aws

開発者ガイド

AWS Key Management Service

AWS Key Management Service: 開発者ガイド

Table of Contents

AWS Key Management Service

AWS Key Management Service (AWS KMS) は、データの保護に使用される暗号化キーの作成と制御を容易にするマネージドサービスです。AWS KMS はハードウェアセキュリティモジュール (HSM) を使用して AWS KMS keys を保護し、FIPS 140-2 暗号化モジュール検証プログラムで検証します。中国 (北京) および中国 (寧夏) リージョンでは、FIPS 140-2 暗号化モジュール検証プログラムはサポートされていません。AWS KMS は OSCCA 認定 HSM を使用して、中国リージョンの KMS キーを保護します。

AWS KMS はデータを暗号化するほとんどの他の AWS サービスと統合されています。AWS KMS は AWS CloudTrail とも統合されており、監査、規制、コンプライアンスのニーズに応じて KMS キーの使用を口グに記録します。

AWS KMS APIを使用して KMS キーやカスタムキーストアなどの特別な機能の作成および管理ができ、KMS キーは暗号化オペレーションで使用できます。詳細については、「AWS Key Management Service API リファレンス」を参照してください。

AWS KMS keys を作成および管理できます。

- HMAC キーを含めた、対称および非対称 KMS キーを作成、編集、および表示します。
- キーポリシー、IAM ポリシー、グラントを使用して KMS キーへのアクセスを制御します。AWS KMS は属性ベースのアクセス制御 (ABAC) をサポートします。条件キーを使用してポリシーを調整することもできます。
- KMS キーのフレンドリ名であるエイリアスを作成、削除、一覧表示、更新します。また、エイリアスを使用して KMS キーへのアクセスを制御することもできます。
- 識別、オートメーション、コスト追跡のために KMS キーにタグ付けします。タグを使用して、KMS キーへのアクセスを制御することもできます。
- KMS キーを有効および無効にします。
- KMS キーの暗号化マテリアルの自動ローテーションを有効および無効にします。
- KMS キーを削除して、キーのライフサイクルを完了します。

KMS キーは暗号化オペレーションで使用できます。例については、「AWS KMS API のプログラミング」を参照してください。

- 対称または非対称 KMS キーを使用して、データを暗号化、復号、再暗号化します。
- 非対称 KMS キーを使用して、メッセージの署名と検証を行います。

- エクスポート可能な対称データキーと非対称データキーペアを生成します。
- HMAC コードを生成して検証します。
- 暗号化アプリケーションに適したランダムな数値を生成します。

AWS KMS のアドバンスト機能を使用できます。

- 異なる AWS リージョン の同じ KMS キーのコピーのように機能するマルチリージョンキーを作成します。
- KMS キーに暗号化マテリアルをインポートします。
- AWS CloudHSM クラスターによってバックアップされている AWS CloudHSM キーストアで KMS キーを作成します。
- AWS の外部の暗号化キーでバックアップされている外部キーストアで KMS キーを作成します。
- VPC のプライベートエンドポイントを介して AWS KMS に直接接続します。
- ハイブリッドポスト量子 TLS を使用して、AWS KMS へ送信するデータの転送時に前方暗号化を提供します。

AWS KMS を使用することで、暗号化するデータへのアクセスをより詳細に制御できます。キー管理および暗号化機能を直接アプリケーションで使用するか、AWS KMS と統合されている AWS サービスを介して使用できます。AWS のアプリケーションに書き込むか、AWS のサービスを使用するかにかかわらず、AWS KMS は AWS KMS keys を使用できるユーザーに対する制御を維持し、暗号化されたデータにアクセスすることができます。

AWS CloudTrail と統合されている AWS KMS は、指定された Amazon S3 バケットにログファイルを配信するサービスです。を使用すると CloudTrail 、KMS キーの使用方法と時期、および使用者をモニタリングして調査できます。

AWS リージョン 内の AWS KMS

AWS KMS がサポートされる AWS リージョン は、AWS Key Management Service エンドポイントとクォータに一覧表示されます。AWS KMS がサポートする AWS リージョン で AWS KMS 機能がサポートされない場合は、その機能に関するトピックでリージョンごとの違いが説明されます。

AWS KMS の料金

他の AWS 製品と同様に、AWS KMS を使用するために必要な契約や最低購入量はありません。AWS KMS の料金の詳細については、「AWS Key Management Service の料金」を参照してください。

サービスレベルアグリーメント

AWS Key Management Service は、当社のサービス可用性ポリシーを定義するサービスレベルアグリーメントによってバックアップされます。

詳細はこちら

- AWS KMS で使用される用語と概念については、「AWS KMS の概念」を参照してください。
- AWS KMS API の詳細については、AWS Key Management Service API リファレンスを参照してください。さまざまなプログラミング言語の例については、「AWS KMS API のプログラミング」を参照してください。
- AWS CloudFormation テンプレートを使用してキーとエイリアスを作成および管理する方法については、AWS CloudFormation ユーザーガイドの AWS CloudFormation での AWS KMS リソースの作成 および AWS Key Management Service リソースタイプリファレンスを参照してください。
- AWS KMS が暗号化を使用し、KMS キーを保護する方法の詳細な技術情報については、「AWS Key Management Service 暗号化の詳細」を参照してください。暗号化の詳細なドキュメントでは、AWS KMS の中国 (北京) および中国 (寧夏) リージョンでの動作方法は説明されていません。
- FIPS エンドポイントを含むそれぞれの AWS リージョン の AWS KMS エンドポイントのリストについては、「AWS 全般のリファレンス」の AWS Key Management Service トピックの「サービスエンドポイント」を参照してください。
- AWS KMS に関する質問のヘルプについては、AWS Key Management Service ディスカッションフォーラムを参照してください。

AWS SDK の AWS KMS

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

AWS KMS の概念

AWS Key Management Service (AWS KMS) で使用される基本的な用語および概念についてと、それらがデータの保護にどのように役立つかについて説明します。

トピック

AWS KMS keys

AWS KMS keys (KMS キー) は、AWS KMS のプライマリリソースです。KMS キーを使用して、データの暗号化、復号、再暗号化を行うことができます。また、AWS KMS の外部で使用できるデー

タキーを生成することもできます。通常は、対称 KMS キーを使用しますが、暗号化や署名には非対称 KMS キーを作成して使用することができ、HMAC KMS キーを作成および使用して、HMAC タグを生成および検証します。

> **Note**
>
> AWS KMS では、カスタマーマスターキー (CMK) という用語が AWS KMS key および KMS キーに置き換えられています。この概念に変更はありません。AWS KMS では、互換性を破る変更を避けるため、この用語にいくつかのバリエーションがあります。

AWS KMS key は、暗号化キーの論理表現です。KMS キーには、キー ID、キー仕様、キー使用法、作成日、説明、およびキーステータスなどのメタデータが含まれます。最も重要なのは、KMS キーを使用して暗号化オペレーションを実行するときに使用されるキーマテリアルへの参照が含まれていることです。

AWS KMS FIPS 検証済みのハードウェアセキュリティモジュールで生成された暗号化キーマテリアルを持つ KMS キーを作成できます。対称 KMS キーのキーマテリアルと非対称 KMS キーのプライベートキーは、AWS KMS を暗号化されないままにしません。KMS キーを使用または管理するには、AWS KMS を使用します。KMS キーの作成と管理に関する詳細については、 キーの管理 を参照してください。KMS キーを使用する方法の詳細については、AWS Key Management Service API リファレンスを参照してください。

デフォルトでは、AWS KMS は KMS キーのキーマテリアルを作成します。このキーマテリアルを抽出、エクスポート、表示、管理することはできません。唯一の例外は、非対称キーペアのパブリックキーで、エクスポートすることで AWS の外部で使用できます。また、このキーマテリアルは削除できません。KMS キーを削除する必要があります。ただし、独自のキーマテリアルを KMS キーにインポートしたり、カスタムキーストアを使用して AWS CloudHSM クラスター内のキーマテリアルを使用する KMS キーを作成したり、AWS の外部で所有および管理している外部キーマネージャーのキーマテリアルを使用したりすることができます。

AWS KMS は 1 つの AWS リージョン でデータを暗号化し、別の AWS リージョン で復号できるマルチリージョンキーもサポートします。

KMS キーの作成と管理に関する詳細については、 キーの管理 を参照してください。KMS キーを使用する方法の詳細については、「AWS Key Management Service API リファレンス」を参照してください。

カスタマーキーと AWS キー

お客様が作成する KMS キーはカスタマーマネージドキーです。サービスリソースの暗号化に KMS キーを使用する AWS のサービス では、多くの場合お客様が使うキーを作成します。AWS のサービス がお客様の AWS アカウントで作成する KMS キー は、AWS マネージドキー です。AWS のサービス がお客様のサービスアカウントで作成する KMS キー は、AWS 所有のキー です。

KMS キーのタイプ	KMS キーメタデータを表示可能	KMS キーを管理可能	AWS アカウント のみに使用	自動ローテーション	料金表
カスタマーマネージドキー	はい	はい	はい	オプション。毎年 (約 365 日)	月額料金 (時間単位の日割り計算) 従量課金料金
AWS マネージドキー	はい	いいえ	はい	必須。毎年 (約 365 日)	月額料金なし 従量課金料金 (一部の AWS のサービス では、この料金をお支払いください)
AWS 所有のキー	いいえ	いいえ	いいえ	可変	料金はかかりません

AWS KMS と統合されている AWS のサービスでは、KMS キーのサポートが異なります。一部の AWS のサービスでは、デフォルトで AWS 所有のキー または AWS マネージドキー を使用してデータが暗号化されます。一部の AWS のサービスでは、カスタマーマネージドキーをサポートするものもあります。また、他の AWS のサービスでは、AWS 所有のキー の使いやすさ、AWS マネージドキー の可視性、カスタマーマネージドキーの制御性を考慮し、すべてのタイプの KMS キーをサポートします。AWS のサービスが提供する暗号化オプションの詳細については、ユーザーガイドの「保管時の暗号化」トピックまたはサービスのデベロッパーガイドを参照してください。

カスタマーマネージドキー

お客様が作成する KMS キーは、カスタマーマネージドキーです。カスタマーマネージドキーは AWS アカウント内の KMS キーで、ユーザーが作成、所有、および管理します。これらの KMS キーは、キーポリシー、IAM ポリシー、グラントの確立と管理、有効化と無効化、暗号化マテリアルのローテーション、タグの追加、KMS キーを参照するエイリアスの作成 、KMS キー削除のスケジューリングなどを完全に制御できます。

カスタマーマネージドキーは、AWS KMS の AWS Management Console のカスタマーマネージドキーページに表示されます。カスタマーマネージドキーを明確に識別するには、DescribeKey オペレーションを使用します。カスタマーマネージドキーでは、`DescribeKey` レスポンスの `KeyManager` フィールドの値は `CUSTOMER` です。

暗号化オペレーションでカスタマーマネージドキーを使用し、AWS CloudTrail ログでその使用を監査できます。さらに、AWS KMS と統合されている多数の AWS のサービスを使用すると、カスタマーマネージドキーを指定して保存および管理するデータを保護できます。

カスタマーマネージドキーの使用には、月額料金と、無料利用枠を超えた使用に対する料金がかかります。これらは、お客様のアカウントの AWS KMS クォータに対して影響があります。詳細については、「AWS Key Management Service 料金表」および「クォータ」を参照してください。

AWS マネージドキー

AWS マネージドキー は、AWS KMS と統合されている AWS のサービスがユーザーに代わって作成、管理、使用する、アカウントの KMS キーです。

一部の AWS のサービスでは、AWS マネージドキー かまたは、そのサービスのリソースを保護するためのカスタマーマネージドキーを選択できます。一般に、リソースを保護する暗号化キーを制御する必要がない限り、AWS マネージドキー は良い選択です。キーまたはそのキーポリシーを作成または維持する必要はなく、AWS マネージドキー の月額料金はまったく発生しません。

アカウントで AWS マネージドキー を表示し、それらのキーポリシーを表示し、AWS CloudTrail ログでそれらの使用を監査するアクセス許可が与えられます。ただし、AWS マネージドキー のプロパティの変更、ローテーション、キーポリシーの変更、削除のスケジュールを設定を行うことはできません。また、AWS マネージドキー を暗号化オペレーションで直接使用することはできません。それらを作成したサービスがユーザーに代わってそれらを使用します。

AWS マネージドキー は AWS KMS の AWS Management Console の AWS マネージドキー ページに表示されます。AWS マネージドキー は `aws/redshift` のような `aws/`*`service-name`* 形式のエイリアスも識別できます。を明確に識別するにはAWS マネージドキー、 DescribeKeyオペレーショ

ンを使用します。AWS マネージドキー の場合、DescribeKey レスポンスの KeyManager フィールドの値は AWS です。

すべての AWS マネージドキーは、毎年自動的にローテーションされます。このローテーションスケジュールは変更できません。

> **Note**
>
> 2022 年 5 月、AWS KMS は AWS マネージドキー のローテーションスケジュールを 3 年 (約 1,095 日間隔) ごとから毎年 (約 365 日間隔) に変更しました。
> 新しい AWS マネージドキーは、作成日から 1 年後に自動的にローテーションされ、それ以降はほぼ 1 年ごとにローテーションされます。
> 既存の AWS マネージドキー は、直近のローテーションから 1 年後にローテーションされ、その後毎年ローテーションされます。

AWS マネージドキー には月額料金はかかりません。これらは、無料利用枠を超える量を使用した場合は有料になりますが、一部の AWS サービスでこれらのコストがカバーされます。詳細については、サービスのユーザーガイドまたはデベロッパーガイドの「保管時の暗号化 」トピックを参照してください。詳細については、「AWS Key Management Service の料金表」を参照してください。

AWS マネージドキー は、アカウントの各リージョンの KMS キー数のリソースクォータに対してカウントされません。ただし、アカウントのプリンシパルに代わって使用される場合、KMS キーはリクエストクォータに対してカウントされます。詳細については、「クォータ」を参照してください。

AWS 所有のキー

AWS 所有のキー は、AWS のサービスが複数の AWS アカウント で使用するために所有および管理する KMS キーのコレクションです。AWS 所有のキー は AWS アカウント 内にはありませんが、AWS のサービスは AWS 所有のキー を使用してアカウント内のリソースを保護できます。

一部の AWS サービスでは、AWS 所有のキー またはカスタマーマネージドキーを選択できます。一般に、リソースを保護する暗号化キーを監査または制御する必要がない限り、AWS 所有のキー は良い選択です。AWS 所有のキー は完全に無料 (月額料金や使用料なし) であり、アカウントのAWS KMS クォータに対してカウントされませんし、使いやすいものです。キーまたはそのキーポリシーを作成または管理する必要はありません。

AWS 所有のキー のローテーションは、サービスによって異なります。特定の AWS 所有のキー のローテーションの詳細については、サービスのユーザーガイドまたはデベロッパーガイドの「保管時の暗号化」トピックを参照してください。

対称暗号化 KMS キー

AWS KMS key を作成するときは、デフォルトで対称暗号化用の KMS キーが作成されます。これは、最もよく使用されるタイプの基本的な KMS キーです。

AWS KMS では、対称暗号化 KMS キーは 256 ビットの AES-GCM 暗号化キーを表します。ただし、中国リージョンでは 128 ビット SM4 暗号化キーを表します。対称キーマテリアルが、暗号化されずに AWS KMS 外で使用されることは一切ありません。対称暗号化 KMS キーを使用するには、AWS KMS を呼び出す必要があります。対称暗号化キーは対称暗号化で使用され、同じキーが暗号化と復号に使用されます。タスクが非対称キーを明示的に要求する場合以外は、暗号化されずに AWS KMS 外で使用されることがない対称暗号化 KMS キーが適切な選択肢になります。

AWS KMS と統合された AWS のサービスは、データの暗号化に対称暗号化 KMS キーのみを使用します。これらのサービスは、非対称 KMS キーを使用する暗号化をサポートしません。KMS キーが対称か非対称かを判断する方法については、「非対称 KMS キーの識別」を参照してください。

具体的には、対称キーのキー仕様は SYMMETRIC_DEFAULT、キー使用法は ENCRYPT_DECRYPT、暗号化アルゴリズムは SYMMETRIC_DEFAULT です。詳細については、「SYMMETRIC_DEFAULT キー仕様」を参照してください。

AWS KMS で対称暗号化 KMS キーを使用して、データの暗号化、復号、再暗号化、およびデータキーとデータキーペアの生成を行うことができます。マルチリージョンの対称暗号化 KMS キーの作成、対称暗号化 KMS キーへの独自のキーマテリアルのインポート、およびカスタムキーストアでの対称暗号化 KMS キーの作成が可能です。異なるタイプの KMS キーで実行できるオペレーションを比較した表については、「キータイプリファレンス」を参照してください。

非対称 KMS キー

AWS KMS で非対称 KMS キーを作成できます。非対称 KMS キーは、数学的に関連するパブリックキーとプライベートキーペアを表します。プライベートキーが暗号化されないまま AWS KMS から出ていくことはありません。プライベートキーを使用するには、AWS KMS を呼び出す必要があります。AWS KMS API オペレーションを呼び出すことによって、AWS KMS 内でパブリックキーを使用することも、パブリックキーをダウンロードして AWS KMS の外部で使用することもできます。マルチリージョン非対称 KMS キーを作成することもできます。

パブリックキーの暗号化または署名および検証用の RSA キーペアまたは SM2 キーペア (中国リージョンのみ)、または署名と検証用の楕円曲線キーペアを表す非対称 KMS キーを作成できます。

非対称 KMS キーの作成と使用の詳細については、「AWS KMS の非対称キー」を参照してください。

HMAC KMS キー

HMAC KMS キーは、Hash-based Message Authentication Code (HMAC) の生成と検証に使用されるさまざまな長さの対称キーを表します。HMAC キーのキーマテリアルが、暗号化されずに AWS KMS 外で使用されることはありません。HMAC キーを使用するには、`GenerateMac` または `VerifyMac` API オペレーションを呼び出します。

マルチリージョンの HMAC KMS キーを作成することもできます。

HMAC KMS キーの作成と使用の詳細については、「AWS KMS での HMAC キー」を参照してください。

データキー

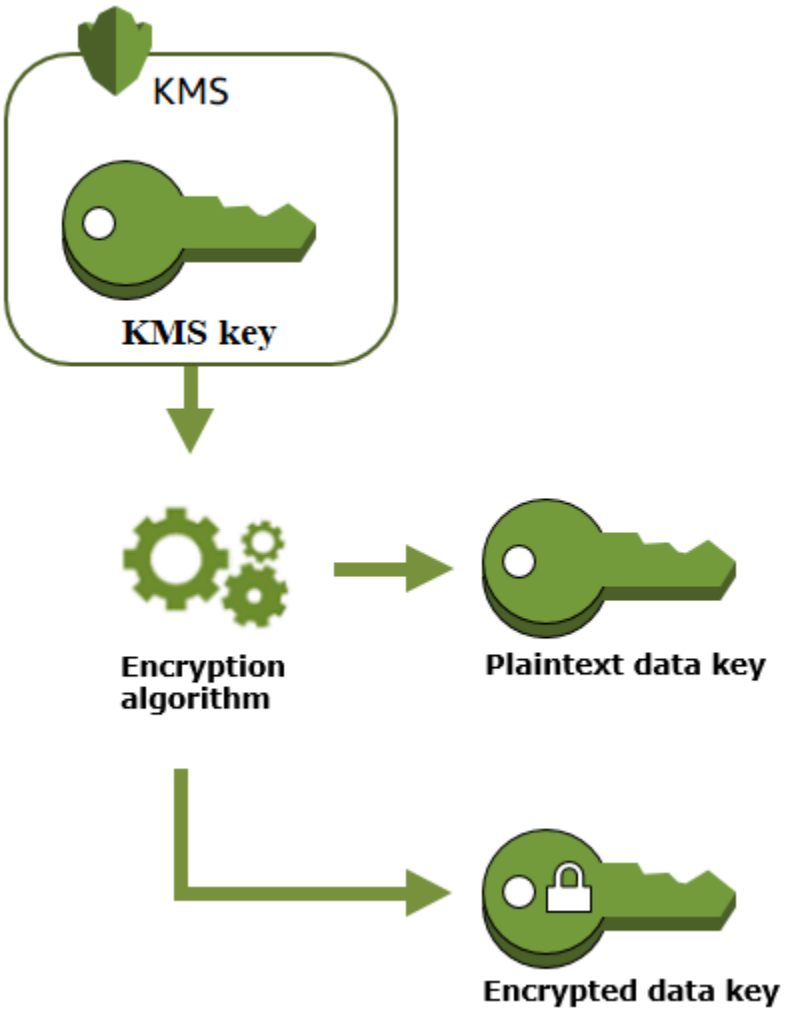
データキーは、大量のデータ、および他のデータ暗号化キーなどのデータを暗号化するために使用できる暗号化キーです。ダウンロードできない対称 KMS キーとは異なり、データキーは AWS KMS 外での使用のために返されます。

AWS KMS はデータキーの生成時、即座に使用 (オプション) できるプレーンテキストのデータキーと、データと共に安全に保存できるデータキーの暗号化されたコピーを返します。データを復号する準備ができたら、最初に AWS KMS を要求して、暗号化されたデータキーを復号します。

AWS KMS はデータキーを生成、暗号化、復号します。ただし、AWS KMS はデータキーの保存、管理、追跡、またはデータキーの暗号化オペレーションを実行しません。AWS KMS の外部でデータキーを使用して管理する必要があります。データキーを安全に使用する方法については、「AWS Encryption SDK」を参照してください。

データキーの作成

データキーを作成するには、 GenerateDataKeyオペレーションを呼び出します。 はデータキーAWS KMSを生成します。次に、ユーザーが指定する対称暗号化 KMS キーのデータキーのコピーを暗号化します。このオペレーションでは、データキーのプレーンテキストコピーと KMS キーで暗号化されたデータキーのコピーが返されます。以下の図では、このオペレーションを示しています。

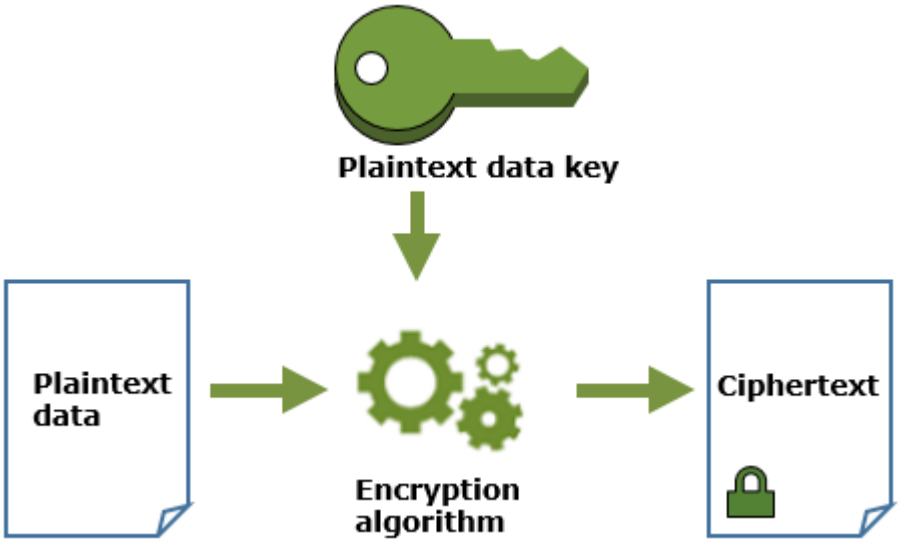


AWS KMS は、暗号化されたデータキーのみを返す GenerateDataKeyWithoutPlaintextオペレーションもサポートしています。データキーを使用する必要がある場合、AWS KMS にそのデータキーを復号するように求めます。

データキーでデータを暗号化する

AWS KMS は、データキーを使用してデータを暗号化することはできません。ただし、OpenSSL を使用するか、AWS Encryption SDK のような暗号化ライブラリを使用することで、AWS KMS の外部でデータキーを使用できます。

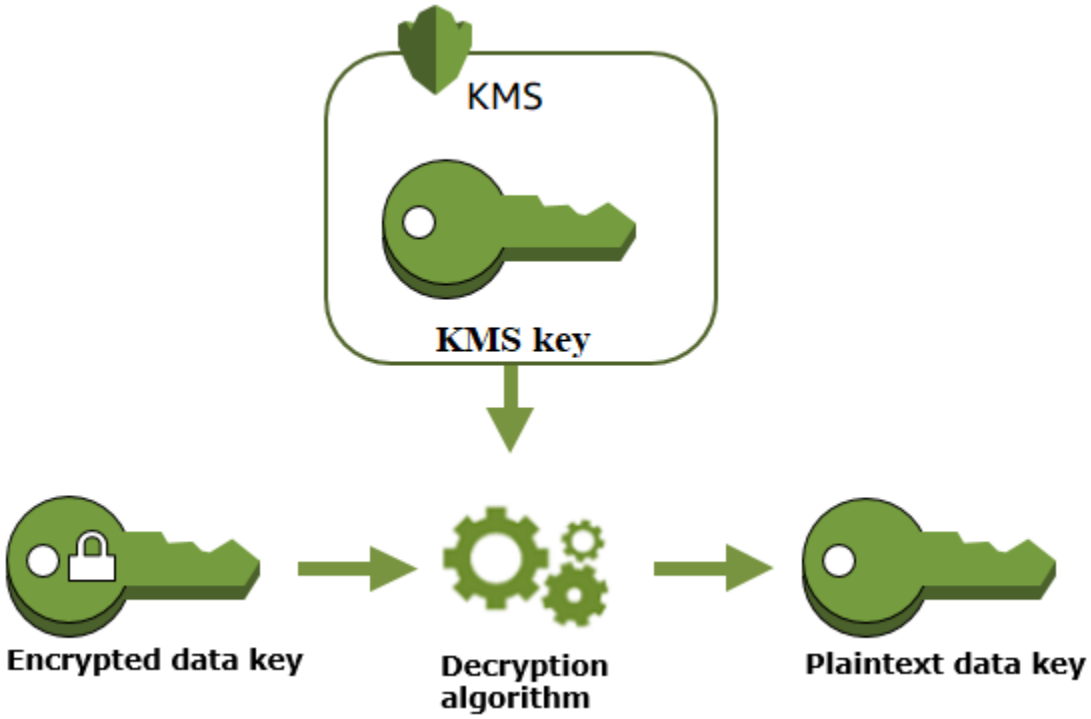
プレーンテキストのデータキーを使用してデータを暗号化したら、できる限り早急にメモリからそれを削除します。暗号化したデータと一緒に暗号化データキーを安全に保存して、データの復号に利用することができます。



データキーでデータを復号する

データを復号するには、Decrypt オペレーションに暗号化されたデータキーを渡します。AWS KMS は KMS キーを使用してデータキーを復号し、プレーンテキストのデータキーを返します。プレーンテキストのデータキーを使ってデータを復号し、続いてできる限り早急にメモリからプレーンテキストのデータキーを削除します。

以下の図では、Decrypt オペレーションを使用して暗号化されたデータキーを復号する方法を示しています。



使用できない KMS キーがデータキーに及ぼす影響

KMS キーが使用できなくなると、その影響はほぼ即時に表れます (最終的な一貫性の対象となります)。KMS キーのキーステータスは新しい条件を反映して変化し、暗号化オペレーションで KMS キーを使用するすべてのリクエストは失敗します。

ただし、KMS キーで暗号化されたデータキー、およびデータキーで暗号化されたデータへの影響は、データキーの復号などに KMS キーが再度使用されるまで遅延します。

KMS キーは、実行する可能性のある次のアクションなどのような、さまざまな原因によって使用できなくなる可能性があります。

- KMS キーを無効にする
- KMS キーの削除をスケジュールする
- キーマテリアルがインポートされた KMS キーからキーマテリアルを削除するか、インポートされたキーマテリアルの有効期限が切れるようにします。
- KMS キーをホストする AWS CloudHSM キーストアを切断するか、KMS キーのキーマテリアルとして機能する AWS CloudHSM クラスターからキーを削除します。
- KMS キーをホストする外部キーストアの切断、または外部キーストアプロキシへの暗号化および復号リクエストを妨げるその他のアクション (外部キーマネージャーからの外部キーの削除など) です。

