダウンメニューからドメインを選択し、リクエスト を選択します。
   - 別の AWS アカウント またはリージョンのクラスターに接続するには、リモートドメインの ARN を選択し、リクエスト を選択します。リージョン間でドメインを接続するには、両方のドメインで Elasticsearch バージョン 7.10 以降または が実行されている必要がありますOpenSearch。
5. クラスタークエリで使用できないクラスターをスキップするには、 [使用できないものをスキップ] を選択します。この設定を行うと、1 つまたは複数のリモートクラスターで障害が発生しても、クラスター間のクエリで部分的な結果が返されるようになります。
6. クラスター間検索では、まず接続リクエストを検証することで、前提条件が満たされているか確認が行われます。ドメインに互換性がないことがわかった場合、接続リクエストは `Validation failed` の状態になります。
7. 接続要求が正常に検証されると、接続リクエストはターゲットドメインに送信され、承認を受けます。この承認が行われるまで、接続は `Pending acceptance` の状態のままです。接続要求がターゲットドメインで承諾されると、状態が `Active` に変わり、ターゲットドメインをクエリに使用できるようになります。

   - ドメインページには、ターゲットドメインについて、ドメインの全体的な正常性とインスタンスの正常性に関する詳細が表示されます。ドメインとの間の接続を柔軟に作成、表示、削除、モニタリングできるのは、ドメイン所有者のみです。

接続が確立されると、接続されたドメインのノード間ではすべてのトラフィックフローが暗号化されます。VPC ドメインを非 VPC ドメインに接続し、非 VPC ドメインがインターネットからのトラフィックを受信できるパブリックエンドポイントである場合も、ドメイン間のクラスター間トラフィックは暗号化され安全です。

接続の削除

接続を削除すると、そのインデックスに対するクラスター間のオペレーションが停止します。

1. ドメインダッシュボードで、[接続] タブに移動します。
2. 削除するドメイン接続を選択し、[削除] を選択したら、削除を確定します。

接続を削除するには、ソースドメインまたはターゲットドメインのいずれかでこれらの手順を実行できます。削除した接続は 15 日間 Deleted ステータスで表示されます。

アクティブなクラスター間接続のあるドメインは削除できません。ドメインを削除するには、まずそのドメインからすべての受信接続と送信接続を削除します。この操作は、ドメインを削除する前にクラスター間ドメインのユーザーを考慮するために行います。

セキュリティのセットアップとサンプルチュートリアル

1. クラスター間検索リクエストをソースドメインに送信します。
2. ソースドメインでは、ドメインアクセスポリシーに照らし合わせてリクエストを評価します。クラスター間検索にはきめ細かなアクセス制御が必要になるため、ソースドメインでオープンアクセスポリシーを使用することをお勧めします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> パスにリモートインデックスを含める場合は、ドメイン ARN で URI を URL エンコードする必要があります。例えば、arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst:remote_index ではなく arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst%3Aremote_index を使用します。

きめ細かなアクセス制御に加えて制限付きアクセスポリシーを使用する場合は、ポリシーでは最低限でも es:ESHttpGet へのアクセスを許可する必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

3. ソースドメインに対するきめ細かなアクセス制御では、要求が次のように評価されます。

   - リクエストが、有効な IAM または HTTP 基本認証情報で署名されているか。
   - その場合、検索を実行し、データにアクセスするためのアクセス許可がユーザーにあるか。

   要求がターゲットドメインのデータのみを検索する場合 (例: dest-alias:dest-index/_search)、ターゲットドメインに対するアクセス許可のみが必要になります。

   両方のドメインを対象にしたデータ検索のリクエストの場合 (例: source-index,dest-alias:dest-index/_search)、両方のドメインに対するアクセス許可が必要です。

   きめ細かなアクセスコントロールでは、ユーザーは、関連するインデックスの標準readまたはindices:admin/shards/search_shardsアクセス許可に加えて、 アクセスsearch許可を持っている必要があります。

4. ソースドメインは、ターゲットドメインにリクエストを渡します。ターゲットドメインでは、このリクエストをドメインアクセスポリシーに照らし合わせて評価します。ターゲットドメインに対する es:ESCrossClusterGet アクセス許可も含める必要があります。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/dst-domain"
    }
  ]
}
```

es:ESCrossClusterGet アクセス許可が /dst-domain/* ではなく /dst-domain に適用されていることを確認してください。

ただし、この最小ポリシーでは、クラスター間検索のみが許可されます。ドキュメントのインデックス作成や標準検索の実行など、他のオペレーションを実行するには、追加のアクセス許可が必要です。ターゲットドメインでは、以下のポリシーの使用をお勧めします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/dst-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/dst-domain"
```

```
    }
  ]
}
```

> **Note**
>
> ドメイン間のクラスター間検索リクエストはすべて、デフォルトで暗号化の一部として転送中に node-to-node 暗号化されます。

5. ターゲットドメインで検索が実行され、結果がソースドメインに返されます。
6. ソースドメインにより、独自の結果 (存在する場合) とターゲットドメインの結果を組み合わせたものが返されます。
7. テストリクエストには、Postman の使用をお勧めします。

   - ターゲットドメインで、ドキュメントのインデックスを作成します。

   ```
   POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1

   {
     "Dracula": "Bram Stoker"
   }
   ```

   - ソースドメインからこのインデックスを照会するには、クエリ内にターゲットドメインの接続エイリアスを含めます。

   ```
   GET https://src-domain.us-east-1.es.amazonaws.com/<connection_alias>:books/
   _search

   {
       ...
     "hits": [
       {
         "_index": "source-destination:books",
         "_type": "_doc",
         "_id": "1",
         "_score": 1,
         "_source": {
           "Dracula": "Bram Stoker"
         }
       }
     ]
   ```

```
}
```

接続エイリアスは、ドメインダッシュボードの [接続] タブで確認できます。

- domain-a -> domain-b の接続を接続エイリアス cluster_b でセットアップし、domain-a -> domain-c 接続を接続エイリアス cluster_c でセットアップする場合は、次のように domain-a、domain-b、および domain-c を検索します。

```
GET https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_search
{
  "query": {
    "match": {
      "user": "domino"
    }
  }
}
```

レスポンス

```
{
  "took": 150,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "failed": 0,
    "skipped": 0
  },
  "_clusters": {
    "total": 3,
    "successful": 3,
    "skipped": 0
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "local_index",
        "_type": "_doc",
        "_id": "0",
```

```
        "_score": 1,
        "_source": {
          "user": "domino",
          "message": "Lets unite the new mutants",
          "likes": 0
        }
      },
      {
        "_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
          "user": "domino",
          "message": "I'm different",
          "likes": 0
        }
      },
      {
        "_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
          "user": "domino",
          "message": "So am I",
          "likes": 0
        }
      }
    ]
  }
}
```

接続のセットアップで使用できないクラスターをスキップしなかった場合、検索リクエストを正常に実行するには、検索するすべての宛先クラスターが使用可能である必要があります。そうでない場合は、リクエスト全体が失敗します ドメインのうち、いずれか 1 つだけが使用できない場合でも、検索結果は返されません。

OpenSearch ダッシュボード

接続された複数ドメインのデータは、単一のドメインの場合と同じ方法で可視化できますが、`connection-alias:index` を使用してリモートインデックスにアクセスする必要があります。インデックスパターンは `connection-alias:index` に一致する必要があります。

Amazon OpenSearch Service のランキングを学ぶ

OpenSearch は、BM-25 と呼ばれる確率的ランキングフレームワークを使用して関連性スコアを計算します。特定のキーワードがドキュメント内で頻繁に表示される場合、BM-25 はそのドキュメントに対して高い関連性スコアを割り当てます。ただし、このフレームワークでは、クリックスルーデータなどのユーザーの行動は考慮されないため、関連性がさらに改善します。

Learning to Rank はオープンソースの OpenSearch プラグインで、機械学習と行動データを使用してドキュメントの関連性を調整できます。XGBoost および Ranklib ライブラリのモデルを使用して、検索結果のリスコアを行います。Elasticsearch LTR プラグインは当初、OpenSource Connections によって開発され、Wikimedia Foundation、Snagajob Engineering、Bonsai、Yelp Engineering によって多大な貢献をしました。プラグイン OpenSearch のバージョンは、Elasticsearch LTR プラグインから派生しています。

Learning to Rank には、 OpenSearch または Elasticsearch 7.7 以降が必要です。Learning to Rank プラグインを使用するには、完全な管理者許可が必要です。詳細については、「the section called “マスターユーザーの変更”」を参照してください。

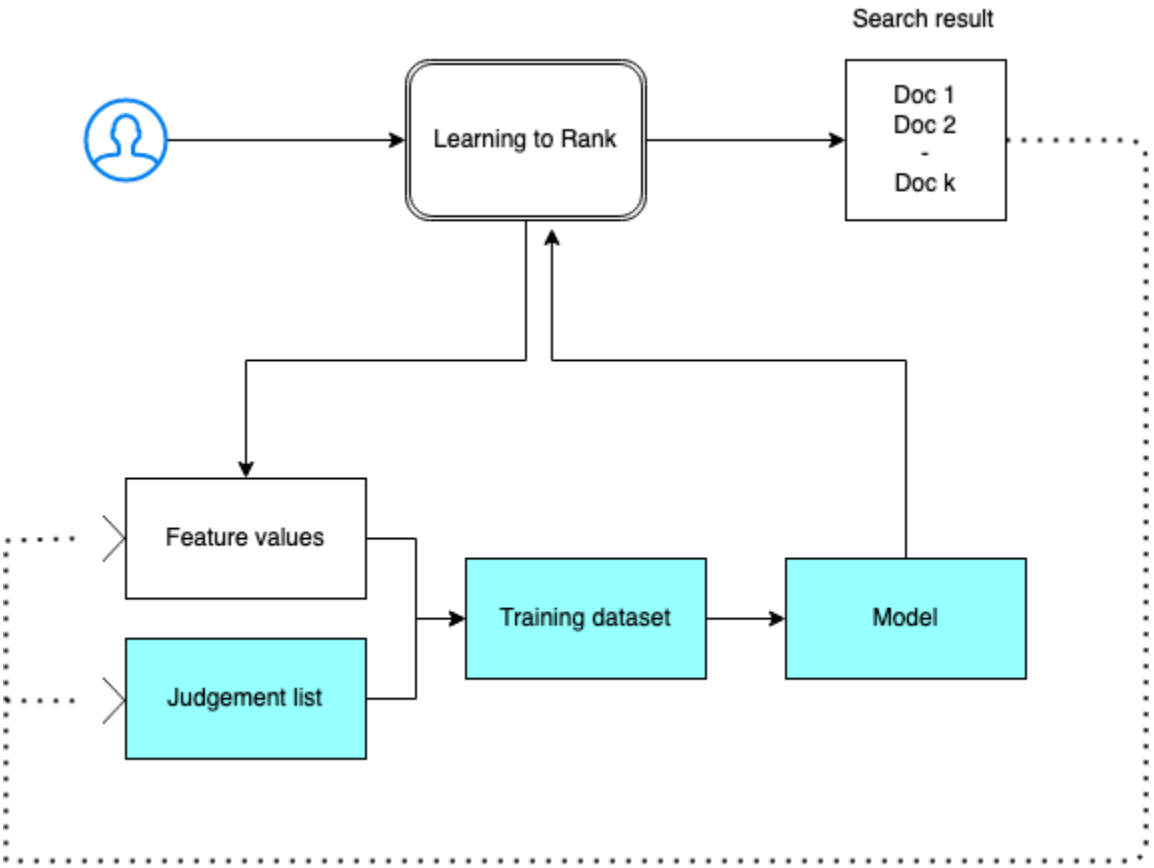
> **Note**
>
> このドキュメントでは、 Learning to Rank プラグインの一般的な概要を説明し、使用を開始するのに役立ちます。詳細なステップや API の説明など、完全なドキュメントについては、Learning to Rank ドキュメントで入手可能です。

トピック

- Learning to Rank を開始する
- Learning to Rank API

Learning to Rank を開始する

判断リストを提供し、トレーニングデータセットを準備し、Amazon OpenSearch Service の外部でモデルをトレーニングする必要があります。青色の部分は、 OpenSearch Service の外部で発生します。



ステップ 1: プラグインを初期化する

Learning to Rank プラグインを初期化するには、次のリクエストを OpenSearch サービスドメインに送信します。

```
PUT _ltr
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : ".ltrstore"
}
```

このコマンドは、機能セットやモデルなどのメタデータ情報を保存する非表示の `.ltrstore` インデックスを作成します。

ステップ 2: 判断リストを作成する

> **Note**
>
> このステップは、 OpenSearch サービスの外部で実行する必要があります。

判断リストは、機械学習モデルが学習する例を集めたものです。判断リストには、自分にとって重要なキーワードと、各キーワードのグレード付けされたドキュメントのセットを含める必要があります。

この例では、ムービーデータセットの判断リストがあります。4 のグレードは完全一致を示します。0 のグレードは最も悪い一致を示します。

グレード	キーワード	ドキュメントID	ムービー名
4	rambo	7555	Rambo
3	rambo	1370	Rambo III
3	rambo	1369	Rambo: First Blood Part II
3	rambo	1368	First Blood

判断リストを以下の形式で準備します。

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood

where qid:1 represents "rambo"
```

判定リストのより完全な例については、「映画の判断」を参照してください。。

この判断リストは、人間の注釈者の助けを借りて手動で作成することも、分析データからプログラムで推測することもできます。

ステップ 3: 機能セットを構築する

機能とは、ドキュメントの関連性に対応するフィールドです。例えば、title、overview、popularity score (再生数) などが表示されます。

各機能の Mustache テンプレートを用いて機能セットを構築します。機能の詳細については、「機能を用いた操作」を参照してください。

この例では、title および overview フィールドを用いて movie_features 機能セットを構築します

```
POST _ltr/_featureset/movie_features
{
  "featureset" : {
      "name" : "movie_features",
      "features" : [
        {
          "name" : "1",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "title" : "{{keywords}}"
            }
          }
        },
        {
          "name" : "2",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "overview" : "{{keywords}}"
            }
          }
        }
```

```
      ]
    }
}
```

元の .ltrstore インデックスのクエリを行うと、機能セットが返されます。

```
GET _ltr/_featureset
```

ステップ 4: 機能値のログを取る

機能値は、各機能について BM-25 によって計算された関連性スコアです。

機能セットと判断リストを組み合わせて、機能値のログを取ります。機能のログギングの詳細については、「機能のログギングスコア」を参照してください。

この例では、bool クエリは、フィルターを用いてグレード付けされたドキュメントを取得してから、sltr クエリを用いて機能セットを選択します。ltr_log クエリは、ドキュメントと機能を組み合わせて、対応する機能値のログを取ります。

```
POST tmdb/_search
{
  "_source": {
    "includes": [
      "title",
      "overview"
    ]
  },
  "query": {
    "bool": {
      "filter": [
        {
          "terms": {
            "_id": [
              "7555",
              "1370",
              "1369",
              "1368"
            ]
          }
        },
        {
          "sltr": {
```

```
            "_name": "logged_featureset",
            "featureset": "movie_features",
            "params": {
              "keywords": "rambo"
            }
          }
        }
      ]
    }
  },
  "ext": {
    "ltr_log": {
      "log_specs": {
        "name": "log_entry1",
        "named_query": "logged_featureset"
      }
    }
  }
}
```

以下にレスポンスの例を示します。

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : 0.0,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 0.0,
```

```
      "_source" : {
        "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
        "title" : "First Blood"
      },
      "fields" : {
        "_ltrlog" : [
          {
            "log_entry1" : [
              {
                "name" : "1"
              },
              {
                "name" : "2",
                "value" : 10.558305
              }
            ]
          }
        ]
      },
      "matched_queries" : [
        "logged_featureset"
      ]
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "7555",
      "_score" : 0.0,
      "_source" : {
        "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      },
      "fields" : {
        "_ltrlog" : [
          {
            "log_entry1" : [
              {
```

```
                  "name" : "1",
                  "value" : 11.2569065
                },
                {
                  "name" : "2",
                  "value" : 9.936821
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 6.334839
                },
                {
                  "name" : "2",
                  "value" : 10.558305
                }
              ]
            }
          ]
        },
        "matched_queries" : [
```

```
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
 find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
 his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
 when Trautman is captured.",
          "title" : "Rambo III"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 9.425955
                },
                {
                  "name" : "2",
                  "value" : 11.262714
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      }
    ]
  }
}
```

前の例では、1 番目の機能には機能値がありません。これは、1368 に等しい ID が付いたドキュメントのタイトルフィールドに「rambo」というキーワードが表示されていないからです。これは、トレーニングデータで欠落している機能値です。

ステップ 2: トレーニングデータセットを作成する

> **Note**
>
> このステップは、 OpenSearch サービスの外部で実行する必要があります。

次のステップでは、判断リストと機能値を組み合わせて、トレーニングデータセットを作成します。元の判断リストは次のようになります。

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood
```

それを最終的なトレーニングデータセットに変換します。これは次のようになります。

```
4 qid:1 1:12.318474 2:10.573917 # 7555 rambo
3 qid:1 1:10.357875 2:11.950391 # 1370 rambo
3 qid:1 1:7.010513 2:11.220095 # 1369 rambo
3 qid:1 1:0.0 2:11.220095 # 1368 rambo
```

このステップは手動で実行することも、プログラムを記述して自動化することもできます。

ステップ6：アルゴリズムを選択してモデルを構築する

> **Note**
>
> このステップは、 OpenSearch サービスの外部で実行する必要があります。

トレーニングデータセットを配置したら、次のステップは XGBoost ライブラリまたは Ranklib ライブラリを使用してモデルを構築することです。XgBoost ライブラリと Ranklib ライブラリを使用すると、Lambdamart や Random Forest などの人気モデルを構築できます。

XGBoost と Ranklib を使用してモデルを構築する手順については、それぞれ XGBoost とのRankLibドキュメントを参照してください。Amazon を使用して XGBoost モデル SageMaker を構築するには、XGBoost アルゴリズム」を参照してください。

ステップ 7: モデルをデプロイする

モデルを構築したら、それを Learning to Rank プラグインに展開します。モデルのデプロイの詳細については、「トレーニングされたモデルのアップロード」を参照してください。。

この例では、Ranklib ライブラリを使用して `my_ranklib_model` モデルを構築します。

```
POST _ltr/_featureset/movie_features/_createmodel?pretty
{
  "model": {
    "name": "my_ranklib_model",
    "model": {
      "type": "model/ranklib",
      "definition": """## LambdaMART
## No. of trees = 10
## No. of leaves = 10
## No. of threshold candidates = 256
## Learning rate = 0.1
## Stop early = 100

<ensemble>
   <tree id="1" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-2.0</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-2.0</output>
               </split>
               <split pos="right">
                  <output>-2.0</output>
               </split>
            </split>
         </split>
         <split pos="right">
```

```
            <output>2.0</output>
         </split>
      </split>
   </tree>
   <tree id="2" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.67031991481781</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.67031991481781</output>
               </split>
               <split pos="right">
                  <output>-1.6703200340270996</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.6703201532363892</output>
         </split>
      </split>
   </tree>
   <tree id="3" weight="0.1">
      <split>
         <feature>2</feature>
         <threshold>10.573917</threshold>
         <split pos="left">
            <output>1.479954481124878</output>
         </split>
         <split pos="right">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
```

```
                <output>-1.4799546003341675</output>
             </split>
             <split pos="right">
                <output>-1.479954481124878</output>
             </split>
          </split>
          <split pos="right">
             <output>-1.479954481124878</output>
          </split>
       </split>
    </split>
 </tree>
 <tree id="4" weight="0.1">
    <split>
       <feature>1</feature>
       <threshold>10.357875</threshold>
       <split pos="left">
          <feature>1</feature>
          <threshold>0.0</threshold>
          <split pos="left">
             <output>-1.3569872379302979</output>
          </split>
          <split pos="right">
             <feature>1</feature>
             <threshold>7.010513</threshold>
             <split pos="left">
                <output>-1.3569872379302979</output>
             </split>
             <split pos="right">
                <output>-1.3569872379302979</output>
             </split>
          </split>
       </split>
       <split pos="right">
          <output>1.3569873571395874</output>
       </split>
    </split>
 </tree>
 <tree id="5" weight="0.1">
    <split>
       <feature>1</feature>
       <threshold>10.357875</threshold>
       <split pos="left">
          <feature>1</feature>
```

```
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.2721362113952637</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.2721363306045532</output>
               </split>
               <split pos="right">
                  <output>-1.2721363306045532</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.2721362113952637</output>
         </split>
      </split>
   </tree>
   <tree id="6" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.2110036611557007</output>
               </split>
               <split pos="right">
                  <output>-1.2110036611557007</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.2110037803649902</output>
            </split>
         </split>
         <split pos="right">
            <output>1.2110037803649902</output>
         </split>
```

```
      </split>
   </tree>
   <tree id="7" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.165616512298584</output>
               </split>
               <split pos="right">
                  <output>-1.165616512298584</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.165616512298584</output>
            </split>
         </split>
         <split pos="right">
            <output>1.165616512298584</output>
         </split>
      </split>
   </tree>
   <tree id="8" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.131177544593811</output>
               </split>
               <split pos="right">
                  <output>-1.131177544593811</output>
               </split>
```

```
            </split>
            <split pos="right">
               <output>-1.131177544593811</output>
            </split>
         </split>
         <split pos="right">
            <output>1.131177544593811</output>
         </split>
      </split>
   </tree>
   <tree id="9" weight="0.1">
      <split>
         <feature>2</feature>
         <threshold>10.573917</threshold>
         <split pos="left">
            <output>1.1046180725097656</output>
         </split>
         <split pos="right">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.1046180725097656</output>
               </split>
               <split pos="right">
                  <output>-1.1046180725097656</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.1046180725097656</output>
            </split>
         </split>
      </split>
   </tree>
   <tree id="10" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
```

```
                <feature>1</feature>
                <threshold>0.0</threshold>
                <split pos="left">
                   <output>-1.0838804244995117</output>
                </split>
                <split pos="right">
                   <output>-1.0838804244995117</output>
                </split>
             </split>
             <split pos="right">
                <output>-1.0838804244995117</output>
             </split>
          </split>
          <split pos="right">
             <output>1.0838804244995117</output>
          </split>
       </split>
    </tree>
</ensemble>
"""
    }
  }
}
```

モデルを表示するには、次のリクエストを送信します。

```
GET _ltr/_model/my_ranklib_model
```

ステップ 8: Learning to Rank を用いて検索する

モデルをデプロイしたら、検索する準備ができています。

使用する機能と実行するモデルの名前を用いて `sltr` クエリを実行します。

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
  },
  "query": {
    "multi_match": {
      "query": "rambo",
      "fields": ["title", "overview"]
```

```
    }
  },
  "rescore": {
    "query": {
      "rescore_query": {
        "sltr": {
          "params": {
            "keywords": "rambo"
          },
          "model": "my_ranklib_model"
        }
      }
    }
  }
}
```

Learning to Rank では、判断リストの最高グレードを割り当てたため、「Rambo」が最初の結果として表示されます。

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 7,
      "relation" : "eq"
    },
    "max_score" : 13.096414,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "7555",
        "_score" : 13.096414,
        "_source" : {
          "overview" : "When governments fail to act on behalf of captive missionaries,
 ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
```

```
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1370",
      "_score" : 11.17245,
      "_source" : {
        "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
        "title" : "Rambo III"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1368",
      "_score" : 10.442155,
      "_source" : {
        "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
        "title" : "First Blood"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1369",
      "_score" : 10.442155,
      "_source" : {
        "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
        "title" : "Rambo: First Blood Part II"
      }
```

```
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "31362",
      "_score" : 7.424202,
      "_source" : {
        "overview" : "It is 1985, and a small, tranquil Florida town is being rocked
by a wave of vicious serial murders and bank robberies. Particularly sickening to the
authorities is the gratuitous use of violence by two “Rambo” like killers who dress
themselves in military garb. Based on actual events taken from FBI files, the movie
depicts the Bureau’s efforts to track down these renegades.",
        "title" : "In the Line of Duty: The F.B.I. Murders"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "13258",
      "_score" : 6.43182,
      "_source" : {
        "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
his family's stifling home life when he encounters Lee Carter (Will Poulter), the
school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
to make cinematic history by filming his own action-packed video epic. Together, these
two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
– and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
own!""",
        "title" : "Son of Rambow"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "61410",
      "_score" : 3.9719706,
      "_source" : {
        "overview" : "It's South Africa 1990. Two major events are about to happen:
The release of Nelson Mandela and, more importantly, it's Spud Milton's first year
at an elite boys only private boarding school. John Milton is a boy from an ordinary
background who wins a scholarship to a private school in Kwazulu-Natal, South Africa.
Surrounded by boys with nicknames like Gecko, Rambo, Rain Man and Mad Dog, Spud has
his hands full trying to adapt to his new home. Along the way Spud takes his first
tentative steps along the path to manhood. (The path it seems could be a rather long
```

```
 road). Spud is an only child. He is cursed with parents from well beyond the lunatic
 fringe and a senile granny. His dad is a fervent anti-communist who is paranoid that
 the family domestic worker is running a shebeen from her room at the back of the
 family home. His mom is a free spirit and a teenager's worst nightmare, whether it's
 shopping for Spud's underwear in the local supermarket",
          "title" : "Spud"
        }
      }
    ]
  }
}
```

Learning to Rank プラグインを使用せずに検索すると、 は異なる結果 OpenSearch を返します。

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
  },
  "query": {
    "multi_match": {
      "query": "Rambo",
      "fields": ["title", "overview"]
    }
  }
}
```

```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 5,
      "relation" : "eq"
    },
    "max_score" : 11.262714,
    "hits" : [
```

```
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1370",
      "_score" : 11.262714,
      "_source" : {
        "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
        "title" : "Rambo III"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "7555",
      "_score" : 11.2569065,
      "_source" : {
        "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
        "title" : "Rambo"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1368",
      "_score" : 10.558305,
      "_source" : {
        "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
        "title" : "First Blood"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1369",
```

```
        "_score" : 10.558305,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
 secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
 rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
 life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "13258",
        "_score" : 6.4600153,
        "_source" : {
          "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
 his family's stifling home life when he encounters Lee Carter (Will Poulter), the
 school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
 to make cinematic history by filming his own action-packed video epic. Together, these
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 – and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      }
    ]
  }
}
```

モデルがどの程度うまく実行されているかに基づいて、判断リストと機能を調整します。次に、ステップ 2~8 を繰り返して、時間の経過とともにランキング結果を改善します。

Learning to Rank API

Learning to Rank オペレーションを使用して、機能セットとモデルをプログラムで操作します。

ストアを作成する

機能セットやモデルなどのメタデータ情報を保存する非表示の `.ltrstore` インデックスを作成します。

```
PUT _ltr
```

ストアを削除する

非表示の .ltrstore インデックスを削除し、プラグインをリセットします。

```
DELETE _ltr
```

機能セットを作成する

機能セットを作成します。

```
POST _ltr/_featureset/<name_of_features>
```

機能セットを削除する

機能セットを削除します。

```
DELETE _ltr/_featureset/<name_of_feature_set>
```

機能セットを取得する

機能セットを取得します。

```
GET _ltr/_featureset/<name_of_feature_set>
```

モデルを作成する

モデルを作成します。

```
POST _ltr/_featureset/<name_of_feature_set>/_createmodel
```

モデルを削除する

モデルを削除します。

```
DELETE _ltr/_model/<name_of_model>
```

モデルを取得する

モデルを取得します。

```
GET _ltr/_model/<name_of_model>
```

統計を取得する

プラグインがどのように動作するかについての情報を提供します。

```
GET _ltr/_stats
```

また、フィルターを使用して単一の統計を取得することもできます:

```
GET _ltr/_stats/<stat>
```

さらに、情報をクラスター内の単一のノードに制限することもできます。

```
GET _ltr/_stats/<stat>/nodes/<nodeId>

{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "873043598401:ltr-77",
  "stores" : {
    ".ltrstore" : {
      "model_count" : 1,
      "featureset_count" : 1,
      "feature_count" : 2,
      "status" : "green"
    }
  },
  "status" : "green",
  "nodes" : {
    "DjelK-_ZSfyzstO5dhGGQA" : {
      "cache" : {
        "feature" : {
          "eviction_count" : 0,
          "miss_count" : 0,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
```

```
        },
        "featureset" : {
          "eviction_count" : 2,
          "miss_count" : 2,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
        },
        "model" : {
          "eviction_count" : 2,
          "miss_count" : 3,
          "entry_count" : 1,
          "memory_usage_in_bytes" : 3204,
          "hit_count" : 1
        }
      },
      "request_total_count" : 6,
      "request_error_count" : 0
    }
  }
}
```

統計は、次のテーブルに示すように、ノードとクラスターの 2 つのレベルで提供されます。

ノードレベルの統計

フィールド名	説明
request_total_count	ランク付けリクエストの合計数。
request_error_count	失敗したリクエストの合計数。
cache	すべてのキャッシュ (機能、機能セット、モデル) の統計情報。キャッシュヒットは、ユーザーがプラグインのクエリを行い、モデルがすでにメモリにロードされているときに発生します。
cache.eviction_count	キャッシュ削除の数。
cache.hit_count	キャッシュヒットの数。

フィールド名	説明
cache.miss_count	キャッシュミスの数。キャッシュミスは、ユーザーがプラグインのクエリを行い、モデルがまだメモリにロードされていない場合に発生します。
cache.entry_count	キャッシュのエントリの数。
cache.memory_usage_in_bytes	使用メモリの合計 (バイト単位)。
cache.cache_capacity_reached	キャッシュ制限に達したかどうかを示します。

クラスターレベルの統計

フィールド名	説明
保存	機能セットとモデルメタデータが保存される場所を示します。(デフォルトは「.ltrstore」です。 それ以外の場合は、ユーザーが指定した名前で、接頭辞に「.ltrstore_」が付きます)。
stores.status	インデックスのステータス。
stores.feature_sets	機能セットの数
stores.features_count	機能の数。
stores.model_count	モデルの数。
ステータス	Feature Store インデックスのステータス (赤、黄、緑) およびサーキットブレーカーの状態 (開または閉) に基づくプラグインのステータス。
cache.cache_capacity_reached	キャッシュ制限に達したかどうかを示します。

キャッシュ統計を取得

キャッシュとメモリの使用状況についての統計を返します。

```
GET _ltr/_cachestats

{
    "_nodes": {
        "total": 2,
        "successful": 2,
        "failed": 0
    },
    "cluster_name": "opensearch-cluster",
    "all": {
        "total": {
            "ram": 612,
            "count": 1
        },
        "features": {
            "ram": 0,
            "count": 0
        },
        "featuresets": {
            "ram": 612,
            "count": 1
        },
        "models": {
            "ram": 0,
            "count": 0
        }
    },
    "stores": {
        ".ltrstore": {
            "total": {
                "ram": 612,
                "count": 1
            },
            "features": {
                "ram": 0,
                "count": 0
            },
            "featuresets": {
                "ram": 612,
                "count": 1
```

```
            },
            "models": {
                "ram": 0,
                "count": 0
            }
        }
    },
    "nodes": {
        "ejF6uutERF20wOFNOXB61A": {
            "name": "opensearch1",
            "hostname": "172.18.0.4",
            "stats": {
                "total": {
                    "ram": 612,
                    "count": 1
                },
                "features": {
                    "ram": 0,
                    "count": 0
                },
                "featuresets": {
                    "ram": 612,
                    "count": 1
                },
                "models": {
                    "ram": 0,
                    "count": 0
                }
            }
        },
        "Z2RZNWRLSveVcz2c6lHf5A": {
            "name": "opensearch2",
            "hostname": "172.18.0.2",
            "stats": {
                ...
            }
        }
    }
}
```

キャッシュをクリア

プラグインキャッシュをクリアします。これを使用して、モデルをリフレッシュします。

```
POST _ltr/_clearcache
```

Amazon OpenSearch Service での非同期検索

Amazon OpenSearch Service の非同期検索を使用すると、バックグラウンドで実行される検索クエリを送信し、リクエストの進行状況をモニタリングして、後の段階で結果を取得できます。検索が完了する前に部分的な結果が取得できるようになります。検索が終了したら、後で取得および分析できるように結果を保存します。

非同期検索には OpenSearch 1.0 以降、または Elasticsearch 7.10 以降が必要です。

このドキュメントでは、非同期検索の概要を説明します。また、オープンソース OpenSearch クラスターではなくマネージド Amazon OpenSearch Service ドメインで非同期検索を使用する際の制限についても説明します。使用可能な設定、アクセス許可、完全な API リファレンスなど、非同期検索の完全なドキュメントについては、 OpenSearch ドキュメントの「非同期検索」を参照してください。

サンプルの検索コール

非同期検索を実行するには、以下の形式を使用して HTTP リクエストを `_plugins/_asynchronous_search` に送信します。

```
POST opensearch-domain/_plugins/_asynchronous_search
```

> **Note**
>
> OpenSearch バージョンの代わりに Elasticsearch 7.10 を使用している場合は、すべての非同期検索リクエスト_opendistroで _plugins を に置き換えます。

次の非同期検索オプションを指定できます。

オプション	説明	デフォルト値	必須
wait_for_completio	結果を待機する予定の時間を指定します。通常の検索と同様に、この時間内に得られるどの	1 秒	いいえ

オプション	説明	デフォルト値	必須
n_timeout	ような結果も確認できます。ID に基づいて残りの結果をポーリングできます。最大値は 300 秒です。		
keep_on_completion	検索の完了後に結果をクラスターに保存するかどうかを指定します。保存された結果は、後で確認できます。	false	いいえ
keep_alive	結果がクラスターに保存される時間を指定します。例えば、2d は、結果が 48 時間クラスターに保存されることを意味します。保存された検索結果は、この期間の後、または検索がキャンセルされた場合、削除されます。これにはクエリランタイムが含まれることに注意してください。この時間にクエリがオーバーランすると、プロセスではこのクエリが自動的にキャンセルされます。	12 時間	いいえ

リクエスト例

```
POST _plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=1ms&keep_on_completion=true&request_cache=false
{
  "aggs": {
    "city": {
      "terms": {
        "field": "city",
        "size": 10
      }
    }
  }
}
```

> **Note**
>
> 標準 _search クエリに適用されるすべてのリクエストパラメータがサポートされています。 OpenSearch バージョンの代わりに Elasticsearch 7.10 を使用している場合は、 を _pluginsに置き換えます_opendistro。

非同期検索アクセス許可

非同期検索は、きめ細かなアクセスコントロールをサポートしています。ユースケースに適合する許可のミキシングとマッチングの詳細については、非同期検索セキュリティを参照してください。

きめ細かなアクセスコントロールが有効になっているドメインの場合は、ロールに対して以下の最小限の許可が必要です。

```
# Allows users to use all asynchronous search functionality
asynchronous_search_full_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/*'
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - 'indices:data/read/search*'

# Allows users to read stored asynchronous search results
asynchronous_search_read_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/get'
```

きめ細かなアクセスコントロールが無効になっているドメインの場合は、IAM アクセスとシークレットキーを使用してすべてのリクエストに署名します。非同期検索 ID を用いて結果にアクセスできます。

非同期検索設定

OpenSearch では、 _cluster/settings API を使用して、使用可能なすべての非同期検索設定を変更できます。 OpenSearch サービスでは、次の設定のみ変更できます。

- plugins.asynchronous_search.node_concurrent_running_searches
- plugins.asynchronous_search.persist_search_failures

クラスター間検索

クラスター間で非同期検索を実行できますが、次の小さな制限があります。

- 非同期検索は、ソースドメインでのみ実行できます。
- クロスクラスター検索クエリの一部としてネットワークラウンドトリップを最小化することはできません。

接続エイリアス cluster_b を用いた domain-a -> domain-b と接続エイリアス cluster_c を用いた domain-a -> domain-c の間の接続をセットアップする場合は、次のように domain-a、domain-b、および domain-c を非同期検索します。

```
POST https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=500ms&keep_on_completion=true&request_cache=false
{
  "size": 0,
  "_source": {
    "excludes": []
  },
  "aggs": {
    "2": {
      "terms": {
        "field": "clientip",
        "size": 50,
        "order": {
          "_count": "desc"
        }
      }
    }
  },
  "stored_fields": [
    "*"
  ],
  "script_fields": {},
  "docvalue_fields": [
    "@timestamp"
```

```
  ],
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "status:404",
            "analyze_wildcard": true,
            "default_field": "*"
          }
        },
        {
          "range": {
            "@timestamp": {
              "gte": 1483747200000,
              "lte": 1488326400000,
              "format": "epoch_millis"
            }
          }
        }
      ],
      "filter": [],
      "should": [],
      "must_not": []
    }
  }
}
```

レスポンス

```
{
  "id" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAB",
  "state" : "RUNNING",
  "start_time_in_millis" : 1609329314796,
  "expiration_time_in_millis" : 1609761314796
}
```

詳細については、「the section called “クラスター間検索”」を参照してください。

UltraWarm

UltraWarm インデックスを使用した非同期検索は引き続き機能します。詳細については、「the section called “UltraWarm ストレージ”」を参照してください。

> **Note**
>
> で非同期検索統計をモニタリングできます CloudWatch。メトリクスの一覧については、「the section called “非同期検索メトリクス”」を参照してください。

Amazon OpenSearch Service でのポイントインタイム検索

Point in Time (PIT) は、時間に固定されたデータセットに対して異なるクエリを実行できるようにする検索の一種です。ドキュメントは絶えずインデックス付けされ、更新され、削除されているため、同じインデックスに対して同じクエリを異なる時点で実行すると、通常は異なる結果が返されます。PIT では、一定の状態に保たれたデータセット対して、クエリを実行できます。

PIT 検索の主な用途は、それを`search_after`機能と組み合わせることです。これは OpenSearch、特にディープページ分割の場合、 で推奨されるページ分割方法です。これは、時間的にフリーズされたデータセットで動作し、クエリにバインドされず、前後に一貫したページ分割をサポートしているためです。PIT は、 OpenSearch バージョン 2.5 を実行しているドメインで使用できます。

> **Note**
>
> このトピックでは、PIT の概要と、セルフマネージド OpenSearch クラスターではなくマネージド Amazon OpenSearch Service ドメインで PIT を使用する際に考慮すべき点について説明します。包括的な API リファレンスを含む PIT の完全なドキュメントについては、オープンソース OpenSearch ドキュメントの「ポイントインタイム」を参照してください。

考慮事項

PIT　検索を設定するときは、以下を考慮します。

- OpenSearch バージョン 2.3 を実行しているドメインからアップグレードしていて、PIT アクションに対してきめ細かなアクセスコントロールが必要な場合は、これらのアクションとロールを手動で追加する必要があります。

- PIT には回復力はありません。ノードの再起動、ノードの終了、ブルー/グリーンデプロイ、および OpenSearch プロセスの再起動により、すべての PIT データが失われます。
- ブルー/グリーンデプロイ中にシャードが再配置された場合、新しいノードに転送されるのはライブデータのセグメントのみです。PIT が保持していたシャードのセグメント (独占的に保持していたものとライブデータと共有されたものの両方) は、古いノードに残ります。
- PIT 検索は現在、非同期検索では使用できません。

PIT の作成

PIT クエリを実行するには、次の形式_search/point_in_timeを使用して HTTP リクエストを に送信します。

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

次の PIT オプションを指定できます。

オプション	説明	デフォルト値	必須
keep_alive	PIT を維持する時間。検索リクエストで PIT にアクセスするたびに、PIT の有効期間が keep_alive パラメータと同じ時間だけ延長されます。このクエリパラメータは、PIT を作成するときは必須ですが、検索リクエストでは任意となります。		はい
preference	検索の実行に使用されるノードまたはシャードを指定する文字列。	ランダム	いいえ
routing	検索リクエストを特定のシャードにルーティングするよう指定する文字列。	ドキュメントの _id	いいえ
expand_wildcards	ワイルドカードパターンと一致するインデックスのタイプを指定する文字列。コンマ区切り値をサポート。有効な値は以下のとおりです。 • all: 任意のインデックスまたはデータストリームと一致。非表示のものを含む。	open	いいえ

オプション	説明	デフォルト値	必須
	• open: 開いているインデックス、非表示でないインデックス、非表示でないデータストリームと一致。 • closed: 閉じているインデックス、非表示でないインデックス、非表示でないデータストリームと一致。 • hidden: 非表示のインデックスまたはデータストリームと一致。開いているものか閉じているもの、または、開いているものと閉じているもの両方と組み合わせねばならない。 • none: ワイルドカードのパターンは使用できない。		
allow_partial_pit_creation	部分的な障害を含む PIT を作成するかどうかを指定するブール値。	true	いいえ

レスポンス例

```
{
    "pit_id":
 "o463QQEPbXktaW5kZXgtMDAwMDAxFnNOWU43ckt3U3IyaFVpbGE1UWEtMncAFjFyeXBsRGJmVFM2RTB6eVg1aVVqQncAA
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "creation_time": 1658146050064
}
```

PIT を作成すると、PIT ID が届きます。これは、PIT を使って検索するときに使用する ID です。

ポイントインタイムアクセス許可

PIT は、詳細なアクセス制御をサポートしています。 OpenSearch バージョン 2.5 ドメインにアップグレードし、きめ細かなアクセスコントロールが必要な場合は、次のアクセス許可を持つロールを手動で作成する必要があります。

```
# Allows users to use all point in time search search functionality
point_in_time_full_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/point_in_time/readall"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"


# Allows users to use point in time search search functionality for specific index
# All type operations like list all PITs, delete all PITs are not supported in this
 case

point_in_time_index_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - 'my-index-1'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"
```

OpenSearch バージョン 2.5 以降のドメインでは、組み込み`point_in_time_full_access`ロールを使用できます。詳細については、 OpenSearch ドキュメントの「セキュリティモデル」を参照してください。

PIT の設定

OpenSearch では、 `_cluster/settings` API を使用して使用可能なすべての PIT 設定を変更できます。 OpenSearch サービスでは、現在設定を変更できません。

クラスター間検索

PIT の作成、PIT ID を使った検索、PIT の一覧表示、クラスター全体での PIT の削除には、次のような軽微の制限があります。

- PIT の一覧を表示したり、すべての PIT を削除したりできるのは、ソースドメイン上でのみです。
- クロスクラスター検索クエリの一部としてネットワークラウンドトリップを最小化することはできません。

詳細については、「the section called “クラスター間検索”」を参照してください。

UltraWarm

UltraWarm インデックスを使用した PIT 検索は引き続き機能します。詳細については、「the section called “UltraWarm ストレージ”」を参照してください。

> **Note**
>
> で PIT 検索統計をモニタリングできます CloudWatch。メトリクスの一覧については、「the section called “ポイントインタイムメトリクス”」を参照してください。

Amazon OpenSearch Service でのセマンティック検索

OpenSearch バージョン 2.9 以降では、セマンティック検索を使用して検索クエリを理解し、検索の関連性を向上させることができます。セマンティック検索は、ニューラル検索と k-Narest Neighbor (k-NN) 検索の 2 つの方法のいずれかで使用できます。

Service を使用すると OpenSearch 、 および外部サービス の AI コネクタ AWS のサービスを設定できます。コンソールを使用して、 AWS CloudFormation テンプレートで ML モデルを作成することもできます。詳細については、「the section called “CloudFormation テンプレート統合”」を参照してください。

セマンティック検索を使用するための step-by-step ガイドを含むセマンティック検索の完全なドキュメントについては、オープンソース OpenSearch ドキュメントの「セマンティック検索」を参照してください。

Amazon OpenSearch Service での同時セグメント検索

OpenSearch バージョン 2.13 以降では、同時セグメント検索を使用して、クエリフェーズ中にセグメントを並行して検索できます。同時セグメント検索の完全なドキュメントについては、オープンソース OpenSearch ドキュメントの「同時セグメント検索」を参照してください。同時セグメント検索に関連する Amazon CloudWatch メトリクスの詳細については、「インスタンスメトリクスとUltraWarm メトリクス」を参照してください。

Amazon OpenSearch Service で現在のセグメント検索を使用する場合、いくつかの追加の制限が適用されます。

- OpenSearch サービスのインデックスレベルで同時セグメント検索を有効にすることはできません。
- デフォルトでは、 OpenSearch Service は最大スライス数メカニズムで 2 つのスライス数を使用します。

Amazon OpenSearch Service での OpenSearch Dashboards の使用

OpenSearch Dashboards は、 で動作するように設計されたオープンソースの視覚化ツールです OpenSearch。Amazon OpenSearch Service は、すべての OpenSearch サービスドメインで OpenSearch Dashboards をインストールします。 OpenSearch ダッシュボードは、ドメイン内のホットデータノードで実行されます。

ダッシュボードへのリンクは、 OpenSearch サービスコンソール OpenSearch のドメインダッシュボードにあります。を実行しているドメインの場合 OpenSearch、URL は です*domain-endpoint*/_dashboards/。レガシー Elasticsearch を実行しているドメインの場合、URL は です*domain-endpoint*/_plugin/kibana。

このデフォルトの OpenSearch Dashboards インストールを使用するクエリのタイムアウトは 300 秒です。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service のコンテキストにおける OpenSearch Dashboards について説明します。これには、ダッシュボードに接続するさまざまな方法が含まれます。入門ガイド、ダッシュボードの作成手順、ダッシュボード管理、DQL (Dashboards Query Language) などの包括的なドキュメントについては、オープンソース OpenSearch ドキュメントの「OpenSearch Dashboards」を参照してください。

以下のセクションでは、 OpenSearch Dashboards の一般的なユースケースについて説明します。

- the section called “ OpenSearch Dashboards へのアクセスの制御”
- the section called “WMS マップサーバーを使用するように OpenSearch Dashboards を設定する”
- the section called “ローカル Dashboards サーバーを OpenSearch サービスに接続する”

OpenSearch Dashboards へのアクセスの制御

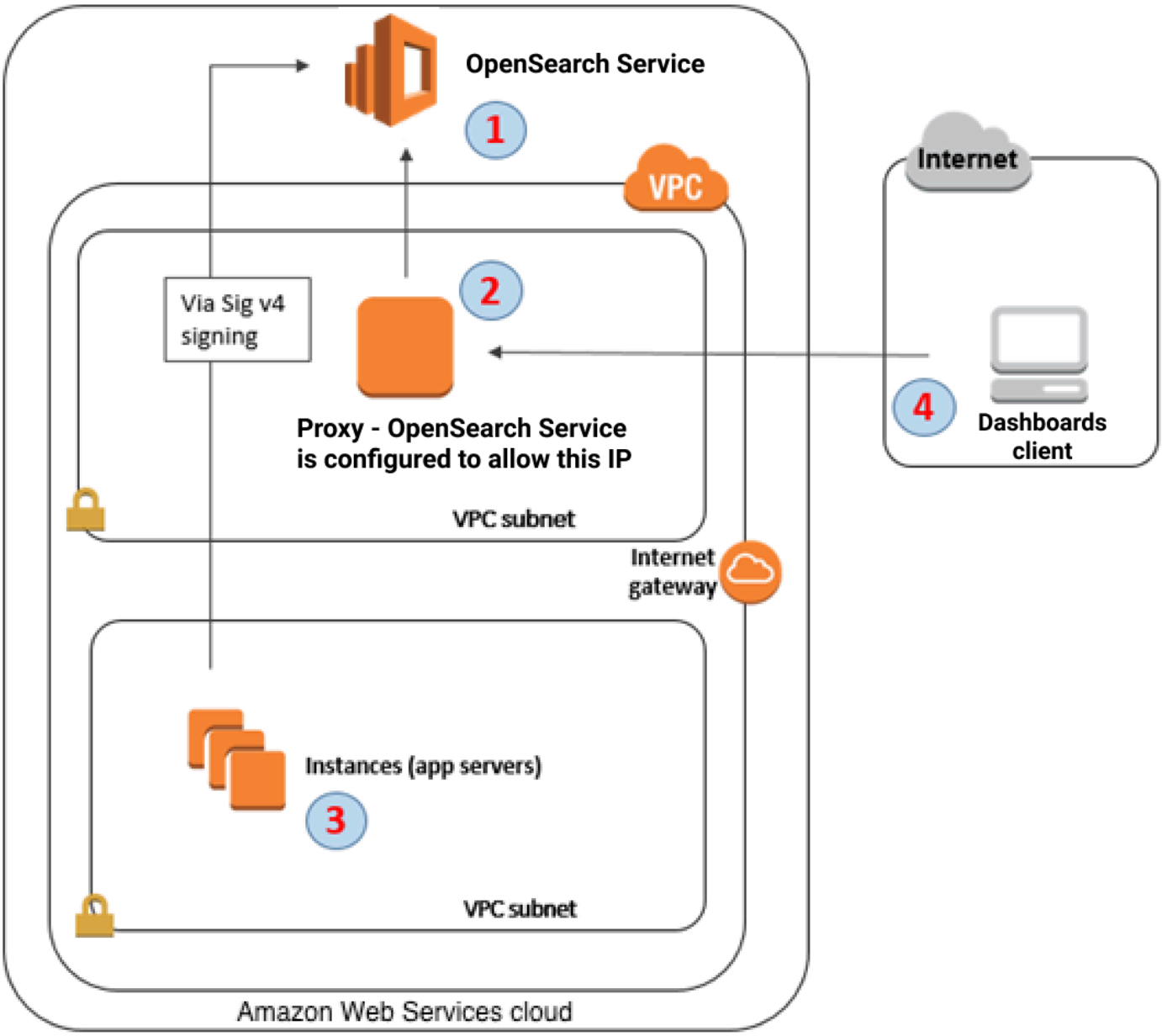
Dashboards は IAM ユーザーとロールをネイティブにサポートしていませんが、 OpenSearch サービスには Dashboards へのアクセスを制御するためのソリューションがいくつか用意されています。

- Dashboards の SAML 認証を有効にします。
- HTTP 基本認証を用いたきめ細かなアクセスコントロールを使用します。
- ダッシュボードの Cognito 認証を設定します。
- パブリックアクセスドメインの場合、プロキシサーバーを使用する (または使用しない) IP ベースのアクセスポリシーを設定します。
- VPC アクセスドメインの場合、プロキシサーバーを使用する (または使用しない) オープンアクセスポリシー、およびセキュリティグループを使用してアクセス許可を制御します。詳細については、「the section called “VPC ドメインのアクセスポリシーについて”」を参照してください。

プロキシを使用して OpenSearch Dashboards から OpenSearch サービスにアクセスする

> **Note**
>
> このプロセスは、ドメインでパブリックアクセスが使用されており、Cognito 認証を使用しない場合にのみ適用できます。the section called “ OpenSearch Dashboards へのアクセスの制御” を参照してください。

Dashboards は JavaScript アプリケーションであるため、リクエストはユーザーの IP アドレスから送信されます。IP ベースのアクセスコントロールは、膨大な数の IP アドレスをホワイトリストに登録する必要があるため、各ユーザーに Dashboards へのアクセスを許可する方法として実用的とは言えません。回避策の 1 つは、Dashboards OpenSearch と OpenSearch Service の間にプロキシサーバーを配置することです。これにより、IP ベースのアクセスポリシーを追加し、唯一の IP アドレス (プロキシの IP アドレス) からのリクエストを許可できます。この設定は以下の図のようになります。



1. これは OpenSearch サービスドメインです。IAM は、このドメインへの承認済みアクセスを提供します。追加の IP ベースのアクセスポリシーは、プロキシサーバーへのアクセスを提供します。
2. これは、Amazon EC2 インスタンスで実行されているプロキシサーバーです。
3. 他のアプリケーションは、署名バージョン 4 の署名プロセスを使用して、認証されたリクエストを OpenSearch サービスに送信できます。
4. OpenSearch Dashboards クライアントは、プロキシを介して OpenSearch サービスドメインに接続します。

この種の設定を有効にするには、ロールと IP アドレスを指定するリソースベースのポリシーが必要です。ポリシー例を次に示します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*",
         "Principal":{
            "AWS":"arn:aws:iam::111111111111:role/allowedrole1"
         },
         "Action":[
            "es:ESHttpGet"
         ],
         "Effect":"Allow"
      },
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"*"
         },
         "Action":"es:*",
         "Condition":{
            "IpAddress":{
               "aws:SourceIp":[
                  "203.0.113.0/24",
                  "2001:DB8:1234:5678::/64"
               ]
            }
         },
         "Resource":"arn:aws:es:us-west-2:111111111111:domain/my-domain/*"
      }
   ]
}
```

プロキシサーバーを実行する EC2 インスタンスを Elastic IP アドレスを使用して設定することをお勧めします。これにより、必要に応じてインスタンスを置き換え、各インスタンスに同じパブリック IP アドレスをアタッチできます。詳細については、Amazon EC2 ユーザーガイド」の「Elastic IP アドレス」を参照してください。

プロキシサーバーおよび Cognito 認証を使用している場合、redirect_mismatch エラーを回避するため、Dashboards と Amazon Cognito に追加の設定が必要になる場合があります。次の nginx.conf の例を参照してください。

```
server {
    listen 443;
    server_name $host;
    rewrite ^/$ https://$host/_plugin/_dashboards redirect;

    ssl_certificate           /etc/nginx/cert.crt;
    ssl_certificate_key       /etc/nginx/cert.key;

    ssl on;
    ssl_session_cache  builtin:1000  shared:SSL:10m;
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    location /_plugin/_dashboards {
        # Forward requests to Dashboards
        proxy_pass https://$dashboards_host/_plugin/_dashboards;

        # Handle redirects to Cognito
        proxy_redirect https://$cognito_host https://$host;

        # Update cookie domain and path
        proxy_cookie_domain $dashboards_host $host;
        proxy_cookie_path / /_plugin/_dashboards/;

        # Response buffer settings
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
    }

    location ~ \/(log|sign|fav|forgot|change|saml|oauth2) {
        # Forward requests to Cognito
        proxy_pass https://$cognito_host;

        # Handle redirects to Dashboards
        proxy_redirect https://$dashboards_host https://$host;

        # Update cookie domain
```

```
        proxy_cookie_domain $cognito_host $host;
    }
}
```

WMS マップサーバーを使用するように OpenSearch Dashboards を設定する

OpenSearch Dashboards for OpenSearch Service のデフォルトインストールには、インドおよび中国リージョンのドメインを除くマップサービスが含まれています。マップサービスは、最大 10 のズームレベルをサポートします。

リージョンに関係なく、座標マップの可視化に別の Web Map Service (WMS) サーバーが使用されるように Dashboards を設定できます。リージョンマップの可視化では、デフォルトのマップサービスのみがサポートされます。

WMS マップサーバーを使用できるように Dashboards を設定するには:

1. Dashboards を開きます。
2. [スタックの管理] を選択します。
3. [詳細設定] を選択します。
4. visualization:tileMap:WMSdefaults を見つけます。
5. `enabled` を `true` に変更し、`url` を有効な WMS マップサーバーの URL に変更します。

```
{
  "enabled": true,
  "url": "wms-server-url",
  "options": {
    "format": "image/png",
    "transparent": true
  }
}
```

6. [変更を保存] を選択します。

新しいデフォルト値を可視化に適用するには、Dashboards の再ロードが必要になることがあります。可視化結果を保存した場合は、可視化結果を開いた後に [オプション] を選択します。[WMS マップサーバー] が有効で、[WMS URL)] に優先マップサーバーが指定されていることを確認し、[変更の適用] を選択します。

> **Note**
>
> マップサービスは多くの場合、ライセンス料や制限事項を伴います。マップサーバーを指定する際には、このような点を考慮する必要があります。テストを行うには、アメリカ地質調査所のマップサービスが便利です。

ローカル Dashboards サーバーを OpenSearch サービスに接続する

既に独自の OpenSearch Dashboards インスタンスの設定に多大な時間を費やしている場合は、OpenSearch サービスが提供するデフォルトの Dashboards インスタンスの代わりに (または追加で) 使用できます。以下の手順は、オープンアクセスポリシーできめ細かなアクセス制御を使用するドメイン用です。

ローカル OpenSearch Dashboards サーバーを OpenSearch サービスに接続するには

1. OpenSearch サービスドメインで、適切なアクセス許可を持つユーザーを作成します。
   a. Dashboards で、[セキュリティ]、[内部ユーザー] に進み、[内部ユーザーの作成] を選択します。
   b. ユーザーネームとパスワードを入力し、[作成] を選択します。
   c. [ロール] に進み、ロールを選択します。
   d. [マッピングされたユーザー] を選択し、[マッピングの管理] を選択します。
   e. [ユーザー] で、ユーザーネームを追加し、[マップ] を選択します。
2. セルフマネージド Dashboards OSS インストールに適切なバージョンの OpenSearch セキュリティプラグインをダウンロードしてインストールします。
3. ローカル Dashboards サーバーで `config/opensearch_dashboards.yml` ファイルを開き、前に作成したユーザー名とパスワードを使用して OpenSearch サービスエンドポイントを追加します。

   ```
   opensearch.hosts: ['https://domain-endpoint']
   opensearch.username: 'username'
   opensearch.password: 'password'
   ```

   以下のサンプル `opensearch_dashboards.yml` ファイルを使用できます。

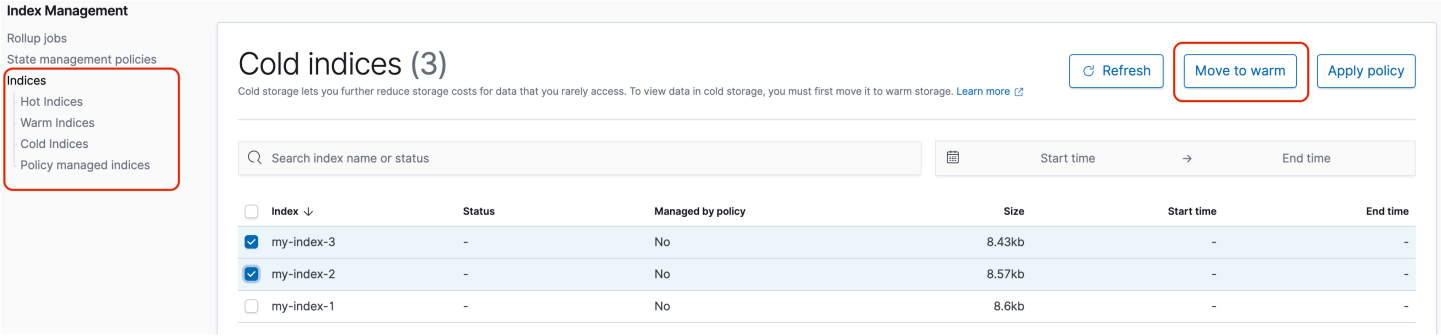
```
server.host: '0.0.0.0'

opensearch.hosts: ['https://domain-endpoint']

opensearchDashboards.index: ".username"

opensearch.ssl.verificationMode: none # if not using HTTPS

opensearch_security.auth.type: basicauth
opensearch_security.auth.anonymous_auth_enabled: false
opensearch_security.cookie.secure: false # set to true when using HTTPS
opensearch_security.cookie.ttl: 3600000
opensearch_security.session.ttl: 3600000
opensearch_security.session.keepalive: false
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ['opensearch_dashboards_read_only']
opensearch_security.auth.unauthenticated_routes: []
opensearch_security.basicauth.login.title: 'Please log in using your username and
 password'

opensearch.username: 'username'
opensearch.password: 'password'
opensearch.requestHeadersWhitelist: [authorization, securitytenant,
 security_tenant]
```

OpenSearch サービスインデックスを表示するには、ローカル Dashboards サーバーを起動し、開発ツールに移動して次のコマンドを実行します。

```
GET _cat/indices
```

OpenSearch Dashboards でのインデックスの管理

OpenSearch サービスドメインに OpenSearch Dashboards をインストールすると、ドメインの異なるストレージ階層でインデックスを管理するための便利な UI が提供されます。Dashboards のメインメニューからインデックス管理を選択すると、ホットストレージ、、UltraWarmコールドストレージのすべてのインデックスと、インデックスステート管理 (ISM) ポリシーによって管理されるインデックスが表示されます。インデックス管理を使用して、ウォームストレージとコールドストレージ間でインデックスを移動し、3 つの階層間の移行をモニタリングします。



UltraWarm および/またはコールドストレージが有効になっていない限り、ホットインデックス、ウォームインデックス、コールドインデックスのオプションは表示されないことに注意してください。

その他の機能

各 OpenSearch サービスドメインへのデフォルトの OpenSearch Dashboards インストールには、いくつかの追加機能があります。

- さまざまなOpenSearchプラグインのユーザーインターフェイス
- テナント
- レポート

  レポート作成メニューを使用して、Discover ページおよびダッシュボードまたは可視化の PDF レポートまたは PNG レポートからオンデマンド CSV レポートを生成します。CSV レポートには 10,000 行までの制限があります。
- ガントチャート
- ノートブック

Amazon OpenSearch Service でのインデックス管理

Amazon OpenSearch Service にデータを追加した後には、そのデータのインデックスを再作成したり、インデックスのエイリアスを設定したり、インデックスをよりコスト効率の高いストレージに移動したり、インデックスを完全に削除したりすることが必要になる場合がよくあります。この章では、UltraWarm ストレージとインデックスステート管理について説明します。OpenSearch インデックス API の詳細については、「OpenSearch ドキュメント」を参照してください。

トピック

- UltraWarm Amazon OpenSearch Service の ストレージ
- Amazon OpenSearch Service のコールドストレージ
- Amazon OpenSearch サービス用 OR1 ストレージ
- Amazon OpenSearch Service でのインデックス状態管理
- インデックスロールアップによる Amazon OpenSearch Service でのインデックスのサマリー
- Amazon OpenSearch Service でのインデックスの変換
- Amazon OpenSearch Service のクラスター間レプリケーション
- リモート再インデックスを使用した Amazon OpenSearch Service インデックスの移行
- データストリームを使用した Amazon OpenSearch Service での時系列データの管理

UltraWarm Amazon OpenSearch Service の ストレージ

UltraWarm は、大量の読み取り専用データを Amazon OpenSearch Service に保存するための費用対効果の高い方法を提供します。標準データノードは「ホット」ストレージを使用します。このストレージは、各ノードにアタッチされたインスタンスストアまたは Amazon EBS ボリュームの形をとります。ホットストレージは、新しいデータのインデックス作成と検索において、可能な限り高速なパフォーマンスを提供します。

UltraWarm ノードは、アタッチされたストレージではなく、Amazon S3 と高度なキャッシュソリューションを使用してパフォーマンスを向上させます。アクティブに書き込んでいないインデックス、クエリの頻度が低く、同じパフォーマンスを必要としないインデックスの場合、 UltraWarm はデータ 1 GiB あたりのコストを大幅に削減します。ウォームインデックスはホットストレージに戻さない限り読み取り専用であるため、 UltraWarm は ログなどのイミュータブルなデータに最適です。

では OpenSearch、ウォームインデックスは他のインデックスと同じように動作します。同じ APIs、それらを使用して OpenSearch Dashboards で視覚化を作成したりできます。

トピック

前提条件

UltraWarm にはいくつかの重要な前提条件があります。

- UltraWarm には、 OpenSearch または Elasticsearch 6.8 以降が必要です。
- ウォームストレージを使用するには、ドメインに専用のマスターノードが必要です。
- スタンバイドメインでマルチ AZ を使用する場合、ウォームノードの数は、使用するアベイラビリティーゾーンの数の倍数である必要があります。
- ドメインでデータノードに T2 または T3 インスタンスタイプが使用されている場合、ウォームストレージを使用することはできません。
- インデックスがおおよその k-NN (`"index.knn": true`) を使用している場合、ウォームストレージに移すことはできません。

- ドメインがきめ細かなアクセスコントロール を使用している場合、ユーザーは API コールを行うには OpenSearch Dashboards の ultrawarm_manager ロールにマッピングされている必要があります。 UltraWarm

> **Note**
>
> ultrawarm_manager ロールは、既存の一部の OpenSearch サービスドメインで定義されていない場合があります。Dashboards にロールが表示されない場合は、それを手動で作成する必要があります。

許可の設定

既存の OpenSearch サービスドメイン UltraWarm で を有効にした場合、ultrawarm_managerロールがドメインで定義されていない可能性があります。きめ細かなアクセスコントロールを使用してドメインのウォームインデックスを管理するには、管理者以外のユーザーがこのロールにマッピングされている必要があります。ultrawarm_manager ロールを手動で作成するには、以下のステップを実行します。

1. OpenSearch ダッシュボードで、セキュリティに移動し、アクセス許可 を選択します。
2. [アクショングループの作成] を選択し、以下のグループを設定します。

グループ名	許可
ultrawarm_cluster	• cluster:admin/ultrawarm/migration/list • cluster:monitor/nodes/stats
ultrawarm_index_read	• indices:admin/ultrawarm/migration/get • indices:admin/get
ultrawarm_index_write	• indices:admin/ultrawarm/migration/warm • indices:admin/ultrawarm/migration/hot • indices:monitor/stats • indices:admin/ultrawarm/migration/cancel

3. [ロール]、[ロールの作成] の順に選択します。

4. ロールに ultrawarm_manager という名前を付けます。
5. [クラスターの許可] では、`ultrawarm_cluster` および `cluster_monitor` を選択します。
6. [インデックス] では、`*` と入力します。
7. [インデックスの許可] では、`ultrawarm_index_read`、`ultrawarm_index_write`、および `indices_monitor` を選択します。
8. [作成] を選択します。
9. ロールを作成したら、 UltraWarm インデックスを管理する任意のユーザーまたはバックエンドロールにマッピングします。

UltraWarm ストレージ要件とパフォーマンスに関する考慮事項

で説明されているようにthe section called “ストレージ要件の計算”、ホットストレージ内のデータには、レプリカ、Linux リザーブドスペース、 OpenSearch サービスリザーブドスペースという大きなオーバーヘッドが発生します。例えば、1 つのレプリカシャードを持つ 20 GiB プライマリシャードには、約 58 GiB のホットストレージが必要です。

Amazon S3 を使用するため、 ではこのオーバーヘッド UltraWarm は発生しません。 UltraWarm ストレージ要件を計算するときは、プライマリシャードのサイズのみを考慮します。S3 のデータの耐久性はレプリカの必要性を排除し、S3 はオペレーティングシステムやサービスの考慮事項を抽象化します。同じ 20 GiB シャードには、20 GiB のウォームストレージが必要です。`ultrawarm1.large.search` インスタンスをプロビジョニングする場合、プライマリシャードには最大ストレージの 20 TiB すべてを使用できます。インスタンスタイプの概要と、それぞれが対応できるストレージの最大容量については、「the section called “UltraWarm ストレージクォータ”」を参照してください。

では UltraWarm、最大シャードサイズを 50 GiB にすることをお勧めします。各 UltraWarm インスタンスタイプに割り当てられる CPU コアの数と RAM の量によって、同時に検索できるシャードの数を把握できます。プライマリシャードのみが S3 のストレージに UltraWarmカウントされますが、OpenSearch Dashboards は`_cat/indices`インデックス UltraWarm サイズをすべてのプライマリシャードとレプリカシャードの合計としてレポートすることに注意してください。

例えば、各 `ultrawarm1.medium.search` インスタンスには 2 つの CPU コアがあり、S3 で最大 1.5 TiB のストレージに対応できます。これらのインスタンスのうちの 2 つには、3 TiB のストレージが組み合わされており、各シャードが 50 GiB の場合、約 62 のシャードになります。クラスターへのリクエストがこれらのシャードのうちの 4 つしか検索しない場合、パフォーマンスが優れている可能性があります。リクエストが広範で、それらの 62 すべてを検索する場合、4 つの

CPU コアがオペレーションを実施しにくくなる可能性があります。WarmCPUUtilization および WarmJVMMemoryPressureUltraWarm メトリクスをモニタリングして、インスタンスがワークロードをどのように処理するかを理解します。

検索が広範または頻繁である場合、インデックスをホットストレージに残すことを検討してください。他の OpenSearch ワークロードと同様に、ニーズ UltraWarm が満たされているかどうかを判断するための最も重要なステップは、現実的なデータセットを使用して代表的なクライアントテストを実行することです。

UltraWarm 料金

ホットストレージでは、プロビジョニングした分に対して料金を支払います。インスタンスにはアタッチされた Amazon EBS ボリュームが必要で、インスタンスストアが含まれるインスタンスもあります。そのストレージが空かいっぱいかに関係なく、同じ料金を支払います。

UltraWarm ストレージでは、使用した分に対して料金が発生します。ultrawarm1.large.search インスタンスは S3 で最大 20 TiB のストレージに対応できますが、1 TiB のデータを格納する場合、1 TiB のデータに対してのみ課金されます。他のすべてのノードタイプと同様に、 UltraWarm ノードごとに時間単位の料金も支払います。詳細については、「the section called “料金”」を参照してください。

の有効化 UltraWarm

コンソールは、ウォームストレージを使用するドメインを作成する最も簡単な方法です。ドメインの作成時に、 UltraWarm データノードを有効にすると、必要なウォームノードの数を選択します。前提条件を満たしていれば、既存のドメインでも同じ基本プロセスを使用できます。ドメインの状態が Processing から Active に変わった後でも、数時間は使用できない UltraWarm 場合があります。

スタンバイドメインでマルチ AZ を使用する場合、ウォームノードの数は、使用するアベイラビリティーゾーンの数の倍数である必要があります。詳細については、「the section called “Multi-AZ with Standby”」を参照してください。

AWS CLI または 設定 API を使用して UltraWarm、 、特に の WarmEnabled、WarmCount、および WarmTypeオプションを有効にすることもできますClusterConfig。

> **Note**
>
> ドメインはウォームノードの最大数をサポートします。詳細については、「the section called “クォータ”」を参照してください。

CLI コマンドの例

次の AWS CLI コマンドは、3 つのデータノード、3 つの専用マスターノード、6 つのウォームノード、およびきめ細かなアクセスコントロールが有効になっているドメインを作成します。

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-config
 InstanceCount=3,InstanceType=r6g.large.search,DedicatedMasterEnabled=true,DedicatedMasterType=
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["123456789012"]},"Action":
["es:*"],"Resource":"arn:aws:es:us-west-1:123456789012:domain/my-domain/*"}]}' \
  --region us-east-1
```