この効果は、データキーを使用して、サービスが管理するリソースを保護している多くの AWS のサービス ユーザーにとって特に重要です。次の例では、Amazon Elastic Block Store (Amazon EBS) と Amazon Elastic Compute Cloud (Amazon EC2) を使用しています。異なる AWS のサービス は様々な方法でデータキーを使用します。詳細については、AWS のサービス の「セキュリティ」の章の「データ保護」セクションを参照してください。

例えば、次のシナリオが考えられます。

1. 暗号化された EBS ボリュームを作成し、KMS キーを指定して保護します。Amazon EBS は、KMS キーを使用してボリュームの暗号化されたデータキーを生成するよう、AWS KMS に要求します。Amazon EBS は、暗号化されたデータキーをボリュームのメタデータとともに保存します。
2. EBS ボリュームを EC2 インスタンスにアタッチすると、Amazon EC2 は、KMS キーを使用して EBS ボリュームの暗号化されたデータキーを復号します。Amazon EC2 は、EBS ボリュームに対するすべてのディスク I/O を暗号化する責任を担う Nitro ハードウェア内のデータキーを使用します。データキーは、EBS ボリュームが EC2 インスタンスにアタッチされる限り、Nitro ハードウェア内で維持されます。
3. KMS キーを使用不可能にするアクションを実行しました。これによって EC2 インスタンスまたは EBS ボリュームにただちに影響が出ることはありません。Amazon EC2 は、KMS キーではなくデータキーを使用して、ボリュームがインスタンスにアタッチされている限り、すべてのディスク I/O を暗号化します。
4. ただし、暗号化された EBS ボリュームが EC2 インスタンスからデタッチされると、Amazon EBS は Nitro ハードウェアからデータキーを削除します。次回、暗号化された EBS ボリュームが EC2 インスタンスにアタッチされると、アタッチメントは失敗します。これは、Amazon EBS は KMS キーを使用してボリュームの暗号化されたデータキーを復号できないためです。EBS ボリュームを再度使用するには、KMS キーを再度使用可能にする必要があります。

データキーペア

データキーペアは、数学的に関連するパブリックキーとプライベートキーで構成される非対称データキーです。これらは、クライアント側の暗号化と復号、または AWS KMS の外部での署名と検証に使用されるように設計されています。

OpenSSL などのツールが生成するデータキーペアとは異なり、AWS KMS は、ユーザーが指定する AWS KMS 内の対称暗号化 KMS キーの各データキーペアにあるプライベートキーを保護します。ただし、AWS KMS はデータキーペアの保存、管理、追跡、またはデータキーペアの暗号化オペレーションを実行しません。AWS KMS の外部でデータキーペアを使用して管理する必要があります。

AWS KMS は、次のタイプのデータキーペアをサポートしています。

- RSA キーペア: RSA_2048、RSA_3072、RSA_4096
- 楕円曲線キーペア、ECC_NIST_P256、ECC_NIST_P384、ECC_NIST_P521、ECC_SECG_P256K1
- SM キーペア (中国リージョンのみ): SM2

選択するデータキーペアのタイプは、通常、ユースケースまたは規制要件によって異なります。ほとんどの証明書には RSA キーが必要です。楕円曲線キーは、デジタル署名によく使用されます。ECC_SECG_P256K1 キーは、一般的に暗号化通貨に使用されます。AWS KMS では、署名に ECC キーペアを使用し、暗号化または署名に RSA キーペアを使用することが推奨されていますが、両方には使用しないでください。ただし、AWS KMS の外部でデータキーペアを使用することを AWS KMS で強制的に制限することはできません。

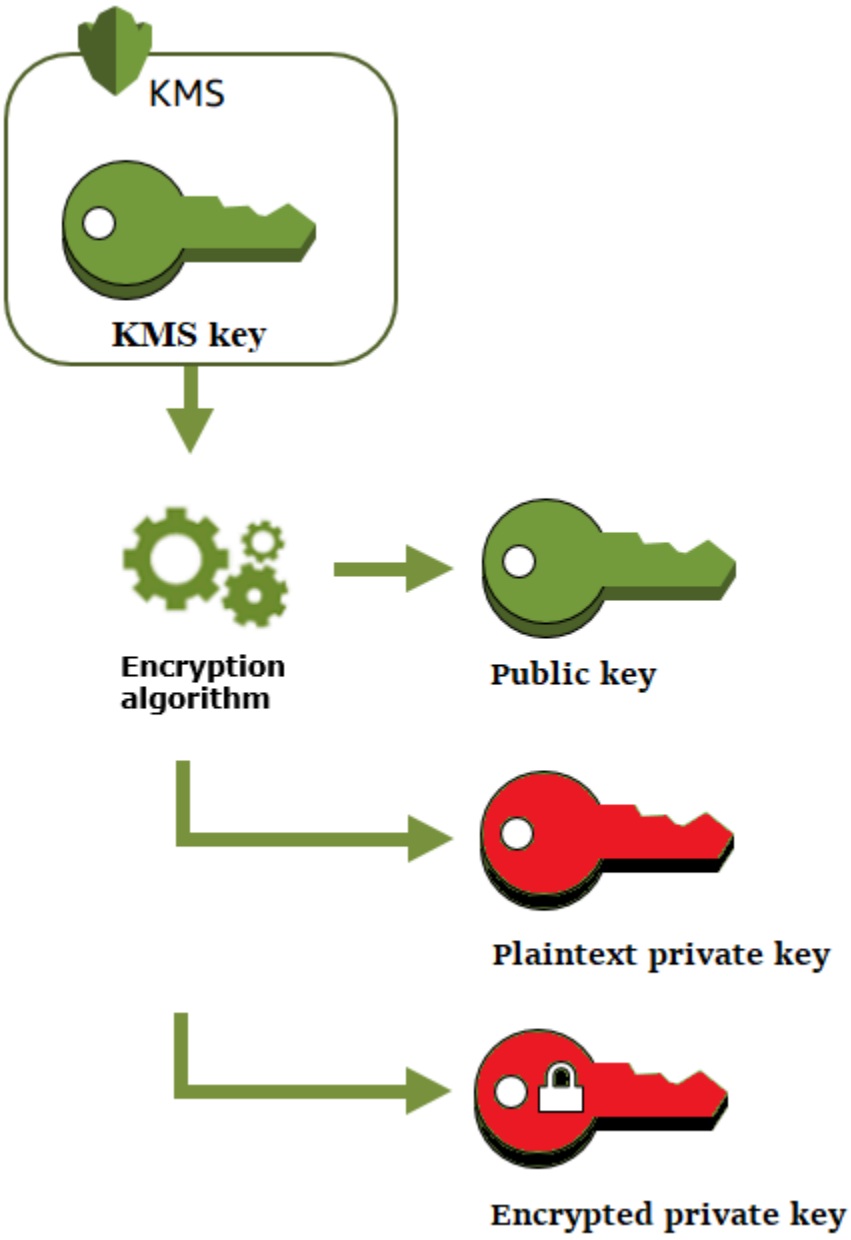
データキーペアを作成する

データキーペアを作成するには、 GenerateDataKeyPairまたは GenerateDataKeyPairWithoutPlaintextオペレーションを呼び出します。プライベートキーの暗号化に使用する対称暗号化 KMS キーを指定します。

`GenerateDataKeyPair` は、プレーンテキストパブリックキー、プレーンテキストプライベートキー、暗号化されたプライベートキーを返します。このオペレーションは、デジタル署名を生成する場合など、プレーンテキストプライベートキーをすぐに必要とする場合に使用します。

`GenerateDataKeyPairWithoutPlaintext` は、プレーンテキストのパブリックキーと暗号化されたプライベートキーを返しますが、プレーンテキストプライベートキーは返しません。このオペレーションは、パブリックキーで暗号化する場合など、プレーンテキストプライベートキーがすぐに必要ない場合に使用します。後で、データを復号化するために平文の秘密鍵が必要な場合は、Decrypt オペレーションを呼び出すことができます。

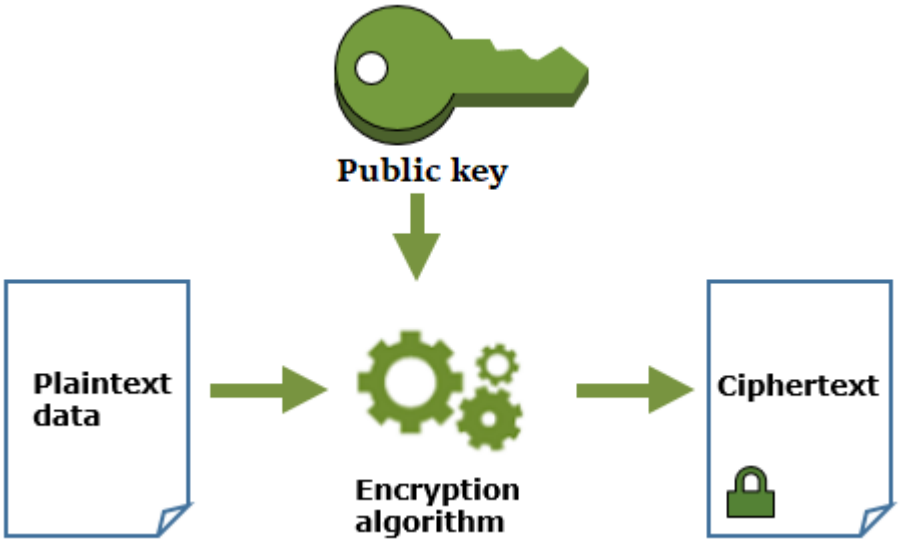
以下の図では、`GenerateDataKeyPair` オペレーションを示しています。`GenerateDataKeyPairWithoutPlaintext` オペレーションは、プレーンテキストプライベートキーを省略します。



データキーペアでデータを暗号化する

データキーペアを使用して暗号化する場合、ペアのパブリックキーを使用してデータを暗号化し、同じペアのプライベートキーを使用してデータを復号します。通常、データキーペアは、プライベートキーを保持している当事者のみが復号できるデータを、多くの当事者が暗号化する必要がある場合に使用されます。

次の図に示すように、パブリックキーを持つ当事者は、そのキーを使用してデータを暗号化します。

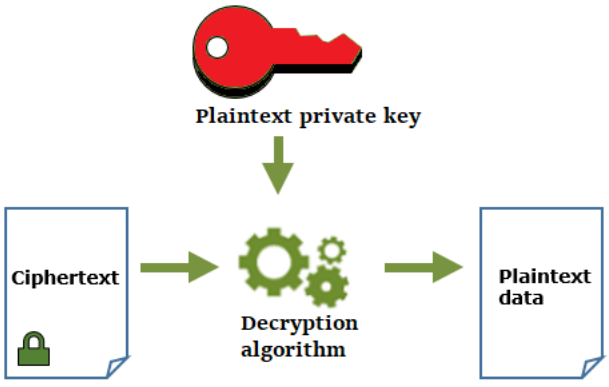


データキーペアでデータを復号する

データを復号するには、データキーペアでプライベートキーを使用します。オペレーションを成功させるには、パブリックキーとプライベートキーが同じデータキーペアのものである必要があり、また、同じ暗号化アルゴリズムを使用する必要があります。

暗号化された秘密鍵を復号化するには、復号オペレーションに渡します。プレーンテキストプライベートキーを使用してデータを復号します。その後、できるだけ早くプレーンテキストのプライベートキーをメモリから削除します。

次の図は、データキーペアのプライベートキーを使用して暗号テキストを復号する方法を示しています。



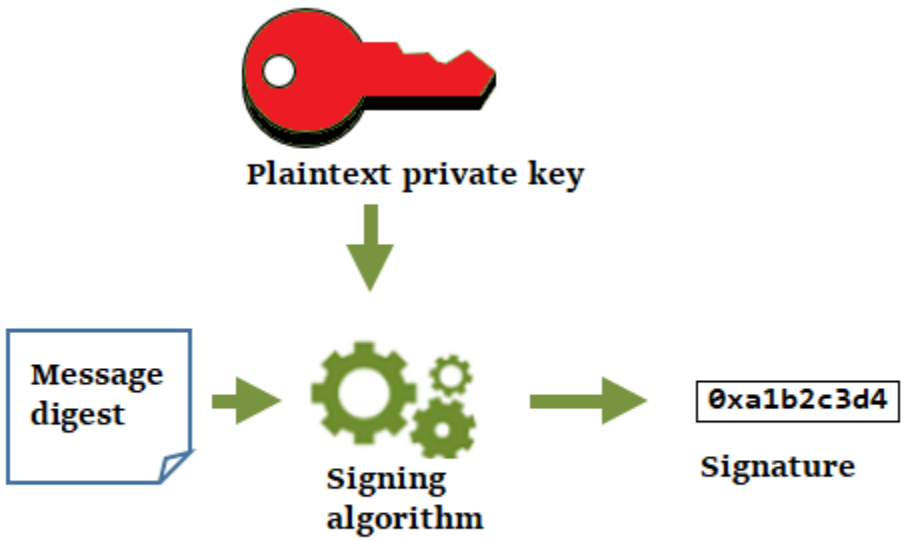
データキーペアでメッセージに署名する

メッセージの暗号化署名を生成するには、データキーペアのプライベートキーを使用します。パブリックキーを持つすべてのユーザーは、メッセージが自分のプライベートキーで署名されたこと、および署名されてから変更されていないことを確認するために使用できます。

プライベートキーが暗号化されている場合は、暗号化されたプライベートキーを Decrypt オペレーションに渡します。AWS KMS は KMS キーを使用してデータキーを復号し、プレーンテキストのプライベートキーを返します。署名を生成するには、プレーンテキストプライベートキーを使用します。その後、できるだけ早くプレーンテキストのプライベートキーをメモリから削除します。

メッセージに署名するには、OpenSSL で `dgst` コマンドなどの暗号化ハッシュ関数を使用してメッセージダイジェストを作成します。次に、プレーンテキストプライベートキーを署名アルゴリズムに渡します。結果は、メッセージの内容を表す署名です。(最初にダイジェストを作成しなくても、短いメッセージで署名できる場合があります。メッセージの最大サイズは、使用する署名ツールによって異なります)。

次の図は、データキーペアのプライベートキーを使用してメッセージに署名する方法を示しています。



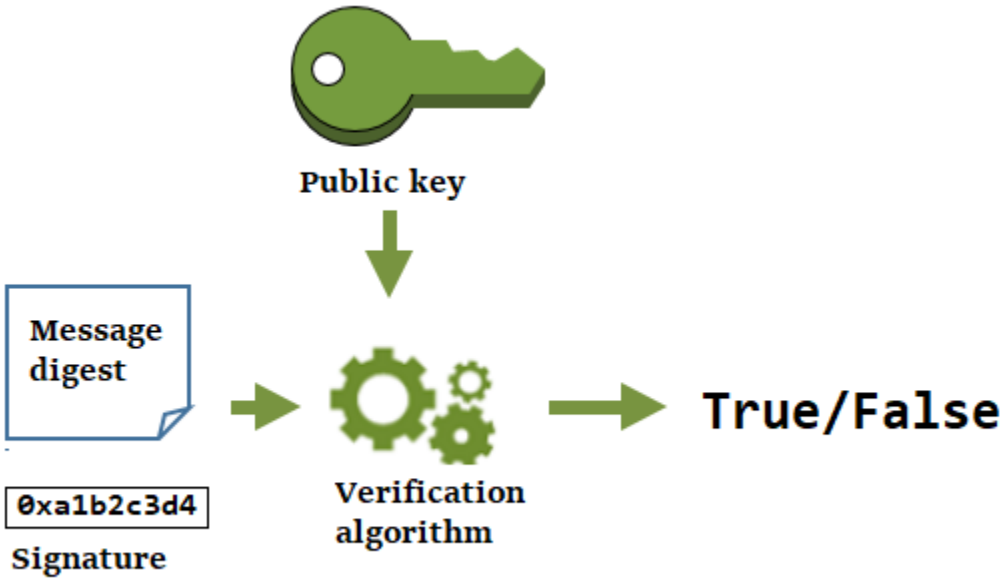
データキーペアを使用した署名の検証

データキーペアにパブリックキーを持っているユーザーは、誰でもそれを使用して、プライベートキーで生成した署名を検証できます。検証では、承認されたユーザーが指定したプライベートキーと

署名アルゴリズムを使用してメッセージに署名し、メッセージが署名後に変更されていないことを確認します。

署名を検証する当事者は、同じタイプのダイジェストを生成し、同じアルゴリズムを使用して、メッセージの署名に使用されるプライベートキーに対応するパブリックキーを使用する必要があります。

次の図は、データキーペアのパブリックキーを使用してメッセージ署名を検証する方法を示しています。



エイリアス

KMS キーのフレンドリ名としてエイリアスを使用します。例えば、KMS キーを 1234abcd-12ab-34cd-56ef-1234567890ab ではなく、テストキーとして参照できます。

エイリアスを使用すると、AWS Management Console でKMS キーを識別しやすくなります。エイリアスを使用して、暗号化オペレーションなどの一部の AWS KMS オペレーションで KMS キーを識別することもできます。アプリケーションでは、1 つのエイリアスを使用して各 AWS リージョン の異なる KMS キー を参照できます。

ポリシーを編集したり、グラントを管理したりすることなく、エイリアスに基づいて KMS キーへのアクセスを許可および拒否することもできます。この機能は、属性ベースのアクセスコントロール (ABAC) の AWS KMS サポートの一部です。詳細については、「AWS KMS の ABAC」を参照してください。

AWS KMS では、エイリアスは KMS キーのプロパティではなく、独立したリソースです。そのため、関連付けられた KMS キーに影響を与えずに、エイリアスを追加、変更、削除できます。

> **Important**
>
> エイリアス名には、機密情報や重要情報を含めないでください。エイリアスは、 CloudTrail ログやその他の出力にプレーンテキストで表示されます。

詳細はこちら:

- エイリアスの詳細については、「エイリアスの使用」を参照してください。
- エイリアスを含むキー識別子の形式については、「キー識別子 (KeyId) 」を参照してください。
- KMS キーに関連付けられているエイリアスの検索については、エイリアス名とエイリアス ARN を見つける を参照してください。
- 複数のプログラミング言語でのエイリアスの作成と管理の例については、「エイリアスの使用」を参照してください。

カスタムキーストア

カスタムキーストアは、ユーザーが所有および管理している AWS KMS の外部のキーマネージャーによってバックアップされている AWS KMS リソースです。暗号化オペレーションでカスタムキーストアの KMS キーを使用すると、暗号化オペレーションは、実際には暗号化キーを使用してキーマネージャーで実行されます。

AWS KMS は、AWS CloudHSM クラスターによってバックアップされている AWS CloudHSM キーストアと、AWS の外部にある外部キーマネージャによってバックアップされている外部キーストアをサポートしています。

詳細については、「カスタムキーストア」を参照してください。

暗号化オペレーション

AWS KMS の場合、暗号化オペレーションとは、KMS キーを使用してデータを保護する API オペレーションです。KMS キーは AWS KMS 内にあるため、暗号化オペレーションで KMS キーを使用するには AWS KMS を呼び出す必要があります。

KMS キーで暗号化オペレーションを実行するには、AWS SDK、AWS CLI (AWS Command Line Interface)、AWS Tools for PowerShell を使用します。AWS KMS コンソールで暗号化オペレーショ

ンを実行することはできません。いくつかのプログラミング言語で暗号化オペレーションを呼び出す例については、を参照してください AWS KMS API のプログラミング。

以下の表では、AWS KMS 暗号化オペレーションを示しています。また、オペレーションで使用される KMS キーのキータイプとキー使用法の要件も示します。

オペレーション	キーのタイプ	キーの用途
Decrypt	対称または非対称	ENCRYPT_DECRYPT
暗号化	対称または非対称	ENCRYPT_DECRYPT
GenerateDataKey	対称	ENCRYPT_DECRYPT
GenerateDataKeyPair	非対称 [1] カスタムキーストアの KMS キーではサポートされません。	ENCRYPT_DECRYPT
GenerateDataKeyPairWithoutPlaintext	非対称 [1] カスタムキーストアの KMS キーではサポートされません。	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	対称	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_VERIFY_MAC
GenerateRandom	該当なし。このオペレーションでは KMS キーを使用しません。	該当なし
ReEncrypt	対称または非対称	ENCRYPT_DECRYPT
Sign	非対称	SIGN_VERIFY

オペレーション	キーのタイプ	キーの用途
検証	非対称	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_VERIFY_MAC

[1] 対称暗号化 KMS キーによって保護される非対称データキーペアを生成します。

暗号化オペレーションのアクセス許可については、「the section called “アクセス許可に関するリファレンス”」を参照してください。

すべてのユーザーに対する AWS KMS の応答性と機能性を向上させるために、AWS KMS では、1 秒あたりに呼び出すことができる暗号化オペレーションの数にクォータを設定します。詳細については、「the section called “暗号化オペレーションの共有クォータ”」を参照してください。

キー識別子 (KeyId)

キー識別子は、KMS キーの名前のように機能します。これらは、コンソールで KMS キーを識別するのに役立ちます。キー識別子を使用して、AWS KMS API オペレーション、キーポリシー、IAM ポリシー、グラントで使用する KMS キーを指定します。キー識別子の値は、KMS キーに関連付けられているキーマテリアルとはまったく関係ありません。

AWS KMS は、いくつかのキー識別子を定義します。KMS キーを作成すると、AWS KMS は KMS キーのプロパティであるキー ARN とキー ID を生成します。エイリアスを作成すると、AWS KMS は、ユーザー定義のエイリアス名に基づいてエイリアス ARN を生成します。キーおよびエイリアスの識別子は、AWS Management Console および AWS KMS API で表示できます。

AWS KMS コンソールでは、KMS キーをキー ARN、キー ID、エイリアス名で表示およびフィルタリングし、キー ID とエイリアス名でソートできます。コンソールでキー ID を検索する方法については、「the section called “キー ID とキー ARN を検索する”」を参照してください。

AWS KMS API では、KMS キーの識別に使用するパラメータ名は、`KeyId` またはバリエーション (`TargetKeyId` や `DestinationKeyId` など) になります。ただし、これらのパラメータの値はキー ID に限定されません。いくつかは、任意の有効なキー識別子を受け取ることができます。各パラメータ値の詳細については、AWS Key Management Service API リファレンスのパラメータの説明を参照してください。

> **Note**
>
> AWS KMS API を使用する際は、使用するキー識別子に注意してください。API ごとに異なるキー識別子が必要です。通常、タスクに対して実用的で最も完全なキー識別子を使用します。

AWS KMS では、次のキー識別子がサポートされています。

キー ARN

キー ARN は、KMS キーの Amazon リソースネーム (ARN) です。これは KMS キーの、一意の完全修飾識別子です。キー ARN には、AWS アカウント、リージョン、キー ID が含まれます。KMS キーのキー ARN を検索する方法については、the section called “キー ID とキー ARN を検索する” を参照してください。

キー ARN の形式は次のとおりです。

```
arn:<partition>:kms:<region>:<account-id>:key/<key-id>
```

以下は単一リージョン KMS キーのサンプルキー ARN です。

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

##########のキー ARN のkey-id 要素は、mrk- プレフィックスで始まります。以下はマルチリージョンキーのサンプルキー ARN です。

```
arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab
```

キー ID

キー ID は、アカウントとリージョン内の KMS キーを一意に識別します。KMS キーのキー ID を検索する方法については、the section called “キー ID とキー ARN を検索する” を参照してください。

以下は単一リージョン KMS キーのサンプルキー ID です。

```
1234abcd-12ab-34cd-56ef-1234567890ab
```

マルチリージョンキーのキー ID は、mrk- プレフィックスで始まります。以下はマルチリージョンキーのサンプルキー ID です。

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

エイリアス ARN

エイリアス ARN は、AWS KMS エイリアスの Amazon リソースネーム (ARN) です。これは、エイリアスとそれが表す KMS キーの、一意の完全修飾識別子です。エイリアス ARN には、AWS アカウント、リージョン、エイリアス名が含まれます。

エイリアス ARN は、任意の時点で特定の 1 つの KMS キーを識別します。ただし、エイリアスに関連付けられている KMS キーは変更できるため、エイリアス ARN は異なる時点でさまざまな KMS キーを識別できます。KMS キーのエイリアス ARN を検索する方法については、エイリアス名とエイリアス ARN を見つける を参照してください。

エイリアス ARN の形式は次のとおりです。

```
arn:<partition>:kms:<region>:<account-id>:alias/<alias-name>
```

以下は、架空の ExampleAlias のエイリアス ARN です。

```
arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias
```

エイリアス名

エイリアス名は、最大 256 文字の文字列です。エイリアス名は、アカウントとリージョン内で関連付けられた KMS キーを一意に識別します。AWS KMS API では、エイリアス名は常に alias/ で始まります。KMS キーのエイリアス名を検索する方法については、エイリアス名とエイリアス ARN を見つける を参照してください。

エイリアス名の形式は次のとおりです。

```
alias/<alias-name>
```

例:

```
alias/ExampleAlias
```

エイリアス名の aws/ プレフィックスは、AWS マネージドキー 用に予約されます。このプレフィックスでエイリアスを作成することはできません。例えば、Amazon Simple Storage Service (Amazon S3) の AWS マネージドキー でのエイリアス名は、次のとおりです。

```
alias/aws/s3
```

キーマテリアル

キーマテリアルは、暗号化アルゴリズムで使用されるビット単位の文字列です。キーマテリアルを使用する暗号化オペレーションを保護するには、シークレットキーを秘密にしておく必要があります。パブリックキーマテリアルは共有されるように設計されています。

各 KMS キーには、メタデータにキーマテリアルへの参照が含まれています。対称暗号化 KMS キーのキーマテリアルのオリジンは、異なる可能性があります。AWS KMS が生成するキーマテリアル、またはカスタムキーストアの AWS CloudHSM のクラスターで生成されるキーマテリアルを使用するか、独自のキーマテリアルをインポートすることができます。対称暗号化 KMS キーに AWS KMS キーマテリアルを使用する場合は、キーマテリアルの自動ローテーションを有効にできます。

デフォルトでは、各 KMS キーに一意のキーマテリアルがあります。ただし、同じキーマテリアルでマルチリージョンキーのセットを作成することができます。

キーマテリアルのオリジン

キーマテリアルのオリジンは、KMS キーのキーマテリアルのソースを識別する KMS キープロパティです。KMS キーの作成時にキーマテリアルのオリジンを選択し、それを変更することはできません。キーマテリアルのソースは、KMS キーのセキュリティ、耐久性、可用性、レイテンシー、スループット特性に影響します。

KMS キーのキーマテリアルのオリジンを見つけるには、 DescribeKeyオペレーションを使用するか、AWS KMSコンソールの KMS キーの詳細ページの暗号化設定タブでオリジン値を確認します。ヘルプについては、「キーの表示」を参照してください。

KMS キーでは、次のうち 1 つのキーマテリアルオリジン値を指定できます。

AWS_KMS

AWS KMS は、独自のキーストアで、KMS キーのキーマテリアルを作成および管理します。これは、ほとんどの KMS キーのデフォルト値であり、レコメンデーション値です。

AWS KMS からキーマテリアルを持つキーを作成する方法については、「キーの作成」を参照してください。

EXTERNAL (Import key material)

KMS キーには、インポートされたキーマテリアルがあります。External キーマテリアルオリジンを持つ KMS キーを作成するとき、KMS キーにはキーマテリアルがありません。後で、KMS キーにキーマテリアルをインポートすることができます。インポートしたキーマテリアルを使用する場合は、そのキーマテリアルを AWS KMS の外部で保護して管理する必要があります。これには、キーマテリアルの有効期限が切れた場合の置き換えも含まれます。詳細については、「インポートしたキーマテリアルについて」を参照してください。

インポートされたキーマテリアルを持つ KMS キーの作成方法については、「ステップ 1: キーマテリアルなしで KMS キーを作成する」を参照してください。

AWS_CLOUDHSM

AWS KMS は、AWS CloudHSM キーストアの AWS CloudHSM クラスターにキーマテリアルを作成します。

AWS CloudHSM キーストアで KMS キーを作成する方法については、「AWS CloudHSM キーストアでの KMS キーの作成」を参照してください。