詳細については、「AWS CLI コマンドのリファレンス」を参照してください。

設定 API リクエストの例

設定 API に対する次のリクエストは、有効にされたきめ細かなアクセスコントロールおよび制限されたアクセスポリシーを持つ、3 つのデータノード、3 つの専用マスターノード、および 6 つのウォームノードを持つドメインを作成します。

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
```

```
    },
    "WarmEnabled": true,
    "WarmCount": 6,
    "WarmType": "ultrawarm1.medium.search"
  },
  "EBSOptions": {
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain",
  "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/my-domain/*\"}]}"
}
```

詳細については、「Amazon OpenSearch Service API リファレンス」を参照してください。

インデックスを UltraWarm ストレージに移行する

インデックスへの書き込みが完了し、可能な限り高速な検索パフォーマンスが不要になった場合は、ホットから に移行します UltraWarm。

```
POST _ultrawarm/migration/my-index/_warm
```

次に、移行のステータスを確認します。

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_SHARD_RELOCATION",
    "migration_type": "HOT_TO_WARM",
    "shard_level_status": {
      "running": 0,
      "total": 5,
      "pending": 3,
      "failed": 0,
      "succeeded": 2
    }
  }
}
```

移行を実施するには、インデックスヘルスが緑である必要があります。複数のインデックスを連続してすばやく移行する場合、_cat API と同様に、すべての移行の概要をプレーンテキストで取得できます。

```
GET _ultrawarm/migration/_status?v

index    migration_type state
my-index HOT_TO_WARM    RUNNING_SHARD_RELOCATION
```

OpenSearch サービスは、一度に 1 つのインデックスを に移行します UltraWarm。キューには最大 200 の移行を設定できます。制限を超えるリクエストは拒否されます。キュー内の移行の現在の個数を確認するには、HotToWarmMigrationQueueSize メトリクスをモニタリングします。インデックスは、移行プロセス全体を通して使用でき、ダウンタイムは発生しません。

移行プロセスには次の状態があります。

```
PENDING_INCREMENTAL_SNAPSHOT
RUNNING_INCREMENTAL_SNAPSHOT
FAILED_INCREMENTAL_SNAPSHOT
PENDING_FORCE_MERGE
RUNNING_FORCE_MERGE
FAILED_FORCE_MERGE
```

```
PENDING_FULL_SNAPSHOT
RUNNING_FULL_SNAPSHOT
FAILED_FULL_SNAPSHOT
PENDING_SHARD_RELOCATION
RUNNING_SHARD_RELOCATION
FINISHED_SHARD_RELOCATION
```

これらの状態が示すように、スナップショット、シャード再配置、強制マージ中に移行が失敗する可能性があります。スナップショットまたはシャード再配置中の障害は、通常、ノードの障害または S3 接続の問題が原因です。通常、ディスク領域の不足は、強制マージ失敗の根本的な原因です。

移行が完了すると、同じ _status リクエストでエラーが返されます。その時点でインデックスをチェックすると、ウォームインデックスに固有の設定が表示されます。

```
GET my-index/_settings

{
  "my-index": {
    "settings": {
      "index": {
        "refresh_interval": "-1",
        "auto_expand_replicas": "false",
        "provided_name": "my-index",
        "creation_date": "1599241458998",
        "unassigned": {
          "node_left": {
            "delayed_timeout": "5m"
          }
        },
        "number_of_replicas": "1",
        "uuid": "GswyCdR0RSq0SJYmzsIpiw",
        "version": {
          "created": "7070099"
        },
        "routing": {
          "allocation": {
            "require": {
              "box_type": "warm"
            }
          }
        },
        "number_of_shards": "5",
        "merge": {
```

```
          "policy": {
            "max_merge_at_once_explicit": "50"
          }
        }
      }
    }
  }
}
```

- number_of_replicas は、ディスク領域を消費しないパッシブレプリカの数です。
- routing.allocation.require.box_type は、インデックスが標準データノードではなくウォームノードを使用するように指定します。
- merge.policy.max_merge_at_once_explicit は、移行中に同時にマージするセグメントの数を指定します。

ウォームストレージのインデックスは、ホットストレージ に返されない限り読み取り専用です。ホットストレージ は、ログなどのイミュータブ UltraWarm ルなデータに最適です。インデックスのクエリを行ってそれらを削除することはできますが、個々のドキュメントを追加、更新、削除することはできません。それらを行おうとすると、次のエラーが発生する場合があります。

```
{
  "error" : {
    "root_cause" : [
      {
        "type" : "cluster_block_exception",
        "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage
 exceeded flood-stage watermark, index has read-only-allow-delete block];"
      }
    ],
    "type" : "cluster_block_exception",
    "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage exceeded
 flood-stage watermark, index has read-only-allow-delete block];"
  },
  "status" : 429
}
```

移行を自動化する

インデックスが特定の経過時間に達した後、または他の条件を満たした後の移行プロセスは、the section called “インデックスステート管理” を使用して自動化することをお勧めします。このワークフローを示すサンプルポリシーを参照してください。

移行の調整

インデックスを UltraWarm ストレージに移行するには、強制マージが必要です。各 OpenSearch インデックスはいくつかのシャードで構成され、各シャードはいくつかの Lucene セグメントで構成されます。強制マージオペレーションは、削除対象としてマークされたドキュメントをパージし、ディスク領域を節約します。デフォルトでは、 はインデックスを 1 つのセグメントに UltraWarm マージします。

`index.ultrawarm.migration.force_merge.max_num_segments` 設定を使用して、この値を最大 1,000 のセグメントに変更することができます。値を大きくすると、移行プロセスが高速になりますが、移行終了後のウォームインデックスのクエリレイテンシーが長くなります。設定を変更するには、次のリクエストを行います。

```
PUT my-index/_settings
{
  "index": {
    "ultrawarm": {
      "migration": {
        "force_merge": {
          "max_num_segments": 1
        }
      }
    }
  }
}
```

移行プロセスのこの段階でかかる時間を確認するには、`HotToWarmMigrationForceMergeLatency` メトリクスをモニタリングします。

移行をキャンセルする

UltraWarm は、キュー内で移行を順番に処理します。移行がキューに入っていても、まだ開始していない場合は、次のリクエストを使用して移行をキューから削除できます。

```
POST _ultrawarm/migration/_cancel/my-index
```

ドメインできめ細かなアクセスコントロールを使用する場合は、このリクエストを行うための indices:admin/ultrawarm/migration/cancel 許可を持っている必要があります。

ホットインデックスとウォームインデックスを一覧表示する

UltraWarm では、ホットインデックスとウォームインデックスの管理に役立つ のような _all2 つのオプションが追加されています。すべてのウォームインデックスまたはホットインデックスのリストについては、次のリクエストを実行します。

```
GET _warm
GET _hot
```

これらのオプションは、インデックスを指定する次のようなリクエストで使用できます。

```
_cat/indices/_warm
_cluster/state/_all/_hot
```

ウォームインデックスをホットストレージに戻す

インデックスに再度書き込む必要がある場合は、ホットストレージに移行し直します。

```
POST _ultrawarm/migration/my-index/_hot
```

ウォームストレージからホットストレージへのキュー移行は、一度に最大 10 個まで実行できます。OpenSearch サービスは、キューに入れられた順序で、移行リクエストを一度に 1 つずつ処理します。現在の個数を確認するには、WarmToHotMigrationQueueSize メトリクスをモニタリングします。

移行が完了したら、インデックス設定をチェックして、ニーズを満たしていることを確認します。インデックスはホットストレージに戻り、1 つのレプリカが作成されます。

スナップショットからのウォームインデックスの復元

自動スナップショットの標準リポジトリに加えて、 はウォームインデックスの 2 番目のリポジトリ UltraWarm を追加しますcs-ultrawarm。このリポジトリ内の各スナップショットには、1 つのイ

ンデックスしか含まれていません。ウォームインデックスを削除した場合、そのスナップショットは cs-ultrawarm リポジトリに 14 日間残ります。これは、他の自動スナップショットと同様です。

cs-ultrawarm からスナップショットを復元すると、ホットストレージではなくウォームストレージに復元されます。cs-automated-enc リポジトリと cs-automated リポジトリのスナップショットは、ホットストレージに復元されます。

UltraWarm スナップショットをウォームストレージに復元するには

1. 復元するインデックスを含む最新のスナップショットを特定します。

```
GET _snapshot/cs-ultrawarm/_all?verbose=false

{
  "snapshots": [{
    "snapshot": "snapshot-name",
    "version": "1.0",
    "indices": [
      "my-index"
    ]
  }]
}
```

> **Note**
>
> デフォルトでは、GET _snapshot/<repo> オペレーションは、リポジトリ内の各スナップショットの開始時間、終了時間、期間などの詳細なデータ情報を表示します。GET _snapshot/<repo> オペレーションは、リポジトリ内の各スナップショットのファイルから情報を取得します。開始時間、終了時間、期間は不要で、スナップショットの名前とインデックス情報のみが必要な場合、処理時間を最小限に抑えてタイムアウトを防ぐために、スナップショットを一覧表示する際に verbose=false パラメータを使用することをお勧めします。

2. インデックスがすでに存在する場合は、それを削除します。

```
DELETE my-index
```

インデックスを削除しない場合は、それをホットストレージに戻し、それの再インデックスを行います。

3. スナップショットの復元:

```
POST _snapshot/cs-ultrawarm/snapshot-name/_restore
```

UltraWarm は、この復元リクエストで指定したインデックス設定を無視しますが、`rename_pattern`や などのオプションを指定できます`rename_replacement`。 OpenSearch スナップショットの復元オプションの概要については、「」のOpenSearch ドキュメントを参照してください。

ウォームインデックスの手動スナップショット

ウォームインデックスの手動スナップショットを取得できますが、推奨されません。自動化された `cs-ultrawarm` リポジトリには、移行中に取得された各ウォームインデックスのスナップショットがすでに含まれており、追加料金は発生しません。

デフォルトでは、 OpenSearch サービスには手動スナップショットにウォームインデックスは含まれません。例えば、次の呼び出しには、ホットインデックスしか含まれていません。

```
PUT _snapshot/my-repository/my-snapshot
```

ウォームインデックスの手動スナップショットを取得することを選択した場合は、いくつかの重要な考慮事項が適用されます。

- ホットインデックスとウォームインデックスを混合することはできません。例えば、以下のコマンドは失敗します。

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1,hot-index-1",
  "include_global_state": false
}
```

ホットインデックスとウォームインデックスの混合が含まれている場合は、ワイルドカード (*) ステートメントも失敗します。

- スナップショットあたり 1 つのウォームインデックスしか含めることができません。例えば、以下のコマンドは失敗します。

```
PUT _snapshot/my-repository/my-snapshot
```

```
{
  "indices": "warm-index-1,warm-index-2,other-warm-indices-*",
  "include_global_state": false
}
```

このリクエストは成功します。

```
PUT _snapshot/my-repository/my-snapshot
{
  "indices": "warm-index-1",
  "include_global_state": false
}
```

- 手動スナップショットは、もともとウォームインデックスが含まれていても、常にホットストレージに復元されます。

ウォームインデックスをコールドストレージへ移行する

UltraWarm クエリの頻度が低い にデータがある場合は、コールドストレージへの移行を検討してください。コールドストレージは、ときどきしかアクセスしないデータや、使用されなくなったデータを対象としています。コールドインデックスを読み書きすることはできませんが、クエリを行う必要があるときはいつでも無償でウォームストレージに移行できます。手順については、「the section called “コールドストレージへのインデックスの移行”」を参照してください。

無効化 UltraWarm

コンソールは、 を無効にする最も簡単な方法です UltraWarm。ドメインを選択し、[アクション] から [クラスター設定の編集] を選択します。データ UltraWarm ノードを有効にする を選択し、変更の保存 を選択します。 AWS CLI および設定 API で `WarmEnabled` オプションを使用することもできます。

を無効にする前に UltraWarm、すべてのウォームインデックスを削除するか、ホットストレージに戻す必要があります。ウォームストレージが空になったら、 を無効にする前に 5 分待ちます UltraWarm。

Amazon OpenSearch Service のコールドストレージ

コールドストレージを使用すると、アクセス頻度の低いデータや履歴データを Amazon OpenSearch Service ドメインに保存し、他のストレージ階層よりも低コストでオンデマンドで分析できます。古

いデータに対して定期的な調査や法医学分析を行う必要がある場合は、コールドストレージが適しています。コールドストレージに適したデータの実際的な例としては、アクセス頻度の低いログ、コンプライアンス要件を満たすために保存する必要があるデータ、履歴価値のあるログなどがあります。

UltraWarm ストレージと同様に、コールドストレージは Amazon S3 によってバックアップされます。コールドデータをクエリする必要がある場合は、既存の UltraWarmノードに選択的にアタッチできます。コールドデータの移行とライフサイクルは、手動で、またはインデックスステート管理ポリシーを使用して管理できます。

トピック

前提条件

コールドストレージには、次の前提条件があります。

- コールドストレージには、 OpenSearch または Elasticsearch バージョン 7.9 以降が必要です。
- OpenSearch サービスドメインでコールドストレージを有効にするには、同じドメイン UltraWarm でも を有効にする必要があります。
- コールドストレージを使用するには、ドメインに専用のマスターノードがある必要があります。

- ドメインでデータノードに T2 または T3 インスタンスタイプが使用されている場合、コールドストレージを使用することはできません。
- インデックスがおおよその k-NN ("index.knn": true) を使用している場合、コールドストレージに移すことはできません。
- ドメインがきめ細かなアクセスコントロール を使用している場合、管理者以外のユーザーは、コールドインデックスを管理するために OpenSearch Dashboards の cold_manager ロールにマッピングされる必要があります。

> **Note**
>
> cold_manager ロールは、既存の一部の OpenSearch サービスドメインに存在しない可能性があります。Dashboards にロールが表示されない場合は、それを手動で作成する必要があります。

許可の設定

既存の OpenSearch サービスドメインでコールドストレージを有効にすると、cold_managerロールがドメインで定義されない場合があります。ドメインがきめ細かなアクセスコントロールを使用する場合、管理者以外のユーザーは、コールドインデックスを管理するためにこのロールにマッピングされている必要があります。cold_manager ロールを手動で作成するには、以下のステップを実行します。

1. OpenSearch ダッシュボードで、セキュリティに移動し、アクセス許可 を選択します。
2. [アクショングループの作成] を選択し、以下のグループを設定します。

グループ名	許可
cold_cluster	• cluster:monitor/nodes/stats • cluster:admin/ultrawarm* • cluster:admin/cold/*
cold_index	• indices:monitor/stats • indices:data/read/minmax • indices:admin/ultrawarm/migration/get

グループ名	許可
	• indices:admin/ultrawarm/migration/cancel

3. [ロール]、[ロールの作成] の順に選択します。
4. ロールに cold_manager という名前を付けます。
5. [クラスターの許可] の場合、作成した cold_cluster グループを選択します。
6. [インデックス] の場合、* と入力します。
7. インデックスの許可] の場合、作成した cold_index グループを選択します。
8. [作成] を選択します。
9. ロールを作成したら、コールドインデックスを管理する任意のユーザーまたはバックエンドロールにそれをマッピングします。

コールドストレージの要件とパフォーマンスに関する考慮事項

コールドストレージは Amazon S3 を使用するため、レプリカ、Linux リザーブドスペース、OpenSearch サービスリザーブドスペースなどのホットストレージのオーバーヘッドは発生しません。コールドストレージには、コンピューティング性能がアタッチされていないため、特定のインスタンスタイプはありません。コールドストレージには、任意の量のデータを保存できます。Amazon の ColdStorageSpaceUtilizationメトリクスをモニタリング CloudWatch して、使用しているコールドストレージ領域を確認します。

コールドストレージの料金

UltraWarm ストレージと同様に、コールドストレージではデータストレージに対してのみ料金が発生します。コールドデータにはコンピューティングコストはないので、コールドストレージにデータがない場合、課金されることはありません。

コールドストレージとウォームストレージ間でデータを移動する場合、転送料金は発生しません。インデックスがウォームストレージとコールドストレージ間で移行されている間、インデックスの 1 つのコピーに対してのみ、継続して料金が発生します。移行が完了すると、インデックスは、移行先のストレージ階層に従って課金されます。コールドストレージの料金の詳細については、「Amazon OpenSearch Service の料金」を参照してください。

コールドストレージを有効にする

コンソールは、コールドストレージを使用するドメインを作成する最も簡単な方法です。ドメインを作成している間に、[コールドストレージを有効にする] を選択します。前提条件を満たしている限り、既存のドメインでも同じプロセスを使用できます。ドメインの状態が [処理中] から [アクティブ] になっても、コールドストレージが使用可能になるには数時間かかることがあります。

AWS CLI または設定 API を使用して、コールドストレージを有効にすることもできます。

CLI コマンドの例

次の AWS CLI コマンドは、3 つのデータノード、3 つの専用マスターノード、コールドストレージが有効で、きめ細かなアクセスコントロールが有効になっているドメインを作成します。

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-
config ColdStorageOptions={Enabled=true},WarmEnabled=true,WarmCount=4,WarmType=ultrawarm1.mediu
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --region us-east-2
```

詳細については、「AWS CLI コマンドのリファレンス」を参照してください。

設定 API リクエストの例

設定 API に対する次のリクエストにより、3 つのデータノード、3 つの専用マスターノード、有効になっているコールドストレージ、および有効になっているきめ細かなアクセスコントロールがあるドメインが作成されます。

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
```

```
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
     },
    "WarmEnabled": true,
    "WarmCount": 4,
    "WarmType": "ultrawarm1.medium.search",
    "ColdStorageOptions": {
       "Enabled": true
     }
  },
  "EBSOptions": {
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain"
}
```

詳細については、「Amazon OpenSearch Service API リファレンス」を参照してください。

OpenSearch Dashboards でのコールドインデックスの管理

ホットインデックス、ウォームインデックス、コールドインデックスは、 OpenSearch サービスドメイン内の既存の Dashboards インターフェイスで管理できます。Dashboards を使用すると、CLI や設定 API を使用することなく、ウォームストレージとコールドストレージ間でインデックスを移行し、インデックスの移行ステータスをモニタリングできます。詳細については、 OpenSearch「Dashboards でのインデックスの管理」を参照してください。

コールドストレージへのインデックスの移行

インデックスをコールドストレージに移行する場合、データの時間範囲を指定して簡単に検出できるようにします。インデックス内のデータに基づいてタイムスタンプフィールドを選択するか、開始タイムスタンプと終了タイムスタンプを手動で指定するか、それを指定しないように選択することができます。

パラメータ	サポートされている値	説明
`timestamp_field`	インデックスマッピングの日付/時刻フィールド。	指定されたフィールドの最小値と最大値が計算され、コールドインデックスの `start_time` および `end_time` メタデータとして保存されます。
`start_time` ：および `end_time`	以下の形式のいずれか。 • strict_date_optional_time。例えば、`yyyy-MM-dd'T'HH:mm:ss.SSSZ` 、`yyyy-MM-dd` などです。 • エポック時間 (ミリ秒)	指定された値は、コールドインデックスの `start_time` および `end_time` メタデータとして保存されます。

タイムスタンプを指定しない場合は、代わりに `?ignore=timestamp` をリクエストに追加します。

次のリクエストは、ウォームインデックスをコールドストレージに移行し、そのインデックスのデータの開始時刻と終了時刻を提供します。

```
POST _ultrawarm/migration/my-index/_cold
  {
    "start_time": "2020-03-09",
    "end_time": "2020-03-09T23:00:00Z"
  }
```

次に、移行のステータスを確認します。

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_METADATA_RELOCATION",
    "migration_type": "WARM_TO_COLD"
  }
}
```

OpenSearch サービスは、一度に 1 つのインデックスをコールドストレージに移行します。キューには最大 100 の移行を設定できます。制限を超えるリクエストは拒否されます。キュー内の移行の現在の個数を確認するには、WarmToColdMigrationQueueSize メトリクスをモニタリングします。移行プロセスには次の状態があります。

```
ACCEPTED_COLD_MIGRATION - Migration request is accepted and queued.
RUNNING_METADATA_MIGRATION - The migration request was selected for execution and
 metadata is migrating to cold storage.
FAILED_METADATA_MIGRATION - The attempt to add index metadata has failed and all
 retries are exhausted.
PENDING_INDEX_DETACH - Index metadata migration to cold storage is completed. Preparing
 to detach the warm index state from the local cluster.
RUNNING_INDEX_DETACH - Local warm index state from the cluster is being removed. Upon
 success, the migration request will be completed.
FAILED_INDEX_DETACH - The index detach process failed and all retries are exhausted.
```

コールドストレージへの移行の自動化

インデックスが特定の経過時間に達した後、または他の条件を満たした後の移行プロセスは、インデックスステート管理を使用して自動化することができます。ホットストレージからコールドストレージにインデックスを自動的に移行する方法を示すサンプルポリシー UltraWarm を参照してください。

> **Note**
>
> インデックスステート管理ポリシーを使用してインデックスをコールドストレージに移動するには、明示的な `timestamp_field` が必要です。

コールドストレージへの移行のキャンセル

コールドストレージへの移行がキューに追加されているか、失敗状態にある場合は、次のリクエストを使用して移行をキャンセルできます。

```
POST _ultrawarm/migration/_cancel/my-index

{
  "acknowledged" : true
}
```

ドメインがきめ細かなアクセスコントロールを使用している場合、このリクエストを行うには `indices:admin/ultrawarm/migration/cancel` 許可が必要です。

コールドインデックスの一覧表示

クエリを実行する前に、コールドストレージのインデックスを一覧表示して、詳細な分析 UltraWarm のためにどのインデックスに移行するかを決定できます。次のリクエストは、すべてのコールドインデックスをインデックス名でソートして一覧表示します。

```
GET _cold/indices/_search
```

レスポンス例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 3,
  "indices" : [
    {
      "index" : "my-index-1",
      "index_cold_uuid" : "hjEoh26mRRCFxRIMdgvLmg",
      "size" : 10339,
      "creation_date" : "2021-06-28T20:23:31.206Z",
```

```
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-2",
      "index_cold_uuid" : "0vIS2n-oROmOWDFmwFIgdw",
      "size" : 6068,
      "creation_date" : "2021-07-15T19:41:18.046Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-3",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

フィルタリング

プレフィックスベースのインデックスパターンと時間範囲のオフセットに基づいて、コールドインデックスをフィルタリングできます。

次のリクエストは、event-* のプレフィックスパターンに一致するインデックスを一覧表示します。

```
GET _cold/indices/_search
 {
   "filters":{
      "index_pattern": "event-*"
   }
 }
```

レスポンス例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
```

```
  "indices" : [
    {
      "index" : "events-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

次のリクエストは、2019-03-01 および 2020-03-01 の間の start_time と end_time のメタデータフィールドを持つインデックスを返します。

```
GET _cold/indices/_search
{
  "filters": {
    "time_range": {
      "start_time": "2019-03-01",
      "end_time": "2020-03-01"
    }
  }
}
```

レスポンス例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "my-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2019-05-09T00:00Z",
      "end_time" : "2019-09-09T23:00Z"
    }
  ]
}
```

ソート

コールドインデックスは、インデックス名やサイズなどのメタデータフィールドによって並べ替えることができます。次のリクエストは、サイズを降順で並べ替えたすべてのインデックスを一覧表示します。

```
GET _cold/indices/_search
 {
 "sort_key": "size:desc"
 }
```

レスポンス例

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 5,
  "indices" : [
    {
      "index" : "my-index-6",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-9",
      "index_cold_uuid" : "mbD3ZRVDRI6ONqgEOsJyUA",
      "size" : 57922,
      "creation_date" : "2021-07-07T23:41:35.640Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-5",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
```

```
}
```

その他の有効なソートキーは、start_time:asc/desc、end_time:asc/desc、および index_name:asc/desc です。

ページ分割

コールドインデックスのリストをページ分割できます。page_size パラメータを使用して、ページごとに返されるインデックスの数を設定します (デフォルトは 10)。コールドインデックスのすべての _search リクエストは、後続の呼び出しに使用できる pagination_id を返します。

次のリクエストは、コールドインデックスの _search リクエストの結果をページ分割で表示し、次の 100 件の結果を表示します。

```
GET _cold/indices/_search?page_size=100
{
"pagination_id": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
}
```

ウォームストレージへのインデックスの移行

前のセクションのフィルタリング条件を使用してコールドインデックスのリストを絞り込んだら、データをクエリして視覚化を作成できる UltraWarm 場所に戻します。

次のリクエストは、2 つのコールドインデックスをウォームストレージに移行します。

```
POST _cold/migration/_warm
 {
 "indices": "my-index1,my-index2"
 }


{
  "acknowledged" : true
}
```

移行のステータスを確認して移行 ID を取得するには、次のリクエストを送信します。

```
GET _cold/migration/_status
```

レスポンス例

```
{
  "cold_to_warm_migration_status" : [
    {
      "migration_id" : "tyLjXCA-S76zPQbPVHkOKA",
      "indices" : [
        "my-index1,my-index2"
      ],
      "state" : "RUNNING_INDEX_CREATION"
    }
  ]
}
```

インデックス固有の移行情報を取得するには、インデックス名を含めます。

```
GET _cold/migration/my-index/_status
```

インデックスを指定するのではなく、現在の移行ステータス別にインデックスを一覧表示できます。有効な値は、_failed、_accepted、_all です。

次のコマンドは、単一の移行リクエスト内のすべてのインデックスのステータスを取得します。

```
GET _cold/migration/_status?migration_id=my-migration-id
```

ステータスリクエストを使用して移行 ID を取得します。移行の詳細については、&verbose=true を追加します。

コールドストレージからウォームストレージには 10 またはそれ未満のバッチ数でインデックスを移行できます。最大で 100 インデックスを同時に移行できます。制限を超えるリクエストは拒否されます。現在、移行中の個数を確認するには、ColdToWarmMigrationQueueSize メトリクスをモニタリングします。移行プロセスには次の状態があります。

```
ACCEPTED_MIGRATION_REQUEST - Migration request is accepted and queued.
RUNNING_INDEX_CREATION - Migration request is picked up for processing and will create
 warm indexes in the cluster.
PENDING_COLD_METADATA_CLEANUP - Warm index is created and the migration service will
 attempt to clean up cold metadata.
RUNNING_COLD_METADATA_CLEANUP - Cleaning up cold metadata from the indexes migrated to
 warm storage.
```

```
FAILED_COLD_METADATA_CLEANUP - Failed to clean up metadata in the cold tier.
FAILED_INDEX_CREATION - Failed to create an index in the warm tier.
```

スナップショットからのコールドインデックスの復元

削除されたコールドインデックスを復元する必要がある場合は、「the section called “スナップショットからのウォームインデックスの復元”」の手順に従ってから、そのインデックスを再度コールド階層に移行することで、ウォーム層に復元できます。削除されたコールドインデックスをコールド階層に直接復元することはできません。 OpenSearch コールドインデックスは、削除されてから 14 日間保持されます。

コールドストレージからウォームストレージへの移行のキャンセル

コールドストレージからウォームストレージへのインデックスの移行がキューに追加されているか、失敗状態にある場合は、次のリクエストを用いてそれをキャンセルできます。

```
POST _cold/migration/my-index/_cancel

{
  "acknowledged" : true
}
```

インデックスのバッチ (一度に最大 10 個) の移行をキャンセルするには、移行 ID を指定します。

```
POST _cold/migration/_cancel?migration_id=my-migration-id

{
  "acknowledged" : true
}
```

ステータスリクエストを使用して移行 ID を取得します。

コールドインデックスメタデータの更新

コールドインデックスの start_time および end_time フィールドを更新できます。

```
PATCH _cold/my-index
 {
 "start_time": "2020-01-01",
 "end_time": "2020-02-01"
```

```
}
```

コールドストレージのインデックスの timestamp_field を更新することはできません。

> **Note**
>
> OpenSearch Dashboards は PATCH メソッドをサポートしていません。curl、Postman、または他のメソッドを使用して、コールドメタデータを更新します。

コールドインデックスの削除

ISM ポリシーを使用していない場合は、コールドインデックスを手動で削除できます。次のリクエストは、コールドインデックスを削除します。

```
DELETE _cold/my-index

{
  "acknowledged" : true
}
```

コールドストレージを無効にする

コールドストレージを無効にする最も簡単な方法は、 OpenSearch サービスコンソールです。ドメインを選択し、[アクション] から [クラスター設定の編集] を選択し、[コールドストレージを有効にする] の選択を解除します。

AWS CLI または設定 API を使用するには、 で ColdStorageOptionsを設定します"Enabled"="false"。

コールドストレージを無効にする前に、すべてのコールドインデックスを削除するか、ウォームストレージに移行する必要があります。そうしないと、無効アクションは失敗します。

Amazon OpenSearch サービス用 OR1 ストレージ

OR1 は Amazon OpenSearch Service のインスタンスファミリーで、大量のデータを費用対効果の高い方法で保存できます。OR1 インスタンスのあるドメインは、Amazon Elastic Block Store (Amazon EBS) gp3 io1 またはボリュームをプライマリストレージとして使用し、データは到着時に Amazon S3 に同期的にコピーされます。このストレージ構造により、高い耐久性を備えたイン

デックス作成スループットが改善します。また、OR1 インスタンスファミリーは、障害発生時の自動データ回復をサポートします。OR1 インスタンスタイプのオプションについては、「the section called “現行世代のインスタンスタイプ”」を参照してください。

ログ分析、オブザーバビリティ、セキュリティ分析など、負荷の高い運用分析ワークロードのインデックスを作成する場合、OR1 インスタンスのパフォーマンスと計算効率の向上によるメリットがあります。さらに、OR1 インスタンスが提供する自動データ復旧により、ドメインの全体的な信頼性が向上します。

OpenSearch サービスはストレージ関連のOR1メトリックをAmazonに送信します。 CloudWatch使用可能なメトリクスのリストについては、??? を参照してください。

OR1 インスタンスは、オンデマンドまたはリザーブドインスタンス料金でご利用いただけます。Amazon EBS と Amazon S3 でプロビジョニングされたインスタンスとストレージには、時間単位の料金が適用されます。

トピック

- 制限事項
- OR1 とストレージの違い UltraWarm
- OR1 インスタンスの使用

制限事項

ドメインに OR1 インスタンスを使用する際には、以下の制限を考慮してください。

- OpenSearch ドメインはバージョン 2.11 以降を実行している必要があります。
- ドメインでは保管時の暗号化が有効になっている必要があります。詳細については、「???」を参照してください。
- ドメインは新しいドメインでなければなりません。既存のドメインを OR1 インスタンスを使用するように変更することはできません。
- ドメインで専用マスターノードを使用している場合は、Graviton インスタンスを使用する必要があります。専用マスターノードの詳細については、「」を参照してください。???
- OR1 インスタンスのシャードサイズは 100 GiB 未満でなければなりません。100 GiB を超えるシャードは、回復時間を遅くする可能性があります。OR1 インスタンスで 100 GiB を超えるシャードを作成すると、 OpenSearch Service はドメインへの書き込みリクエストをブロックします。それでも 100 GiB を超えるシャードを使用したい場合は、AWS Support問い合わせてクォータの増額をリクエストしてください。

- OR1 インスタンスのインデックスの更新間隔は 10 秒以上でなければなりません。OR1 インスタンスのデフォルト更新間隔は 10 秒です。

OR1 とストレージの違い UltraWarm

OpenSearch Service は、 UltraWarm ウォームデータの保存コストを削減するように最適化されたインスタンスを提供します。OR1 UltraWarm とインスタンスはどちらも Amazon EBS にローカルでデータを保存し、Amazon S3 にリモートでデータを保存します。ただし、OR1 UltraWarm とインスタンスはいくつかの重要な点で異なります。

- OR1 インスタンスは、ローカルストレージとリモートストレージの両方にデータのコピーを保持します。 UltraWarm インスタンスは、ストレージコストを削減するために、データを主にリモートストレージに保持します。使用パターンによっては、データをローカルストレージに移動する場合があります。
- OR1 インスタンスはアクティブで、読み取り/書き込み操作を受け付けますが、 UltraWarm インスタンスのデータは、手動でホットストレージに戻すまでは読み取り専用です。
- UltraWarm データの耐久性はインデックスのスナップショットに依存しています。それに比べて、OR1 インスタンスはバックグラウンドでレプリケーションとリカバリを実行します。インデックスが赤色の場合、OR1 インスタンスは失われたシャードを Amazon S3 のリモートストレージから自動的に復元します。復旧時間は、復旧するデータ量によって異なります。

UltraWarm ストレージの詳細については、を参照してください???。

OR1 インスタンスの使用

、 AWS Command Line Interface (AWS CLI)、または AWS SDK を使用して新しいドメインを作成するときに AWS Management Console、データノードに OR1 インスタンスを選択できます。その後、既存のツールを使用してデータのインデックス作成とクエリを行うことができます。

コンソール

1. にある Amazon OpenSearch サービスコンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。
3. [ドメインの作成] をクリックします。
4. ドメインの名前と、その他の希望オプションを入力します。[インスタンスファミリー] で、[OR1] を選択します。[作成] を選択して、ドメイン作成プロセスを開始します。

AWS CLI

1. AWS CLI ターミナルに移動します。をインストールする必要がある場合は AWS CLI、「の最新バージョンのインストールまたは更新」を参照してください AWS CLI。
2. OR1 ストレージを使用するには、ドメインの作成時に特定の OR1 InstanceType インスタンスタイプサイズの値をフィールドに入力する必要があります。保存時の暗号化も有効にする必要があります。

   次の例では、サイズが 2xlarge である OR1 インスタンスを使用するドメインを作成します。

```
aws opensearch create-domain \
  --domain-name test-domain \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=or1.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMast
 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=200" \
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-east-1:account-
id:domain/test-domain/*"}]}'
```

Amazon OpenSearch Service でのインデックス状態管理

Amazon OpenSearch Service のインデックス状態管理 (ISM) では、ルーチンタスクを自動化するカスタム管理ポリシーを定義し、インデックスとインデックスパターンに適用できます。これにより、インデックスオペレーションを実行するために、外部プロセスを設定および管理する必要がなくなります。

ポリシーには、デフォルトの状態と、移行するインデックスの状態一覧が含まれています。各状態内で、実行するアクションのリストと、移行をトリガーする条件を定義できます。一般的なユースケースは、一定期間が経過した古いインデックスを定期的に削除することです。例えば、30 日後にインデックスを read_only 状態に移行して、90 日後に最終的に削除するポリシーを定義できます。

ポリシーをインデックスにアタッチすると、ISM は 5～8 分ごと (1.3 より前のクラスターに対しては 30～48 分ごと) に実行されるジョブを作成し、ポリシーアクションを実行して条件を確認してから、インデックスを別の状態に移行します。このジョブを実行する基本時間は 5 分ごとです。さらに、0～60% のランダムジッターが追加されることで、すべてのインデックスからのアクティビティが同時に急増することがなくなります。クラスターの状態が赤の場合、ISM はジョブを実行しません。

ISM には OpenSearch または Elasticsearch 6.8 以降が必要です。

> **Note**
>
> このドキュメントでは、ISM の簡単な概要といくつかのサンプルポリシーについて説明します。また、Amazon OpenSearch Service ドメインの ISM がセルフマネージド OpenSearch クラスターの ISM とどのように異なるかについても説明します。包括的なパラメータリファレンス、各設定の説明、API リファレンスを含む ISM の完全なドキュメントについては、OpenSearch ドキュメントの「インデックス状態管理」を参照してください。

> **Important**
>
> インデックステンプレートを使用して、新しく作成されたインデックスに ISM ポリシーを適用することはできなくなりました。ISM テンプレートフィールドを使用して、新しく作成されたインデックスを引き続き自動的に管理することができます。この更新では、この設定を使用して既存の CloudFormation テンプレートに影響を与える重大な変更が導入されました。

ISM ポリシーを作成する

インデックス状態管理を開始するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. ISM ポリシーを作成するドメインを選択します。
3. ドメインのダッシュボードから Dashboards URL OpenSearch に移動し、マスターユーザー名とパスワードでサインインします。URL はこの形式に従います。

```
domain-endpoint/_dashboards/
```

4. OpenSearch Dashboards 内の左側のナビゲーションパネルを開き、インデックス管理 を選択し、ポリシー を作成します。
5. ビジュアルエディタ または JSON エディタ を使用してポリシーを作成します。より構造化された方法でポリシーを定義できるビジュアルエディタを使用することをお勧めします。ポリシーの作成については、以下の サンプルポリシー を参照してください。
6. ポリシーを作成したら、それを 1 つ以上のインデックスにアタッチします。

```
POST _plugins/_ism/add/my-index
{
  "policy_id": "my-policy-id"
}
```

> **Note**
>
> ドメインで従来の Elasticsearch バージョンを実行している場合は、_plugins の代わりに _opendistro を使用します。

または、 OpenSearch Dashboards でインデックスを選択し、ポリシーの適用 を選択します。

サンプルポリシー

次のサンプルポリシーは、一般的な ISM ユースケースを自動化する方法を示しています。

ホットからウォーム、またコールドストレージまで

このサンプルポリシーは、インデックスをホットストレージから に移動しUltraWarm、最終的に に移動します。
コールドストレージ。次に、インデックスを削除します。

インデックスは、最初は hot 状態です。10 日後、ISM はインデックスを warm 状態に移動します。その 80 日後、90 日が経過したインデックスは cold 状態に移動します。1 年後、サービスは、インデックスを削除中であるという通知を Amazon Chime ルームに送信してから、完全にインデックスを削除します。

コールドインデックスでは、通常の delete オペレーションではなく、cold_delete オペレーションが必要であることに注意してください。また、ISM でコールドインデックスを管理するには、データに明示的な timestamp_field が必要であることに注意してください。

```
{
  "policy": {
    "description": "Demonstrate a hot-warm-cold-delete workflow.",
    "default_state": "hot",
    "schema_version": 1,
    "states": [{
        "name": "hot",
        "actions": [],
        "transitions": [{
          "state_name": "warm",
          "conditions": {
            "min_index_age": "10d"
          }
        }]
      },
      {
        "name": "warm",
        "actions": [{
          "warm_migration": {},
          "retry": {
            "count": 5,
            "delay": "1h"
          }
        }],
        "transitions": [{
          "state_name": "cold",
          "conditions": {
            "min_index_age": "90d"
          }
        }]
      },
      {
        "name": "cold",
        "actions": [{
            "cold_migration": {
              "timestamp_field": "<your timestamp field>"
            }
          }
        ],
```

```
        "transitions": [{
          "state_name": "delete",
          "conditions": {
             "min_index_age": "365d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "notification": {
            "destination": {
              "chime": {
                "url": "<URL>"
              }
            },
            "message_template": {
              "source": "The index {{ctx.index}} is being deleted."
            }
          }
        },
        {
          "cold_delete": {}
        }]
      }
    ]
  }
}
```

レプリカの数を減らす

このサンプルポリシーはもう少し単純で、ディスク容量を節約するために 7 日後にレプリカ数をゼロに減らし、21 日後にインデックスを削除します。このポリシーは、インデックスが重要ではなく、書き込みリクエストをこれ以上受信しないことを前提としています。レプリカ数をゼロにすると、データ損失のリスクがあります。

```
{
  "policy": {
    "description": "Changes replica count and deletes.",
    "schema_version": 1,
    "default_state": "current",
    "states": [{
        "name": "current",
```

```
        "actions": [],
        "transitions": [{
          "state_name": "old",
          "conditions": {
            "min_index_age": "7d"
          }
        }]
      },
      {
        "name": "old",
        "actions": [{
          "replica_count": {
            "number_of_replicas": 0
          }
        }],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
            "min_index_age": "21d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "delete": {}
        }],
        "transitions": []
      }
    ]
  }
}
```

インデックスのスナップショットを撮る

このサンプルポリシーでは、`snapshot` オペレーションを使用して、インデックスに少なくとも 1 つのドキュメントが含まれているとすぐに、インデックスのスナップショットを撮ります。`repository` は、Amazon S3 に登録した手動スナップショットリポジトリの名前です。`snapshot` は、スナップショットの名前です。スナップショットの前提条件およびリポジトリを登録する手順については、「the section called “インデックススナップショットの作成”」を参照してください。

```
{
  "policy": {
    "description": "Takes an index snapshot.",
    "schema_version": 1,
    "default_state": "empty",
    "states": [{
        "name": "empty",
        "actions": [],
        "transitions": [{
          "state_name": "occupied",
          "conditions": {
            "min_doc_count": 1
          }
        }]
      },
      {
        "name": "occupied",
        "actions": [{
          "snapshot": {
            "repository": "<my-repository>",
            "snapshot": "<my-snapshot>"
            }
          }],
          "transitions": []
      }
    ]
  }
}
```

ISM テンプレート

ism_template フィールドをポリシーにセットアップすると、テンプレートパターンと一致するインデックスを作成したときに、ポリシーがそのインデックスに自動的にアタッチされるようになります。この例では、「log」で始まる名前で作成したインデックスは、自動的に ISM ポリシーmy-policy-id と一致します。

```
PUT _plugins/_ism/policies/my-policy-id
{
  "policy": {
    "description": "Example policy.",
    "default_state": "...",
    "states": [...],
```

```
    "ism_template": {
      "index_patterns": ["log*"],
      "priority": 100
    }
  }
}
```

より詳細な例については、「自動ロールオーバー用 ISM テンプレートを使用したサンプルポリシー」を参照してください。

差異

OpenSearch と Elasticsearch と比較して、Amazon OpenSearch Service の ISM にはいくつかの違いがあります。

ISM オペレーション

- OpenSearch サービスは、、、および の 3 つの一意の ISM `warm_migration`オペレーションをサポートします`cold_migration``cold_delete`。
  - ドメインで UltraWarmが有効になっている場合、 `warm_migration`アクションはインデックスをウォームストレージに移行します。
  - ドメインでコールドストレージが有効になっている場合、`cold_migration` アクションはインデックスをコールドストレージに移行し、`cold_delete` アクションはインデックスをコールドストレージから削除します。

  これらのアクションのいずれかが設定されたタイムアウト期間内に完了しない場合でも、ウォームインデックスへの移行は継続されます。上記のアクションのいずれかに error_notification を設定すると、タイムアウト期間内にアクションが完了しなかった場合に、そのアクションが失敗したことが通知されますが、この通知は参考用です。実際のオペレーションに固有のタイムアウトはなく、最終的に成功または失敗するまで実行が継続されます。
- ドメインが OpenSearch または Elasticsearch 7.4 以降を実行している場合、 OpenSearch Service は ISM openおよび `close`オペレーションをサポートします。
- ドメインが OpenSearch または Elasticsearch 7.7 以降を実行している場合、 OpenSearch サービスは ISM `snapshot`オペレーションをサポートします。

コールドストレージ ISM オペレーション

コールドインデックスでは、次の ISM API を使用するときに ?type=_cold パラメータを指定する必要があります。

- ポリシーを追加する
- ポリシーを削除する
- ポリシーを更新する
- 失敗したインデックスを再試行する
- インデックスを説明する

これらのコールドインデックス用の API には、次のような違いがあります。

- ワイルドカード演算子は、最後に使用する場合を除き、サポートされていません。例えば、_plugins/_ism/<add, remove, change_policy, retry, explain>/logstash-* はサポートされていますが、_plugins/_ism/<add, remove, change_policy, retry, explain>/iad-*-prod はサポートされていません。
- 複数のインデックスの名前とパターンはサポートされていません。例えば、_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs はサポートされていますが、_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs,sample-data はサポートされていません。

ISM の設定

OpenSearch と Elasticsearch では、 _cluster/settings API を使用して使用可能なすべての ISM 設定を変更できます。Amazon OpenSearch Service では、次の ISM 設定のみ変更できます。

- クラスターレベルの設定:
  - plugins.index_state_management.enabled
  - plugins.index_state_management.history.enabled
- インデックスレベルの設定:
  - plugins.index_state_management.rollover_alias

チュートリアル: Index State Management プロセスの自動化

このチュートリアルでは、日常的なインデックス管理タスクを自動化して、それらをインデックスおよびインデックスパターンに適用する ISM ポリシーの実装方法を示します。

Amazon OpenSearch Service のインデックス状態管理 (ISM) を使用すると、定期的なインデックス管理アクティビティを自動化できるため、インデックスのライフサイクルを管理するための追加のツールを使用できなくなります。インデックスの経過時間、サイズ、その他の条件に基づいて、これらのオペレーションを自動化するポリシーを Amazon OpenSearch Service ドメイン内から作成できます。

OpenSearch サービスは 3 つのストレージ階層をサポートしています。アクティブ書き込みと低レイテンシーの分析にはデフォルトの「ホット」状態、最大 3 ペタバイトの読み取り専用データ UltraWarm には 、無制限の長期アーカイブにはコールドストレージです。

このチュートリアルでは、日次インデックスで時系列データを処理するサンプルユースケースをご紹介します。このチュートリアルでは、アタッチされた各インデックスの自動スナップショットを 24 時間後に作成するポリシーを設定します。次に、2 日後にインデックスをデフォルトのホット状態から UltraWarm ストレージに移行し、30 日後にコールドストレージに移行し、60 日後にインデックスを削除します。

前提条件

- OpenSearch サービスドメインは Elasticsearch バージョン 6.8 以降を実行している必要があります。
- ドメインでは、 UltraWarmとコールドストレージが有効になっている必要があります。
- ドメインの手動スナップショットリポジトリを登録する必要があります。
- ユーザーロールには、 OpenSearch サービスコンソールにアクセスするための十分なアクセス許可が必要です。必要に応じて、ドメインへのアクセスを検証して設定します。

ステップ 1: ISM ポリシーを設定する

まず、Dashboards で ISM OpenSearch ポリシーを設定します。

1. OpenSearch サービスコンソールのドメインダッシュボードから、 OpenSearch Dashboards URL に移動し、マスターユーザー名とパスワードでサインインします。URL はこの形式に従います: *domain-endpoint*/_dashboards/。

2. OpenSearch ダッシュボードで、サンプルデータを追加を選択し、1 つ以上のサンプルインデックスをドメインに追加します。
3. 左側のナビゲーションパネルを開き、[Index Management] (インデックス管理)、[Create policy] (ポリシーの作成) の順に選択します。
4. ポリシーには `ism-policy-example` という名前を付けます。
5. デフォルトのポリシーを次のポリシーに置き換えます。

```
{
  "policy": {
    "description": "Move indexes between storage tiers",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [],
        "transitions": [
          {
            "state_name": "snapshot",
            "conditions": {
              "min_index_age": "24h"
            }
          }
        ]
      },
      {
        "name": "snapshot",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "30m"
            },
            "snapshot": {
              "repository": "snapshot-repo",
              "snapshot": "ism-snapshot"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
```

```
            "conditions": {
              "min_index_age": "2d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "1h"
            },
            "warm_migration": {}
          }
        ],
        "transitions": [
          {
            "state_name": "cold",
            "conditions": {
              "min_index_age": "30d"
            }
          }
        ]
      },
      {
        "name": "cold",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "1h"
            },
            "cold_migration": {
              "start_time": null,
              "end_time": null,
              "timestamp_field": "@timestamp",
              "ignore": "none"
            }
          }
        ],
```

```
        "transitions": [
          {
            "state_name": "delete",
            "conditions": {
              "min_index_age": "60d"
            }
          }
        ]
      },
      {
        "name": "delete",
        "actions": [
          {
            "cold_delete": {}
          }
        ],
        "transitions": []
      }
    ],
    "ism_template": [
      {
        "index_patterns": [
          "index-*"
        ],
        "priority": 100
      }
    ]
  }
}
```

> **Note**
>
> ism_template フィールドは、新しく作成され、かつ、指定された index_patterns のいずれかに一致するインデックスにポリシーを自動的にアタッチします。この場合は、index- で始まるすべてのインデックスです。このフィールドは、環境内のインデックス形式に一致するように変更できます。詳細については、「ISM テンプレート」を参照してください。

6. ポリシーの snapshot セクションで、*snapshot-repo* を、ドメイン用に登録したスナップショットリポジトリの名前に置き換えます。オプションで、*ism-snapshot* を置き換えることもできます。これは、作成時のスナップショットの名前となります。
7. [作成] を選択します。ポリシーが [State management policies] (状態管理ポリシー) ページに表示されるようになりました。

ステップ 2: ポリシーを 1 つ以上のインデックスにアタッチする

ポリシーを作成したら、クラスター内の 1 つ以上のインデックスにアタッチします。

1. [Hot indicies] (ホットインデックス) タブにアクセスし、ステップ 1 で追加したすべてのサンプルインデックスをリスト化する opensearch_dashboards_sample を検索します。
2. すべてのインデックスを選択し、ポリシーの適用 を選択し、先ほど作成したism-policy-exampleポリシーを選択します。
3. [適用] を選択します。

[Policy managed indices] (ポリシー管理インデックス) ページで、インデックスがさまざまな状態を経るのをモニタリングできます。

インデックスロールアップによる Amazon OpenSearch Service でのインデックスのサマリー

Amazon OpenSearch Service のインデックスロールアップでは、古いデータを要約されたインデックスに定期的にロールアップすることで、ストレージコストを削減できます。

関心のあるフィールドを選択し、インデックスロールアップを使用して、より粗いタイムバケットに集約されたフィールドのみを持つ新しいインデックスを作成します。同じクエリパフォーマンスで、数か月または数年の履歴データをわずかなコストで保存できます。

インデックスのロールアップには、 OpenSearch または Elasticsearch 7.9 以降が必要です。

> **Note**
>
> このドキュメントは、Amazon OpenSearch Service でインデックスロールアップジョブの作成を開始するのに役立ちます。使用可能なすべての設定のリストや完全な API リファレンス

を含む包括的なドキュメントについては、 OpenSearch ドキュメントの「インデックスロールアップ」を参照してください。

インデックスロールアップジョブの作成

開始するには、ダッシュボードで OpenSearchインデックス管理を選択します。[ロールアップジョブ] を選択し、[ロールアップジョブの作成] を選択します。

ステップ 1: インデックスを設定する

ソースインデックスとターゲットインデックスを設定します。ソースインデックスは、ロールアップするインデックスです。ターゲットインデックスは、インデックスロールアップの結果が保存される場所です。

インデックスロールアップジョブを作成した後は、そのインデックス選択を変更することはできません。

ステップ 2: 集計とメトリクスを定義する

ロールアップする集計 (用語およびヒストグラム) とメトリクス (平均、合計、最大、最小、および値のカウント) を持つ属性を選択します。多くのスペースを節約することはないので、非常にきめ細かな属性を多く追加しないようにしてください。

ステップ 3: スケジュールを指定する

インデックスが取り込まれるときにロールアップするスケジュールを指定します。インデックスロールアップジョブは、デフォルトでは有効になっています。

ステップ 4: 確認して作成する

設定を確認し、[作成] を選択します。

ステップ5：ターゲットインデックスを検索する

標準 `_search` API を使用して、ターゲットインデックスを検索できます。ターゲットインデックス内のデータの内部構造にアクセスすることはできません。これは、プラグインがターゲットインデックスに合わせてバックグラウンドでクエリを自動的に書き換えるからです。これは、ソースインデックスとターゲットインデックスで同じクエリを使用できるようにするためのものです。

ターゲットインデックスのクエリを行うには、`size` を 0 に設定します。

```
GET target_index/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "avg_cpu": {
      "avg": {
        "field": "cpu_usage"
      }
    }
  }
}
```

> **Note**
>
> OpenSearch バージョン 2.2 以降では、1 つの request. OpenSearch versions で複数のロールアップインデックスの検索がサポートされています。バージョン 2.2 以前およびレガシー Elasticsearch OSS バージョンでは、検索ごとに 1 つのロールアップインデックスのみがサポートされています。

Amazon OpenSearch Service でのインデックスの変換

インデックスロールアップジョブでは、古いデータを集約されたインデックスにロールアップすることでデータ粒度を減らすことができますが、変換ジョブでは、特定のフィールドを中心とするデータの別の要約ビューを作成できるため、さまざまな方法でデータを視覚化または分析できます。

インデックス変換には、 OpenSearch Dashboards ユーザーインターフェイスと REST API があります。この機能には OpenSearch 1.0 以降が必要です。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service ドメインでインデックス変換の使用を開始するのに役立つ、インデックス変換の簡単な概要を提供します。包括的なドキュメントと REST API リファレンスについては、オープンソース OpenSearch ドキュメントの「インデックス変換」を参照してください。

インデックス変換ジョブの作成

クラスターにデータがない場合は、Dashboards OpenSearch 内のサンプルフライトデータを使用して変換ジョブを試してください。データを追加したら、 OpenSearch Dashboards を起動します。次に [インデックス管理]、[変換ジョブ]、および [変換ジョブの作成] を選択します。

ステップ 1: インデックスを選択する

[インデックス] セクションで、ソースとターゲットインデックスを選択します。既存のターゲットインデックスを選択することも、そのインデックス名を入力して新しいターゲットインデックスを作成することもできます。

ソースインデックスのサブセットのみを変換する場合は、データフィルターの追加 を選択し、OpenSearch クエリ DSL を使用してソースインデックスのサブセットを指定します。

ステップ 2: フィールドを選択する

インデックスを選択したら、変換ジョブで使用するフィールドと、グループ化と集計のどちらを使用するかを選択します。

- グループ化を使用して、変換されたインデックス内の別のバケットにデータを配置できます。例えば、サンプルフライトデータ内のすべての空港の目的地をグループ化する場合は、`DestAirportID` フィールドを `DestAirportID_terms` フィールドのターゲットフィールドにグループ化すると、変換ジョブの終了後に、変換されたインデックスにグループ化された空港 ID を見つけることができます。
- 一方、集約では、簡単な計算を実行できます。例えば、変換ジョブに集約を含めると、すべての飛行機のチケットの合計を計算する `sum_of_total_ticket_price` の新しいフィールドを定義できます。その後、変換されたインデックス内の新しいデータを分析できます。

ステップ 3: スケジュールを指定する

変換ジョブはデフォルトで有効になっており、スケジュールに基づいて実行されます。変換実行間隔については、間隔を分、時間、日数で指定します。

ステップ 4: 確認してモニタリングする

設定を確認し、[作成] を選択します。次に、[ジョブステータスの変換] 列をモニタリングします。

ステップ5：ターゲットインデックスを検索する

ジョブが終了したら、標準 _search API を使用してターゲットインデックスを検索します。

例えば、DestAirportID フィールドに基づいてフライトデータを変換する変換ジョブを実行した後、SFO の値を持つすべてのフィールドを返すために、次のリクエストを実行することができます。

```
GET target_index/_search
{
  "query": {
    "match": {
      "DestAirportID_terms" : "SFO"
    }
  }
}
```

Amazon OpenSearch Service のクラスター間レプリケーション

Amazon OpenSearch Service のクラスター間レプリケーションを使用すると、ユーザーインデックス、マッピング、メタデータを 1 つの OpenSearch サービスドメインから別のサービスドメインにレプリケートできます。クラスター間レプリケーションでは、障害発生時の災害対策が保証され、地理的に離れたデータセンター間でデータをレプリケートしてレイテンシーを削減できます。ドメイン間でAWS 転送されるデータには、標準のデータ転送料金がかかります。

クラスター間レプリケーションは、ローカルまたはフォロワーインデクスがリモートまたはリーダーインデックスからデータを取り出すアクティブ/パッシブレプリケーションモデルに従います。リーダーインデックスは、データのソース (データの複製元) のインデックスを意味します。リーダーインデックスは、データのターゲット (データの複製先) のインデックスを意味します。

クラスター間レプリケーションは、Elasticsearch 7.10 または OpenSearch 1.1 以降を実行しているドメインで使用できます。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service の観点からクラスター間レプリケーションを設定する方法について説明します。これには、 を使用してクラスター間接続 AWS Management Console を設定することが含まれます。これは、セルフマネージド OpenSearch クラスターでは不可能です。設定リファレンスや包括的な API リファレンスを

含む詳細なドキュメントについては、 OpenSearch ドキュメントの「クラスター間レプリケーション」を参照してください。

トピック

制限事項

クラスター間レプリケーションには、次の制約事項があります。

- Amazon OpenSearch Service ドメインとセルフマネージドクラスター OpenSearch または Elasticsearch クラスター間でデータをレプリケートすることはできません。
- 1 つのフォロワードメインから別のフォロワードメインにインデックスをレプリケートすることはできません。インデックスを複数のフォロワードメインにレプリケートする場合、レプリケートできるのは単一のリーダードメインからのみです。
- ドメインは、インバウンド接続とアウトバウンド接続の組み合わせを介して最大 20 のその他のドメインに接続できます。
- クラスター間接続を最初に設定する際には、リーダードメインがフォロワードメインと同じか、またはそれ以降のバージョンである必要があります。
- を使用してドメイン AWS CloudFormation を接続することはできません。
- M3 インスタンスとバースト可能 (T2 および T3) インスタンスではクラスター間レプリケーションを使用できません。

- UltraWarm またはコールドインデックス間でデータをレプリケートすることはできません。両方のインデックスがホットストレージにある必要があります。
- リーダードメインからインデックスを削除してもフォロワードメインの対応するインデックスが自動的に削除されることはありません。

前提条件

クラスター間レプリケーションをセットアップする前に、ドメインが次の要件を満たしていることを確認してください。

- Elasticsearch 7.10 または OpenSearch 1.1 以降
- Fine-grained access controlが有効
- N ode-to-node 暗号化が有効

アクセス許可の要件

レプリケーションを開始するには、リモート (リーダー) ドメインの `es:ESCrossClusterGet` アクセス許可を含める必要があります。リモートドメイン上で以下の IAM ポリシーが推奨されます。このポリシーでは、ドキュメントのインデックス作成や標準検索の実行など、他のオペレーションも実行できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/leader-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/leader-domain"
    }
  ]
}
```

es:ESCrossClusterGet アクセス許可が /leader-domain/* ではなく /leader-domain に適用されていることを確認してください。

管理者以外のユーザーがレプリケーションアクティビティを実行するには、適切なアクセス許可にマッピングする必要もあります。ほとんどのアクセス許可は、特定の REST API オペレーションに対応します。例えば、indices:admin/plugins/replication/index/_resume アクセス許可を使用すると、インデックスのレプリケーションを再開できます。アクセス許可の完全なリストについては、 OpenSearch ドキュメントの「レプリケーションアクセス許可」を参照してください。

> **Note**
>
> レプリケーションを開始してレプリケーションルールを作成するコマンドは、特殊なケースです。リーダードメインとフォロワードメインでバックグラウンドプロセスを呼び出すため、リクエストfollower_cluster_roleで leader_cluster_roleと を渡す必要があります。 OpenSearch サービスは、すべてのバックエンドレプリケーションタスクでこれらのロールを使用します。これらのロールのマッピングと使用については、 ドキュメントの「リーダーとフォロワーのクラスターロールをマッピングする」を参照してください。 OpenSearch

クラスター間接続のセットアップ

1 つのドメインから別のドメインにインデックスをレプリケートするには、ドメイン間でクラスター間接続を設定する必要があります。ドメイン間の接続する最も簡単な方法は、ドメインダッシュボードの [接続] タブを使用することです。設定 API または AWS CLI を使用することもできます。クラスター間レプリケーションは「プル」モデルに従うため、フォロワードメインから接続を開始します。

> **Note**
>
> 以前にクラスター間検索を実行するために 2 つのドメインを接続したことのある場合、その同じ接続をレプリケーションに使用することはできません。コンソールで、接続は

SEARCH_ONLY とマークされます。以前に接続された 2 つのドメイン間でレプリケーションを実行するには、接続を削除し、再度作成する必要があります。これを実行すれば、クラスター間検索とクラスター間レプリケーションの両方で接続が使用できるようになります。

接続をセットアップするには

1. Amazon OpenSearch Service コンソールで、フォロワードメインを選択し、接続タブに移動し、リクエスト を選択します。
2. [接続のエイリアス] に接続の名前を入力します。
3. AWS アカウント とリージョンのドメインに接続するか、別のアカウントまたはリージョンに接続するかを選択します。

   - AWS アカウント とリージョンのドメインに接続するには、ドメインを選択し、リクエスト を選択します。
   - 別の AWS アカウント またはリージョンのドメインに接続するには、リモートドメインの ARN を指定し、リクエスト を選択します。

OpenSearch サービスは接続リクエストを検証します。ドメインに互換性がない場合、接続は失敗します。正常に完了した検証は承認のために宛先ドメインに送信されます。宛先ドメインでリクエストが承認されたら、レプリケーションを開始できます。

クラスター間レプリケーションは双方向レプリケーションをサポートします。したがって、ドメイン A からドメイン B へのアウトバウンド接続と、ドメイン B からドメイン A への別のアウトバウンド接続を作成できます。その後、ドメイン A がドメイン B のインデックスに従い、ドメイン B がドメイン A のインデックスに従うようにレプリケーションを設定できます。

レプリケーションの開始

クラスター間接続を確立したら、データのレプリケーションを開始できます。まず、レプリケートするインデックスをリーダードメインに作成します。

```
PUT leader-01
```

そのインデックスをレプリケートするには、次のコマンドをフォロワードメインに送信します。

```
PUT _plugins/_replication/follower-01/_start
{
```

```
    "leader_alias": "connection-alias",
    "leader_index": "leader-01",
    "use_roles":{
       "leader_cluster_role": "all_access",
       "follower_cluster_role": "all_access"
    }
}
```

接続エイリアスは、ドメインダッシュボードの [接続] タブで確認できます。

この例では、管理者がリクエストを発行していると仮定し、説明を簡単にするために all_access を leader_cluster_role と follower_cluster_role に使用します。ただし、本番環境では、リーダーインデックスとフォロワーインデックスの両方でレプリケーションユーザーを作成してマッピングすることをお勧めします。ユーザー名は同一である必要があります。これらのロールとそのマッピング方法については、 ドキュメントの「リーダーとフォロワーのクラスターロールをマッピングする」を参照してください。 OpenSearch

レプリケーションの確認

レプリケーションが行われていることを確認するには、レプリケーションステータスを取得します。

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "SYNCING",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01",
  "syncing_details" : {
    "leader_checkpoint" : -5,
    "follower_checkpoint" : -5,
    "seq_no" : 0
  }
}
```