EXTERNAL_KEY_STORE

キーマテリアルは、AWS の外部にある外部キーマネージャー内の暗号化キーです。このオリジンは、外部キーストアの KMS キーでのみサポートされています。

外部キーストアで KMS キーを作成する方法については、「外部キーストアで KMS キーを作成する」を参照してください。

キー仕様

キー仕様は、キーの暗号化設定を表すプロパティです。キー仕様の意味は、キータイプによって異なります。

- AWS KMS キー — キー仕様は、KMS キーが対称か非対称かを決定します。また、そのキーマテリアルのタイプとサポートするアルゴリズムも決定します。キー仕様は、KMS キーの作成時に選択します。キー仕様を変更することはできません。デフォルトのキースペックである SYMMETRIC_DEFAULT は、256 ビット対称暗号化キーを表します。

> **Note**
>
> KMS キーの KeySpec は、CustomerMasterKeySpec と呼ばれていました。CreateKey オペレーションの CustomerMasterKeySpecパラメータは廃止されました。代わりに、同様に機能する KeySpec パラメータを使用します。重大な変更を防ぐために、 CreateKeyおよび DescribeKeyオペレーションのレスポンスに、同じ値を持つ KeySpecおよび CustomerMasterKeySpecメンバーの両方が含まれるようになりました。

主要仕様のリストと主要仕様の選択に関するヘルプについては、「キー仕様の選択」を参照してください。KMS キーのキー仕様を確認するには、 DescribeKeyオペレーションを使用するか、 AWS KMSコンソールの KMS キーの詳細ページの暗号化設定タブを参照してください。ヘルプについては、「キーの表示」を参照してください。

KMS キーの作成時にプリンシパルが使用できるキー仕様を制限するには、kms:KeySpec 条件キーを使用します。kms:KeySpec 条件キーを使用して、特定のキー仕様を持つ KMS キーのみで AWS KMS オペレーションを呼び出すことをプリンシパルに許可することもできます。例えば、RSA_4096 キー仕様を持つ KMS キーの削除をスケジュールする許可を拒否できます。

- データキー (GenerateDataKey) — キー仕様によって AES データキーの長さが決まります。
- データキーペア (GenerateDataKeyPair) — キーペア仕様によって、データキーペア内のキーマテリアルのタイプが決まります。

キーの用途

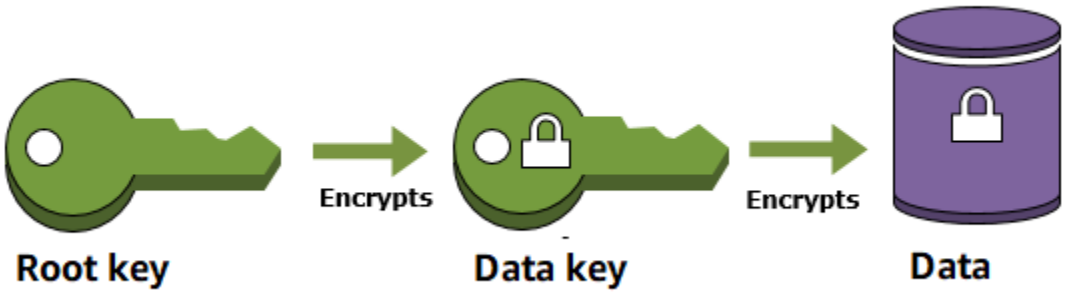
キー用途は、キーがサポートする暗号化オペレーションを決定するプロパティです。KMS キーには、ENCRYPT_DECRYPT、SIGN_VERIFY、GENERATE_VERIFY_MAC のキー用途を指定できます。各 KMS キーに指定できるキー用途は 1 つだけです。KMS キーを複数タイプのオペレーションに使用すると、両方のオペレーションの成果が攻撃に対してより脆弱になります。

KMS キーの用途を選択する方法については、「キーの用途の選択」を参照してください。KMS キーのキーの使用方法を確認するには、 DescribeKeyオペレーションを使用するか、 AWS KMSコンソールの KMS キーの詳細ページで暗号化設定タブを選択します。ヘルプについては、「キーの表示」を参照してください。

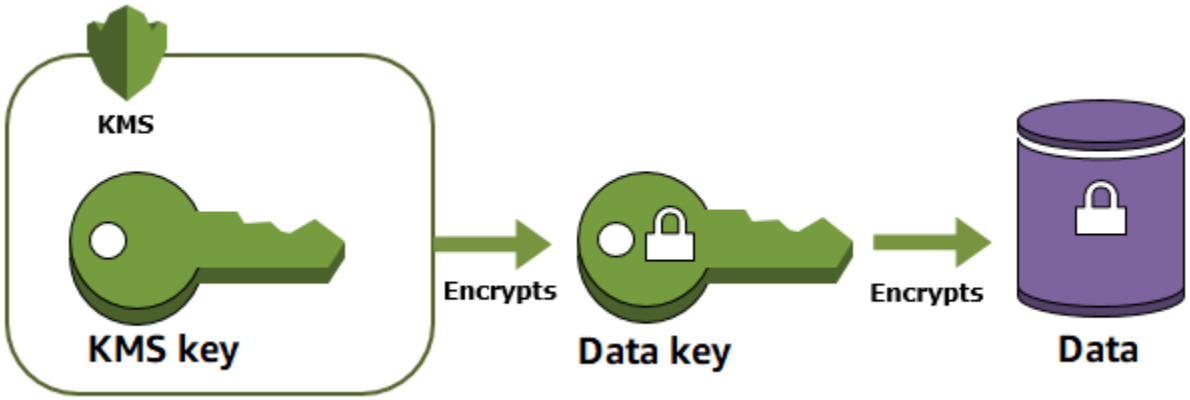
エンベロープ暗号化

データを暗号化するとデータは保護されますが、暗号化キーを保護する必要があります。1 つの方法としては、それを暗号化します。エンベロープ暗号化は、データキーでプレーンテキストデータを暗号化してから、そのデータキーを別のキーで暗号化する手法です。

データ暗号化キーを別の暗号化キーで暗号化し、その暗号化キーを別の暗号化キーで暗号化することもできます。しかし、最終的には、キーとデータを復号するために、1 つのキーをプレーンテキストで保持する必要があります。この最上位プレーンテキストキーの暗号化キーは、ルートキーと呼ばれます。



AWS KMS は、暗号化キーを安全に保存、管理して保護します。AWS KMS に保存されているルートキーは AWS KMS keys と呼ばれ、AWS KMS の FIPS 検証済みハードウェアセキュリティモジュールを暗号化されないままにしません。KMS キーを使用するには、AWS KMS を呼び出す必要があります。



エンベロープ暗号化には、いくつかの利点があります。

- データキーの保護

データキーを暗号化する場合、暗号化されたデータキーの保存について心配する必要がありません。これは、そのデータキーが暗号化によって本質的に保護されているためです。暗号化されたデータとともに、暗号化されたデータキーを安全に保存できます。

- 複数のキーを使用した同じデータの暗号化

  暗号化オペレーションには時間がかかります (特に、暗号化するデータが大きいオブジェクトである場合)。異なるキーで raw データを複数回にわたって再暗号化する代わりに、raw データを保護するデータキーのみを再暗号化できます。

- 複数のアルゴリズムの強度の結合

  一般に、対称キーアルゴリズムは、パブリックキーアルゴリズムよりも高速で、より小さい暗号文を生成します。一方、パブリックキーのアルゴリズムはロールを本質的に分離し、キー管理を簡単にします。エンベロープ暗号化により、それぞれの方法を組み合わせることができます。

暗号化コンテキスト

対称暗号化 KMS キーを使用するすべての AWS KMS 暗号化オペレーションは、暗号化コンテキストを受け入れます。これは、データに関する追加のコンテキスト情報が含まれている場合があるオプションのシークレットではないキーバリューペアのセットです。AWS KMS は、認証付き暗号化をサポートするための追加認証データ (AAD) として暗号化コンテキストを使用します。

暗号化リクエストに暗号化コンテキストが含まれている場合、暗号文に暗号化されてバインドされます。このため、データを復号する (または復号して再暗号化する) には、同じ暗号化コンテキストが必要です。復号リクエストで指定された暗号化コンテキストが大文字と小文字を区別して完全に一致しない場合、復号リクエストは失敗します。暗号化コンテキストでのキーと値のペアの順序のみを変更できます。

> **Note**
>
> 非対称 KMS キーまたは HMAC KMS キーを使用する暗号化オペレーションで暗号化コンテキストを指定することはできません。非対称アルゴリズムと MAC アルゴリズムは、暗号化コンテキストをサポートしません。

この暗号化コンテキストは秘密ではなく、暗号化されていません。これは AWS CloudTrail ログにプレーンテキストで表示されるため、それを使用して暗号化オペレーションを識別して分類できます。暗号化コンテキストには機密情報を含めないようにしてください。暗号化または復号されるデータの

内容を説明した暗号化テキストの使用をお勧めします。例えば、ファイルを暗号化するときは、ファイルパスの一部を暗号化コンテキストとして使用することもあります。

```
"encryptionContext": {
    "department": "10103.0"
}
```

例えば、Amazon Elastic Block Store (Amazon EBS) CreateSnapshotオペレーションで作成されたボリュームとスナップショットを暗号化する場合、Amazon EBS はボリューム ID を暗号化コンテキスト値として使用します。

```
"encryptionContext": {
  "aws:ebs:id": "vol-abcde12345abc1234"
}
```

暗号化コンテキストを使用して、アカウントの AWS KMS keys へのアクセスを絞り込むか、制限することもできます。暗号化コンテキストはグラントの制約として、およびポリシーステートメントの条件として使用できます。

暗号化コンテキストを使用して暗号化されたデータの整合性を保護する方法については、 AWS セキュリティブログの「 AWS Key Management Serviceと を使用して暗号化されたデータの整合性を保護する方法 EncryptionContext」を参照してください。

暗号化コンテキストの詳細。

暗号化コンテキストのルール

AWS KMS は、暗号化コンテキストのキーと値に以下のルールを適用します。

- 暗号化コンテキストペアのキーと値はシンプルなリテラル文字列であることが必要です。整数や浮動小数点数など別の型を使用する場合、AWS KMS では文字列として解釈されます。
- 暗号化コンテキストのキーと値には、Unicode 文字を含めることができます。暗号化コンテキストにキーポリシーまたは IAM ポリシーで許可されていない文字が含まれている場合は、`kms:EncryptionContext:context-key` および `kms:EncryptionContextKeys` などのポリシー条件キーで暗号化コンテキストを指定できなくなります。キーポリシードキュメントのルールに関する詳細については、「キーポリシー形式」を参照してください。IAM ポリシードキュメントのルールに関する詳細については、「IAM ユーザーガイド」の「IAM 名前の要件」を参照してください。

ポリシーでの暗号化コンテキスト

暗号化コンテキストは主に整合性と信頼性を検証するために使用されます。ただし、キーポリシーおよび IAM ポリシーで、対称暗号化 AWS KMS keysへのアクセスを制御するための暗号化コンテキストを使用することも可能です。

kms:EncryptionContext： および kms:EncryptionContextKeys 条件キーは、リクエストに特定の暗号化コンテキストキーまたはキーと値のペアが含まれている場合にのみ、アクセス許可を許可 (または拒否) します。

例えば、以下のキーポリシーステートメントでは、RoleForExampleApp ロールに Decrypt オペレーションでの KMS キーの使用を許可します。また、kms:EncryptionContext:*context-key* 条件キーを使用して、リクエストの暗号化コンテキストに AppName:ExampleApp 暗号化コンテキストペアが含まれる場合にのみ、このアクセス許可を付与します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

これらの暗号化コンテキスト条件キーの詳細については、「の条件キー AWS KMS」を参照してください。

グラントでの暗号化コンテキスト

グラントを作成するときは、グラント許可の条件を確立するグラント制約を含めることができます。AWS KMS は、EncryptionContextEquals と EncryptionContextSubset の 2 つのグラント制約をサポートし、これらの両方が暗号化オペレーションのリクエストに暗号化コンテキストを必要とします。これらのグラント制約を使用するとき、グラントの許可は、暗号化オペレーションのリクエストで暗号化コンテキストがグラント制約の要件を満たしている場合にのみ有効になります。

例えば、 GenerateDataKeyオペレーションを許可するグラントに 許可EncryptionContextEqualsの制約を追加できます。この制約により、このグラントは、リクエストの暗号化コンテキストがグラント制約の暗号化コンテキストで大文字と小文字を区別して一致する場合にのみ、この操作を許可します。

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/exampleUser \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --operations GenerateDataKey \
    --constraints EncryptionContextEquals={Purpose=Test}
```

被付与者プリンシパルからの次のようなリクエストは、EncryptionContextEquals の制約を満たします。

```
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --encryption-context Purpose=Test
```

グラント制約の詳細については、「グラントの制約の使用」を参照してください 。グラントの詳細については、「the section called “グラント”」を参照してください。

暗号化コンテキストのログ記録

AWS KMS が AWS CloudTrail を使用して暗号化コンテキストを記録することにより、アクセスされた KMS キーやデータを特定することができます。ログエントリには、ログエントリ内の暗号化コンテキストによって参照された特定のデータの暗号化または復号に、どの KMS キーが使用されたかが正確に示されます。

> **Important**
>
> 暗号化コンテキストは記録されるため、機密情報が含まれていてはなりません。

暗号化コンテキストの保存

Decrypt または ReEncrypt オペレーションを呼び出すときに暗号化コンテキストを簡単に使用できるように、暗号化コンテキストを暗号化されたデータとともに格納できます。暗号化や復号で必要

になったときに完全な暗号化コンテキストを作成できる十分な暗号化コンテキストのみを保存することをお勧めします。

例えば、暗号化コンテキストがファイルへの完全修飾パスである場合、そのパスの一部のみを、暗号化されたファイルの内容とともに保存します。次に、完全な暗号化コンテキストが必要になったら、保存されたフラグメントからコンテキストを再構築します。誰かがファイルを改ざんした場合 (名前の変更、別の場所への移動など)、暗号化コンテキスト値が変更され、復号リクエストは失敗します。

キーポリシー

KMS キーの作成時に、KMS キーを使用および管理できるユーザーを決定します。これらのアクセス許可は、キーポリシーと呼ばれるドキュメントに含まれます。キーポリシーを使用して、カスタマーマネージドキーに対するアクセス許可をいつでも追加、削除、変更できます。ただし、AWS マネージドキー のキーポリシーは編集できません。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。

グラント

グラントは、AWS プリンシパルに暗号化操作で AWS KMS keys の使用を許可するポリシーツールです。また、KMS キー (DescribeKey) を表示して、グラントの作成、管理をできるようにします。KMS キーへのアクセスを認可する際、グラントはキーポリシーおよび IAM ポリシーと共に考慮されます。グラントは、作成してそのアクセス許可を使用し、キーポリシーまたは IAM ポリシーを変更することなく削除できるため、一時的なアクセス許可としてよく使用されます。グラントは限定的であり、作成と取り消しが容易であるため、一時的なアクセス許可やよりきめ細かなアクセス許可を付与するためによく使用されます。

グラントの用語を含む、グラントの詳細については「AWS KMS でのグラント」を参照してください。

KMS キーの使用状況を監査する

を使用してAWS CloudTrail、キーの使用状況を監査できます。 は、アカウントの AWS API コールと関連イベントの履歴を含むログファイル CloudTrail を作成します。これらのログファイルには、AWS マネジメントコンソール、AWS SDK、コマンドラインツールで行ったすべての AWS KMS API リクエストが含まれます。また、AWS サービスによってお客様に代わって行われた AWS KMS へのリクエストも含まれます。これらのログファイルを使用して、KMS キーが使用された日時、リクエストされたオペレーション、リクエスタのアイデンティティ、ソースの IP アドレスなど

の重要な情報を見つけることができます。詳細については、AWS CloudTrail での口グ記録 および AWS CloudTrail ユーザーガイドを参照してください。

キー管理インフラストラクチャ

暗号化の一般的な手法では、AES (Advanced Encryption Standard) など一般に利用でき専門家によって検証されたアルゴリズムとシークレットキーを使用して暗号化し、復号する必要があります。暗号化での主な問題の 1 つは、キーを秘密にしておくのが非常に困難なことです。これは通常、キー管理インフラストラクチャ (KMI) のジョブです。AWS KMS はユーザーに代わってキーインフラストラクチャを運用します。AWS KMS は AWS KMS keys と呼ばれるルートキーを作成し、安全に保存します。AWS KMS の運用の詳細については、AWS Key Management Service 暗号化の詳細を参照してください。

キーの管理

AWS KMS の使用を開始するには、AWS KMS key を作成します。

このセクションのトピックでは、基本的な KMS キーである対称暗号化 KMS キーの作成から削除までの管理方法を説明します。キーの編集と表示、キーのタグ付け、キーの有効化と無効化、キーマテリアルのローテーション、および KMS キーの使用をモニタリングするための AWS ツールとサービスの使用方法がトピックに含まれます。また、AWS CloudFormation を使用して KMS キーを作成し管理する方法、および各 AWS KMS オペレーションで必要なキーステータスを示すキーステータスリファレンスに関する情報も含まれています。

他のタイプの KMS キーの作成、使用、および管理に関する詳細については、「特定用途のキー」を参照してください。

トピック

キーの作成

は、 AWS KMS keysでAWS Management Console、または CreateKeyオペレーションまたは AWS CloudFormation テンプレートを使用して作成できます。このプロセスの最中に、KMS キーのタイプ、そのリージョナリティ (単一リージョンまたはマルチリージョン)、およびキーマテリアルのオリジン (デフォルトでは AWS KMS がユーザーに代わって作成する) を選択します。KMS キー作成後に

これらのプロパティを変更することはできません。KMS キーのキーポリシーも設定します。これはいつでも変更できます。

このトピックでは、基本的な KMS キーである対称暗号化 KMS キーを、AWS KMS からのキーマテリアルを使用して単一リージョン用に作成する方法を説明します。この KMS キーを使用して、AWS のサービス内のリソースを保護することができます。対称暗号化 KMS キーの詳細については、「SYMMETRIC_DEFAULT キー仕様」を参照してください。他のタイプのキーの作成については、「特定用途のキー」を参照してください。

AWS のサービスで保存または管理するデータを暗号化するための KMS キーを作成している場合は、対称暗号化 KMS キーを作成します。AWS KMS と統合された AWS のサービスは、データの暗号化に対称暗号化 KMS キーのみを使用します。これらのサービスは、非対称 KMS キーを使用する暗号化をサポートしません。作成する KMS キーのタイプを決定する方法については、KMS キータイプの選択 を参照してください。

> **Note**
>
> 対称 KMS キーは、対称暗号化 KMS キーと呼ばれるようになりました。AWS KMS は、対称暗号化 KMS キー (デフォルトのタイプ) と、同じく対称キーである HMAC KMS キーの 2 種類の対称 KMS キーをサポートしています。

AWS KMS コンソールで KMS キーを作成するときは、エイリアス (フレンドリ名) を指定する必要があります。`CreateKey` オペレーションでは、新しい KMS キーのエイリアスは作成されません。新規または既存の KMS キーのエイリアスを作成するには、 CreateAliasオペレーションを使用します。AWS KMS のエイリアスの詳細については、「エイリアスの使用」を参照してください。

このトピックでは、対称暗号化 KMS キーの作成方法を説明します。次の表を使用して、さまざまなタイプの KMS キーを作成する手順を確認してください。

KMS キーの作成手順

KMS キータイプ	手順
対称暗号化キー (SYMMETRIC_DEFAULT)	the section called “対称暗号化 KMS キーの作成”
非対称キー	the section called “非対称 KMS キーを作成する”

KMS キータイプ	手順
HMAC キー	the section called “HMAC キーの作成”
マルチリージョンキー (任意のタイプ)	the section called “キーマテリアルがインポートされたプライマリキーを作成する” the section called “キーマテリアルがインポートされたレプリカキーを作成する”
インポートされたキーマテリアル (「Bring your own key — BYOK」)	the section called “ステップ 1: キーマテリアルなしで KMS キーを作成する”
AWS CloudHSM キーストア	the section called “AWS CloudHSM キーストアでの KMS キーの作成”
外部キーストア (「Hold your own key — HYOK」)	the section called “外部キーストアで KMS キーを作成する”

詳細はこちら:

- クライアント側の暗号化用のデータキーを作成するには、 GenerateDataKeyオペレーションを使用します。
- 暗号化または署名用の非対称 KMS キーを作成するには、「非対称 KMS キーを作成する」を参照してください。
- HMAC KMS キーを作成するには、「HMAC KMS キーの作成」を参照してください。
- インポートされたキーマテリアルで KMS キーを作成 (「Bring Your Own Key」) するには、「キーマテリアルをインポートするステップ 1: キーマテリアルなしで AWS KMS key を作成する」を参照してください。
- マルチリージョンのプライマリキーまたはレプリカキーを作成するには、「マルチリージョンキーを作成する」を参照してください。
- カスタムキーストアで KMS キーを作成するには (キーマテリアルのオリジンがカスタムキーストア (CloudHSM))、「AWS CloudHSM キーストアでの KMS キーの作成」を参照してください。
- AWS CloudFormation テンプレートを使用して KMS キーを作成するには、 AWS CloudFormation ユーザーガイドのAWS::KMS::Key「」を参照してください。

- 既存の KMS キーが対称か非対称かを判断するには、「非対称 KMS キーの識別」を参照してください。
- プログラムおよびコマンドラインインターフェイスのオペレーションで KMS キーを使用するには、キー ID またはキー ARN が必要です。詳細な手順については、「キー ID とキー ARN を検索する」を参照してください。
- KMS キーに適用されるクォータの詳細については、クォータ を参照してください。

トピック

- KMS キーを作成するためのアクセス許可
- 対称暗号化 KMS キーの作成

KMS キーを作成するためのアクセス許可

コンソールで、または API を使用して KMS キーを作成するには、IAM ポリシーで次のアクセス許可を持っている必要があります。可能な限り、条件キーを使用してアクセス許可を制限します。例えば、IAM ポリシーで kms:KeySpec 条件キーを使用して、プリンシパルが対称暗号化キーのみを作成することを許可できます。

キーを作成するプリンシパルの IAM ポリシーの例については、「KMS キーの作成をユーザーに許可する」を参照してください。

> **Note**
>
> タグとエイリアスを管理する許可をプリンシパルに付与する場合は注意が必要です。タグまたはエイリアスを変更すると、カスタマーマネージドキーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」を参照してください。

- kms:CreateKey は必須です。
- kms:CreateAlias は、新しい KMS キーごとにエイリアスが必要なコンソールで KMS キーを作成するために必要です。
- kms:TagResource は、KMS キーの作成時にタグを追加するために必要です。
- iam:CreateServiceLinkedRole マルチリージョンのプライマリキーを作成するには が必要です。詳細については、「マルチリージョンキーへのアクセスを制御する」を参照してください。

kms:PutKeyPolicy KMS キーの作成に アクセス許可は必要ありません。kms:CreateKey アクセス許可には、初期キーポリシーを設定する許可が含まれています。ただし、KMS キーの作成時にこのアクセス許可をキーポリシーに追加して、KMS キーへのアクセスを制御できるようにする必要があります。代わりに、 BypassLockoutSafetyCheckパラメータを使用する方法がありますが、これは推奨されません。

KMS キーは、作成された AWS アカウントに属します。KMS キーを作成する IAM ユーザーはキー所有者とはみなされないため、作成した KMS キーを使用または管理するためのアクセス許可が自動的に付与されません。他のプリンシパルと同様に、キー作成者は、キーポリシー、IAM ポリシー、またはグラントを使用してアクセス許可を取得する必要があります。ただし、kms:CreateKey アクセス許可を持つプリンシパルは、初期キーポリシーを設定し、キーを使用または管理するためのアクセス許可を自分自身に付与できます。

対称暗号化 KMS キーの作成

KMS キーは、AWS Management Console、または AWS KMS API を使用して作成することができます。

このトピックでは、基本的な KMS キーである対称暗号化 KMS キーを、AWS KMS からのキーマテリアルを使用して単一リージョン用に作成する方法を説明します。この KMS キーを使用して、AWS のサービス内のリソースを保護することができます。他のタイプのキーの作成については、「特定用途のキー」を参照してください。

対称暗号化 KMS キーの作成 (コンソール)

AWS Management Console を使用して AWS KMS keys (KMS キー) を作成します。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。

4. [Create key] (キーの作成) を選択します。
5. 対称暗号化 KMS キーを作成するには、[Key type]] (キーのタイプ) で [Symmetric] (対称) を選択します。

   AWS KMS コンソールで非対称 KMS キーを作成する方法については、「非対称 KMS キーを作成する (コンソール)」を参照してください。
6. [Key usage] (キーの使用) では、[Encrypt and decrypt] (暗号化および復号) オプションがすでに選択されています。

   MAC コードを生成して検証する KMS キーの作成方法については、「HMAC KMS キーの作成」を参照してください。
7. [Next] (次へ) を選択します。

   [Advanced options] (詳細オプション) については、「特定用途のキー」を参照してください。
8. KMS キーのエイリアスを入力します。エイリアス名の先頭を **aws/** にすることはできません。この **aws/** プレフィックスは、アカウント内の AWS マネージドキー を表すために、Amazon Web Services によって予約されます。

   > **Note**
   >
   > エイリアスを追加、削除、更新すると、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「エイリアスを使用して KMS キーへのアクセスを制御する」を参照してください。

   エイリアスは KMS キーを識別するために使用する表示名です。保護する予定のデータタイプ、または KMS キーで使用する予定のアプリケーションを示すエイリアスを選択することをお勧めします。

   エイリアスは AWS Management Console で KMS キーを作成するときに必要です。CreateKey オペレーションを使用する場合、これらはオプションです。
9. (オプション) KMS キーの説明を入力します。

   今すぐ説明を追加するか、キーの状態が `Pending Deletion` または `Pending Replica Deletion` でない限り、後でいつでも更新できます。既存のカスタマーマネージドキーの説明を追加、変更、または削除するには、 で説明を編集するAWS Management Consoleか、UpdateKeyDescriptionオペレーションを使用します。

10. (オプション) タグキーとオプションのタグ値を入力します。KMS キーに複数のタグを追加するには、[Add tag] (タグを追加) を選択します。

    > **Note**
    >
    > KMS キーのタグ付けまたはタグ解除により、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「タグを使用してKMS キーへのアクセスを制御する」を参照してください。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

11. [次へ] をクリックします。

12. KMS キーを管理できる IAM ユーザーとロールを選択します。

    > **Note**
    >
    > このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシパルに KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
    >
    > IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

13. (オプション) 選択した IAM ユーザーとロールがこの KMS キーを削除しないようにするには、ページの下部にある [Key deletion] (キーの削除) セクションで、[Allow key administrators to delete this key] (キー管理者にこのキーの削除を許可する) のチェックボックスをオフにします。

14. [次へ] をクリックします。

15. 暗号化オペレーションでキーを使用できる IAM ユーザーとロールを選択します。

> **Note**
>
> このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシパルに暗号化オペレーションで KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

16. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID 番号を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

> **Note**
>
> 外部アカウントでプリンシパルが KMS キーを使用できるようにするには、外部アカウントの管理者が、これらのアクセス許可を付与する IAM ポリシーを作成する必要があります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

17. [次へ] を選択します。
18. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。
19. [Finish] (完了) を選択し、KMS キーを作成します。

対称暗号化 KMS キーの作成 (AWS KMS API)

CreateKey オペレーションを使用して、すべてのタイプの AWS KMS keys を作成できます。以下の例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

> **Important**
>
> Description フィールドまたは Tags フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

次のオペレーションでは、最も一般的に使用される KMS キーを作成します。これは、AWS KMS で生成されたキーマテリアルにバックアップされた、単一リージョンの対称暗号化キーです。このオペレーションには必須パラメータはありません。ただし、キーポリシーを指定するために Policy パラメータが必要になる場合もあります。キーポリシー (PutKeyPolicy) を変更し、説明やタグなどのオプション要素をいつでも追加できます。また、非対称キー、マルチリージョンキー、インポートされたキーマテリアルを含むキー、およびカスタムキーストア内のキーも作成できます。

CreateKey オペレーションではエイリアスを指定することはできませんが、 CreateAliasオペレーションを使用して新しい KMS キーのエイリアスを作成できます。

次の例では、パラメータを指定せずに CreateKey オペレーションを呼び出します。このコマンドでは、すべてのデフォルト値が使用されます。これは、AWS KMS が生成したキーマテリアルで対称暗号化 KMS キーを作成します。