リーダーとフォロワーのチェックポイントの値は負の整数で始まり、ユーザーが持っているシャードの数を反映します (1 つのシャードの場合は -1、5 つのシャードの場合は -5 など)。変更を加えるたびに、値は正の整数で増加します。2 つの値が同じ場合、インデックスが完全に同期されていることを意味します。これらのチェックポイント値を使用して、ドメイン間のレプリケーションのレイテンシーを測定できます。

レプリケーションをさらに検証するには、リーダーインデックスにドキュメントを追加します。

```
PUT leader-01/_doc/1
{
   "Doctor Sleep":"Stephen King"
}
```

そして、それがフォロワーインデックスに表示されることを確認します。

```
GET follower-01/_search

{
    ...
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "follower-01",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "Doctor Sleep" : "Stephen King"
        }
      }
    ]
  }
}
```

レプリケーションの一時停止と再開

問題を修復する場合や、リーダードメインの負荷を削減する必要がある場合は、レプリケーションを一時的に一時停止できます。次のリクエストをフォロワードメインに送信します。空のリクエストボディを含めてください。

```
POST _plugins/_replication/follower-01/_pause
{}
```

次に、ステータスを取得して、レプリケーションが一時停止していることを確認します。

```
GET _plugins/_replication/follower-01/_status
```

```
{
  "status" : "PAUSED",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01"
}
```

変更作業が完了したら、レプリケーションを再開します。次のリクエストをフォロワードメインに送信します。空のリクエストボディを含めてください。

```
POST _plugins/_replication/follower-01/_resume
{}
```

12 時間以上一時停止してからレプリケーションを再開することはできません。レプリケーションを停止し、フォロワーインデックスを削除し、リーダーのレプリケーションを再開する必要があります。

レプリケーションの開始

レプリケーションを完全に停止すると、フォロワーインデックスはリーダーのフォローを解除し、標準インデックスになります。停止したレプリケーションを再開することはできません。

フォロアードメインからレプリケーションを停止します。空のリクエストボディを含めてください。

```
POST _plugins/_replication/follower-01/_stop
{}
```

自動フォロー

指定したパターンに一致するインデックスを自動的にレプリケートする 1 つのリーダードメインに対して、レプリケーションルールのセットを定義できます。リーダードメインのインデックスがパターン (例: books*) の 1 つに一致すると、一致するフォロワーインデックスがフォロワードメインに作成されます。 OpenSearch サービスは、パターンに一致する既存のインデックスと、作成した新しいインデックスをレプリケートします。フォロワードメインにすでに存在するインデックスはレプリケートされません。

すべてのインデックス (システムによって作成されたインデックスおよびフォロワードメインに既に存在するインデックスを除く) をレプリケートするには、ワイルドカード (*) パターンを使用します。

レプリケーションルールの作成

フォロアードメインにレプリケーションルールを作成し、クラスター間接続の名前を指定します。

```
POST _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name",
   "pattern": "books*",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

接続エイリアスは、ドメインダッシュボードの [接続] タブで確認できます。

この例では、管理者がリクエストを発行していると仮定し、説明を簡単にするために all_access をリーダードメインルールとフォロワードメインルールとして使用します。ただし、本番環境では、リーダーインデックスとフォロワーインデックスの両方でレプリケーションユーザーを作成してマッピングすることをお勧めします。ユーザー名は同一である必要があります。これらのロールとそのマッピング方法については、 ドキュメントの「リーダーとフォロワーのクラスターロールをマッピングする」を参照してください。 OpenSearch

ドメイン上の既存のレプリケーションルールのリストを取得するには、auto-follow stats API オペレーションを使用します。

ルールをテストするには、リーダードメインのパターンに一致するインデックスを作成します。

```
PUT books-are-fun
```

そして、そのレプリカがフォロアードメインに表示されていることを確認します。

```
GET _cat/indices

health status index          uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   books-are-fun  ldfHO78xYYdxRMULuiTvSQ     1   1          0            0
     208b           208b
```

レプリケーションルールの削除

レプリケーションルールを削除すると、 OpenSearch サービスはパターンに一致する新しいインデックスのレプリケーションを停止しますが、それらのインデックスのレプリケーションを停止するまで既存のレプリケーションアクティビティを続行します。

フォロアードメインからレプリケーションルールを削除します。

```
DELETE _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name"
}
```

接続されたドメインのアップグレード

クラスター間接続が可能な 2 つのドメインのエンジンバージョンをアップグレードするには、まずフォロワードメインをアップグレードしてから、リーダードメインをアップグレードします。それらの間の接続を削除しないでください。削除すると、レプリケーションが一時停止し、再開できなくなります。

リモート再インデックスを使用した Amazon OpenSearch Service インデックスの移行

リモート再インデックスを使用すると、ある Amazon OpenSearch Service ドメインから別のドメインにインデックスをコピーできます。インデックスは、任意の OpenSearch サービスドメイン、セルフマネージドクラスター OpenSearch 、Elasticsearch クラスターから移行できます。

リモートドメインとインデックスは、データのソース (データのコピー元) のドメインとインデックスを意味します。ローカルドメインとインデックスは、データのターゲット (データのコピー先) のドメインとインデックスを意味します。

リモート再インデックス作成には、ローカルドメインで OpenSearch 1.0 以降、または Elasticsearch 6.7 以降が必要です。リモートドメインは、ローカルドメインと同じまたはそれより前のメジャーバージョンである必要があります。Elasticsearch のバージョンは バージョンよりも OpenSearch低いと見なされ、Elasticsearch ドメインから OpenSearch ドメインにデータを再インデックスできます。同じメジャーバージョン内では、リモートドメインは任意のマイナーバージョン

にすることができます。例えば、Elasticsearch 7.10.x から 7.9 へのリモートインデックス再作成はサポートされていますが、Elasticsearch 7.10.x OpenSearch への 1.0 はサポートされていません。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service ドメイン間でデータを再インデックスする方法について説明します。詳細な手順やサポートされているオプションなど、`reindex`オペレーションの完全なドキュメントについては、ドキュメントの「インデックスの再作成 OpenSearch 」ドキュメントを参照してください。

トピック

前提条件

リモート再インデックスには、次の要件があります。

- リモートドメインは、ローカルドメインからアクセス可能である必要があります。VPC 内に存在するリモートドメインでは、ローカルドメインに VPC へのアクセスが必要です。このプロセスはネットワーク構成によって異なりますが、VPN またはマネージドネットワークへの接続、もしくは、ネイティブの VPC エンドポイント接続の使用が必要となる場合がほとんどです。詳細については、「the section called “VPC サポート”」を参照してください。
- リクエストは、他の REST リクエストと同様にリモートドメインによって承認される必要があります。リモートドメインできめ細かなアクセスコントロールが有効になっている場合は、リモートドメインで再インデックスを実行し、リモートドメインでインデックスを読み取るための許可が必要です。セキュリティに関するその他の考慮事項については、「the section called “きめ細かなアクセスコントロール”」を参照してください。
- 再インデックスプロセスを開始する前に、ローカルドメインに目的の設定でインデックスを作成することをお勧めします。

- ドメインでデータノードに T2 または T3 インスタンスタイプが使用されている場合、リモート再インデックスを使用することはできません。

OpenSearch サービスインターネットドメイン間でデータを再インデックスする

最も基本的なシナリオは、リモートインデックスがパブリックにアクセス可能なエンドポイントを持つローカルドメイン AWS リージョン と同じ にあり、IAM 認証情報に署名していることです。

リモートドメインから、再インデックス元のリモートインデックスと、再インデックス先のローカルインデックスを指定します。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

検証チェックを行うには、リモートドメインのエンドポイントの末尾に 443 を追加する必要があります。

インデックスがローカルドメインにコピーされたことを確認するには、次のリクエストをローカルドメインに送信します。

```
GET local_index/_search
```

リモートインデックスがローカルドメインとは異なるリージョンにある場合は、次のサンプルリクエストのように、リージョン名を渡します。

```
POST _reindex
{
  "source": {
    "remote": {
```

```
      "host": "https://remote-domain-endpoint:443",
      "region": "eu-west-1"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

AWS GovCloud (US) や中国リージョンのように分離されたリージョンの場合、IAM ユーザーがそれらのリージョンで認識されないため、エンドポイントにアクセスできない場合があります。

リモートドメインが基本認証 で保護されている場合は、ユーザー名とパスワードを指定します。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

リモートが VPC 内にあるときに OpenSearch サービスドメイン間でデータを再インデックスする

すべての OpenSearch サービスドメインは、独自の内部仮想プライベートクラウド (VPC) インフラストラクチャで構成されています。既存の OpenSearch サービス VPC に新しいドメインを作成すると、VPC 内のデータノードごとに Elastic Network Interface が作成されます。

リモート再インデックスオペレーションはリモート OpenSearch サービスドメインから実行されるため、独自のプライベート VPC 内で実行されるため、ローカルドメインの VPC にアクセスする方法が必要です。これを行うには、組み込みの VPC エンドポイント接続機能を使用して 経由で接続を確立するか AWS PrivateLink、プロキシを設定します。

ローカルドメインが OpenSearch バージョン 1.0 以降を使用している場合は、コンソールまたは AWS CLI を使用して AWS PrivateLink 接続を作成できます。 AWS PrivateLink 接続により、ローカル VPC 内のリソースは、同じ 内のリモート VPC 内のリソースにプライベートに接続できます AWS リージョン。

でデータを再インデックスする AWS Management Console

コンソールでリモートの再インデックスを使用すれば、VPC エンドポイント接続を共有している 2 つのドメイン間でインデックスをコピーできます。

1. で Amazon OpenSearch Service コンソールに移動しますhttps://console.aws.amazon.com/aos/。
2. 左側のナビゲーションペインで [Domains] (ドメイン) を選択します。
3. ローカルドメイン (データをコピーするドメイン) を選択します。選択すると、ドメインの詳細ページが開きます。一般情報の下にある [接続] タブを選択し、[リクエスト] を選択します。
4. [接続のリクエスト] ページで、接続モードに [VPC エンドポイント接続] を選択し、その他の関連情報を入力します。これらの詳細には、データをコピーするドメインであるリモートドメインが含まれます。次に、[Request] (リクエスト) を選択します。
5. リモートドメインの詳細ページに移動し、[接続] タブを選択してから [インバウンド接続] テーブルを見つけます。先ほど接続を作成したドメイン (ローカルドメイン) の名前の横にあるチェックボックスをオンにします。[承認] を選択します。
6. ローカルドメインに戻り、[Connections] (接続) タブを選択してから [Outbound connections] (アウトバウンド接続) テーブルを見つけます。2 つのドメイン間の接続がアクティブになると、テーブルの [Endpoint] (エンドポイント) 列でエンドポイントが利用可能になります。エンドポイントをコピーします。
7. ローカルドメインのダッシュボードを開き、左側のナビゲーションにある [Dev Tools] (開発ツール) を選択します。リモートのドメインインデックスがローカルのドメインにまだ存在していないことを確認するには、次の GET リクエストを実行します。を独自のインデックス名*remote-domain-index-name*に置き換えます。

```
GET remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

出力に、インデックスが見つからなかったことを示すエラーが表示されます。

8. GET リクエストの下で、以下のような POST リクエストを作成し、リモートホストとしてエンドポイントを使用します。

```
POST _reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

このリクエストを実行します。

9. GET リクエストをもう 1 度実行します。そうすると、出力にローカルインデックスが存在すると表示されます。このインデックスをクエリして、 が OpenSearchリモートインデックスからすべてのデータをコピーしたことを確認できます。

OpenSearch サービス API オペレーションでデータを再インデックスする

API でリモートの再インデックスを使用すれば、VPC エンドポイント接続を共有している 2 つのドメイン間でインデックスをコピーできます。

1. CreateOutboundConnection API オペレーションを使用して、ローカルドメインからリモートドメインへの新しい接続をリクエストします。

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/cc/outboundConnection

{
   "ConnectionAlias": "remote-reindex-example",
   "ConnectionMode": "VPC_ENDPOINT",
   "LocalDomainInfo": {
      "AWSDomainInformation": {
```

```
            "DomainName": "local-domain-name",
            "OwnerId": "aws-account-id",
            "Region": "region"
        }
    },
    "RemoteDomainInfo": {
        "AWSDomainInformation": {
            "DomainName": "remote-domain-name",
            "OwnerId": "aws-account-id",
            "Region": "region"
        }
    }
}
```

ConnectionId が応答で返されます。この ID は次のステップで使用するので保存しておきます。

2. 接続 ID で AcceptInboundConnection API オペレーションを使用して、ローカルドメインからのリクエストを承認します。

```
PUT https://es.region.amazonaws.com/2021-01-01/opensearch/cc/
inboundConnection/ConnectionId/accept
```

3. DescribeOutboundConnections API オペレーションを使用して、リモートドメインのエンドポイントを取得します。

```
{
    "Connections": [
        {
            "ConnectionAlias": "remote-reindex-example",
            "ConnectionId": "connection-id",
            "ConnectionMode": "VPC_ENDPOINT",
            "ConnectionProperties": {
                "Endpoint": "connection-endpoint"
            },
            ...
        }
    ]
}
```

connection-endpoint はステップ 5 で使用するので保存しておきます。

4. リモートのドメインインデックスがローカルのドメインにまだ存在していないことを確認するには、次の GET リクエストを実行します。を独自のインデックス名*remote-domain-index-name*に置き換えます。

```
GET local-domain-endpoint/remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

   出力に、インデックスが見つからなかったことを示すエラーが表示されます。

5. POST リクエストを作成し、次のようにエンドポイントをリモートホストとして使用します。

```
POST local-domain-endpoint/_reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

   このリクエストを実行します。

6. GET リクエストをもう 1 度実行します。そうすると、出力にローカルインデックスが存在すると表示されます。このインデックスをクエリして、 が OpenSearchリモートインデックスからすべてのデータをコピーしたことを確認できます。

リモートドメインが VPC 内でホストされ、VPC エンドポイントの接続機能の使用を希望しない場合は、パブリックにアクセス可能なエンドポイントを使ってプロキシを設定します。この場合、OpenSearch サービスには VPC にトラフィックを送信する機能がないため、パブリックエンドポイントが必要です。

VPC モードでドメインを実行する場合、1 つまたは複数のエンドポイントが VPC に配置されます。ただし、これらのエンドポイントは VPC 内のドメインに入るトラフィックのみを対象としており、VPC 自体へのトラフィックは許可されません。

リモートの再インデックスコマンドは、リモートドメインから実行されるため、発信元のトラフィックはそれらのエンドポイントを使ってリモートドメインにアクセスすることはできません。このユースケースでプロキシが必要となるのはそのためです。プロキシドメインには、公開認証機関 (CA) によって署名された証明書が必要です。自己署名証明書またはプライベート CA 署名証明書はサポートされていません。

OpenSearch サービス以外のドメイン間でデータを再インデックスする

セルフマネージド EC2 インスタンスのように、リモートインデックスが OpenSearch Service の外部でホストされている場合は、 `external`パラメータを に設定しますtrue。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password",
      "external": true
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

この場合、ユーザー名とパスワードを使用した基本認証のみがサポートされます。リモートドメインには、パブリックにアクセス可能なエンドポイント (ローカル OpenSearch サービスドメインと同じ VPC にある場合でも) と、パブリック CA によって署名された証明書が必要です。自己署名証明書またはプライベート CA 署名証明書はサポートされていません。

大規模なデータセットの再インデックスを行う

リモート再インデックスは、次のデフォルト値を使用してリモートドメインにスクロールリクエストを送信します。

- 5 分の検索コンテキスト
- 30 秒のソケットタイムアウト
- 1,000 のバッチサイズ

データに合わせてこれらのパラメータを調整することをお勧めします。大きなドキュメントの場合は、バッチサイズを小さくするか、タイムアウトを長くするかを検討してください。詳細については、「スクロール検索」を参照してください。

```
POST _reindex?pretty=true&scroll=10h&wait_for_completion=false
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "socket_timeout": "60m"
    },
    "size": 100,
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

パフォーマンスを向上させるために、ローカルインデックスに次の設定を追加することもお勧めします。

```
PUT local_index
{
  "settings": {
    "refresh_interval": -1,
    "number_of_replicas": 0
  }
}
```

再インデックスプロセスが完了したら、目的のレプリカの数を設定し、更新間隔の設定を削除できます。

クエリを介して選択したドキュメントのサブセットのみの再インデックスを行うには、次のリクエストをローカルドメインに送信します。

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index",
    "query": {
      "match": {
        "field_name": "text"
      }
    }
  },
  "dest": {
    "index": "local_index"
  }
}
```

リモート再インデックスはスライスをサポートしていないため、同じリクエストに対して複数のスクロールオペレーションを並行して実行することはできません。

リモート再インデックス設定

標準のインデックス再作成オプションに加えて、 OpenSearch Service は次のオプションをサポートしています。

オプション	有効な値	説明	必須
外部	ブール値	リモートドメインが OpenSearch サービスドメインでない場合、または 2 つの VPC ドメイン間でインデックスを再作成する場合は、 を として指定しますtrue。	いいえ
region	文字列	リモートドメインが別のリージョンにあ	いいえ

オプション	有効な値	説明	必須
		る場合は、リージョン名を指定します。	

データストリームを使用した Amazon OpenSearch Service での時系列データの管理

時系列データを管理する一般的なワークフローには、ロールオーバーインデックスエイリアスの作成、書き込みインデックスの定義、バッキングインデックスの共通のマッピングと設定の定義など、複数の手順が含まれます。

Amazon OpenSearch Service のデータストリームは、この初期設定プロセスを簡素化するのに役立ちます。Data Streams は、通常は追加のみであるアプリケーションログなどの時間ベースのデータに対してすぐに機能します。

データストリームには OpenSearch バージョン 1.0 以降が必要です。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service ドメインでデータストリームを開始するのに役立つ基本的な手順について説明します。包括的なドキュメントについては、OpenSearch ドキュメントの「データストリーム」を参照してください。

Data Streams の使用開始

データストリームは、内部的に複数のバッキングインデックスで構成されます。検索リクエストはすべてのバッキングインデックスにルーティングされますが、インデックス作成リクエストは最新の書き込みインデックスにルーティングされます。

ステップ 1: インデックステンプレートを作成する

データストリームを作成するには、まずインデックスのセットをデータストリームとして設定するインデックステンプレートを作成する必要があります。`data_stream` オブジェクトは、通常のインデックステンプレートではなくデータストリームであることを示します。インデックスパターンは、次のデータストリームの名前と一致します。

```
PUT _index_template/logs-template
{
  "index_patterns": [
    "my-data-stream",
    "logs-*"
  ],
  "data_stream": {},
  "priority": 100
}
```

この場合、取り込まれた各ドキュメントには @timestamp フィールドがある必要があります。また、独自のカスタムタイムスタンプフィールドを data_stream オブジェクトのプロパティとして定義できます。

```
PUT _index_template/logs-template
{
  "index_patterns": "my-data-stream",
  "data_stream": {
    "timestamp_field": {
      "name": "request_time"
    }
  }
}
```

ステップ 2: データストリームの作成

インデックステンプレートを作成したら、データストリームを作成せずにデータの取り込みを直接開始できます。

data_stream オブジェクトと一致するインデックステンプレートがあるため、 OpenSearch は自動的にデータストリームを作成します。

```
POST logs-staging/_doc
{
  "message": "login attempt failed",
  "@timestamp": "2013-03-01T00:00:00"
}
```

手順 3: データストリームにデータを取り込む

データストリームにデータを取り込むには、通常のインデックス API を使用できます。インデックスを作成するすべてのドキュメントにタイムスタンプフィールドがあることを確認します。タイムスタンプフィールドのないドキュメントを取り込もうとすると、エラーが発生します。

```
POST logs-redis/_doc
{
  "message": "login attempt",
  "@timestamp": "2013-03-01T00:00:00"
}
```

ステップ 4: データストリームの検索

通常のインデックスやインデックスエイリアスを検索するのと同じように、データストリームを検索できます。検索オペレーションは、すべてのバッキングインデックス (ストリーム内に存在するすべてのデータ) に適用されます。

```
GET logs-redis/_search
{
  "query": {
    "match": {
      "message": "login"
    }
  }
}
```

手順 5: データストリームのロールオーバー

インデックスステート管理 (ISM) ポリシーをセットアップして、データストリームのロールオーバープロセスを自動化できます。ISM ポリシーは、作成時にバッキングインデックスに適用されます。ポリシーをデータストリームに関連付けると、そのデータストリームの将来のバッキングインデックスにのみ影響します。また、`rollover_alias` 設定を提供する必要はありません。ISM ポリシーがバッキングインデックスからこの情報を推測するからです。

> **Note**
>
> バッキングインデックスをコールドストレージ に移行すると、 OpenSearch はこのインデックスをデータストリームから削除します。インデックスを に戻してもUltraWarm、インデックスは元のデータストリームの一部ではなく、独立したままになります。インデックスが

データストリームから削除された後、ストリームに対して検索しても、インデックスのデータは返されません。

> **Warning**
>
> データストリームの書き込みインデックスは、コールドストレージに移行できません。データストリーム内のデータをコールドストレージに移行する場合は、移行前にデータストリームをロールオーバーする必要があります。

ステップ 6: OpenSearch Dashboards でデータストリームを管理する

OpenSearch Dashboards からデータストリームを管理するには、OpenSearch Dashboards を開き、インデックス管理 を選択し、インデックス またはポリシーマネージドインデックス を選択します。

ステップ 7: データストリームを削除する

削除オペレーションは、まずデータストリームのバッキングインデックスを削除してから、データストリーム自体を削除します。

データストリームとその非表示のバッキングインデックスすべてを削除するには、次の手順に従います。

```
DELETE _data_stream/name_of_data_stream
```

Amazon OpenSearch サービスでのデータのモニタリング

アラートと異常検出を用いて Amazon OpenSearch Service 内のデータの積極的なモニタリングを行います。アラートをセットアップして、データが特定のしきい値を超えたときに通知を受信します。異常検出では、機械学習の使用により、ストリーミングデータ内の異常値が自動的に検出されます。異常検出とアラートを組み合わせると、異常が検出された場合にすぐに通知を受けることができます。

トピック

Amazon OpenSearch Service でのアラートの設定

1 つ以上のインデックスのデータが特定の条件を満たしたときに通知を受け取るように Amazon OpenSearch Service のアラートを設定します。例えば、アプリケーションで 1 時間に 5 つを超える HTTP 503 エラーが記録された場合に E メールが届くようにしたり、過去 20 分以内に新しいドキュメントにインデックスが付けられなかった場合にデベロッパーに連絡したりできます。

アラートには、 OpenSearch または Elasticsearch 6.2 以降が必要です。

> **Note**
>
> このドキュメントでは、アラートの概要と、Amazon OpenSearch Service ドメインでのアラートとオープンソース OpenSearch クラスターでのアラートの違いについて説明します。包括的な API リファレンス、複合モニターで使用可能なリクエストフィールドのリスト、使用可能なトリガー変数とアクション変数の説明など、アラートに関する詳細なドキュメントについては、 OpenSearch ドキュメントの「アラート」を参照してください。

トピック

アラートの許可

アラートはきめ細かなアクセスコントロールをサポートしています。ユースケースに合わせたアクセス許可の混在とマッチングの詳細については、 OpenSearch ドキュメントの「アラートセキュリティ」を参照してください。

OpenSearch Dashboards 内のアラートページにアクセスするには、少なくともalerting_read_access事前定義されたロールにマッピングされているか、同等のアクセス許可が付与されている必要があります。このロールは、アラート、送信先、モニターを表示するアクセス許可を付与しますが、アラートの確認や送信先やモニターの変更を行うアクセス許可は付与しません。

アラートの開始方法

アラートを作成するには、モニター を設定します。これは、定義されたスケジュールで実行され、OpenSearch インデックスをクエリするジョブです。また、1 つまたは複数のトリガーも設定します。イベントを生成する条件をトリガーで定義します。最後に、アクションを設定します。アラートがトリガーされるとアクションが実行されます。

アラートを開始するには

1. OpenSearch ダッシュボードのメインメニューからアラートを選択し、モニターの作成を選択します。
2. クエリごと、バケットごと、クラスターごとのメトリクス、またはドキュメントごとのモニターを作成します。手順については、「Create a monitor」(モニターの作成) を参照してください。
3. [Triggers] (トリガー) では、1 つまたは複数のトリガーを作成します。手順については、「Create triggers」(トリガーの作成) を参照してください。
4. [Actions] (アクション) では、アラートの通知チャネルを設定します。Slack、Amazon Chime、カスタム webhook、Amazon SNS のいずれかを選択します。ご想像のとおり、通知にはチャネルへの接続が必要です。例えば、 OpenSearch サービスドメインがインターネットに接続して Slack チャンネルに通知したり、カスタムウェブフックをサードパーティーサーバーに送信したりできる必要があります。 OpenSearch サービスドメインがアラートを送信するには、カスタムウェブフックにパブリック IP アドレスが必要です。

> **Tip**
>
> アクションが正常にメッセージを送信した後、そのメッセージへのアクセスの保護 (Slack チャネルへのアクセスなど) はユーザーの責任となります。ドメインに機密デー

タが含まれている場合は、アクションなしでトリガーを使用し、Dashboards でアラートを定期的にチェックすることを検討してください。

通知

アラートは、通知用の統合システムである通知と統合されます OpenSearch。通知により、どの通信サービスを使用するかを設定したり、関連する統計やトラブルシューティング情報を確認したりできます。包括的なドキュメントについては、 OpenSearch ドキュメントの「通知」を参照してください。

通知を使用するには、ドメインが OpenSearch バージョン 2.3 以降を実行している必要があります。

> **Note**
>
> OpenSearch 通知は、 OpenSearch サービスソフトウェアの更新、Auto-Tune ???の機能強化、およびその他の重要なドメインレベルの情報に関する詳細を提供するサービス通知 とは別のものです。 OpenSearch 通知はプラグイン固有です。

通知チャネルは、バージョン 2.0 以降の OpenSearchアラート送信先に置き換えられました。送信先は正式に廃止され、今後すべてのアラート通知はチャネルを通じて管理されます。

ドメインをバージョン 2.3 以降にアップグレードすると (2.x のサービス OpenSearch サポートは 2.3 で始まるため)、既存の送信先は自動的に通知チャネルに移行されます。送信先が移行できなかった場合、モニターは通知チャネルに移行されるまでその送信先を使用し続けます。詳細については、OpenSearch ドキュメントの「送信先に関する質問」を参照してください。

通知を開始するには、 OpenSearch Dashboards にサインインし、通知 、チャネル 、チャネル の作成 を選択します。

Amazon Simple Notification Service (Amazon SNS) は、通知でサポートされているチャネルタイプです。ユーザーを認証するには、Amazon SNS へのフルアクセス権をユーザーに付与するか、Amazon SNS へのアクセス許可を持つ IAM ロールをユーザーに割り当てる必要があります。手順については、[Amazon SNS as a channel type] (チャネルタイプとしての Amazon SNS) を参照してください。

差異

のオープンソースバージョンと比較して OpenSearch、Amazon OpenSearch Service でのアラートにはいくつかの大きな違いがあります。

アラート設定

OpenSearch サービスでは、次のアラート設定を変更できます。

- `plugins.scheduled_jobs.enabled`
- `plugins.alerting.alert_history_enabled`
- `plugins.alerting.alert_history_max_age`
- `plugins.alerting.alert_history_max_docs`
- `plugins.alerting.alert_history_retention_period`
- `plugins.alerting.alert_history_rollover_period`
- `plugins.alerting.filter_by_backend_roles`

他のすべての設定では、変更できないデフォルト値が使用されます。

アラートを無効にするには、次のリクエストを送信します。

```
PUT _cluster/settings
{
  "persistent" : {
    "plugins.scheduled_jobs.enabled" : false
  }
}
```

次のリクエストは、デフォルトの 30 日ではなく、7 日後に履歴インデックスを自動的に削除するようにアラートを設定します。

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.alerting.alert_history_retention_period": "7d"
  }
}
```

以前にモニターを作成し、毎日のアラートインデックスの作成を停止する場合は、すべてのアラート履歴インデックスを削除します。

```
DELETE .plugins-alerting-alert-history-*
```

履歴インデックスのシャード数を減らすには、インデックステンプレートを作成します。次のリクエストは、1 つのシャードと 1 つのレプリカにアラートを送信するための履歴インデックスを設定します。

```
PUT _index_template/template-name
{
  "index_patterns": [".opendistro-alerting-alert-history-*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 1
    }
  }
}
```

データ損失の許容値によっては、ゼロレプリカの使用を検討することもできます。インデックステンプレートの作成と管理の詳細については、 OpenSearch ドキュメントの「インデックステンプレート」を参照してください。

Amazon OpenSearch Service での異常検出

Amazon OpenSearch Service での異常検出は、Random Cut Forest (RCF) アルゴリズムを使用して、 OpenSearch データの異常をほぼリアルタイムで自動的に検出します。RCFは、受信データストリームのスケッチをモデル化する教師なしの機械学習アルゴリズムです。このアルゴリズムにより、受信データポイントごとに、`anomaly grade` および `confidence score` の値が計算されます。異常検出は、これらの値を使用して、データの通常の変動と異常を区別します。

異常検出プラグインとアラートプラグインをペアリングして、異常が検出されるとすぐに通知できます。

異常検出は、任意の OpenSearch バージョンまたは Elasticsearch 7.4 以降を実行しているドメインで使用できます。`t2.micro` と `t2.small` を除くすべてのインスタンスタイプで、異常検出がサポートされています。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service のコンテキストにおける異常検出の概要を説明します。詳細なステップ、API リファレンス、使用可能なすべての設定のリファレンス、視覚化とダッシュボードを作成するステップを含む包括的なドキュメントについては、オープンソース OpenSearch ドキュメントの「異常検出」を参照してください。

前提条件

異常検出には、次のような前提条件があります。

- 異常検出には、 OpenSearch または Elasticsearch 7.4 以降が必要です。
- 異常検出は、Elasticsearch バージョン 7.9 以降および のすべてのバージョンで、きめ細かなアクセスコントロールのみをサポートします OpenSearch。Elasticsearch 7.9 より前のバージョンでは、ディテクターを作成、表示、管理できるのは管理者ユーザーのみです。
- ドメインがきめ細かなアクセスコントロールを使用している場合、管理者以外のユーザーは、ディテクターを表示したり、ディテクターを作成および管理したりするために、 OpenSearch Dashboards `anomaly_full_access` の`anomaly_read_access`ロールにマッピングする必要があります。

異常検出の開始方法

開始するには、 OpenSearch ダッシュボードで異常検出を選択します。

ステップ 1: ディテクターを作成する

ディテクターは、個別の異常検出タスクです。複数のディテクターを作成し、すべてのディテクターを同時に実行すると、それぞれ別々のソースからのデータを分析できます。

ステップ 2: ディテクターに機能を追加する

機能とは、異常をチェックするインデックス内のフィールドを指します。ディテクターでは、1 つ以上の機能について異常を検出できます。機能ごとに、以下の集計の 1 つを選択する必要があります: `average()`、`sum()`、`count()`、`min()`、または `max()`。

> **Note**
>
> `count()` 集約方法は、 OpenSearch および Elasticsearch 7.7 以降でのみ使用できます。Elasticsearch 7.4では、次のようなカスタム式を使用します。
>
> ```
> {
>   "aggregation_name": {
>      "value_count": {
>         "field": "field_name"
>      }
>   }
> }
> ```

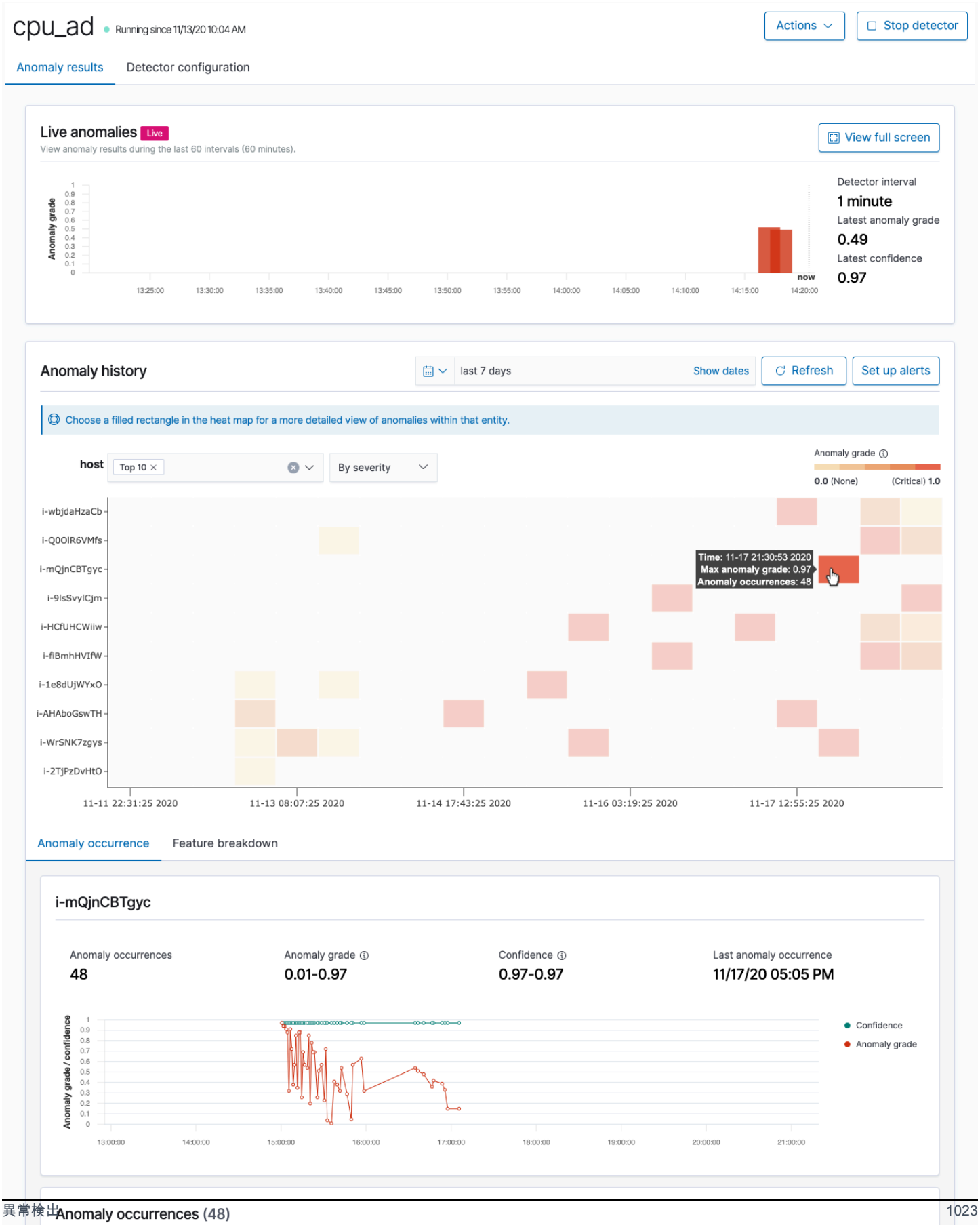
集計方法によって、検出される異常の性質も異なります。例えば、`min()` を選択した場合、ディテクターは機能の最小値に基づいて異常を検出します。`average()` を選択した場合、ディテクターは機能の平均値に基づいて異常を検出します。ディテクターごとに追加できる機能は、最大 5 つです。

以下のオプション設定を指定できます (Elasticsearch 7.7 以降で使用可能) 。

- Category field (カテゴリフィールド) - IPアドレス、製品 ID、国コードなどのディメンションでデータを分類またはスライスします。
- Window size (ウィンドウサイズ) - 検出ウィンドウで考慮するデータストリームからの集計間隔の数を設定します。

機能を設定したならば、サンプルの異常をプレビューし、必要に応じて機能の設定を調整します。

ステップ 3: 結果を観察する



- Live anomalies (ライブ異常) - 過去 60 間隔のライブ異常結果を表示します。たとえば、間隔を 10 に設定すると、過去 600 分間の結果が表示されます。このグラフは 30 秒ごとに更新されます。
- Anomaly history (異常履歴) - 対応する測定指標とともに、異常の等級を示します。
- Feature breakdown (機能分類) - 集計方法に基づいて機能を分類します。ディテクターの日時の範囲は変更できます。
- Anomaly occurrence (異常出現) - 検出された各異常の `Start time`、`End time`、`Data confidence`、および `Anomaly grade` を示します。

  カテゴリフィールドを設定すると、異常エンティティの結果を関連付ける、追加のヒートマップチャートが表示されます。塗りつぶされた長方形を選択すると、異常のより詳細なビューが表示されます。

ステップ 4: アラートのセットアップ

異常が検出されたときに通知を受け取るためのモニターを作成するには、[アラートのセットアップ] を選択します。プラグインにより、アラートを設定できる [モニターの追加] ページにリダイレクトされます。

チュートリアル: 異常検出で高い CPU 使用率を検出する

このチュートリアルでは、Amazon OpenSearch Service で異常ディテクターを作成して CPU 使用率が高いことを検出する方法を示します。 OpenSearch Dashboards を使用して、CPU 使用率をモニタリングし、CPU 使用率が指定されたしきい値を超えたときにアラートを生成するようにディテクターを設定します。

> **Note**
>
> これらの手順は の最新バージョンに適用 OpenSearch され、過去のバージョンでは若干異なる場合があります。

前提条件

- Elasticsearch 7.4 以降、または任意の OpenSearchバージョンを実行している OpenSearch サービスドメインが必要です。
- CPU 使用率データを含むアプリケーションログファイルをクラスターに取り込む必要があります。

ステップ 1: ディテクターを作成する

まず、CPU 使用率データの異常を識別するディテクターを作成します。

1. OpenSearch ダッシュボードの左側のパネルメニューを開き、異常検出 を選択し、ディテクターの作成 を選択します。
2. ディテクターに **high-cpu-usage** という名前を付けます。
3. データソース用に、異常を特定する CPU 使用率ログファイルを含むインデックスを選択します。
4. データの [Timestamp field] (Timestamp フィールド) を選択します。必要に応じて、データフィルターを追加できます。このデータフィルターは、データソースのサブセットのみを分析し、無関係なデータのノイズを低減します。
5. [Detector interval] (ディテクターの間隔) を 2 分に設定します。この間隔は、ディテクターがデータを収集する時間 (1 分単位) を定義します。
6. [Window delay] (ウィンドウの遅延) で、[1-minute] (1 分間) の遅延を追加します。この遅延により、ウィンドウ内のすべてのデータが確実に存在しているようにするために、処理時間が長くなります。
7. [次へ] をクリックします。異常検出ダッシュボードで、ディテクター名の下にある [Configure model] (モデルを設定) を選択します。
8. [Feature name] (機能名) で、**max_cpu_usage** を入力します。[Feature state] (機能の状態) で、[Enable feature] (機能を有効にする) を選択します。
9. [Find anomalies based on] (次に基づいて異常を検索) で、[Field value] (フィールド値) を選択します。
10. [Aggregation method] (集計方法) で、[**max()**] を選択します。
11. [Field] (フィールド) で、異常をチェックするデータ内のフィールドを選択します。例えば、`cpu_usage_percentage` と呼ばれることがあります。
12. 他のすべての設定をデフォルトのままにして、[Next] (次へ) を選択します。
13. ディテクタージョブの設定を無視して、[Next] (次へ) を選択します。
14. ポップアップウィンドウで、(自動または手動で) ディテクターをいつ開始するかを選択し、[Confirm] (確認) を選択します。

これでディテクターが設定され、初期化された後に、ディテクターパネルの [Real-time results] (リアルタイム結果) セクションで CPU 使用率のリアルタイム結果を確認できるようになります。[Live

anomalies] (ライブ異常) セクションには、データがリアルタイムで取り込まれているときに発生する異常が表示されます。

ステップ 2: アラートを設定する

ディテクターを作成したので、ディテクター設定で指定された条件を満たす CPU 使用率を検出した場合に Slack にメッセージを送信するよう促すアラートを呼び出すモニターを作成します。1 つ以上のインデックスのデータがアラートを呼び出す条件を満たすと、Slack の通知が届きます。

1. OpenSearch ダッシュボードの左側のパネルメニューを開き、アラート を選択し、モニターの作成 を選択します。
2. モニターの名前を指定します。
3. [Monitor type] (モニターの種類) で、[Per-query monitor] (クエリごとのモニター) を選択します。クエリごとのモニターは、指定されたクエリを実行し、トリガーを定義します。
4. [Monitor defining method] (モニタリングの定義方法) で [Anomaly detector] (異常ディテクター) を選択し、前のセクションで作成したディテクターを [Detector] (ディテクター) ドロップダウン メニューから選択します。
5. [Schedule] (スケジュール) で、モニターがデータを収集する頻度とアラートを受け取る頻度を選択します。このチュートリアルでは、7 分ごとに実行するようにスケジュールを設定します。
6. [Triggers] (トリガー) セクションで、[Add trigger] (トリガーを追加) を選択します。[Trigger name] (トリガー名) で、**`High CPU usage`** を入力します。このチュートリアルでは、[Severity level] (重大度レベル) で、最高レベルの重大度である [1] を選択します。
7. [Anomaly grade threshold] (異常グレードのしきい値) で、[IS ABOVE] (超える) を選択します。その下のメニューで、適用するグレードしきい値を選択します。このチュートリアルでは、[Anomaly grade] (異常グレード) を [0.7] に設定します。
8. [Anomaly confidence threshold] (異常の信頼度のしきい値) で、[IS ABOVE] (超える) を選択します。その下のメニューで、[Anomaly grade] (異常グレード) と同じ数値を入力します。このチュートリアルでは、[Anomaly confidence threshold] (異常の信頼度のしきい値) を [0.7] に設定します。
9. [Actions] (アクション) セクションで、[Destination] (宛先) を選択します。[Name] (名前) フィールドで、宛先の名前を選択します。[Type] (タイプ) メニューで、[Slack] を選択します。[Webhook URL] (ウェブフック URL) フィールドで、アラートを受信するウェブフック URL を入力します。詳細については、「Sending messages using incoming webhooks」(着信ウェブフックを使用したメッセージの送信) を参照してください。

10[作成] を選択します。

関連リソース

- the section called “アラート”

- the section called “異常検出”
- 異常検出 API

Amazon OpenSearch Service の機械学習

ML Commons は、トランスポートおよび REST API コールを通じて一般的な機械学習 (ML) アルゴリズムのセットを提供する OpenSearch プラグインです。これらの呼び出しは、ML リクエストごとに適切なノードとリソースを選択し、ML タスクを監視して稼働時間を確保します。これにより、既存のオープンソースの ML アルゴリズムを活用し、新しい ML 機能の開発に必要な労力を削減できます。プラグインの詳細については、 OpenSearch ドキュメントの「機械学習」を参照してください。この章では、Amazon OpenSearch Service で プラグインを使用する方法について説明します。

トピック

- 用の Amazon OpenSearch Service ML コネクタ AWS のサービス
- サードパーティープラットフォーム用の Amazon OpenSearch Service ML コネクタ
- AWS CloudFormation を使用してセマンティック検索用のリモート推論を設定する
- サポートされていない ML Commons 設定
- OpenSearch サービスフローフレームワークテンプレート

用の Amazon OpenSearch Service ML コネクタ AWS のサービス

Amazon OpenSearch Service 機械学習 (ML) コネクタを別の で使用する場合は AWS のサービス、IAM ロールを設定して OpenSearch 、サービスを安全にそのサービスに接続する必要があります。 AWS のサービス これにより、Amazon SageMaker と Amazon Bedrock を含めるようにコネクタを設定できます。このチュートリアルでは、 OpenSearch サービスから SageMaker ランタイムへのコネクタの作成方法について説明します。コネクタの詳細については、「サポートされているコネクタ」を参照してください。

トピック

- 前提条件
- OpenSearch サービスコネクタを作成する

前提条件

コネクタを作成するには、Amazon SageMaker Domain エンドポイントと、 OpenSearch サービスアクセスを許可する IAM ロールが必要です。

Amazon SageMaker ドメインをセットアップする

機械学習モデルをデプロイするには、Amazon SageMaker SageMaker デベロッパーガイドの「Amazon でモデルをデプロイする」を参照してください。AI コネクタを作成するために必要な、モデルのエンドポイント URL をメモしておきます。

IAM ロールを作成する

ランタイムアクセス許可を OpenSearch サービスに委任する IAM SageMaker ロールを設定します。新しいロールを作成するには、IAM ユーザーガイドの「IAM ロール (コンソール) の作成」を参照してください。なお、同じ権限セットを持っていれば、既存のロールを使用できます。 AWS マネージドロールを使用する代わりに新しいロールを作成する場合は、このチュートリアルopensearch-sagemaker-roleの を自分のロールの名前に置き換えます。

1. 次の マネージド IAM ポリシーを新しいロールにアタッチして、 OpenSearch サービスが SageMaker エンドポイントにアクセスできるようにします。ポリシーをロールにアタッチするには、「IAM ID アクセス許可の追加」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sagemaker:InvokeEndpointAsync",
                "sagemaker:InvokeEndpoint"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

2. ロールの信頼関係を編集するには、「ロールの信頼ポリシーの変更」の手順に従ってください。Principal ステートメントで OpenSearch サービスを指定する必要があります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
```

```
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "opensearchservice.amazonaws.com"
                ]
            }
        }
    ]
}
```

aws:SourceAccount および aws:SourceArn 条件キーを使用してアクセスを特定のドメインに制限することをおすすめします。SourceAccount はドメインの所有者に属する AWS アカウント ID で、 SourceArnはドメインの ARN です。例えば、次の条件ブロックを信頼ポリシーに追加できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

のアクセス許可を設定します。

コネクタを作成するには、IAM ロールを OpenSearch サービスに渡すアクセス許可が必要です。さらに、es:ESHttpPost アクションへのアクセスも必要です。これらの両方の許可を付与するには、リクエストの署名に認証情報が使用されている IAM ロールまたはユーザーに次のポリシーをアタッチします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
    },
```

```
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

ユーザーまたはロールがロールを渡すための iam:PassRole アクセス許可を持っていない場合、次のステップでリポジトリを登録しようとすると、認証エラーが発生することがあります。

OpenSearch Dashboards で ML ロールをマッピングする (きめ細かなアクセスコントロールを使用している場合)

きめ細かいアクセス制御では、コネクタの設定時に追加の手順が必要になります。HTTP 基本認証を他のすべての目的で使用する場合でも、opensearch-sagemaker-roleを渡すための iam:PassRole 許可を持っている IAM ロールまたはユーザーに ml_full_access ロールをマップする必要があります。

1. OpenSearch サービスドメインの OpenSearch Dashboards プラグインに移動します。Dashboards エンドポイントは、 OpenSearch サービスコンソールのドメインダッシュボードにあります。
2. メインメニューから [セキュリティ]、[ロール] を選択し、[ml_full_access] ロールを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。
4. [バックエンドロール] で、opensearch-sagemaker-role を渡すためのアクセス許可があるロールの ARN を追加します。

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

5. [マップ] を選択し、ユーザーまたはロールが [マッピングされたユーザー] の下に表示されていることを確認します。

OpenSearch サービスコネクタを作成する

コネクタを作成するには、 OpenSearch サービスドメインエンドポイントにPOSTリクエストを送信します。署名付きリクエストを送信するには、curl、サンプル Python クライアント、Postman、またはその他の方法を使用できます。Kibana コンソールでは、POST リクエストは使用できないことにご注意ください。リクエストは以下のような形式です。

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
```

ドメインが仮想プライベートクラウド (VPC) に存在する場合は、リクエストが正常に AI コネクタを作成するにはコンピュータが VPC に接続されていることが必要です。VPC へのアクセスはネットワーク構成によって異なりますが、通常は VPN あるいは社内ネットワークへの接続が必要になります。 OpenSearch サービスドメインにアクセスできることを確認するには、ウェブブラウザで `https://your-vpc-domain.region.es.amazonaws.com` に移動し、デフォルトの JSON レスポンスを受け取ることを確認します。

Python クライアントのサンプリング

Python クライアントは、HTTP リクエストよりも自動化が容易で、再利用性が向上します。Python クライアントで AI コネクタを作成するには、以下のサンプルコードを Python ファイルに保存します。クライアントには AWS SDK for Python (Boto3)、`requests`、`requests-aws4auth` のパッケージが必要です。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository
path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
headers = {"Content-Type": "application/json"}
```

```
r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

サードパーティープラットフォーム用の Amazon OpenSearch Service ML コネクタ

このチュートリアルでは、 OpenSearch Service to Cohere からコネクタを作成する方法について説明します。コネクタの詳細については、「サポートされているコネクタ」を参照してください。

外部リモートモデルで Amazon OpenSearch Service 機械学習 (ML) コネクタを使用する場合は、特定の認証認証情報を に保存する必要があります AWS Secrets Manager。これは API キーでも、ユーザー名とパスワードの組み合わせでもかまいません。つまり、Secrets Manager からの読み取りを OpenSearch サービスアクセスに許可する IAM ロールも作成する必要があります。

トピック

前提条件

Cohere または OpenSearch Service を使用する外部プロバイダーのコネクタを作成するには、認証情報 AWS Secrets Managerを保存する への OpenSearch サービスアクセスを許可する IAM ロールが必要です。また、認証情報は Secrets Manager に保存する必要があります。

IAM ロールを作成する

IAM ロールを設定して、Secrets Manager のアクセス許可を OpenSearch サービスに委任します。既存の `SecretManagerReadWrite` ロールも使用できます。新しいロールを作成するには、IAM ユーザーガイドの「IAM ロール (コンソール) の作成」を参照してください。 AWS マネージドロールを使用する代わりに新しいロールを作成する場合は、このチュートリアル`opensearch-secretmanager-role`の を自分のロールの名前に置き換えます。

1. 次の マネージド IAM ポリシーを新しいロールにアタッチして、 OpenSearch サービスが Secrets Manager の値にアクセスできるようにします。ロールにポリシーをアタッチするには、「IAM ID アクセス許可の追加」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

2. ロールの信頼関係を編集するには、「ロールの信頼ポリシーの変更」の手順に従ってください。Principal ステートメントで OpenSearch サービスを指定する必要があります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "opensearchservice.amazonaws.com"
                ]
            }
        }
    ]
}
```

aws:SourceAccount および aws:SourceArn 条件キーを使用してアクセスを特定のドメインに制限することをお勧めします。SourceAccount はドメインの所有者に属する AWS アカウント ID で、 SourceArnはドメインの ARN です。例えば、次の条件ブロックを信頼ポリシーに追加できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
```

```
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

のアクセス許可を設定します。

コネクタを作成するには、IAM ロールを OpenSearch サービスに渡すアクセス許可が必要です。さらに、es:ESHttpPost アクションへのアクセスも必要です。これらの両方の許可を付与するには、リクエストの署名に認証情報が使用されている IAM ロールまたはユーザーに次のポリシーをアタッチします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

ユーザーまたはロールがロールを渡すための iam:PassRole アクセス許可を持っていない場合、次のステップでリポジトリを登録しようとすると、認証エラーが発生することがあります。

セットアップ AWS Secrets Manager

認証の認証情報を Secrets Manager に保存するには、AWS Secrets Manager ユーザーガイドの「AWS Secrets Manager シークレットの作成」を参照してください。

Secrets Manager がキーと値のペアをシークレットとして受け入れると、次の形式の ARN を受け取ります: arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3。次のステップでコネクタを作成するときに、この ARN はキーと使用するので記録しておいてください。

OpenSearch Dashboards で ML ロールをマッピングする (きめ細かなアクセスコントロールを使用している場合)

きめ細かいアクセス制御では、コネクタの設定時に追加の手順が必要になります。HTTP 基本認証を他のすべての目的で使用する場合でも、`opensearch-sagemaker-role`を渡すための `iam:PassRole` 許可を持っている IAM ロールまたはユーザーに `ml_full_access` ロールをマップする必要があります。

1. OpenSearch サービスドメインの OpenSearch Dashboards プラグインに移動します。Dashboards エンドポイントは、 OpenSearch サービスコンソールのドメインダッシュボードにあります。
2. メインメニューから [セキュリティ]、[ロール] を選択し、[ml_full_access] ロールを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。
4. [バックエンドロール] で、`opensearch-sagemaker-role` を渡すためのアクセス許可があるロールの ARN を追加します。

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

5. [マップ] を選択し、ユーザーまたはロールが [マッピングされたユーザー] の下に表示されていることを確認します。

OpenSearch サービスコネクタを作成する

コネクタを作成するには、 OpenSearch サービスドメインエンドポイントに`POST`リクエストを送信します。署名付きリクエストを送信するには、curl、サンプル Python クライアント、Postman、またはその他の方法を使用できます。Kibana コンソールでは、`POST` リクエストは使用できないことにご注意ください。リクエストは以下のような形式です。

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
```

```
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}
```

このリクエストのリクエスト本文は、オープンソースのコネクタリクエストのリクエスト本文と 2 つの点で異なります。credential フィールド内では、Secrets Manager からの読み取りを OpenSearch サービスに許可する IAM ロールの ARN と、シークレットの ARN を渡します。headers フィールドでは、シークレットキーとそれが ARN からのものであるという事実を用いてシークレットを参照します。

ドメインが仮想プライベートクラウド (VPC) に存在する場合は、リクエストが AI コネクタを正常に作成するには、コンピュータが VPC に接続されていることが必要です。VPC へのアクセスはネットワーク構成によって異なりますが、通常は VPN あるいは社内ネットワークへの接続が必要になります。 OpenSearch サービスドメインにアクセスできることを確認するには、ウェブブラウザで https://*your-vpc-domain*.*region*.es.amazonaws.com に移動し、デフォルトの JSON レスポンスを受け取ることを確認します。

Python クライアントのサンプリング

Python クライアントは、HTTP リクエストよりも自動化が容易で、再利用性が向上します。Python クライアントで AI コネクタを作成するには、以下のサンプルコードを Python ファイルに保存します。クライアントには AWS SDK for Python (Boto3)、requests、requests-aws4auth のパッケージが必要です。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
```

```
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}

headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

AWS CloudFormation を使用してセマンティック検索用のリモート推論を設定する

OpenSearch バージョン 2.9 以降では、セマンティック検索でリモート推論を使用して、独自の機械学習 (ML) モデルをホストできます。リモート推論では、ML Commons プラグインを使用

して、モデル推論を Amazon SageMaker や Amazon などの ML サービスでリモートでホストし BedRock、ML コネクタを使用して Amazon OpenSearch Service に接続できます。

リモート推論のセットアップを容易にするために、Amazon OpenSearch Service はコンソールに AWS CloudFormation テンプレートを提供します。 CloudFormation は、インフラストラクチャをコードとして扱うことで、 AWS およびサードパーティーのリソースをモデル化、プロビジョニング、管理 AWS のサービス できる です。

OpenSearch CloudFormation テンプレートはモデルプロビジョニングプロセスを自動化するため、OpenSearch サービスドメインにモデルを簡単に作成し、モデル ID を使用してデータを取り込んでニューラル検索クエリを実行できます。

OpenSearch サービスバージョン 2.12 以降でニューラルスパースエンコーダーを使用する場合は、リモートでデプロイするのではなく、トークナイザモデルをローカルで使用することをお勧めします。詳細については、 OpenSearch ドキュメントの「スパースエンコーディングモデル」を参照してください。

トピック

- 前提条件
- Amazon SageMaker テンプレート
- Amazon Bedrock テンプレート

前提条件

OpenSearch サービスで CloudFormation テンプレートを使用するには、次の前提条件を満たします。

OpenSearch サービスドメインをセットアップする

CloudFormation テンプレートを使用する前に、バージョン 2.9 以降できめ細かなアクセスコントロールが有効になっている Amazon OpenSearch Service ドメインを設定する必要があります。OpenSearch サービスバックエンドロールを作成して、コネクタを作成するアクセス許可を ML Commons プラグインに付与します。

CloudFormation テンプレートは、デフォルトの名前 で Lambda IAM ロールを作成します。このロールは`LambdaInvokeOpenSearchMLCommonsRole`、別の名前を選択した場合に上書きできます。テンプレートがこの IAM ロールを作成したら、Lambda 関数に OpenSearch サービスドメインを呼び出すアクセス許可を付与する必要があります。これを行うには、 という名前のロールをサービスバックエンドロールに次のステップでマッピングします。 `ml_full_access` OpenSearch

1. OpenSearch サービスドメインの OpenSearch Dashboards プラグインに移動します。Dashboards エンドポイントは、 OpenSearch サービスコンソールのドメインダッシュボードにあります。
2. メインメニューから [セキュリティ]、[ロール] を選択し、[ml_full_access] ロールを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。
4. [バックエンドロール] で、ドメインを呼び出すアクセス許可を必要とする Lambda ロールの ARN を追加します。

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

5. [マップ] を選択し、ユーザーまたはロールが [マッピングされたユーザー] の下に表示されていることを確認します。

ロールをマッピングしたら、ドメインのセキュリティ設定に移動し、Lambda IAM ロールを OpenSearch サービスアクセスポリシーに追加します。

AWS アカウントでアクセス許可を有効にする

には、 CloudFormation と Lambda へのアクセス許可と、ランタイムまたは Amazon SageMaker のいずれかのテンプレートで AWS のサービス 選択したアクセス許可 AWS アカウント が必要です BedRock。

Amazon Bedrock を使用している場合、モデルも登録する必要があります。モデルを登録するには、「Amazon Bedrock ユーザーガイド」の「モデルアクセス」を参照してください。

独自の Amazon S3 バケットを使用してモデルアーティファクトを提供する場合は、CloudFormation IAM ロールを S3 アクセスポリシーに追加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「IAM ID アクセス許可の追加および削除」を参照してください。

Amazon SageMaker テンプレート

Amazon SageMaker CloudFormation テンプレートは、ニューラルプラグインとセマンティック検索をセットアップするために複数の AWS リソースを定義します。

まず、Amazon テンプレートを使用したテキスト埋め込みモデルとの統合 SageMakerを使用して、SageMaker ランタイムにテキスト埋め込みモデルをサーバーとしてデプロイします。モデルエンドポイントを指定しない場合、 SageMaker はランタイムが Amazon S3 からモデルアーティファクト

をダウンロードしてサーバーにデプロイできるようにする IAM ロール CloudFormation を作成します。エンドポイントを指定すると、 は Lambda 関数が OpenSearch サービスドメインにアクセスできるようにする IAM ロール CloudFormation を作成します。ロールが既に存在する場合は、ロールを更新して再利用します。エンドポイントは、ML Commons プラグインを使用して ML コネクタに使用されるリモートモデルを提供します。

次に、Amazon Sagemaker テンプレートを使用したスパースエンコーダーとの統合を使用して、ドメインがリモート推論コネクタをセットアップした Lambda 関数を作成します。 OpenSearch サービスでコネクタが作成されると、リモート推論は SageMaker ランタイムのリモートモデルを使用してセマンティック検索を実行できます。テンプレートはドメイン内のモデル ID を に返して、検索を開始できるようにします。

Amazon SageMaker CloudFormation テンプレートを使用するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインから、[統合] を選択します。
3. 各 Amazon SageMaker テンプレートで、ドメイン の設定、パブリックドメイン の設定 を選択します。
4. CloudFormation コンソールのプロンプトに従ってスタックをプロビジョニングし、モデルを設定します。

> **Note**
>
> OpenSearch また、VPC ドメインを設定するための別のテンプレートも用意されています。このテンプレートを使用する場合は、Lambda 関数の VPC ID を指定する必要があります。

Amazon Bedrock テンプレート

Amazon SageMaker CloudFormation テンプレートと同様に、Amazon Bedrock CloudFormation テンプレートは、 OpenSearch Service と Amazon Bedrock の間にコネクタを作成するために必要な AWS リソースをプロビジョニングします。

まず、テンプレートは、将来の Lambda 関数が OpenSearch サービスドメインにアクセスできるようにする IAM ロールを作成します。次に、テンプレートは Lambda 関数を作成します。Lambda 関数には、ML Commons プラグインを使用してコネクタがドメインによって作成されます。

OpenSearch サービスがコネクタを作成すると、リモート推論の設定が完了し、Amazon Bedrock API オペレーションを使用してセマンティック検索を実行できます。

Amazon Bedrock は独自の ML モデルをホストするため、モデルを SageMaker ランタイムにデプロイする必要はありません。代わりに、テンプレートは Amazon Bedrock の事前定義されたエンドポイントを使用し、エンドポイントのプロビジョニングステップをスキップします。

Amazon Bedrock CloudFormation テンプレートを使用するには

1. https://console.aws.amazon.com/aos/home で Amazon OpenSearch Service コンソールを開きます。
2. 左側のナビゲーションペインから、[統合] を選択します。
3. Amazon Bedrock を使用して Amazon Titan Text Embeddings モデルと統合 で、ドメイン の設定、パブリックドメイン の設定 を選択します。
4. プロンプトの指示に従ってモデルをセットアップします。

> **Note**
>
> OpenSearch また、VPC ドメインを設定するための別のテンプレートも用意されています。このテンプレートを使用する場合は、Lambda 関数の VPC ID を指定する必要があります。

さらに、 OpenSearch Service には、Cohere モデルと Amazon Titan マルチモーダル埋め込みモデルに接続するための以下の Amazon Bedrock テンプレートが用意されています。

- `Integration with Cohere Embed through Amazon Bedrock`
- `Integrate with Amazon Bedrock Titan Multi-modal`

サポートされていない ML Commons 設定

Amazon OpenSearch Service では、次の ML Commons 設定の使用をサポートしていません。

- `plugins.ml_commons.allow_registering_model_via_url`
- `plugins.ml_commons.allow_registering_model_via_local_file`

ML Commons 設定の詳細については、「ML Commons クラスター設定」を参照してください。

OpenSearch サービスフローフレームワークテンプレート

Amazon OpenSearch Service フローフレームワークテンプレートを使用すると、一般的なユースケース用のテンプレートを提供することで、複雑な OpenSearch サービスのセットアップと前処理タスクを自動化できます。例えば、フローフレームワークテンプレートを使用して、機械学習のセットアップタスクを自動化できます。Amazon OpenSearch Service フローフレームワークテンプレートは、JSON または YAML ドキュメントでセットアッププロセスに関する簡単な説明を提供します。これらのテンプレートは、会話型チャットまたはクエリ生成、AI コネクタ、ツール、エージェント、および生成モデル用のバックエンド使用のために OpenSearch サービスを準備するその他のコンポーネントの自動ワークフロー設定について説明します。

Amazon OpenSearch Service フローフレームワークテンプレートは、特定のニーズに合わせてカスタマイズできます。カスタムフローフレームワークテンプレートの例については、「flow-framework」を参照してください。 OpenSearch サービスが提供するテンプレートについては、「 workflow-templates」を参照してください。詳細な手順、API リファレンス、使用可能なすべての設定のリファレンスを含む包括的なドキュメントについては、オープンソース OpenSearch ドキュメントの「設定の自動化」を参照してください。

Service での ML コネクタの作成 OpenSearch

Amazon OpenSearch Service フローフレームワークテンプレートを使用すると、ml-commons で提供されている Create Connector API を利用して ML コネクタを設定およびインストールできます。ML コネクタを使用して、 OpenSearch サービスを他の AWS サービスまたはサードパーティープラットフォームに接続できます。詳細については、「サードパーティーの ML プラットフォーム用のコネクタの作成」を参照してください。Amazon OpenSearch Service フローフレームワーク API を使用すると、 OpenSearch サービスのセットアップと前処理タスクを自動化し、ML コネクタの作成に使用できます。

OpenSearch Service でコネクタを作成する前に、次の操作を行う必要があります。

- Amazon SageMaker ドメインを作成します。
- IAM ロールを作成します。
- パスロールのアクセス許可を設定します。
- OpenSearch Dashboards でフローフレームワークロールと ml-commons ロールをマッピングします。

AWS サービスの ML コネクタを設定する方法の詳細については、「 のサービス用の Amazon OpenSearch Service ML コネクタ AWS」を参照してください。サードパーティープラットフォームでの OpenSearch Service ML コネクタの使用の詳細については、「サードパーティープラットフォーム用の Amazon OpenSearch Service ML コネクタ」を参照してください。

フローフレームワークサービスを使用したコネクタの作成

コネクタを使用してフローフレームワークテンプレートを作成するには、 OpenSearch サービスドメインエンドポイントにPOSTリクエストを送信する必要があります。cURL、サンプル Python クライアント、Postman、または別の方法を使用して、署名付きリクエストを送信できます。POST リクエストの形式は次のとおりです。

```
POST /_plugins/_flow_framework/workflow
{
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to BedRock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
                  "content-type": "application/json"
                },
                "method": "POST",
```

```
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
              "content_type": "application/json",
              "auth": "Sig_V4",
              "max_tokens_to_sample": "8000",
              "service_name": "bedrock",
              "temperature": "0.0001",
              "response_filter": "$.completion",
              "region": "us-west-2",
              "anthropic_version": "bedrock-2023-05-31"
            }
          }
        }
      ]
    }
  }
}
```

ドメインが Virtual Private Cloud (Amazon VPC) 内にある場合、AI コネクタを正常に作成するには、リクエストが Amazon VPC に接続されている必要があります。Amazon VPC へのアクセスはネットワーク設定によって異なりますが、通常は VPN または企業ネットワークへの接続が含まれます。OpenSearch サービスドメインにアクセスできることを確認するには、ウェブブラウザ`https://your-vpc-domain.region.es.amazonaws.com`で に移動し、デフォルトの JSON レスポンスを受信していることを確認します。

Python クライアントのサンプリング

Python クライアントはHTTPリクエストよりも自動化が簡単で、再利用性が向上します。Python クライアントで AI コネクタを作成するには、以下のサンプルコードを Python ファイルに保存し

ます。クライアントには、AWS SDK for Python (Boto3)、Requests:HTTP for Humans、および requests-aws4auth 1.2.3 パッケージが必要です。

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

path = '_plugins/_flow_framework/workflow'
url = host + path

payload = {
  "name": "Deploy Claude Model",
  "description": "Deploy a model using a connector to Claude",
  "use_case": "PROVISION",
  "version": {
    "template": "1.0.0",
    "compatibility": [
      "2.12.0",
      "3.0.0"
    ]
  },
  "workflows": {
    "provision": {
      "nodes": [
        {
          "id": "create_claude_connector",
          "type": "create_connector",
          "user_inputs": {
            "name": "Claude Instant Runtime Connector",
            "version": "1",
            "protocol": "aws_sigv4",
            "description": "The connector to BedRock service for Claude model",
            "actions": [
              {
                "headers": {
                  "x-amz-content-sha256": "required",
```

```
                  "content-type": "application/json"
                },
                "method": "POST",
                "request_body": "{ \"prompt\":\"${parameters.prompt}\",
 \"max_tokens_to_sample\":${parameters.max_tokens_to_sample},
 \"temperature\":${parameters.temperature},  \"anthropic_version\":
\"${parameters.anthropic_version}\" }",
                "action_type": "predict",
                "url": "https://bedrock-runtime.us-west-2.amazonaws.com/model/
anthropic.claude-instant-v1/invoke"
              }
            ],
            "credential": {
                "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-
role"
             },
            "parameters": {
              "endpoint": "bedrock-runtime.us-west-2.amazonaws.com",
              "content_type": "application/json",
              "auth": "Sig_V4",
              "max_tokens_to_sample": "8000",
              "service_name": "bedrock",
              "temperature": "0.0001",
              "response_filter": "$.completion",
              "region": "us-west-2",
              "anthropic_version": "bedrock-2023-05-31"
            }
          }
        }
      ]
    }
  }
}

headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

事前定義されたワークフローテンプレート

Amazon OpenSearch Service には、一般的な機械学習 (ML) のユースケース用に複数のワークフローテンプレートが用意されています。テンプレートを使用すると、複雑なセットアップが簡単になり、セマンティック検索や会話検索などのユースケースに多くのデフォルト値が提供されます。ワークフローの作成 API を呼び出すときに、ワークフローテンプレートを指定できます。

- OpenSearch サービスが提供するワークフローテンプレートを使用するには、use_caseクエリパラメータとしてテンプレートのユースケースを指定します。
- カスタムワークフローテンプレートを使用するには、リクエスト本文に完全なテンプレートを指定します。カスタムテンプレートの例については、JSON テンプレートの例または YAML テンプレートの例を参照してください。

テンプレートのユースケース

この表は、使用可能なさまざまなテンプレートの概要、テンプレートの説明、および必要なパラメータを示しています。

テンプレートのユースケース	説明	必須パラメータ
bedrock_titan_embedding_model_deploy	Amazon Bedrock 埋め込みモデル (デフォルトでは titan-embed-text-v1	create_connector.credential.roleArn
bedrock_titan_embedding_model_deploy	Amazon Bedrock マルチモーダル埋め込みモデル (デフォルトでは titan-embed-text-v1	create_connector.credential.roleArn
cohere_embedding_model_deploy	Cohere 埋め込みモデル (デフォルトでは embed-english-v3.0) を作成してデプロイします。	create_connector.credential.roleArn , create_connector.credential.secretArn

テンプレートのユースケース	説明	必須パラメータ
cohere_chat_model_deploy	Cohere チャットモデル (デフォルトでは Cohere コマンド) を作成してデプロイします。	create_connector.credential.roleArn , create_connector.credential.secretArn
open_ai_embedding_model_deploy	OpenAI 埋め込みモデル (デフォルトでは text-embedding-ada-002) を作成してデプロイします。	create_connector.credential.roleArn , create_connector.credential.secretArn
openai_chat_model_deploy	OpenAI チャットモデル (デフォルトでは gpt-3.5-turbo) を作成してデプロイします。	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding	セマンティック検索を設定し、Cohere 埋め込みモデルをデプロイします。Cohere モデルの API キーを指定する必要があります。	create_connector.credential.roleArn , create_connector.credential.secretArn
semantic_search_with_cohere_embedding_query_enricher	セマンティック検索を設定し、Cohere 埋め込みモデルをデプロイします。ニューラルクエリのデフォルトのモデル ID を設定する query_enricher 検索プロセッサを追加します。Cohere モデルの API キーを指定する必要があります。	create_connector.credential.roleArn , create_connector.credential.secretArn

テンプレートのユースケース	説明	必須パラメータ
`multimodal_search_with_bedrock_titan`	Amazon Bedrock マルチモーダルモデルをデプロイし、text_image_embedding プロセッサとマルチモーダル検索用の k-NN インデックスを使用して取り込みパイプラインを設定します。 AWS 認証情報を指定する必要があります。	`create_connector.credential.roleArn`

> **Note**
>
> シークレット ARN を必要とするすべてのテンプレートの場合、デフォルトでは「key」のキー名でシークレットを AWS Secrets マネージャーに保存します。

事前トレーニング済みモデルを含むデフォルトテンプレート

Amazon OpenSearch Service には、オープンソース OpenSearch サービスでは利用できない 2 つの追加のデフォルトワークフローテンプレートが用意されています。

テンプレートのユースケース	説明
`semantic_search_with_local_model`	セマンティック検索を設定し、事前トレーニング済みのモデル () をデプロイしますmsmarco-distilbert-base-tas-b 。ニューラルクエリのデフォルトのモデル ID を設定し、「」というリンクされた k-NN インデックスを作成するneural_query_enricher 検索プロセッサを追加しますmy-nlp-index。
`hybrid_search_with_local_model`	ハイブリッド検索を設定し、事前トレーニング済みモデル () をデプロイしますmsmarco-distilbert-base-tas-b 。ニューラルクエリのデフォルトのモデル ID を設定し、「」

テンプレートのユースケース	説明
	というリンクされた k-NN インデックスを作成する[neural_query_enricher](#) 検索プロセッサを追加しますmy-nlp-index。

のアクセス許可を設定します。

バージョン 2.13 以降で新しいドメインを作成する場合、アクセス許可はすでに設定されています。バージョン 2.11 以前の既存の OpenSearch サービスドメインでフローフレームワークを有効にしてからバージョン 2.13 以降にアップグレードする場合は、`flow_framework_manager`ロールを定義する必要があります。きめ細かなアクセスコントロールを使用してドメインのウォームインデックスを管理するには、管理者以外のユーザーがこのロールにマッピングされている必要があります。`flow_framework_manager` ロールを手動で作成するには、以下のステップを実行します。

1. OpenSearch ダッシュボードで、セキュリティに移動し、アクセス許可 を選択します。
2. [アクショングループの作成] を選択し、以下のグループを設定します。

グループ名	許可
`flow_framework_full_access`	• `cluster:admin/opensearch/flow_framework/*` • `cluster_monitor`
`flow_framework_read_accesss`	• `cluster:admin/opensearch/flow_framework/workflow/get` • `cluster:admin/opensearch/flow_framework/workflow/search` • `cluster:admin/opensearch/flow_framework/workflow_state/get` • `cluster:admin/opensearch/flow_framework/workflow_state/search`

3. [ロール]、[ロールの作成] の順に選択します。
4. ロール flow_framework_manager に名前を付けます。

5. [クラスターの許可] では、flow_framework_full_access および flow_framework_read_access を選択します。
6. [インデックス] では、* と入力します。
7. [インデックスの許可] では、indices:admin/aliases/get、indices:admin/mappings/get、および indices_monitor を選択します。
8. [作成] を選択します。
9. ロールを作成したら、フローフレームワークインデックスを管理する任意のユーザーまたはバックエンドロールにマッピングします。

Amazon OpenSearch Service のセキュリティ分析

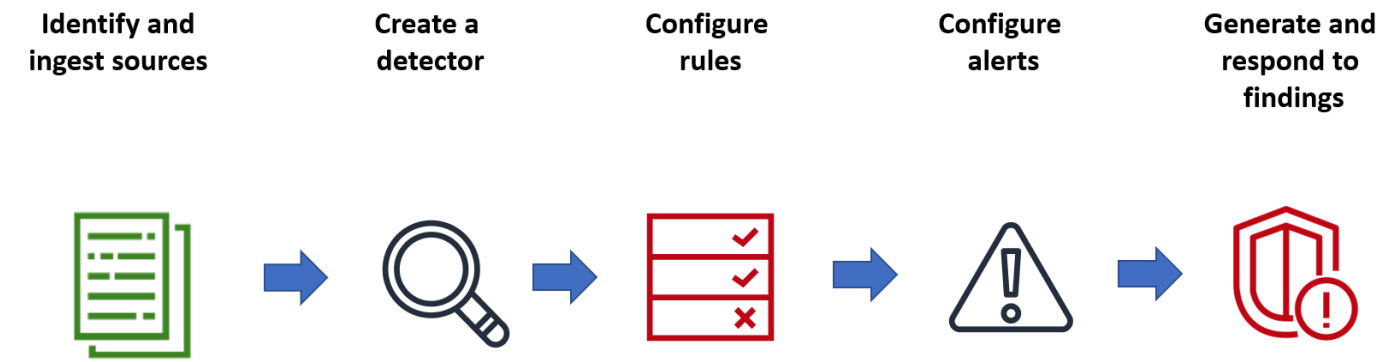
Security Analytics は、組織のインフラストラクチャを可視化し、異常なアクティビティをモニタリングし、潜在的なセキュリティ脅威をリアルタイムで検出し、事前設定された送信先にアラートをトリガーする OpenSearch ソリューションです。セキュリティルールを継続的に評価し、自動生成されたセキュリティ検出結果を確認することにより、セキュリティイベントのログの、悪意のあるアクティビティをモニタリングすることができます。さらにセキュリティ分析では、自動アラートを作成し、これを Slack やメールなどの指定された通知チャネルに送信することができます。

Security Analytics プラグインを使用すると、一般的な脅威を検出 out-of-the-box し、ファイアウォールログ、Windows ログ、認証監査ログなどの既存のセキュリティイベントログから重要なセキュリティインサイトを生成できます。Security Analytics を使用するには、ドメインが OpenSearch バージョン 2.5 以降を実行している必要があります。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service のセキュリティ分析の概要を説明します。主要な概念を定義し、アクセス許可を設定する手順を提供します。セットアップガイド、API リファレンス、使用可能なすべての設定のリファレンスを含む包括的なドキュメントについては、 OpenSearch ドキュメントの「 Security Analytics」を参照してください。

セキュリティ分析のコンポーネントと概念

数多くのツールや機能が、セキュリティ分析のオペレーションの基盤を構成しています。セキュリティ分析のプラグインを構成している主な要素には、ディテクター、ログタイプ、ルール、検出結果、アラートなどがあります。



ログタイプ

OpenSearch は、複数のタイプのログをサポートし、各タイプの out-of-the-box マッピングを提供します。ディテクターを作成するときにログタイプを指定して時間間隔を設定すると、そこから、セキュリティ分析は関連するルールセットを自動的にアクティブ化し、指定された間隔で実行します。

ディテクター

ディテクターは、ログタイプのさまざまなサイバーセキュリティ脅威を、データインデックス全体で特定します。ユーザーは、カスタムルールと、システムで発生するイベントを評価するパッケージ済みの Sigma ルールの、両方を使用するようにディテクターを設定します。すると、ディテクターがこれらのイベントからセキュリティ検出結果を生成します。ディテクターの詳細については、OpenSearch ドキュメントの「ディテクターの作成」を参照してください。

ルール

脅威検出ルールは、ディテクターが、取り込まれたログデータに適用してセキュリティイベントを特定する際の条件を定義します。セキュリティ分析では、ユーザーの要件に合うように、ルールをインポート、作成、カスタマイズすることができます。また、ログから一般的な脅威を検出するための、パッケージ済みの、オープンソースの Sigma ルールも用意されています。セキュリティ分析は、MITRE ATT&CK 組織が管理している、敵対相手の戦術やテクニックに関する、増加を続けるナレッジベースに、数多くのルールをマッピングします。 OpenSearch Dashboards または APIs の両方を使用して、ルールを作成および使用できます。ルールの詳細については、 OpenSearch ドキュメントの「ルールの使用」を参照してください。

結果

ディテクターによって、ルールとログイベントが一致すると、検出結果が生成されます。各検出結果には、選択したルール、ログタイプ、ルールの重要度から成る、一意の組み合わせが含まれています。検出結果に示される脅威は、必ずしも、今すぐ対処しなければならないものではありませんが、常に重要なイベントが取り出されています。検出結果の詳細については、 OpenSearch ドキュメントの「検出結果の使用」を参照してください。

アラート

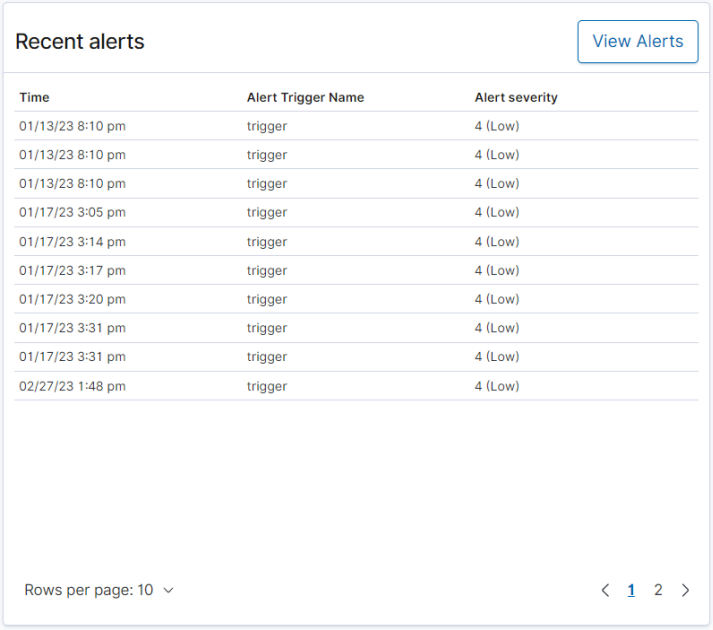
ディテクターを作成するときは、アラートをトリガーする条件を 1 つ以上指定することができます。アラートは、Slack やメールなど、優先チャンネルに送信される通知です。ディテクターが 1 つまたは複数のルールに一致したときにアラートがトリガーされるように設定します。通知メッセージ

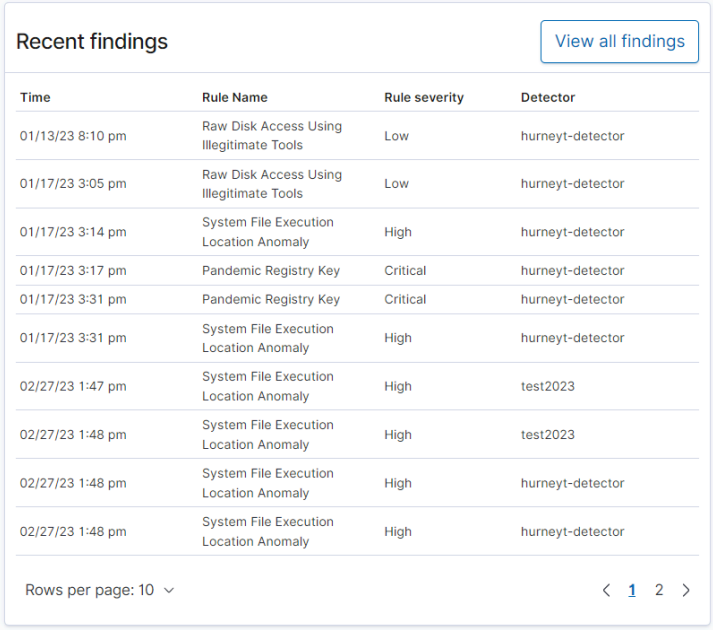
はカスタマイズできます。アラートの詳細については、 OpenSearch ドキュメントの「アラートの使用」を参照してください。

セキュリティ分析を理解する

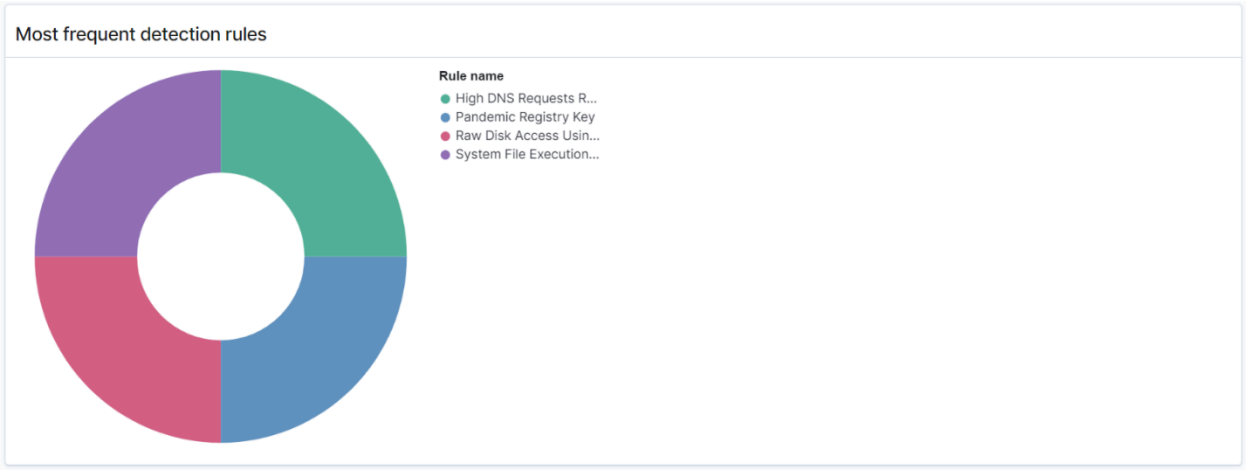
OpenSearch Dashboards を使用して、Security Analytics プラグインを可視化し、インサイトを得ることができます。概要ビューには、検出結果とアラート数、最近の検出結果とアラート、頻繁な検出ルール、ディテクターのリストなどの情報が表示されます。複数の画面を、それらをまとめた 1 か所の画面で確認できます。例えば、次のグラフには、特定の期間におけるさまざまなログタイプの検出結果とアラートの傾向が示されています。

ページの下へ進むと、最新の検出結果とアラートを確認できます。





さらに、最も多くトリガーされたルールが、アクティブなディテクター全体に広がっていることを確認できます。ログタイプ全体でさまざまな種類の悪意あるアクティビティを検出し、調査するのに役立ちます。



最後に、設定済みのディテクターのステータスを確認できます。このパネルからは、ディテクターのワークフローの作成に移動することもできます。

Security Analytics の設定を行うには、[Rules] (ルール) ページでルールを作成し、このルールを使って [Detectors] (ディテクター) ページにディテクターを記述します。Security Analytics の検出結果に特化した画面を表示するには、[Findings] (検出結果) と [Alerts] (アラート) のページを使用します。

のアクセス許可を設定します。

既存の OpenSearch サービスドメインで Security Analytics を有効にすると、そのドメインで`security_analytics_manager`ロールが定義されていない可能性があります。きめ細かなアクセスコントロールを使用してドメインのウォームインデックスを管理するには、管理者以外のユーザーがこのロールにマッピングされている必要があります。`security_analytics_manager` ロールを手動で作成するには、以下のステップを実行します。

1. OpenSearch ダッシュボードで、セキュリティに移動し、アクセス許可 を選択します。
2. [アクショングループの作成] を選択し、以下のグループを設定します。

グループ名	許可
`security_analytics_full_access`	• `cluster:admin/opensearch/securityanalytics/alerts/*` • `cluster:admin/opensearch/securityanalytics/detector/*` • `cluster:admin/opensearch/securityanalytics/findings/*` • `cluster:admin/opensearch/securityanalytics/mapping/*` • `cluster:admin/opensearch/securityanalytics/rule/*`

グループ名	許可
security_analytics_read_access	• cluster:admin/opensearch/securityanalytics/alerts/get • cluster:admin/opensearch/securityanalytics/detector/get • cluster:admin/opensearch/securityanalytics/detector/search • cluster:admin/opensearch/securityanalytics/findings/get • cluster:admin/opensearch/securityanalytics/mapping/get • cluster:admin/opensearch/securityanalytics/mapping/view/get • cluster:admin/opensearch/securityanalytics/rule/get • cluster:admin/opensearch/securityanalytics/rule/search

3. [ロール]、[ロールの作成] の順に選択します。
4. ロールに security_analytics_manager という名前を付けます。
5. [クラスターの許可] では、security_analytics_full_access および security_analytics_read_access を選択します。
6. [インデックス] では、* と入力します。
7. [Index permissions] (インデックスのアクセス許可) で、indices:admin/mapping/put と indices:admin/mappings/get を選択します。
8. [作成] を選択します。
9. ロールを作成したら、任意のユーザーまたは Security Analytics インデックスを管理するバックエンドロールに、それをマッピングします。

トラブルシューティング

そのようなインデックスエラーはありません

ディテクターがない状態で Security Analytics ダッシュボードを開くと、右下に `[index_not_found_exception] no such index [.opensearch-sap-detectors-config]` と書かれた通知が表示されることがあります。この通知は無視してかまいません。ディテクターを作成すると数秒で消え、再び表示されることはありません。

Amazon OpenSearch Service のオブザーバビリティ

OpenSearch Dashboards for Amazon OpenSearch Service のデフォルトインストールにはオブザーバビリティプラグインが含まれています。オブザーバビリティプラグインを使用すると、パイプ処理言語 (PPL) を使用してデータ駆動型イベントを視覚化し、 に保存されているデータを探索、検出、クエリできます OpenSearch。プラグインには OpenSearch 1.2 以降が必要です。

可観測性プラグインは、一般的なデータソースからメトリクス、ログ、トレースを収集およびモニタリングするための統合エクスペリエンスを提供します。データ収集とモニタリングを 1 か所で行うことで、インフラストラクチャ全体のフルスタックのオブザー end-to-end バビリティを実現できます。

> **Note**
>
> このドキュメントでは、 OpenSearch サービスにおけるオブザーバビリティの概要を説明します。アクセス許可を含むオブザーバビリティプラグインの包括的なドキュメントについては、「オブザーバビリティ」を参照してください。

データを探索するプロセスはユーザーごとに異なります。データを調べて視覚化を作成するのが初めての場合は、次のようなワークフローを試すことをお勧めします。

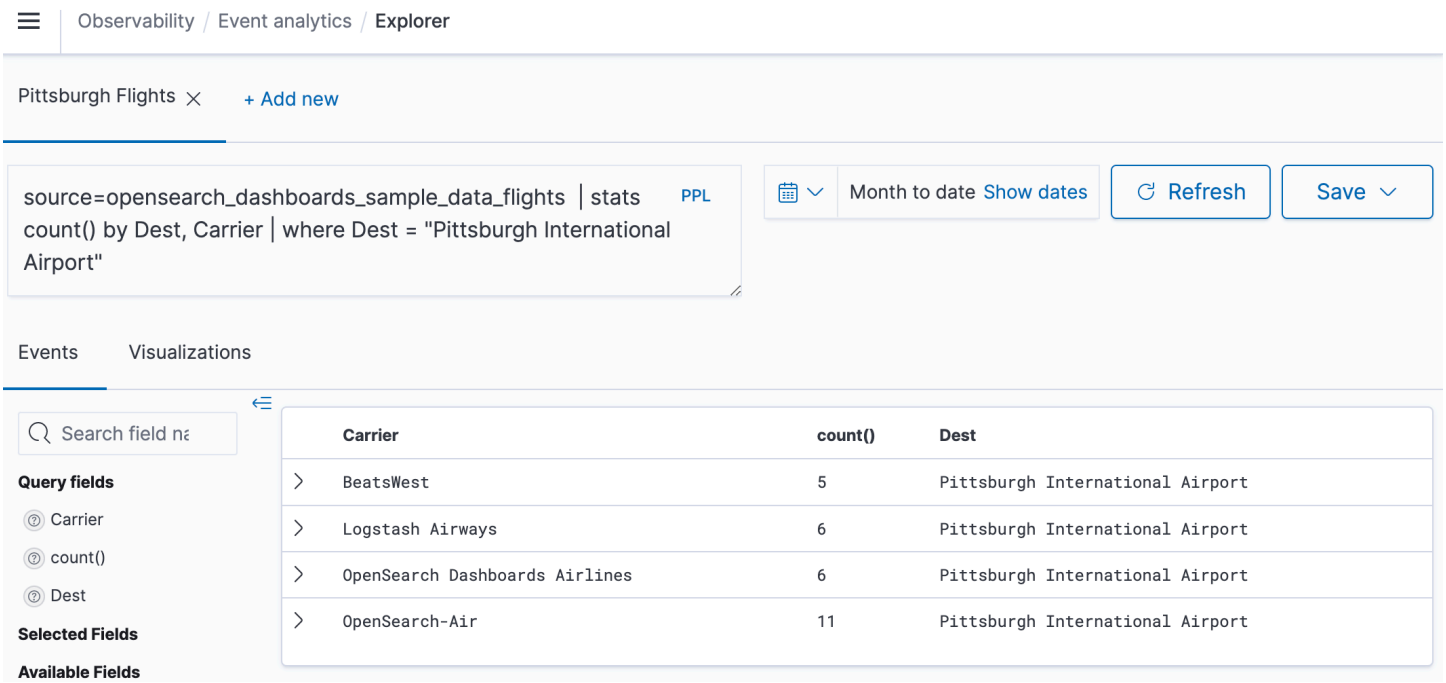
イベント分析でのデータの探索

まず、 OpenSearch サービスドメインでフライトデータを収集していて、先月ピッツバーグ国際空港に到着したフライトが最も多い航空会社を調べたいとします。次の PPL クエリを記述します。

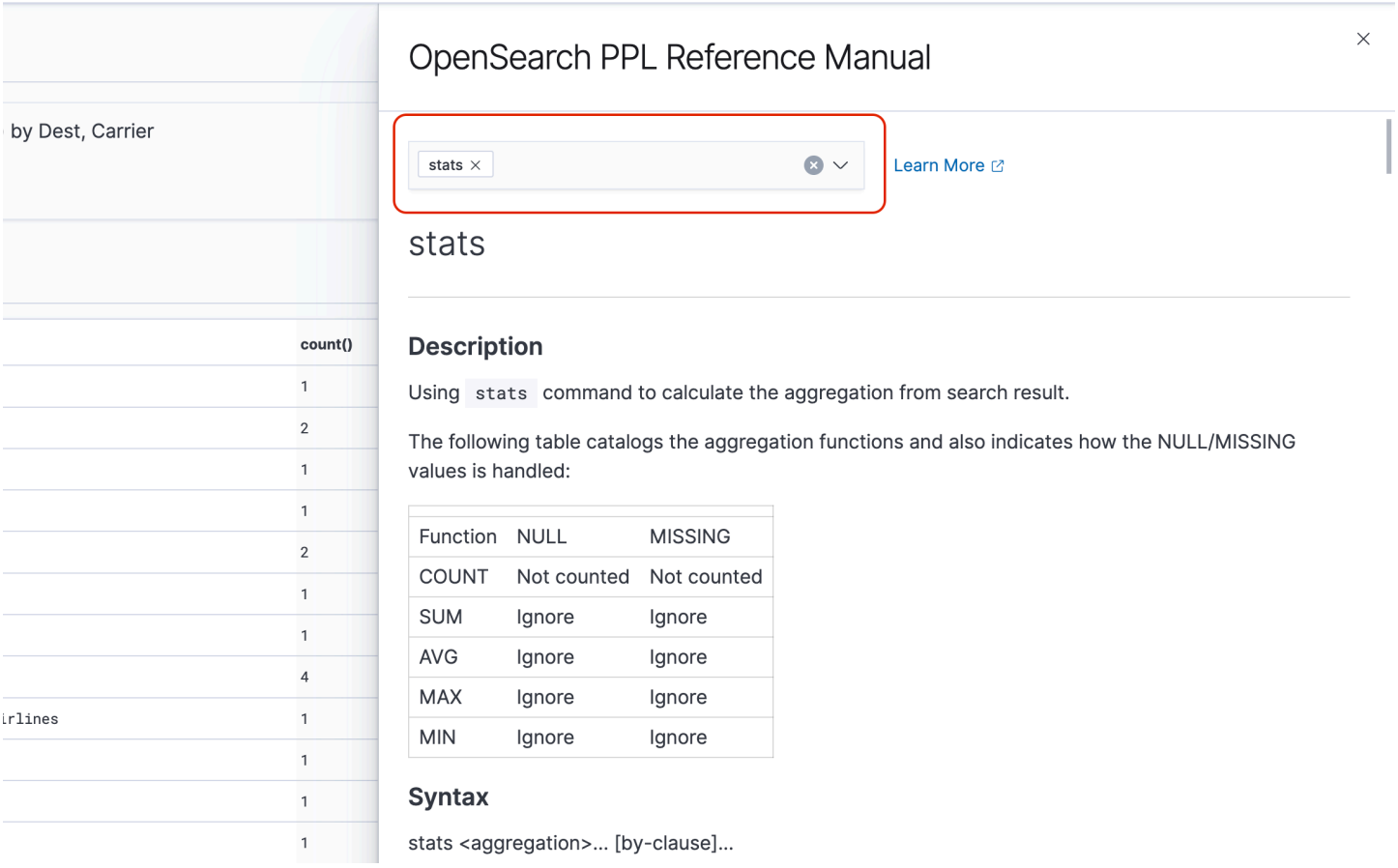
```
source=opensearch_dashboards_sample_data_flights |
    stats count() by Dest, Carrier |
    where Dest = "Pittsburgh International Airport"
```

このクエリは、opensearch_dashboards_sample_data_flights という名前のインデックスからデータを取り出します。次に、クエリは stats コマンドを使用して便の総数を取得し、目的地の空港と航空会社に従ってグループ化します。最後に、where 句を使用して、ピッツバーグ国際空港に到着する便の結果がフィルターされます。

先月のデータは次のようになります。



クエリエディタの PPL ボタンを選択すると、各 PPL コマンドの使用方法と例が表示されます。



便の遅延に関する情報をクエリする複雑な例を見てみましょう。

```
source=opensearch_dashboards_sample_data_flights |
    where FlightDelayMin > 0 |
    stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier,
 Dest |
    eval avg_delay=minimum_delay / total_delayed |
    sort - avg_delay
```

クエリの各コマンドは、最終出力に影響します。

- `source=opensearch_dashboards_sample_data_flights` - 前の例と同じインデックスからデータを取り出します
- `where FlightDelayMin > 0` - 遅延した便でデータをフィルターします
- `stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier` - 各航空会社について、合計最小遅延時間と遅延便の合計数を取得します
- `eval avg_delay=minimum_delay / total_delayed` - 最小遅延時間を遅延便の総数で除算し、各航空会社の平均遅延時間を計算します
- `sort - avg_delay` - 結果を平均遅延で降順でソートします

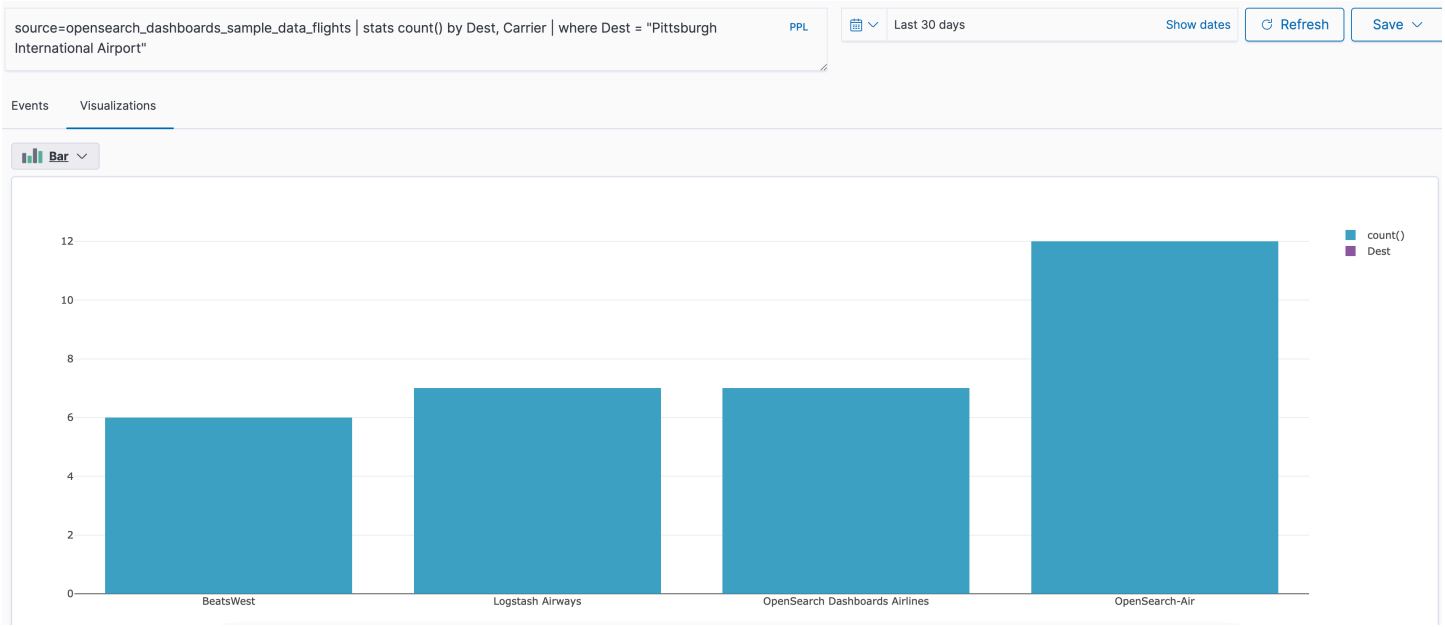
このクエリを使用すると、 OpenSearch Dashboards" の遅延が平均して少ないことを確認できます。

	avg_delay	Carrier	minimum_delay	total_delayed
>	212	Logstash Airways	4470	21
>	184	OpenSearch-Air	4245	23
>	155	BeatsWest	2025	13
>	153	OpenSearch Dashboards Airlines	4305	28

その他のサンプル PPL クエリについては、イベント分析ページの「Queries and Visualizations」(クエリと可視化) を参照してください。

可視化の作成

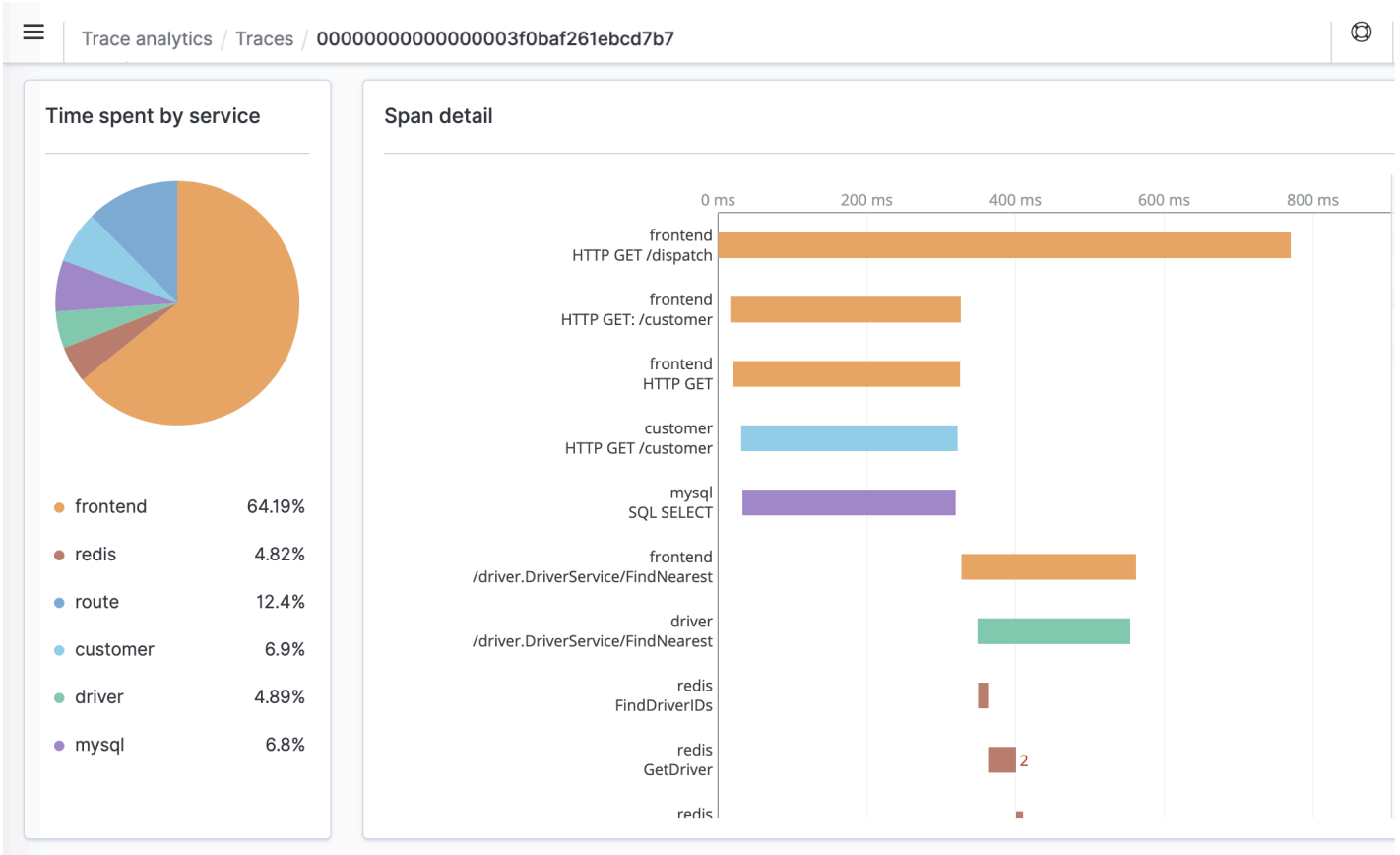
関心のあるデータを正しくクエリしたら、これらのクエリを可視化として保存できます。



次に、これらの可視化をオペレーションパネルに追加して、さまざまなデータを比較します。ノートブックを活用して、チームメンバーと共有できるさまざまな可視化とコードブロックを組み合わせます。

トレース分析による詳細な分析

トレース分析は、 OpenSearch データ内のイベントフローを視覚化して、分散アプリケーションのパフォーマンス問題を特定して修正する方法を提供します。



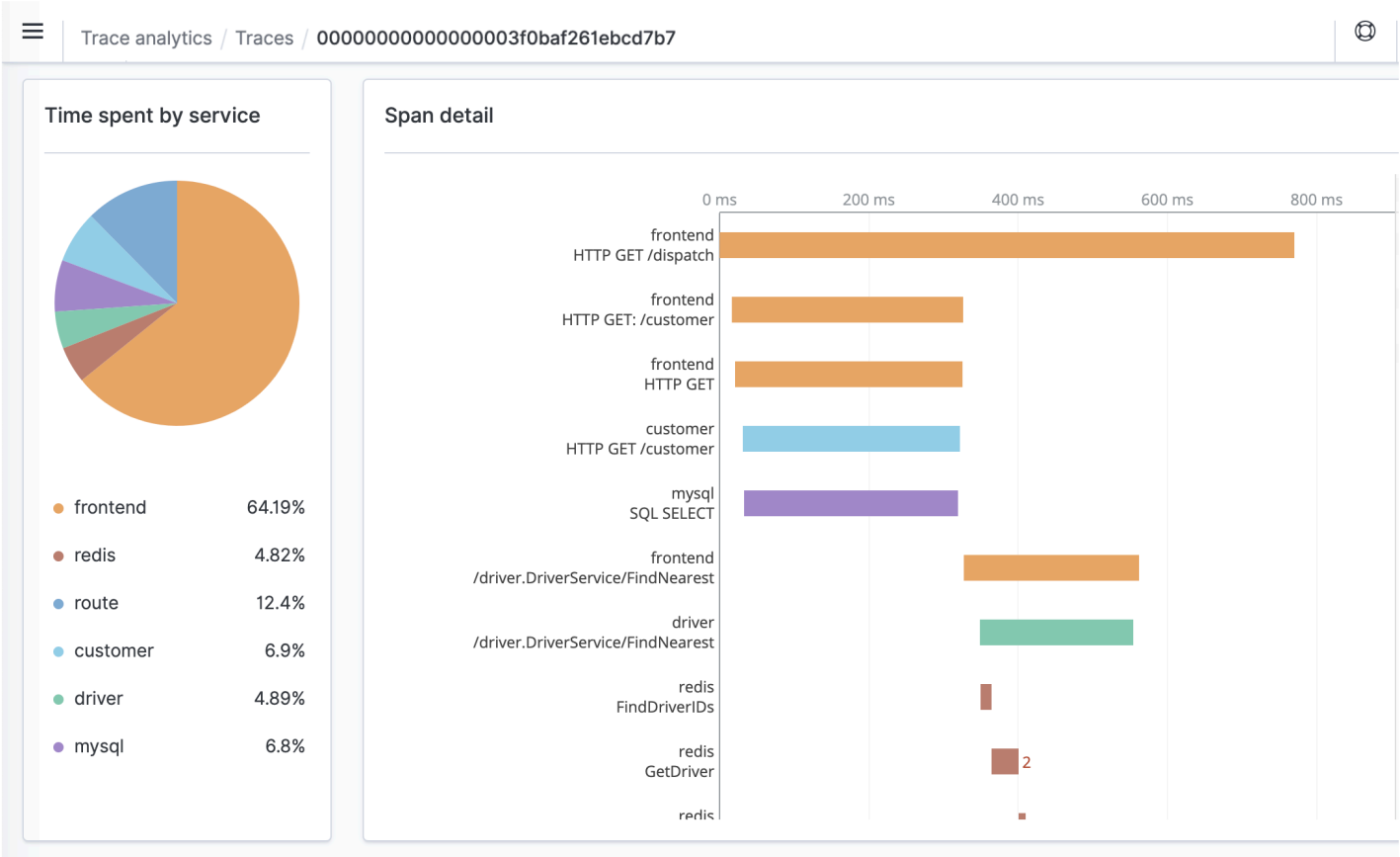
Amazon OpenSearch Service のトレース分析

OpenSearch オブザーバビリティプラグインの一部であるトレース分析を使用して、分散アプリケーションからのトレースデータを分析できます。トレース分析には、 OpenSearch または Elasticsearch 7.9 以降が必要です。

分散アプリケーションでは、ユーザーがボタンをクリックするなどの単一のオペレーションで、一連の拡張イベントをトリガーできます。例えば、アプリケーションのフロントエンドがバックエンドサービスを呼び出し、バックエンドサービスはデータベースのクエリを実行し、クエリを処理して結果を返します。次に、最初のバックエンドサービスがフロントエンドに確認を送信し、フロントエンドは UI を更新します。

トレース分析を使用すると、このイベントのフローを可視化し、パフォーマンスの問題を特定できます。

> **Note**
>
> このドキュメントでは、トレース分析の概要を説明します。包括的なドキュメントについては、オープンソース OpenSearch ドキュメントの「トレース分析」を参照してください。



前提条件

トレース分析では、Jaeger や Zipkin OpenTelemetryなどの がサポートするライブラリを使用して、アプリケーションに計測を追加し、トレースデータを生成する必要があります。このステップは、完全に OpenSearch Service の外部で行われます。AWS Distro for OpenTelemetry ドキュメントには、Java、Python、Go、 など、開始に役立つ多くのプログラミング言語のサンプルアプリケーションが含まれています JavaScript。

アプリケーションに計測を追加すると、OpenTelemetryコレクターはアプリケーションからデータを受信し、 OpenTelemetry データにフォーマットします。「」のレシーバーのリストを参照してくださいGitHub。 の AWS ディストリビューションには、 のレシーバー AWS X-Ray OpenTelemetry が含まれています。

最後に、 Amazon OpenSearch Ingestionを使用して、 で使用する OpenTelemetry データをフォーマットできます OpenSearch。

OpenTelemetry コレクターのサンプル設定

で OpenTelemetry コレクターを使用するにはAmazon OpenSearch Ingestion、次のサンプル設定を試してください。

```
extensions:
  sigv4auth:
    region: "us-east-1"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/
opentelemetry.proto.collector.trace.v1.TraceService/Export"
    auth:
      authenticator: sigv4auth
    compression: none

service:
  extensions: [sigv4auth]
  pipelines:
    traces:
      receivers: [jaeger]
      exporters: [otlphttp]
```

OpenSearch 取り込みサンプル設定

トレースデータを OpenSearch サービスドメインに送信するには、次のサンプル OpenSearch 取り込みパイプライン設定を試してください。パイプラインを作成する手順については、「」を参照してくださいthe section called “パイプラインの作成”。

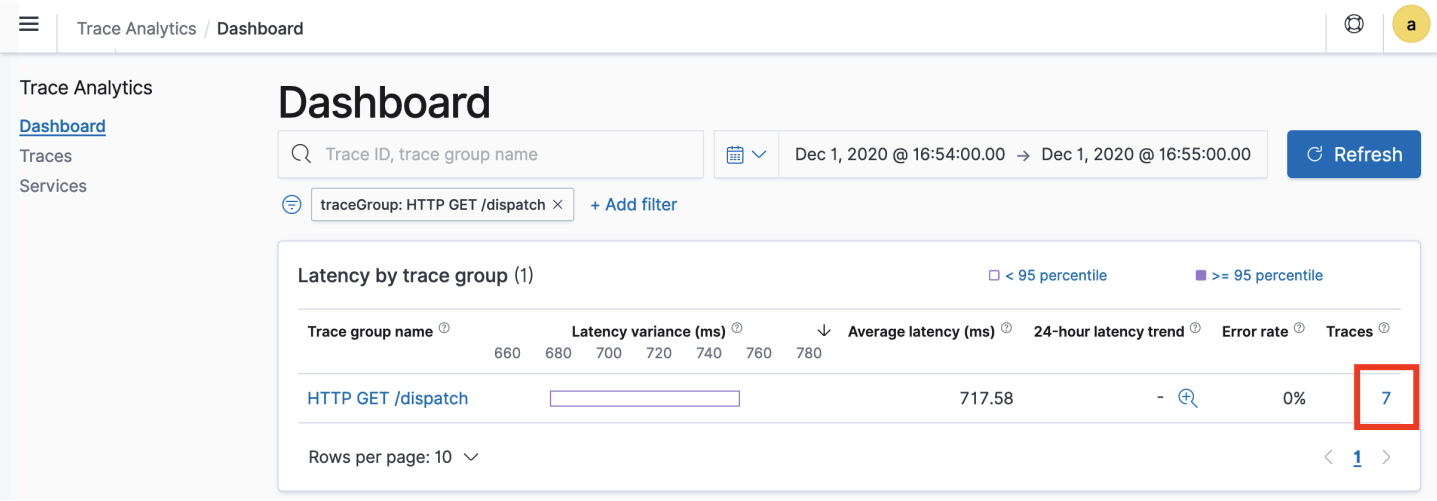
```
version: "2"
otel-trace-pipeline:
```

```
  source:
    otel_trace_source:
      "/${pipelineName}/ingest"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace_pipeline"
    - pipeline:
        name: "service_map_pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"

service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-service-map
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

sts_role_arn オプションで指定するパイプラインロールには、シンクへの書き込みアクセス許可が必要です。パイプラインロールのアクセス許可を設定する手順については、「」を参照してくださいthe section called “ロールとユーザーの設定”。

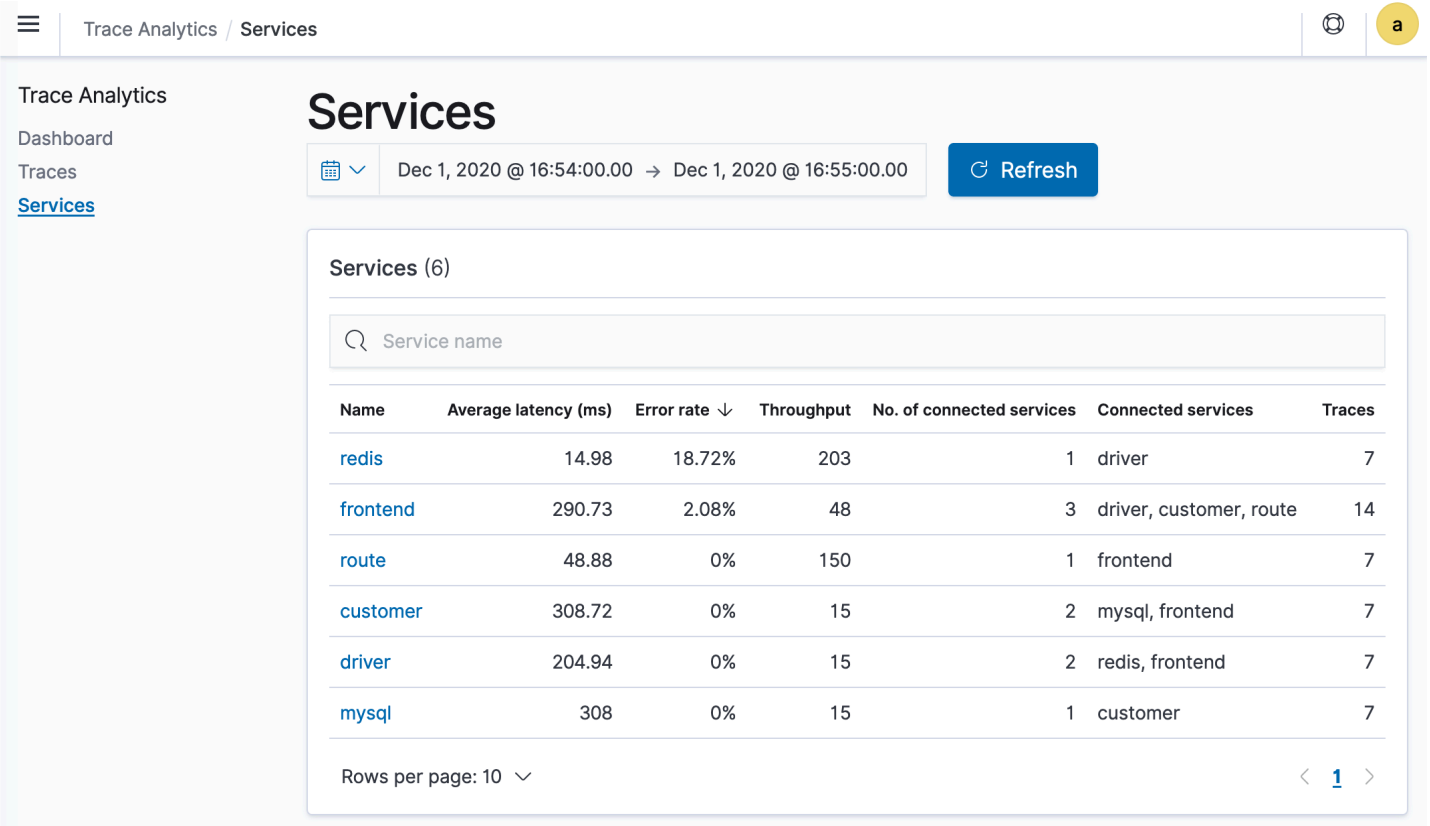
トレースデータの探索

Dashboards ビューでは、HTTP メソッドとパス別にトレースをグループ化して、特定のオペレーションに関連付けられた平均レイテンシー、エラー率、傾向を確認できるようにします。より焦点を絞ったビューについては、トレースグループ名でフィルタリングを試みてください。



トレースグループを構成するトレースをドリルダウンするには、右側の列でトレースの数を選択します。次に、詳細な概要については、個々のトレースを選択します。

Services ビューでは、アプリケーション内のすべてのサービスと、さまざまなサービスが相互に接続する方法を示す対話型マップを一覧表示します。ダッシュボード (オペレーションごとに問題を特定するのに役立つ) とは対照的に、サービスマップはサービスごとに問題を特定するのに役立ちます。エラー率またはレイテンシーでソートを試みて、アプリケーションの潜在的な問題領域を把握してください。



パイプ処理言語を使用した Amazon OpenSearch Service データのクエリ

Piped Processing Language (PPL) は、パイプ (|) 構文を使用して Amazon OpenSearch Service に保存されているデータをクエリできるクエリ言語です。PPL には、 OpenSearch または Elasticsearch 7.9 以降が必要です。

> **Note**
>
> このドキュメントでは、Amazon OpenSearch Service の PPL の概要を説明します。詳細な手順と完全なコマンドリファレンスについては、オープンソース OpenSearch ドキュメントの「PPL」を参照してください。

PPL 構文は、パイプ文字 (|) で区切られているコマンドで構成されており、データは各パイプラインを左から右に流れます。例えば、HTTP 403 または 503 エラーのあるホストの数を調べ、ホストごとに集計し、影響順に並べ替える PPL 構文は、次のとおりです。

```
source = dashboards_sample_data_logs | where response='403' or response='503' | stats
 count(request) as request_count by host, response | sort -request_count
```

開始するには、 OpenSearch Dashboards で Query Workbench を選択し、PPL を選択します。bulk オペレーションを使用して、いくつかのサンプルデータのインデックスを作成します。

```
PUT accounts/_bulk?refresh
{"index":{"_id":"1"}}
{"account_number":1,"balance":39225,"firstname":"Amber","lastname":"Duke","age":32,"gender":"M"
 Holmes
 Lane","employer":"Pyrami","email":"amberduke@pyrami.com","city":"Brogan","state":"IL"}
{"index":{"_id":"6"}}
{"account_number":6,"balance":5686,"firstname":"Hattie","lastname":"Bond","age":36,"gender":"M"
 Bristol
 Street","employer":"Netagy","email":"hattiebond@netagy.com","city":"Dante","state":"TN"}
{"index":{"_id":"13"}}
{"account_number":13,"balance":32838,"firstname":"Nanette","lastname":"Bates","age":28,"gender"
 Mady Street","employer":"Quility","city":"Nogal","state":"VA"}
{"index":{"_id":"18"}}
{"account_number":18,"balance":4180,"firstname":"Dale","lastname":"Adams","age":33,"gender":"M"
 Hutchinson Court","email":"daleadams@boink.com","city":"Orick","state":"MD"}
```

次の例は、18 より大きい age を持つアカウントインデックス内のドキュメントの firstname および lastname フィールドを返します。

```
search source=accounts | where age > 18 | fields firstname, lastname
```

レスポンス例

id	firstname	lastname
0	Amber	Duke
1	Hattie	Bond
2	Nanette	Bates
3	Dale	Adams

search、where、fields、rename、dedup、stats、sort、eval、head、top、および rare のような読み取り専用コマンドの完全なセットを使用することができます。PPL プラグインは、数学、三角法、日付時刻、文字列、集計、アドバンスト演算子および式を含むすべての SQL 関数をサポートします。詳細については、OpenSearch PPL リファレンスマニュアル を参照してください。

Amazon OpenSearch Service の運用上のベストプラクティス

この章では、Amazon OpenSearch Service ドメインを運用するためのベストプラクティスと、多くのユースケースに適用される一般的なガイドラインについて説明します。各ワークロードは一意で、固有の特性があるため、すべてのユースケースに完全に適した一般的な推奨事項はありません。最も重要なベストプラクティスは、ドメインを連続的なサイクルでデプロイ、テスト、調整して、ワークロードに最適な設定、安定性、およびコストを見つけることです。

トピック

モニタリングとアラート

OpenSearch サービスドメインのモニタリングには、次のベストプラクティスが適用されます。

CloudWatch アラームを設定する

OpenSearch サービスは、パフォーマンスメトリクスを Amazon に発行します CloudWatch。クラスターとインスタンスのメトリクスを定期的に確認し、ワークロードのパフォーマンスに基づいて推奨 CloudWatch アラームを設定します。

ログの発行を有効にする

OpenSearch サービスは、Amazon CloudWatch Logs の OpenSearch エラーログ、スローログの検索、スローログのインデックス作成、監査ログを公開します。検索スローログ、インデックス作成スローログ、およびエラーログは、パフォーマンスと安定性の問題のトラブルシューティングに役立ちます。きめ細かいアクセスコントロールを有効にした場合にのみ使用できる監査ログは、ユーザーアクティビティを追跡します。詳細については、 OpenSearch ドキュメントの「ログ」を参照してください。

検索スローログとインデックス作成スローログは、検索およびインデックス作成オペレーションのパフォーマンスを理解し、トラブルシューティングするための重要なツールです。すべての本番ドメインのために、検索およびインデックスの低速ログ配信を有効にします。また、ログ記録のしきい値を設定する必要があります。そうしないと、ログはキャプチャ CloudWatch されません。

シャード戦略

シャードは、 OpenSearch サービスドメインのデータノード全体にワークロードを分散します。インデックスを適切に設定すると、ドメイン全体のパフォーマンスを向上させるのに役立ちます。

OpenSearch Service にデータを送信するときは、そのデータをインデックスに送信します。インデックスはデータベーステーブルに似ています。ドキュメントが行、フィールドが列に相当します。インデックスを作成するときに、作成する OpenSearch プライマリシャードの数を指定します。プライマリシャードは、データセット全体の独立したパーティションです。 OpenSearch サービスは、インデックス内のプライマリシャード全体にデータを自動的に分散します。インデックスのレプリカを設定することもできます。各レプリカシャードは、そのインデックスのプライマリシャードのコピーの完全なセットで構成されます。

OpenSearch サービスは、クラスター内のデータノード全体で各インデックスのシャードをマッピングします。これにより、インデックスのプライマリシャードとレプリカシャードが別々のデータノードに確実に存在するようになります。最初のレプリカは、インデックスにデータのコピーが確実に 2 個あるようにします。常に少なくとも 1 個のレプリカを使用する必要があります。追加のレプリカは、さらなる冗長性と読み取りキャパシティーを提供します。

OpenSearch は、インデックスに属するシャードを含むすべてのデータノードにインデックス作成リクエストを送信します。インデックス作成リクエストは、まずプライマリシャードを含むデータノードに送信され、その後にレプリカシャードを含むデータノードに送信されます。検索リクエストは、コーディネーターノードによって、インデックスに属するすべてのシャードのプライマリシャードまたはレプリカシャードにルーティングされます。

例えば、5 個のプライマリシャードと 1 個のレプリカがあるインデックスの場合、各インデックス作成リクエストは 10 個のシャードにタッチします。一方、検索リクエストは n 個のシャードに送信されます。n はプライマリシャードの数です。5 個のプライマリシャードと 1 個のレプリカがあるインデックスの場合、各検索クエリは、そのインデックスの 5 個のシャード (プライマリまたはレプリカ) にタッチします。

シャードとデータノード数を決定する

次のベストプラクティスを使用して、ドメインのシャードとデータノード数を決定します。

シャードのサイズ — ディスク上のデータのサイズは、ソースデータのサイズの直接的な結果であり、インデックスを作成するデータが増えるにつれて変化します。 source-to-index 比率は 1:10 から 10:1 以上まで大きく異なる場合がありますが、通常は 1:1.10 前後です。この比率を使用して、ディスク上のインデックスサイズを予測できます。また、一部のデータにインデックスを付け、実際のインデックスサイズを取得して、ワークロードの比率を判断することもできます。インデックスのサイズを予測したら、各シャードが 10～30 GiB (検索ワークロードの場合) または 30～50 GiB (ログワークロードの場合) になるようにシャード数を設定します。50 GiB が最大になるはずです。増量に備えて準備しておいてください。

シャード数 — データノードへのシャードの分散は、ドメインのパフォーマンスに大きな影響を与えます。複数のシャードを持つインデックスがある場合は、シャードがデータノード数の偶数倍になるように試みます。これは、シャードがデータノード全体に均等に分散されるようにするのに役立ち、ノードがホットになるのを防ぎます。例えば、プライマリシャードが 12 個ある場合、データノード数は 2、3、4、6、または 12 になります。ただし、シャード数よりもシャードサイズの方が重要です。5 GiB のデータがある場合、1 個のシャードを使用するべきです。

データノードあたりのシャード — ノードが保持できるシャードの総数は、ノードの Java 仮想マシン (JVM) ヒープメモリに比例します。ヒープメモリの GiB あたりのシャード数が 25 個以下になるように試みます。例えば、32 GiB のヒープメモリを持つノードのシャード数は 800 個以下である必要があります。シャードの分散は、ワークロードのパターンに基づいて異なる場合がありますが、シャード数はノードあたり 1,000 個という制限があります。cat/allocation API は、シャードの数とデータノード全体のシャードストレージの合計のクイックビューを提供します。

シャードと CPU の比率 — シャードがインデックス作成または検索リクエストに関わっている場合、vCPU を使用してリクエストを処理します。ベストプラクティスとして、シャードあたり 1.5 vCPU の初期スケールポイントを使用します。インスタンスタイプに 8 個の vCPUs がある場合は、各ノードのシャード数が 6 個以下になるようにデータノード数を設定します。これは近似値であることに注意してください。必ずワークロードをテストし、それに応じてクラスターをスケールしてください。

ストレージボリューム、シャードサイズ、インスタンスタイプに関する推奨事項については、次のリソースを参照してください。

- the section called “ドメインのサイジング”
- the section called “ペタバイトスケール”

ストレージスキューを回避する

ストレージスキューは、クラスター内の 1 つ以上のノードが、1 つ以上のインデックスについて、他のノードよりも高い割合のストレージを保持している場合に発生します。不均等な CPU 使用率、断続的で不均等なレイテンシー、データノード全体での不均等なキューイングなどはストレージスキューを示します。スキューの問題があるかどうかを判断するには、次のトラブルシューティングセクションを参照してください。

- the section called “ノードシャードとストレージスキュー”
- the section called “インデックスシャードとストレージスキュー”

安定性

以下のベストプラクティスは、安定した正常な OpenSearch サービスドメインの維持に適用されます。

で最新の状態に保つ OpenSearch

サービスソフトウェア更新

OpenSearch サービスは、機能を追加したり、ドメインを改善したりするソフトウェア更新を定期的にリリースします。更新によって OpenSearch または Elasticsearch エンジンのバージョンは変更されません。DescribeDomain API オペレーションを実行する定期的な時間をスケジュールし、 `UpdateStatus` が の場合はサービスソフトウェアの更新を開始することをお勧めしますELIGIBLE。特定の期間 (通常は 2 週間) 内にドメインを更新しない場合、 OpenSearch Service は自動的に更新を実行します。

OpenSearch バージョンアップグレード

OpenSearch サービスでは、コミュニティが管理するバージョンの のサポートが定期的に追加されています OpenSearch。利用可能な場合は、常に OpenSearch 最新バージョンにアップグレードしてください。

OpenSearch サービスは、 OpenSearch と OpenSearch Dashboards (ドメインがレガシーエンジンを実行している場合は Elasticsearch と Kibana) の両方を同時にアップグレードします。クラスターに専用マスターノードがある場合、アップグレードはダウンタイムなしで完了します。そうしないと、クラスターがマスターノードを選択している間、アップグレード後数秒間応答しない可能性があります。アップグレードの一部またはすべて中に OpenSearch ダッシュボードが使用できない場合があります。

ドメインをアップグレードするには 2 つの方法があります。

- インプレースアップグレード — 同じクラスターを維持するため、このオプションの方が簡単です。
- スナップショット/リストアアップグレード — このオプションは、新しいクラスターで新しいバージョンをテストしたり、クラスター間で移行したりするのに適しています。

使用するアップグレードプロセスにかかわらず、開発とテスト専用のドメインを維持し、本番ドメインをアップグレードする前に新しいバージョンにアップグレードすることをお勧めします。テストドメインを作成するときに、デプロイタイプのために [Development and testing] (開発とテスト) を選択します。ドメインのアップグレードの直後に、必ずすべてのクライアントを互換性のあるバージョンにアップグレードしてください。

スナップショットパフォーマンスの向上

スナップショットの処理が停止するのを防ぐには、専用マスターノードのインスタンスタイプがシャードカウントに一致する必要があります。詳細については、「the section called “専用マスターノードのインスタンスタイプの選択”」を参照してください。また、各ノードの Java ヒープメモリの GiB あたりのシャード数は、推奨される 25 シャードよりも少なくしてください。詳細については、「the section called “シャード数の選択”」を参照してください。

専用マスターノードを有効にする

専用のマスターノードにより、クラスターの安定性が向上します。専用マスターノードではクラスター管理タスクを実行しますが、インデックスデータは保持せず、クライアントリクエストにも応答しません。このクラスター管理タスクのオフロードにより、ドメインの安定性が向上し、一部の設定変更をダウンタイムなしで行うことができます。

3 つのアベイラビリティーゾーンにまたがるドメインの安定性を最適化するために、3 個の専用マスターノードを有効にして使用します。スタンバイが有効のマルチ AZ でデプロイすると、3 つの専用

マスターノードが設定されます。インスタンスタイプの推奨事項については、「the section called "専用マスターノードのインスタンスタイプの選択"」を参照してください。

複数のアベイラビリティゾーンにデプロイする

サービス中断が発生した場合にデータの損失を防ぎ、クラスターのダウンタイムを最小限に抑えるため、同じ AWS リージョン内の 2 つまたは 3 つのアベイラビリティーゾーン間にノードを分散できます。ベストプラクティスは、スタンバイが有効のマルチ AZ を使用してデプロイすることです。これにより、3 つのアベイラビリティーゾーンが設定され、2 つのゾーンがアクティブ、1 つのゾーンがスタンバイとして機能し、インデックスごとに 2 つのレプリカシャードが割り当てられます。この設定により、 OpenSearch サービスは対応するプライマリシャードとは異なる AZs にレプリカシャードを分散できます。アベイラビリティーゾーン間のクラスター通信には、AZ 間のデータ転送料金はかかりません。

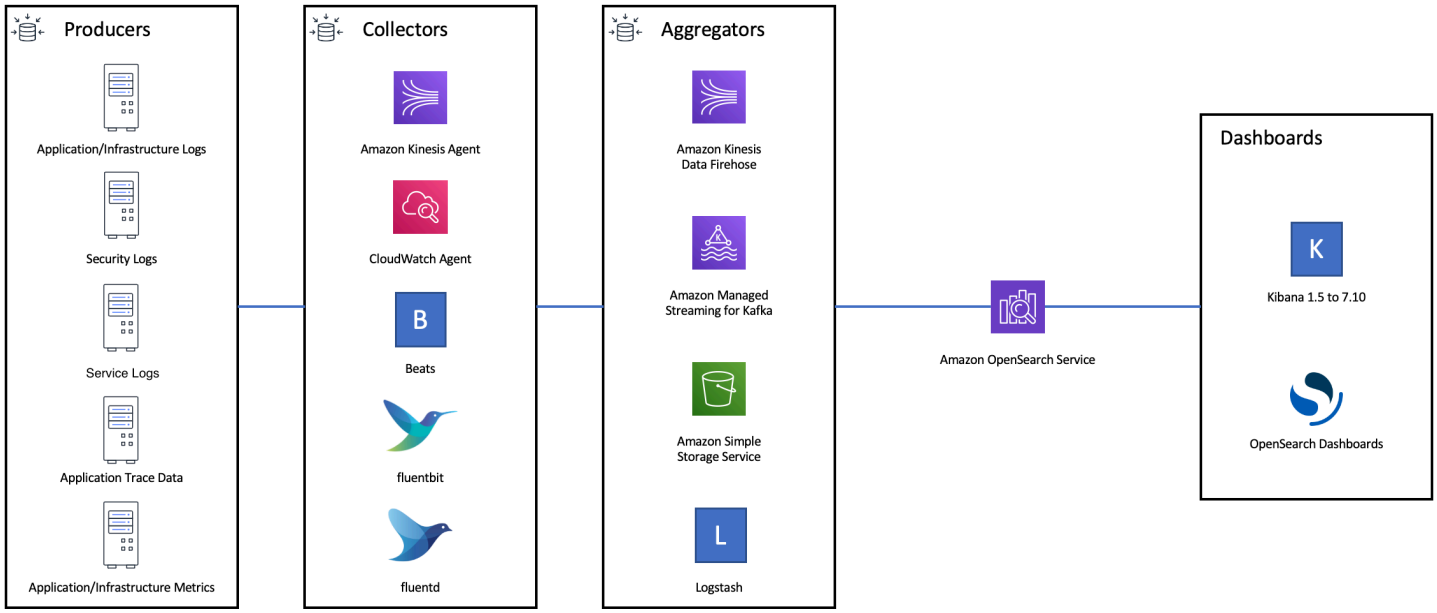
アベイラビリティーゾーンは、各 リージョン内の独立した場所です。2 つの AZ 設定では、1 つのアベイラビリティーゾーンを失うことは、すべてのドメインキャパシティーの半分を失うことを意味します。3 つのアベイラビリティゾーンに移行すると、1 つのアベイラビリティゾーンを失うことによる影響が軽減されます。

取り込みフローとバッファリングを制御する

_bulk API オペレーションを使用して、全体のリクエスト数を制限することをお勧めします。1 つのドキュメントを含む 5,000 のリクエストを送信するよりも、5,000 のドキュメントを含む 1 つの `_bulk` リクエストを送信する方が効率的です。

最適な運用安定性のために、インデックス作成リクエストのアップストリームフローを制限したり、一時停止したりする必要がある場合があります。インデックス作成リクエストのレートを制限することは、対処しなければクラスターで過負荷となる可能性のある、予期しないまたは時折のリクエストの急増に対処するための重要なメカニズムです。アップストリームアーキテクチャにフロー制御メカニズムを構築することを検討してください。

次の図は、ログ取り込みアーキテクチャの複数のコンポーネントオプションを示しています。急激なトラフィックの急増や短時間のドメインメンテナンスのために受信データをバッファリングするのに十分なスペースを確保できるように、集約レイヤーを設定します。



検索ワークロードのマッピングを作成する

検索ワークロードの場合、 がドキュメントとそのフィールド OpenSearch を保存およびインデックス化する方法を定義するマッピングを作成します。新しいフィールドが誤って追加されるのを防ぐために dynamic を strict に設定します。