```
$ aws kms create-key
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "MultiRegion": false
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
```

```
    }
}
```

新規の KMS キーにキーポリシーを指定しない場合、CreateKey が適用するデフォルトのキーポリシーは、新規の KMS キーを作成するときに使用され、コンソールが適用するデフォルトのキーポリシーとは異なります。

例えば、 GetKeyPolicyオペレーションへのこの呼び出しは、 CreateKeyが適用されるキーポリシーを返します。これは、AWS アカウント に KMS キーへのアクセス許可を付与して、KMS キーの AWS Identity and Access Management (IAM) ポリシーを作成できるようにします。IAM ポリシーおよび KMS キーのキーポリシーの詳細については、「AWS KMS の認証とアクセスコントロール」を参照してください。

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

エイリアスの使用

エイリアスは、AWS KMS key のわかりやすい名前です。例えば、エイリアスを使用すると、1234abcd-12ab-34cd-56ef-1234567890ab の代わりに KMS キーを test-key として参照できます。

エイリアスを使用して、AWS KMSコンソール、 DescribeKeyオペレーション、および Encrypt やなどの暗号化オペレーション で KMS キーを識別できますGenerateDataKey。エイリアスを使用すると、AWS マネージドキー の認識が容易になります。これらの KMS キーのエイリアスは、常に aws/*<service-name>* の形式になります。例えば、Amazon DynamoDB の AWS マネージドキー

のエイリアスは aws/dynamodb です。プロジェクトやカテゴリの名前をエイリアスの前に付けるなど、プロジェクトに対して同様のエイリアス標準を設定できます。

ポリシーを編集したり、権限を管理したりすることなく、エイリアスに基づいて KMS キーへのアクセスを許可および拒否することもできます。この機能は、属性ベースのアクセスコントロール (ABAC) の AWS KMS サポートの一部です。詳細については、「エイリアスを使用して KMS キーへのアクセスを制御する」を参照してください。

エイリアスの機能の強みは、エイリアスに関連付けられている KMS キーをいつでも変更できることです。エイリアスを使用すると、コードの記述と保守が容易になります。例えば、エイリアスを使用して特定の KMS キーを参照し、KMS キーを変更するとします。この場合は、単にエイリアスを別の KMS キーに関連付けます。コードを変更する必要はありません。

エイリアスを使用すると、別の AWS リージョン で同じコードを再利用することも容易になります。複数のリージョンで同じ名前のエイリアスを作成し、各エイリアスをそのリージョンの KMS キーに関連付けます。コードが各リージョンで実行されると、エイリアスは関連付けられた KMS キーをそのリージョンで参照します。例については「アプリケーションでのエイリアスの使用」を参照してください。

KMS キーのエイリアスは、AWS KMSコンソール、 CreateAlias API、または AWS CloudFormation テンプレート を使用して作成できます。

AWS KMS API では、各アカウントとリージョンのエイリアスを完全に制御できます。API には、エイリアスの作成 (CreateAlias) 、エイリアス名とエイリアス ARNs の表示 (ListAliases) 、エイリアスに関連付けられた KMS キーの変更 (UpdateAlias) 、エイリアスの削除 () を行うオペレーションが含まれていますDeleteAlias。複数のプログラミング言語のエイリアスの管理例については、を参照してください the section called “エイリアスの使用”。

詳細については、次のリソースを参照してください。

- エイリアスを含む KMS キーの識別子については、キー識別子 (KeyId) を参照してください。
- AWS CloudFormation テンプレートを使用して KMS キーのエイリアスを作成する方法については、 AWS CloudFormation ユーザーガイドAWS::KMS::Aliasの「」を参照してください。
- KMS キーに関連付けられているエイリアスの検索については、エイリアス名とエイリアス ARN を見つける を参照してください。
- エイリアスのリソースクォータおよびエイリアスに関連する API オペレーションのレートクォータについては、「クォータ」を参照してください。
- 複数のプログラミング言語でのエイリアスの作成と管理の例については、「エイリアスの使用」を参照してください。

トピック

エイリアスについて

AWS KMS におけるエイリアスの仕組みについて説明します。

エイリアスは独立した AWS リソース

エイリアスは、KMS キーのプロパティではありません。エイリアスに対して実行するアクションは、エイリアスに関連付けられた KMS キーには影響しません。KMS キーのエイリアスを作成してエイリアスを更新し、別の KMS キーに関連付けることができます。関連付けられた KMS キーに影響を与えずに、エイリアスを削除することもできます。ただし、KMS キーを削除すると、その KMS キーに関連付けられているすべてのエイリアスが削除されます。

IAM ポリシーでリソースとしてエイリアスを指定した場合、ポリシーは、関連付けられた KMS キーではなく、エイリアスを参照します。

各エイリアスには 2 つの形式があります。

エイリアスを作成するときは、エイリアス名を指定します。AWS KMS は、エイリアス ARN を作成します。

- エイリアス ARN は、エイリアスを一意に識別する Amazon リソースネーム (ARN) です。

```
# Alias ARN
arn:aws:kms:us-west-2:111122223333:alias/<alias-name>
```

- エイリアス名は、アカウントとリージョンで一意です。AWS KMS API では、エイリアス名は常に `alias/` によってプレフィックスされます。そのプレフィックスは、AWS KMS コンソールで、除外されます。

```
# Alias name
```

```
alias/<alias-name>
```

エイリアスはシークレットではありません

エイリアスは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。エイリアス名には、機密情報や重要情報を含めないでください。

各エイリアスは、一度に 1 つの KMS キーに関連付けられる

エイリアスとその KMS キーは、同じアカウントとリージョンに存在する必要があります。

エイリアスは、同じ AWS アカウント およびリージョン内のカスタマーマネージドキーに関連付けることができます。ただし、エイリアスを AWS マネージドキー に関連付ける許可はありません。

例えば、このListAliases出力は、test-keyエイリアスが TargetKeyIdプロパティで表される 1 つのターゲット KMS キーにのみ関連付けられていることを示しています。

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
}
```

複数のエイリアスを同じ KMS キーに関連付けることができる

例えば、test-key と project-key のエイリアスを同じ KMS キーに関連付けることができます。

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
},
{
     "AliasName": "alias/project-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1516435200.399,
```

```
    "LastUpdatedDate": 1516435200.399
}
```

エイリアスは、アカウントとリージョン内で一意である必要があります

例えば、各アカウントとリージョンに test-key エイリアスを 1 つだけ持つことができます。エイリアスでは大文字と小文字が区別されますが、大文字と小文字だけが異なるエイリアスは非常にエラーが発生しやすくなります。エイリアス名は変更できません。ただし、エイリアスを削除して、目的の名前で新しいエイリアスを作成することはできます。

異なるリージョンに同じ名前のエイリアスを作成することができます

例えば、米国東部 (バージニア北部) に finance-key エイリアスを持つことができ、ヨーロッパ (フランクフルト) に finance-key エイリアスを持つことができます。各エイリアスは、そのリージョンの KMS キーに関連付けられます。コードが alias/finance-key のようなエイリアス名を参照している場合は、複数のリージョンで実行できます。各リージョンでは、異なる KMS キーが使用されます。詳細については、「アプリケーションでのエイリアスの使用」を参照してください。

エイリアスに関連付けられている KMS キーを変更できる

UpdateAlias オペレーションを使用して、エイリアスを別の KMS キーに関連付けることができます。例えば、finance-key エイリアスが 1234abcd-12ab-34cd-56ef-1234567890ab KMS キーに関連付けられている場合、0987dcba-09fe-87dc-65ba-ab0987654321 KMS キーに関連付けられるようにエイリアスを更新できます。

ただし、現在の KMS キーと新しい KMS キーが同じタイプ (両方とも対称または非対称、もしくは両方とも HMAC) であり、同じキー使用法 (ENCRYPT_DECRYPT、SIGN_VERIFY、GENERATE_VERIFY_MAC のいずれか) である必要があります。この制限により、エイリアスを使用するコードのエラーが防止されます。エイリアスを別のタイプのキーに関連付ける必要があり、リスクを軽減した場合は、エイリアスを削除して再作成できます。

一部の KMS キーにはエイリアスがない

AWS KMS コンソールで KMS キーを作成する場合、KMS キーに新しいエイリアスを割り当てる必要があります。ただし、 CreateKeyオペレーションを使用して KMS キーを作成する場合、エイリアスは必要ありません。また、 UpdateAliasオペレーションを使用してエイリアスに関連付けられている KMS キーを変更し、 DeleteAliasオペレーションを使用してエイリアスを削除することもできます。そのため、KMS キーには、複数のエイリアスを持つものもあれば、エイリアスを持たないものもあります。

AWS によってアカウントにエイリアスが作成される

AWS では AWS マネージドキー のアカウントでエイリアスを作成します。これらのエイリアスには alias/aws/*<service-name>*, 、のような形式の名前 alias/aws/s3があります。

一部の AWS エイリアスには KMS キーがありません。これらの定義済みエイリアスは、通常、サービスの使用をスタートすると、AWS マネージドキー に関連付けられます。

エイリアスを使用して KMS キーを識別する

エイリアス名またはエイリアス ARN を使用して、暗号化オペレーション 、、DescribeKeyおよび で KMS キーを識別できますGetPublicKey。(KMS キーが別の AWS アカウント にある場合は、そのキー ARN またはエイリアス ARN を使用する必要があります)。エイリアスは、他の AWS KMS オペレーションでは、KMS キーの有効な識別子ではありません 各 AWS KMS API オペレーションの有効なキー識別子については、AWS Key Management Service API リファレンスの KeyId パラメータの説明を参照してください。

エイリアス名、またはエイリアス ARN を使用して、IAM ポリシーの KMS キーを識別することはできません。エイリアスに基づいて KMS キーへのアクセスを制御するには、kms:RequestAlias または kms:ResourceAliases 条件キーを使用します。詳細については、「AWS KMS の ABAC」を参照してください。

エイリアスを管理する

認可されたユーザーは、エイリアスを作成、表示、削除できます。エイリアスを更新することもできます。その場合、既存のエイリアスは別の KMS キーに関連付けられます。

トピック

- エイリアスの作成
- エイリアスの表示
- エイリアスの更新
- エイリアスの削除

エイリアスの作成

エイリアスは AWS KMS コンソールで、または AWS KMS API オペレーションを使用して作成できます。

エイリアスは 1 ~ 256 文字の文字列である必要があります。エイリアス名に使用できるのは、英数字、スラッシュ (/)、アンダースコア (_)、およびダッシュ (-) のみです。カスタマーマネージドキーのエイリアス名を alias/aws/ で始めることはできません。alias/aws/ プレフィックスは AWS マネージドキー のために予約されます。

新規の KMS キーまたは既存の KMS キーのエイリアスを作成できます。エイリアスを追加すると、特定の KMS キーをプロジェクトまたはアプリケーションで使用できます。

エイリアスを作成する (コンソール)

AWS KMS コンソールで KMS キーを作成する場合、新しい KMS キーのエイリアスを作成する必要があります。既存の KMS キーのエイリアスを作成するには、KMS キーの詳細ページにある [Aliases] (エイリアス) タブをクリックします。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。AWS マネージドキー または AWS 所有のキー のエイリアスを管理することはできません。
4. テーブルで、KMS キーのキー ID またはエイリアスを選択します。次に、KMS キーの詳細ページで、[Aliases] (エイリアス) タブをクリックします。

   KMS キーに複数のエイリアスがある場合は、テーブルの [Aliases] (エイリアス) 列に、1 つのエイリアスと (+n 個以上) などのエイリアスの概要が表示されます。エイリアスの概要を選択すると、KMS キーの詳細ページの [Aliases] (エイリアス) タブを直接表示します。
5. [Aliases] (エイリアス) タブで、[Create alias] (エイリアスの作成) を選択します。エイリアス名を入力し、[Create alias] (エイリアスの作成) を選択します。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

> **Note**
>
> alias/ プレフィックスを追加しないでください。コンソールが自動的に追加します。alias/ExampleAlias を入力すると、実際のエイリアス名は alias/alias/ExampleAlias になります。

エイリアスを作成する (AWS KMS API)

エイリアスを作成するには、CreateAliasオペレーションを使用します。コンソールで KMS キーを作成するプロセスとは異なり、CreateKeyオペレーションは新しい KMS キーのエイリアスを作成しません。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

CreateAlias オペレーションを使用して、エイリアスを持たない新規 KMS キーのエイリアスを作成できます。CreateAlias オペレーションを使用して、既存の KMS キーにエイリアスを追加したり、誤って削除されたエイリアスを再作成したりできます。

AWS KMS API オペレーションでは、エイリアス名は alias/ で始まり、その後に名前が続きます (alias/ExampleAlias など)。エイリアスはアカウントと リージョンで一意であることが必要です。既に使用されているエイリアス名を検索するには、ListAliasesオペレーションを使用します。エイリアス名では、大文字と小文字が区別されます。

TargetKeyId は、同じ AWS リージョン 内の任意のカスタマーマネージドキーです。KMS キーを識別するには、そのキー ID またはキー ARN を使用します。別のエイリアスを使用することはできません。

次の例では、example-key エイリアスを作成し、指定した KMS キーに関連付けます。これらの例では、AWS Command Line Interface (AWS CLI) を使用します。複数のプログラミング言語の例については、「エイリアスの使用」を参照してください。

```
$ aws kms create-alias \
    --alias-name alias/example-key \
```

```
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

CreateAlias は出力を返しません。新しいエイリアスを確認するには、 ListAliases オペレーションを使用します。詳細については、「エイリアスを表示する (AWS KMS API)」を参照してください。

エイリアスの表示

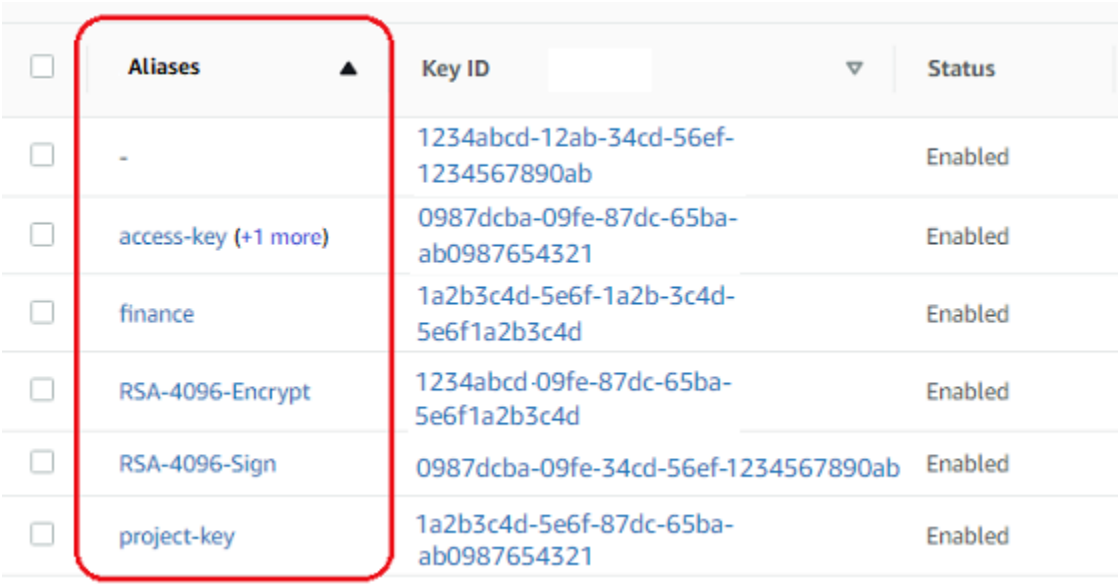
エイリアスを使用すると、AWS KMS コンソールで KMS キーを簡単に認識できます。KMS キーのエイリアスは、AWS KMSコンソールで、または ListAliasesオペレーションを使用して表示できます。KMS キーのプロパティを返す DescribeKeyオペレーションには、エイリアスは含まれません。

エイリアスの表示 (コンソール)

AWS KMS コンソールの [Customer managed keys] (カスタマーマネージドキー) ページおよび [AWS マネージドキー] ページには、各 KMS キーに関連付けられたエイリアスが表示されます。エイリアスに基づいて、KMS キーを検索、ソート、フィルタリングすることもできます。

次の AWS KMS コンソールの図では、サンプルアカウントの [カスタマー管理型のキー] ページのエイリアスを示します。イメージで示されているように、一部の KMS キーにはエイリアスがありません。

KMS キーに複数のエイリアスがある場合は、[Aliases] (エイリアス) 列に、1 つのエイリアスとエイリアスの概要 (+n 個以上) が表示されます。エイリアスの概要では、KMS キーに関連付けられている追加のエイリアスの数と、KMS キーのすべてのエイリアスの表示へのリンクが、[Aliases] (エイリアス) タブに表示されます。

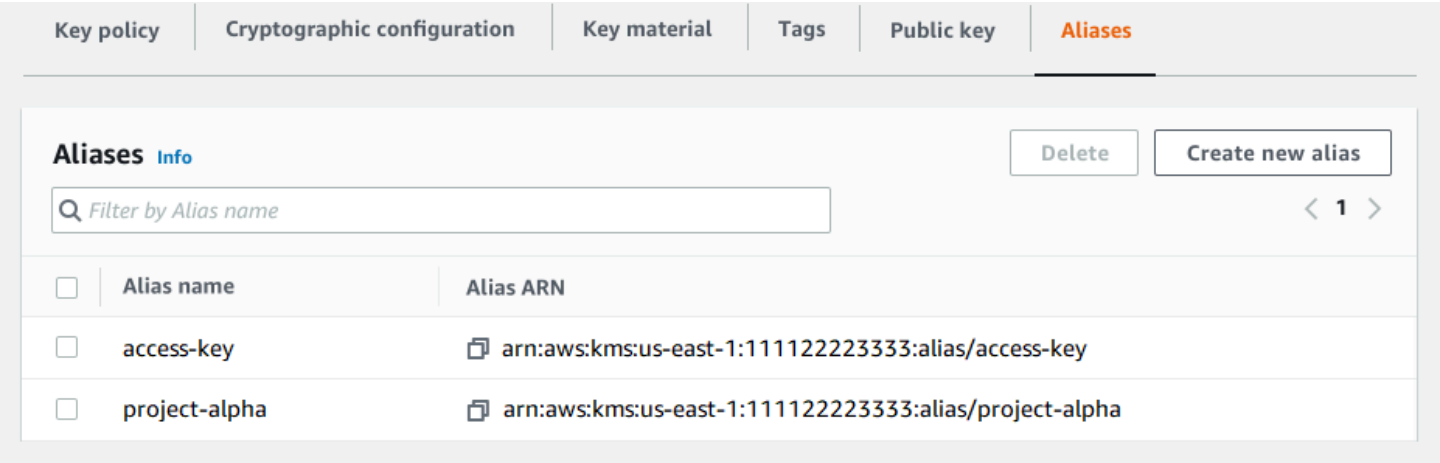


各 KMS キーの詳細ページの [Aliases] (エイリアス) タブには、AWS アカウント とリージョンの KMS キーのすべてのエイリアスのエイリアス名とエイリアス ARN が表示されます。[Aliases] (エイリアス) を使用して、エイリアスの作成とエイリアスの削除を行うこともできます。

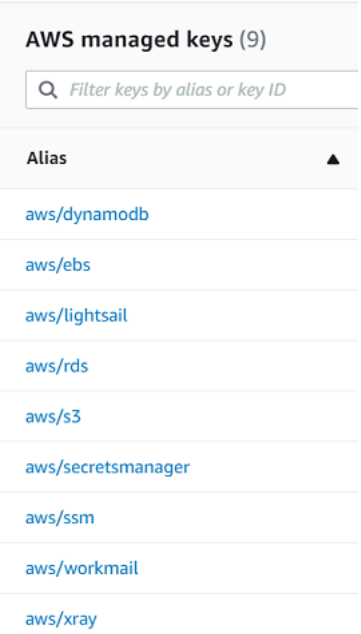
KMS キーのすべてのエイリアスのエイリアス名とエイリアス ARN を検索するには、[Aliases] (エイリアス) タブを使用します。

- [Aliases] (エイリアス) タブに直接移動するには、[Aliases] (エイリアス) 列で、エイリアスの概要 (+n個以上) を選択します。エイリアスの概要は、KMS キーに複数のエイリアスがある場合にのみ表示されます。
- または、KMS キーのエイリアスまたはキー ID を選択し (KMS キーの詳細ページが開きます)、[Aliases] (エイリアス) タブを選択します。これらのタブは、[General configuration] (一般設定) セクションにあります。

次のイメージは、サンプル KMS キーの [Aliases] (エイリアス) タブを示しています。



このエイリアスを使用して、この AWS マネージドキー ページの例に示されているように、AWS マネージドキー を認識できます。AWS マネージドキー のエイリアスでは、形式は常に aws/*<service-name>* です。例えば、Amazon DynamoDB の AWS マネージドキー のエイリアスは aws/dynamodb です。



エイリアスを表示する (AWS KMS API)

ListAliases オペレーションは、アカウントとリージョンのエイリアスのエイリアス名とエイリアス ARN を返します。出力には、AWS マネージドキー および カスタマーマネージドキーのエイリアスが含まれます。AWS マネージドキー のエイリアスは、 aws/*<service-name>*の形式になります (例: aws/dynamodb)。

レスポンスには、TargetKeyId フィールドがないエイリアスが含まれている場合もあります。これらは AWS が作成した定義済みのエイリアスですが、まだ KMS キーとは関連付けられていません。

```
$ aws kms list-aliases
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
```

```
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        }
    ]
}
```

特定の KMS キーに関連付けられているすべてのエイリアスを取得するには、ListAliases オペレーションのオプションの KeyId パラメータを使用します。KeyId パラメータは、KMS キーのキー ID またはキー ARN を受け取ります。

この例では、0987dcba-09fe-87dc-65ba-ab0987654321 KMS キーに関連付けられているすべてのエイリアスを取得します。

```
$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
    "Aliases": [
        {
```

```
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": "2018-01-20T15:23:10.194000-07:00",
            "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        }
    ]
}
```

KeyId パラメータはワイルドカード文字を使用しませんが、プログラミング言語の機能を使用して応答をフィルタリングできます。

例えば、次の AWS CLI コマンドでは、AWS マネージドキー のエイリアスのみを取得します。

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

次のコマンドは、 access-key エイリアスのみを取得します。エイリアス名では、大文字と小文字が区別されます。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/access-key`]'
[
    {
        "AliasName": "alias/access-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": "2018-01-20T15:23:10.194000-07:00",
        "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    }
]
```

エイリアスの更新

エイリアスは独立したリソースであるため、エイリアスに関連付けられている KMS キーを変更することができます。例えば、エイtest-keyリアスが 1 つの KMS キーに関連付けられている場合、UpdateAliasオペレーションを使用して別の KMS キーに関連付けることができます。これは、キー

マテリアルを変更せずに KMS キーを手動でローテーションする方法の 1 つです。KMS キーを更新し、新しいリソースに特定の KMS キーを使用していたアプリケーションで、別の KMS キーを使用することもできます。

AWS KMS コンソールでエイリアスを更新することはできません。また、 UpdateAlias （ または他のオペレーション) を使用してエイリアス名を変更することはできません。エイリアス名を変更するには、現在のエイリアスを削除してから KMS キーの新しいエイリアスを作成します。

エイリアスを更新するときは、現在の KMS キーと新しい KMS キーが同じタイプ (両方とも対称または非対称、もしくは HMAC) である必要があります。これらのキーの用途も同じである必要もあります (ENCRYPT_DECRYPT、SIGN_VERIFY、または GENERATE_VERIFY_MAC)。この制限により、エイリアスを使用するコードの暗号化エラーが防止されます。

次の例では、 ListAliasesオペレーションを使用して、エイリアスが現在 test-key KMS キー に関連付けられていることを示します1234abcd-12ab-34cd-56ef-1234567890ab。

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/test-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        }
    ]
}
```

次に、UpdateAlias オペレーションを使用して、test-key エイリアスに関連付けられている KMS キーを、KMS キー 0987dcba-09fe-87dc-65ba-ab0987654321 に変更します 現在関連付けられている KMS キーを指定する必要はありません。新しい (「ターゲット」) KMS キーのみを指定します。エイリアス名では、大文字と小文字が区別されます。

```
$ aws kms update-alias --alias-name 'alias/test-key' --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321
```

エイリアスが現在、ターゲット KMS キーに関連付けられていることを確認するには、再度、ListAliases オペレーションを使用します。この AWS CLI コマンドは、--query

パラメータを使用して、test-key エイリアスのみを取得します。TargetKeyId および LastUpdatedDate フィールドが更新されます。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[
    {
        "AliasName": "alias/test-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1593622000.191,
        "LastUpdatedDate": 1604958290.154
    }
]
```

エイリアスの削除

エイリアスは、AWS KMSコンソールで、または DeleteAliasオペレーションを使用して削除できます。エイリアスを削除する前に、そのエイリアスが使用されていないことを確認してください。エイリアスを削除しても関連付けられている KMS キーには影響しませんが、そのエイリアスを使用するアプリケーションに問題が発生する可能性があります。エイリアスを誤って削除した場合は、同じ名前の新しいエイリアスを作成し、同じまたは別の KMS キーに関連付けることができます。

KMS キーを削除すると、その KMS キーに関連付けられているすべてのエイリアスが削除されます。

エイリアスを削除する (コンソール)

AWS KMS コンソールでエイリアスを削除するには、KMS キーの詳細ページの [Aliases] (エイリアス) タブを使用します。KMS キーの複数のエイリアスを一度に削除できます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。AWS マネージドキー または AWS 所有のキー のエイリアスを管理することはできません。
4. テーブルで、KMS キーのキー ID またはエイリアスを選択します。次に、KMS キーの詳細ページで、[Aliases] (エイリアス) タブをクリックします。

KMS キーに複数のエイリアスがある場合は、テーブルの [Aliases] (エイリアス) 列に、1 つのエイリアスと (+n 個以上) などのエイリアスの概要が表示されます。エイリアスの概要を選択すると、KMS キーの詳細ページの [Aliases] (エイリアス) タブを直接表示します。

5. [Aliases] (エイリアス) タブで、削除するエイリアスの横にあるチェックボックスをオンにします。その後、[削除] をクリックします。

エイリアスを削除する (AWS KMS API)

エイリアスを削除するには、 DeleteAliasオペレーションを使用します。このオペレーションでは、エイリアスは一度に 1 つずつ削除されます。エイリアス名では、大文字と小文字が区別されます。また、エイリアス名の先頭には必ず `alias/` プレフィックスが付いています。

例えば、次のコマンドは、 `test-key` エイリアスを削除します。このコマンドは出力を返しません。

```
$ aws kms delete-alias --alias-name alias/test-key
```

エイリアスが削除されたことを確認するには、 ListAliasesオペレーションを使用します。次のコマンドでは、AWS CLI の `--query` パラメータを使用して、`test-key` エイリアスのみを取得します。応答の空の括弧は、 `ListAliases` 応答に `test-key` エイリアスが含まれていないことを示します。括弧を削除するには、 `--output text` パラメータと値を使用します。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[]
```

アプリケーションでのエイリアスの使用

エイリアスを使用して、アプリケーションコードで KMS キーを表示できます。AWS KMS 暗号化オペレーション 、DescribeKey、および の `KeyId`パラメータは、エイリアス名またはエイリアス ARN GetPublicKeyを受け入れます。

例えば、次の `GenerateDataKey` コマンドでは、エイリアス名 (`alias/finance`) を使用して KMS キーを識別します。エイリアス名は、 `KeyId` パラメータの値です。

```
$ aws kms generate-data-key --key-id alias/finance --key-spec AES_256
```

KMS キーが別の AWS アカウント にある場合は、これらのオペレーションで、キー ARN またはエイリアス ARN を使用する必要があります。エイリアス ARN を使用する際、KMS キーのエイリアス

は KMS キーを所有するアカウントで定義され、リージョンごとに異なる場合があることに注意してください。エイリアス ARN を検索する方法については、エイリアス名とエイリアス ARN を見つける を参照してください。

例えば、次の GenerateDataKey コマンドでは、発信者のアカウントに含まれていない KMS キーを使用します。ExampleAlias エイリアスは、指定したアカウントおよびリージョンの KMS キーに関連付けられます。

```
$ aws kms generate-data-key --key-id arn:aws:kms:us-west-2:444455556666:alias/
ExampleAlias --key-spec AES_256
```

エイリアスの最も強力な使用法の 1 つは、アプリケーションを複数の AWS リージョン で実行する場合です。例えば、署名と検証に RSA 非対称 KMS キーを使用するグローバルなアプリケーションがあるとします。

- 米国西部 (オレゴン) (s-west-2) では、arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab を使用します。
- ヨーロッパ (フランクフルト) (eu-central-1) では arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321 を、
- アジアパシフィック (シンガポール) (ap-southeast-1) では、arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d を使用します。

各リージョンで異なるバージョンのアプリケーションを作成するか、ディクショナリまたはスイッチステートメントを使用して、各リージョンに適切な KMS キーを選択できます。ただし、各リージョンで同じエイリアス名を持つエイリアスを作成する方がはるかに簡単です。エイリアス名では、大文字と小文字が区別されます。

```
aws --region us-west-2 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

aws --region eu-central-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321
```

```
aws --region ap-southeast-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:ap-
southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
```

次に、コード内でエイリアスを使用します。コードが各リージョンで実行されると、エイリアスは、そのリージョンの関連付けられた KMS キーを参照します。例えば、このコードは、エイリアス名を使用して Sign オペレーションを呼び出します。

```
aws kms sign --key-id alias/new-app \
    --message $message \
    --message-type RAW \
    --signing-algorithm RSASSA_PSS_SHA_384
```

ただし、エイリアスが削除または更新され、別の KMS キーに関連付けられるリスクがあります。この場合、エイリアス名を使用した署名の検証に失敗し、エイリアスの再作成または更新が必要になる場合があります。

このリスクを軽減するには、アプリケーションで使用するエイリアスを管理する権限をプリンシパルに与えることに注意する必要があります。詳細については、「エイリアスへのアクセスの制御」を参照してください。

複数の AWS リージョン (AWS Encryption SDK など) のデータを暗号化するアプリケーションには、他にも複数のソリューションがあります。

エイリアスへのアクセスの制御

エイリアスを作成または変更すると、エイリアスとそのエイリアスに関連付けられた KMS キーが影響を受けます。このため、エイリアスを管理するプリンシパルには、エイリアスおよび影響を受けるすべての KMS キーに対してエイリアスオペレーションを呼び出す許可が必要です。これらの許可は、キーポリシー、IAM ポリシー、権限を使用することで付与できます。

> **Note**
>
> タグとエイリアスを管理する許可をプリンシパルに付与する場合は注意が必要です。タグまたはエイリアスを変更すると、カスタマーマネージドキーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「エイリアスを使用して KMS キーへのアクセスを制御する」を参照してください。

すべての AWS KMS オペレーションに対するアクセスの制御については、「アクセス許可に関するリファレンス」を参照してください。

エイリアスを作成および管理するための権限は、次のように機能します。

kms:CreateAlias

エイリアスを作成するには、プリンシパルに、エイリアスおよび関連付けられた KMS キーの両方に対する次の許可が必要です。

- `kms:CreateAlias` エイリアスの場合。エイリアスの作成を許可されたプリンシパルにアタッチされた IAM ポリシーでこのアクセス許可を提供します。

  次のポリシーステートメントの例では、`Resource` エレメントの特定のエイリアスを指定します。ただし、複数のエイリアス ARN を一覧表示したり、「test*」などのエイリアスパターンを指定したりできます。`Resource` 値を "*" に指定すると、プリンシパルがアカウントとリージョンで任意のエイリアスを作成できるようになります。エイリアスを作成する権限は、アカウントおよびリージョン内のすべてのリソースに対する `kms:Create*` 権限に含めることもできます。

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- KMS キーの `kms:CreateAlias` このアクセス許可は、キーポリシーまたはキーポリシーから委任された IAM ポリシーで指定する必要があります。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:CreateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
```

```
}
```

条件キーを使用して、エイリアスと関連付けることができる KMS キーを制限できます。例えば、kms:KeySpec 条件キーを使用して、プリンシパルが非対称 KMS キーでのみエイリアスを作成できるようにします。KMS キーリソースに対する kms:CreateAlias 許可の制限に使用できる条件キーの詳細なリストについては、AWS KMS アクセス許可 を参照してください。

kms:ListAliases

アカウントとリージョンのエイリアスを一覧表示するには、プリンシパルに IAM ポリシーの kms:ListAliases アクセス権限が必要です。このポリシーは特定の KMS キーまたはエイリアスリソースに関連付けられていないため、ポリシー内のリソース要素の値が "*" である必要があります。

例えば、次の IAM ポリシーステートメントでは、アカウントとリージョン内のすべての KMS キーとエイリアスを一覧表示する許可をプリンシパルに付与します。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

kms:UpdateAlias

エイリアスに関連付けられている KMS キーを変更するには、プリンシパルにエイリアス用、現在の KMS キー用、新規の KMS キー用の 3 つの許可要素が必要です。

例えば、test-key エイリアスをキー ID 1234abcd-12ab-34cd-56ef-1234567890ab の KMS キーから、キー ID 0987dcba-09fe-87dc-65ba-ab0987654321 の KMS キーに変更するとします。その場合は、このセクションの例に似たポリシーステートメントを含めます。

- kms:UpdateAlias エイリアスの場合。このアクセス権限は、プリンシパルにアタッチされている IAM ポリシーで提供します。次の IAM ポリシーは、特定のエイリアスを指定します。ただし、

複数のエイリアス ARN を一覧表示したり、"test*" などのエイリアスパターンを指定したりできます。"*" の Resource 値を指定することで、プリンシパルがアカウントとリージョンで任意のエイリアスを更新できるようにもなります。

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:UpdateAlias",
    "kms:ListAliases",
    "kms:ListKeys"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- 現在エイリアスに関連付けられている KMS キーの kms:UpdateAlias。このアクセス許可は、キーポリシーまたはキーポリシーから委任された IAM ポリシーで指定する必要があります。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

- kms:UpdateAliasオペレーションでエイリアスに関連付けられている KMS キーの 。このアクセス許可は、キーポリシーまたはキーポリシーから委任された IAM ポリシーで指定する必要があります。

```
{
  "Sid": "Key policy for 0987dcba-09fe-87dc-65ba-ab0987654321",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
```

```
}
```

条件キーを使用して、UpdateAlias オペレーションの KMS キーの一方または両方を制限できます。例えば、kms:ResourceAliases 条件キーを使用して、ターゲット KMS キーに既に特定のエイリアスがある場合にのみ、プリンシパルがエイリアスを更新できるようにします。KMS キーリソースに対する kms:UpdateAlias 許可の制限に使用できる条件キーの詳細なリストについては、AWS KMS アクセス許可 を参照してください。

kms:DeleteAlias

エイリアスを削除するには、プリンシパルにエイリアスおよび関連付けられた KMS キーに対する許可が必要です。

プリンシパルにリソースを削除する権限を与えるときは、いつものように注意する必要があります。ただし、エイリアスを削除しても関連付けられた KMS キーには影響しません。エイリアスに依存するアプリケーションでエラーが発生する可能性がありますが、エイリアスを誤って削除した場合は、エイリアスを再作成することができます。

- kms:DeleteAlias エイリアスの場合。エイリアスの削除を許可されているプリンシパルにアタッチされた IAM ポリシーでこのアクセス許可を提供します。

  次のポリシーステートメントの例は、 Resource エレメントのエイリアスを指定します。ただし、複数のエイリアス ARN を一覧表示したり、"test*" などのエイリアスパターンを指定したりできます。"*" の Resource 値を指定して、プリンシパルにアカウントとリージョンで任意のエイリアスを削除させることもできます。

  ```
  {
    "Sid": "IAMPolicyForAnAlias",
    "Effect": "Allow",
    "Action": [
      "kms:CreateAlias",
      "kms:UpdateAlias",
      "kms:DeleteAlias"
    ],
    "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
  }
  ```

- 関連付けられた KMS キーの kms:DeleteAlias。このアクセス許可は、キーポリシーまたはキーポリシーから委任された IAM ポリシーで指定する必要があります。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"
  },
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

エイリアスのアクセス許可を制限する

リソースが KMS キーの場合、条件キーを使用してエイリアスのアクセス許可を制限できます。例えば、次の IAM ポリシーでは、特定のアカウントとリージョンの KMS キーに対するエイリアスオペレーションを許可します。ただし、kms:KeyOrigin 条件キーを使用して、 のキーマテリアルを持つ KMS キーへのアクセス許可をさらに制限しますAWS KMS。

KMS キーリソースに対するエイリアスのアクセス許可の制限に使用できる条件キーの詳細なリストについては、AWS KMS アクセス許可 を参照してください。

```
{
  "Sid": "IAMPolicyKeyPermissions",
  "Effect": "Allow",
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_KMS"
    }
  }
}
```

リソースがエイリアスのポリシーステートメントでは、条件キーを使用できません。プリンシパルが管理できるエイリアスを制限するには、エイリアスへのアクセスを制御する IAM ポリシーステートメントの Resource 要素の値を使用します。例えば、次のポリシーステートメントでは、エイリアスが Restricted で始まる場合を除き、プリンシパルが AWS アカウント とリージョンで任意のエイリアスを作成、更新、削除できるようにします。

```
{
  "Sid": "IAMPolicyForAnAliasAllow",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/*"
},
{
  "Sid": "IAMPolicyForAnAliasDeny",
  "Effect": "Deny",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/Restricted*"
}
```

エイリアスを使用して KMS キーへのアクセスを制御する

KMS キーに関連付けられたエイリアスに基づいて、KMS キーへのアクセスを制御できます。そのためには、kms:RequestAlias および kms:ResourceAliases 条件キーを使用します。この機能は、属性ベースのアクセスコントロール (ABAC) の AWS KMS サポートの一部です。

kms:RequestAlias 条件キーは、リクエスト内のエイリアスに基づいて、KMS キーへのアクセスを許可または拒否します。kms:ResourceAliases 条件キーは、KMS キーに関連付けられたエイリアスに基づいて、KMS キーへのアクセスを許可または拒否します。

これらの機能では、ポリシーステートメントの resource 要素のエイリアスを使用して KMS キーを識別することはできません。エイリアスが resource 要素の値の場合、ポリシーは関連付けられている KMS キーではなく、エイリアスリソースに適用されます。

> **Note**
>
> タグとエイリアスの変更が KMS キーの認可に影響を及ぼすまでに最長 5 分かかることがあります。最近の変更は、認可に影響を与える前に API オペレーションで表示される場合があります。

エイリアスを使用して KMS キーへのアクセスを制御する際は、次の点を考慮してください。

- エイリアスを使用して、最小特権アクセスのベストプラクティスを強化します。IAM プリンシパルでは、使用または管理する必要がある KMS キーのみに対して、必要なアクセス許可のみを付与します。例えば、エイリアスを使用してプロジェクトに使用される KMS キーを識別します。次に、プロジェクトエイリアスを持つ KMS キーのみを使用する許可をプロジェクトチームに付与します。
- プリンシパルにエイリアスを追加、編集、削除できる `kms:CreateAlias`、`kms:UpdateAlias`、`kms:DeleteAlias` 許可を付与する際は注意してください。エイリアスを使用して KMS キーへのアクセスを制御する際、タグを変更することで、使用許可のない KMS キーに対する使用許可をプリンシパルに付与してしまう可能性があります。他のプリンシパルがジョブを実行するために必要な KMS キーへのアクセスを拒否することもできます。
- 現在、エイリアスを管理する許可を持つ AWS アカウント でプリンシパルを確認し、必要に応じて許可を調整します。キーポリシーを変更したり、権限を作成したりする許可のないキー管理者も、エイリアスを管理する許可があれば、KMS キーへのアクセスを制御できます。

  例えば、コンソールのキー管理者のデフォルトキーポリシーには、`kms:CreateAlias`、`kms:DeleteAlias`、`kms:UpdateAlias` アクセス許可があります。IAM ポリシーでは、AWS アカウント のすべての KMS キーに対するエイリアスのアクセス許可を付与する可能性があります。例えば、 AWSKeyManagementServicePowerUser管理ポリシーでは、プリンシパルはすべての KMS キーのエイリアスを作成、削除、一覧表示できますが、更新することはできません。
- エイリアスに依存するポリシーを設定する前に、AWS アカウント で KMS キーのエイリアスを確認します。含めるエイリアスにのみポリシーが適用されることを確認します。CloudTrail ログとCloudWatch アラームを使用して、KMS キーへのアクセスに影響を与える可能性のあるエイリアスの変更を警告します。また、各エイリアスの作成日と最終更新日もListAliasesレスポンスに含まれます。

- エイリアスポリシー条件はパターンマッチングを使用します。エイリアスの特定のインスタンスには関連付けられません。エイリアスベースの条件キーを使用するポリシーは、パターンに一致するすべての新規および既存のエイリアスに影響します。ポリシー条件に一致するエイリアスを削除して再作成すると、古いエイリアスの場合と同様に、新しいエイリアスに条件が適用されます。

kms:RequestAlias 条件キーは、オペレーションリクエストで明示的に指定されたエイリアスに依存します。kms:ResourceAliases 条件キーは、リクエストに表示されない場合でも、KMS キーに関連付けられているエイリアスに依存します。

kms:RequestAlias

リクエスト内の KMS キーを識別するエイリアスに基づいて、KMS キーへのアクセスを許可または拒否します。kms:RequestAlias 条件キーは、キーポリシーまたは IAM ポリシーで使用できます。これは、エイリアスを使用してリクエスト内の KMS キーを識別するオペレーション、つまり暗号化オペレーション 、、DescribeKeyおよび に適用されますGetPublicKey。CreateAlias や などのエイリアスオペレーションには無効ですDeleteAlias。

条件キーで、エイリアス名またはエイリアス名パターンを指定します。エイリアス ARN を指定することはできません。

例えば、次のポリシーステートメントでは、プリンシパルが KMS キーに対して、指定されたオペレーションを使用できるようにします。アクセス許可は、KMS キーを識別する alpha を含むエイリアスをリクエストが使用する場合にのみ有効です。

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/alpha-developer"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:RequestAlias": "alias/*alpha*"
    }
  }
```

```
}
```

次の認可済みプリンシパルからのサンプルリクエストは条件を満たします。ただし、これらの値が同じ KMS キーを識別したとしても、キー ID、キー ARN、別のエイリアスが条件を満たさない可能性があります。

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-
alpha"
```

kms:ResourceAliases

KMS キーに関連付けられたエイリアスに基づいて、エイリアスがリクエストで使用されていない場合でも、KMS キーへのアクセスを許可または拒否します。kms:ResourceAliases 条件キーを使用すると、 などのエイリアスまたはエイリアスパターンを指定できるためalias/test*、IAM ポリシーで使用して、同じリージョン内の複数の KMS キーへのアクセスを制御できます。これは、KMS キーを使用する AWS KMS オペレーションに有効です。

例えば、次の IAM ポリシーでは、プリンシパルが 2 つの AWS アカウント で KMS キーの自動キーローテーションを管理できるようにします。ただし、アクセス許可は restricted で始まるエイリアスに関連付けられた KMS キーにのみ適用されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:EnableKeyRotation",
        "kms:DisableKeyRotation",
        "kms:GetKeyRotationStatus"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/*",
        "arn:aws:kms:*:444455556666:key/*"
      ],
      "Condition": {
        "ForAnyValue:StringLike": {
          "kms:ResourceAliases": "alias/restricted*"
        }
      }
```

```
    }
  ]
}
```

kms:ResourceAliases 条件はリクエストではなく、リソースの条件です。したがって、エイリアスを指定しないリクエストは、引き続き条件を満たすことができます。

次のリクエスト例は一致するエイリアスを指定し、条件を満たしています。

```
$ aws kms enable-key-rotation --key-id "alias/restricted-project"
```

ただし、次のリクエスト例では、指定された KMS キーに restricted で始まるエイリアスがあれば、そのエイリアスがリクエストで使用されなくても、条件を満たします。

```
$ aws kms enable-key-rotation --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

AWS CloudTrail ログでのエイリアスの検索

エイリアスを使用して、AWS KMS API オペレーションで AWS KMS key を表すことができます。これを行うと、KMS キーのエイリアスとキー ARN がイベントの AWS CloudTrail ログエントリに記録されます。エイリアスが requestParameters フィールドに表示されます。キー ARN が resources フィールドに表示されます。これは、AWS サービスがアカウントで AWS マネージドキー を使用する場合にも当てはまります。

例えば、次のGenerateDataKeyリクエストではproject-key、 エイリアスを使用して KMS キーを表します。