```
PUT my-index
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "title": { "type" : "text" },
      "author": { "type" : "integer" },
      "year": { "type" : "text" }
    }
  }
}
```

インデックステンプレートを使用する

インデックステンプレートは、作成時にインデックスを設定する OpenSearch 方法として使用できます。インデックスを作成する前に、インデックステンプレートを設定します。その後、インデックスを作成すると、テンプレートから設定とマッピングが継承されます。1 個のインデックスに複数のテンプレートを適用できるため、1 個のテンプレートで設定を指定し、別のテンプレートでマッピン

グを指定できます。この戦略では、複数のインデックスにまたがる共通設定用の 1 個のテンプレートと、より具体的な設定やマッピング用の個別のテンプレートを使用できます。

次の設定は、テンプレートでの設定に役立ちます。

- プライマリシャードとレプリカシャードの数
- 更新間隔 (インデックスを更新して最近の変更を検索できるようにする頻度)
- 動的マッピングコントロール
- 明示的なフィールドマッピング

次のテンプレート例には、これらの各設定が含まれています。

```
{
   "index_patterns":[
      "index-*"
   ],
   "order": 0,
   "settings": {
      "index": {
         "number_of_shards": 3,
         "number_of_replicas": 1,
         "refresh_interval": "60s"
      }
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "field_name1": {
            "type": "keyword"
         }
      }
   }
}
```

ほとんど変更されない場合でも、複数のアップストリームクライアントを更新するよりも、設定とマッピングを で一元的に定義 OpenSearch するのが簡単です。

Index State Management でインデックスを管理する

ログや時系列データを管理している場合は、Index State Management (ISM) を使用することをお勧めします。ISM では、通常のインデックスライフサイクル管理タスクを自動化できます。ISM を使

用すると、インデックスエイリアスのロールオーバーを呼び出すポリシーを作成し、インデックスのスナップショットを作成し、ストレージ階層間でインデックスを移動し、古いインデックスを削除できます。シャードスキューを回避するための代替データライフサイクル管理戦略として、ISM ロールオーバーオペレーションを使用することもできます。

まず、ISM ポリシーを設定します。例については、「the section called “サンプルポリシー”」を参照してください。その後、ポリシーを 1 つ以上のインデックスにアタッチします。ポリシーに ISM テンプレートフィールドを含めると、 OpenSearch Service は指定されたパターンに一致するインデックスにポリシーを自動的に適用します。

未使用インデックスの削除

クラスター内のインデックスを定期的に確認し、使用されていないインデックスを特定します。これらのインデックスが S3 に保存されるようにスナップショットを作成してから、削除します。未使用のインデックスを削除すると、シャード数が減り、ノード全体でよりバランスの取れたストレージの分散とリソースの使用が可能になります。アイドル状態であっても、インデックスは内部的なインデックスのメンテナンス作業中に一部のリソースを消費します。

未使用のインデックスを手動で削除する代わりに、ISM を使用して自動的にスナップショットを作成し、一定期間後にインデックスを削除することができます。

高可用性を実現するために複数のドメインを使用する

複数のリージョンで 99.9% のアップタイムを超える高可用性を実現するには、2 つのドメインの使用を検討してください。小規模またはゆっくりと変化するデータセットの場合、クロスクラスターレプリケーションを設定して、アクティブ/パッシブモデルを維持できます。このモデルでは、リーダードメインのみが書き込みの対象となりますが、どちらのドメインも読み取りの対象とすることができます。データセットが大きく、データが急速に変化する場合は、すべてのデータがアクティブ-アクティブモデルの両方のドメインに独立して書き込まれるように、取り込みパイプラインで二重配信を設定します。

フェイルオーバーを念頭に置いて、アップストリームとダウンストリームのアプリケーションを設計します。フェイルオーバープロセスは、他のディザスタリカバリプロセスと一緒にテストしてください。

パフォーマンス

次のベストプラクティスは、最適なパフォーマンスを実現するために、ドメインの調整に適用されます。

一括リクエストのサイズと圧縮を最適化する

一括サイズ設定は、データ、分析、およびクラスター設定によって異なりますが、適切な開始点は一括リクエストあたり 3～5 MiB です。

gzip 圧縮を使用してリクエストとレスポンスのペイロードサイズを小さくすることで、 OpenSearch ドメインからリクエストを送信し、レスポンスを受信します。gzip 圧縮は、OpenSearch Python クライアント で使用するか、クライアント側から次のヘッダーを含めることで使用できます。

- `'Accept-Encoding': 'gzip'`
- `'Content-Encoding': 'gzip'`

一括リクエストのサイズを最適化するには、3 MiB の一括リクエストサイズから始めます。その後、インデックス作成のパフォーマンスが改善しなくなるまで、リクエストサイズを徐々に大きくします。

> **Note**
>
> Elasticsearch バージョン 6.x を実行しているドメインで gzip 圧縮を有効にするには、クラスターレベルで `http_compression.enabled` を設定する必要があります。この設定は、Elasticsearch バージョン 7.x およびすべてのバージョンの でデフォルトで適用されます OpenSearch。

一括リクエストのレスポンスのサイズを小さくする

OpenSearch レスポンスのサイズを小さくするには、 `filter_path`パラメータを使用して不要なフィールドを除外します。失敗したリクエストを識別または再試行するために必要なフィールドを除外しないようにしてください。詳細な説明と例については、「the section called “レスポンスサイズの削減”」を参照してください。

更新間隔を調整する

OpenSearch インデックスには結果の読み取り整合性があります。更新オペレーションにより、インデックスに対して実行されたすべての更新が検索可能になります。デフォルトの更新間隔は 1 秒です。つまり、 はインデックスの書き込み中に毎秒更新 OpenSearch を実行します。

インデックスの更新頻度が低い (更新間隔が長い) ほど、インデックス作成の全体的なパフォーマンスが向上します。更新間隔を長くすることのトレードオフは、インデックスが更新されてから新し

いデータが検索可能になるまでの時間が長くなることです。全体的なパフォーマンスを向上させるには、更新間隔を許容できる限り長く設定します。

すべてのインデックスの `refresh_interval` パラメータを 30 秒以上に設定することをお勧めします。

Auto-Tune を有効にする

Auto-Tune は、 OpenSearch クラスターのパフォーマンスと使用状況のメトリクスを使用して、ノードのキューサイズ、キャッシュサイズ、Java 仮想マシン (JVM) 設定の変更を提案します。これらのオプションの変更により、クラスターの速度と安定性が向上します。デフォルトの OpenSearch サービス設定はいつでも元に戻すことができます。Auto-Tune は、明示的に無効にしない限り、新しいドメインでデフォルトで有効になります。

すべてのドメインで Auto-Tune を有効にし、定期的なメンテナンスウィンドウを設定するか、定期的に推奨事項を確認することをお勧めします。

セキュリティ

ドメインのセキュリティ保護には、次のベストプラクティスが適用されます。

きめ細かなアクセスコントロールを有効にする

きめ細かなアクセスコントロールにより、 OpenSearch サービスドメイン内の特定のデータにアクセスできるユーザーを制御できます。一般化されたアクセスコントロールと比較して、きめ細かいアクセスコントロールでは、各クラスター、インデックス、ドキュメント、およびフィールドに、独自の指定アクセスポリシーが設定されます。アクセス基準は、アクセスをリクエストする人の役割や、データに対して実行しようとしているアクションなど、さまざまな要因に基づくことができます。例えば、あるユーザーにはインデックスへの書き込みアクセスを許可し、別のユーザーには変更を加えずにインデックスのデータを読み取るためだけのアクセスを許可する場合があります。

きめ細かなアクセスコントロールは、セキュリティやコンプライアンスの問題を生じさせずに、アクセス要件の異なるデータが同じストレージスペースに存在できるようにします。

ドメインできめ細かいアクセスコントロールを有効にすることをお勧めします。

VPC 内にドメインをデプロイする

仮想プライベートクラウド (VPC) 内に OpenSearch サービスドメインを配置すると、インターネットゲートウェイ、NAT デバイス、VPN 接続を必要とせずに、VPC 内の OpenSearch サービスと他

のサービス間の安全な通信が可能になります。すべてのトラフィックは AWS クラウド内で安全に保持されます。論理的な隔離により、VPC 内に存在するドメインには、パブリックエンドポイントを使用するドメインに比較して、より拡張されたセキュリティレイヤーがあります。

VPC 内にドメインを作成することをお勧めします。

制限的なアクセスポリシーを適用する

ドメインが VPC 内にデプロイされている場合でも、セキュリティをレイヤーで実装するのがベストプラクティスです。現在のアクセスポリシーの設定を確認してください。

制限的なリソースベースのアクセスポリシーをドメインに適用し、設定 API および OpenSearch API オペレーションへのアクセスを許可するときは、最小特権の原則に従います。原則として、アクセスポリシーで匿名ユーザープリンシパル `"Principal": {"AWS": "*" }` を使用することは避けてください。

ただし、きめ細かなアクセスコントロールを有効にする場合など、オープンアクセスポリシーの使用が許容される状況もあります。オープンアクセスポリシーを使用すると、特定のクライアントやツールなど、リクエストの署名が困難または不可能な場合にもドメインにアクセスできます。

保管中の暗号化を有効にする

OpenSearch サービスドメインは、データへの不正アクセスを防ぐために、保管中のデータを暗号化します。保管時の暗号化では AWS Key Management Service 、 (AWS KMS) を使用して暗号化キーを保存および管理し、256 ビットキー (AES-256) を使用した Advanced Encryption Standard アルゴリズムを使用して暗号化を実行します。

ドメインに機密データが保存されている場合、保管中のデータの暗号化を有効にします。

node-to-node 暗号化を有効にする

N ode-to-node 暗号化は、 OpenSearch サービス内のデフォルトのセキュリティ機能に加えて、セキュリティレイヤーを追加します。内でプロビジョニングされるノード間のすべての通信に Transport Layer Security (TLS) を実装します OpenSearch。N ode-to-node 暗号化。HTTPS 経由で OpenSearch サービスドメインに送信されるデータは、ノード間で分散およびレプリケートされている間、転送中に暗号化されたままになります。

ドメインに機密データが保存されている場合は、暗号化 を有効にします node-to-node。

によるモニタリング AWS Security Hub

を使用して、セキュリティのベストプラクティスに関連する OpenSearch サービスの使用状況をモニタリングしますAWS Security Hub。Security Hub は、セキュリティコントロールを使用してリソース設定とセキュリティ標準を評価し、お客様がさまざまなコンプライアンスフレームワークに準拠できるようサポートします。Security Hub を使用して OpenSearch サービスリソースを評価する方法の詳細については、「 ユーザーガイド」の「 Amazon OpenSearch Service コントロールAWS Security Hub 」を参照してください。

コスト最適化

以下のベストプラクティスは、 OpenSearch サービスコストの最適化と削減に適用されます。

最新世代のインスタンスタイプを使用する

OpenSearch サービスは常に、低コストでパフォーマンスを向上させる新しい Amazon EC2 インスタンスタイプを採用しています。常に最新世代のインスタンスを使用することをお勧めします。

本番稼働用ドメインで T2 や `t3.small` インスタンスを使用しないようにしてください。それらは負荷の高い状態では不安定になることがあるためです。`r6g.large` インスタンスは、小規模な本番ワークロードのためのオプションです (データノードと専用マスターノードの両方として)。

最新の Amazon EBS gp3 ボリュームを使用する

OpenSearch データノードは、高速なインデックス作成とクエリを提供するために、低レイテンシーと高スループットのストレージを必要とします。Amazon EBS gp3 ボリュームを使用することによって、以前提供されていた Amazon EBS gp2 ボリュームタイプよりも 9.6% 低いコストで、より優れたベースラインパフォーマンス (IOPS およびスループット) を得ることができます。gp3 を使用すると、ボリュームサイズにかかわらず、追加の IOPS とスループットをプロビジョニングできます。これらのボリュームはバーストクレジットを使用しないため、前世代のボリュームよりも安定性が優れています。gp3 ボリュームタイプは、gp2 per-data-node ボリュームタイプのボリュームサイズ制限も倍増します。これらのより大きなボリュームを使用すると、データノードあたりのストレージ容量を増やすことによって、パッシブデータのコストを削減できます。

時系列ログデータに UltraWarm および コールドストレージを使用する

ログ分析 OpenSearch に を使用している場合は、コストを削減するためにデータを UltraWarm またはコールドストレージに移動してください。Index State Management (ISM) を使用して、ストレージ階層間でデータを移行し、データ保持を管理します。

UltraWarm は、大量の読み取り専用データを OpenSearch Service. UltraWarm uses Amazon S3 に保存するためのコスト効率の高い方法を提供します。つまり、データはイミュータブルで、必要なコピーは 1 つだけです。インデックス内のプライマリシャードのサイズに相当するストレージの料金のみをお支払いいただきます。 UltraWarm クエリのレイテンシーは、クエリの処理に必要な S3 データの量に応じて増加します。ノードにデータがキャッシュされると、 UltraWarm インデックスへのクエリはホットインデックスへのクエリと同様に実行されます。

コールドストレージも S3 を使用します。コールドデータをクエリする必要がある場合は、既存の UltraWarm ノードに選択的にアタッチできます。コールドデータには と同じマネージドストレージコストが発生しますが UltraWarm、コールドストレージ内のオブジェクトは UltraWarm ノードリソースを消費しません。したがって、コールドストレージは、 UltraWarm ノードのサイズや数に影響を与えることなく、大量のストレージ容量を提供します。

UltraWarm ホットストレージから移行するデータが約 2.5 TiB ある場合、 はコスト効率が高くなります。フィルレートをモニタリングし、そのデータ量に達する前にインデックスを UltraWarm に移動することを計画します。

リザーブドインスタンスの推奨事項を確認する

パフォーマンスとコンピューティング消費量の適切なベースラインを把握したら、リザーブドインスタンス (RI) の購入を検討してください。割引は、前払いなしの 1 年間の予約で約 30% から始まり、全額前払いの 3 年間の契約では最大 50% まで増加する可能性があります。

少なくとも 14 日間安定した動作を確認したら、Cost Explorer で [Reserved Instance recommendations] (リザーブドインスタンスの推奨事項) を確認します。Amazon OpenSearch Service の見出しには、特定の RI 購入のレコメンデーションと削減額の予測が表示されます。

Amazon OpenSearch Service ドメインのサイズ設定

Amazon OpenSearch Service ドメインのサイジングには最適な方法はありません。ただし、ストレージのニーズ、サービス、および OpenSearch それ自体を理解することから始めることで、ハードウェアのニーズについて知識に基づいた初期見積もりを行うことができます。ドメインのサイジングに取りかかる際の最も重要な検討材料として、そうした予測結果を代表的なワークロードを使用したテストに活用し、パフォーマンスを監視します。

トピック

- ストレージ要件の計算
- シャード数の選択

- インスタンスタイプとテストの選択

ストレージ要件の計算

ほとんどの OpenSearch ワークロードは、次の 2 つのカテゴリのいずれかに分類されます。

- 存続期間の長いインデックス: データを 1 つ以上の OpenSearch インデックスに処理し、ソースデータの変更に応じてそれらのインデックスを定期的に更新するコードを記述します。一般的な例としては、ウェブサイト、ドキュメント、e コマースの検索などがあります。
- ローリングインデックス: データは継続的に一時的なインデックスに受け渡され、インデックス作成時間や保持期間が指定されます (例: 2 週間保持される一連の日次のインデックス)。代表的な例は、ログ分析、時系列処理、クリックストリーム分析などです。

長期存続インデックスのワークロードの場合、使用されるストレージ容量はディスク上のソースデータを調べることで簡単に判断できます。データのソースが複数ある場合は、すべてのソースを単純に合計します。

ローリングインデックスについては、一般的な期間内に生成されるデータ量に保持期間を掛けます。たとえば、1 時間あたり 200 MiB のログデータが生成されるとすると、1 日で 4.7 GiB になります。この場合、保持期間が 2 週間であれば、いずれかの時点でデータは 66 GiB に達します。

ただし、ソースデータのサイズはストレージ要件の 1 つの要素に過ぎません。次の点についても考慮する必要があります。

- レプリカの数: レプリカとはインデックス全体をコピーしたものであり、同量のディスク容量を必要とします。デフォルトでは、各 OpenSearch インデックスには 1 つのレプリカがあります。データ損失を防ぐため、少なくとも 1 つは作成することをお勧めします。レプリカによって、検索パフォーマンスを改善することもできます。読み込みが多いワークロードが発生する場合は、より多くのレプリカがあるとよいでしょう。`PUT /my-index/_settings` を使用して、インデックスの `number_of_replicas` 設定を更新します。
- OpenSearch インデックス作成オーバーヘッド : インデックスのディスク上のサイズは異なります。多くの場合、ソースデータとインデックスの合計サイズはソースの 110% で、インデックスはソースデータの 10% までです。データのインデックスを作成したら、`_cat/indices?v` API と `pri.store.size` 値を使用して正確なオーバーヘッドを計算できます。`_cat/allocation?v` も有用な要約を提供します。

- オペレーティングシステムの予約スペース: デフォルトでは、Linux は、重要なプロセス、システム復元、そしてディスクの断片化問題に対する保護目的で、root ユーザー用にファイルシステムの 5% を予約します。
- OpenSearch サービスオーバーヘッド: OpenSearch サービスは、セグメントマージ、ログ、その他の内部オペレーションのために、各インスタンスのストレージ領域の 20% (最大 20 GiB) を予約します。

  この 20 GiB の上限があるため、予約容量の合計はドメインに存在するインスタンスの数によって大きく異なります。たとえば、ドメインに 3 つの m6g.xlarge.search インスタンスがあり、それぞれのストレージ容量が 500 GiB であるとすると、合計は 1.46 TiB になります。この場合、予約される容量はわずか 60 GiB です。ドメインに 10 の m3.medium.search インスタンスがあり、それぞれのストレージ容量が 100 GiB の場合は、合計で 0.98 TiB になります。最初のドメインの方が 50% 大きいですが、後者の予約容量は合計 200 GiB です。

  次の式では、オーバーヘッドの「ワーストケース」の見積もりを適用します。この見積もりには、ノード障害やアベイラビリティーゾーンの停止による影響を最小限に抑えるのに役立つ追加の空き容量が含まれています。

つまり、任意の時点で 66 GiB のデータが存在し、レプリカを 1 つ必要とする場合、最小ストレージ要件は 66 * 2 * 1.1/0.95/0.8 = 191 GiB になります。この計算は、次のように定型化できます。

ソースデータ * (1 + レプリカの数) * (1 + インデックス作成オーバーヘッド) / (1 - Linux 予約スペース) / (1 - OpenSearch サービスオーバーヘッド) = 最小ストレージ要件

あるいは、この簡素化されたバージョンを使用することもできます。

ソースデータ * (1 + レプリカの数) * 1.45 = 必要な最小ストレージ

ストレージ容量の不足は、クラスターが不安定になる最も一般的な原因の 1 つです。したがって、インスタンスタイプ、インスタンス数、およびストレージボリュームを選択するときは、数値をクロスチェックする必要があります。

ストレージに関するその他の考慮事項は次のとおりです。

- 最小ストレージ要件が 1 PB を超える場合は、「the section called “ペタバイトスケール”」を参照してください。
- ローリングインデックスがあり、ホットウォームアーキテクチャを使用する場合は、「the section called “UltraWarm ストレージ”」を参照してください。

シャード数の選択

ストレージ要件を理解したら、インデックス作成の戦略を調査できます。デフォルトでは、OpenSearch Service では、各インデックスは 5 つのプライマリシャードと 1 つのレプリカ (合計 10 個のシャード) に分割されます。この動作は OpenSearch、デフォルトで 1 つのプライマリシャードと 1 つのレプリカシャードになるオープンソース とは異なります。既存のインデックスに対してプライマリシャードの数を簡単に変更することはできません。したがって、シャード数は最初のドキュメントでインデックスを作成する前に決定する必要があります。

シャード数を選択することの全体的な目標は、クラスター内のすべてのデータノード間でインデックスを均等に分散させることです。ただし、これらのシャードは大きすぎたり多すぎたりしてはいけません。一般的なガイドラインとして、検索レイテンシーが主要なパフォーマンス目標であるワークロードではシャードのサイズを 10～30 GiB、ログ分析などの書き込みが多いワークロードでは 30～50 GiB の範囲に維持することをお勧めします。

シャードが大きいと、障害からの OpenSearch 復旧が困難になる可能性がありますが、各シャードはある程度の CPU とメモリを使用するため、シャードが小さすぎると、パフォーマンスの問題やメモリ不足エラーが発生する可能性があります。つまり、シャードは基盤となる OpenSearch サービスインスタンスが処理できるほど小さくする必要がありますが、ハードウェアに不要な負荷をかけるほど小さくはありません。

たとえば、66 GiB のデータがあるとします。その数値は今後も増加する予定がなく、各シャードのサイズは約 30 GiB を維持する必要があります。この場合、シャード数は約 3 つとなります (66 * 1.1/30 = 3)。この計算は、次のように定型化できます。

(ソースデータ + 拡張の余地) * (1 + インデックス作成オーバーヘッド) / 必要なシャードサイズ = プライマリシャードの概数

この式は、時間の経過に伴うデータ拡張にも対応できます。同じ 66 GiB のデータが来年は 4 倍になると想定される場合、シャード数はおよそ (66 + 198) * 1.1/30 = 10 となります。ただし、追加分の 198 GiB のデータは現時点では存在しません。将来に向けてのこのような準備として不必要な小さいシャードを作成し、現時点で CPU とメモリを大量に消費することがないようにしてください。この場合、シャード当たりのサイズは 66 * 1.1/10 シャード = 7.26 GiB となります。これは余分なリソースを消費する場合があり、推奨サイズ範囲を下回ります。6 つのシャードの middle-of-the-road アプローチをより多く検討すると、現在は 12-GiB のシャード、将来は 48-GiB のシャードになります。その後、シャードが 50 GiB を超えたときには、3 つのシャードを使用してデータのインデックスを再作成することもできます。

かなりまれな問題の場合、ノードあたりのシャード数を制限する必要があります。シャードのサイズを適切に設定した場合、通常はこの制限に達するかなり前にディスク容量が不足します。たとえば、m6g.large.search インスタンスの最大ディスクサイズは 512 GiB です。ディスク使用率が 80% 未満に維持されており、シャードのサイズを 20 GiB に設定した場合、約 20 個のシャードを収容できます。Elasticsearch 7.x 以降、および のすべてのバージョンでは OpenSearch、ノードあたり 1,000 個のシャードに制限があります。ノードあたりの最大シャードを調整するには、cluster.max_shards_per_node 設定を設定してください。例については、「Cluster settings」(クラスターの設定) を参照してください。

シャードのサイズを適切に設定すると、ほとんどの場合この制限未満に維持できますが、Java ヒープの GiB ごとにシャードの数を検討することもできます。ノードごとに、Java ヒープの GiB あたりのシャード数を 25 以下にしてください。例えば、m5.large.search インスタンスに 4 GiB のヒープがあるとすると、各ノードのシャード数は 100 以下にすることになります。そのシャード数では、各シャードのサイズが約 5 GiB になり、推奨値をかなり下回ります。

インスタンスタイプとテストの選択

ストレージ要件を計算して必要なシャード数を選択したら、ハードウェアに関する決定ができるようになります。ハードウェア要件はワークロードによって大きく異なるものの、基本的な推奨事項は提供されています。

各インスタンスタイプのストレージ制限は一般的に、軽度のワークロードで必要とされる CPU とメモリの量にマッピングされています。たとえば、m6g.large.search インスタンスが最大で 512 GiB の EBS ボリュームサイズ、vCPU x 2 コア、8 GiB のメモリを備えているとします。クラスターには多数のシャードがあり、高負荷の集計処理、頻繁なドキュメント更新、または大量のクエリ処理が発生している場合、それらのリソースではニーズを満たせない可能性があります。クラスターがこのようなカテゴリに分類される場合はまず、ストレージ要件の容量 100 GiB ごとに vCPU x 2 コア、メモリ 8 GiB に近い構成になるようにします。

> **Tip**
>
> 各インスタンスタイプに割り当てられるハードウェアリソースの概要については、「Amazon OpenSearch Service の料金」を参照してください。

ただし、上記に記載されたリソースでも不十分になる場合があります。一部の OpenSearch ユーザーは、要件を満たすためにそれらのリソースが何倍も必要であると報告しています。ワークロードに適したハードウェアを見つけるには、知識に基づいた初期予測を作成し、代表的なワークロードでテストし、調整して、再度テストする必要があります。

ステップ 1: 初期見積もりを作成する

開始するには、分割されたブレイン状態 (通信が遅れてクラスターにマスターノードが 2 つある場合) などの潜在的な OpenSearch問題を回避するために、最低 3 つのノードをお勧めします。専用マスターノードが 3 つの場合は、レプリケーション用のデータノードは少なくとも 2 つにすることをお勧めします。

ステップ 2: ノードごとのストレージ要件を計算する

ストレージ要件が 184 GiB で、推奨最小数である 3 つのノードがある場合、各ノードで必要とするストレージの容量は 184/3 = 61 GiB という計算になります。この例では、3 つの `m6g.large.search` インスタンスを選択してそれぞれで 90 GiB の EBS ストレージボリュームを使用すれば、安全策として時間の経過に伴う拡張にも備えることができます。この構成では 6 つの vCPU コアと 24 GiB のメモリを利用でき、軽度のワークロードに適しています。

より大規模な例として、ストレージ要件が 14 TiB (14,336 GiB) で高い負荷が発生する状況を想定します。この場合、まずは 2 * 144 = 288 の vCPU コアと 8 * 144 = 1,152 GiB のメモリを使用したテストが考えられます。これらの数値は、約 18 `i3.4xlarge.search` インスタンスを使用する結果となります。高速なローカルストレージが必要ない場合、それぞれが 1 TiB の EBS ストレージボリュームを使用する 18 個の `r6g.4xlarge.search` インスタンスをテストすることもできます。

クラスターに数百テラバイトのデータが含まれる場合は、「the section called “ペタバイトスケール”」を参照してください。

ステップ 3: 代表的なテストを実行する

クラスターを設定したら、前に計算したシャードの数を使用してインデックスを追加し、現実的なデータセットを使用して代表的なクライアントテストを実行し、 CloudWatch メトリクスをモニタリングして、クラスターがワークロードをどのように処理するかを確認できます。

ステップ 4: 成功または反復する

パフォーマンスがニーズを満たし、テストが成功し、 CloudWatch メトリクスが正常であれば、クラスターを使用する準備が整います。異常なリソースの使用状況を検出する CloudWatch には、アラームを設定することを忘れないでください。

パフォーマンスが許容できないものであれば、テストが失敗したか、または `CPUUtilization` か `JVMMemoryPressure` が高いことになります。別のインスタンスタイプを選択 (またはインスタンスを追加) して、テストを続行する必要があるかもしれません。インスタンスを追加すると、 はクラスター全体のシャードの分散 OpenSearch を自動的に再調整します。

過負荷クラスターの過剰容量は、負荷の低いクラスターの不足よりも簡単に測定できるため、必要以上に大きなクラスターから開始することをお勧めします。次に、追加のリソースを持つ効率的なクラスターにテストおよびスケールダウンして、アクティビティの増加期間中に安定した運用を確保します。

本番稼働用クラスター (1 つまたは複数) では複雑なステータスが発生しますが、専用マスターノードを活用することでパフォーマンスとクラスターの信頼性を向上させることができます。

Amazon OpenSearch Service のペタバイトスケール

Amazon OpenSearch Service ドメインは、最大 3 PB のアタッチされたストレージを提供します。`i3.16xlarge.search` インスタンスタイプが 200 個あり、それぞれに 15 TB のストレージがあるドメインを設定できます。規模がまったく異なるため、このサイズのドメインに関する推奨事項は一般的な推奨事項とは異なります。このセクションでは、ドメインの作成、コスト、ストレージ、シャードのサイズに関する考慮事項について説明します。

このセクションでは、`i3.16xlarge.search` インスタンスタイプを頻繁に参照していますが、他のいくつかのインスタンスタイプを使用して 1 PB の合計ドメインストレージに達することができます。

ドメインの作成

このサイズのドメインは、ドメインあたり 80 インスタンスというデフォルトの制限を超えています。ドメインあたり 200 インスタンスへのサービス制限引き上げをリクエストするには、AWS サポートセンターでサポートケースを作成します。

料金

このサイズのドメインを作成する前に、Amazon OpenSearch Service の料金ページをチェックして、関連するコストが想定どおりであることを確認します。ホットウォームアーキテクチャがユースケースに合っているかどうかを確認するために the section called “UltraWarm ストレージ” を調べます。

[Storage (ストレージ)]

`i3` インスタンスタイプは、高速な不揮発性メモリエクスプレス (NVMe) ローカルストレージを提供するよう特別に設計されたものです。Amazon Elastic Block Store と比較すると、このローカルストレージはパフォーマンス上の利点をもたらす傾向があるため、EBS ボリュームは、OpenSearch サービスでこれらのインスタンスタイプを選択する場合のオプションではありません。必要に応じて EBS ストレージを使用する場合は、`r6.12xlarge.search` など、別のインスタンスタイプを使用します。

シャードのサイズと数

一般的な OpenSearch ガイドラインは、シャードあたり 50 GB を超えないようにすることです。大きなドメインに対応するために必要なシャードの数および `i3.16xlarge.search` インスタンスが利用可能なリソースが指定されている場合は、シャードのサイズは 100 GB をお勧めします。

たとえば、450 TB のソースデータがあり、レプリカを 1 つにする場合、最小ストレージ要件は 450 TB * 2 * 1.1/0.95 = 1.04 PB です。この計算の説明については、「the section called “ストレージ要件の計算”」を参照してください。1.04 PB/15 TB = 70 インスタンスですが、時間と共に変動するデータ量を考慮して、ストレージセーフティネットを提供し、ノードの障害を処理するために、90 個以上の `i3.16xlarge.search` インスタンスを選択することができます。各インスタンスにより最小ストレージ要件に 20 GiB が追加されますが、このサイズのディスクでは、この 20 GiB はほぼ無視できます。

シャード数の制御は、trigy. OpenSearch users はインデックスを毎日ローテーションし、1～2 週間データを保持することがよくあります。このような状況では、「アクティブ」なシャードと「非アクティブ」なシャードを区別すると便利な場合があります。アクティブなシャードとは、書き込みや読み取りがアクティブに行われているシャードです。非アクティブなシャードとは、いくつかの読み取りリクエストがあるものの大部分はアイドルな状態のサービスです。一般的には、アクティブなシャードの数を数千個未満に維持する必要があります。アクティブなシャードの数が 10,000 に近づくと、パフォーマンスと安定性に大きなリスクが出現します。

プライマリシャードの数は次の式を使用して計算します。450,000 GB * 1.1/(100 GB/シャード) = 4,950 シャード。レプリカを考慮してこの数を 2 倍にすると 9,900 シャードになり、すべてのシャードがアクティブの場合は大きな懸念事項になります。ただし、インデックスをローテーションして、任意の指定日にシャードの数の 1/7 または 1/14 (それぞれ 1,414 または 707 シャード) のみがアクティブな場合、クラスターはうまく機能します。この場合も、ドメインのサイズ決定と設定において最も重要なステップは、現実的なデータセットを使用して代表的なクライアントテストを実行することです。

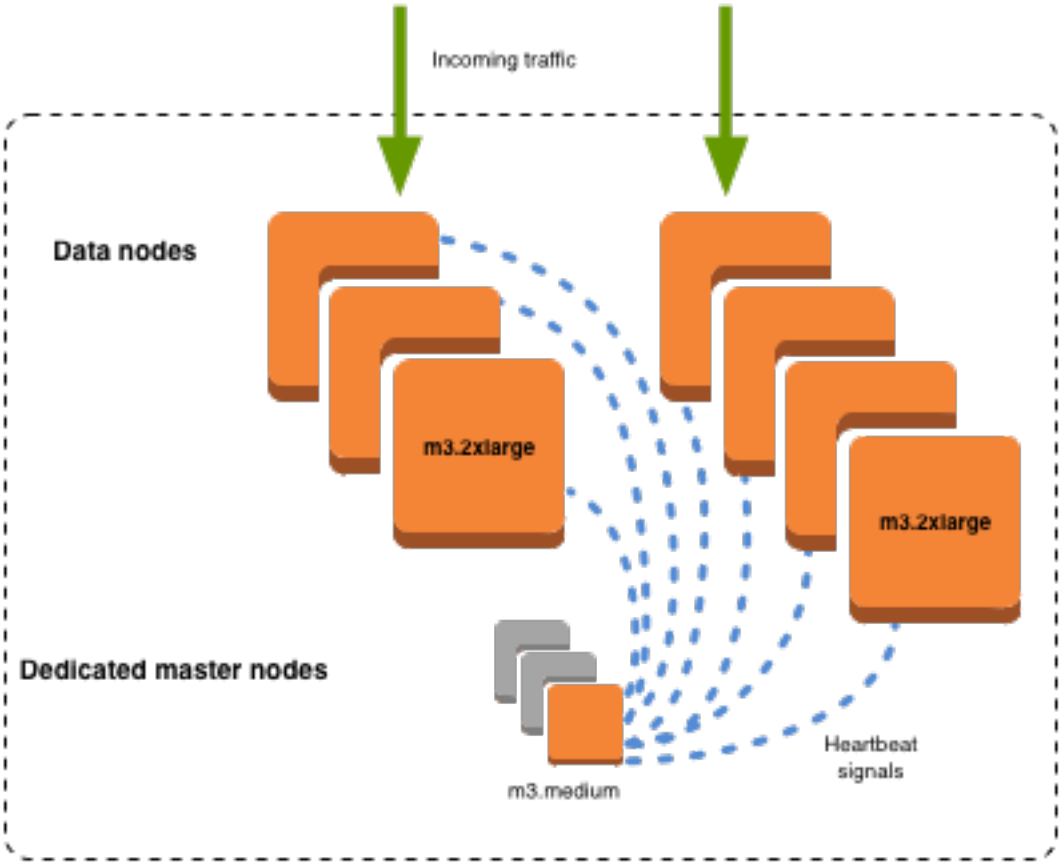
Amazon OpenSearch Service の専用マスターノード

Amazon OpenSearch Service は、専用マスターノードを使用してクラスターの安定性を向上させます。専用マスターノードではクラスター管理タスクを実行しますが、データは保持せず、データのアップロードリクエストにも応答しません。このように、クラスター管理タスクをオフロードすると、ドメインの安定性が向上します。他のすべてのノードタイプと同様に、専用マスターノードごとに時間料金を支払います。

専用マスターノードで実行するクラスター管理タスクは次のとおりです。

- クラスター内のすべてのノードを追跡します。
- クラスター内のインデックスの数を追跡します。
- 各インデックスに属するシャード数を追跡します。
- クラスター内のノードのルーティング情報を保持します。
- クラスター内のインデックス作成やノードの作成/削除など、状態が変化したときにクラスターの状態を更新します。
- クラスターの状態に施された変更を、クラスター内のすべてのノードにわたって複製します。
- クラスター内のデータノードの可用性をモニタリングするハートビートシグナル (定期的なシグナル) を送信することで、すべてのクラスターノードの状態をモニタリングします。

次の図は、10 個のインスタンスを持つ OpenSearch サービスドメインを示しています。インスタンスのうち 7 つはデータノードで、3 つは専用マスターノードです。アクティブになっているのは、専用マスターノードの 1 つだけです。2 つの灰色の専用マスターノードは、アクティブな専用マスターノードに障害が発生した場合のバックアップとして待機します。すべてのデータアップロードリクエストは、7 つのデータノードにより保存され、すべてのクラスター管理タスクは、アクティブな専用マスターノードにオフロードされます。



専用マスターノードの数の選択

マルチ AZ とスタンバイを使用することをお勧めします。スタンバイは、各本番 OpenSearch サービスドメインに 3 つの専用マスターノードを追加します。スタンバイなしのマルチ AZ 、またはシングル AZ でデプロイする場合でも、3 つの専用マスターノードを推奨します。偶数の専用マスターノードを選択しないでください。専用マスターノードの数を選択するときは、次の点を考慮してください。

- 1 つの専用マスターノードは、障害発生時にバックアップがないため、 OpenSearch サービスによって明示的に禁止されています。1 つの専用マスターノードのみを持つドメインを作成しようとすると、検証例外が表示されます。
- 2 つの専用マスターノードがある場合、クラスターには、障害が発生した場合に新しいマスターノードを選択するために必要なノードのクォーラムがありません。

クォーラムは、専用マスターノードの数/2 + 1 (直近の整数まで切り捨て) です。ここでは 2/2 + 1 = 2 となります。専用マスターノード 1 つが失敗し、バックアップは 1 つのみであれば、クラスターにはクォーラムがなく、新しいマスターを選択できません。

- 推奨される専用マスターノード数は 3 つです。マスターノードが失敗したときに 2 つのバックアップノードが使用でき、必要なクォーラム (2) で新しいマスターを選択できます。
- 専用マスターノードを 4 つにすると 3 つより良くなるわけではなく、複数のアベイラビリティーゾーンを使用した場合、問題が発生する可能性があります。
  - マスターノード 1 つが失敗した場合、クォーラム (3) で新しいマスターを選択します。2 つのノードが失敗したした場合、そのクォーラムは失われ、専用マスターノードが 3 つの場合と同じ状況になります。
  - 3 つのアベイラビリティーゾーン (AZ) 設定では、2 つの AZ に 1 つの専用マスターノードがあり、1 つの AZ に 2 つの専用マスターノードがあります。その AZ で中断が発生した場合、残りの 2 つの AZ には、新しいマスターを選択するために必要なクォーラム (3) がありません。
- 5 つの専用マスターノードは 3 つと同様に機能し、クォーラムを維持しながら 2 つのノードを失うことができます。ただし、一度にアクティブになる専用マスターノードは 1 つしかないため、この設定では 4 つのアイドルノードに対して料金を支払うことになります。多くのお客様は、このレベルのフェイルオーバーは過剰だと考えています。

クラスターに偶数のマスター対象ノードがあり、Elasticsearch OpenSearch バージョン 7.x 以降では 1 つのノードを無視して、投票設定が常に奇数になるようにした場合。この場合、基本的に専用マスターノード 4 つは 3 つ (2 つは 1 つ) に相当します。

> **Note**
>
> 新しいマスターノードを選択するために必要なクォーラムがクラスターにない場合は、クラスターへの書き込みおよび読み取りリクエストの両方が失敗します。この動作は OpenSearch デフォルトとは異なります。

専用マスターノードのインスタンスタイプの選択

専用マスターノードでは検索およびクエリリクエストを処理しませんが、そのサイズは、管理可能なインスタンスサイズや、インスタンス数、インデックス数、シャード数と大きな相関があります。本番稼働用クラスターでは、最低でも専用マスターノードに以下のインスタンスタイプをお勧めします。

これらの推奨事項は一般的なワークロードに基づいており、お客様の要件によって異なります。多数のシャードまたはフィールドのマッピングがあるクラスターは、大きなインスタンスタイプからメリットを得ることができます。専用マスターノードのメトリクスを監視し、より大きいインスタンスタイプを使用する必要があるかどうかを調べてください。

インスタンス数	マスターノードの RAM サイズ	サポートされるシャードカウントの上限	推奨される最小専用マスターインスタンスタイプ
1 ~ 10	8 GiB	10K	`m5.large.search` または `m6g.large.search`
11 ~ 30	16 GiB	30K	`c5.2xlarge.search` または `c6g.2xlarge.search`
31 ~ 75	32 GiB	40K	`r5.xlarge.search` または `r6g.xlarge.search`
76 ~ 125	64 GiB	75K	`r5.2xlarge.search` または `r6g.2xlarge.search`
126 ~ 200	128 GiB	75K	`r5.4xlarge.search` または `r6g.4xlarge.search`

- 特定の構成変更が専用マスターノードに与える影響の詳細については、「the section called “設定変更”」を参照してください。
- インスタンス数の制限の詳細については、OpenSearch 「サービスドメインとインスタンスクォータ」を参照してください。

- vCPU、メモリ、料金など、特定のインスタンスタイプの詳細については、「Amazon OpenSearch Service の料金」を参照してください。

Amazon OpenSearch Service の推奨 CloudWatch アラーム

CloudWatch アラームは、 CloudWatch メトリクスが指定された値を一定時間超えたときにアクションを実行します。例えば、クラスター AWS のヘルスステータスが 1 red 分以上の場合は、E メールで送信できます。このセクションでは、Amazon OpenSearch Service に推奨されるアラームとその対応方法について説明します。

これらのアラームは、 を使用して自動的にデプロイできます AWS CloudFormation。サンプルスタックについては、関連するGitHubリポジトリ を参照してください。

> **Note**
>
> CloudFormation スタックをデプロイすると、 KMSKeyErrorおよび KMSKeyInaccessibleアラームは Insufficient Data状態になります。これらのメトリクスは、ドメインで暗号化キーに問題がある場合にのみ表示されるためです。

アラームの設定の詳細については、「Amazon CloudWatch ユーザーガイド」の「Amazon アラームの作成」を参照してください。 CloudWatch

アラーム	問題
ClusterStatus.red maximum is >= 1 for 1 minute, 1 consecutive time	少なくとも 1 つのプライマリシャードとそのレプリカがノードに割り当てられていません。「the section called “赤のクラスター状態”」を参照してください。
ClusterStatus.yellow maximum is >= 1 for 1 minute, 5 consecutive times	少なくとも 1 つのレプリカシャードがノードに割り当てられていません。「the section called “黄色のクラスター状態”」を参照してください。
FreeStorageSpace minimum is <=	クラスターのノードの空きストレージ容量が 20 GiB に下がっています。「the section called “使用可能なストレージ領域の不足”」を参照し

アラーム	問題
20480 for 1 minute, 1 consecutive time	てください。この値は MiB 単位です。20480 ではなく、各ノードのストレージ容量の 25% に設定することをお勧めします。
`ClusterIndexWritesBlocked` is >= 1 for 5 minutes, 1 consecutive time	クラスターは書き込みリクエストをブロックしています。「the section called “ClusterBlockException”」を参照してください。
`Nodes` minimum is < x for 1 day, 1 consecutive time	x はクラスター内のノード数です。このアラームは、クラスター内の少なくとも 1 つのノードが 1 日間にわたってアクセスできない状態を意味します。「the section called “障害が発生したクラスターノード”」を参照してください。

`AutomatedSnapshotFailure` maximum is >= 1 for 1 minute, 1 consecutive time	自動スナップショットが失敗しました。多くの場合、この失敗によってクラスター状態が赤になります。「the section called “赤のクラスター状態”」を参照してください。
	
	すべての自動スナップショットの概要および障害に関する情報を取得するには、次のリクエストのいずれかを試してください。
	
	`GET domain_endpoint /_snapshot/cs-automated/_all`
	`GET domain_endpoint /_snapshot/cs-automated-enc/_all`

`CPUUtilization` または `WarmCPUUtilization` maximum is >= 80% for 15 minutes, 3 consecutive times	100% の CPU 使用率が発生することもありますが、高い使用率が持続するのは問題です。より大規模なインスタンスタイプを使用するか、インスタンスを追加することを検討してください。

アラーム	問題
`JVMMemory Pressure` maximum is >= 95% for 1 minute, 3 consecutive times	使用量が増加した場合にクラスターでメモリ不足エラーが発生する可能性があります。垂直スケーリングを検討してください。 OpenSearch サービスは Java ヒープにインスタンスの RAM の半分を使用し、ヒープサイズは 32 GiB までです。インスタンスは最大 64 GiB の RAM まで垂直スケーリングでき、それ以上はインスタンスを追加することで水平方向にスケーリングできます。
`OldGenJVM MemoryPressure` maximum is >= 80% for 1 minute, 3 consecutive times	
`MasterCPU Utilization` maximum is >= 50% for 15 minutes, 3 consecutive times	より大規模なインスタンスタイプを専用マスターノードとして使用することを検討してください。クラスターの安定性と Blue/Green デプロイに関わるため、専用マスターノードの CPU 使用率はデータノードよりも低くする必要があります。
`MasterJVM MemoryPressure` maximum is >= 95% for 1 minute, 3 consecutive times	
`MasterOld GenJVMMem oryPressure` maximum is >= 80% for 1 minute, 3 consecutive times	
`KMSKeyError` is >= 1 for 1 minute, 1 consecutive time	ドメイン内の保管中のデータの AWS KMS 暗号化に使用される暗号化キーは無効になっています。通常のオペレーションを復元するために、再度有効にしてください。詳細については、「the section called “保管中の暗号化”」を参照してください。

アラーム	問題
KMSKeyInaccessible is >= 1 for 1 minute, 1 consecutive time	ドメイン内の保管中のデータの暗号化に使用される AWS KMS 暗号化キーが削除されたか、 OpenSearch サービスへの許可が取り消されました。この状態にあるドメインを復元することはできません。ただし、手動スナップショットがある場合は、それを使用して新しいドメインに移行できます。詳細については、「the section called “保管中の暗号化”」を参照してください。
shards.active is >= 30000 for 1 minute, 1 consecutive time	アクティブなプライマリとレプリカの両方のシャードの合計数は、30,000よりも多くなります。インデックスを頻繁にローテーションさせ過ぎている可能性があります。特定の年齢に達したら、ISM を使用してインデックスを削除することを検討してください。
5*xx* alarms >= 10% of OpenSearchRequests	1 つ以上のデータノードが過負荷になっているか、アイドルタイムアウト時間内にリクエストが完了しない可能性があります。より大きなインスタンスタイプに切り替えるか、クラスターにさらにノードを追加することを検討してください。シャードおよびクラスターアーキテクチャ用のベストプラクティスをフォローしていることを確認してください。
MasterReachableFromNode 最大 は 5 分間 < 1、連続 1 回	このアラームは、マスターノードが停止しているか、連絡不能であることを示します。これらの障害は通常、ネットワーク接続の問題または AWS 依存関係の問題が原因です。
ThreadpoolWriteQueue average is >= 100 for 1 minute, 1 consecutive time	クラスタ-では、インデックス作成の同時実行性が高くなっています。インデックス作成リクエストを点検して抑制するか、クラスターリソースを増やします。
ThreadpoolSearchQueue average is >= 500 for 1 minute, 1 consecutive time	クラスターでは、検索の同時実行性が高くなっています。クラスターのスケーリングを検討してください。検索キューのサイズを大きくすることもできますが、そうすると、メモリ不足エラーが発生する可能性があります。

アラーム	問題
`Threadpool SearchQueue` maximum is >= 5000 for 1 minute, 1 consecutive time	
Increase in `Threadpool SearchRejected` SUM is >=1{ math expression DIFF ( )} for 1 minute, 1 consecutive time	これらのアラームは、パフォーマンスや安定性に影響を及ぼす可能性のあるドメインの問題を通知します。
Increase in `Threadpool WriteRejected` SUM is >=1{ math expression DIFF ( )} for 1 minute, 1 consecutive time	

> **Note**
>
> メトリクスを表示するのみであれば、「the section called “クラスターメトリクスのモニタリング”」を参照してください。

検討した方が良いその他のアラーム

定期的に使用する OpenSearch サービス機能に応じて、次のアラームを設定することを検討してください。

アラーム	問題
WarmFreeStorageSpace は >= 10%	空きウォームストレージの合計の 10% に達しました。 は、空きウォームストレージ容量の合計を MiB でWarmFreeStorageSpace 測定します。 は、アタッチされたディスクではなく Amazon S3 UltraWarm を使用します。 MiB
HotToWarmMigrationQueueSize is >= 20 for 1 minute, 3 consecutive times	多数のインデックスがホットから UltraWarm ストレージに同時に移動しています。クラスターのスケーリングを検討してください。
HotToWarmMigrationSuccessLatency is >= 1 day, 1 consecutive time	デイリーインデックスを動かそうとして HotToWarmMigrationSuccessCount x レイテンシーが 24 時間を超える場合に通知されるよう、このアラームを設定してください。
WarmJVMMemoryPressure maximum is >= 95% for 1 minute, 3 consecutive times	使用量が増加した場合にクラスターでメモリ不足エラーが発生する可能性があります。垂直スケーリングを検討してください。 OpenSearch サービスは Java ヒープにインスタンスの RAM の半分を使用し、ヒープサイズは 32 GiB までです。インスタンスは最大 64 GiB の RAM まで垂直スケーリングでき、それ以上はインスタンスを追加することで水平方向にスケーリングできます。
WarmOldGenJVMMemoryPressure maximum is >= 80% for 1 minute, 3 consecutive times	
WarmToColdMigrationQueueSize is >= 20 for 1 minute, 3 consecutive times	多数のインデックスが同時に から UltraWarm コールドストレージに移行しています。クラスターのスケーリングを検討してください。

アラーム	問題
HotToWarmMigrationFailureCount is >= 1 for 1 minute, 1 consecutive time	スナップショット、シャード再配置、または強制マージ中に、移行が失敗する可能性があります。スナップショットまたはシャード再配置中の障害は、通常、ノードの障害または S3 接続の問題が原因です。通常、ディスク領域の不足は、強制マージ失敗の根本的な原因です。
WarmToColdMigrationFailureCount is >= 1 for 1 minute, 1 consecutive time	インデックスメタデータをコールドストレージに移動させようとして失敗すると、通常、移行は失敗します。ウォームインデックスクラスター状態が削除されたときにも、障害が発生する可能性があります。
WarmToColdMigrationLatency is >= 1 day, 1 consecutive time	デイリーインデックスを動かそうとして WarmToColdMigrationSuccessCount x レイテンシーが 24 時間を超える場合に通知されるよう、このアラームを設定してください。
AlertingDegraded is >= 1 for 1 minute, 1 consecutive time	アラートインデックスが赤色であるか、1 つ以上のノードがスケジュールどおりでないことを意味します。
ADPluginUnhealthy is >= 1 for 1 minute, 1 consecutive time	異常検出プラグインが正しく動作していません。これは、障害率が高いか、使用されているインデックスの 1 つが赤色であるためです。
AsynchronousSearchFailureRate is >= 1 for 1 minute, 1 consecutive time	少なくとも 1 つの非同期検索が直前に失敗しました。これは、おそらくコーディネータノードが失敗したことを意味します。非同期検索リクエストのライフサイクルは、コーディネータノードでのみ管理されているので、コーディネータがダウンすると、リクエストは失敗します。

アラーム	問題
`AsynchronousSearchStoreHealth` is >= 1 for 1 minute, 1 consecutive time	残存するインデックス内の非同期検索レスポンスストアの状態は、赤色です。大量の非同期レスポンスを保存している可能性があり、クラスターが不安定になる可能性があります。非同期検索レスポンスを 10 MB 以下に制限してください。
`SQLUnhealthy` is >= 1 for 1 minute, 3 consecutive times	SQL プラグインが 5xx レスポンスコードを返すか、無効なクエリ DSL を に渡しています OpenSearch。クライアントがプラグインに対して行っているリクエストのトラブルシューティングを行います。
`LTRStatus.red` is >= 1 for 1 minute, 1 consecutive time	Learning to Rankプラグインの実行に必要なインデックスの内、少なくとも 1 つにプライマリシャードがなく、機能しません。

Amazon OpenSearch Service の一般的なリファレンス

Amazon OpenSearch Service は、さまざまなインスタンス、オペレーション、プラグイン、その他のリソースをサポートしています。

トピック

Amazon OpenSearch Service でサポートされているインスタンスタイプ

Amazon OpenSearch Service では、次のインスタンスタイプがサポートされています。すべてのリージョンで、すべてのインスタンスタイプがサポートされているわけではありません。可用性の詳細については、「Amazon OpenSearch Service の料金」を参照してください。

お客様のユースケースに適切なインスタンスタイプ関する情報については the section called “ドメインのサイジング”、the section called “EBS ボリュームサイズのクォータ”、および the section called “ネットワークのクォータ” を参照してください。

現行世代のインスタンスタイプ

最高のパフォーマンスを得るには、新しい OpenSearch サービスドメインを作成するときに、次のインスタンスタイプを使用することをお勧めします。

インスタンスタイプ	インスタンス	制限事項
OR1	`or1.medium.search`	• OR1 インスタンスタイプには OpenSearch 2.11 以降が必要です。

インスタンスタイプ	インスタンス	制限事項
	or1.large.search or1.xlarge.search or1.2xlarge.search or1.4xlarge.search or1.8xlarge.search or1.12xlarge.search or1.16xlarge.search	• OR1 インスタンスは、他の Graviton インスタンスタイプのマスターノード (C6g、M6g、R6g) とのみ互換性があります。

インスタンスタイプ	インスタンス	制限事項

Im4gn	`im4gn.large.search`
	`im4gn.xlarge.search`
	`im4gn.2xlarge.search`
	`im4gn.4xlarge.search`
	`im4gn.8xlarge.search`
	`im4gn.16xlarge.search`
		• Im4gn インスタンスタイプには、Elasticsearch 7.9 以降またはの任意のバージョンが必要であり OpenSearch、EBS ストレージボリュームはサポートされていません。
		• Im4gn インスタンスは、他の Graviton インスタンスタイプ (C6g、M6g、R6g、R6gd) とのみ互換性があります。Graviton インスタンスと非 Graviton インスタンスは、同じクラスター内で結合できません。

インスタンスタイプ	インスタンス	制限事項
C5	c5.large.search c5.xlarge.search c5.2xlarge.search c5.4xlarge.search c5.9xlarge.search c5.18xlarge.search	C5 インスタンスタイプには、Elasticsearch 5.1 以降または の任意のバージョンが必要です OpenSearch。

インスタンスタイプ	インスタンス	制限事項
C6g	c6g.large.search c6g.xlarge.search c6g.2xlarge.search c6g.4xlarge.search c6g.8xlarge.search c6g.12xlarge.search	• C6g インスタンスタイプには、Elasticsearch 7.9 以降または の任意のバージョンが必要です OpenSearch。 • C6g インスタンスは、他の Graviton インスタンスタイプ (Im4gn、M6g、R6g、R6gd) とのみ互換性があります。Graviton インスタンスと非 Graviton インスタンスは、同じクラスター内で結合できません。

インスタンスタイプ	インスタンス	制限事項
I3	i3.large.search i3.xlarge.search i3.2xlarge.search i3.4xlarge.search i3.8xlarge.search i3.16xlarge.search	I3 インスタンスタイプには、Elasticsearch 5.1 以降または の任意のバージョンが必要であり OpenSearch、EBS ストレージボリュームはサポートされていません。
M5	m5.large.search m5.xlarge.search m5.2xlarge.search m5.4xlarge.search m5.12xlarge.search	M5 インスタンスタイプには、Elasticsearch 5.1 以降または の任意のバージョンが必要です OpenSearch。

インスタンスタイプ	インスタンス	制限事項
M6g	m6g.large.search m6g.xlarge.search m6g.2xlarge.search m6g.4xlarge.search m6g.8xlarge.search m6g.12xlarge.search	• M6g インスタンスタイプには、Elasticsearch 7.9 以降または の任意のバージョンが必要です OpenSearch。 • M6g インスタンスは、他の Graviton インスタンスタイプ (Im4gn、C6g、R6g、R6gd) とのみ互換性があります。Graviton インスタンスと非 Graviton インスタンスは、同じクラスター内で結合できません。

インスタンスタイプ	インスタンス	制限事項
R5	r5.large.search r5.xlarge.search r5.2xlarge.search r5.4xlarge.search r5.12xlarge.search	R5 インスタンスタイプには、Elasticsearch 5.1 以降または の任意のバージョンが必要です OpenSearch。

インスタンスタイプ	インスタンス	制限事項
R6g	r6g.large.search r6g.xlarge.search r6g.2xlarge.search r6g.4xlarge.search r6g.8xlarge.search r6g.12xlarge.search	• R6g インスタンスタイプには、Elasticsearch 7.9 以降または の任意のバージョンが必要です OpenSearch。 • R6g インスタンスは、他の Graviton インスタンスタイプ (Im4gn、C6g、M6g、R6gd) とのみ互換性があります。Graviton インスタンスと非 Graviton インスタンスは、同じクラスター内で結合できません。

インスタンスタイプ	インスタンス	制限事項
R6gd	r6gd.large.search r6gd.xlarge.search r6gd.2xlarge.search r6gd.4xlarge.search r6gd.8xlarge.search r6gd.12xlarge.search r6gd.16xlarge.search	• R6gd インスタンスタイプには、Elasticsearch 7.9 以降またはの任意のバージョンが必要で OpenSearch、EBS ストレージボリュームはサポートされていません。 • R6gd インスタンスは、他の Graviton インスタンスタイプ (Im4gn、C6g、M6g、R6g) とのみ互換性があります。Graviton インスタンスと非 Graviton インスタンスは、同じクラスター内で結合できません。

インスタンスタイプ	インスタンス	制限事項
T3	t3.small.search t3.medium.search	• T3 インスタンスタイプには、Elasticsearch 5.6 以降または の任意のバージョンが必要です OpenSearch。 • T3 インスタンスタイプは、ドメインがスタンバイなしでプロビジョニングされている場合にのみ使用できます。詳細については、「the section called “Multi-AZ without Standby”」を参照してください。 • T3 インスタンスタイプは、ドメインのインスタンス数が 10 以下の場合にのみ使用できます。 • T3 インスタンスタイプは、ストレージ、コールドストレージ、または Auto-Tune をサポート UltraWarmしていません。

旧世代のインスタンスタイプ

OpenSearch サービスでは、アプリケーションを最適化し、まだアップグレードしていないユーザー向けに、旧世代のインスタンスタイプを提供しています。最高のパフォーマンスを得るには、現世代のインスタンスタイプの使用をお勧めしますが、以下の旧世代のインスタンスタイプも引き続きサポートします。

インスタンスタイプ	インスタンス	制限事項
C4	c4.large.search c4.xlarge.search c4.2xlarge.search c4.4xlarge.search	

インスタンスタイプ	インスタンス	制限事項
	c4.8xlarge.search	
I2	i2.xlarge.search i2.2xlarge.search	
M3	m3.medium.search m3.large.search m3.xlarge.search m3.2xlarge.search	• M3 インスタンスタイプでは、保管時のデータの暗号化、きめ細かなアクセス制御、クラスター間検索がサポートされません。 • M3 インスタンスタイプには、 OpenSearch バージョン別の追加の制限があります。詳細については、「the section called “無効な M3 インスタンスタイプ”」を参照してください。
M4	m4.large.search m4.xlarge.search m4.2xlarge.search m4.4xlarge.search m4.10xlarge.search	

インスタンスタイプ	インスタンス	制限事項
R3	r3.large.search r3.xlarge.search r3.2xlarge.search r3.4xlarge.search r3.8xlarge.search	R3 インスタンスタイプは、保管時のデータの暗号化またはきめ細かなアクセスコントロールをサポートしていません。
R4	r4.large.search r4.xlarge.search r4.2xlarge.search r4.4xlarge.search r4.8xlarge.search r4.16xlarge.search	

インスタンスタイプ	インスタンス	制限事項

T2	`t2.micro.search`
	`t2.small.search`
	`t2.medium.search`
		• T2 インスタンスタイプは、ドメインのインスタンス数が 10 以下の場合のみ使用できます。
		• `t2.micro.search` インスタンスタイプは、Elasticsearch 1.5 および 2.3 のみをサポートします。
		• T2 インスタンスタイプは、保管中のデータの暗号化、きめ細かなアクセスコントロール、 UltraWarmストレージ、コールドストレージ、クラスター間検索、または Auto-Tune をサポートしていません。

> **Tip**
>
> 専用マスターノードおよびデータノードの異なるインスタンスタイプをお勧めする場合があります。

Amazon OpenSearch Service のエンジンバージョン別の機能

多くの OpenSearch サービス機能には、最小 OpenSearch バージョン要件またはレガシー Elasticsearch OSS バージョン要件があります。機能の最小バージョンを満たしていても、その機能がドメインで利用できない場合は、ドメインのサービスソフトウェアを更新します。

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
VPC サポート	1.0	1.0
ドメインへのすべてのトラフィックに HTTPS を要求する		

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
マルチ AZ のサポート		
専用マスターノード		
カスタムパッケージ		
カスタムエンドポイント		
スローログの公開		
エラーログの公開	1.0	5.1
保管中のデータの暗号化		
OpenSearch Dashboards の Cognito 認証		
インプレースアップグレード		
Curator のサポート	含まれない	5.1

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
時間単位の自動スナップショット	1.0	5.3
N ode-to-node 暗号化	1.0	6.0
Java 高レベル REST クライアントのサポート		
HTTP リクエストとレスポンス圧縮		
アラート	1.0	6.2
SQL	1.0	6.5
クラスター間検索	1.0	6.7
きめ細かなアクセスコントロール		
OpenSearch Dashboards の SAML 認証		
Auto-Tune		

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
リモート再インデックス		
UltraWarm	1.0	6.8
インデックスステート管理		
K-NN (ユークリッド距離)	1.0	7.1
異常検出	1.0	7.4
コサイン類似度による k-NN	1.0	7.7
Learning to Rank		
パイプ処理言語	1.0	7.9
OpenSearch Dashboards レポート		
OpenSearch ダッシュボードトレース分析		

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
ARM ベースの Graviton インスタンス		
コールドストレージ		
ハミング距離、L1 ノルム距離、k-NN 用 Painless スクリプティング	1.0	7.10
非同期検索		
インデックス変換	1.0	含まれない
クラスター間レプリケーション	1.1	7.10
ML Commons	1.3	含まれない
通知	2.3	含まれない
特定時点検索	2.5	含まれない
パイプラインを検索	2.9	含まれない

機能	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
機械学習コネクタ	2.9	含まれない
マルチモーダルセマンティック検索	2.11	含まれない
Amazon S3 のダイレクトクエリのデータソース	2.11	含まれない

これらの機能の一部と追加の機能を有効にするプラグインについては、「the section called “エンジンバージョンに応じたプラグイン”」を参照してください。各バージョンの OpenSearch API の詳細については、「」を参照してくださいthe section called “サポートされているオペレーション”。

Amazon OpenSearch Service のエンジンバージョン別のプラグイン

Amazon OpenSearch Service ドメインには、 OpenSearch コミュニティのプラグインがあらかじめパッケージ化されています。このサービスはプラグインを自動的にデプロイおよび管理しますが、ドメイン用に選択した OpenSearch またはレガシー Elasticsearch OSS のバージョンに応じて異なるプラグインをデプロイします。

次の表に、プラグインを OpenSearch バージョン別、および互換性のあるレガシー Elasticsearch OSS のバージョン別に一覧表示します。これには、操作できるプラグインのみが含まれます。これは包括的ではありません。 OpenSearch サービスでは、追加のプラグインを使用して、スナップショット用の S3 リポジトリプラグインや、最適化とモニタリング用の OpenSearch Performance Analyzer プラグインなどのコアサービス機能を有効にします。ドメインで実行されているすべてのプラグインの完全なリストについては、次のリクエストを実行します。