```
$ aws kms generate-data-key --key-id alias/project-key --key-spec AES_256
```

このリクエストが CloudTrail ログに記録されると、ログエントリには、実際に使用された KMS キーのエイリアスとキー ARN の両方が含まれます。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDE",
        "arn": "arn:aws:iam::111122223333:role/ProjectDev",
        "accountId": "111122223333",
        "accessKeyId": "FFHIJ",
```

```
        "userName": "example-dev"
    },
    "eventTime": "2020-06-29T23:36:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.205.123.000",
    "userAgent": "aws-cli/1.18.89 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.12",
    "requestParameters": {
        "keyId": "alias/project-key",
        "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "d93f57f5-d4c5-4bab-8139-5a1f7824a363",
    "eventID": "d63001e2-dbc6-4aae-90cb-e5370aca7125",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

CloudTrail ログでの AWS KMSオペレーションのログ記録の詳細については、「」を参照してくださいAWS KMS による AWS CloudTrail API コールのログ記録。

キーの表示

AWS Management Console または AWS Key Management Service (AWS KMS) API を使用して、ユーザーが管理する KMS キー、および AWS によって管理される KMS キーを含む AWS KMS keys を表示します。

トピック

- コンソールで KMS キーを表示する
- API で KMS キーを表示する

- KMS キーの暗号化設定の表示
- キー ID とキー ARN を検索する
- エイリアス名とエイリアス ARN を見つける

コンソールで KMS キーを表示する

AWS Management Console では、アカウントとリージョンの KMS キーのリストと、各 KMS キーの詳細を表示できます。

> **Note**
>
> AWS KMS コンソールには、アカウントとリージョンで表示する許可 がある KMS キーが表示されます。他の AWS アカウント の KMS キーは、表示、管理、および使用する許可がある場合でも、コンソールには表示されません。他のアカウントの KMS キーを表示するには、 DescribeKeyオペレーションを使用します。

トピック

- キーテーブルへの移動
- キーの詳細へ移動する
- KMS キーをソートおよびフィルタリングする
- KMS キーの詳細を表示する
- KMS キーテーブルをカスタマイズする

キーテーブルへの移動

各アカウントとリージョンの AWS KMS keys がテーブルに表示されます。作成する KMS キーと、AWS サービスによって作成される KMS キー には別々のテーブルがあります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージド

キー] を選択します。KMS キーの各種タイプの詳細については、AWS KMS keys を参照してください。

> **Tip**
>
> エイリアスのない AWS マネージドキー を表示するには、[Customer managed keys] (カスタマーマネージドキー) ページを使用します。
> AWS KMS コンソールには、アカウントとリージョンのカスタムキーストアも表示されます。カスタムキーストアで作成した KMS キーは、[Customer managed keys] (カスタマーマネージドキー) ページに表示されます。カスタムキーストアの詳細については、「カスタムキーストア」を参照してください。

キーの詳細へ移動する

アカウントおよびリージョンのすべての AWS KMS key には詳細ページがあります 詳細ページには KMS キーの [General configuration] (一般設定) セクションが表示され、ユーザーに [Cryptographic configuration] (暗号化の設定) およびキーの [Key policy] (キーポリシー) の表示と管理を認可するタブが含まれています。キーの種類によっては、詳細ページに [Aliases] (エイリアス)、[Key material] (キーマテリアル)、[Key rotation] (キーローテーション)、[Public key] (パブリックキー)、[Regionality] (リージョナリティー)、[Tags] (タグ) タブが含まれる場合もあります。

KMS キーのキーの詳細ページに移動するには

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。KMS キーの各種タイプの詳細については、AWS KMS key を参照してください。
4. キーの詳細ページを開くには、キーテーブルで KMS キーのキー ID またはエイリアスを選択します。

   KMS キーに複数のエイリアスがある場合、エイリアスの概要 (+n 個追加) が、エイリアスの 1 つの名前の横に表示されます。エイリアスのサマリーを選択すると、キーの詳細ページの Aliases (エイリアス) タブを直接表示します。

KMS キーをソートおよびフィルタリングする

コンソールで KMS キーを検索しやすくするために、キーテーブルをソートしてフィルタリングできます。

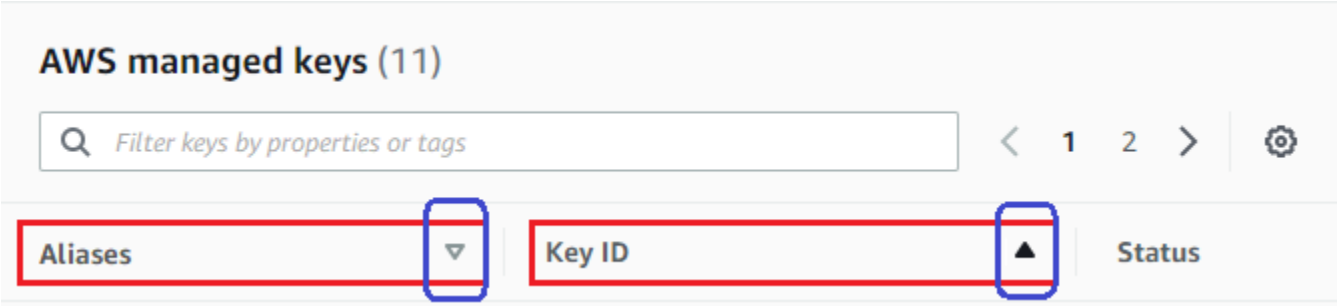
並べ替え

KMS キーを列の値で昇順または降順にソートできます。この機能は、テーブル内の KMS キーが現在のテーブルページに表示されていない場合も、すべての KMS キーをソートします。

ソート可能な列は、列名の横の矢印で示されます。AWS マネージドキー ページでは、[Aliases] (エイリアス) または [Key ID] (キー ID) でソートできます。[Customer managed keys] (カスタマーマネージドキー) ページでは、[Aliases] (エイリアス)、[Key ID] (キー ID)、または [Key type] (キータイプ) でソートできます。

昇順で並べ替えるには、矢印が上を向くように列見出しを選択します。降順で並べ替えるには、矢印が下を向くように列見出しを選択します。一度に 1 つの列のみでソートすることができます。

例えば、デフォルトのエイリアスではなく、キー ID により昇順で KMS キーをソートできます。



[Customer managed keys] (カスタマーマネージドキー) ページの KMS キーを [Key type] (キータイプ) で昇順にソートすると、すべての非対称キーがすべての対称キーの前に表示されます。

フィルター

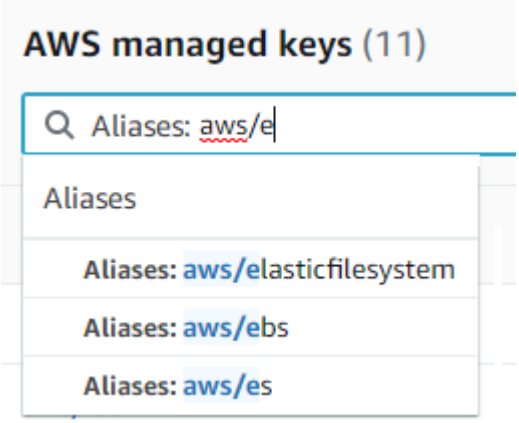
KMS キーは、プロパティ値またはタグでフィルタリングできます。フィルターは、現在のテーブルページに表示されていない場合でも、テーブル内のすべての KMS キーに適用されます。フィルターでは、大文字と小文字は区別されません。

フィルタリング可能なプロパティがフィルターボックスに一覧表示されます。AWS マネージドキー ページでは、エイリアスおよびキー IDでフィルタリングできます。[Customer managed keys] (カスタマーマネージドキー) ページでは、エイリアス、キー ID、キータイププロパティ、タグでフィルタリングできます。

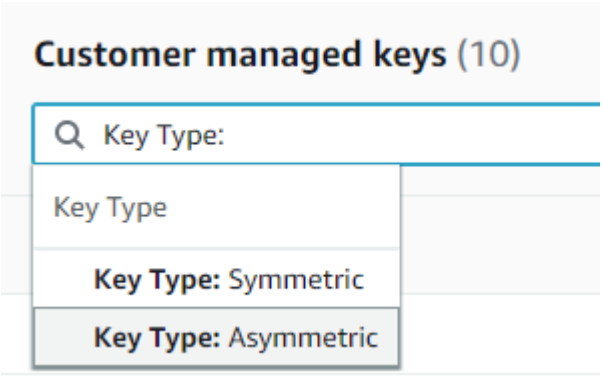
- AWS マネージドキー ページでは、エイリアスおよびキー IDでフィルタリングできます。
- [Customer managed keys] (カスタマーマネージドキー) ページでは、タグ、エイリアス、キー ID、キータイプ、リージョナリティのプロパティでフィルタリングできます。

プロパティ値でフィルタリングするには、フィルター、プロパティ名の順に選択し、実際のプロパティ値のリストから選択します。タグでフィルタリングするには、タグキーを選択し、実際のタグ値のリストから選択します。プロパティまたはタグキーを選択した後、プロパティ値またはタグ値のすべてまたは一部を入力することもできます。選択を行う前に、結果のプレビューが表示されます。

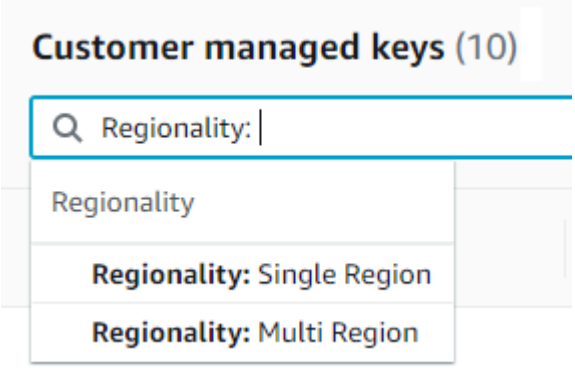
例えば、aws/e を含むエイリアス名で KMS キーを表示するには、フィルターボックス、[Alias (エイリアス)]、タイプ aws/e の順に選択し、Enter または Return を押してフィルターを追加します。



[Customer managed keys] (カスタマーマネージドキー) ページに非対称 KMS キーのみを表示するには、フィルターボックスをクリックして、[Key type] (キータイプ)、[Key type: Asymmetric] (キータイプ: 非対称) の順に選択します。非対称オプションは、テーブルに非対称 KMS キーがある場合にのみ表示されます。非対称 KMS キーの識別の詳細については、非対称 KMS キーの識別を参照してください。



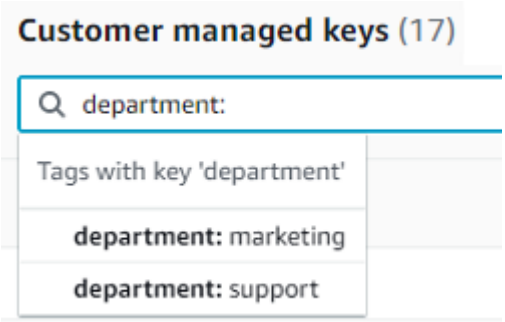
マルチリージョンキーのみを表示するには、[Customer managed keys] (カスタマーマネージドキー) ページで、フィルターボックス、[Regionality] (リージョナリティー)、[Regionality: Multi-Region] (リージョナリティー: マルチリージョン) の順に選択します。マルチリージョンキーオプションは、テーブルにマルチリージョンキーがある場合にのみ表示されます。マルチリージョンキーの識別の詳細については、マルチリージョンキーを表示する を参照してください。



タグのフィルタリングは少し異なります。特定のタグを持つ KMS キーのみを表示するには、フィルターボックス、タグキーの順に選択し、実際のタグ値の中から選択します。タグ値のすべてまたは一部を入力することもできます。

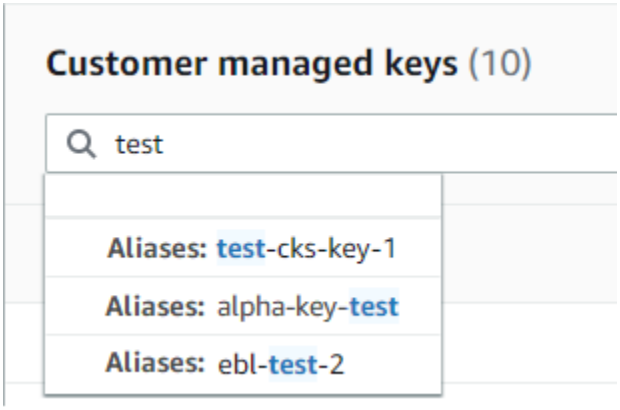
結果のテーブルには、選択したタグを持つすべての KMS キーが表示されます。ただし、タグは表示されません。タグを表示するには、KMS キーのキー ID またはエイリアスを選択し、その詳細ページで [Tags] (タグ) タブを選択します。これらのタブは、一般設定セクションにあります。

このフィルターには、タグキーとタグ値の両方が必要です。タグキーのみを入力しても、その値のみを入力しても KMS キーは検索されません。タグキーまたは値のすべてまたは一部でタグをフィルタリングするには、 ListResourceTagsオペレーションを使用してタグ付き KMS キーを取得し、プログラミング言語のフィルタリング機能を使用します。例については「ListResourceTags: KMS キーのタグを取得する」を参照してください。

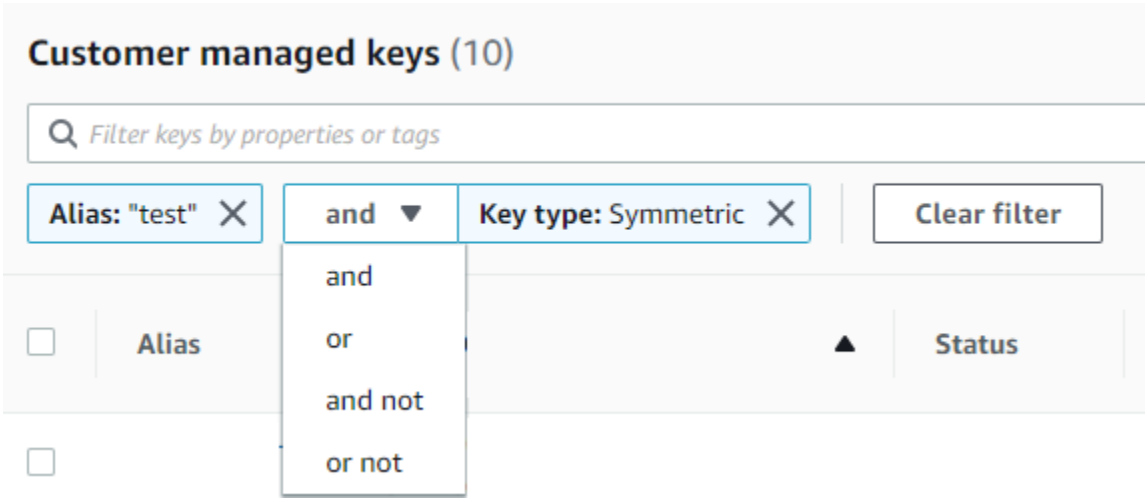


テキストを検索するには、フィルターボックスに、エイリアス、キー ID、キータイプ、タグキーのすべてまたは一部を入力します。(タグキーの選択後、タグ値を検索できます)。選択を行う前に、結果のプレビューが表示されます。

例えば、KMS キーをタグキーまたはフィルタリング可能なプロパティの `test` で表示するには、フィルターボックスの `test` を入力します。プレビューには、フィルターが選択する KMS キーが表示されます。この場合、`test` はエイリアスのプロパティにのみ表示されます。



複数のフィルターを同時に使用できます。フィルターを追加する場合は、論理演算子を選択することもできます。

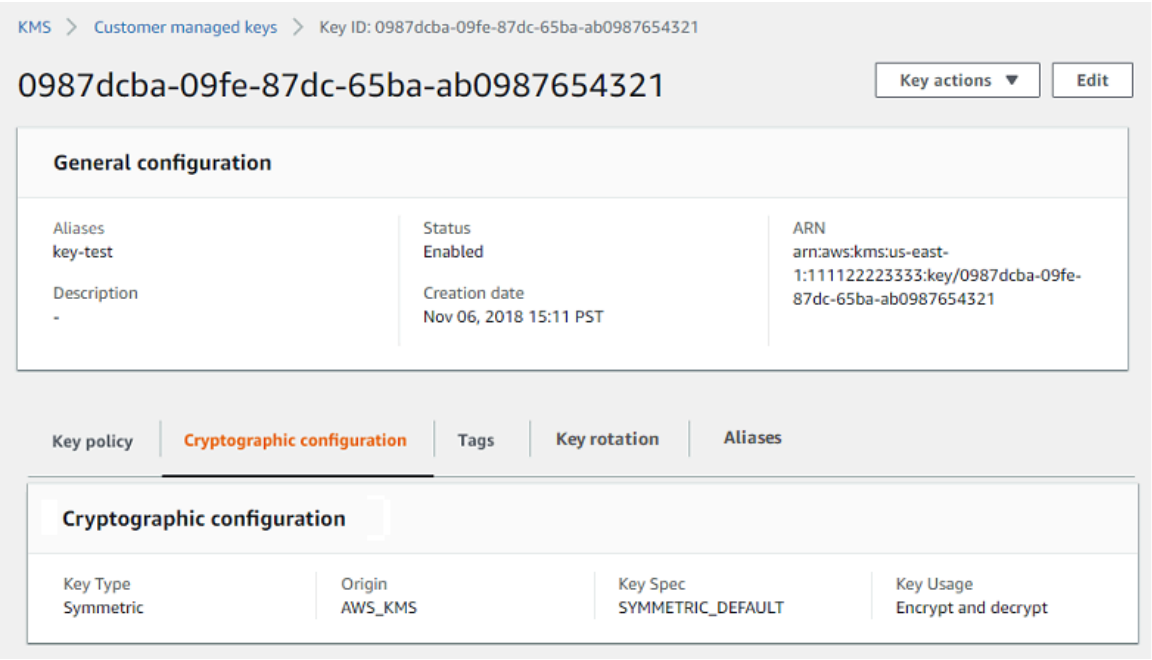


KMS キーの詳細を表示する

各 KMS キーの詳細ページには、KMS キーのプロパティが表示されます。KMS キーの種類によって若干異なります。

KMS キーに関する詳細情報を表示するには、AWS マネージドキー または [Customer managed keys] (カスタマーマネージドキー) ページで、KMS キーのエイリアスまたはキー ID を選択します。

KMS キーの詳細ページには、KMS キーのベーシックプロパティを表示する [General configuration] (一般設定) セクションがあります。これには、[Key policy] (キーポリシー)、[Cryptographic configuration] (暗号化設定)、[Tags] (タグ)、[Key material] (キーマテリアル) (インポートされたキーマテリアルを持つ KMS キーの場合)、[Key rotation] (キーローテーション) (対称暗号化 KMS キーの場合)、[Regionality] (リージョナリティ) (マルチリージョンキーの場合)、および [Public key] (パブリックキー) (非対称 KMS キーの場合) などの、KMS キーのプロパティを表示して編集できるタブもあります。



次のリストでは、タブ内のフィールドを含め、詳細表示のフィールドについて説明します。これらのフィールドの一部は、テーブル表示の列としても使用できます。

エイリアス

場所: [Aliases] (エイリアス) タブ

KMS キーのわかりやすい名前。エイリアスを使用して、コンソールおよび一部の AWS KMS API で KMS キーを識別できます。詳細については、「エイリアスの使用」を参照してください。

エイリアスタブは、AWS アカウント およびリージョンで KMS キーに関連付けられたすべてのエイリアスを表示します。

ARN

場所: [General configuration] (一般設定) セクション

KMS キーの Amazon リソースネーム (ARN)。この値は KMS キーを一意に識別します。この値を使用して、AWS KMS API オペレーションで KMS キーを識別できます。

接続状態

カスタムキーストアがバッキングキーストアに接続しているかどうかを示します。このフィールドは、KMS キーがカスタムキーストアで作成された場合にのみ表示されます。

このフィールドの値の詳細については、 AWS KMS API リファレンスのConnectionState「」を参照してください。

作成日

場所: [General configuration] (一般設定) セクション

KMS キーが作成された日時。この値は、デバイスの現地時間で表示されます。タイムゾーンはリージョンに依存しません。

有効期限切れとは異なり、作成時は KMS キーのみを参照し、キーマテリアルは参照しません。

CloudHSM クラスター ID

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーのキーマテリアルを含む AWS CloudHSM クラスターのクラスター ID。このフィールドは、KMS キーがカスタムキーストアで作成された場合にのみ表示されます。

CloudHSM クラスター ID を選択すると、AWS CloudHSM コンソールでクラスターページが開きます。

カスタムキーストア ID

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーを含むカスタムキーストアの ID。このフィールドは、KMS キーがカスタムキーストアで作成された場合にのみ表示されます。

カスタムキーストア ID を選択すると、AWS KMS コンソールでカスタムキーストアページが開きます。

カスタムキーストア名

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーを含むカスタムキーストアの名前。このフィールドは、KMS キーがカスタムキーストアで作成された場合にのみ表示されます。

カスタムキーストアのタイプ

場所: [Cryptographic configuration] (暗号化設定) タブ

カスタムキーストアが AWS CloudHSM キーストアと外部キーストアのいずれであるのかを示します。このフィールドは、KMS キーがカスタムキーストアで作成された場合にのみ表示されます。

説明

場所: [General configuration] (一般設定) セクション

書き込みおよび編集できる KMS キーの簡単な説明 (オプション)。カスタマーマネージドキーの説明を追加または更新するには、上記の一般設定で編集を選択します。

暗号化アルゴリズム

場所: [Cryptographic configuration] (暗号化設定) タブ

AWS KMS で KMS キーとともに使用できる暗号化アルゴリズムを一覧表示します。このフィールドは、[Key type] (キーのタイプ) が [Asymmetric] (非対称) で、[Key usage] (キーの用途) が [Encrypt and decrypt] (暗号化と復号) の場合にのみ表示されます。AWS KMS がサポートする暗号化アルゴリズムについては、「SYMMETRIC_DEFAULT キー仕様」および「暗号化および復号の RSA キー仕様」を参照してください。

有効期限日

場所:[キーマテリアル] タブ

KMS キーのキーマテリアルの有効期限が切れる日時。このフィールドは、インポートされたキーマテリアルを持つ KMS キーに対してのみ表示されます。つまり、[Origin] (オリジン) が [External] (外部) で、KMS キーに有効期限付きキーマテリアルがある場合です。

外部キー ID

場所: [Cryptographic configuration] (暗号化設定) タブ

外部キーストアの KMS キーに関連付けられている外部キーのID。このフィールドは、外部キーストアの KMS キーにのみ表示されます。

外部キーのステータス

場所: [Cryptographic configuration] (暗号化設定) タブ

外部キーストアプロキシが KMS キーに関連付けられた外部キーについて報告した最新のステータス。このフィールドは、外部キーストアの KMS キーにのみ表示されます。

外部キーの使用

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーに関連付けられた外部キーで有効になっている、暗号化のオペレーション。このフィールドは、外部キーストアの KMS キーにのみ表示されます。

キーポリシー

場所: [Key policy] (キーポリシー) タブ

IAM ポリシーおよびグラントとともに KMS キーへのアクセスを制御します。KMS キーそれぞれに 1 つのキーポリシーがあります。これは、唯一の必須認証要素です。カスタマーマネージドキーのキーポリシーを変更するには、キーポリシータブで編集を選択します。詳細については、「the section called “キーポリシー”」を参照してください。

キーローテーション

場所: [Key rotation] (キーローテーション) タブ

カスタマーマネージド KMS キーのキーマテリアルの自動ローテーションを有効化または無効化します。カスタマーマネージドキーのキーローテーションステータスを変更するには、[Key rotation] (キーローテーション) タブのチェックボックスを使用します。

AWS マネージドキーのキーマテリアルのローテーションを有効または無効にすることはできません。AWS マネージドキーは毎年自動的にローテーションされます。

キー仕様

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーのキーマテリアルのタイプ。AWS KMS は、対称暗号化 KMS キー (SYMMETRIC_DEFAULT)、異なる長さの HMAC KMS キー、異なる長さの RSA KMS キー、異なる曲線を持つ楕円曲線キーをサポートします。詳細については、「キー仕様」を参照してください。

キーのタイプ

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーが対称か非対称かを示します。

キーの用途

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーを [Encrypt and decrypt] (暗号化および復号)、[Sign and verify] (署名および検証)、または [Generate and verify MAC] (MAC の生成と検証) のどれに使用できるかを示します。詳細については、「キーの用途」を参照してください。

オリジン

場所: [Cryptographic configuration] (暗号化設定) タブ

KMS キーのキーマテリアルのソース。有効な値は次のとおりです。

- AWS KMS が生成するキーマテリアル用の AWS KMS
- AWS CloudHSM キーストアの KMS キー用の AWS CloudHSM
- インポートされたキーマテリアルの外部 (BYOK)
- 外部キーストアの KMS キー用の外部キーストア

MAC アルゴリズム

場所: [Cryptographic configuration] (暗号化設定) タブ

AWS KMS で HMAC KMS キーと使用できる MAC アルゴリズムのリストです。このフィールドは、[Key spec] (キーの仕様) が HMAC キー仕様 (HMAC_*) である場合にのみ表示されます。AWS KMS がサポートする MAC アルゴリズムについては、「HMAC KMS キーの主な仕様」を参照してください。

プライマリキー

場所: [リージョナリティ] タブ

この KMS キーがマルチリージョンのプライマリキーであることを示します。認可されたユーザーはこのセクションを使用して、別の関連するマルチリージョンキーにプライマリキーを変更できます。このフィールドは、KMS キーがマルチリージョンのプライマリキーである場合にのみ表示されます。

パブリックキー

場所: [Public key] (パブリックキー) タブ

非対称 KMS キーのパブリックキーを表示します。承認されたユーザーは、このタブを使用してパブリックキーをコピーおよびダウンロードできます。

リージョナリティー

場所: 一般設定セクションおよびリージョナリティータブ

KMS キーが単一リージョンキー、マルチリージョンのプライマリキー、またはマルチリージョンのレプリカキーのいずれであるかを示します。このフィールドは、KMS キーがマルチリージョンキーである場合にのみ表示されます。

関連するマルチリージョンキー

場所: [リージョナリティ] タブ

関連するすべてのマルチリージョンのプライマリキーとレプリカキー (現在の KMS キーを除く) を表示します。このフィールドは、KMS キーがマルチリージョンキーである場合にのみ表示されます。

プライマリキーの [Related multi-Region keys] (関連するマルチリージョンキー) セクションでは、承認されたユーザーが新しいレプリカキーを作成できます。

レプリカキー

場所: [リージョナリティ] タブ

この KMS キーがマルチリージョンのレプリカキーであることを示します。このフィールドは、KMS キーがマルチリージョンのレプリカキーである場合にのみ表示されます。

署名アルゴリズム

場所: [Cryptographic configuration] (暗号化設定) タブ

AWS KMS で KMS キーとともに使用できる署名アルゴリズムを一覧表示します。このフィールドは、[Key type] (キーのタイプ) が [Asymmetric] (非対称) で、[Key usage] (キーの用途) が [Sign and verify] (署名と検証) の場合にのみ表示されます。AWS KMS がサポートする署名アルゴリズムについては、「署名および検証用の RSA キー仕様」および「楕円曲線のキー仕様」を参照してください。

ステータス

場所: [General configuration] (一般設定) セクション

KMS キーのキーステータス。KMS キーはステータスが有効の場合にのみ、暗号化オペレーションで使用できます。各 KMS キーステータスと KMS キーで実行されるオペレーションへの影響の詳細については、AWS KMS キーのキーステータス を参照してください。

タグ

場所: [Tags] (タグ) タブ

KMS キーを記述するオプションのキーバリューペア。KMS キーのタグを追加または変更するには、[Tags] (タグ) タブで [Edit] (編集) を選択します。

AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

KMS キーテーブルをカスタマイズする

必要に応じて、AWS マネージドキー、および AWS Management Console の [Customer managed keys] (カスタマーマネージドキー) ページに表示されるテーブルをカスタマイズできます。テーブルの列、各ページ (ページサイズ) の AWS KMS keys の数、テキストの折り返しを選択できます。選択した設定は、確認すると保存され、ページを開くたびに再適用されます。

KMS キーテーブルをカスタマイズするには

1. AWS マネージドキー または [Customer managed keys] (カスタマーマネージドキー) ページで、ページの右上隅にある設定アイコン ( ) を選択します。
2. [Preferences] (設定) ページで、必要な設定を選択してから [Confirm] (確認) を選択します。

ページサイズ設定を使用して、特に、スクロールしやすいデバイスを通常使用する場合、各ページに表示される KMS キーの数を増やすことを検討します。

表示されるデータ列は、テーブル、ジョブロール、アカウントとリージョンでの KMS キーのタイプによって異なる場合があります。次の表に、推奨される設定を示します。列の説明については、「KMS キーの詳細を表示する」を参照してください。

推奨される KMS キーテーブルの設定

KMS キーテーブルに表示される列をカスタマイズして、KMS キーに関する必要な情報を表示できます。

AWS マネージドキー

デフォルトでは、AWS マネージドキー テーブルにエイリアス、キー ID、ステータスの各列が表示されます。これらの列は、ほとんどのユースケースに最適です。

対称暗号化 KMS キー

AWS KMS が生成したキーマテリアルを持つ対称暗号化 KMS キーのみを使用する場合は、[Aliases] (エイリアス)、[Key ID] (キー ID)、[Status] (ステータス)、[Creation date] (作成日) の各列が最も役に立つと思われます。

非対称 KMS キー

非対称 KMS キーを使用する場合は、[Aliases] (エイリアス)、[Key ID] (キー ID)、[Status] (ステータス) 列に加えて、[Key type] (キータイプ)、[Key spec] (キー仕様)、[Key usage] (キーの使用法) 列を追加することを検討します。これらの列には、KMS キーが対称か非対称か、キーマテリアルのタイプ、KMS キーを暗号化または署名に使用できるかが表示されます。

HMAC KMS キー

HMAC KMS キーを使用する場合は、[Aliases] (エイリアス)、[Key ID] (キー ID)、[Status] (ステータス) 列に加えて、[Key type] (キーのタイプ)、[Key spec] (キーの仕様)、[Key usage] (キーの使用) 列を追加することを検討します。これらの列は、KMS キーが HMAC キーであるかどうかを示します。KMS キーをキーの仕様やキーの用途でソートすることはできないため、エイリアスとタグを使用して HMAC キーを識別してから、AWS KMS コンソールのフィルタリング機能を使用してエイリアスまたはタグによるフィルタリングを実行します。

インポートされたキーマテリアル

インポートされたキーマテリアルを持つ KMS キーの場合、[Origin] (オリジン) および[Expiration date] (有効期限日) 列を追加することを検討します これらの列には、KMS キーのキーマテリアルが AWS KMS によってインポートまたは生成されたかどうか、およびキーマテリアルの有効期限が切れるタイミングが表示されます。[Creation date] (作成日)フィールドには、KMS キーが (キーマテリアルなしで) 作成された日付が表示されます。キーマテリアルの特性を反映していません。

カスタムキーストアのキー

KMS キーがカスタムキーストアにある場合、[Origin] (オリジン) 列と [Custom key store ID] (カスタムキーストア ID) 列を追加することを検討します。これらの列は、KMS キーがカスタムキーストアにあることを示し、カスタムキーストアのタイプを表し、カスタムキーストアを識別します。

マルチリージョンキー

マルチリージョンキーがある場合、リージョナリティー列を追加することを検討します。これは、KMS キーが単一リージョンキー、マルチリージョンのプライマリキー、マルチリージョンのレプリカキーのいずれであるかを示します。

API で KMS キーを表示する

AWS Key Management Service (AWS KMS) API を使用して、KMS キーを表示できます。このセクションでは、既存の KMS キーに関する詳細を返すいくつかのオペレーションを示します。例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用できます。

トピック

- ListKeys: すべての KMS キーの ID と ARN を取得する
- DescribeKey: KMS キーに関する詳細情報を取得する
- GetKeyPolicy: KMS キーにアタッチされたキーポリシーを取得する
- ListAliases: KMS キーのエイリアス名と ARNs を取得する
- ListResourceTags: KMS キーのタグを取得する

ListKeys: すべての KMS キーの ID と ARN を取得する

ListKeys オペレーションは、アカウントとリージョン内のすべての KMS キーの ID と Amazon リソースネーム (ARN) を返します。

例えば、`ListKeys` オペレーションに対するこの呼び出しでは、架空のアカウントの各 KMS キーの ID と ARN を返します。複数のプログラミング言語の例については、「KMS キーのキー ID とキー ARN を取得する」を参照してください。

```
$ aws kms list-keys

{
  "Keys": [
    {
      "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
      "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
    },
    {
```

```
      "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
      "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    }
}
```

DescribeKey: KMS キーに関する詳細情報を取得する

DescribeKey オペレーションは、指定された KMS キーに関する詳細を返します。KMS キーを識別するには、そのキー ID、キー ARN、エイリアス名、エイリアス ARN を使用します。

発信者のアカウントとリージョンに KMS キーのみを表示する ListKeysオペレーションとは異なり、認可されたユーザーは DescribeKeyオペレーションを使用して、他のアカウントの KMS キーに関する詳細を取得できます。

> **Note**
>
> DescribeKey レスポンスは、同じ値を持つ KeySpec および CustomerMasterKeySpec メンバーの両方を含みます。CustomerMasterKeySpec メンバーは非推奨です。

例えば、DescribeKey に対するこの呼び出しは、対称暗号化 KMS キーに関する情報を返します。レスポンスのフィールドは、AWS KMS key の仕様、キーの状態、キーマテリアルのオリジンによって異なります。複数のプログラミング言語の例については、「AWS KMS key の表示」を参照してください。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1499988169.234,
        "MultiRegion": false,
```

```
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

この例では、署名と検証に使用される非対称 KMS キーで DescribeKey オペレーションを呼び出します。レスポンスには、この KMS キーに対して AWS KMS がサポートする署名アルゴリズムが含まれます。

```
$ aws kms describe-key --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
    "KeyMetadata": {
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "Origin": "AWS_KMS",
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "CreationDate": 1569973196.214,
        "Description": "",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "AWSAccountId": "111122223333",
        "Enabled": true,
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

GetKeyPolicy: KMS キーにアタッチされたキーポリシーを取得する

GetKeyPolicy オペレーションは、KMS キーにアタッチされているキーポリシーを取得します。KMS キーを識別するには、そのキー ID またはキー ARN を使用します。ポリシー名も指定する必要があり、これは、常に default になります。(出力の読み込みが困難な場合には、コマンドに --

output text オプションを追加します。) GetKeyPolicy は発信者のアカウントとリージョンの KMS キーでのみ機能します。

複数のプログラミング言語の例については、「キーポリシーの取得」を参照してください。

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default

{
  "Version" : "2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

ListAliases: KMS キーのエイリアス名と ARNs を取得する

ListAliases オペレーションは、アカウントとリージョンのエイリアスを返します。レスポンスの TargetKeyId は、エイリアスが参照する KMS キーのキー ID (存在する場合) を示します。

デフォルトでは、ListAliases コマンドはアカウントとリージョンのすべてのエイリアスを返します。これには、お客様が作成してカスタマーマネージドキーに関連付けたエイリアスと、AWS が作成してアカウントの AWS マネージドキー に関連付けたエイリアスが含まれます。AWS のエイリアスは aws/dynamodb のような aws/*<service-name>* 形式を使用するので認識できます。

このレスポンスには、TargetKeyId フィールドがないエイリアスも含まれます (この例の aws/redshift エイリアスなど)。これらは AWS が作成した定義済みのエイリアスですが、まだ KMS キーとは関連付けられていません。

複数のプログラミング言語の例については、「エイリアスのリスト化」を参照してください。

```
$ aws kms list-aliases

{
    "Aliases": [
```

```
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasName": "alias/financeKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        },
        {
            "AliasName": "alias/aws/redshift",
```

```
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/redshift"
        },
    ]
}
```

特定の KMS キーを参照するエイリアスを取得するには、KeyId パラメータを使用します。パラメータ値には、キー ID または キー ARN を指定できます。エイリアス名またはエイリアス ARN を指定することはできません。

次の例のコマンドは、カスタマーマネージドキーを参照するエイリアスを取得します。ただし、このようなコマンドを使用して、AWS マネージドキー を参照するエイリアスを検索することもできます。

```
$ aws kms list-aliases --key-id arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/financeKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "AliasName": "alias/financeKey",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
    ]
}
```

AWS マネージドキー のエイリアスのみを取得するには、プログラミング言語の機能を使用して、レスポンスをフィルタリングします。

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

ListResourceTags: KMS キーのタグを取得する

ListResourceTags オペレーションは、指定された KMS キーのタグを返します。API では 1 つの KMS キーのタグを返しますが、コマンドをループで実行すると、アカウントとリージョン内のすべての KMS キー、または選択した KMS キーのセットに対するタグを取得できます。 この API は一度に 1 ページずつ返されるため、多数の KMS キーに多数のタグがある場合は、プログラミング言語のページ割りを使用して、必要なタグをすべて取得する必要があります。

ListResourceTags オペレーションは、すべての KMS キーのタグを返しますが、AWS マネージドキー はタグ付けされません。発信者のアカウントとリージョンの KMS キーでのみ機能します。

KMS キーのタグを検索するには、ListResourceTags オペレーションを使用します。KeyId パラメータは必須です。このオペレーションは、キー ID またはキー ARN を受け入れます。この例を実行する前に、サンプルキー ARN を有効な ARN に置き換えます。

```
$  aws kms list-resource-tags --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
      {
    "Tags": [
        {
            "TagKey": "Department",
            "TagValue": "IT"
        },
        {
            "TagKey": "Purpose",
            "TagValue": "Test"
        }
    ],
    "Truncated": false
}
```

ListResourceTags オペレーションを使用して、特定のタグ、タグキー、タグ値を持つアカウントおよびリージョンで、すべての KMS キーを取得します。これを行うには、プログラミング言語のフィルタリング機能を使用します。

例えば、次の Bash スクリプトでは、 ListKeysおよび ListResourceTagsオペレーションを使用して、アカウントとリージョンのすべての KMS キーをProjectタグキーで取得します。これらのオペレーションは両方とも、結果の最初のページのみを取得します。多数の KMS キーまたは多数のタグがある場合は、言語のページ割り機能を使用して、各オペレーションの結果全体を取得します。この例を実行する前に、サンプルキー ID を有効な ID に置き換えます。

```
TARGET_TAG_KEY='Project'

for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text); do
  key_tags=$(aws kms list-resource-tags --key-id "$key" --query "Tags[?TagKey==\`
$TARGET_TAG_KEY\`]")
  if [ "$key_tags" != "[]" ]; then
    echo "Key: $key"
    echo "$key_tags"
  fi
done
```

出力は、次の出力例のようにフォーマットされます。

```
Key: 0987dcba-09fe-87dc-65ba-ab0987654321
[
    {
        "TagKey": "Project",
        "TagValue": "Gamma"
    }
]
Key: 1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
Key: 0987ab65-43cd-21ef-09ab-87654321cdef
[
    {
        "TagKey": "Project",
        "TagValue": "Alpha"
    }
]
```