```
GET _cat/plugins?v
```

プラグイン	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
ICU Analysis	1.0	すべてのドメインで含まれる
Japanese (kuromoji) Analysis		
Phonetic Analysis	1.0	2.3
Seunjeon 韓国語分析	1.0	5.1
Smart Chinese Analysis		
Stempel Polish Analysis		
Ingest Attachment Processor		
Ingest User Agent Processor		
Mapper Murmur3		

プラグイン	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
Mapper Size	1.0	5.3
Ukrainian Analysis		
OpenSearch アラート	1.0	6.2
OpenSearch SQL	1.0	6.5
OpenSearch セキュリティ	1.0	6.7
OpenSearch インデックスステート管理	1.0	6.8
OpenSearch k-NN	1.0	7.1
OpenSearch 異常検出	1.0	7.4
IK (中国語) 分析	1.0	7.7
ベトナム語分析		
タイ語分析		

プラグイン	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
Learning to Rank		
OpenSearch 非同期検索	1.0	7.10
OpenSearch クラスター間のレプリケーション	1.1	7.10
OpenSearch オブザーバビリティ	1.2	サポートされていません
Nori 分析	1.3	サポートされていません
ピンイン分析	1.3	サポートされていません
STConvert	1.3	サポートされていません
スダチ分析	1.3	サポートされていません
ML Commons	1.3	サポートされていません
OpenSearch 通知	2.3	サポートされていません
セキュリティ分析	2.5	サポートされていません

プラグイン	最低限必要な OpenSearch バージョン	最低限必要な Elasticsearch バージョン
Neural Search	2.9	サポートされていません
Amazon Personalize の検索ランキング	2.9	サポートされていません
ヘブライ語分析	2.11	サポートされていません
HanLP	2.11	サポートされていません

オプションプラグイン

Amazon OpenSearch Service は、プリインストールされているデフォルトのプラグインに加えて、いくつかのオプションの言語アナライザープラグインをサポートしています。 AWS Management Console および を使用して AWS CLI 、プラグインをドメインに関連付けたり、ドメインからプラグインの関連付けを解除したり、すべてのプラグインを一覧表示したりできます。オプションのプラグインパッケージは、特定の OpenSearch バージョンと互換性があり、そのバージョンのドメインにのみ関連付けることができます。

Sudachi プラグイン の場合、辞書ファイルを再関連付けしても、すぐにドメインに反映されないことに注意してください。設定変更やその他の更新の一環として、次のブルー/グリーンデプロイがドメインで実行されると、辞書が更新されます。または、更新されたデータを使用して新しいパッケージを作成し、この新しいパッケージを使用して新しいインデックスを作成し、既存のインデックスを新しいインデックスに再インデックスしてから、古いインデックスを削除することもできます。インデックスの再作成方法を使用する場合は、トラフィックが中断されないようにインデックスエイリアスを使用してください。

オプションプラグインは `ZIP-PLUGIN` パッケージタイプを使用します。オプションのプラグインについての詳細は、「the section called “カスタムパッケージ”」を参照してください。

Amazon OpenSearch Service でサポートされているオペレーション

OpenSearch サービスは、 OpenSearch およびレガシー Elasticsearch OSS の多くのバージョンをサポートしています。以下のセクションでは、 OpenSearch サービスが各バージョンでサポートするオペレーションについて説明します。

トピック

- Elasticsearch バージョン 6.0
- Elasticsearch バージョン 5.6
- Elasticsearch バージョン 5.5
- Elasticsearch バージョン 5.3
- Elasticsearch バージョン 5.1
- Elasticsearch バージョン 2.3
- Elasticsearch バージョン 1.5

API の重要な相違点

設定と統計

OpenSearch サービスは、「フラット」設定フォームを使用する `_cluster/settings` API への PUT リクエストのみを受け入れます。拡張設定フォームを使用するリクエストは拒否します。

```
// Accepted
PUT _cluster/settings
{
  "persistent" : {
    "action.auto_create_index" : false
  }
}

// Rejected
PUT _cluster/settings
{
  "persistent": {
    "action": {
      "auto_create_index": false
    }
  }
}
```

高レベル Java REST クライアントは拡張フォームを使用するため、設定リクエストを送信する必要がある場合は、低レベルクライアントを使用します。

Elasticsearch 5.3 以前は、次の例に示すように、 OpenSearch サービスドメインの `_cluster/settings` API は HTTP PUTメソッドのみをサポート OpenSearch していましたが、 `GET` メソッド

はサポートしていません。以降のバージョンの Elasticsearch は GETメソッドをサポートしています。

```
GET https://domain-name.region.es.amazonaws.com/_cluster/settings?pretty
```

戻り値の例を次に示します。

```
{
  "persistent": {
    "cluster": {
      "routing": {
        "allocation": {
          "cluster_concurrent_rebalance": "2",
          "node_concurrent_recoveries": "2",
          "disk": {
            "watermark": {
              "low": "1.35gb",
              "flood_stage": "0.45gb",
              "high": "0.9gb"
            }
          },
          "node_initial_primarirecoveries": "4"
        }
      }
    },
    "indices": {
      "recovery": {
        "max_bytper_sec": "40mb"
      }
    }
  }
}
```

特定の設定および統計 API について、オープンソース OpenSearch クラスターと OpenSearch サービスからのレスポンスを比較すると、フィールドが欠落していることがあります。 OpenSearch サービスは、 からのファイルシステムデータパス_nodes/statsや、 からのオペレーティングシステムの名前とバージョンなど、サービス内部を公開する特定の情報を編集します_nodes。 APIs

縮小

_shrink API により、アップグレード、設定変更、ドメインの削除が失敗する場合があります。Elasticsearch バージョン 5.3 または 5.1 を実行するドメインでは、使用しないことをお勧めし

ます。これらのバージョンは収縮させたインデックススナップショットの復元の失敗を引き起こす可能性があるバグがあります。

他の Elasticsearch または OpenSearchバージョンで _shrink API を使用する場合は、縮小オペレーションを開始する前に次のリクエストを行います。

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": "name-of-the-node-to-shrink-to",
    "index.blocks.read_only": true
  }
}
```

縮小操作の完了後に次のリクエストを作成します。

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}

PUT https://domain-name.region.es.amazonaws.com/shrunken-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}
```

OpenSearch バージョン 2.13

OpenSearch 2.13 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (/*index-nam*
- /_delete_by_query [1]
- /_explain
- /_refresh
- /_reindex [1]

e /_forcemerge 、/*index-name* /update/*id*、および /*index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
  - cluster.search.request.slowlog.level
  - cluster.search.request.slowlog.threshold.warn
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

    - cluster.search.request.slowlog.threshold.info
    - cluster.search.request.slowlog.threshold.debug
    - cluster.search.request.slowlog.threshold.trace
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 2.11

OpenSearch 2.11 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 2.9

OpenSearch 2.9 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-*
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

name /update/*id*、および /*index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 2.7

OpenSearch 2.7 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。

4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 2.5

OpenSearch 2.5 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 2.3

OpenSearch 2.3 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- _plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_percolate
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 1.3

OpenSearch 1.3 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-*
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

name /update/*id*、および /*index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 1.2

OpenSearch 1.2 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。

4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 1.1

OpenSearch 1.1 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. the section called “縮小” を参照してください。

OpenSearch バージョン 1.0

OpenSearch 1.0 では、 OpenSearch Service は次のオペレーションをサポートしています。ほとんどのオペレーションの詳細については、OpenSearchREST API リファレンス または特定のプラグインの API リファレンスを参照してください。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用 OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 7.10

Elasticsearch 7.10 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index

- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_index_template
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_asynchronous_search
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_plugins/_replication
- /_rank_eval
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。

2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。
6. 従来のインデックステンプレート (_template) は、Elasticsearch 7.8 で開始する組み合わせ可能なテンプレート (_index_template) によって置き換えられました。組み合わせ可能なテンプレートは、従来のテンプレートよりも優先されます。指定されたインデックスに一致する組み合わせ可能なテンプレートがない場合、従来のテンプレートは引き続き一致して、適用できます。_template オペレーションは引き続き OpenSearch 以降のバージョンの Elasticsearch OSS で機能しますが、2 つのテンプレートタイプへの GET 呼び出しでは異なる結果が返されます。

Elasticsearch バージョン 7.9

Elasticsearch 7.9 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot

- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_ppl
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch サービスがサポートする汎用

OpenSearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。

5. 「the section called “縮小”」を参照してください。
6. 従来のインデックステンプレート (_template) は、Elasticsearch 7.8 で開始する組み合わせ可能なテンプレート (_index_template) によって置き換えられました。組み合わせ可能なテンプレートは、従来のテンプレートよりも優先されます。指定されたインデックスに一致する組み合わせ可能なテンプレートがない場合、従来のテンプレートは引き続き一致して、適用できます。_template オペレーションは引き続き OpenSearch 以降のバージョンの Elasticsearch OSS で機能しますが、2 つのテンプレートタイプへの GET 呼び出しでは異なる結果が返されます。

Elasticsearch バージョン 7.8

Elasticsearch 7.8 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。
6. 従来のインデックステンプレート (_template) は、Elasticsearch 7.8 で開始する組み合わせ可能なテンプレート (_index_template) によって置き換えられました。組み合わせ可能なテンプレートは、従来のテンプレートよりも優先されます。指定されたインデックスに一致する組み合わせ可能なテンプレートがない場合、従来のテンプレートは引き続き一致して、適用できます。_template オペレーションは引き続き OpenSearch 以降のバージョンの Elasticsearch OSS で機能しますが、2 つのテンプレートタイプへの GET 呼び出しでは異なる結果が返されます。

Elasticsearch バージョン 7.7

Elasticsearch 7.7 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

• cluster.max_shards_per_node		

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 7.4

Elasticsearch 7.4 では、 OpenSearch Service は次のオペレーションをサポートしています。

• インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge 、*/index-name* /update/*id*、および */index-name* /_close など) • /_alias • /_aliases • /_all • /_analyze • /_bulk	• /_cluster/state • /_cluster/stats • /_count • /_delete_by_query [1] • /_explain • /_field_caps • /_field_stats • /_flush • /_ingest/pipeline • /_mapping	• /_refresh • /_reindex [1] • /_render • /_rollover • /_scripts [3] • /_search[2] • /_search profile • /_shard_stores • /_shrink[5] • /_snapshot

- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般

的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。

5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 7.1

Elasticsearch 7.1 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.8

Elasticsearch 6.8 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile

- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
  - cluster.blocks.read_only
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.7

Elasticsearch 6.7 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit
- cluster.max_shards_per_node

- /_plugin/kibana
- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.5

Elasticsearch 6.5 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および

- /_cluster/state
- /_cluster/stats
- /_count

- /_refresh
- /_reindex [1]
- /_render

/*index-name* /update/*id* など)、/*index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデ

フォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.4

Elasticsearch 6.4 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.3

Elasticsearch 6.3 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および

- /_cluster/state
- /_cluster/stats
- /_count

- /_refresh
- /_reindex [1]
- /_render

/*index-name* /update/*id* など)、/*index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデ

フォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.2

Elasticsearch 6.2 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

- /_rank_eval

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 6.0

Elasticsearch 6.0 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および
- /_cluster/state
- /_cluster/stats
- /_count
- /_render
- /_rollover
- /_scripts [3]

/*index-name* /update/*id* など)、/*index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデ

フォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 5.6

Elasticsearch 5.6 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [4]:
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 5.5

Elasticsearch 5.5 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および
- /_cluster/state
- /_cluster/stats
- /_count
- /_render
- /_rollover
- /_scripts [3]

/*index-name* /update/*id* など)、/*index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit

- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]

- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデ

フォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. スクリプトの使用に関する考慮事項については、「the section called “サポートされている他のリソース”」を参照してください。
4. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
5. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 5.3

Elasticsearch 5.3 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /update/*id* など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ [3]:
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_search [2]
- /_search profile
- /_shard_stores
- /_shrink [4]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- action.auto_create_index
- action.search.shard_count.limit
- indices.breaker.fielddata.limit
- indices.breaker.request.limit
- indices.breaker.total.limit

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデフォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。
3. PUT メソッドを参照してください。GET メソッドの詳細については、「the section called “API の重要な相違点”」を参照してください。このリストは、 OpenSearch Service がサポートする一般的な Elasticsearch オペレーションのみを参照し、異常検出、ISM などのプラグイン固有のサポートされているオペレーションは含まれません。
4. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 5.1

Elasticsearch 5.1 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (/*index-name* /_forcemerge および /*index-name* /update/*id* な
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_render
- /_rollover
- /_search[2]
- /_search profile
- /_shard_stores

ど)、/*index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings を使用する複数のプロパティ (PUT のみ):
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_shrink[3]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. クラスター設定を変更すると、これらの操作が完了する前に中断される可能性があります。リクエストが正常に完了したことを確認するには、これらの操作とともに /_tasks 操作を使用することをお勧めします。
2. メッセージ本文の /_search/scroll に対する DELETE リクエストでは、"Content-Length" を HTTP ヘッダーに指定する必要があります。ほとんどのクライアントでは、このヘッダーがデ

フォルトで追加されます。scroll_id 値の=文字に関する問題を回避するには、クエリ文字列ではなくリクエスト本文を使用して、scroll_id値を OpenSearch サービスに渡します。

3. 「the section called “縮小”」を参照してください。

Elasticsearch バージョン 2.3

Elasticsearch 2.3 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション (*/index-name* /_forcemerge および */index-name* /_recovery など)、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear (インデックスのみ)
- /_cat (/_cat/nodeattrs を除く)
- /_cluster/health
- /_cluster/settings を使用する複数のプロパティ (PUT のみ):
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
  - threadpool.percolate.queue_size
  - threadpool.search.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_render
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

- threadpool.suggest.queue_size

Elasticsearch バージョン 1.5

Elasticsearch 1.5 では、 OpenSearch Service は次のオペレーションをサポートしています。

- インデックスパス内のすべてのオペレーション、*/index-name* /_optimize および */index-name* /_warmer など、*/index-name* /_close を除く
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat
- /_cluster/health
- /_cluster/settings を使用する複数のプロパティ (PUT のみ):
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - threadpool.get.queue_size
  - threadpool.bulk.queue_size
  - threadpool.index.queue_size
  - threadpool.percolate.queue_size
  - threadpool.search.queue_size
  - threadpool.suggest.queue_size
- /_cluster/stats
- /_count
- /_flush
- /_mapping
- /_mget
- /_msearch
- /_nodes
- /_percolate
- /_plugin/kibana
- /_plugin/kibana3
- /_plugin/migration
- /_refresh
- /_search
- /_snapshot
- /_stats
- /_status
- /_template

Amazon OpenSearch サービスクォータ

AWS アカウントには、以前は制限と呼ばれていたデフォルトクォータが各サービスに設定されています。 AWS 特に明記されていない限り、クォータは地域固有です。

OpenSearch サービスドメインとインスタンス、Amazon OpenSearch Serverless、Amazon OpenSearch Ingestion のクォータを確認するには、の「Amazon OpenSearch サービスクォータ」を参照してください。AWS 全般のリファレンス

サービスのクォータをで表示するには AWS Management Console、 OpenSearch サービService Quotas コンソールを開きます。ナビゲーションペインで [AWS サービス] を選択し、[Amazon OpenSearch Service] を選択します。クォータの引き上げをリクエストするには、「Service Quotas ユーザーガイド」の「クォータ引き上げリクエスト」を参照してください。

トピック

UltraWarm ストレージクォータ

次の表は、 UltraWarm インスタンスタイプと、各タイプが使用できる最大ストレージ容量を示しています。の詳細については UltraWarm、「」を参照してくださいthe section called “UltraWarm ストレージ”。

インスタンスタイプ	最大ストレージ
`ultrawarm1.medium.search`	1.5 TiB
`ultrawarm1.large.search`	20 TiB

EBS ボリュームサイズのクォータ

次の表は、 OpenSearch Service がサポートする各インスタンスタイプの EBS ボリュームの最小サイズと最大サイズを示しています。どのインスタンスタイプにインスタンスストレージやその他のハードウェア詳細が含まれるかについては、「Amazon OpenSearch Service の料金表」を参照してください。

- ドメインの作成時に EBS ボリュームタイプの下で磁気ストレージを選択した場合、最大ボリュームサイズは `t2.small` および `t2.medium` を除くすべてのインスタンスタイプ、および磁気ストレージをサポートしていないすべての Graviton インスタンス (M6g、C6g、R6g、R6gd)) で 100 GiB になります。次のテーブルに示されている最大サイズを使用する場合は、いずれかの SSD オプションを選択してください。
- 一部の旧世代のインスタンスタイプにはインスタンスストレージが含まれますが、EBS ストレージをサポートしています。これらのインスタンスタイプの 1 つに対して EBS ストレージを選択した場合、ストレージボリュームは加算されません。EBS ボリュームまたはインスタンスストレージを使用できますが、両方を使用することはできません。

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
`t2.micro.search`	10 GiB	35 GiB	該当なし
`t2.small.search`	10 GiB	35 GiB	該当なし
`t2.medium.search`	10 GiB	35 GiB	該当なし
`t3.small.search`	10 GiB	100 GiB	100 GiB
`t3.medium.search`	10 GiB	200 GiB	200 GiB
`m3.medium.search`	10 GiB	100 GiB	該当なし
`m3.large.search`	10 GiB	512 GiB	該当なし
`m3.xlarge.search`	10 GiB	512 GiB	該当なし
`m3.2xlarge.search`	10 GiB	512 GiB	該当なし

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
m4.large.search	10 GiB	512 GiB	該当なし
m4.xlarge.search	10 GiB	1 TiB	該当なし
m4.2xlarge.search	10 GiB	1.5 TiB	該当なし
m4.4xlarge.search	10 GiB	1.5 TiB	該当なし
m4.10xlarge.search	10 GiB	1.5 TiB	該当なし
m5.large.search	10 GiB	512 GiB	1 TiB
m5.xlarge.search	10 GiB	1 TiB	2 TiB
m5.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m5.4xlarge.search	10 GiB	3 TiB	6 TiB
m5.12xlarge.search	10 GiB	9 TiB	18 TiB
m6g.large.search	10 GiB	512 GiB	1 TiB
m6g.xlarge.search	10 GiB	1 TiB	2 TiB
m6g.2xlarge.search	10 GiB	1.5 TiB	3 TiB
m6g.4xlarge.search	10 GiB	3 TiB	6 TiB
m6g.8xlarge.search	10 GiB	6 TiB	12 TiB
m6g.12xlarge.search	10 GiB	9 TiB	18 TiB
c4.large.search	10 GiB	100 GiB	該当なし
c4.xlarge.search	10 GiB	512 GiB	該当なし
c4.2xlarge.search	10 GiB	1 TiB	該当なし
c4.4xlarge.search	10 GiB	1.5 TiB	該当なし

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
c4.8xlarge.search	10 GiB	1.5 TiB	該当なし
c5.large.search	10 GiB	256 GiB	256 GiB
c5.xlarge.search	10 GiB	512 GiB	512 GiB
c5.2xlarge.search	10 GiB	1 TiB	1 TiB
c5.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c5.9xlarge.search	10 GiB	3.5 TiB	3.5 TiB
c5.18xlarge.search	10 GiB	7 TiB	7 TiB
c6g.large.search	10 GiB	256 GiB	256 GiB
c6g.xlarge.search	10 GiB	512 GiB	512 GiB
c6g.2xlarge.search	10 GiB	1 TiB	1 TiB
c6g.4xlarge.search	10 GiB	1.5 TiB	1.5 TiB
c6g.8xlarge.search	10 GiB	3 TiB	3 TiB
c6g.12xlarge.search	10 GiB	4.5 TiB	4.5 TiB
r3.large.search	10 GiB	512 GiB	該当なし
r3.xlarge.search	10 GiB	512 GiB	該当なし
r3.2xlarge.search	10 GiB	512 GiB	該当なし
r3.4xlarge.search	10 GiB	512 GiB	該当なし
r3.8xlarge.search	10 GiB	512 GiB	該当なし
r4.large.search	10 GiB	1 TiB	該当なし
r4.xlarge.search	10 GiB	1.5 TiB	該当なし

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
r4.2xlarge.search	10 GiB	1.5 TiB	該当なし
r4.4xlarge.search	10 GiB	1.5 TiB	該当なし
r4.8xlarge.search	10 GiB	1.5 TiB	該当なし
r4.16xlarge.search	10 GiB	1.5 TiB	該当なし
r5.large.search	10 GiB	1 TiB	2 TiB
r5.xlarge.search	10 GiB	1.5 TiB	3 TiB
r5.2xlarge.search	10 GiB	3 TiB	6 TiB
r5.4xlarge.search	10 GiB	6 TiB	12 TiB
r5.12xlarge.search	10 GiB	12 TiB	24 TiB
r6g.large.search	10 GiB	1 TiB	2 TiB
r6g.xlarge.search	10 GiB	1.5 TiB	3 TiB
r6g.2xlarge.search	10 GiB	3 TiB	6 TiB
r6g.4xlarge.search	10 GiB	6 TiB	12 TiB
r6g.8xlarge.search	10 GiB	8 TiB	16 TiB
r6g.12xlarge.search	10 GiB	12 TiB	24 TiB
r6gd.large.search	該当なし	該当なし	該当なし
r6gd.xlarge.search	該当なし	該当なし	該当なし
r6gd.2xlarge.search	該当なし	該当なし	該当なし
r6gd.4xlarge.search	該当なし	該当なし	該当なし
r6gd.8xlarge.search	該当なし	該当なし	該当なし

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
r6gd.12xlarge.search	該当なし	該当なし	該当なし
r6gd.16xlarge.search	該当なし	該当なし	該当なし
i2.xlarge.search	10 GiB	512 GiB	該当なし
i2.2xlarge.search	10 GiB	512 GiB	該当なし
i3.large.search	該当なし	該当なし	該当なし
i3.xlarge.search	該当なし	該当なし	該当なし
i3.2xlarge.search	該当なし	該当なし	該当なし
i3.4xlarge.search	該当なし	該当なし	該当なし
i3.8xlarge.search	該当なし	該当なし	該当なし
i3.16xlarge.search	該当なし	該当なし	該当なし
or1.medium.search	20 GiB	該当なし	768 GiB
or1.large.search	20 GiB	該当なし	1532 GiB
or1.xlarge.search	20 GiB	該当なし	3 TiB
or1.2xlarge.search	20 GiB	該当なし	6 TiB
or1.4xlarge.search	20 GiB	該当なし	12 TiB
or1.8xlarge.search	20 GiB	該当なし	16 TiB
or1.12xlarge.search	20 GiB	該当なし	24 TiB
or1.16xlarge.search	20 GiB	該当なし	36 TiB
im4gn.large.search	該当なし	該当なし	該当なし
im4gn.xlarge.search	該当なし	該当なし	該当なし

インスタンスタイプ	最小 EBS サイズ	最大 EBS サイズ (gp2)	最大 EBS サイズ (gp3)
im4gn.2xlarge.search	該当なし	該当なし	該当なし
im4gn.4xlarge.search	該当なし	該当なし	該当なし
im4gn.8xlarge.search	該当なし	該当なし	該当なし
im4gn.16xlarge.search	該当なし	該当なし	該当なし

ネットワークのクォータ

次の表は、HTTP リクエストペイロードの最大サイズを示しています。

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
t2.micro.search	10 MiB
t2.small.search	10 MiB
t2.medium.search	10 MiB
t3.small.search	10 MiB
t3.medium.search	10 MiB
m3.medium.search	10 MiB
m3.large.search	10 MiB
m3.xlarge.search	100 MiB
m3.2xlarge.search	100 MiB
m4.large.search	10 MiB
m4.xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
m4.2xlarge.search	100 MiB
m4.4xlarge.search	100 MiB
m4.10xlarge.search	100 MiB
m5.large.search	10 MiB
m5.xlarge.search	100 MiB
m5.2xlarge.search	100 MiB
m5.4xlarge.search	100 MiB
m5.12xlarge.search	100 MiB
m6g.large.search	10 MiB
m6g.xlarge.search	100 MiB
m6g.2xlarge.search	100 MiB
m6g.4xlarge.search	100 MiB
m6g.8xlarge.search	100 MiB
m6g.12xlarge.search	100 MiB
c4.large.search	10 MiB
c4.xlarge.search	100 MiB
c4.2xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
c4.4xlarge.search	100 MiB
c4.8xlarge.search	100 MiB
c5.large.search	10 MiB
c5.xlarge.search	100 MiB
c5.2xlarge.search	100 MiB
c5.4xlarge.search	100 MiB
c5.9xlarge.search	100 MiB
c5.18xlarge.search	100 MiB
c6g.large.search	10 MiB
c6g.xlarge.search	100 MiB
c6g.2xlarge.search	100 MiB
c6g.4xlarge.search	100 MiB
c6g.8xlarge.search	100 MiB
c6g.12xlarge.search	100 MiB
r3.large.search	10 MiB
r3.xlarge.search	100 MiB
r3.2xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
r3.4xlarge.search	100 MiB
r3.8xlarge.search	100 MiB
r4.large.search	100 MiB
r4.xlarge.search	100 MiB
r4.2xlarge.search	100 MiB
r4.4xlarge.search	100 MiB
r4.8xlarge.search	100 MiB
r4.16xlarge.search	100 MiB
r5.large.search	100 MiB
r5.xlarge.search	100 MiB
r5.2xlarge.search	100 MiB
r5.4xlarge.search	100 MiB
r5.12xlarge.search	100 MiB
r6g.large.search	100 MiB
r6g.xlarge.search	100 MiB
r6g.2xlarge.search	100 MiB
r6g.4xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
r6g.8xlarge.search	100 MiB
r6g.12xlarge.search	100 MiB
r6gd.large.search	100 MiB
r6gd.xlarge.search	100 MiB
r6gd.2xlarge.search	100 MiB
r6gd.4xlarge.search	100 MiB
r6gd.8xlarge.search	100 MiB
r6gd.12xlarge.search	100 MiB
r6gd.16xlarge.search	100 MiB
i2.xlarge.search	100 MiB
i2.2xlarge.search	100 MiB
i3.large.search	100 MiB
i3.xlarge.search	100 MiB
i3.2xlarge.search	100 MiB
i3.4xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
i3.8xlarge.search	100 MiB
i3.16xlarge.search	100 MiB
or1.medium.search	10 MiB
or1.large.search	100 MiB
or1.xlarge.search	100 MiB
or1.2xlarge.search	100 MiB
or1.4xlarge.search	100 MiB
or1.8xlarge.search	100 MiB
or1.12xlarge.search	100 MiB
or1.16xlarge.search	100 MiB
im4gn.large.search	100 MiB
im4gn.xlarge.search	100 MiB
im4gn.2xlarge.search	100 MiB
im4gn.4xlarge.search	100 MiB

インスタンスタイプ	HTTP リクエストペイロードの最大サイズ
im4gn.8xlarge.search	100 MiB
im4gn.16xlarge.search	100 MiB

シャードサイズのクォータ

次のセクションでは、さまざまなインスタンスファミリーの最大シャードサイズを示します。

インスタンスタイプ	Multi-AZ without Standby	Multi-AZ with Standby
R5、C5、M5	該当なし	65 GiB
I3	該当なし	65 GiB
R6g、C6g、M6g、R6gd	該当なし	65 GiB
OR1	100 GiB	65 GiB
Im4gn	該当なし	65 GiB

クォータの引き上げをリクエストするには、AWS サポートまでお問い合わせください。

Java プロセスのクォータ

OpenSearch サービスは Java プロセスをヒープサイズを 32 GiB に制限します。上級ユーザーは、フィールド データに使用されるヒープのパーセンテージを指定できます。詳細については、the section called “高度なクラスター設定”およびthe section called “JVM OutOfMemoryError”を参照してください。

ドメインポリシーのクォータ

OpenSearch サービスはドメインのアクセスポリシーを100 KiBに制限します。

Amazon OpenSearch Service のリザーブドインスタンス

Amazon OpenSearch Service のリザーブドインスタンス (RI) では、標準オンデマンドインスタンスと比べて大幅な割引を受けられます。インスタンス自体は同一です。RI は、アカウントでのオンデマンドインスタンスに適用される請求の割引にすぎません。使用状況が予想可能な存続時間の長いアプリケーションについては、RI により時間の経過と共に大幅なコスト削減が可能になります。

OpenSearch Service RI には 1 年または 3 年の期間が必要で、割引レートに影響を与える 3 つのお支払い方法があります。

- 前払いなし – 前払いはありません。期間内の時間ごとに割引された時間料金を支払います。
- 一部前払い – 料金の一部を前払いし、期間内の時間ごとに割引された時間料金を支払います。
- 全前払い – 料金を全額前払いします。期間内に時間料金は支払いません。

一般的に、前払い料金が多いほど割引率がなくなります。リザーブドインスタンスをキャンセルすることはできず (予約するときに、全期間に対して支払う契約を結びます)、前払い料金は払い戻しできません。

RI は柔軟ではなく、予約したインスタンスタイプにのみ適用されます。例えば、8 個の `c5.2xlarge.search` インスタンスのための予約は、16 個の `c5.xlarge.search` インスタンスまたは 4 個の `c5.4xlarge.search` インスタンスには適用されません。詳細については、「Amazon OpenSearch Service の料金」および「よくある質問」を参照してください。

トピック

- リザーブドインスタンスの購入 (コンソール)
- リザーブドインスタンスを購入する (AWS CLI)
- リザーブドインスタンスを購入する (AWS SDK)
- コストを確認する

リザーブドインスタンスの購入 (コンソール)

コンソールでは、既存のリザーブドインスタンスを表示したり、新しいリザーブドインスタンスを購入したりできます。

予約を購入するには

1. https://aws.amazon.com にアクセスし、[コンソールにサインイン] を選択します。

2. [分析] の下で、[Amazon OpenSearch Service] を選択します。
3. ナビゲーションペインで [リザーブドインスタンスのリース] を選択します。

   このページでは、既存の予約を見ることができます。多くの予約がある場合、フィルタリングして特定の予約を識別して見やすくすることができます。

   > **Tip**
   >
   > [リザーブドインスタンスのリース] リンクが表示されない場合は、AWS リージョンでドメインを作成します。

4. [リザーブドインスタンスの注文] を選択します。
5. 一意のわかりやすい名前を指定します。
6. インスタンスタイプ、サイズ、およびインスタンスの数を選択します。ガイダンスについては、「the section called “ドメインのサイジング”」を参照してください。
7. 期間の長さと支払いオプションを選択します。支払いの詳細を慎重に確認します。
8. [Next] を選択します。
9. 購入概要を慎重に確認します。リザーブドインスタンスの購入は払い戻しできません。
10. [注文] を選択します。

リザーブドインスタンスを購入する (AWS CLI)

AWS CLI には、提供タイプの表示、予約の購入、予約の表示を行うためのコマンドがあります。次のコマンドとサンプルレスポンスは、指定された AWS リージョン の提供タイプを示しています。

```
aws opensearch describe-reserved-instance-offerings --region us-east-1
{
  "ReservedInstanceOfferings": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
```

```
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
}
```

各戻り値の説明については、次の表を参照してください。

フィールド	説明
FixedPrice	予約の前払いコスト。
ReservedInstanceOfferingId	提供 ID。製品を予約する場合は、この値をメモしておきます。
RecurringCharges	予約の時間料金。
UsagePrice	レガシーフィールド。OpenSearch Service では、この値は常に 0 です。
PaymentOption	前払いなし、一部前払い、または全額前払い。
Duration	期間の長さ (秒)。 • 31536000 秒が 1 年です。 • 94608000 秒が 3 年です。
InstanceType	予約のインスタンスタイプ。各インスタンスタイプに割り当てられるハードウェアリソースについては、「Amazon OpenSearch Service の料金表」を参照してください。
CurrencyCode	FixedPrice と RecurringChargeAmount の通貨。

この次の例では、予約を購入します。

```
aws opensearch purchase-reserved-instance-offering --reserved-instance-offering-
id 1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a --reservation-name my-reservation --instance-
count 3 --region us-east-1
{
  "ReservationName": "my-reservation",
  "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
}
```

最後に、以下の例を使用して特定のリージョンの予約をリストできます。

```
aws opensearch describe-reserved-instances --region us-east-1
{
  "ReservedInstances": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "ReservationName": "my-reservation",
      "PaymentOption": "PARTIAL_UPFRONT",
      "UsagePrice": 0.0,
      "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "State": "payment-pending",
      "StartTime": 1522872571.229,
      "InstanceCount": 3,
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
}
```

> **Note**
>
> StartTime は Unix エポック時間 (1970 年 1 月 1 日の午前 00:00 UTC から経過した秒数)。たとえば、1522872571 エポック時間は 2018 年 4 月 4 日の 20:09:31 UTC です。オンラインコンバーターを使用することができます。

前の例で使用されているコマンドの詳細については、AWS CLI コマンドリファレンスを参照してください。

リザーブドインスタンスを購入する (AWS SDK)

AWS SDK (Android および iOS SDK を除く) は、「Amazon OpenSearch Service API リファレンス」に定義されているすべてのオペレーションをサポートします。これには、次も含まれます。

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

このサンプルスクリプトは、AWS SDK for Python (Boto3) の OpenSearchService 低レベル Python クライアントを使用して、リザーブドインスタンスを購入します。`instance_type` の値を指定する必要があります。

```
import boto3
from botocore.config import Config

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-east-1'
)

client = boto3.client('opensearch', config=my_config)

instance_type = '' # e.g. m4.2xlarge.search


def describe_RI_offerings(client):
    """Gets the Reserved Instance offerings for this account"""

    response = client.describe_reserved_instance_offerings()
    offerings = (response['ReservedInstanceOfferings'])
    return offerings
```

```
def check_instance(offering):
    """Returns True if instance type is the one you specified above"""

    if offering['InstanceType'] == instance_type:
        return True

    return False


def get_instance_id():
    """Iterates through the available offerings to find the ID of the one you
 specified"""

    instance_type_iterator = filter(
        check_instance, describe_RI_offerings(client))
    offering = list(instance_type_iterator)
    id = offering[0]['ReservedInstanceOfferingId']
    return id


def purchase_RI_offering(client):
    """Purchase Reserved Instances"""

    response = client.purchase_reserved_instance_offering(
        ReservedInstanceOfferingId = get_instance_id(),
        ReservationName = 'my-reservation',
        InstanceCount = 1
    )
    print('Purchased reserved instance offering of type ' + instance_type)
    print(response)


def main():
    """Purchase Reserved Instances"""
    purchase_RI_offering(client)
```

AWS SDK のインストールと使用の詳細については、「AWS Software Development Kits」を参照してください。

コストを確認する

Cost Explorer は、過去 13 か月間の使用量データを表示できる無料のツールです。このデータを分析すると、傾向を見分けて、RI がユースケースに合うかどうかを理解できます。既に RI がある場合、

[購入オプション] によってグループ分けし、償却コストを表示してその使用量とオンデマンドインスタンスの使用量を比較できます。また、使用状況の予算を設定して、予約を活用していることを確認することもできます。詳細については、AWS Billing ユーザーガイドの「Cost Explorer によるコストの分析」を参照してください。

Amazon OpenSearch Service でサポートされているその他のリソース

このトピックでは、Amazon OpenSearch Service がサポートする追加のリソースについて説明します。

bootstrap.memory_lock

OpenSearch サービスでは、 `bootstrap.memory_lock`で を有効にします。これによりopensearch.yml、JVM メモリが ロックされ、オペレーティングシステムがディスクにスワップするのを防ぎます。これは、以下を除いて、サポートされているすべてのインスタンスタイプに適用されます。

- `t2.micro.search`
- `t2.small.search`
- `t2.medium.search`
- `t3.small.search`
- `t3.medium.search`

スクリプト モジュール

OpenSearch サービスは、Elasticsearch 5.x 以降のドメインのスクリプティングをサポートしています。このサービスでは、1.5 または 2.3 用のスクリプティングがサポートされていません。

サポートされているスクリプトオプションには、以下が含まれます。

- Painless
- Lucene Expressions
- Mustache

Elasticsearch 5.5 以降のドメイン、およびすべての OpenSearch ドメインでは、 OpenSearch Service は `_scripts`エンドポイントを使用して保存されたスクリプトをサポートしま

す。Elasticsearch 5.3 および 5.1 ドメインでは、インラインスクリプトのみがサポートされています。

TLS トランスポート

OpenSearch サービスは、ポート 80 で HTTP をサポートし、ポート 443 で HTTPS をサポートしますが、TLS トランスポートはサポートしていません。

Amazon OpenSearch Service のチュートリアル

この章には、サービスへの移行、簡単な検索アプリケーションの構築、OpenSearch Dashboards での可視化の作成など、Amazon OpenSearch Service を使用するための全般的なチュートリアルがいくつか含まれています。

トピック

チュートリアル: Amazon OpenSearch Service でドキュメントを作成および検索する

このチュートリアルでは、Amazon OpenSearch Service でドキュメントを作成および検索する方法を説明します。データを JSON ドキュメント形式でインデックスに追加します。OpenSearch Service は、追加した最初のドキュメントに関するインデックスを作成します。

このチュートリアルでは、HTTP リクエストを実行してドキュメントを作成し、ドキュメントの ID を自動的に生成し、ドキュメントに対して基本的な検索および高度な検索を実行する方法を説明します。

> **Note**
>
> このチュートリアルでは、オープンアクセスを持つドメインを使用します。最高レベルのセキュリティを実現するため、仮想プライベートクラウド (VPC) 内にドメインを配置することをお勧めします。

前提条件

このチュートリアルには、次のような前提条件があります。

- AWS アカウント が必要です。
- アクティブな OpenSearch Service のドメインが必要です。

ドキュメントのインデックスへの追加

ドキュメントをインデックスに追加するために、Postman、cURL、または OpenSearch Dashboards コンソールなどの任意の HTTP ツールを使用できます。これらの例は、OpenSearch Dashboards でデベロッパーコンソールを使用していることを前提としています。別のツールを使用している場合は、必要に応じて完全な URL と認証情報を入力して、適宜調整してください。

ドキュメントをインデックスに追加するには

1. 使用しているドメインの OpenSearch Dashboards URL に移動します。この URL は、OpenSearch Service コンソールのドメインダッシュボードに表示されています。URL はこの形式に従います。

   ```
   domain-endpoint/_dashboards/
   ```

2. プライマリユーザー名とパスワードを使ってサインインします。
3. 左側のナビゲーションパネルを開き、[Dev Tools] (開発ツール) を選択します。
4. 新しいリソースを作成するための HTTP 動詞は PUT です。これは、新しいドキュメントとインデックスの作成に使用します。コンソールに次のコマンドを入力します。

   ```
   PUT fruit/_doc/1
   {
     "name":"strawberry",
     "color":"red"
   }
   ```

   PUT リクエストは、fruit という名前のインデックスを作成し、ID が 1 である単一のドキュメントをインデックスに追加します。次のレスポンスが生成されます。

   ```
   {
     "_index" : "fruit",
     "_type" : "_doc",
     "_id" : "1",
     "_version" : 1,
     "result" : "created",
     "_shards" : {
   ```

```
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

自動的に生成される ID の作成

OpenSearch Service は、ドキュメントの ID を自動的に生成できます。ID を生成するコマンドは、PUT リクエストの代わりに POST リクエストを使用し、(前のリクエストと比較して) ドキュメント ID を必要としません。

デベロッパーコンソールに次のリクエストを入力します。

```
POST veggies/_doc
{
  "name":"beet",
  "color":"red",
  "classification":"root"
}
```

このリクエストは、veggies という名前のインデックスを作成し、ドキュメントをインデックスに追加します。次のレスポンスが生成されます。

```
{
  "_index" : "veggies",
  "_type" : "_doc",
  "_id" : "3WgyS4IB5DLqbRIvLxtF",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

レスポンスの追加 _id フィールドに注意してください。これは、ID が自動的に作成されたことを示します。

> **Note**
>
> URL の _doc の後には何も入力しません。ここには通常 ID が挿入されます。生成される ID を使用してドキュメントを作成しようとしているため、まだ ID を入力しません。これは更新の際に行います。

POST コマンドを使用したドキュメントの更新

ドキュメントを更新するには、ID 番号を持つ HTTP POST コマンドを使用します。

最初に、ID が 42 のドキュメントを作成します。

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow"
}
```

その後、その ID を使用してドキュメントを更新します。

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow",
  "classification":"berries"
}
```

このコマンドは、ドキュメントを新しいフィールド classification で更新します。次のレスポンスが生成されます。

```
{
  "_index" : "fruits",
  "_type" : "_doc",
  "_id" : "42",
  "_version" : 2,
  "result" : "updated",
```

```
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 1,
  "_primary_term" : 1
}
```

> **Note**
>
> 存在しないドキュメントを更新しようとすると、OpenSearch Service はドキュメントを作成します。

一括アクションの実行

POST _bulk API オペレーションを使用して、1 回のリクエストで 1 つ以上のインデックスに対して複数のアクションを実行できます。一括アクションコマンドは、次の形式になります。

```
POST /_bulk
<action_meta>\n
<action_data>\n
<action_meta>\n
<action_data>\n
```

各アクションには 2 行の JSON が必要です。まず、アクションの説明またはメタデータを入力します。次の行で、データを入力します。各パートは改行 (\n) で区切られます。挿入のアクションの説明は次のようになります。

```
{ "create" : { "_index" : "veggies", "_type" : "_doc", "_id" : "7" } }
```

また、データを含む次の行は次のようになります。

```
{ "name":"kale", "color":"green", "classification":"leafy-green" }
```

メタデータとデータをまとめると、一括オペレーションでの 1 つのアクションが表されます。次のように、1 つのリクエストで多くのオペレーションを実行できます。

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "35" } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "36" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "37" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "38" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "39" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
{ "delete" : { "_index" : "vegetables", "_id" : "1" } }
```

最後のアクションが delete であることに注意してください。delete アクションに続くデータはありません。

ドキュメントの検索

これでクラスターにデータが存在するようになったので、そのデータを検索できます。例えば、すべての根菜類を検索したり、すべての葉菜類の数を取得したり、1 時間あたりに口グ記録された工ラーの数を調べたりすることができます。

基本的な検索

基本的な検索は次のようになります。

```
GET veggies/_search?q=name:l*
```

リクエストにより、レタスドキュメントを含む JSON レスポンスが生成されます。

高度な検索

リクエスト本文でクエリオプションを JSON として指定すると、より高度な検索を実行できます。

```
GET veggies/_search
{
  "query": {
    "term": {
      "name": "lettuce"
    }
```

```
  }
}
```

この例では、レタスドキュメントを含む JSON レスポンスも生成します。

ソート

ソートを使用すると、このタイプのクエリをさらに多く実行できます。まず、インデックスを再作成する必要があります。これは、自動フィールドマッピングが、デフォルトではソートできないタイプを選択するためです。次のリクエストを送信して、インデックスを削除して再作成します。

```
DELETE /veggies

PUT /veggies
{
   "mappings":{
      "properties":{
         "name":{
            "type":"keyword"
         },
         "color":{
            "type":"keyword"
         },
         "classification":{
            "type":"keyword"
         }
      }
   }
}
```

その後、インデックスにデータを再入力します。

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "7"  } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "8" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "9" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "10" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "11" } }
```

```
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
```

これで、ソートを使用して検索できるようになりました。このリクエストでは、分類による昇順のソートが追加されます。

```
GET veggies/_search
{
  "query" : {
    "term": { "color": "green" }
  },
  "sort" : [
      "classification"
  ]
}
```

関連リソース

詳細については、以下のリソースを参照してください。

- はじめに
- データのインデックス作成
- データの検索

チュートリアル: Amazon OpenSearch Service への移行

インデックススナップショットは、セルフマネージド OpenSearch または従来の Elasticsearch クラスターから Amazon OpenSearch Service に移行する一般的な方法です。そのプロセスは、大きく分けて以下のステップで構成されています。

1. 既存のクラスターのスナップショットを作成し、そのスナップショットを Amazon S3 バケットにアップロードする。
2. OpenSearch Service ドメインを作成する。
3. バケットにアクセスするための許可を OpenSearch Service に付与し、スナップショットを使用するための許可があることを確認する。
4. OpenSearch Service ドメインでスナップショットを復元する。

このチュートリアルでは、より詳細な手順と代替のオプション (ある場合) について説明します。

スナップショットの作成とアップロード

repository-s3 プラグインを使用すると、スナップショットを S3 に直接作成できます。ただし、このプラグインをすべてのノードにインストールし、opensearch.yml (または Elasticsearch クラスターを使用している場合は elasticsearch.yml) を設定し、各ノードを再起動し、AWS 認証情報を追加した上で、スナップショットを作成する必要があります。このプラグインは、継続使用や大規模なクラスターの移行の場合に役立ちます。

小規模なクラスターの場合は、1 回ごとに共有ファイルシステムのスナップショットを作成して、AWS CLI を使用して S3 にアップロードする方法があります。すでにスナップショットを作成している場合は、手順 4 に進んでください。

スナップショットを取り、Amazon S3 にアップロードするには

1. すべてのノードで opensearch.yml (または Elasticsearch.yml) に path.repo 設定を追加して、各ノードを再起動します。

```
path.repo: ["/my/shared/directory/snapshots"]
```

2. スナップショットを作成する前に必要な スナップショットリポジトリ を登録します。リポジトリは単なる保存場所です。共有ファイルシステム、Amazon S3、File system distribuito Hadoop (HDFS) などです。この場合、共有ファイルシステム (「fs」) を使用します。

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "fs",
  "settings": {
    "location": "/my/shared/directory/snapshots"
  }
}
```

3. スナップショットを作成します。

```
PUT _snapshot/my-snapshot-repo-name/my-snapshot-name
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

4. AWS CLI をインストールし、aws configure を実行して認証情報を追加します。

5. スナップショットのディレクトリに移動します。次のコマンドを実行して新しい S3 バケットを作成し、スナップショットのディレクトリの中身をそのバケットにアップロードします。

```
aws s3 mb s3://bucket-name --region us-west-2
aws s3 sync . s3://bucket-name --sse AES256
```

スナップショットのサイズとインターネット接続の速度によっては、この操作に時間がかかる場合があります。

ドメインの作成

コンソールはドメインを作成する最も簡単な方法です。この例では、すでにターミナルを開いており、AWS CLI がインストールされています。次のコマンドを変更して、ニーズに合わせてドメインを作成してください。

```
aws opensearch create-domain \
  --domain-name migration-domain \
  --engine-version OpenSearch_1.0 \
  --cluster-config InstanceType=c5.large.search,InstanceCount=2 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-user-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["*"]},"Action":
["es:ESHttp*"],"Resource":"arn:aws:es:us-west-2:123456789012:domain/migration-domain/
*"}]}' \
  --region us-west-2
```

このコマンドでは、それぞれ 100 GiB のストレージを持つ 2 つのデータノードがあるインターネットにアクセス可能なドメインが作成されます。また、HTTP Basic 認証とすべての暗号化の設定により、きめ細かなアクセスコントロールが可能になります。VPC などのアドバンストセキュリティ設定が必要な場合は、OpenSearch Service コンソールを使用してください。

このコマンドを発行する前に、ドメイン名、マスターユーザーの認証情報、アカウント番号を変更します。S3 バケットに使用したものと同じAWS リージョンと、スナップショットと互換性のある OpenSearch/Elasticsearch のバージョンを指定します。

> **Important**
>
> スナップショットには上位互換性のみがあり、その対象は 1 つのメジャーバージョンのみです。例えば、Elasticsearch 7.x クラスター上の OpenSearch 1 x クラスターからスナップショットを復元することはできず、OpenSearch 1.x または 2.x クラスターのみです。また、マイナーバージョンも同様です。セルフマネージド 5.3.3 クラスターのスナップショットを 5.3.2 の OpenSearch Service ドメインに復元することはできません。OpenSearch または Elasticsearch は、スナップショットがサポートしている最新バージョンを選択することをお勧めします。互換性のあるバージョンのテーブルについては、「the section called “スナップショットを使用してデータを移行する”」を参照してください。

S3 バケットへの許可を提供します。

AWS Identity and Access Management (IAM) コンソールで、以下の許可と信頼関係を持つロールを作成します。ロールの作成時に、AWS サービスとして S3 を選択します。このロールには、分かりやすいように `OpenSearchSnapshotRole` と名前を付けます。

アクセス許可

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
```

```
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

信頼関係

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

次に、個人的な IAM ロールに、OpenSearchSnapshotRole を引き受ける許可を付与します。以下のポリシーを作成して、アイデンティティにアタッチします。

アクセス許可

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

OpenSearch Dashboards にスナップショットロールをマッピングします (きめ細かなアクセスコントロールを使用している場合)

きめ細かなアクセスコントロールを有効にした場合は、HTTP 基本認証をその他すべての目的で使用する場合でも、manage_snapshots ロールを IAM ロールにマップして、スナップショットを使用できるようにする必要があります。

スナップショットを使用するためのアイデンティティの許可を提供するには

1. OpenSearch Service ドメインの作成時に指定したマスターユーザーの認証情報を使用して Dashboards にログインします。Dashboards URL は、OpenSearch Service コンソールに表示されます。`https://domain-endpoint/_dashboards/` の形式です。
2. メインメニューから [セキュリティ]、[ロール] を選択し、[manage_snapshots] ロールを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。
4. 適切なフィールドに、個人的な IAM ロールのドメイン ARN を追加します。ARN は次のいずれかの形式となります。

   ```
   arn:aws:iam::123456789123:user/user-name
   ```

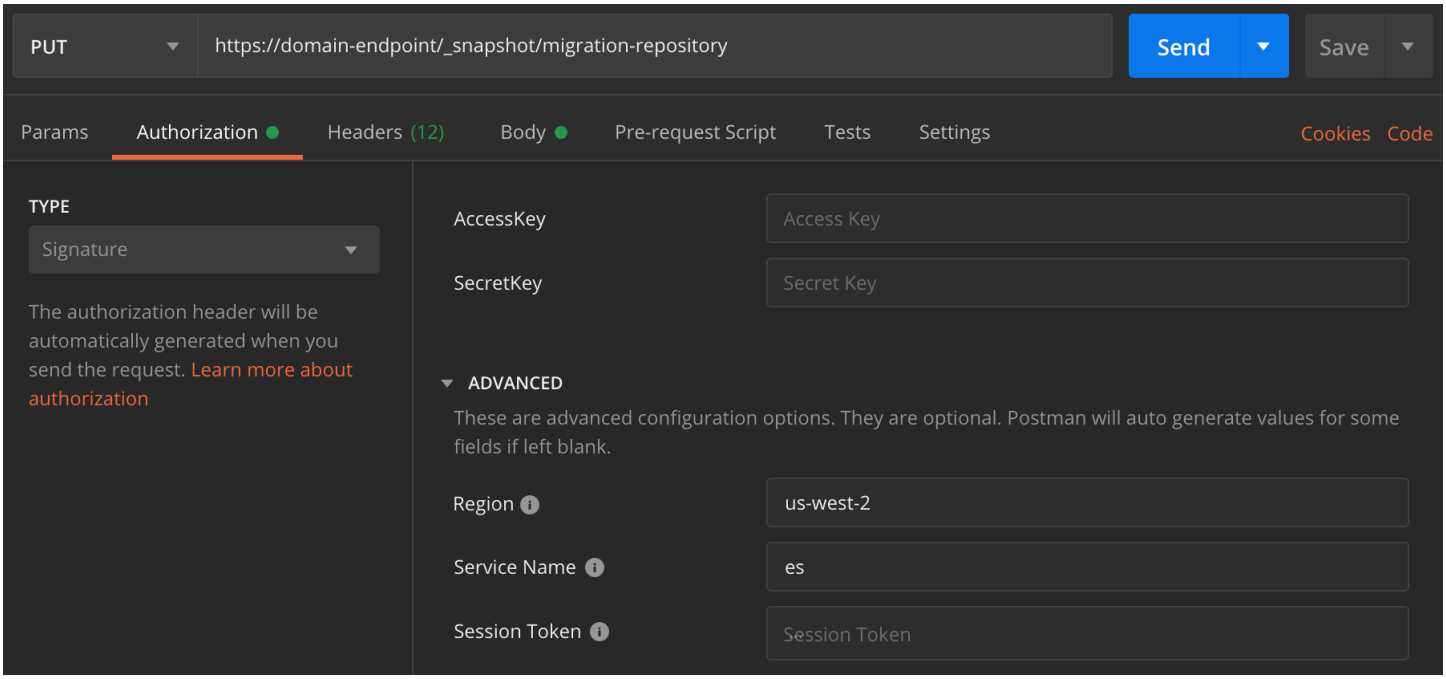
   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. マップを選択し、ロールがマッピングされたユーザーに表示されていることを確認します。

スナップショットを復元する

ここでは、OpenSearch Service ドメインにアクセスする方法が 2 つあります。1 つはマスターユーザーの認証情報を使用した HTTP Basic 認証で、もう 1 つは IAM 認証情報を使用した AWS 認証です。スナップショットはマスターユーザーの概念がない Amazon S3 を使用するため、IAM 認証情報を使用してスナップショットのリポジトリを OpenSearch Service ドメインに登録する必要があります。

ほとんどのプログラミング言語にはリクエストの署名に役立つライブラリがありますが、Postman などのツールを使用して IAM 認証情報を [認可] セクションに入力するのが簡単な方法です。



スナップショットを復元するには

1. リクエストにどのような方法で署名するかにかかわらず、まずリポジトリを登録します。

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "bucket-name",
    "region": "us-west-2",
    "role_arn": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
  }
}
```

2. 次に、リポジトリ内のスナップショットを一覧表示し、復元するスナップショットを見つけます。この時点で、Postman を続けて使用するか、curl などのツールに切り替えることができます。

   短縮構文

   ```
   GET _snapshot/my-snapshot-repo-name/_all
   ```

   curl

```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/_all
```

3. スナップショットを復元します。

   短縮構文

```
POST _snapshot/my-snapshot-repo-name/my-snapshot-name/_restore
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

   curl

```
curl -XPOST -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/my-snapshot-name/_restore \
  -H 'Content-Type: application/json' \
  -d '{"indices":"migration-index1,migration-index2,other-indices-
*","include_global_state":false}'
```

4. 最後に、インデックスが正常に復元されていることを検証します。

   短縮構文

```
GET _cat/indices?v
```

   curl

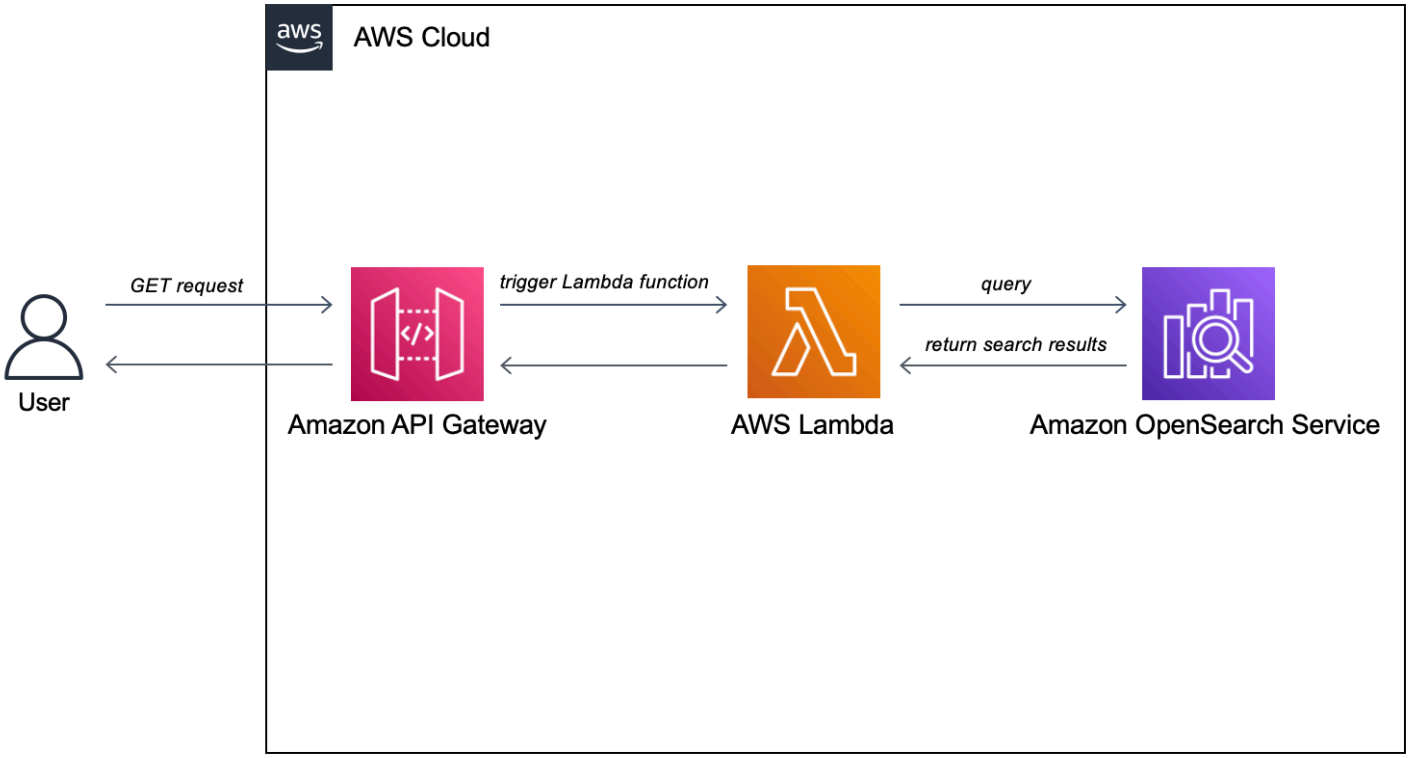
```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
indices?v
```

これで、移行は完了です。続いて、新しい OpenSearch Service エンドポイントを使用するようにクライアントを設定したり、ワークロードに合わせてドメインのサイズを変更したり、インデックスのシャード数を確認したり、IAM マスターユーザーに切り替えたり、OpenSearch Dashboards での可視化の構築を開始したりすることができます。

チュートリアル: Amazon OpenSearch Service を用いて検索アプリケーションを作成する

Amazon OpenSearch Service を用いて検索アプリケーションを作成する一般的な方法は、サーバーへのユーザークエリを送信するウェブフォームを使用することです。次に、OpenSearch API を直接呼び出すようサーバーを承認し、サーバーが OpenSearch Service にリクエストを送信するようにします。ただし、サーバーに依存しないクライアント側のコードを記述する場合は、セキュリティとパフォーマンスのリスクを補正する必要があります。OpenSearch API への、署名されていないパブリックアクセスを許可することはお勧めしません。ユーザーは、保護されていないエンドポイントにアクセスしたり、過度に広範なクエリ (または多すぎるクエリ) によりクラスターのパフォーマンスに影響を及ぼす可能性があります。

この章では、解決方法を提示します。Amazon API Gateway を使用してユーザーを OpenSearch API のサブセットに制限し、AWS Lambda を使用して API Gateway から OpenSearch Service へのリクエストに署名します。



> **Note**
>
> スタンダードの API Gateway および Lambda 料金表が適用されますが、このチュートリアル内での使用は限定されているため、費用はごくわずかです。

前提条件

このチュートリアルの前提条件は、OpenSearch Service ドメインです。まだドメインを持っていない場合は、「OpenSearch Service ドメインを作成する」のステップに従って、ドメインを作成します。

ステップ 1: サンプルデータのインデックス作成

sample-movies.zip をダウンロードして解凍します。その後 _bulk API オペレーションを使用して、movies インデックスに 5,000 個のドキュメントを追加します。

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/_bulk
{ "index": { "_index": "movies", "_id": "tt1979320" } }
{"directors":["Ron
 Howard"],"release_date":"2013-09-02T00:00:00Z","rating":8.3,"genres":
["Action","Biography","Drama","Sport"],"image_url":"http://ia.media-imdb.com/images/
M/MV5BMTQyMDE0MTY0OV5BMl5BanBnXkFtZTcwMjI2OTI0OQ@@._V1_SX400_.jpg","plot":"A re-
creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
 Niki Lauda.","title":"Rush","rank":2,"running_time_secs":7380,"actors":["Daniel
 Brühl","Chris Hemsworth","Olivia Wilde"],"year":2013,"id":"tt1979320","type":"add"}
{ "index": { "_index": "movies", "_id": "tt1951264" } }
{"directors":["Francis Lawrence"],"release_date":"2013-11-11T00:00:00Z","genres":
["Action","Adventure","Sci-Fi","Thriller"],"image_url":"http://ia.media-imdb.com/
images/M/
MV5BMTAyMjQ3OTAxMzNeQTJeQWpwZ15BbWU4MDU0NzA1MzAx._V1_SX400_.jpg","plot":"Katniss
 Everdeen and Peeta Mellark become targets of the Capitol after
 their victory in the 74th Hunger Games sparks a rebellion in
 the Districts of Panem.","title":"The Hunger Games: Catching
 Fire","rank":4,"running_time_secs":8760,"actors":["Jennifer Lawrence","Josh
 Hutcherson","Liam Hemsworth"],"year":2013,"id":"tt1951264","type":"add"}
...
```

上記は、利用可能なデータの、ごく一部のコマンド例であることにご注意ください。_bulk オペレーションを実行するには、sample-movies ファイルのコンテンツ全体をコピーして貼り付ける

必要があります。詳細な手順については、「the section called “オプション 2: 複数のドキュメントをアップロードする”」を参照してください。

次のように curl コマンドを使用しても同じ結果が得られます。

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-binary
 @bulk_movies.json -H 'Content-Type: application/json'
```

ステップ 2: Lambda 関数を作成してデプロイする

API Gateway で API を作成する前に、リクエストを渡す Lambda 関数を作成します。

Lambda 関数を作成する

このソリューションでは、API Gateway は Lambda 関数にリクエストを渡します。この関数は OpenSearch Service にクエリを送り、結果を返します。このサンプル関数は外部ライブラリを使用するため、デプロイパッケージを作成して Lambda にアップロードする必要があります。

デプロイパッケージを作成するには

1. コマンドプロンプトを開き、my-opensearch-function プロジェクトディレクトリを作成します。例えば、macOS では次のようになります。

   ```
   mkdir my-opensearch-function
   ```

2. my-sourcecode-function プロジェクトディレクトリに移動します。

   ```
   cd my-opensearch-function
   ```

3. 次のサンプル Python コードの内容をコピーし、opensearch-lambda.py という名前の新しいファイルに保存します。リージョンとホストエンドポイントをファイルに追加します。

   ```
   import boto3
   import json
   import requests
   from requests_aws4auth import AWS4Auth

   region = '' # For example, us-west-1
   service = 'es'
   credentials = boto3.Session().get_credentials()
   awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
    session_token=credentials.token)
   ```

```
host = '' # The OpenSearch domain endpoint with https:// and without a trailing
 slash
index = 'movies'
url = host + '/' + index + '/_search'

# Lambda execution starts here
def lambda_handler(event, context):

    # Put the user query into the query DSL for more accurate search results.
    # Note that certain fields are boosted (^).
    query = {
        "size": 25,
        "query": {
            "multi_match": {
                "query": event['queryStringParameters']['q'],
                "fields": ["title^4", "plot^2", "actors", "directors"]
            }
        }
    }

    # Elasticsearch 6.x requires an explicit Content-Type header
    headers = { "Content-Type": "application/json" }

    # Make the signed HTTP request
    r = requests.get(url, auth=awsauth, headers=headers, data=json.dumps(query))

    # Create the response and add some extra content to support CORS
    response = {
        "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": '*'
        },
        "isBase64Encoded": False
    }

    # Add the search results to the response
    response['body'] = r.text
    return response
```

4. 新しい package ディレクトリに外部のライブラリをインストールします。

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
```

```
pip3 install --target ./package requests_aws4auth
```

5. ルートにインストール済みライブラリを含むデプロイパッケージを作成します。次のコマンドを使ってプロジェクトディレクトリに `my-deployment-package.zip` ファイルを生成します。

```
cd package
zip -r ../my-deployment-package.zip .
```

6. zip ファイルのルートに `opensearch-lambda.py` ファイルを追加します。

```
cd ..
zip my-deployment-package.zip opensearch-lambda.py
```

Lambda 関数とデプロイパッケージの作成の詳細については、AWS Lambda デベロッパーガイドの「.zip ファイルアーカイブで Python Lambda 関数をデプロイする」と本ガイドの「the section called “Lambda デプロイパッケージを作成する”」を参照してください。

Lambda コンソールを使って関数を作成するには

1. Lambda コンソール (https://console.aws.amazon.com/lambda/home) に進みます。ナビゲーションペインで、[Functions] (関数) を選択します。
2. [Create Function] (関数を作成) を選択します。
3. 次のフィールドを設定します。

   - 関数名: opensearch-function
   - ランタイム: Python 3.9
   - アーキテクチャ: x86_64

   他のオプションはすべてデフォルトのままにして、[Create function] を選択します。
4. 関数の概要ページの[Code source] (コードソース) セクションで、ドロップダウンから [Upload from] (アップロード元) を選択し、[.zip file] (.zip ファイル) を選択します。`my-deployment-package.zip` 作成したファイルを見つけて、[Save] (保存) を選択します。
5. ハンドラーとは、イベントを処理する関数コード内のメソッドです。[Runtime settings] (ランタイム設定) の下で [Edit] (編集) を選択し、Lambda 関数が配置されているデプロイパッケージ内の、ファイルの名前に従ってハンドラー名を変更します。例えば、ファイルの名前が `opensearch-lambda.py` の場合、ハンドラー名を *`opensearch-lambda`*`.lambda_handler` に変更します。詳細については、「Python の Lambda 関数ハンドラー」を参照してください。

ステップ 3: API Gateway で API を作成する

API Gateway を使用してさらに制限された API を作成することで、OpenSearch `_search` API とのやり取りのプロセスを簡素化することができます。API Gateway により、Amazon Cognito 認証やリクエストスロットリングなどのセキュリティ機能を有効にすることもできます。API を作成してデプロイするには、以下のステップを実施します。

API の作成と設定

API Gateway コンソールを使用して API を作成するには

1. API Gateway コンソール (https://console.aws.amazon.com/apigateway/home) に移動します。ナビゲーションペインで、[API] を選択します。
2. [REST API] (プライベートではない) を見つけ、[構築] を選択します。
3. 次のページの [Create new API] (新しい API を作成) セクションで、[New API] (新規 API) が選択されていることを確認します。
4. 次のフィールドを設定します。
   - API 名: opensearch-api
   - 説明: Amazon OpenSearch Service ドメイン検索用のパブリック API
   - エンドポイントタイプ: リージョン別
5. API の作成 を選択します。
6. [アクション] および [メソッドの作成] を選択します。
7. ドロップダウンで [GET] を選択し、チェックマークをクリックして確定します。
8. 以下の設定を行い、[保存] を選択します。

設定	Value
統合タイプ	Lambda 関数
Lambda プロキシ統合の使用	はい
Lambda のリージョン	*us-west-1*
Lambda 関数	`opensearch-lambda`

設定	Value
デフォルトタイムアウトの使用	はい

メソッドリクエストの設定

[メソッドリクエスト] を選択して、以下の設定を行います。

設定	Value
認証	なし
リクエストの検証	クエリ文字列パラメータ、およびヘッダーの検証
API キーが必要です	false

[URL Query String Parameters] (URL クエリ文字列パラメータ) の下で [Add query string] (クエリ文字列を追加) を選択し、次のパラメータを設定します。

設定	Value
名前	q
必須	はい

API のデプロイとステージの設定

API Gateway コンソールでは、デプロイを作成して新規または既存のステージに関連付けることで API をデプロイできます。

1. [アクション] および [API のデプロイ] を選択します。
2. [デプロイステージ] では、[新しいステージ] を選択し、ステージ `opensearch-api-test` に名前を付けます。
3. [Deploy] (デプロイ) をクリックします。
4. ステージエディタで以下の設定を行い、[変更の保存] を選択します。

設定	Value
スロットリングの有効化	はい
Rate	1,000
バースト	500

これらの設定は、エンドポイントのルート (https://*some-id*.execute-api.*us-west-1*.amazonaws.com/search-es-api-test) への GET リクエストという 1 つのメソッドのみを持つ API を設定します。リクエストに必要なのは、検索するクエリ文字列という 1 つのパラメータ (q) です。呼び出されると、メソッドは opensearch-lambda 関数を実行する Lambda にリクエストを渡します。詳細については、「Amazon API Gateway での API の作成」および「Amazon API Gateway での REST API のデプロイ」を参照してください。

ステップ 4 (オプション): ドメインアクセスポリシーを変更する

OpenSearch Service ドメインで、Lambda 関数が movies インデックスへの GET リクエストを行うことを許可する必要があります。きめ細かなアクセスコントロールが有効になっているドメインで未処理のアクセスポリシーがある場合は、そのまま使用できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

または、所持しているドメインアクセスポリシーをよりきめ細かにすることもできます。例えば、次の最小限のポリシーは、movies インデックスへの opensearch-lambda-role (Lambda を通じて作成された) 読み取りアクセスを提供します。Lambda が自動的に作成するロールの正確な名前を

取得するには、AWS Identity and Access Management (IAM) コンソールに進み、ロールを選択し、「lambda」を検索します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/service-role/opensearch-lambda-role-1abcdefg"
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/movies/_search"
    }
  ]
}
```

> **Important**
>
> きめ細かなアクセスコントロールがドメインに対して有効になっている場合は、OpenSearch Dashboards でロールをユーザーにマッピングする必要もあります。マッピングを行わないと、アクセス許可エラーが表示されます。

アクセスポリシーの詳細については、「the section called “アクセスポリシーの設定”」を参照してください。

Lambda ロールをマッピングする (きめ細かなアクセスコントロールを使用している場合)

きめ細かなアクセスコントロールでは、アプリケーションをテストする前に追加のステップが導入されます。HTTP 基本認証を他のすべての目的で使用する場合でも、ユーザーに Lambda ロールをマッピングする必要があります。マッピングを行わないと、アクセス許可エラーが表示されます。

1. ドメインの OpenSearch Dashboards URL に移動します。
2. メインメニューから [Security] (セキュリティ)、[Roles] (ロール) の順に選択し、Lambda ロールをマッピングするロールである `all_access` へのリンクを選択します。
3. [マッピングされたユーザー]、[マッピングの管理] を選択します。

4. [Backend roles] (バックエンドロール) で、Lambda ロールの Amazon リソースネーム (ARN) を追加します。ARN は `arn:aws:iam::123456789123:role/service-role/opensearch-lambda-role-1abcdefg` の形式にします。
5. [マップ] を選択し、ユーザーまたはロールが [マッピングされたユーザー] の下に表示されていることを確認します。

ステップ 5: ウェブアプリケーションをテストする

ウェブアプリケーションをテストするには

1. sample-site.zip をダウンロードして解凍し、お気に入りのテキストエディタで `scripts/search.js` を開きます。
2. API Gateway エンドポイントを指すように `apigatewayendpoint` 変数を更新し、所定のパスの最後にバックスラッシュを追加します。[Stages] (ステージ) と API の名前を選択することで、API ゲートウェイでエンドポイントをすばやく見つけることができます。`apigatewayendpoint` 変数は `https://some-id.execute-api.us-west-1.amazonaws.com/opensearch-api-test/` の形式にします。
3. `index.html` を開き、thor、house、および他のいくつかの用語の検索を実行してみてください。

Movie Search

thor

Found 7 results.

Thor

2011 — The powerful but arrogant god Thor is cast out of Asgard to live amongst humans in Midgard (Earth), where he soon becomes one of their finest defenders.

Thor: The Dark World

2013 — Faced with an enemy that even Odin and Asgard cannot withstand, Thor must embark on his most perilous and personal journey yet, one that will reunite him with Jane Foster and force him to sacrifice everything to save us all.

Vikingdom

2013 — A forgotten king, Eirick, is tasked with the impossible odds to defeat Thor, the God of Thunder.

CORS エラーのトラブルシューティング

Lambda 関数で CORS をサポートするための応答にコンテンツが含まれていても、次のエラーが表示される場合があります。

```
Access to XMLHttpRequest at '<api-gateway-endpoint>' from origin 'null' has been
 blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present in the
 requested resource.
```

このような場合は、以下のことを試してみます。

1. GET リソースで CORS を有効化します。[Advanced] (アドバンスト) の下で、[Access-Control-Allow-Credentials] を `'true'` にセットします。
2. API ゲートウェイ ([Actions] (アクション)、[Deploy API] (API のデプロイ)) で API を再デプロイします。
3. Lambda 関数トリガーを削除して再追加します。[re-add] (再追加) を追加し、[Add trigger] (トリガーを追加) を選択して、関数を呼び出す HTTP エンドポイントを作成します。トリガーには、次の設定がある必要があります。

トリガー	API	デプロイされるステージ	セキュリティ
API Gateway	opensearch-api	opensearch-api-test	開く

次のステップ

この章は、概念を示すための出発点にすぎません。例えば、次のような変更が考えられます。

- OpenSearch Service ドメインに独自のデータを追加します。
- API にメソッドを追加します。
- Lambda 関数では、検索クエリを変更するか、異なるフィールドを追加します。
- 結果のスタイルを変更するか、`search.js` を変更してユーザーに異なるフィールドを表示します。

チュートリアル: OpenSearch Service と OpenSearch Dashboards によるカスタマーサポートへの問い合わせを可視化する

この章は、次のような状況の完全なチュートリアルです。ビジネスである程度の数のカスタマーサポートコールを受けており、それらを分析したいと考えています。各問い合わせの件名は何でしょうか? 肯定的なやり取りの数はいくつでしょうか? 否定的なやり取りの数はいくつでしょうか? マネージャーはこれらの問い合わせのトランスクリプトをどのように検索または確認することができますか?

手動ワークフローでは、従業員が通話記録を聴き、各問い合わせの件名をメモし、顧客とのやり取りが肯定的であったかどうかを判断することが考えられます。

このようなプロセスには非常に大きな労力がかかります。1 回の問い合わせの平均時間が 10 分とすると、各従業員が 1 日あたり聴くことができる問い合わせの数は 48 件にすぎません。人間の先入観や偏見を除外することで、生成されるデータは非常に正確なものとなる一方で、データの量は最小限になります。つまり、問い合わせの件名と、顧客が満足したかどうかのブール値のみとなります。完全なトランスクリプトなど、それ以上の内容を伴う場合、非常に長い時間がかかる可能性があります。

Amazon S3、Amazon Transcribe、Amazon Comprehend、および Amazon OpenSearch Service を使用することで、ごくわずかなコードで同様のプロセスを自動化し、はるかに多くのデータを得ることができます。例えば、問い合わせの完全なトランスクリプト、トランスクリプトからのキーワード、および問い合わせの全体的な「センチメント」(肯定的、否定的、中立、混在) を取得できます。次に、OpenSearch と OpenSearch Dashboards を使用して、データを検索して可視化できます。

このチュートリアルはそのとおりに使用できますが、その意図は、OpenSearch Service でインデックス化する前に、JSON ドキュメントを強化する方法に関するアイデアを生み出すことです。

推定コスト

一般的に、このチュートリアルのステップを実行するコストは 2 USD 未満です。このチュートリアルでは、以下のリソースを使用します。

- 転送および保存されるデータが 100 MB 未満である S3 バケット

  詳細については、「Amazon S3 料金表」を参照してください。
- 1 つの `t2.medium` インスタンスを持つ OpenSearch Service ドメインと、数時間に対応する 10 GiB の EBS ストレージ

  詳細については、「Amazon OpenSearch Service 料金表」を参照してください。

- Amazon Transcribe への数回の呼び出し

  詳細については、「Amazon Transcribe 料金表」を参照してください。
- Amazon Comprehend への数回の自然言語処理の呼び出し

  詳細については、「Amazon Comprehend 料金表」を参照してください。

トピック

ステップ 1: 前提条件を設定する

続行する前に、以下のリソースが必要です。

前提条件	説明
Amazon S3 バケット	詳細については、Amazon Simple Storage Service コンソールユーザーガイドの「バケットの作成」を参照してください。
OpenSearch Service ドメイン	データのコピー先。詳細については、「OpenSearch Service ドメインの作成」を参照してください。

これらのリソースがない場合は、次の AWS CLI コマンドを使用して作成できます。

```
aws s3 mb s3://my-transcribe-test --region us-west-2
```

```
aws opensearch create-domain --domain-name my-transcribe-test --engine-version
 OpenSearch_1.0 --cluster-config  InstanceType=t2.medium.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-
policies '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789012:root"},"Action":"es:*","Resource":"arn:aws:es:us-
west-2:123456789012:domain/my-transcribe-test/*"}]}' --region us-west-2
```

> **Note**
>
> これらのコマンドでは us-west-2 リージョンを使用しますが、Amazon Comprehend がサポートしている任意のリージョンを使用することもできます。詳細については、AWS 全般のリファレンス を参照してください。

ステップ 2: サンプルコードをコピーする

1. 次の Python 3 サンプルコードをコピーし、call-center.py という新しいファイルに貼り付けます。

```
import boto3
import datetime
import json
import requests
from requests_aws4auth import AWS4Auth
import time
import urllib.request

# Variables to update
audio_file_name = '' # For example, 000001.mp3
bucket_name = '' # For example, my-transcribe-test
domain = '' # For example, https://search-my-transcribe-test-12345.us-
west-2.es.amazonaws.com
index = 'support-calls'
type = '_doc'
region = 'us-west-2'

# Upload audio file to S3.
s3_client = boto3.client('s3')

audio_file = open(audio_file_name, 'rb')

print('Uploading ' + audio_file_name + '...')
response = s3_client.put_object(
    Body=audio_file,
    Bucket=bucket_name,
    Key=audio_file_name
)

# # Build the URL to the audio file on S3.
```

```
# # Only for the us-east-1 region.
# mp3_uri = 'https://' + bucket_name + '.s3.amazonaws.com/' + audio_file_name

# Get the necessary details and build the URL to the audio file on S3.
# For all other regions.
response = s3_client.get_bucket_location(
    Bucket=bucket_name
)
bucket_region = response['LocationConstraint']
mp3_uri = 'https://' + bucket_name + '.s3-' + bucket_region + '.amazonaws.com/' +
 audio_file_name

# Start transcription job.
transcribe_client = boto3.client('transcribe')

print('Starting transcription job...')
response = transcribe_client.start_transcription_job(
    TranscriptionJobName=audio_file_name,
    LanguageCode='en-US',
    MediaFormat='mp3',
    Media={
        'MediaFileUri': mp3_uri
    },
    Settings={
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 2 # assumes two people on a phone call
    }
)

# Wait for the transcription job to finish.
print('Waiting for job to complete...')
while True:
    response =
 transcribe_client.get_transcription_job(TranscriptionJobName=audio_file_name)
    if response['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    else:
        print('Still waiting...')
    time.sleep(10)

transcript_uri = response['TranscriptionJob']['Transcript']['TranscriptFileUri']

# Open the JSON file, read it, and get the transcript.
```

```
response = urllib.request.urlopen(transcript_uri)
raw_json = response.read()
loaded_json = json.loads(raw_json)
transcript = loaded_json['results']['transcripts'][0]['transcript']

# Send transcript to Comprehend for key phrases and sentiment.
comprehend_client = boto3.client('comprehend')

# If necessary, trim the transcript.
# If the transcript is more than 5 KB, the Comprehend calls fail.
if len(transcript) > 5000:
    trimmed_transcript = transcript[:5000]
else:
    trimmed_transcript = transcript

print('Detecting key phrases...')
response = comprehend_client.detect_key_phrases(
    Text=trimmed_transcript,
    LanguageCode='en'
)

keywords = []
for keyword in response['KeyPhrases']:
    keywords.append(keyword['Text'])

print('Detecting sentiment...')
response = comprehend_client.detect_sentiment(
    Text=trimmed_transcript,
    LanguageCode='en'
)

sentiment = response['Sentiment']

# Build the Amazon OpenSearch Service URL.
id = audio_file_name.strip('.mp3')
url = domain + '/' + index + '/' + type + '/' + id

# Create the JSON document.
json_document = {'transcript': transcript, 'keywords': keywords, 'sentiment':
 sentiment, 'timestamp': datetime.datetime.now().isoformat()}

# Provide all details necessary to sign the indexing request.
credentials = boto3.Session().get_credentials()
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region,
 'opensearchservice', session_token=credentials.token)

# Index the document.
print('Indexing document...')
response = requests.put(url, auth=awsauth, json=json_document, headers=headers)

print(response)
print(response.json())
```

2. 最初の 6 つの変数を更新します。
3. 次のコマンドを使用して必要なパッケージをインストールします。

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. call-center.py と同じディレクトリに MP3 を配置し、スクリプトを実行します。サンプル出力を次に示します。

```
$ python call-center.py
Uploading 000001.mp3...
Starting transcription job...
Waiting for job to complete...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Detecting key phrases...
Detecting sentiment...
Indexing document...
<Response [201]>
{u'_type': u'call', u'_seq_no': 0, u'_shards': {u'successful': 1, u'failed': 0,
 u'total': 2}, u'_index': u'support-calls4', u'_version': 1, u'_primary_term': 1,
 u'result': u'created', u'_id': u'000001'}
```

call-center.py は多数のオペレーションを実行します。

1. このスクリプトは、音声ファイル (この例では MP3 ですが、Amazon Transcribe は複数の形式をサポートします) を S3 バケットにアップロードします。
2. オーディオファイルの URL を Amazon Transcribe に送信し、書き起こしジョブの完了を待機します。

   書き起こしジョブが完了するまでの時間は、オーディオファイルの長さによって異なります。数秒ではなく、数分であると想定してください。

   > **Tip**
   >
   > 書き起こしの品質を向上させるため、Amazon Transcribe 用のカスタム語彙を設定できます。

3. 書き起こしジョブが完了すると、スクリプトはトランスクリプトを抽出し、それを 5,000 文字に切り捨て、キーワードとセンチメントの分析のため Amazon Comprehend に送信します。
4. 最後に、スクリプトは完全なトランスクリプト、キーワード、センチメント、および現在のタイムスタンプを JSON ドキュメントに追加し、そのインデックスを OpenSearch Service に作成します。

> **Tip**
>
> LibriVox には、テストに使用できるパブリックドメインオーディオブックがあります。

(オプション) ステップ 3: サンプルデータのインデックスを作成する

多くの通話記録が手元にない場合 (これは一般的です)、sample-calls.zip にサンプルドキュメントのインデックスを作成できます。これは `call-center.py` で生成されるものと同等です。

1. `bulk-helper.py` という名前のファイルを作成します。

```
import boto3
from opensearchpy import OpenSearch, RequestsHttpConnection
import json
from requests_aws4auth import AWS4Auth

host = '' # For example, my-test-domain.us-west-2.es.amazonaws.com
region = '' # For example, us-west-2
```

```
service = 'es'

bulk_file = open('sample-calls.bulk', 'r').read()

credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

response = search.bulk(bulk_file)
print(json.dumps(response, indent=2, sort_keys=True))
```

2. host および region の最初の 2 つの変数を更新します。
3. 次のコマンドを使用して必要なパッケージをインストールします。

```
pip install opensearch-py
```

4. sample-calls.zip をダウンロードして解凍します。
5. sample-calls.bulk と同じディレクトリに bulk-helper.py を配置し、ヘルパーを実行します。サンプル出力を次に示します。

```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "support-calls",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
```

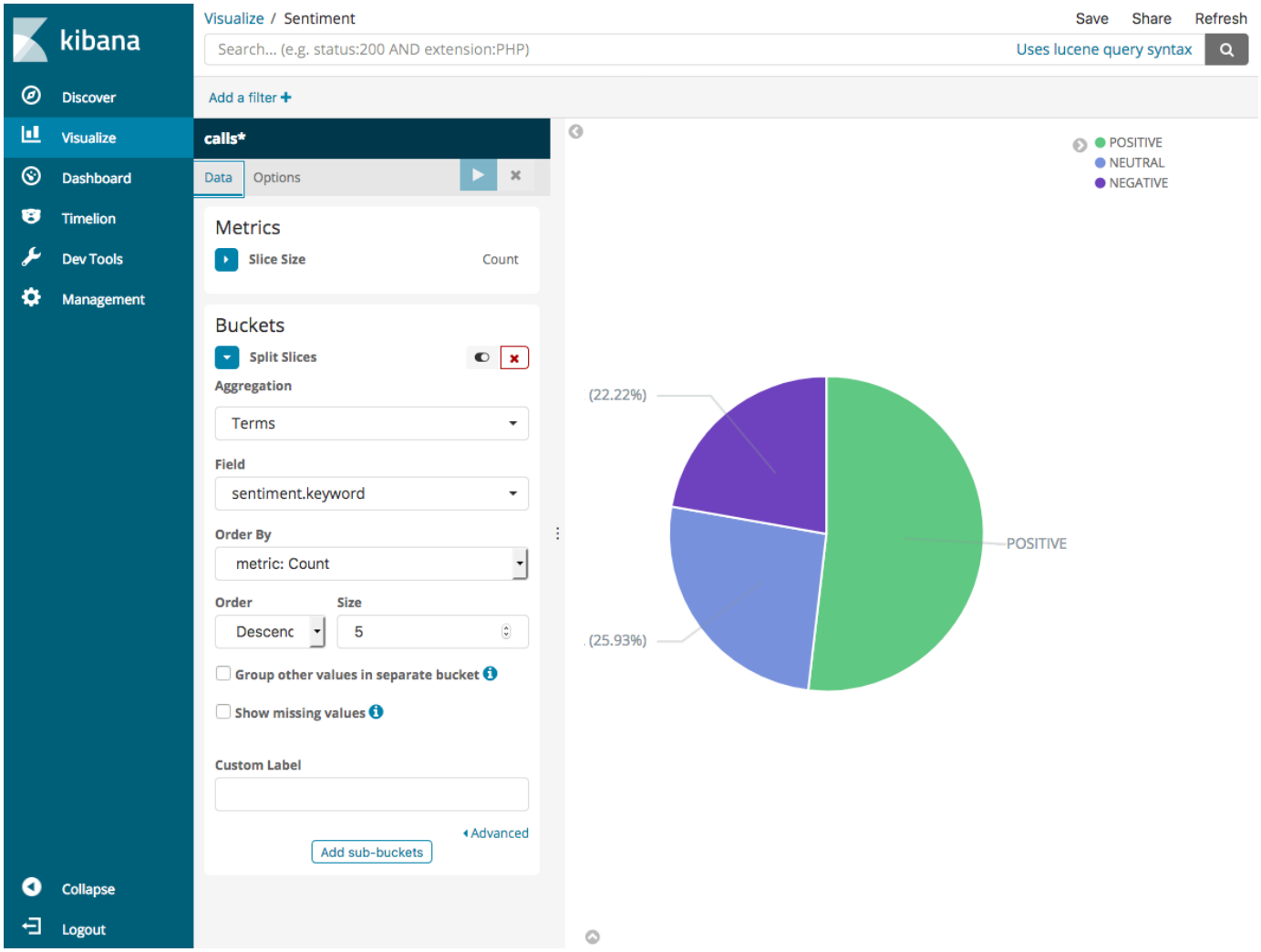
```
        },
        "_type": "_doc",
        "_version": 9,
        "result": "updated",
        "status": 200
      }
    },
    ...
  ],
  "took": 27
}
```

ステップ 4: データを分析し、可視化する

OpenSearch Service にデータを配置したので、OpenSearch Dashboards を使用してそのデータを可視化することができます。

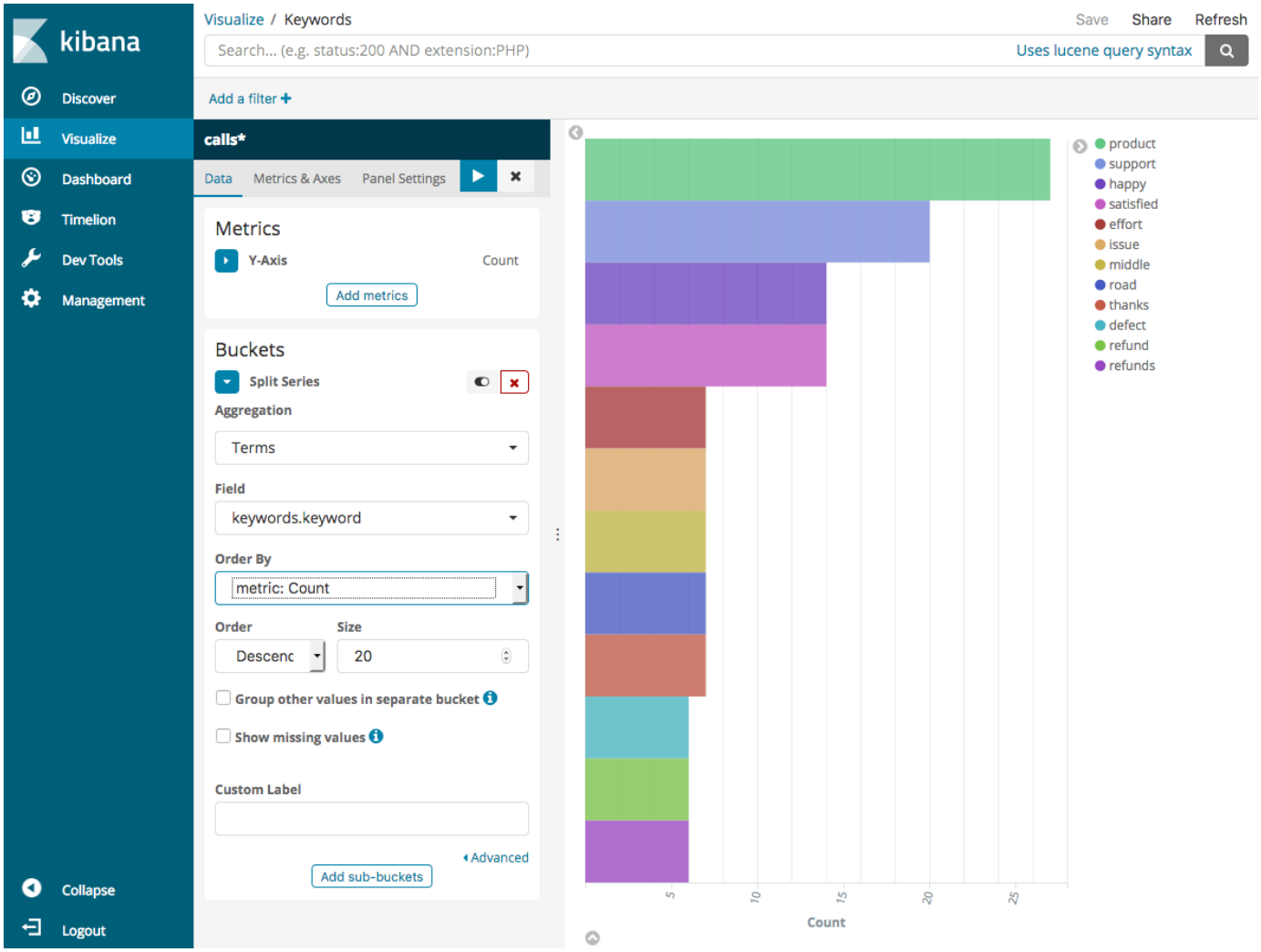
1. `https://search-domain.region.es.amazonaws.com/_dashboards` に移動します。
2. OpenSearch Dashboards を使用するには、インデックスパターンが必要です。Dashboards はインデックスパターンを使用して、分析を 1 つまたは複数のインデックスに絞り込みます。`call-center.py` が作成した `support-calls` インデックスと一致させるには、[スタックの管理]、[インデックスパターン] に移動し、`support*` のインデックスパターンを定義してから、[次のステップ] を選択します。
3. [タイムフィルターフィールド名] で、[タイムスタンプ] を選択します。
4. これで、可視化の作成を開始できます。[可視化] を選択し、新しい視覚化を追加します。
5. 円グラフと `support*` インデックスパターンを選択します。
6. デフォルトの可視化は基本的です。そこで、より魅力的な視覚化を作成するため、[分割スライス] を選択します。

   [集約] で、[用語] を選択します。[フィールド] で、[sentiment.keyword] を選択します。次に、[変更の適用] を選択し、[保存] を選択します。

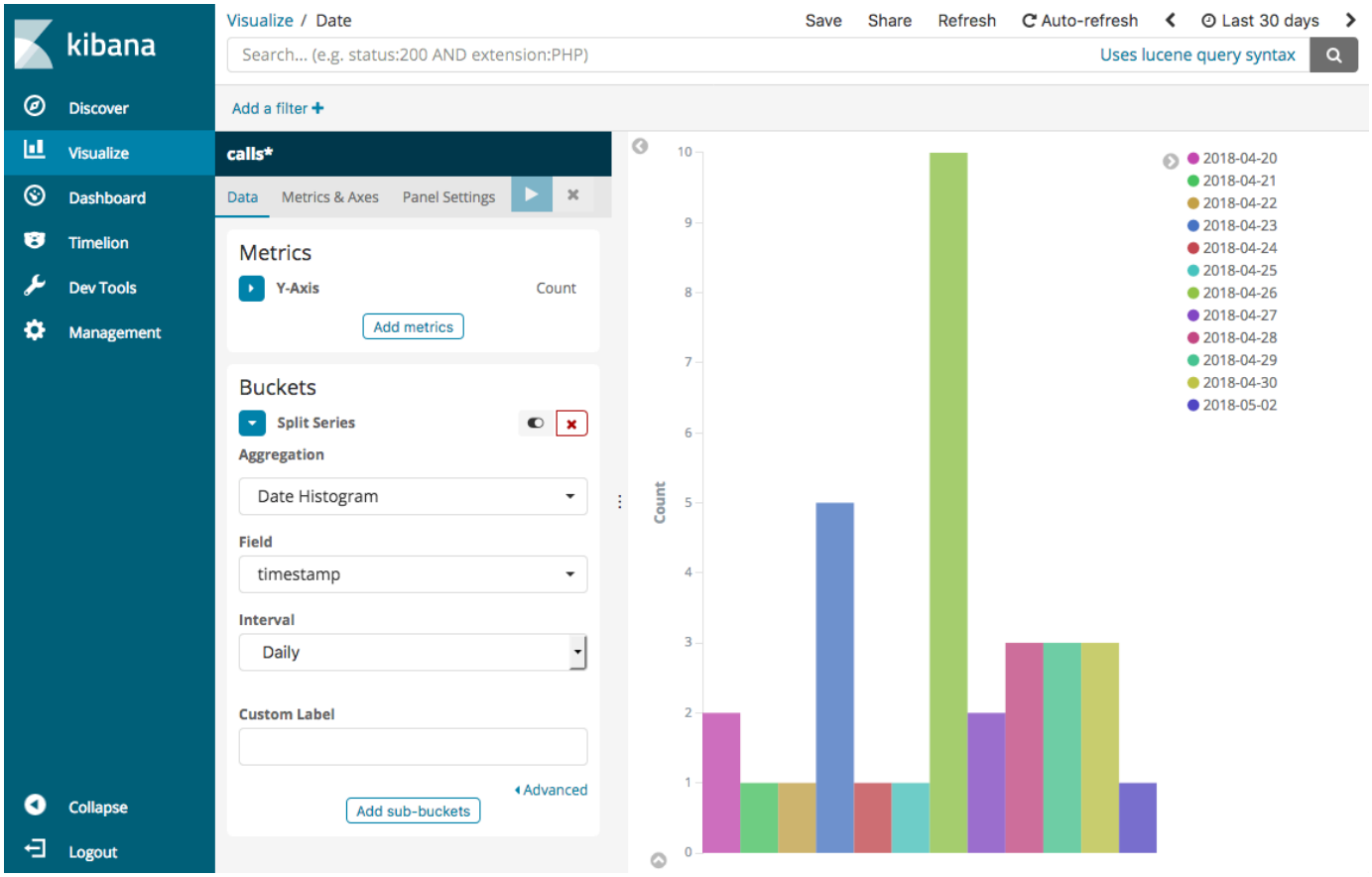


7. [可視化] ページに戻り、別の可視化を追加します。今回は、水平棒グラフを選択します。
8. [シリーズの分割] を選択します。

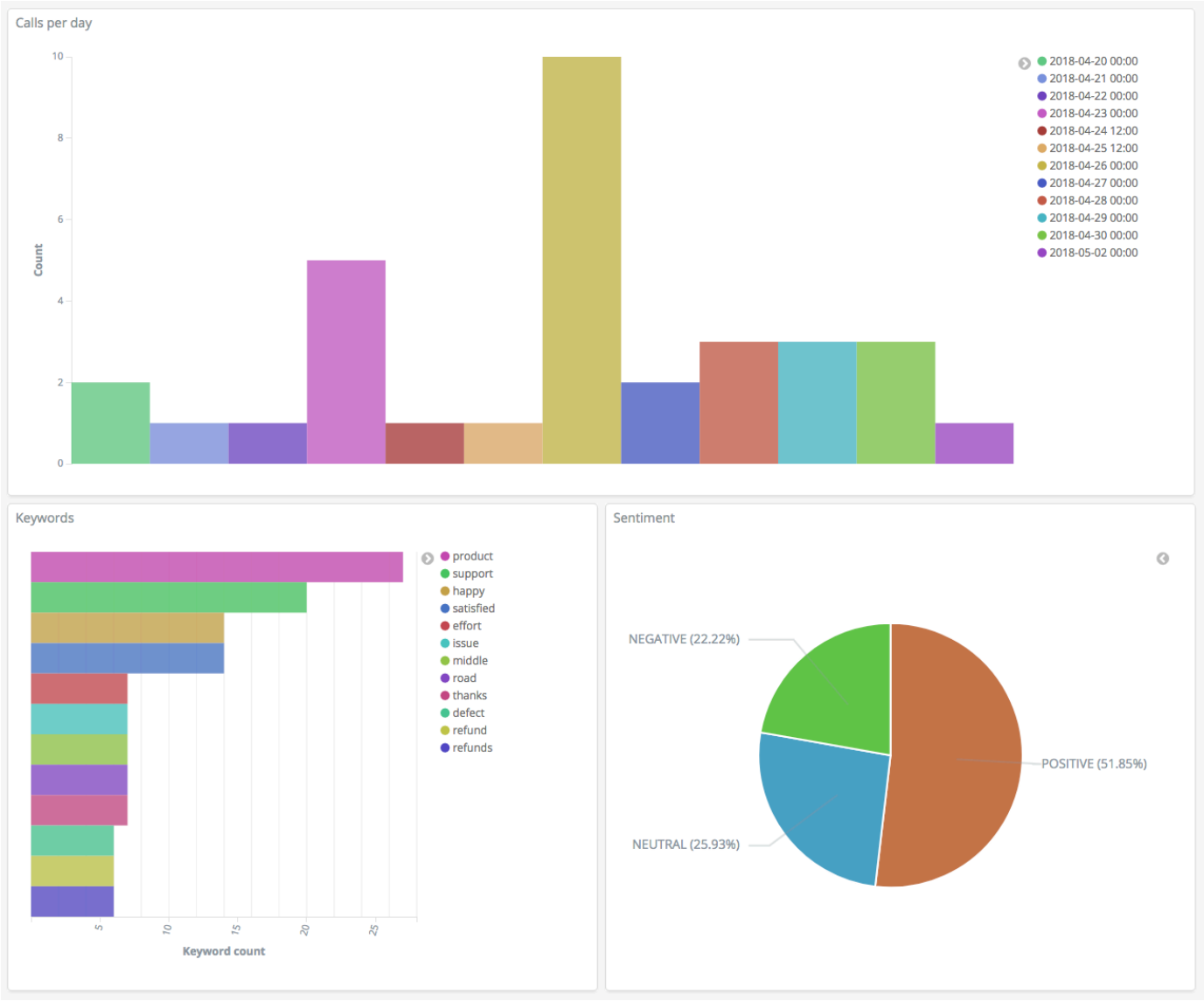
   [集約] で、[用語] を選択します。[フィールド] で、[keywords.keyword] を選択し、[サイズ] を 20 に変更します。次に、[変更の適用] を選択し、[保存] を選択します。



9. [可視化] ページに戻り、最終的な 1 つの視覚化である垂直棒グラフを追加します。
10. [シリーズの分割] を選択します。[集約] で、[日付ヒストグラム] を選択します。[フィールド] の [タイムスタンプ] を選択し、[間隔] を [日別] に設定します。
11. [メトリクス & 軸] を選択し、[モード] を [法線] に変更します。
12. [変更の適用] を選択し、[保存] を選択します。



13. これで 3 つの可視化を作成したので、Dashboards ダッシュボードに追加することができます。[ダッシュボード] を選択し、ダッシュボードを作成して、可視化を追加します。



ステップ 5: リソースのクリーンアップと次のステップ

不要な料金が発生しないようにするため、S3 バケットおよび OpenSearch Service ドメインを削除します。詳細については、Amazon Simple Storage Service ユーザーガイドの「バケットの削除」およびこのガイドの「OpenSearch Service ドメインを削除する」を参照してください。

トランスクリプトは、MP3 ファイルよりもはるかに少ないディスク容量で済みます。MP3 の保持期間を短くする (例えば、通話記録の保持期間を 3 か月から 1 か月にする) ことも、数年間のトランスクリプトを保持することもでき、いずれの場合もストレージコストを低減できます。

また、AWS Step Functions と Lambda を使用した書き起こしプロセスの自動化、インデックス作成前のメタデータの追加、またはユースケースに正確に合わせたより複雑な視覚化の作成もできます。

Amazon OpenSearch Service 名称変更 - 変更の概要

2021 年 9 月 8 日に、検索および分析スイートの名称が Amazon OpenSearch Service に変更されました。OpenSearch Service は、OpenSearch だけでなく、レガシーの Elasticsearch もサポートします。以下のセクションでは、名称変更によって変更されたサービスのさまざまな部分と、ドメインが正常に機能し続けるために実行する必要があるアクションについて説明します。

これらの変更の一部は、ドメインを Elasticsearch から OpenSearch にアップグレードするときにのみ適用されます。請求およびコストマネジメントコンソールなど、エクスペリエンスは速やかに変更されます。

これはすべてを網羅したリストではないことに注意してください。製品の他の部分も変更されましたが、これらの更新が最も関連性があります。

トピック

API の新バージョン

OpenSearch Service 設定API の新バージョン (2021-01-01) は、レガシーの Elasticsearch OSS と同様に OpenSearch で動作します。21 の API オペレーションは、より簡潔でエンジンに依存しない名前に置き換えられました (例えば、`CreateElasticsearchDomain` は `CreateDomain` に変更されました) が、OpenSearch Service は引き続き両方の API バージョンをサポートしています。

今後は、新しい API オペレーションを使用して、ドメインを作成および管理することをお勧めします。新しい API オペレーションを使用してドメインを作成するときは、バージョン番号だけではな

く、形式 Elasticsearch_X.Y または OpenSearch_X.Y で EngineVersion パラメータを指定する必要があります。バージョンを指定しない場合、デフォルトは OpenSearch の最新バージョンになります。

aws opensearch ... を使用してドメインの作成と管理を行うために、AWS CLI をバージョン 1.20.40 以降に更新します。新しい CLI 形式については、「OpenSearch CLI リファレンス」を参照してください。

名称変更されたインスタンスタイプ

Amazon OpenSearch Service のインスタンスタイプは形式 <type>.<size>.search になりました (例えば、m6g.large.*elasticsearch* ではなくm6g.large.*search*)。ご自身では特に何もする必要はありません。既存のドメインは、API 内と、請求およびコストマネジメントコンソール内で新しいインスタンスタイプを自動的に参照し始めます。

リザーブドインスタンス (RI) がある場合、契約は変更の影響を受けません。古い設定 API バージョンはまだ古い名前付け形式と互換性がありますが、新しい API バージョンを使用する場合は、新しい形式を使用する必要があります。

アクセスポリシーの変更

次のセクションでは、アクセスポリシーを更新するために必要なアクションについて説明します。

IAM ポリシー

名称変更された API オペレーションを使用するように IAM ポリシーを更新することをお勧めします。ただし、OpenSearch Service は、古い API 許可を内部で複製することで、既存のポリシーを引き続き優先します。例えば、CreateElasticsearchDomain オペレーションを実行するための許可を現在持っている場合、CreateElasticsearchDomain (古い API オペレーション) とCreateDomain (新しい API オペレーション) 両方への呼び出しを行えるようになりました。同じことが明示的拒否に適用されます。更新された API オペレーションのリストについては、「ポリシーエレメントリファレンス」を参照してください。

SCP ポリシー

サービスコントロールポリシー (SCP)では、標準の IAM に比べてさらに複雑な追加のレイヤーが導入されています。SCP ポリシーを中断しないようにするには、古いおよび新しい API オ

ペレーションの両方を各 SCP ポリシーに追加する必要があります。例えば、ユーザーが現在 CreateElasticsearchDomain の許可権限を持っている場合、彼らがドメインを作成する能力を保持できるように、CreateDomain の許可権限も彼らに付与する必要があります。同じことが明示的拒否に適用されます。

例:

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:CreateDomain"
         ...
      ],
    },
      "Effect": "Deny",
      "Action:" [
        "es:DeleteElasticsearchDomain",
        "es:DeleteDomain"
         ...
```

新しいリソースタイプ

OpenSearch Service では、次の新しいリソースタイプが導入されました。

リソース	説明
AWS::OpenSearchService::Domain	Amazon OpenSearch Service ドメインを表します。このリソースはサービスレベルに存在し、ドメインで実行されているソフトウェアに固有のものではありません。これは、AWS CloudFormation および AWS Resource Groups のようなサービスに適用されます。そこで、サービス全体のリソースが作成および管理されます。 CloudFormation 内で定義されたドメインを Elasticsearch から OpenSearch にアップグ

リソース	説明
	レードする手順については、CloudFormation ユーザーガイドの解説を参照してください。
AWS::OpenSearch::Domain	ドメインで実行されている OpenSearch/ ElasticSearch ソフトウェアを表します。このリソースは AWS CloudTrail および AWS Config のようなサービスに適用されます。これは、OpenSearch Service 全体ではなくドメイン上で実行しているソフトウェアを参照しています。これらのサービスには、OpenSearch を実行するドメイン (AWS::OpenSearch::Domain) に対して Elasticsearch を実行するドメイン (AWS::Elasticsearch::Domain) 用の個別のリソースタイプが含まれるようになりました。

> **Note**
>
> AWS Config で、1 つ以上のドメインを OpenSearch にアップグレードした場合でも、数週間は既存の AWS::Elasticsearch::Domain リソースタイプの下で引き続きデータを確認できます。

Kibana は OpenSearch Dashboards に名称変更されました

Kibana の AWS 代替機能である OpenSearch Dashboards は、OpenSearch で機能するように設計されたオープンソースの可視化ツールです。ドメインを Elasticsearch から OpenSearch にアップグレードすると、/_plugin/kibana エンドポイントは /_dashboards に変わります。OpenSearch Service はすべてのリクエストを新しいエンドポイントにリダイレクトしますが、いずれかの IAM ポリシーで Kibana エンドポイントを使用している場合は、新しい /_dashboards エンドポイントも含めるように、それらのポリシーを更新します。

the section called “ OpenSearch Dashboards の SAML 認証” を使用している場合は、ドメインを OpenSearch にアップグレードする前に、ID プロバイダー (IdP) によって設定されたすべての

Kibana URL を /_plugin/kibana から /_dashboards に変更する必要があります。最も一般的な URL は、Assertion Consumer Service (ACS) URL と受信者 URL です。

デフォルトの OpenSearch Dashboards の kibana_read_only ロールが opensearch_dashboards_read_only、kibana_user ロールが opensearch_dashboards_user に名称変更されました。この変更は、サービスソフトウェア R20211203 以降を実行して新しく作成されるすべての OpenSearch 1.x ドメインに適用されます。既存のドメインをサービスソフトウェア R20211203 にアップグレードしても、ロール名は変わりません。

名称変更された CloudWatch メトリクス

OpenSearch を実行するドメインでは、いくつかの CloudWatch メトリックスが変更されます。ドメインを OpenSearch にアップグレードすると、メトリクスが自動的に変更され、現在の CloudWatch アラームが中断されます。クラスターを Elasticsearch バージョンから OpenSearch バージョンにアップグレードする前に、新しいメトリクスを使用するように CloudWatch アラームを更新してください。

以下のメトリクスが変更されました。

元のメトリクス名	新しい名称
KibanaHealthyNodes	OpenSearchDashboardsHealthyNodes
KibanaConcurrentConnections	OpenSearchDashboardsConcurrentConnections
KibanaHeapTotal	OpenSearchDashboardsHeapTotal
KibanaHeapUsed	OpenSearchDashboardsHeapUsed
KibanaHeapUtilization	OpenSearchDashboardsHeapUtilization
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal

元のメトリクス名	新しい名称
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

OpenSearch Service が Amazon CloudWatch に送信するメトリクスの完全なリストについては、「the section called “クラスターメトリクスのモニタリング”」を参照してください。

請求およびコストマネジメントコンソールの変更

請求およびコストマネジメントコンソール内およびコストと使用状況レポート内の履歴データは引き続き古いサービス名を使用するため、データを検索するときは、[Amazon OpenSearch Service] と従来の Elasticsearch の名前の両方でフィルターの使用をスタートする必要があります。既存の保存済みレポートがある場合は、フィルターを更新して、OpenSearch Service も含まれていることを確認します。Elasticsearch の使用量が減少し、OpenSearch の使用量が増加すると、最初はアラートを受け取ることがありますが、それは数日以内に消えます。

サービス名の変更に加えて、次のフィールドは、すべてのレポート、請求書、および価格表の API オペレーションについて変更されます。

フィールド	古い形式	新しい形式
インスタンスタイプ	m5.large.elasticsearch	m5.large.search

フィールド	古い形式	新しい形式
製品ファミリー	Elasticsearch インスタンス Elasticsearch ボリューム	Amazon OpenSearch Service インスタンス Amazon OpenSearch Service ボリューム
料金の説明	c5.18xlarge.elasticsearch インスタンス時間 (または時間の端数分) あたり5.098 USD - EU	c5.18xlarge.search インスタンス時間 (または時間の端数分) あたり 5.098 USD - EU
インスタンスファミリー	`ultrawarm.elasticsearch`	`ultrawarm.search`

新しいイベント形式

OpenSearch サービスが Amazon EventBridge および Amazon CloudWatch に送信するイベントの形式が変更されました。具体的には、`detail-type` フィールドです。ソースフィールド (`aws.es`) は同じままです。各イベントタイプの完全な形式については、「the section called “イベントのモニタリング”」を参照してください。古い形式に依存する既存のイベントルールがある場合は、新しい形式に準拠するように更新してください。

同じままのものは何ですか？

以下の特徴と機能は、特に記載されていませんが、同じままです。

- サービスプリンシパル (`es.amazonaws.com`)
- ベンダーコード
- ドメイン ARN
- ドメインエンドポイント

使用開始: ドメインを OpenSearch 1.x にアップグレードします

OpenSearch 1.x は Elasticsearch バージョン 6.8 および 7.x からのアップグレードをサポートします。ドメインをアップグレードする手順については、「the section called “アップグレードの開始 (コ

ンソール)"」を参照してください。ドメインをアップグレードするために AWS CLI または設定 API を使用している場合は、 `TargetVersion` を `OpenSearch_1.x` と指定する必要があります。

OpenSearch 1.x では、[互換モードを有効にする] という追加のドメイン設定を導入しています。特定の Elasticsearch OSS クライアントおよびプラグインが、接続する前にクラスターのバージョンをチェックするので、互換モードでは OpenSearch がそのバージョンを 7.10 とレポートするように設定され、これらのクライアントが引き続き動作するようにします。

OpenSearch ドメインを初めて作成するとき、またはElasticsearch バージョンから OpenSearch バージョンにアップグレードするときに、互換モードを有効にできます。それが設定されていない場合、パラメータはデフォルトで、ドメインの作成時は `false` に、ドメインのアップグレード時は `true` になります。

設定 API を使用して互換モードを有効にするには、`override_main_response_version` を `true` に設定します。

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/upgradeDomain
{
  "DomainName": "domain-name",
  "TargetVersion": "OpenSearch_1.0",
  "AdvancedOptions": {
    "override_main_response_version": "true"
   }
}
```

既存の OpenSearch ドメインで互換モードを有効または無効にするには、OpenSearch _cluster/設定 API オペレーションを使用する必要があります。

```
PUT /_cluster/settings
{
  "persistent" : {
    "compatibility.override_main_response_version" : true
  }
}
```

Amazon OpenSearch サービスのトラブルシューティング

このトピックでは、Amazon OpenSearch Service の一般的な問題を特定して解決する方法について説明します。AWS サポートに問い合わせる前に、このセクションの情報を参照してください。

OpenSearch ダッシュボードにアクセスできない

OpenSearch ダッシュボードエンドポイントは署名付きリクエストをサポートしていません。ドメインのアクセスコントロールポリシーで、特定の IAM ロールにのみアクセス権が付与されており、Amazon Cognito 認証を設定していない場合は、Dashboards にアクセスしようとするときに次のエラーが発生する場合があります。

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

OpenSearch サービスドメインが VPC アクセスを使用している場合、このエラーは表示されませんが、リクエストがタイムアウトする可能性があります。この問題の修正、および利用できるさまざまな設定オプションの詳細については、「the section called “ OpenSearch Dashboards へのアクセスの制御”」、「the section called “VPC ドメインのアクセスポリシーについて”」、および「the section called “Identity and Access Management”」を参照してください。

VPC ドメインにアクセスできない

「the section called “VPC ドメインのアクセスポリシーについて”」および「the section called “VPC ドメインをテストする”」を参照してください。

読み取り専用状態のクラスター

以前のバージョンの Elasticsearch OpenSearch や Elasticsearch 7 と比較してみてください。 x はクラスターの調整に別のシステムを使用しています。この新しいシステムでは、クラスターがクォーラムを失うと、アクションを実行するまでクラスターは使用できなくなります。クォーラム損失には 2 つの形式があります。

- クラスターが専用マスターノードを使用している場合は、半分以上が利用できないときにクォーラム損失が発生します。
- クラスターで専用マスターノードを使用していない場合は、データノードの半分以上が利用できないときにクォーラム損失が発生します。

クォーラム損失が発生し、クラスターに複数のノードがある場合、 OpenSearch Service はクォーラムを復元し、クラスターを読み取り専用状態にします。これには 2 つのオプションがあります。

- 読み取り専用状態を削除し、クラスターをそのまま使用します。
- スナップショットからクラスターまたは個々のインデックスを復元します。

クラスターをそのまま使用する場合は、次のリクエストを使用してクラスターの状態が緑色であることを確認します。

```
GET _cat/health?v
```

クラスターの状態が赤の場合、スナップショットからクラスターを復元することをお勧めします。トラブルシューティングのステップについては、the section called “赤のクラスター状態” も合わせてお読みください。クラスターの状態が緑色の場合は、次のリクエストを使用して、予想されるすべてのインデックスが存在することを確認します。

```
GET _cat/indices?v
```

次に、いくつかの検索を実行して、予想されるデータが存在することを確認します。存在する場合は、次のリクエストを使用して読み取り専用状態を削除できます。

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.blocks.read_only": false
  }
}
```

クォーラム損失が発生し、クラスターにノードが 1 つしかない場合、 OpenSearch Service はそのノードを置き換え、クラスターを読み取り専用状態にはしません。それ以外の場合、オプションは同じです。つまり、クラスターをそのまま使用するか、スナップショットから復元します。

いずれの場合でも、 OpenSearch Service はに 2 つのイベントを送信します。AWS Health Dashboard最初のステップでは、クォーラムの損失が通知されます。2 つ目は、 OpenSearch Service がクォーラムを正常に復元した後に発生します。の使用方法について詳しくは AWS Health Dashboard、『ユーザーガイド』を参照してください。AWS Health

赤のクラスター状態

赤いクラスターステータスは、少なくとも 1 つのプライマリシャードとそのレプリカがノードに割り当てられていないことを意味します。 OpenSearch サービスはインデックスの状態に関係なくすべてのインデックスの自動スナップショットを取得しようとしますが、赤色のクラスターステータスが続く間はスナップショットは失敗します。

赤色のクラスターステータスの最も一般的な原因は、クラスターノードの障害と、 OpenSearch 継続的な高負荷によるプロセスのクラッシュです。

> **Note**
>
> OpenSearch Service は、クラスターの状態に関係なく、自動スナップショットを 14 日間保存します。したがって、赤のクラスター状態が 2 週間を超えて続くと、最後に正常な自動スナップショットが削除され、クラスターのデータが完全に失われることになります。 OpenSearch Service Domainのクラスタステータスが赤色になった場合は、自分で問題に対処するのか、 AWS Support それともサポートチームに支援してもらいたいのかを尋ねる場合があります。 CloudWatch 赤色のクラスターステータスが発生したときに通知するようにアラームを設定できます。

最終的に、赤のシャードにより赤のクラスターが発生し、赤のインデックスにより赤のシャードが発生します。赤色のクラスター状態の原因となっているインデックスを特定するのに役立つ API がいくつかあります。 OpenSearch

- `GET /_cluster/allocation/explain` は、見つかった最初の割り当てられていないシャードを選択し、そのシャードをノードに割り当てることができない理由について説明します。

```
{
    "index": "test4",
    "shard": 0,
    "primary": true,
    "current_state": "unassigned",
    "can_allocate": "no",
    "allocate_explanation": "cannot allocate because allocation is not permitted to
 any of the nodes"
}
```

- `GET /_cat/indices?v` はヘルス状態、ドキュメントの数、および各インデックスのディスク使用量を表示します。

```
health status index            uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   test1            30h1EiMvS5uAFr2t5CEVoQ   5   0        820            0
       14mb           14mb
green  open   test2            sdIxs_WDT56afFGu5KPbFQ   1   0          0            0
       233b           233b
green  open   test3            GGRZp_TBRZuSaZpAGk2pmw   1   1          2            0
     14.7kb          7.3kb
red    open   test4            BJxfAErbTtu5HBjIXJV_7A   1   0
green  open   test5            _8C6MIXOSxCqVYicH3jsEA   1   0          7            0
     24.3kb         24.3kb
```

赤のインデックスを削除することが、赤のクラスター状態を修正するための最速の方法です。赤いクラスターのステータスの理由によっては、より大きなインスタンスタイプ、より多くのインスタンス、または EBS OpenSearch ベースのストレージを使用するようにサービスドメインをスケーリングし、問題のあるインデックスを再作成してみることもできます。

問題のあるインデックスを削除することが可能でない場合、スナップショットを復元する、インデックスからドキュメントを削除する、インデックスの設定を変更する、レプリカの数を減らす、または他のインデックスを削除してディスク領域を解放することができます。重要なステップは、サービス· ドメインを再構成する前に、赤いクラスターの状態を解決することです。 OpenSearch 赤のクラスター状態のドメインを再設定すると、問題が複雑化し、状態を解決するまで、設定状態が [処理中] のままドメインがスタックする可能性があります。

赤いクラスターの自動修復

クラスターのステータスが 1 時間以上赤く表示され続けている場合、 OpenSearch Service は割り当てられていないシャードを再ルーティングするか、過去のスナップショットから復元することで、自動的に問題を解決しようとします。

1 つ以上の赤色のインデックスの修正に失敗し、クラスターのステータスが合計 14 日間赤色のままである場合、 OpenSearch Service はクラスターが次の基準の少なくとも 1 つを満たしている場合にのみ追加のアクションを実行します。

- アベイラビリティーゾーンが 1 つだけである
- 専用マスターノードがある
- バースト可能なインスタンスタイプ (T2 または T3) が含まれる

この時点で、クラスタがこれらの基準のいずれかを満たしている場合、 OpenSearch Service は次の 7 日間にわたって、これらのインデックスを修正しないと割り当てられていないシャードがすべて削除されることを説明する通知を毎日送信します。21 日経ってもクラスターのステータスが赤色のままの場合、 OpenSearch Service は赤色のインデックスにある未割り当てのシャード (ストレージとコンピューティング) をすべて削除します。これらのイベントのそれぞれについて、 OpenSearch サービスコンソールの通知パネルに通知が届きます。詳細については、「the section called “クラスターヘルスイベント”」を参照してください。

処理の継続的な高負荷からの復旧

赤のクラスター状態がデータノードの処理の継続的な高負荷によるものであるかどうかを判断するには、次のクラスターメトリクスをモニタリングします。

関連するメトリクス	説明	復旧
JVM MemoryPressure	クラスター内のすべてのデータノードで使用する Java ヒープのパーセンテージを指定します。このメトリクスの [最大] の統計を表示し、Java ガベージコレクターが十分なメモリの回収に失敗したことで生じる少量ずつのメモリプレッシャーを検出します。このパターンは通常、複雑なクエリまたは大きいデータフィールドが原因です。 x86 インスタンスタイプは、コンカレントマークスイープ (CMS) ガベージコレクターを使用します。このガベージコレクターは、アプリケーションスレッドとともに一時停止を短くします。通常の収集処理中に CMS が十分なメモリを再利用できない場合、完全なガベージコレクションがトリガーされ、アプリケーションが長時間一時停止し、クラスターの安定性に影響する可能性があります。	JVM のメモリ サーキットブレーカーを設定します。詳細については、「the section called “JVM OutOfMemoryError”」を参照してください。 問題が解決しない場合は、不要なインデックスを削除する、ドメインへのリクエストの数または複雑性を減少する、インスタンスを追加する、あるいはより大きなインスタンスタイプを使用します。

関連するメトリクス	説明	復旧
	ARM ベースの Graviton インスタンスタイプは、Garbage-First (G1) ガベージコレクターを使用します。G1 ガベージコレクターは CMS に似ていますが、追加の短い一時停止とヒープの最適化を使用して、完全なガベージコレクションの必要性をさらに減らします。 いずれの場合も、ガベージコレクションが満杯になったときにガベージコレクターが再利用できる量を超えてメモリー使用量が増え続けると、 OpenSearch メモリー不足エラーでクラッシュします。すべてのインスタンスタイプで使用率を 80% 以下にすることをお勧めします。 _nodes/stats/jvm API は、JVM 統計の有用な要約、メモリプールの使用量、およびガベージコレクションの情報を提供します。 `GET domain-endpoint /_nodes/stats/jvm?pretty`	
CPUUtilization	クラスター内のデータノードで使用する CPU リソースのパーセンテージを指定します。このメトリクスの [最大] の統計を表示し、使用量の多い継続的なパターンを探します。	データノードを追加するか、既存のデータノードのインスタンスタイプのサイズを大きくします。

関連するメトリクス	説明	復旧
ノード	クラスターのノード数の指定。このメトリクスの [最小] の統計を表示します。サービスがクラスターの新しいインスタンス群をデプロイすると、この値が変動します。	データノードを追加します。

黄色のクラスター状態

黄色のクラスター状態は、すべてのインデックスのプライマリシャードがクラスター内のノードに割り当てられ、少なくとも 1 つのインデックスのレプリカシャードは割り当てられていないことを意味します。 OpenSearch Service がレプリカを割り当てることができるノードは他にないため、単一ノードクラスターは常に黄色のクラスターステータスで初期化されます。緑のクラスター状態を確保するには、ノード数を増やします。詳細については、「the section called “ドメインのサイジング”」を参照してください。

マルチノードクラスターは、新しいインデックスを作成した後、またはノード障害の後に、一時的に黄色のクラスター状態になることがあります。このステータスは、 OpenSearch クラスター全体でデータを複製することで自動的に解決されます。ディスク容量の不足も黄色のクラスター状態を引き起こす可能性があります。クラスターは、ノードにレプリカシャードを収容するディスク領域がある場合のみ、レプリカシャードを配布できます。

ClusterBlockException

以下の理由により、`ClusterBlockException` エラーを受け取る場合があります。

使用可能なストレージ領域の不足

クラスター内の 1 つ以上のノードのストレージ容量が、1) 使用可能なストレージスペースの 20%、または 2) 20 GiB のストレージスペースの最小値未満の場合、ドキュメントの追加やインデックスの作成などの基本的な書き込み操作が失敗し始める可能性があります。 the section called “ストレージ要件の計算” OpenSearch Service がディスク容量をどのように使用するかの概要を示します。

問題を回避するには、`FreeStorageSpace` OpenSearch サービスコンソールでメトリックスを監視し、 CloudWatch `FreeStorageSpace`特定のしきい値を下回ったときにトリガーするアラームを

作成します。 `GET /_cat/allocation?v`また、シャード割り当てとディスク使用量の便利な概要も表示されます。ストレージ容量不足に関連する問題を解決するには、 OpenSearch サービストメインをスケーリングして、より大きなインスタンスタイプ、より多くのインスタンス、または EBS ベースのストレージを使用するようにします。

JVM メモリ負荷が高い

JVM MemoryPressure メトリックスが 30 分間 92% を超えると、 OpenSearch Service は保護メカニズムをトリガーし、すべての書き込み操作をブロックして、クラスターが赤のステータスになるのを防ぎます。保護が有効な状態では、書き込みオペレーションは `ClusterBlockException` エラーで失敗し、新しいインデックスは作成できず `IndexCreateBlockException` エラーがスローされます。

JVM MemoryPressure メトリクスが 88% 以下に戻って 5 分間続くと、保護は無効になり、クラスターへの書き込み操作のブロックは解除されます。

高い JVM のメモリ負荷は、クラスターに対するリクエスト数の急増、ノード全体でのシャード割り当ての不均衡、クラスター内の過度に多いシャード、大量のフィールドデータまたはインデックスマッピング、または入ってくる負荷を処理できないインスタンスタイプによって引き起こされることがあります。また、集計やワイルドカードを使用したり、クエリで広い時間範囲を使用したりすることによって発生することもあります。

クラスターへのトラフィックを減らし、JVM のメモリ負荷が高い問題を解決するには、次の 1 つ以上を試してください。

- ノードあたりの最大ヒープサイズが 32 GB になるようにドメインをスケールする。
- 古いインデックスや未使用のインデックスを削除して、シャードの数を減らす。
- `POST` *`index-name`*`/_cache/clear?fielddata=true` API オペレーションでデータキャッシュをクリアする。キャッシュをクリアすると、進行中のクエリが中断される可能性があることに注意してください。

一般的に、将来 JVM メモリの負荷が高くなるのを避けるために、次のベストプラクティスに従ってください。

- テキストフィールドでの集計を避けるか、インデックスのマッピングタイプを `keyword` に変更します。
- 適切な数のシャードを選択して、検索とインデックス作成のリクエストを最適化します。

- 定期的に未使用のインデックスを削除するように Index State Management (ISM) ポリシーを設定します。

Multi-AZ with Standby への移行中にエラーが発生した

既存のドメインを Multi-AZ with Standby に移行する際に、次の問題が発生する場合があります。

スタンバイのないドメインからスタンバイのあるドメインへの移行している最中に、インデックス、インデックステンプレート、ISM ポリシーのいずれかを作成する

スタンバイのないマルチ AZ からスタンバイのあるマルチ AZ に移行する際にインデックスを作成し、そのインデックスのテンプレートまたは ISM ポリシーが、推奨されているデータコピーのガイドラインに従っていない場合、データの不一致が生じて移行に失敗する可能性があります。この状況を回避するには、3 の倍数のデータコピー数 (プライマリノードとレプリカの両方を含む) で新しいインデックスを作成します。移行の進行状況は、`DescribeDomainChangeProgress` API を使って確認できます。レプリカ数のエラーが発生した場合は、エラーを修正してからAWS サポート に連絡し、移行を再試行します。

データコピーの数が間違っている

ドメインに適切な数のデータコピーがないと、Multi-AZ with Standby への移行は失敗します。

JVM OutOfMemoryError

JVM の `OutOfMemoryError` は、一般的に次のいずれかの JVM サーキットブレーカーに到達したことを意味します。

サーキットブレーカー	説明	クラスター設定プロパティ
親ブレーカー	すべてのサーキットブレーカーで JVM ヒープメモリのパーセンテージの合計に許可されます。デフォルト値は 95% です。	`indices.breaker.total.limit`

サーキットブレーカー	説明	クラスター設定プロパティ
フィールド データ ブレーカー	JVM ヒープメモリのパーセンテージは、メモリに単一のデータフィールドをロードすることを許可します。デフォルト値は 40% です。大きいフィールドを用いてデータをアップロードする場合は、この上限を引き上げる必要があります。	`indices.breaker.fielddata.limit`
リクエストブレーカー	JVM ヒープメモリのパーセンテージは、サービスリクエストに応答するために使用されるデータ構造に許可されます。デフォルト値は 60% です。サービスリクエストが集計の計算を含む場合は、この上限を引き上げる必要がある場合があります。	`indices.breaker.request.limit`

障害が発生したクラスターノード

Amazon EC2 インスタンスでは、予期しない終了と再起動が発生する場合があります。通常、OpenSearch サービスはノードを自動的に再起動します。ただし、 OpenSearchクラスター内の 1 つ以上のノードが障害状態のままになることがあります。

この状態を確認するには、 OpenSearch サービスコンソールでドメインダッシュボードを開きます。[クラスターのヘルス] タブに移動し、[ノード合計数] メトリクスを見つけます。報告されるノード数がクラスターに設定した数より小さいかどうかを調べます。メトリクスが、1 つ以上のノードが 1 日以上ダウンしていることを示している場合は、AWS サポートまでお問い合わせください。

また、 CloudWatch この問題が発生したときに通知するアラームを設定することもできます。

> **Note**
>
> [合計ノード] メトリクスは、クラスター設定の変更中およびサービスの定期的なメンテナンス中は、正確ではありません。この動作は想定されるものです。メトリクスは、すぐに正しい数のクラスターノードを報告します。詳細については、「the section called “設定変更”」を参照してください。

クラスターを予期しないノード終了や再起動から保護するには、 OpenSearch サービスドメイン内のインデックスごとに少なくとも 1 つのレプリカを作成します。

シャードの最大制限を超えました

OpenSearch 7 も同様です。 x バージョンの Elasticsearch では、ノードあたりのシャード数は 1,000 個以下のデフォルト設定になっています。 OpenSearch/Elasticsearch は、新しいインデックスの作成などのリクエストによってこの制限を超えるとエラーを返します。このエラーが発生した場合は、いくつかのオプションがあります。

- さらにデータノードをクラスターに追加します。
- `_cluster/settings/cluster.max_shards_per_node` 設定を増加します。
- _shrink API を使用して、ノードのシャード数を減らします。

ドメインが処理状態でスタックしている

OpenSearch 設定変更の途中、サービスドメインは「処理中」状態になります。設定変更を開始すると、 OpenSearch Service が新しい環境を作成する間、ドメインのステータスが「処理中」に変わります。新しい環境では、 OpenSearch Service は適用可能なノードの新しいセット (データ、マスター、など UltraWarm) を起動します。移行が完了すると、古いノードは終了します。