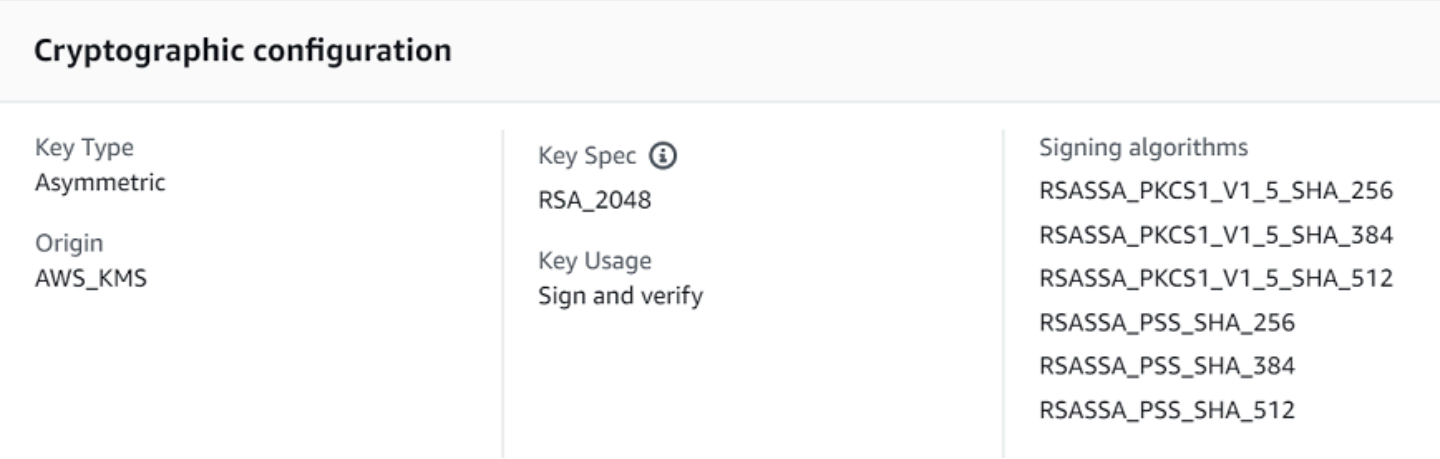
KMS キーの暗号化設定の表示

KMS キーの作成後、その暗号化設定を表示できます。KMS キーの作成後にその設定を変更することはできません。別の設定を使用したい場合は、KMS キーを削除して再度作成します。

AWS KMS コンソールで、または AWS KMS API を使用して、KMS キーの、キー仕様、キーの使用方法、サポートされている暗号化アルゴリズムまたは署名アルゴリズムを含む暗号化設定を確認することができます。詳細については、「非対称 KMS キーの識別」を参照してください。

AWS KMS コンソールでは、各 KMS キーの詳細ページに、KMS キーに関する暗号化の詳細を表示する暗号化設定タブが含まれます。例えば次のイメージは、署名と検証に使用される RSA KMS キーの暗号化設定タブを示しています。

特別な目的をもつ一部の KMS キーの [Cryptographic configuration] (暗号化設定) タブには、他にも専用のセクションがあります。例えば、カスタムキーストアの KMS キーの [Cryptographic configuration] (暗号化設定) タブには、[Custom key stores] (カスタムキーストア) セクションがあります。外部キーストア の KMS キーの [Cryptographic configuration] (暗号化設定) タブには、[External key] (外部キー) セクションがあります。



AWS KMS API では、 DescribeKeyオペレーションを使用します。レスポンスの KeyMetadata 構造には、KMS キーの暗号化設定が含まれます。例えば、DescribeKey は、署名と検証に使用される RSA KMS キーに対して次のレスポンスを返します。

```
{
  "KeyMetadata": {

    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": 1571767572.317,
    "CustomerMasterKeySpec": "RSA_2048",
    "Description": "",
    "Enabled": true,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
```

```
    "KeyState": "Enabled",
    "MultiRegion": false,
    "Origin": "AWS_KMS",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
        "RSASSA_PKCS1_V1_5_SHA_256",
        "RSASSA_PKCS1_V1_5_SHA_384",
        "RSASSA_PKCS1_V1_5_SHA_512",
        "RSASSA_PSS_SHA_256",
        "RSASSA_PSS_SHA_384",
        "RSASSA_PSS_SHA_512"
    ]
  }
}
```

キー ID とキー ARN を検索する

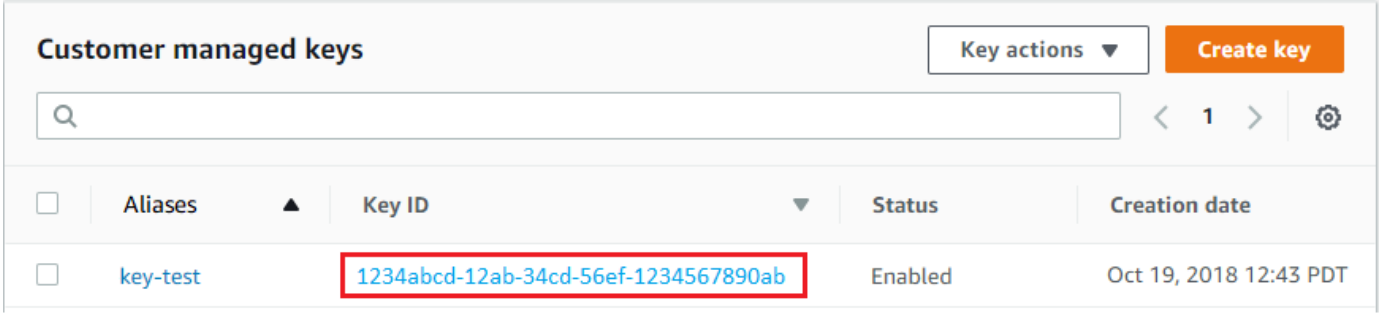
AWS KMS key を識別するには、キー ID または Amazon リソースネーム (キー ARN) を使用します。暗号化オペレーションでは、エイリアス名またはエイリアス ARN を使用することもできます。

AWS KMS がサポートする KMS キー識別子の詳細については、キー識別子 (KeyId) を参照してください。エイリアス名とエイリアス ARN を検索する方法については、「」を参照してください エイリアス名とエイリアス ARN を見つける。

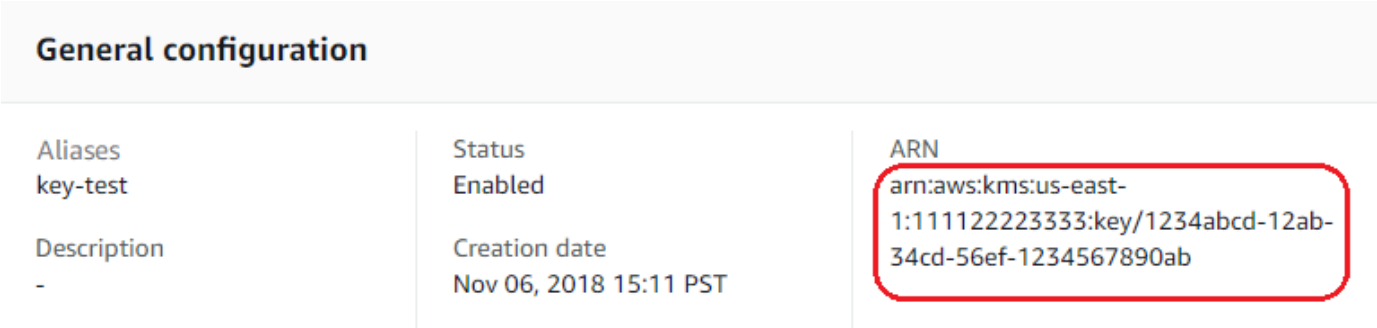
キー ID と ARN を検索するには (コンソール)

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。
4. KMS キーのキー ID を検索するには、KMS キーのエイリアスで始まる行を参照します。

   デフォルトでは、[キー ID] 列がテーブルに表示されます。[キー ID] 列がテーブルに表示されない場合は、「the section called “KMS キーテーブルをカスタマイズする”」で説明されている手順に従って復元します。KMS キーの詳細ページで KMS キーのキー ID を表示することもできます。



5. KMS キーの Amazon リソースネーム (ARN) を検索するには、キー ID またはエイリアスを選択します。キー ARN が [General configuration] セクションに表示されます。



キー ID とキー ARN (AWS KMS API) を検索するには

のキー ID とキー ARN を検索するにはAWS KMS key、 ListKeysオペレーションを使用します。複数のプログラミング言語の例については、「キー ID とキー ARN の取得」 および「キー ID とキー ARN の取得」を参照してください。

ListKeys レスポンスには、アカウントとリージョンの各 KMS キーのキー ID とキー ARN が含まれます。

```
$ aws kms list-keys
{
    "Keys": [
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
```

```
        }
    ]
}
```

エイリアス名とエイリアス ARN を見つける

エイリアスは、AWS KMS AWS KMS keys (KMS キー) のわかりやすい名前です。AWS KMS コンソールまたは AWS KMS API で、エイリアス名およびエイリアス ARN を検索できます。

AWS KMS がサポートする KMS キー識別子の詳細については、キー識別子 (KeyId) を参照してください。キー ID とキー ARN の検索については、「」を参照してください キー ID とキー ARN を検索する。

トピック

- エイリアス名とエイリアス ARN を検索するには (コンソール)
- エイリアス名とエイリアス ARN を検索するには (AWS KMSAPI)

エイリアス名とエイリアス ARN を検索するには (コンソール)

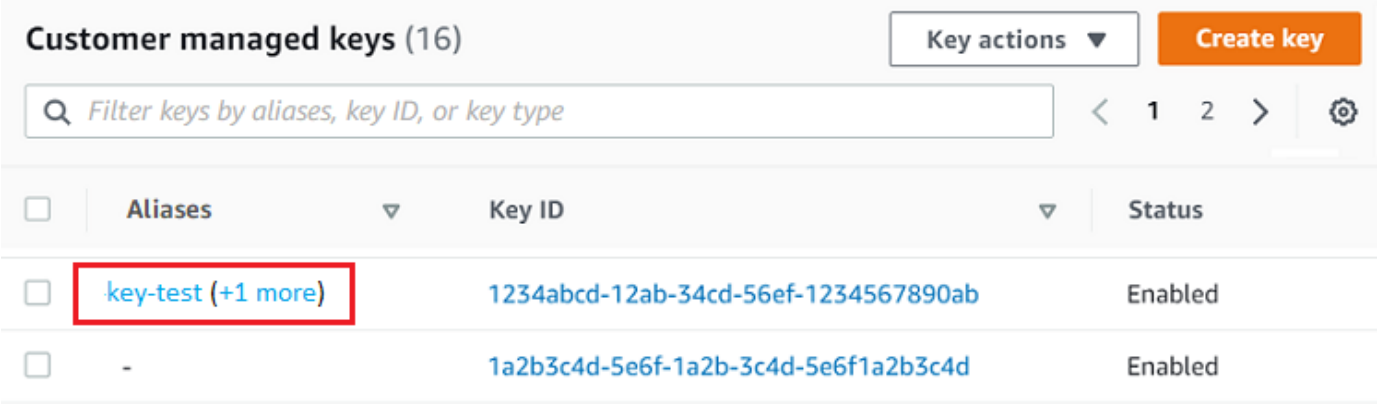
AWS KMS コンソールには、KMS キーに関連付けられたエイリアスが表示されます。

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。
4. エイリアス列には、各 KMS キーのエイリアスが表示されます。KMS キーにエイリアスがない場合は、エイリアス列にダッシュ (-) が表示されます。

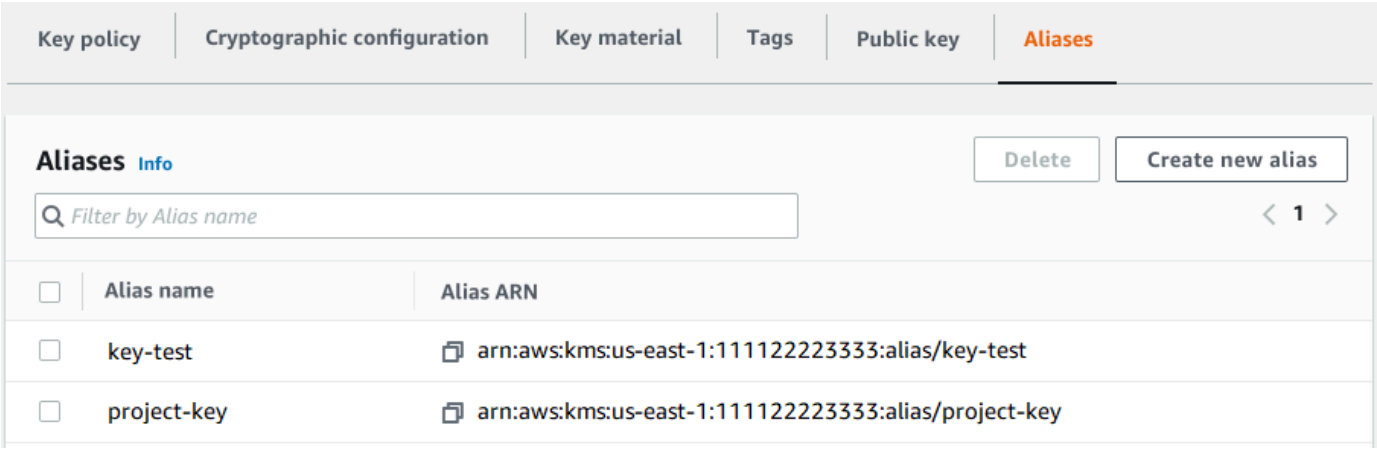
   KMS キーに複数のエイリアスがある場合は、エイリアス列に、(+n 個以上)などのエイリアスの概要も表示されます 。例えば、次の KMS キーには 2 つのエイリアスがあり、そのうちの 1 つは `key-test` です。

   KMS キーのすべてのエイリアスのエイリアス名とエイリアス ARN を検索するには、[Aliases] (エイリアス) タブを使用します。

- [Aliases] (エイリアス) タブに直接移動するには、[Aliases] (エイリアス) 列で、エイリアスの概要 (+n個以上) を選択します。エイリアスの概要は、KMS キーに複数のエイリアスがある場合にのみ表示されます。
- または、KMS キーのエイリアスまたはキー ID を選択し (KMS キーの詳細ページが開きます)、[Aliases] (エイリアス) タブを選択します。これらのタブは、[General configuration] (一般設定) セクションにあります。



5. [Aliases] (エイリアス) タブには、KMS キーのすべてのエイリアスのエイリアス名とエイリアス ARN が表示されます。このタブで、KMS キーのエイリアスを作成または削除することもできます。



エイリアス名とエイリアス ARN を検索するには (AWS KMSAPI)

のエイリアス名とエイリアス ARN を検索するにはAWS KMS key、 ListAliasesオペレーションを使用します。複数のプログラミング言語の例については、「エイリアスのリスト化」 および「エイリアス名と ARN の取得」を参照してください。

デフォルトでは、レスポンスにアカウントとリージョン内のすべてのエイリアスのエイリアス名とエイリアス ARN が示されます。特定の KMS キーのエイリアスのみを取得するには、KeyId パラメータを使用します。

例えば、次のコマンドは、キー ID 1234abcd-12ab-34cd-56ef-1234567890ab を持つサンプル KMS キーのエイリアスのみを取得します 。

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/key-test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/key-test",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        },
        {
            "AliasName": "alias/project-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
        }
    ]
}
```

キーの編集

AWS KMS コンソールおよび AWS KMS API で、カスタマーマネージドキーの次のプロパティを変更できます。

AWS マネージドキー または AWS 所有のキー のプロパティを編集することはできません。これらのキーはそれらを作成した AWS サービスによって管理されます。

説明

カスタマーマネージドキーの説明は、KMS キーの詳細ページで、または UpdateKeyDescriptionオペレーションを使用して変更できます。

コンソールでキーの説明を編集するには、KMS キーの詳細ページの右上隅にある [Edit] (編集) を選択します。

キーポリシー

キーポリシーは、カスタマーマネージドキーの詳細ページにあるキーポリシータブで、または PutKeyPolicyオペレーションを使用して変更できます。

詳細については、「キーポリシーの変更」を参照してください。

タグ

AWS KMS コンソールの [Customer managed keys] (カスタマーマネージドキー) ページで、またはカスタマーマネージドキーの詳細ページの [Tags] (タグ) タブで、タグを作成および削除できます または、 TagResourceおよび UntagResourceオペレーションを使用できます。

詳細については、「キーのタグ付け」を参照してください。

有効化と無効化

AWS KMS コンソールの [Customer managed keys] (カスタマーマネージドキー) ページで、またはカスタマーマネージドキーの詳細ページで、KMS キーを有効化、および無効化できます。または、 EnableKeyおよび DisableKeyオペレーションを使用できます。

詳細については、「キーの有効化と無効化」を参照してください。

自動キーローテーション

自動キーローテーションは、カスタマーマネージドキーの詳細ページのキーローテーションタブで、または EnableKeyRotationおよび DisableKeyRotationオペレーションを使用して有効または無効にできます。

詳細については、「AWS KMS keys ローテーション」を参照してください。

以下も参照してください。

エイリアスの更新

キーのタグ付け

AWS KMS では、KMS キーの作成時に、カスタマーマネージドキーにタグを追加して、削除保留中でない限り、既存の KMS キーにタグ付けまたはタグ解除することができます。他の AWS アカウント で、エイリアス、カスタムキーストア、AWS マネージドキー、AWS 所有のキー、KMS キーにタグ付けはできません。タグはオプションですが、非常に便利です。

詳細については、「キーの作成」および「キーの編集」を参照してください。ベストプラクティス、タグ付け戦略、タグの形式と構文など、タグに関する一般情報については、「Amazon Web Services 全般のリファレンス」の「AWS リソースのタグ付け」を参照してください。

トピック

AWS KMS のタグについて

タグ は、AWS リソースに割り当てる (または AWS が割り当てる) ことができるオプションのメタデータラベルです。各タグは、 タグキーとタグ値で構成され、どちらも大文字と小文字が区別される文字列です。タグ値には、空の (null) 文字列を指定できます。リソースのそれぞれのタグには異なるタグキーが必要ですが、同じタグを複数の AWS リソースに追加できます。各リソースには、最大 50 個のユーザーが作成したタグを含めることができます。

タグキーまたはタグ値には、機密情報や重要情報を含めないでください。タグには、請求など、多くの AWS のサービス からアクセスできます。

AWS KMS では、KMS キーの作成時に、カスタマーマネージドキーにタグを追加して、削除保留中でない限り、既存の KMS キーにタグ付けまたはタグ解除することができます。他の AWS アカウント で、エイリアス、カスタムキーストア、AWS マネージドキー、AWS 所有のキー、KMS キーにタグ付けはできません。タグはオプションですが、非常に便利です。

例えば、Alpha プロジェクトに使用するすべての KMS キーと Amazon S3 バケットに `"Project"="Alpha"` タグを追加できます。

```
TagKey   = "Project"
TagValue = "Alpha"
```

形式や構文など、タグに関する一般情報については、「Amazon Web Services 全般のリファレンス」の「AWS リソースのタグ付け」を参照してください。

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 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、KMS キーおよび Amazon Elastic Block Store (Amazon EBS) ボリュームまたは AWS Secrets Manager シークレットに同じタグを割り当てることができます。タグを使用して、オートメーションのために KMS キーを識別することもできます。
- AWS のコストを追跡します。AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。この機能を使用して、プロジェクト、アプリケーション、またはコストセンターの AWS KMS コストを追跡できます。

  タグを使用したコスト配分の詳細については、AWS Billing ユーザーガイドのコスト配分タグの使用を参照してください。タグキーとタグ値に適用されるルールの詳細については、「AWS Billing ユーザーガイド」の「ユーザー定義タグの制限」を参照してください。
- AWS リソースへのアクセスを制御します。タグに基づく KMS キーへのアクセス許可および拒否は、AWS KMS がサポートする属性ベースのアクセスコントロール (ABAC) の一部です。タグに基づく AWS KMS keys へのアクセス制御の詳細については、タグを使用してKMS キーへのアクセスを制御する を参照してください。AWS リソースへのアクセスを制御するタグ使用の詳細については、IAM ユーザーガイドのリソースタグを使用した AWS リソースへのアクセス制御を参照してください。

AWS KMS は、、TagResource、または ListResourceTagsオペレーションを使用するとUntagResource、AWS CloudTrailログにエントリを書き込みます。

コンソールで KMS キータグを管理する

AWS KMS コンソールで KMS キーを作成するときに、KMS キーにタグを追加できます。コンソールの [Tags] (タグ) タブを使用して、カスタマーマネージドキーのタグを追加、編集、削除することもできます。KMS キーのタグを追加、編集、表示、削除するにはアクセス許可が必要です。詳細については、「タグへのアクセスを制御する」を参照してください。

KMS キーの作成中にタグを追加する

コンソールで KMS キーを作成中にタグを追加するには、KMS キーを作成してコンソールで表示するために必要なアクセス許可に加えて、IAM ポリシーで `kms:TagResource` アクセス許可が必要です。少なくとも、アクセス許可はアカウントとリージョン内のすべての KMS キーを対象にする必要があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。

2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。(AWS マネージドキー のタグを管理することはできません)
4. キータイプを選択し、[Next] (次へ) を選択します。
5. エイリアスおよび説明 (オプション) を入力します。
6. タグキーおよびタグ値 (オプション) を入力します。タグを追加するには、[Add tag] (タグの追加) を選択します。タグを削除するには、[Remove] (削除) を選択します。新しい KMS キーのタグ付けが完了したら、[Next] (次へ) を選択します。
7. KMS キーの作成を完了します。

既存の KMS キーでタグを表示および管理する

コンソールでタグを追加、表示、編集、削除するには、KMS キーでタグ付け許可が必要です。この許可は、KMS キーのキーポリシーから取得できます。または、キーポリシーで許可されている場合、KMS キーを含む IAM ポリシーから取得できます。コンソールで KMS キーを表示するための許可に加えて、これらの許可が必要です。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。(AWS マネージドキー のタグを管理することはできません)
4. テーブルフィルターを使用して、特定のタグを持つ KMS キーのみを表示できます。詳細については、「KMS キーをソートおよびフィルタリングする」を参照してください。
5. KMS キーのエイリアスの横にあるチェックボックスをオンにします。
6. [Key actions]、[Add or edit tags] の順に選択します。
7. KMS キーの詳細ページで、[Tags] (タグ) タブを選択します。

   - 最初のタグを作成するには、[Create tag] (タグの作成) を選択し、タグキー (必須)とタグ値 (オプション) を入力して、[Save] (保存) を選択します。

     タグ値を空白のままにすると、実際のタグ値は NULL または空の文字列になります。
   - タグを追加するには、[Edit] (編集)、[Add tag] (タグの追加) の順に選択し、タグキーとタグ値を入力して、[Save] (保存) を選択します。

- タグの名前または値を変更するには、[Edit (編集)] を選択して変更を加えた後、[Save (保存)] を選択します。
- タグを削除するには、[Edit (編集)] を選択します。そのタグの行で、[Remove (削除)] を選択してから [Save (保存)] を選択します。

8. 変更を保存するには、変更の保存を選択します。

API オペレーションで KMS キータグを管理する

AWS Key Management Service (AWS KMS) API を使用して、管理する KMS キーのタグを追加、削除、一覧表示できます。以下の例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。AWS マネージドキー のタグ付けはできません。

KMS キーのタグを追加、編集、表示、削除するには、アクセス許可が必要です。詳細については、「タグへのアクセスを制御する」を参照してください。

トピック

- CreateKey: 新しい KMS キーにタグを追加する
- TagResource: KMS キーのタグを追加または変更する
- ListResourceTags: KMS キーのタグを取得する
- UntagResource: KMS キーからタグを削除する

CreateKey: 新しい KMS キーにタグを追加する

カスタマーマネージドキーを作成するときにタグを追加できます。タグを指定するには、CreateKeyオペレーションの Tagsパラメータを使用します。

KMS キーの作成時にタグを追加するには、発信者が IAM ポリシーで `kms:TagResource` アクセス許可を取得する必要があります。少なくとも、アクセス許可はアカウントとリージョン内のすべての KMS キーを対象にする必要があります。詳細については、「タグへのアクセスを制御する」を参照してください。

`Tags` パラメータ値の `CreateKey` は、大文字と小文字を区別するタグキーとタグ値のペアのコレクションです。KMS キーのそれぞれのタグは、異なるタグ名を持つ必要があります。タグ値は、NULL または空の文字列にすることができます。

例えば、以下の AWS CLI コマンドは、Project:Alpha タグを持つ対称暗号化 KMS キーを作成します。複数のキーと値のペアを指定する場合は、スペースを使用して各ペアを区切ります。

```
$ aws kms create-key --tags TagKey=Project,TagValue=Alpha
```

このコマンドが成功すると、新しい KMS キーに関する情報を含む KeyMetadata オブジェクトが返されます。ただし、KeyMetadata にはタグは含まれません。タグを取得するには、ListResourceTagsオペレーションを使用します。

TagResource: KMS キーのタグを追加または変更する

TagResource オペレーションは、KMS キーに 1 つ以上のタグを追加します。このオペレーションを使用して、別の AWS アカウント のタグを追加または編集することはできません。

タグを追加するには、新しいタグキーとタグ値を指定します。タグを編集するには、既存のタグキーと新しいタグ値を指定します。KMS キーのそれぞれのタグは、異なるタグキーを持つ必要があります。タグ値は、NULL または空の文字列にすることができます。

例えば、次のコマンドでは、サンプルの KMS キーに **Purpose** タグおよび **Department** タグを追加します。

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Pretest TagKey=Department,TagValue=Finance
```

このコマンドが成功した場合、出力を返しません。KMS キーのタグを表示するには、ListResourceTagsオペレーションを使用します。

TagResource を使用して、既存のタグのタグ値を変更することもできます。タグ値を置き換えるには、同じタグキーを異なる値に指定します。

例えば、このコマンドは Purpose タグの値 Pretest をからに変更します Test。

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Test
```

ListResourceTags: KMS キーのタグを取得する

ListResourceTags オペレーションは、KMS キーのタグを取得します。KeyId パラメータは必須です。このオペレーションを使用して、別の AWS アカウント の KMS キーのタグを表示することはできません。

例えば、次のコマンドでは、サンプルの KMS キーのタグを取得します。

```
$ aws kms list-resource-tags --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

  "Truncated": false,
  "Tags": [
      {
        "TagKey": "Project",
        "TagValue": "Alpha"
     },
     {
       "TagKey": "Purpose",
       "TagValue": "Test"
     },
     {
       "TagKey": "Department",
       "TagValue": "Finance"
     }
  ]
}
```

UntagResource: KMS キーからタグを削除する

UntagResource オペレーションは、KMS キーからタグを削除します。削除するタグを識別するには、タグキーを指定します。このオペレーションを使用して、別の AWS アカウント の KMS キーからタグを削除することはできません。

成功すると、 UntagResource オペレーションは出力を返しません。また、指定したタグキーが KMS キーで見つからない場合、例外をスローしたり、レスポンスを返したりすることもありません。オペレーションが機能したことを確認するには、 ListResourceTagsオペレーションを使用します。

例えば、このコマンドでは、指定した KMS キーから **Purpose** タグとその値を削除します。

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys
 Purpose
```

タグへのアクセスを制御する

タグを追加、表示、削除するには、AWS KMS コンソールまたは API を使用して、プリンシパルがアクセス許可にタグ付けする必要があります。これらのアクセス許可はキーポリシーで付与できます。または、IAM ポリシー (VPC エンドポイントポリシーを含む) で付与することもできますが、キーポリシーが許可する場合に限られます。AWSKeyManagementServicePowerUser 管理ポリシーは、アカウントがアクセスできるすべての KMS キーのタグ、タグ解除、およびタグの一覧表示をプリンシパルに許可します。