次のいずれかの状況が発生した場合、クラスターは「処理中」状態でスタックすることがあります。

- 新しいデータノードセットの起動は失敗します。
- 新しいデータノードセットへのシャード移行は失敗します。
- 検証チェックがエラーで失敗しました。

これらの各状況における詳細な解決手順については、「Amazon OpenSearch Service ドメインが「処理中」状態で停止しているのはなぜですか?」を参照してください。 。

低 EBS バーストバランス

OpenSearch いずれかの汎用 (SSD) ボリュームの EBS バースト残高が 70% を下回るとコンソール通知が送信され、残高が 20% を下回るとフォローアップ通知が送信されます。この問題を解決するには、クラスターをスケールアップするか、読み取り/書き込み IOPS を減らしてバーストバランスをクレジットすることができます。gp3 ボリュームタイプを使用するドメインと、ボリュームサイズが 1,000 GiB を超える gp2 ボリュームを使用するドメインのバーストバランスは 0 のままになります。詳細については、「汎用 SSD ボリューム (gp2)」を参照してください。EBS バースト残高はメトリックスで監視できます。`BurstBalance` CloudWatch

監査ログを有効にできない

OpenSearch サービスコンソールを使用して監査ログの公開を有効にしようとすると、次のエラーが発生することがあります。

CloudWatch Logs ロググループに指定されたリソースアクセスポリシーでは、Amazon OpenSearch Service がログストリームを作成するための十分なアクセス権限を付与していません。リソースアクセスポリシーを確認してください。

このエラーが発生した場合は、ポリシーの `resource` 要素に正しいロググループ ARN が含まれていることを確認してください。その場合は、次のステップを実行します。

1. 数分間待ちます。
2. ウェブブラウザでページを更新します。
3. [既存のグループを選択] を選択します。
4. [既存のロググループ] では、エラーメッセージを受け取る前に作成したロググループを選択します。
5. [アクセスポリシー]セクションで、[既存のポリシーを選択] を選択します。
6. [既存のポリシー] では、エラーメッセージを受け取る前に作成したポリシーを選択します。
7. [有効] を選択します。

プロセスを数回繰り返してもエラーが続く場合は、AWS サポートに連絡してください。

インデックスが閉じない

OpenSearch このサービスは Elasticsearch バージョン 7.4 以降の `_close` OpenSearch API のみをサポートしています。古いバージョンを使用していて、スナップショットからインデックスを復元している場合は、既存のインデックスを削除することができます (それの再インデックスの前または後)。

クライアントライセンスのチェック

LogstashとBeatsのデフォルトディストリビューションには独自のライセンスチェックが含まれており、のオープンソースバージョンには接続できません。 OpenSearchService では、これらのクライアントの Apache 2.0 (OSS) ディストリビューションを必ず使用してください。 OpenSearch

リクエストのスロットリング

永続的に `403 Request throttled due to too many requests` または `429 Too Many Requests` エラーを受け取る場合は、垂直スケーリングを検討してください。ペイロードによってメモリ使用量が Java ヒープの最大サイズを超える場合、Amazon OpenSearch Service はリクエストを調整します。

ノードに SSH 接続できない

SSH OpenSearch を使用してクラスター内のどのノードにもアクセスすることはできず、直接変更することもできません。`opensearch.yml`代わりに、コンソール AWS CLI、または SDK を使用してドメインを設定してください。 OpenSearchREST API を使用してクラスターレベルの設定をいくつか指定することもできます。詳細については、「Amazon OpenSearch サービス API リファレンス」と「」を参照してくださいthe section called “サポートされているオペレーション”。

クラスターのパフォーマンスについてさらに詳しく知りたい場合は、エラーログとスローログをに公開できます CloudWatch。

「オブジェクトのストレージクラスで有効ではない」スナップショットエラー

OpenSearch サービススナップショットは S3 Glacier ストレージクラスをサポートしていません。このエラーは、オブジェクトを S3 Glacier ストレージクラスに移行するライフサイクルルールが S3

バケットに含まれている場合に、スナップショットのリストを取得しようとすると発生する場合があります。

バケットからスナップショットを復元する必要がある場合は、S3 Glacier からオブジェクトを復元し、オブジェクトを新しいバケットにコピーして、スナップショットリポジトリとして新しいバケットを登録します。

無効なホストヘッダー

OpenSearch サービスでは、Hostクライアントがリクエストヘッダーで指定する必要があります。有効な Host 値は、次のような、https:// のないドメインエンドポイントです。

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

Invalid Host Headerリクエストを行う際にエラーが発生した場合は、 OpenSearch Hostクライアントまたはプロキシのヘッダーにサービスドメインエンドポイント (IP アドレスなどではない) が含まれていることを確認してください。

無効な M3 インスタンスタイプ

OpenSearch このサービスは、Elasticsearch バージョン 6.7 以降を実行している既存のドメインへの M3 OpenSearch インスタンスの追加や変更をサポートしていません。Elasticsearch 6.5 以前では、M3 インスタンスを引き続き使用できます。

新しいインスタンスタイプの選択をお勧めします。Elasticsearch 6.7 OpenSearch 以降を実行しているドメインには、以下の制限が適用されます。

- 既存のドメインで M3 インスタンスを使用していない場合、今後これらに変更することはできません。
- 既存のドメインを M3 インスタンスタイプから別のインスタンスタイプに変更した場合、元に戻すことはできません。

ホットクエリは有効化すると動作しなくなります。 UltraWarm

UltraWarm ドメインで有効にすると、search.max_buckets設定に既存のオーバーライドがない場合、 OpenSearch Service 10000 はメモリを大量に消費するクエリがウォームノードを飽和させな

いように、値を自動的にに設定します。ホットクエリが 10,000 個を超えるバケットを使用している場合は、有効にすると動作しなくなる可能性があります。 UltraWarm

Amazon OpenSearch Service はマネージド型の性質上、この設定を変更できないため、サポートケースを開いて制限を増やす必要があります。制限の増加には、プレミアムサポートサブスクリプションは必要ありません。

アップグレード後にダウングレードできない

インプレースアップグレードは元に戻すことができませんが、AWS サポートに連絡すれば、新しいドメインで自動的なアップグレード前のスナップショットを復元する支援が得られます。たとえば、ドメインをElasticsearch 5.6から6.4にアップグレードする場合、 AWS Support はアップグレード前のスナップショットを新しいElasticsearch 5.6ドメインに復元するお手伝いをします。元のドメインの手動スナップショットを作成した場合は、このステップを自分で実行することができます。

すべての AWS リージョンのドメインの概要が必要

次のスクリプトは、Amazon EC2 describe-regions AWS CLI コマンドを使用して、 OpenSearch サービスを利用できるすべてのリージョンのリストを作成します。次に、各リージョンを呼び出しますlist-domain-names。

```
for region in `aws ec2 describe-regions --output text | cut -f4`
do
    echo "\nListing domains in region '$region':"
    aws opensearch list-domain-names --region $region --query 'DomainNames'
done
```

リージョンごとに次の出力が表示されます。

```
Listing domains in region:'us-west-2'...
[
  {
    "DomainName": "sample-domain"
  }
]
```

OpenSearch サービスが利用できないリージョンは「エンドポイント URL に接続できませんでした」を返します。

OpenSearch ダッシュボード使用時のブラウザエラー

ダッシュボードを使用してサービスドメイン内のデータを表示すると、ブラウザはサービスエラーメッセージを HTTP レスポンスオブジェクトにまとめます。 OpenSearch 原因となるサービスエラーを表示し、デバッグを支援するため、Chrome の開発者モードなどのウェブブラウザで一般的に利用されている開発ツールを使用できます。

Chrome でサービスエラーを表示する

1. Chrome のトップメニューバーから、[表示]、[デベロッパー]、[デベロッパーツール] の順に選択します。
2. [ネットワーク] タブを選択します。
3. [状態] 列で、状態が 500 の任意の HTTP セッションを選択します。

Firefox でサービスエラーを表示する

1. メニューで、[ツール]、[ウェブデベロッパー]、[ネットワーク] の順に選択します。
2. 状態が 500 の任意の HTTP セッションを選択します。
3. [レスポンス] タブを選択して、サービス応答を表示します。

ノードシャードとストレージスキュー

「shard skew」のノードは、クラスター内の 1 つ以上のノードに、他のノードに比べて非常に多くのシャードがある場合に発生します。「storage skew」のノードは、クラスター内の 1 つ以上のノードに、他のノードに比べて非常に多くのストレージ (`disk.indices`) がある場合に発生します。ドメインがノードを置き換え、シャードをそのノードに引き続き割り当てているときなど、これらの条件は両方とも一時的に発生する可能性がありますが、それらが持続する場合は対処する必要があります。

両方のタイプのスキューを識別するには、_cat/allocation API オペレーションを実行し、レスポンスの `shards` と `disk.indices` のエントリを比較します。

```
shards    | disk.indices  | disk.used  | disk.avail   | disk.total   | disk.percent |
host      | ip        | node
   264    |      465.3mb  |   229.9mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x   | node1
```

```
   115    |        7.9mb   |    83.7mb   |     49.1gb    |     49.2gb    |           0 |
x.x.x.x   | x.x.x.x   | node2
   264    |      465.3mb   |   235.3mb   |      1.4tb    |      1.5tb    |           0 |
 x.x.x.x   | x.x.x.x   | node3
   116    |        7.9mb   |    82.8mb   |     49.1gb    |     49.2gb    |           0 |
x.x.x.x   | x.x.x.x   | node4
   115    |        8.4mb   |      85mb   |     49.1gb    |     49.2gb    |           0 |
x.x.x.x   | x.x.x.x   | node5
```

一部のストレージスキューは正常ですが、平均から 10% を超えるものは重要です。シャードの分散にスキューがあると、CPU、ネットワーク、およびディスク帯域幅の使用量にもスキューが生じる可能性があります。通常、データが多いほどインデックス作成と検索オペレーションが増大するため、最も重いノードはリソースに最も負荷のかかるノードである傾向があり、軽いノードは十分に活用されていないキャパシティーを表します。

修正: データノード数の倍数であるシャード数を使用して、各インデックスがデータノード全体で均等に分散されるようにします。

インデックスシャードとストレージスキュー

「shard skew」のインデックスは、1 つ以上のノードが、他のノードよりも多くのインデックスのシャードを保持する場合に発生します。「storage skew」のインデックスは、1 つ以上のノードが、不均衡に大量のインデックスの合計ストレージを保持する場合に発生します。

_cat/shards API 出力を操作する必要があるため、インデックススキューはノードスキューよりも識別が困難です。クラスターまたはノードメトリクスに何らかのスキューの兆候がある場合は、インデックスのスキューを調査します。インデックススキューの一般的な兆候は次のとおりです。

- データノードのサブセットで発生する HTTP 429 エラー
- データノード全体における不均等なインデックスまたは検索オペレーションのキューイング
- データノード全体における不均等な JVM ヒープおよび/または CPU 使用率

修正: データノード数の倍数であるシャード数を使用して、各インデックスがデータノード全体で均等に分散されるようにします。それでもインデックスストレージやシャードスキューが発生する場合は、サービスドメインがブルー/グリーンデプロイされるたびにシャードの再割り当てを強制する必要があるかもしれません。 OpenSearch

VPC アクセス選択後の許可されていないオペレーション

OpenSearch サービスコンソールを使用して新しいドメインを作成する場合、VPC またはパブリックアクセスを選択できます。VPC アクセスを選択すると、 OpenSearch Service は VPC 情報をクエリし、適切な権限がない場合は失敗します。

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code:
 403; Error Code: UnauthorizedOperation
```

このクエリを有効にするには、`ec2:DescribeVpcs`、`ec2:DescribeSubnets`、および `ec2:DescribeSecurityGroups` オペレーションへのアクセス権を持っている必要があります。この要件は、コンソールのみを対象としています。 AWS CLI を使用して VPC エンドポイントを使用してドメインを作成および設定する場合、それらの操作にアクセスする必要はありません。

VPC ドメイン作成後、読み込みでスタックする

VPC アクセスを使用した新規のドメインを作成後、ドメインの [設定の状態] が [読み込み中] から先に進行しない場合があります。この問題が発生した場合は、お住まいのリージョンで AWS Security Token Service (AWS STS) が無効になっている可能性があります。

VPC エンドポイントを VPC に追加するには、 OpenSearch Service がその役割を引き受ける必要があります。`AWSServiceRoleForAmazonOpenSearchService`したがって、特定のリージョンで VPC アクセスを使用する新しいドメインを作成するには、 AWS STS を有効にする必要があります。有効化と無効化の詳細については AWS STS、『IAM ユーザーガイド』を参照してください。

API へのリクエストが拒否されました。 OpenSearch

OpenSearch API にタグベースのアクセス制御が導入されたことで、これまでになかったアクセス拒否エラーが表示されるようになるかもしれません。これは、1 つ以上のアクセスポリシーに `ResourceTag` 条件を使用する `Deny` が含まれており、それらの条件が現在尊重されていることが原因である可能性があります。

例えば、次のポリシーは、ドメインにタグ `environment=production` がある場合にのみ、設定 API からの `CreateDomain` アクションへのアクセスを拒否するために使用されます。アクションリストには `ESHttpPut` も含まれていますが、拒否ステートメントはそのアクションまたはその他の `ESHttp*` アクションには適用されませんでした。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:CreateDomain",
      "es:ESHttpPut"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

OpenSearch HTTP メソッドのタグサポートが追加されたことで、上記のような IAM ID ベースのポリシーでは、アタッチされたユーザーはアクションへのアクセスを拒否されるようになります。ESHttpPut以前は、タグの検証がない場合でも、添付されたユーザーは PUT リクエストを送信できました。

ドメインをサービスソフトウェア R20220323 以降に更新した後にアクセス拒否エラーが表示され始めた場合は、ID ベースのアクセスポリシーをチェックして、これが当てはまるかどうかを確認し、必要に応じてアクセスを許可するように更新してください。

Alpine Linux から接続できない

Alpine Linux では、DNS レスポンスのサイズが 512 バイトに制限されています。Alpine Linux バージョン 3.18.0 OpenSearch 以前からサービスドメインに接続しようとすると、ドメインが VPC 内にあり、ノードが 20 を超えると DNS 解決に失敗することがあります。Alpine Linux バージョン 3.18.0 以降を使用している場合は、20 を超えるホストを解決できます。詳細については、Alpine Linux 3.18.0 のリリースノートを参照してください。

ドメインが VPC 内にある場合は、他の Linux ディストリビューション (Debian、Ubuntu、CentOS、Red Hat Enterprise Linux、Amazon Linux 2 など) を使用して接続することをお勧めします。

Search Backpressure のリクエストが多すぎる

CPU ベースのアドミッションコントロールは、トラフィックの自然な増加と急増の両方について、現在のキャパシティに基づいてノードに対するリクエストの数をプロアクティブに制限するゲートキーピングメカニズムです。多すぎるリクエストは、拒否される際に HTTP 429「リクエストが多すぎます」ステータスコードを返します。このエラーは、クラスターリソースの不足、リソースを大量に消費する検索リクエスト、またはワークロードの意図しない急増のいずれかを示します。

Search Backpressure は拒否の理由を提供し、リソースを大量に使用する検索リクエストを微調整するのに役立ちます。トラフィックが急増した場合は、エクスポネンシャルバックオフとジッターを使用してクライアント側で再試行することをお勧めします。

SDK を使用する場合の証明書のエラー

AWS SDK はコンピュータの CA 証明書を使用するため、SDK AWS を使用しようとしたときにサーバー上の証明書を変更すると、接続に失敗する可能性があります。エラーメッセージはさまざまですが、通常は次のテキストが含まれています。

```
Failed to query OpenSearch
...
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

コンピューターの CA up-to-date 証明書とオペレーティングシステムをそのまま使用することで、このような障害を防ぐことができます。ユーザーが自分のコンピュータを管理していない企業環境でこの問題が発生した場合は、必要に応じて管理者から支援を得て更新プロセスを行う必要があります。

以下のリストは、オペレーティングシステムと Java の最小バージョンを示しています。

- 2005 年 1 月以降の更新プログラムがインストールされた Microsoft Windows バージョンでは、必要な CA が信頼リストに 1 つ以上含まれています。
- Mac OS X 10.4 with Java for Mac OS X 10.4 Release 5 (2007 年 2 月)、Mac OS X 10.5 (2007 年 10 月)、および以降のバージョンでは、必要な CA が信頼リストに 1 つ以上含まれています。
- Red Hat Enterprise Linux 5 (2007 年 3 月)、6、7、および CentOS 5、6、および 7 では、必要な CA がデフォルトの CA 信頼リストに 1 つ以上含まれています。
- Java 1.4.2_12 (2006 年 5 月)、5 Update 2 (2005 年 3 月)、および以降のすべてのバージョン (Java 6 (2006 年 12 月)、7、8 を含む) では、必要な CA がデフォルトの CA 信頼リストに 1 つ以上含まれています。

以下に示す 3 つの証明機関があります。

- Amazon Root CA 1
- Starfield Services Root Certificate Authority - G2
- Starfield Class 2 Certification Authority

最初の 2 つの認証局からのルート証明書は Amazon Trust Services から入手できますが、 up-to-date コンピュータを保管しておく方が簡単な解決策です。ACM から提供される証明書の詳細については、「AWS Certificate Manager に関するよくある質問」を参照してください。

> **Note**
>
> 現在、us-east-1 OpenSearch リージョンのサービスドメインは、別の機関の証明書を使用しています。近い将来、これらの新しい証明機関が使用されるように、リージョンを更新する予定です。

Amazon OpenSearch Service のドキュメント履歴

このトピックでは、Amazon OpenSearch Service の重要な変更について説明します。サービスソフトウェアの更新により、新機能、セキュリティパッチ、バグ修正、その他の機能強化のサポートが追加されます。新機能を使用するには、ドメインでのサービスソフトウェアの更新が必要になる場合があります。詳細については、「the section called “サービスソフトウェア更新”」を参照してください。

サービス機能は、サービス AWS リージョン が利用可能な に段階的にロールアウトされます。このドキュメントは、最初のリリースのためにのみ更新されています。リージョンの可用性に関する情報を提供したり、その後のリージョンのロールアウトを発表したりすることはありません。サービス機能のリージョンの可用性、および更新に関する通知のサブスクライブについては、「 の最新情報」を参照してください AWS。

この履歴に関連する日付:

- 現在の製品バージョン – 2021 年 1 月 1 日
- 最新の製品リリース日 — 2024 年 6 月 12 日
- ドキュメントの最終更新日 - 2024 年 6 月 12 日

更新に関する通知については、RSS フィードにサブスクライブできます。

> **Note**
>
> パッチリリース: 末尾が「-P」と数字のサービスソフトウェアバージョン (例: R20211203-P4) がパッチリリースです。パッチには、パフォーマンスの改善、マイナーなバグ修正、セキュリティ修正または体制の改善が含まれる可能性があります。パッチには新機能や重大な変更は含まれないため、一般的にユーザーやドキュメントに直接的な影響はありません。そのため、各パッチの詳細はこのドキュメントの履歴に含まれていません。

変更	説明	日付
新しいサービスにリンクされたロール	Amazon OpenSearch Service は、 というサービスにリンクされたロールを追加しま	2024 年 6 月 12 日

	す。これによりAWSServiceRoleForOpensearchIngestionSelfManagedVpce 、Amazon Ingestion OpenSearch は、セルフマネージド VPC エンドポイントを持つパイプライン Amazon CloudWatch のメトリクスデータを に送信できます。	
Amazon S3 との Amazon OpenSearch Service のゼロ ETL 統合 Amazon S3	Amazon OpenSearch Service は、Amazon S3 内のデータをクエリするための直接クエリをサポートするようになりました。	2024 年 5 月 22 日
OpenSearch 2.13 のサポート	Amazon OpenSearch Service が OpenSearch バージョン 2.13 をサポートするようになりました。このバージョンには、バージョン 2.12 および 2.13 の一部であったすべての機能が含まれています。詳細については、2.12 および 2.13 リリースノートを参照してください。	2024 年 5 月 21 日
Data OpenSearch Prepper バージョン 2.7 の Amazon Ingestion サポート	Amazon Ingestion OpenSearch に Data Prepper バージョン 2.7 のサポートが追加されました。詳細については、「2.7 リリースノート」を参照してください。	2024 年 4 月 4 日

AWS のサービス OpenSearch Serverless コレクションのプライベートアクセス	Amazon Bedrock AWS のサービスなどの特定の に、ネットワークアクセスポリシー内の OpenSearch Serverless コレクションへのアクセスを許可できるようになりました。	2024 年 3 月 28 日
インプレース EBS 更新	Amazon OpenSearch Service でブルー/グリーンデプロイを使用せずに、ドメインに EBS の変更を加えることができるようになりました。	2024 年 2 月 14 日
設定変更の可視性	Amazon OpenSearch Service コンソールおよび 設定 API を使用して、ドメイン設定の変更を追跡できるようになりました。	2024 年 2 月 6 日
ベクトル検索コレクションの一般提供	Amazon OpenSearch Serverless ベクトル検索コレクションが一般利用可能になりました。プレビューフェーズ中、以下の重要な改善が行われました。 • ベクトル検索コレクションは、それぞれ最大 128 次元の数十億のベクトルを使用するワークロードをサポートするようになりました。 • OpenSearch ダッシュボードでベクトル検索コレクションがサポートされるようになりました。	2023 年 11 月 29 日

OR1 インスタンス	Amazon OpenSearch Service で OR1 インスタンスタイプがサポートされるようになりました。	2023 年 11 月 29 日
Amazon S3 でのダイレクトクエリ (プレビュー)	ダイレクトクエリは、トランザクションデータが Amazon S3 バケットに書き込まれてから数秒以内に Amazon OpenSearch Service でトランザクションデータを利用できるようにするためのフルマネージドソリューションを提供します。 Amazon S3	2023 年 11 月 29 日
時系列コレクション用の 10 TiB のキャパシティ	Amazon OpenSearch Serverless は、時系列コレクションに対して最大 10 TiB のインデックスデータのサポートを追加します。また、このリリースは、すべてのタイプのコレクションのために 200 OCU の最大許容キャパシティと、コレクションの作成時にスタンバイレプリカを無効にする機能をサポートしています。	2023 年 11 月 29 日
OpenSearch 2.11 のサポート	Amazon OpenSearch Service が OpenSearch バージョン 2.11 をサポートするようになりました。このバージョンには、バージョン 2.10 と 2.11 に含まれていた機能がすべて含まれています。詳細については、2.10 と 2.11 のリリースノートを参照してください。	2023 年 11 月 17 日

Data OpenSearch Prepper バージョン 2.6 の Amazon Ingestion サポート	Amazon Ingestion OpenSearch に Data Prepper バージョン 2.6 のサポートが追加されました。詳細については、「2.6 リリースノート」を参照してください。さらに、パイプラインソースとして Amazon DynamoDB を指定できます。詳細については、「Amazon DynamoDB OpenSearch での取り込みパイプラインの使用」を参照してください。	2023 年 11 月 17 日
Data OpenSearch Prepper バージョン 2.5 の Amazon Ingestion サポート	Amazon Ingestion OpenSearch で Data Prepper バージョン 2.5 のサポートが追加されました。詳細については、「2.5 リリースノート」を参照してください。さらに、 OpenSearch サービスドメインまたは OpenSearch サーバーレスコレクションをパイプラインソースとして指定できるようになりました。詳細については、Data Prepper ドキュメントのOpenSearch「ソースプラグイン」を参照してください。	2023 年 11 月 17 日

CloudFormation リモート推論用の テンプレート	セマンティック検索用のリモート推論のセットアップを容易にするために、Amazon OpenSearch Service はモデルプロビジョニングプロセスを自動化する AWS CloudFormation テンプレートをコンソールで提供します。	2023 年 11 月 7 日
サービスにリンクされたロールポリシーの更新	サービスにリンクされたロールポリシー `AmazonOpenSearchServiceRolePolicy` が IPv6 アドレスの割り当てと割り当て解除を行うために必要なアクセス許可が追加されました。廃止された Elasticsearch ポリシー `AmazonElasticsearchServiceRolePolicy` も下位互換性を確保するために更新されました。	2023 年 10 月 26 日
Amazon OpenSearch Serverless ライフサイクルポリシー	Amazon OpenSearch Serverless では、データの保持と削除の管理を効率化するためのインデックスライフサイクルポリシーが導入されています。API またはコンソールの設定インターフェイスを使用して「時系列」コレクションのデータ保持ポリシーを設定できるようになったため、日次インデックスを作成する必要や古いデータを削除するためのスクリプトを作成する必要がなくなりました。	2023 年 10 月 25 日

Im4gn インスタンスのサポート	Amazon OpenSearch Service で Im4gn インスタンスタイプがサポートされるようになりました。IM4gn インスタンスは、大規模なデータセットを管理し、vCPU あたりの高いストレージ密度を必要とするワークロード向けに最適化されています。	2023 年 10 月 20 日
管理オプション	Amazon OpenSearch Service では、ドメインに関する問題のトラブルシューティングが必要な場合にきめ細かな制御を提供する複数の管理オプションが提供されるようになりました。これらのオプションには、データノードで OpenSearch プロセスを再起動する機能や、データノードを再起動する機能が含まれます。	2023 年 10 月 17 日
オプションプラグイン	Amazon OpenSearch Service では、Nori (韓国語)、Sudachi (日本語)、Pinyin (中国語)、STConvert Analysis (中国語)、および Amazon Personalize Search Ranking プラグインの 4 つの新しい言語アナライザープラグインのサポートが追加されました。	2023 年 10 月 16 日

OpenSearch 2.9 サポート	Amazon OpenSearch Service で OpenSearch バージョン 2.9 がサポートされるようになりました。このバージョンには、バージョン 2.8 と 2.9 に含まれていた機能がすべて含まれています。詳細については、2.8 と 2.9 のリリースノートを参照してください。	2023 年 10 月 2 日
ML コネクタ	Amazon OpenSearch Service は、機械学習 (ML) コネクタのサポートを追加します。コネクタは、他の AWS のサービスまたはサードパーティーの機械学習 (ML) プラットフォームでホストされている ML モデルへのアクセスを容易にします。	2023 年 9 月 6 日
Amazon Ingestion OpenSearch が Data Prepper バージョン 2.4 のサポートを追加	Amazon Ingestion OpenSearch で Data Prepper バージョン 2.4 のサポートが追加されました。詳細については、「2.4 リリースノート」を参照してください。さらに、パイプラインソースとして Amazon Managed Streaming for Apache Kafka (Amazon MSK) を指定できるようになりました。	2023 年 8 月 31 日

時系列コレクション用の 6 TiB の容量	Amazon OpenSearch Serverless は、時系列コレクションに対して最大 6 TiB のインデックスデータのサポートを追加します。このリリースでは、検索と時系列の両方のコレクションで最大 100 の OCU をサポートします。	2023 年 8 月 15 日
ベクトル検索コレクション	Amazon OpenSearch Serverless には、ベクトル検索コレクションを作成するオプションが追加されました。これを使用してベクトル埋め込みを保存し、類似性検索とセマンティック検索を強化できます。	2023 年 7 月 26 日
OpenSearch 2.7 サポート	Amazon OpenSearch Service が OpenSearch バージョン 2.7 をサポートするようになりました。このバージョンには、バージョン 2.6 と 2.7 に含まれていた機能がすべて含まれています。詳細については、2.6 と 2.7 のリリースノートを参照してください。	2023 年 7 月 10 日
Data Prepper 2.3 のサポート	Amazon Ingestion OpenSearch に Data Prepper バージョン 2.3 のサポートが追加されました。詳細については、「2.3 リリースノート」を参照してください。さらに、Amazon Security Lake をパイプラインソースとして指定できるようになりました。	2023 年 6 月 26 日

Multi-AZ with Standby	Amazon OpenSearch Service は、3 つのアベイラビリティーゾーン (AZsにドメインをデプロイするオプションを追加します。各 AZ にはデータの完全なコピーが含まれ、これらの AZsとして機能します。Multi-AZ with Standby のデプロイオプションにより、インフラストラクチャに障害が発生した場合でも、99.99% の可用性と一貫性のあるパフォーマンスを実現します。	2023 年 5 月 3 日
新しいサービスにリンクされたロール	Amazon OpenSearch Service は、 というサービスにリンクされたロールを追加します。これにより`AWSServiceRoleForAmazonOpenSearchIngestionService` 、Amazon Ingestion OpenSearch はメトリクスデータを に送信できます Amazon CloudWatch。	2023 年 4 月 26 日

Amazon OpenSearch Ingestion	Amazon Ingestion OpenSearch は、リアルタイムのログとトレースデータを OpenSearch サービスドメインと OpenSearch サーバーレスコレクションに配信するフルマネージドデータコレクターです。 OpenSearch Ingestion を使用すると、Logstash や Jaeger などのサードパーティーソリューションを使用してドメインやコレクションにデータを取り込む必要がなくなります。	2023 年 4 月 26 日
OpenSearch 2.5 サポート	Amazon OpenSearch Service で OpenSearch バージョン 2.5 がサポートされるようになりました。このバージョンには、バージョン 2.4 と 2.5 に含まれていた機能がすべて含まれています。詳細については、2.4 と 2.5 のリリースノートを参照してください。	2023 年 3 月 13 日

オフピークのメンテナンスウィンドウ	Amazon OpenSearch Service はオフピークウィンドウを追加します。オフピークウィンドウは、毎日 10 時間のトラフィックが少ないタイムブロックで、ブルー/グリーンデプロイを必要とするサービスソフトウェアの更新と Auto-Tune 最適化をスケジュールできます。オフピークの時間帯に更新を行うことにより、トラフィックの多い時間帯にクラスターの専用マスターノードにかかる負荷を、最小限に抑えることができます。 2 月 16 日以降に作成された新しいドメインの場合、オフピークウィンドウは現地時間の午後 10 時から午前 8 時の間に自動的に設定されます。既存のドメインでは、このウィンドウを明示的に有効にする必要があります。	2023 年 2 月 16 日
ドメイン作成時に SAML 認証を設定する	Amazon OpenSearch Service は、ドメイン作成時の SAML 認証の設定をサポートするようになりました。これまでは、ドメインを既に作成した後で SAML オプションを設定する必要がありました。	2023 年 2 月 1 日

VPC ドメインのリモート再インデックス	Amazon OpenSearch Service は、2 つのドメイン間の VPC エンドポイント接続のオプションを追加します。リモート再インデックスを使用して、リバースプロキシを使用することなく、1 つの VPC ドメインから別の VPC ドメインにインデックスをコピーできるようになりました。この機能を使用するには、VPC ドメインがサービスソフトウェア R20221114 以降を実行している必要があります。	2023 年 1 月 31 日

Amazon OpenSearch Serverless の一般提供	Amazon OpenSearch Serverless が一般公開されました。プレビューフェーズ中、以下の重要な改善が行われました。 • コレクションエンドポイントのトラフィックが減少した場合に、最小限に設定された OCU まで容量をスケールダウンできるようになりました。 • インデックス作成と検索両方の最大許容 OCU が 20 から 50 に引き上げられました。各 OCU には、120 GiB のインデックスデータを保存するのに十分なホットエフェメラルストレージが含まれています。 • データアクセス設定を、個別のワークフローで実行するのではなく、コレクションの作成中に実行できるようになりました。	2023 年 1 月 25 日
非同期ドライラン	Amazon OpenSearch Service では、非同期ドライランがサポートされるようになりました。これにより、設定変更を行う前に検証チェックを実行し、変更によってブルー/グリーンデプロイが発生するかどうかを通知します。	2023 年 1 月 19 日

新しいサービスにリンクされたロール	Amazon OpenSearch Service は、 というサービスにリンクされたロールを追加します。これにより`AWSServiceRoleForAmazonOpenSearchServerless` 、OpenSearch サーバーレスはメトリクスデータを に送信できます Amazon CloudWatch。	2022 年 11 月 29 日
Amazon OpenSearch Serverless プレビュー	Amazon OpenSearch Serverless は、Amazon OpenSearch Service のオンデマンド、自動スケーリング、サーバーレス設定です。Serverless は、 OpenSearch クラスターのプロビジョニング、設定、チューニングの運用上の複雑さを排除します。	2022 年 11 月 29 日
OpenSearch 2.3 サポート	Amazon OpenSearch Service で OpenSearch バージョン 2.3 がサポートされるようになりました。このバージョンには、バージョン 2.0、2.1、2.2 に含まれているすべての機能が含まれています。詳細については、2.0、2.1、2.2、2.3 リリースノートを参照してください。バージョン 2.3 には重大な変更が含まれています。詳細については、「サポートされているアップグレードパス」を参照してください。	2022 年 11 月 15 日

通知プラグインのサポート	Amazon OpenSearch Service は、 通知プラグインをサポートするようになりました。これにより、 OpenSearch プラグインからのすべての通知を一元的に処理できます。バージョン 2.0 以降、アラート送信先は廃止され、通知チャネルに置き換えられました。	2022 年 11 月 15 日
Kibana 7.1.1 のサポート	Elasticsearch 7.1 を実行している Amazon OpenSearch Service ドメインが Kibana 7.1.1 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。7.1 ドメインをサービスソフトウェア R20221114 に更新すると、 OpenSearch サービスは自動的にこのパッチリリースにアップグレードします。	2022 年 11 月 15 日

Kibana 6.8.13 のサポート	Elasticsearch 6.8 を実行している Amazon OpenSearch Service ドメインは、Kibana 6.8.13 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。6.8 ドメインをサービスソフトウェア R20221114 に更新すると、OpenSearch サービスは自動的にこのパッチリリースにアップグレードします。	2022 年 11 月 15 日
Kibana 6.3.2 のサポート	Elasticsearch 6.3 を実行している Amazon OpenSearch Service ドメインが Kibana 6.3.2 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。6.3 ドメインをサービスソフトウェア R20221114 に更新すると、 OpenSearch Service はこのパッチリリースに自動的にアップグレードします。	2022 年 11 月 15 日

AWS PrivateLink	Amazon OpenSearch Service が管理する VPC エンドポイントを使用すると、インターネット経由で接続するのではなく、インターフェイス VPC エンドポイントを使用して OpenSearch サービス VPC ドメインに直接接続できます。OpenSearch サービスマネージド VPC エンドポイントは、ルートテーブルとセキュリティグループで許可されているように、エンドポイントがプロビジョニングされている VPC 内、またはエンドポイントがプロビジョニングされている VPC と VPCs ピアリング接続されている VPC からのみアクセスできます。VPC ドメインがインターフェイス VPC エンドポイントに接続するには、サービスソフトウェア R20220928 以降を実行している必要があります。	2022 年 11 月 7 日
バグ修正とパフォーマンス向上	サービスソフトウェア R20220928には、バグの修正とSAML口ギングの改善を含むパフォーマンスの向上が含まれています。また、この更新により、デフォルトのテナントが `Private` から `Global` に変更されます。	2022 年 10 月 3 日

API リファレンスの改善	Amazon OpenSearch Service は、改善されたオールエンコパス設定 API リファレンスを提供します。新しいリファレンスには、使用可能なすべてのアクションとデータタイプ、サンプルリクエストとレスポンスの構文、サポートされているすべての言語の対応する SDK リファレンスへのリンクが含まれています。	2022 年 9 月 13 日
ブルー/グリーン検証	Amazon OpenSearch Service は、ブルー/グリーンデプロイの前に検証チェックを実行し、ドメインが更新の対象でない場合は検証エラーを表示するようになりました。	2022 年 8 月 16 日
OpenSearch 1.3 サポート	Amazon OpenSearch Service で OpenSearch バージョン 1.3 がサポートされるようになりました。詳細については、「1.3 リリースノート」を参照してください。	2022 年 7 月 27 日
ML Commons プラグインのサポート	Amazon OpenSearch Service は、トランスポートおよび REST API コールを通じて一般的な機械学習アルゴリズムのセットを提供する ML Commons プラグインのサポートを追加します。また、PPL コマンドを通じて ML Commons プラグインを操作することもできます。	2022 年 7 月 27 日

gp3 ボリュームのサポート	Amazon OpenSearch Service は、gp3EBS 汎用 SSD ボリュームタイプのサポートを追加します。ドメインを作成または変更するときに、プロビジョンド IOPS とスループットをさらに指定できます。	2022 年 7 月 26 日
強化されたベストプラクティスに関するドキュメント	Amazon OpenSearch Service ドキュメントには、OpenSearch サービスドメインの作成と運用に関する運用上のベストプラクティスと一般的な推奨事項が記載されています。	2022 年 7 月 6 日
Service Quotas との統合	Service Quotas コンソールから Amazon OpenSearch Service のクォータを表示し、クォータの引き上げをリクエストできるようになりました。	2022 年 1 月 29 日
OpenSearch API のタグベースのアクセスコントロール	タグを使用して OpenSearch APIsへのアクセスを制御できるようになりました。以前は、タグを使用して設定 API へのアクセスを制御することしかできませんでした。	2022 年 6 月 16 日

リージョン間のクロスクラスター検索	クラスター間検索は AWS リージョン 、両方のドメインが Elasticsearch バージョン 7.10 以降、または の任意のバージョンを実行している限り、 でサポートされるようになりましたOpenSearch。	2022 年 6 月 14 日
単一の Kibana 5.6 のサポート	Amazon OpenSearch Service は、単一の Kibana 5.6.16 のサポートを追加します。単一の Kibana 5.6.16 では、Elasticsearch 5.1、5.3、5.5、および 5.6 に接続しながら、フロントエンドとして Kibana 5.6 を使用できます。単一の Kibana 5.6 を使用するには、サービスソフトウェア R20220323 以降が必要です。	2022 年 4 月 4 日
R20220323-P1	Amazon OpenSearch Service は最近、サービスソフトウェア更新 R20220323 をリリースしましたが、問題により更新がロールバックされました。ドメインをパッチリリース R20220323-P1 以降に更新することをお勧めします。これにより、問題が解決されます。	2022 年 4 月 4 日
OpenSearch 1.2 サポート	Amazon OpenSearch Service が OpenSearch バージョン 1.2 をサポートするようになりました。詳細については、「1.2 リリースノート」を参照してください。	2022 年 4 月 4 日

可観測性	OpenSearch Dashboards for Amazon OpenSearch Service のデフォルトインストールには、オブザーバビリティプラグインが含まれています。オブザーバビリティプラグインを使用すると、パイプ処理言語 (PPL) を使用してデータ駆動型イベントを視覚化し、データを探索してクエリできます。プラグインには OpenSearch 1.2 以降とサービスソフトウェア R20220323 以降が必要です。	2022 年 4 月 4 日
Kibana 7.7.1 のサポート	Elasticsearch 7.7 を実行している Amazon OpenSearch Service ドメインが Kibana 7.7 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。7.7 ドメインをサービスソフトウェア R20220323 以降に更新すると、 OpenSearch Service はそれらをこのパッチリリースに自動的にアップグレードします。	2022 年 4 月 4 日

JVM メモリプレッシャーメトリクスの変更	Amazon OpenSearch Service は、メモリ使用率をより正確に反映するようにJVMMemory Pressure CloudWatch メトリクスのロジックを変更しました。以前は、メトリクスでは JVM ヒープの旧世代のメモリプールのみが考慮されていました。この変更により、メトリクスでは新しい世代のメモリープールも考慮されます。ドメインをサービスソフトウェア R20220323 に更新すると、JVMMemory Pressure 、MasterJVM MemoryPressure 、および WarmJVMMemoryPressure メトリクスで増加がみられることがあります。	2022 年 4 月 4 日
IK (中国語) 分析プラグインを使用したカスタム辞書	Amazon OpenSearch Service は、IK (中国語) 分析プラグインでのカスタムディクショナリの使用をサポートするようになりました。	2022 年 4 月 4 日

既存のドメインでのクラスター間レプリケーション	Amazon OpenSearch Service では、2020 年 6 月 3 日以降に作成されたドメインにのみクラスター間検索とクラスター間レプリケーションを実装できるという制限がなくなりました。いつ作成されたかに関係なく、これらの機能をすべてのドメインで有効化できるようになりました。両方のドメインがサービスソフトウェア R20220323 以降である必要があります。	2022 年 4 月 4 日
Blue/Green デプロイの可視性	Amazon OpenSearch Service では、ブルー/グリーンデプロイの進行状況をより詳細に把握できるようになりました。これらの詳細は、コンソールまたは Configuration API を使用してモニタリングできます。	2022 年 1 月 27 日

既存のドメインでのきめ細かなアクセスコントロール	きめ細かなアクセスコントロールを既存のドメインに対して有効にできるようになりました。Open/IP ベースのアクセスポリシーの一時的な移行期間を有効にして、ロールの作成とマッピング中にユーザーが引き続きドメインにアクセスできるようにすることができます。きめ細かなアクセスコントロールを既存のドメインで有効にするには、サービスソフトウェア R20211203 以降が必要です。	2022 年 1 月 6 日
OpenSearch Dashboards ロールの名前を変更しました	サービスソフトウェア R20211203 で、`kibana_user` ロールが `opensearch_dashboards_user` に、`kibana_read_only` が `opensearch_dashboards_read_only` に名称変更されました。この変更は、新しく作成されたすべての 1.x OpenSearch ドメインに適用されます。サービスソフトウェア R20211203 にアップグレードした既存の OpenSearch ドメインの場合、ロールは変わりません。	2022 年 1 月 4 日

OpenSearch 1.1 サポート	Amazon OpenSearch Service で OpenSearch バージョン 1.1 がサポートされるようになりました。詳細については、「1.1 リリースノート」を参照してください。	2022 年 1 月 4 日
ISM ビジュアルエディタ	OpenSearch Dashboards for Amazon OpenSearch Service のデフォルトインストールでは、ISM ポリシーのビジュアルエディタがサポートされるようになりました。この機能には OpenSearch 1.1 以降が必要です。	2022 年 1 月 4 日
サービス間の混乱した代理防止の更新	Amazon OpenSearch Service は、混乱した代理問題を防ぐために、IAM 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の使用をサポートしています。これらの条件キーを使用するには、サービスソフトウェア R20211203 以降にある必要があります。	2022 年 1 月 4 日

Log4j パッチ	サービスソフトウェア R20211203-P2 は、CVE-2021-4422844228 および CVE-2021-45046 のアドバイザリによって推奨されているように、サービスで使用される Log4j のバージョンを更新します。 OpenSearch CVE-2021-45046 パッチは、OpenSearch および Elasticsearch のすべてのバージョンを実行しているドメインに適用されます。 OpenSearch サービスは引き続きさまざまな Log4j バージョンを内部的に更新します。また、最新バージョンの Log4j に制限されるとは限りません。ドメインの Log4j のバージョンは、ドメインが実行しているソフトウェアのバージョンによって異なります。ただし、Log4j のバージョンにかかわらず、R20211203-P2 以降を実行している限り、ドメインには、CVE-2021-44228 および CVE-2021-45046 に対処するために必要な Log4j の更新が含まれています。	2021 年 12 月 15日

クラスター間レプリケーション	クラスター間レプリケーションを使用すると、あるOpenSearch サービスドメインから別のサービスドメインにインデックス、マッピング、メタデータをレプリケートできます。クラスター間レプリケーションには、Elasticsearch 7.10 またはOpenSearch 1.1 以降を実行するドメインが必要です。	2021 年 10 月 5 日
新しい AWSマネージドポリシー	Amazon OpenSearch Service の起動には、新しい AWS管理ポリシーと古いポリシーの非推奨が含まれます。	2021 年 9 月 8 日
Kibana 6.4.3 のサポート	レガシー Elasticsearch バージョン 6.4 を実行しているAmazon OpenSearch Service ドメインが Kibana 6.4 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。OpenSearch サービスは、ドメインをこのパッチリリースに自動的にアップグレードします。	2021 年 9 月 8 日

データストリーム	Amazon OpenSearch Service は、時系列データの管理プロセスを簡素化するデータストリームのサポートを追加します。データストリームを使用するには、ドメインが OpenSearch 1.0 以降を実行している必要があります。	2021 年 9 月 8 日
Amazon OpenSearch サービス	AWS は Amazon OpenSearch Service の名前を変更して、従来の「Elasticsearch」ブランドを削除します。Amazon OpenSearch Service は、OpenSearch およびレガシー Elasticsearch OSS をサポートしています。クラスターを作成するときは、使用する検索エンジンを選択できます。 OpenSearch サービスは、ソフトウェアの最終オープンソースバージョンである Elasticsearch OSS 7.10 との幅広い互換性を提供します。	2021 年 9 月 8 日

コールドストレージ	コールドストレージは、アクセス頻度の低いデータまたは履歴データ用の新しいストレージ階層です。コールドインデックスは、S3 ストレージのみを占有し、コンピューティングはアタッチされません。コールドストレージには、Elasticsearch 7.9 以降を実行しているドメインとサービスソフトウェア R20210426 以降が必要です。	2021 年 5 月 13 日
ARM ベースの Graviton インスタンス	Amazon OpenSearch Service は、ARM ベースの Graviton インスタンスタイプ (M6G、C6G, R6G、および R6GD) をサポートするようになりました。Graviton インスタンスタイプは、Elasticsearch 7.9 以降を実行している新規および既存のドメインおよびサービスソフトウェア R20210331 以降で使用できます。	2021 年 5 月 4 日

ISM テンプレート	Amazon OpenSearch Service は ISM テンプレートのサポートを追加します。これにより、インデックスがポリシーで定義されたパターンと一致する場合に、ISM ポリシーをインデックスに自動的にアタッチできます。ISM テンプレートには、サービスソフトウェア R20210426 以降が必要です。この更新プログラムでは、`policy_id` 設定も非推奨にしました。つまり、インデックステンプレートを使用して、新しく作成されたインデックスに ISM ポリシーを適用できなくなります。この更新では、この設定を使用して既存の CloudFormation テンプレートに重大な変更が導入されました。	2021 年 4 月 27 日
Elasticsearch 7.10 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 7.10 がサポートされるようになりました。詳細については、7.10 リリースノートを参照してください。	2021 年 4 月 21 日

非同期検索	Amazon OpenSearch Service で非同期検索がサポートされるようになりました。これにより、検索リクエストをバックグラウンドで実行できます。非同期検索には、Elasticsearch 7.10 以降を実行しているドメインとサービスソフトウェア R20210331 以降が必要です。	2021 年 4 月 21 日
設定 API 向けのタグベースのアクセスコントロール	AWS タグを使用して Amazon ES 設定 API へのアクセスを制御できるようになりました。	2021 年 3 月 2 日
Auto-Tune	Amazon OpenSearch Service は Auto-Tune を追加します。Auto-Tune は、クラスターのパフォーマンスと使用状況のメトリクスを使用して、ノードの JVM 設定の変更を提案します。Auto-Tune には、Elasticsearch 6.7 以降を実行しているドメインとサービスソフトウェア R20201117 以降が必要です。	2021 年 2 月 24 日

トレース分析	Amazon OpenSearch Service 用 Kibana のデフォルトインストールには、分散アプリケーションからのトレースデータをモニタリングできるトレース分析プラグインが含まれるようになりました。プラグインには、Elasticsearch 7.9 以降を実行しているドメインとサービスソフトウェア R20210201 以降が必要です。	2021 年 2 月 17 日
シャードのメトリクス	Amazon OpenSearch Service は、シャードステータスを追跡するための CloudWatch メトリクス `Shards.active` 、`Shards.unassigned` 、、`Shards.delayedUnassigned` `Shards.activePrimary` `Shards.initializing` 、 を追加しますShards.relocating 。このメトリクスは、サービスソフトウェア R20210201 以降を実行しているドメインで使用できます。	2021 年 2 月 17 日

Kibana レポート	Amazon OpenSearch Service 用 Kibana のデフォルトインストールでは、検出、視覚化、ダッシュボードページのオンデマンドレポートがサポートされるようになりました。この機能には、Elasticsearch 7.9 以降およびサービスソフトウェア R20210201 以降が必要です。	2021 年 2 月 17 日
Kibana 5.6.16 のサポート	Elasticsearch 5.6 を実行している Amazon OpenSearch Service ドメインが Kibana 5.6 の最新パッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。Amazon ES は、ドメインをこのパッチリリースに自動的にアップグレードします。	2021 年 2 月 17 日
既存のドメインの暗号化	Amazon OpenSearch Service は、Elasticsearch 6.7 以降を実行している既存のドメインで、 node-to-node保管中のデータの暗号化の有効化をサポートするようになりました。これらの設定を有効にした後、無効にすることはできません。	2021 年 1 月 27 日

リモート再インデックス	Amazon OpenSearch Service でリモート再インデックスがサポートされるようになりました。これにより、リモートドメインからインデックスを移行できます。この機能には、サービスソフトウェア R20201117 以降が必要です。	2020 年 11 月 24 日
パイプ処理言語	Amazon OpenSearch Service は、パイプ処理言語 (PPL) をサポートするようになりました。PPL は、パイプ (|) 構文を使用して Elasticsearch に保存されているデータをクエリできるクエリ言語です。この機能には、サービスソフトウェア R20201117 以降が必要です。詳細については、「」を参照してください。	2020 年 11 月 24 日
Kibana ノートブック	Amazon OpenSearch Service では、Kibana ノートブックのサポートが追加されました。これにより、ライブビジュアライゼーションと説明文テキストを 1 つのインターフェイスにまとめることができます。この機能には、サービスソフトウェア R20201117 以降が必要です。	2020 年 11 月 24 日

ガントチャート	Amazon OpenSearch Service 用 Kibana のデフォルトインストールでは、新しい視覚化タイプであるガントグラフがサポートされるようになりました。この機能には、サービスソフトウェア R20201117 以降が必要です。	2020 年 11 月 24 日
Elasticsearch 7.9 のサポート	Amazon OpenSearch Service は Elasticsearch バージョン 7.9 をサポートするようになりました。詳細については、7.9 リリースノートを参照してください。	2020 年 11 月 24 日
異常検出の更新	Amazon OpenSearch Service の異常検出により、高基数のサポートが追加され、IP アドレス、製品 ID、国コードなどのディメンションで異常を分類できます。この機能には、サービスソフトウェア R20201117 以降が必要です。	2020 年 11 月 24 日

動的辞書の更新	Amazon OpenSearch Service では、インデックスを再作成せずに検索アナライザーを更新できるようになりました。ドメインの一部またはすべてのドメインで辞書ファイルを更新できます。また、Amazon ES がパッケージのバージョンを長期的に追跡するので、変更内容と変更時期の履歴が得られるようになります。この機能には、サービスソフトウェア R20201019 以降が必要です。	2020 年 11 月 17 日
カスタムエンドポイント	Amazon OpenSearch Service はカスタムエンドポイントをサポートするようになりました。これにより、Amazon ES ドメインに新しい URL を付与できます。ドメインを入れ替える場合は、同じ URL を維持できます。この機能には、サービスソフトウェア R20201019 以降が必要です。	2020 年 11 月 5 日
新しい言語プラグイン	Amazon OpenSearch Service は、サービスソフトウェア R20201019 以降で Elasticsearch 7.7 以降を実行しているドメインで IK (中国語) 分析、ベトナム語分析、タイ語分析プラグインをサポートするようになりました。	2020 年 10 月 28 日

Elasticsearch 7.8 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 7.8 がサポートされるようになりました。詳細については、7.8 リリースノートを参照してください。	2020 年 10 月 28 日
Kibana の SAML 認証	Amazon OpenSearch Service は Kibana の SAML 認証をサポートするようになりました。これにより、サードパーティーの ID プロバイダーを使用して Kibana にログインし、きめ細かなアクセスコントロールを管理し、データを検索し、視覚化を構築できます。この機能には、サービスソフトウェア R20201019 以降が必要です。	2020 年 10 月 27 日
T3 インスタンス	Amazon OpenSearch Service は、 `t3.small` および `t3.medium` インスタンスタイプをサポートするようになりました。	2020 年 9 月 23 日
監査ログ	Amazon OpenSearch Service では、データの監査ログがサポートされるようになりました。これにより、失敗したログイン試行、インデックス、ドキュメント、フィールドへのユーザーアクセスなどを追跡できます。この機能には、サービスソフトウェア R20200910 以降が必要です。	2020 年 9 月 16 日

UltraWarm 更新	UltraWarm for Amazon OpenSearch Service は、新しいメトリクス、新しい設定、より大きな移行キュー、およびキャンセル API を追加します。これらの更新には、サービスソフトウェア R20200910 以降が必要です。詳細については、「」を参照してください。	2020 年 9 月 14 日
Learning to Rank	Amazon OpenSearch Service は、オープンソースの Learning to Rank プラグインをサポートするようになりました。これにより、機械学習テクノロジーを使用して検索の関連性を改善できます。この機能には、サービスソフトウェア R20200721 以降が必要です。	2020 年 7 月 27 日
k-NN コサイン類似度	K 最近傍 (k-NN) では、ユークリッド距離に加えて、コサイン類似度によって「最近隣接」を検索できるようになりました。この機能には、サービスソフトウェア R20200721 以降が必要です。	2020 年 7 月 23 日

gzip 圧縮	Amazon OpenSearch Service は、ほとんどの HTTP リクエストとレスポンスで gzip 圧縮をサポートするようになりました。これにより、レイテンシーが短縮され、帯域幅を節約できます。この機能には、サービスソフトウェア R20200721 以降が必要です。	2020 年 7 月 23 日
Elasticsearch 7.7 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 7.7 がサポートされるようになりました。詳細については、7.7 リリースノートを参照してください。	2020 年 7 月 23 日
Kibana マップサービス	Amazon OpenSearch Service 用 Kibana のデフォルトインストールには、インドおよび中国リージョンのドメインを除き、WMS マップサーバーが含まれるようになりました。	2020 年 6 月 18 日
SQL の改良点	Amazon OpenSearch Service の SQL サポートは、多くの新しいオペレーション、データ探索専用の Kibana ユーザーインターフェイス、インタラクティブ CLI をサポートするようになりました。詳細については、「」を参照してください。	2020 年 6 月 3 日

クラスター間検索	Amazon OpenSearch Service では、接続された複数のドメインでクラスター間のクエリと集約を実行できます。	2020 年 6 月 3 日
異常検出	Amazon OpenSearch Service では、異常をほぼリアルタイムで自動的に検出できます。	2020 年 6 月 3 日
UltraWarm	UltraWarm Amazon OpenSearch Service の ストレージはパブリックプレビューを終了し、一般公開されました。この機能は、より広範なバージョン および をサポートするようになりましたAWS リージョン。詳細については、「」を参照してください。	2020 年 5 月 5 日
カスタム辞書	Amazon OpenSearch Service では、クラスターで使用するカスタムディクショナリファイルをアップロードできます。これらのファイルは、特定の高頻度の単語を無視するか、同等の単語として扱うように Elasticsearch に指示することで、検索結果を改善します。	2020 年 4 月 21 日
Elasticsearch 7.4 のサポート	Amazon OpenSearch Service は Elasticsearch バージョン 7.4 をサポートするようになりました。詳細については、「サポートされるバージョン」を参照してください。	2020 年 3 月 12 日

k-NN	Amazon OpenSearch Service では、k-Narest Neighbor (k-NN) 検索のサポートが追加されました。k-NN にはサービスソフトウェア R20200302 以降が必要です。	2020 年 3 月 3 日
インデックスステート管理	Amazon OpenSearch Service は、インデックス状態管理 (ISM) を追加します。これにより、インデックスが特定の期間に達したときにインデックスを削除するなどのルーチンタスクを自動化できます。この機能には、サービスソフトウェア R20200302 以降が必要です。	2020 年 3 月 3 日
Elasticsearch 5.6.16 のサポート	Amazon OpenSearch Service は、バージョン 5.6 の最新のパッチリリースをサポートするようになりました。これにより、バグ修正が追加され、セキュリティが向上します。Amazon ES は、既存の 5.6 ドメインをこのリリースに自動的にアップグレードします。この Elasticsearch リリースでは、バージョンが 5.6.17 と誤って報告されることに注意してください。	2020 年 3 月 2 日

きめ細かなアクセスコントロール	Amazon OpenSearch Service は、インデックス、ドキュメント、フィールドレベルでのセキュリティ、Kibana マルチテナンシー、クラスターのオプション HTTP 基本認証を提供するきめ細かなアクセスコントロールをサポートするようになりました。	2020 年 2 月 11 日
UltraWarm ストレージ (プレビュー)	Amazon OpenSearch Service は UltraWarm、Amazon S3 を使用する新しいウォームストレージ階層である と、パフォーマンスを向上させるための高度なキャッシュソリューションを追加します。アクティブに書き込みやクエリをあまり行っていないインデックスの場合、 UltraWarm ストレージは GiB あたりのコストを大幅に削減します。	2019 年 12 月 3 日
中国リージョンの暗号化機能	保管中のデータの暗号化と node-to-node 暗号化が、`cn-north-1` 中国 (北京) リージョンと`cn-northwest-1` 中国 (寧夏) リージョンで利用可能になりました。	2019 年 11 月 20 日

HTTPS が必要	Amazon ES ドメインへのすべてのトラフィックが HTTPS 経由で到着することを要求できるようになりました。ドメインを設定するときは、[HTTPS が必要] チェックボックスをオンにします。この機能には、サービスソフトウェア R20190808 以降が必要です。	2019 年 10 月 3 日
Elasticsearch 7.1 および 6.8 のサポート	Amazon OpenSearch Service は、Elasticsearch バージョン 7.1 および 6.8 をサポートするようになりました。詳細については、「サポートされるバージョン」を参照してください。	2019 年 8 月 13 日
毎時スナップショット	Amazon OpenSearch Service は、毎日のスナップショットではなく、Elasticsearch 5.3 以降を実行しているドメインのスナップショットを 1 時間ごとに取得するようになりました。これにより、データを復元するバックアップの頻度が高くなります。	2019 年 7 月 8 日
Elasticsearch 6.7 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 6.7 がサポートされるようになりました。詳細については、「サポートされるバージョン」を参照してください。	2019 年 5 月 29 日

SQL のサポート	Amazon OpenSearch Service では、SQL を使用してデータをクエリできるようになりました。SQL のサポートには、サービスソフトウェア R20190418 以降が必要です。	2019 年 5 月 15 日
5 つのシリーズのインスタンスタイプ	Amazon OpenSearch Service は、M5, C5、および R5 インスタンスタイプをサポートするようになりました。旧世代のインスタンスタイプと比較して、これらの新しいタイプは低価格で優れたパフォーマンスを提供します。詳細については、「制限」を参照してください。	2019 年 4 月 24 日
Elasticsearch 6.5 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 6.5 がサポートされるようになりました。	2019 年 4 月 8 日
アラート	1 つ以上の Amazon ES インデックスからのデータが特定の条件を満たすと、Amazon OpenSearch Service のアラートから通知されます。アラートには、サービスソフトウェア R20190221 以降が必要です。	2019 年 3 月 25 日

3 つのアベイラビリティーゾーンのサポート	Amazon OpenSearch Service は、多くの リージョンで 3 つのアベイラビリティーゾーンをサポートするようになりました。このリリースには、効率化されたコンソール機能も含まれています。このマルチ AZ には、サービスソフトウェア R20181023 以降が必要です。	2019 年 2 月 7 日
Elasticsearch 6.4 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 6.4 がサポートされるようになりました。	2019 年 1 月 23 日
200 ノードクラスター	Amazon ES では、ストレージが合計 3 PB の最大 200 個のデータノードを持つクラスターを作成できるようになりました。	2019 年 1 月 22 日
サービスソフトウェア更新	Amazon ES では、ドメインのサービスソフトウェアを手動で更新して、新機能を迅速に利用したり、トラフィックの少ない時間に更新したりできるようになりました。詳細については、「」を参照してください。	2018 年 11 月 20 日
新しい CloudWatch メトリクス	Amazon ES には、ノードレベルのメトリクスが追加され、新しい [クラスターヘルス] および [インスタンスヘルス] タブが Amazon ES コンソールに追加されました。	2018 年 11 月 20 日

中国 (北京) のサポート	Amazon OpenSearch Service は、M4, C4リージョンで利用可能になりました。 R4	2018 年 10 月 17 日
N ode-to-node 暗号化	Amazon OpenSearch Service で node-to-node 暗号化がサポートされるようになりました。これにより、Amazon ES がクラスター全体にデータを分散する際にデータが暗号化されます。	2018 年 9 月 18 日
インプレースバージョンアップグレード	Amazon OpenSearch Service でインプレースバージョンアップグレードがサポートされるようになりました。	2018 年 8 月 14 日
Elasticsearch 6.3 および 5.6 のサポート	Amazon OpenSearch Service は、Elasticsearch バージョン 6.3 および 5.6 をサポートするようになりました。	2018 年 8 月 14 日
エラーログ	Amazon ES では、Elasticsearch エラーログを Amazon に発行できるようになりました CloudWatch。	2018 年 7 月 31 日
中国 (寧夏) のリザーブドインスタンス	Amazon ES は、中国 (寧夏) リージョンでリザーブドインスタンスの提供を開始しました。	2018 年 5 月 29 日
リザーブドインスタンス	Amazon ES は、リザーブドインスタンスのサポートの提供を開始しました。	2018 年 5 月 7 日

以前の更新

以下のテーブルは、2018 年 5 月以前の Amazon ES の重要な変更点をまとめたものです。

変更	説明	日付
Kibana の Amazon Cognito 認証	Amazon ES で、Kibana のログインページの保護が追加されました。詳細については、「the section called “OpenSearch Dashboards の Amazon Cognito 認証”」を参照してください。	2018 年 4 月 2 日
Elasticsearch 6.2 のサポート	Amazon OpenSearch Service で Elasticsearch バージョン 6.2 がサポートされるようになりました。	2018 年 3 月 14 日
韓国語分析プラグイン	Amazon ES は、メモリ最適化バージョンの Seunjeon 韓国語分析プラグインをサポートするようになりました。	2018 年 3 月 13 日
アクセスコントロールの瞬時の更新	Amazon ES ドメインでのアクセスコントロールポリシーの変更は、すぐに有効化されるようになりました。	2018 年 3 月 7 日
ペタバイトスケール	Amazon ES は、I3 インスタンスタイプと合計で最大 1.5 PB のドメインストレージをサポートするようになりました。詳細については、「the section called “ペタバイトスケール”」を参照してください。	2017 年 12 月 19 日
保管時のデータの暗号化	Amazon ES は、保管中のデータの暗号化をサポートするようになりました。詳細については、「the section called “保管中の暗号化”」を参照してください。	2017 年 12 月 7 日
Elasticsearch 6.0 のサポート	Amazon ES は、Elasticsearch バージョン 6.0 をサポートするようになりました。移行に関する考慮事項と手順については、「the section called “ドメインのアップグレード”」を参照してください。	2017 年 12 月 6 日
VPC サポート	Amazon ES では、Amazon Virtual Private Cloud 内でドメインを起動できるようになりました。VPC サポートはセキュリティに追加のレイヤーを提供し、VPC における Amazon ES とその他のサービス間の通信を簡素化しま	2017 年 10 月 17 日

変更	説明	日付
	す。詳細については、「the section called “VPC サポート”」を参照してください。	
スローログの公開	Amazon ES では、 CloudWatch ログへのスローログの発行がサポートされるようになりました。詳細については、「the section called “ログをモニタリングする”」を参照してください。	2017 年 10 月 16 日
Elasticsearch 5.5 のサポート	Amazon ES は、Elasticsearch バージョン 5.5 をサポートするようになりました。 AWS Support に問い合わせることなく自動スナップショットを復元し、`_scripts` API を使用してスクリプトを保存できるようになりました。	2017 年 9 月 7 日
Elasticsearch 5.3 のサポート	Amazon ES は、Elasticsearch バージョン 5.3 のサポートを追加しました。	2017 年 6 月 1 日
クラスターあたりのインスタンスと EBS キャパシティーが増加	Amazon ES は、クラスターあたり最大 100 個のノードと 150 TB の EBS 容量をサポートするようになりました。	2017 年 4 月 5 日
カナダ (中部) と欧州 (ロンドン) のサポート	Amazon ES は、カナダ (中部) ca-central-1 および欧州 (ロンドン) eu-west-2 リージョンのサポートを追加しました。	2017 年 3 月 20 日
より多くのインスタンスと拡大した EBS ボリューム	Amazon ES は、より多くのインスタンスと拡大した EBS ボリュームのサポートを追加しました。	2017 年 2 月 21 日
Elasticsearch 5.1 のサポート	Amazon ES は、Elasticsearch バージョン 5.1 のサポートを追加しました。	2017 年 1 月 30 日
Phonetic Analysis プラグインのサポート	Amazon ES には、Phonetic Analysis プラグインとの統合が組み込まれるようになりました。これにより、データに対して「同音異義の」クエリ実行が可能になります。	2016 年 12 月 22 日

変更	説明	日付
米国東部 (オハイオ) のサポート	Amazon ES は、米国東部 (オハイオ) リージョン (us-east-2) のサポートを追加しました。	2016 年 10 月 17 日
新しいパフォーマンスメトリクス	Amazon ES が新しいパフォーマンスメトリクス `ClusterUsedSpace` を追加しました。	2016 年 7 月 29 日
Elasticsearch 2.3 のサポート	Amazon ES は、Elasticsearch バージョン 2.3 のサポートを追加しました。	2016 年 7 月 27 日
アジアパシフィック (ムンバイ) のサポート	Amazon ES は、アジアパシフィック (ムンバイ) ap-south-1 リージョンのサポートを追加しました。	2016 年 6 月 27 日
クラスターあたりのさらなるインスタンス	Amazon ES は、クラスターごとのインスタンス (インスタンス数) の最大数を 10 から 20 に増やしました。	2016 年 5 月 18 日
アジアパシフィック (ソウル) のサポート	Amazon ES は、アジアパシフィック (ソウル) ap-northeast-2 リージョンのサポートを追加しました。	2016 年 1 月 28 日
Amazon ES	初回リリース。	2015 年 10 月 1 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。