タグの AWS グローバル条件キーを使用して、これらのアクセス許可を制限することもできます。ではAWS KMS、これらの条件により、 TagResourceや などのタグ付けオペレーションへのアクセスを制御できますUntagResource。

> **Note**
>
> タグとエイリアスを管理する許可をプリンシパルに付与する場合は注意が必要です。タグまたはエイリアスを変更すると、カスタマーマネージドキーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「タグを使用してKMS キーへのアクセスを制御する」を参照してください。

サンプルポリシーおよび詳細については、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。

タグを作成および管理するためのアクセス許可は、次のように機能します。

kms:TagResource

プリンシパルにタグの追加または編集を許可します。KMS キーの作成中にタグを追加するには、プリンシパルが特定の KMS キーに制限されない IAM ポリシーでアクセス許可を持っている必要があります。

kms:ListResourceTags

プリンシパルが KMS キーのタグを表示できるようにします。

kms:UntagResource

プリンシパルが KMS キーからタグを削除できるようにします。

ポリシーのタグ付け許可

キーポリシーまたは IAM ポリシーでタグ付け許可を付与できます。例えば、次のキーポリシーの例では、選択したユーザーに KMS キーに対するタグ付け許可が付与されます。これにより、サンプルの管理者ロールまたはデベロッパーロールを引き受けることができるすべてのユーザーにタグを表示する許可が付与されます。

```
{
  "Version": "2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow all tagging permissions",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/LeadAdmin",
        "arn:aws:iam::111122223333:user/SupportLead"
      ]},
      "Action": [
          "kms:TagResource",
          "kms:ListResourceTags",
          "kms:UntagResource"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Allow roles to view tags",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:role/Administrator",
        "arn:aws:iam::111122223333:role/Developer"
      ]},
      "Action": "kms:ListResourceTags",
      "Resource": "*"
    }
  ]
```

```
}
```

プリンシパルに複数の KMS キーに対するタグ付け許可を付与するには、IAM ポリシーを使用します。このポリシーを有効にするには、各 KMS キーのキーポリシーで、アカウントが IAM ポリシーを使用して KMS キーへのアクセスを制御することを許可する必要があります。

例えば、次の IAM ポリシーではプリンシパルが KMS キーを作成することを許可します。指定したアカウントのすべての KMS キーでタグを作成および管理することもできます。この組み合わせにより、プリンシパルは CreateKeyオペレーションの Tags パラメータを使用して、作成中に KMS キーにタグを追加できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ListResourceTags"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

タグ付け許可を制限する

ポリシー条件を使用して、タグ付け許可を制限できます。次のポリシー条件を `kms:TagResource` および `kms:UntagResource` 許可に適用できます。例えば、`aws:RequestTag/`*`tag-key`* 条件を使用して、プリンシパルが特定のタグのみを追加できるようにするか、プリンシパルが特定のタグキーを持つタグを追加しないように許可できます。または、`kms:KeyOrigin` 条件を使用して、プリンシパルがインポートされたキーマテリアルを持つ KMS キーにタグ付けまたはタグ解除を行わないようにすることができます。

- aws:RequestTag
- aws:ResourceTag/tag-key (IAM ポリシーのみ)
- aws:TagKeys
- kms:CallerAccount
- kms:KeySpec
- kms:KeyUsage
- kms:KeyOrigin
- kms:ViaService

ベストプラクティスとして、タグを使用して KMS キーへのアクセスを制御する場合は、`aws:RequestTag/`*`tag-key`* または `aws:TagKeys` 条件キーを使用して、許可するタグ (またはタグキー) を決定します。

例えば、次の IAM ポリシーは前述のものと似ています。ただしこのポリシーでは、プリンシパルはタグ (`TagResource`) の作成とタグ `UntagResource` の削除を、`Project` タグキーを持つタグに対してのみ実行できます。

`TagResource` および `UntagResource`リクエストには複数のタグを含めることができるため、aws:TagKeys 条件で `ForAllValues`または `ForAnyValue`集合演算子を指定する必要があります。`ForAnyValue` 演算子では、リクエスト内のタグキー 1 つ以上が、ポリシーのタグキーの 1 つと一致する必要があります。`ForAllValues` 演算子では、リクエスト内のタグキーすべてが、ポリシーのタグキーの 1 つと一致する必要があります。`ForAllValues` 演算子は`true`、リクエストにタグがない場合は も返しますが、タグが指定されていない場合は TagResource と UntagResource が失敗します。集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
```

```
      "Action": "kms:ListResourceTags",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
          "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

タグを使用してKMS キーへのアクセスを制御する

KMS キーのタグに基づいて、AWS KMS keys へのアクセスを制御できます。例えば、プリンシパルが特定のタグを持つ KMS キーのみを有効または無効にすることを許可する IAM ポリシーを書き込むことができます。または、IAM ポリシーを使用して、KMS キーに特定のタグがない限り、プリンシパルが暗号化オペレーションで KMS キーを使用しないようにすることもできます。

この機能は、属性ベースのアクセスコントロール (ABAC) の AWS KMS サポートの一部です。AWS リソースへのアクセスを制御するタグの使用に関する情報は、IAM ユーザーガイドの What is ABAC for AWS? および Controlling Access to AWS Resources Using Resource Tags を参照してください。ABAC に関連するアクセス問題の解決方法については、AWS KMS の ABAC トラブルシューティング を参照してください。

> **Note**
>
> タグとエイリアスの変更が KMS キーの認可に影響を及ぼすまでに最長 5 分かかることがあります。最近の変更は、認可に影響を与える前に API オペレーションで表示される場合があります。

AWS KMS は、aws:ResourceTag/tag-key グローバル条件コンテキストキー をサポートします。これにより、KMS キーのタグに基づいて KMS キーへのアクセスを制御できます。複数の KMS キーに同じタグを付けることができるため、この機能を使用すると、KMS キーの選択したセットにアクセ

ス許可を適用できます。それらのタグを変更することで、セット内の KMS キーを簡単に変更することもできます。

AWS KMS では、`aws:ResourceTag/tag-key` 条件キーは IAM ポリシーでのみサポートされます。これは、1 つの KMS キーにのみ適用されるキーポリシー、または ListKeysや オペレーションなど、特定の KMS キーを使用しないListAliasesオペレーションではサポートされていません。

タグを使用してアクセスを制御すると、シンプルでスケーラブルかつ柔軟な方法でアクセス許可を管理できます。ただし、適切に設計および管理されていない場合、KMS キーへのアクセスを誤って許可または拒否する可能性があります。タグを使用してアクセスを制御する場合は、次の方法を検討します。

- タグを使用して、最小特権アクセスのベストプラクティスを強化します。IAM プリンシパルで、使用または管理する必要がある KMS キーのみに対して、必要なアクセス許可のみを付与します。例えば、タグを使用して、プロジェクトに使用する KMS キーにラベルを付けます。次に、プロジェクトタグで KMS キーのみを使用する許可をプロジェクトチームに付与します。
- プリンシパルにタグを追加、編集、削除できる `kms:TagResource` および `kms:UntagResource` 許可を付与する際は注意してください。タグを使用して KMS キーへのアクセスを制御する場合、タグを変更すると、使用許可のない KMS キーに対する使用許可をプリンシパルに付与してしまう可能性があります。他のプリンシパルがジョブを実行するために必要な KMS キーへのアクセスを拒否することもできます。キーポリシーを変更したり、権限を作成したりする許可を持たないキー管理者も、タグを管理する許可があれば、KMS キーへのアクセスを制御できます。

  可能な限り、ポリシー条件 (`aws:RequestTag/tag-key` または `aws:TagKeys` ~プリンシパルのタグ付け許可の制限など) を、特定の KMS キーの特定のタグまたはタグパターンに適用します。
- 現在、タグ付けおよびタグ解除の許可を持つプリンシパルを AWS アカウント で確認し、必要に応じて調整します。例えば、コンソールのキー管理者のデフォルトキーポリシーは、KMS キーの `kms:TagResource` および `kms:UntagResource` アクセス許可を含んでいます。IAM ポリシーでは、すべての KMS キーに対してタグ付けおよびタグ解除を許可する場合があります。例えば、AWSKeyManagementServicePowerUser管理ポリシーでは、プリンシパルがすべての KMS キーのタグのタグ付け、タグ解除、およびタグの一覧表示を行うことを許可します。
- タグに依存するポリシーを設定する前に、AWS アカウント で KMS キーのタグを確認します。含めるタグにのみポリシーが適用されることを確認します。CloudTrail ログとCloudWatch アラームを使用して、KMS キーへのアクセスに影響を与える可能性のある変更にタグを付けるように警告します。

- タグベースのポリシー条件では、パターンマッチングを使用します。タグの特定のインスタンスには関連付けられません。タグベースの条件キーを使用するポリシーは、パターンに一致するすべての新規および既存のタグに影響します。ポリシー条件に一致するタグを削除して再作成すると、古いタグの場合と同様に、新しいタグに条件が適用されます。

例えば、次の IAM ポリシーの例を考えてみます。これにより、プリンシパルは、アジアパシフィック (シンガポール) リージョンで "Project"="Alpha" タグを持つアカウントの KMS キーに対してのみ および GenerateDataKeyWithoutPlaintext Decrypt オペレーションを呼び出すことができます。このポリシーは、サンプルの Alpha プロジェクトでロールにアタッチできます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

次の IAM ポリシーの例では、プリンシパルが、特定の暗号化オペレーションのために、アカウントで任意の KMS キーを使用することを許可します。ただし、プリンシパルが "Type"="Reserved" タグのある、または "Type" タグのない KMS キーにこれらの暗号化オペレーションを使用することを禁止します。

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

```
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/Type": "true"
        }
      }
    }
  ]
}
```

キーの有効化と無効化

カスタマーマネージドキーの有効化と無効化を行うことができます。KMS キー を作成すると、そのキーはデフォルトで有効になります。KMSキーを無効にすると、再度有効にするまで暗号化オペレーションで使用できなくなります。

一時的で簡単に元に戻せるため、KMS キーを無効にすることは、破壊的で元に戻せないアクションである KMS キーを削除することの安全な代替手段です。KMS キーを削除することを検討している場合は、まず無効にし、暗号化されたデータを復号するためにキーを使用する必要がないことを確認するCloudWatch アラームまたは同様のメカニズムを設定します。

KMS キーを無効にするとそのキーはただちに使用できなくなります (結果整合性の影響を受ける)。ただし、KMS キーで保護されたデータキーを使って暗号化されたリソースは、KMS キーがデータキーの復号化などに再び使用されるまでは、影響を受けません。この問題は AWS のサービス に影響します。その多くが、リソースを保護するためにデータキーを使用しています。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

AWS マネージドキー または AWS 所有のキー を有効化および無効化することはできません。AWS マネージドキー は、AWS KMS を使用するサービスでの使用が永続的に有効になっています。AWS 所有のキー は、それらを所有するサービスによってのみ管理されます。

> **Note**
>
> AWS KMS は無効になっているカスタマーマネージドキーのキーマテリアルを回転しません。詳細については、「キーの自動回転の仕組み」を参照してください。

トピック

- KMS キーの有効化と無効化 (コンソール)
- KMS キーの有効化と無効化 (AWS KMS API)

KMS キーの有効化と無効化 (コンソール)

AWS KMS コンソールを使用して、カスタマーマネージドキーの有効化と無効化を行うことができます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 有効または無効にする KMS キーのチェックボックスをオンにします。
5. KMS キーを有効にするには、[Key actions (キーアクション)]、[Enable (有効)] の順に選択します。KMS キーを無効にするには、[Key actions (キーアクション)]、[Disable (無効)] の順に選択します。

KMS キーの有効化と無効化 (AWS KMS API)

EnableKey オペレーションは、無効な を有効にしますAWS KMS key。以下の例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。`key-id` パラメータは必須です。

このオペレーションはどのような出力も返しません。キーのステータスを表示するには、DescribeKeyオペレーションを使用します。

```
$ aws kms enable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

DisableKey オペレーションは、有効な KMS キーを無効にします。`key-id` パラメータは必須です。

```
$ aws kms disable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

このオペレーションはどのような出力も返しません。キーのステータスを確認するには、DescribeKeyオペレーションを使用して、 `Enabled`フィールドを確認します。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "MultiRegion": false,
        "Enabled": false,
        "KeyState": "Disabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
```

```
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333"
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

AWS KMS keys ローテーション

カスタマーマネージドキーの新しい暗号化マテリアルを作成するには、新しい KMS キーを作成し、アプリケーションまたはエイリアスを変更して新しい KMS キーを使用します。または、既存の KMS キーの自動キーローテーションを有効にすることができます。

KMS キーの自動キーローテーションを有効にすると、AWS KMS は KMS キーの新しい暗号化マテリアルを毎年生成します。AWS KMS に古い暗号化マテリアルはすべて永続的に保存されるため、KMS キーで暗号化されたすべてのデータを復号することができます。AWS KMS は KMS キーが削除されるまで、ローテーションされたキーマテリアルを削除しません。Amazon CloudWatch および で KMS キーのキーマテリアルのローテーションを追跡できますAWS CloudTrail。

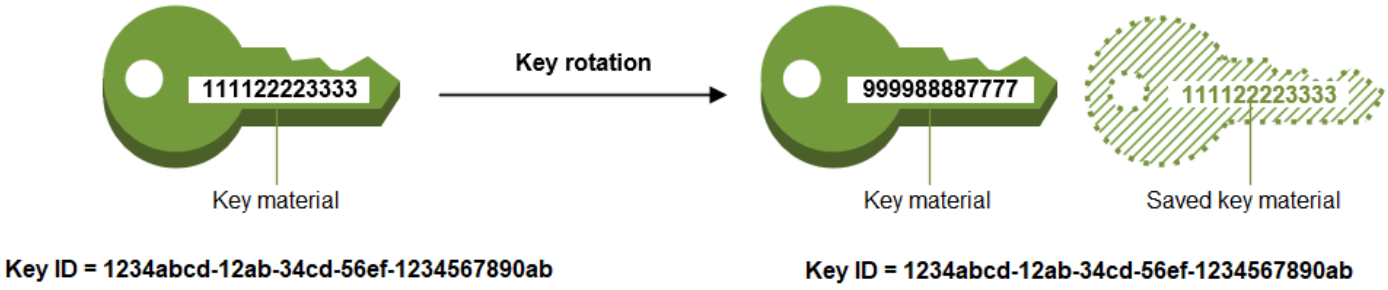
データの暗号化にローテーションされた KMS キーを使用する場合、AWS KMS は現在のキーマテリアルを使用します。暗号文の復号にローテーションされた KMS キーを使用する場合、AWS KMS はそのテキストの暗号化に使用されたキーマテリアルのバージョンを使用します。キーマテリアルの特定のバージョンをリクエストすることはできません。AWS KMS は適切なキーマテリアルを使用して復号を透過的に実行するため、ローテーションされた KMS キーは、コードを変更することなくアプリケーションおよび AWS のサービス で安全に使用することができます。

ただし、自動キーローテーションは、KMS キーで保護されるデータには効果がありません。自動ローテーションは KMS キーが生成したデータキーのローテーションや KMS キーで保護されたデータの再暗号化を実行せず、漏洩したデータキーの影響を軽減しません。

AWS KMS が自動キーローテーションをサポートするのは、AWS KMS が作成するキーマテリアルを持つ対称暗号化 KMS キーのみです。カスタマーマネージド KMS キーについては、自動ローテーションはオプションになります。AWS KMS は常に、AWS マネージド KMS キーのキーマテリアルのローテーションを毎年実行します。AWS 所有の KMS キーのローテーションはそれぞれ異なります。

> **Note**
>
> AWS マネージドキー のローテーション間隔は 2022 年 5 月に変更されました。詳細については、「AWS マネージドキー」を参照してください。

キーローテーションは、暗号化操作で使用される暗号化シークレットであるキーマテリアルのみを変更します。KMS キーは、そのキーマテリアルの変更の有無や回数に関わらず、同じ論理リソースのままです。次のイメージに示されているように、KMS キーのプロパティは変更されません。



自動キーローテーションには次の利点があります。

- キー ID、キー ARN、リージョン、ポリシー、アクセス許可などの KMS キーのプロパティは、キーがローテーションされても変更されません。
- KMS キーのキー ID またはキー ARN を参照するアプリケーションまたはエイリアスを変更する必要はありません。
- キーマテリアルのローテーションは、どの AWS のサービスでの KMS キーの使用にも影響しません。
- キーローテーションを有効にすると、AWS KMS によって KMS キーが毎年自動的にローテーションされます。更新を覚えている、またはスケジュールする必要はありません。

新しい KMS キーを作成し、元の KMS キーの代わりに使用することを決定することができます。これには、既存の KMS キーでキーマテリアルをローテーションするのと同じ効果があり、多くの場合、手動キーローテーションを使用します。キーローテーションのスケジュールを制御する場合は、手動ローテーションすることをお勧めします。これは、非対称 KMS キー、HMAC KMS キー、カスタムキーストア内の KMS キー、およびインポートされたキーマテリアルを持つ KMS キーなど、自動キーローテーションの対象にならない KMS キーをローテーションする手段も提供します。

キーローテーションと料金

AWS KMS では、KMS キー用に維持されるキーマテリアルの各バージョンに対して月額料金が発生します。詳細については、「AWS Key Management Service の料金表」を参照してください。

> **Note**
>
> AWS Cost Explorer Service を使用して、キーストレージ料金の内訳を表示できます。例えば、[使用タイプ] に `$REGION-KMS-Keys` を指定し、データを [API オペレーション] でグループ化することによって、ビューをフィルターし、現在のローテションされた KMS キーとして課金されたキーの合計料金を表示できます。
> 過去の期間のレガシー Unknown API オペレーションのインスタンスが引き続き表示される場合があります。

キーローテーションとクォータ

各 KMS キーは、キーリソースのクォータを計算するときに、ローテーションされたキーマテリアルのバージョン数に関係なく、1 つのキーとしてカウントされます。

キーマテリアルとローテーションの詳細については、AWS Key Management Service 暗号化の詳細を参照してください。

トピック

- KMS キーをローテーションする理由
- キーの自動ローテーションの仕組み
- 「自動キーローテーションを有効または無効にする方法」
- 手動でのキーローテーション

KMS キーをローテーションする理由

暗号化のベストプラクティスでは、AWS KMS が生成するデータキーなどのデータを直接暗号化するキーの広範な再利用を推奨していません。256 ビットデータキーが数百万のメッセージを暗号化すると、キーが枯渇し、微小なパターンを含む暗号文を生成し始める可能性があり、ずる賢い人物が悪用して、キーの中のビットが発見される可能性があります。このキーの枯渇を回避するには、データキーを 1 回または数回だけ使用することをお勧めします。こうすることによって、キーマテリアルが効果的にローテーションされます。

ただし、KMS キーは、ラップキー (キー暗号化キーとも呼ばれます) として最もよく使用されます。データを暗号化する代わりに、ラップキーはデータを暗号化するデータキーを暗号化します。そのた

め、これらはデータキーよりもはるかに使用頻度が低く、キーが枯渇するほど多数回再利用されることはほとんどありません。

このように枯渇するリスクは極めて低いものの、ビジネスルールや契約、政府の規制により、KMS キーのローテーションが必要になる場合があります。KMS キーをローテーションせざるを得ない場合は、自動キーローテーション (これがサポートされている場合) または手動キーローテーション (自動キーローテーションがサポートされていない場合) を使用することをお勧めします。

キーの自動ローテーションの仕組み

AWS KMS のキーローテーションは、透過的で使いやすいように設計されています。AWS KMS は、カスタマーマネージドキーに対してのみ、オプションの自動キーローテーションをサポートします。

キーマテリアルの管理

AWS KMS は、キーローテーションが無効になっている場合でも、KMS キーのすべてのキーマテリアルを保持します。AWS KMS は KMS キーが削除されたときにのみ、キーマテリアルを削除します。

キーマテリアルの使用

データの暗号化にローテーションされた KMS キーを使用する場合、AWS KMS は現在のキーマテリアルを使用します。暗号文の復号にローテーションされた KMS キーを使用する場合、AWS KMS は暗号化に使用したものと同じバージョンのキーマテリアルを使用します。キーマテリアルの特定のバージョンをリクエストすることはできません。

ローテーション日

AWS KMS は、ローテーションが有効になってから 1 年 (約 365 日) 後にキーマテリアルをローテーションし、その後は毎年 (約 365 日間隔で) ローテーションします。

カスタマーマネージドキー

自動キーローテーションは、カスタマーマネージドキーではオプションであり、いつでも有効化および無効化できるため、ローテーション日は、ローテーションの最終有効化日によって異なります。その日付は、キーの有効期間にわたって多数回変更できます。

たとえば、2022 年 1 月 1 日にカスタマーマネージドキーを作成し、2022 年 3 月 15 日に自動キーローテーションを有効にすると、AWS KMS は、2023 年 3 月 15 日、2024 年 3 月 15 日、その後は 365 日ごとにキーマテリアルをローテーションします。

以下は特殊なケースです。

- キーローテーションの無効化 — 任意の時点で自動キーローテーションを無効にすると、KMS キーは、ローテーションが無効になったときに使用していたバージョンのキーマテリアルを使用し続けます。自動キーローテーションを再び有効にすると、AWS KMS は、新たにローテーションを有効にした日から 1 年後にキーマテリアルをローテーションして、その後は毎年 (約 365 日間隔で) ローテーションします。
- 無効にされた KMS キー - KMS キーが無効になっている間、AWS KMS はこれをローテーションしません。ただし、キーローテーションのステータスは変更されず、KMS キーが無効の間は変更することができません。KMS キーが再有効化され、そのキーマテリアルが 1 年以上前のものである場合、AWS KMS はそれを直ちにローテーションしてから、その後毎年ローテーションします。キーマテリアルが 1 年未満のものである場合、AWS KMS は元のキーローテーションのスケジュールを再開します。
- 削除保留中の KMS キー - KMS キーが削除保留中の場合、AWS KMS はローテーションを実行しません。キーローテーションのステータスは `false` に設定されています。削除が保留中の場合は変更することができません。削除をキャンセルすると、以前のキーローテーションのステータスが元に戻ります。キーマテリアルが 1 年以上前のものである場合、AWS KMS はそれを直ちにローテーションしてから、その後毎年 (最後のローテーションから約 365 日間隔で) ローテーションします。キーマテリアルが 1 年未満のものである場合、AWS KMS は元のキーローテーションのスケジュールを再開します。

AWS マネージドキー

AWS KMS は、AWS マネージドキーの自動ローテーションを毎年実行します (約 365 日間隔)。AWS マネージドキーのキーローテーションを有効化または無効化することはできません。

AWS マネージドキー のキーマテリアルは、作成日から 1 年後に最初にローテーションされ、その後は毎年 (最後のローテーションから約 365 日間隔で) ローテーションされます。

> **Note**
>
> 2022 年 5 月、AWS KMS は AWS マネージドキー のローテーションスケジュールを 3 年 (約 1,095 日間隔) ごとから毎年 (約 365 日間隔) に変更しました。
> 新しい AWS マネージドキーは、作成日から 1 年後に自動的にローテーションされ、それ以降はほぼ 1 年ごとにローテーションされます。
> 既存の AWS マネージドキーは、直近のローテーションから 1 年後にローテーションされ、その後毎年ローテーションされます。

AWS 所有のキー

AWS 所有のキーのキーローテーションを有効化または無効化することはできません。AWS 所有のキー のキーローテーション方式は、キーを作成および管理する AWS サービスによって決定されます。詳細については、サービスのユーザーガイドまたはデベロッパーガイドの「保管時の暗号化 」トピックを参照してください。

サポートされる KMS キータイプ

自動キーローテーションがサポートされるのは、AWS KMS が生成するキーマテリアルを持つ対称暗号化 KMS キー (オリジン = AWS_KMS) のみです。

自動キーローテーションは、次のタイプの KMS キーではサポートされませんが、これらの KMS キーは手動でローテーションできます。

- 非対称 KMS キー
- HMAC KMS キー
- カスタムキーストアの KMS キー
- インポートされたキーマテリアルを持つ KMS キー

マルチリージョンキー

自動キーローテーションをマルチリージョンキーで有効および無効にできます。プロパティはプライマリキーにのみ設定します。AWS KMS がキーを同期すると、プライマリキーからレプリカキーにプロパティ設定がコピーされます。プライマリキーのキーマテリアルをローテーションさせると、AWS KMS はそのキーマテリアルをすべてのレプリカキーに自動的にコピーします。詳細については、「マルチリージョンキーをローテーションする」を参照してください。

AWS サービス

AWS サービスのサーバー側の暗号化に使用するカスタマーマネージドキーで、自動キーローテーションを有効にできます。年間ローテーションは透過的で、AWS のサービスと互換性があります。

キーローテーションのモニタリング

が AWS マネージドキーまたはカスタマーマネージドキー のキーマテリアルAWS KMSを自動的にローテーションすると、 `KMS CMK Rotation`イベントが Amazon EventBridge に、RotateKey イベントが AWS CloudTrailログに書き込まれます。 ???これらのレコードを使用して、KMS キーがローテーションされたことを確認できます。

結果整合性

自動キーローテーションは、他の AWS KMS 管理オペレーションと同じ結果整合性の影響下にあります。新しいキーマテリアルが AWS KMS 全体で使用可能になるまで、若干の遅延が生じることがあります。ただし、キーマテリアルのローテーションにより、暗号化オペレーションが中断または遅延することはありません。新しいキーマテリアルが AWS KMS 全体で使用可能になるまで、現在のキーマテリアルが暗号化オペレーションで使用されます。マルチリージョンキーのキーマテリアルが自動的にローテーションされると、関連するマルチリージョンキーを持つすべてのリージョンで新しいキーマテリアルが使用可能になるまで、AWS KMS は現在のキーマテリアルを使用します。

「自動キーローテーションを有効または無効にする方法」

認可されたユーザーが AWS KMS コンソールまたは AWS KMS API を使用して、自動キーローテーションを有効化または無効化し、キーローテーションステータスを表示することができます。

自動キーローテーションを有効にすると、AWS KMS は、有効にした日から 1 年後に KMS キーのキーマテリアルをローテーションし、その後毎年ローテーションします。

トピック

- キーローテーションの有効化と無効化 (コンソール)
- キーローテーションを有効化および無効化する (AWS KMS API)

キーローテーションの有効化と無効化 (コンソール)

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Customer managed keys] (カスタマーマネージドキー) を選択します。(AWS マネージドキー のローテーションを有効化または無効化することはできません。これらのローテーションは毎年自動的に行われます。)
4. KMS キーのエイリアスまたは キー ID を選択します。
5. [キーローテーション] タブを選択します。

   [Key rotation] (キーローテーション) タブは、マルチリージョンの対称暗号化 KMS キーなど、AWS KMS が生成したキーマテリアルを持つ対称暗号化 KMS キー ([Origin] (オリジン) が [AWS_KMS]) の詳細ページにのみ表示されます。

非対称 KMS キー、HMAC KMS キー、インポートされたキーマテリアルを持つ KMS キー、またはカスタムキーストア内の KMS キーを自動的にローテーションすることはできません。ただし、手動でローテーションすることはできます。

6. [Automatically rotate this KMS key every year] (この KMS キーを毎年自動的にローテーションする) チェックボックスをオンまたはオフにします。

> **Note**
>
> KMS キーが無効または削除保留中の場合は [Automatically rotate this KMS key every year] (この KMS キーを毎年自動的にローテーションする) のチェックボックスはオフになっています。これを変更することはできません。KMS キーを有効にするか削除をキャンセルすると、キーローテーションのステータスが復元されます。詳細については、「キーの自動ローテーションの仕組み」および「AWS KMS キーのキーステータス」を参照してください。

7. [Save] (保存) を選択します。

キーローテーションを有効化および無効化する (AWS KMS API)

AWS Key Management Service (AWS KMS) API を使用して、自動キーローテーションを有効および無効にし、すべてのカスタマーマネージドキーの現在のローテーションステータスを表示できます。以下の例では AWS Command Line Interface (AWS CLI) を使用していますが、サポートされている任意のプログラミング言語を使用できます。

EnableKeyRotation オペレーションは、指定された KMS キーの自動キーローテーションを有効にします。DisableKeyRotation オペレーションによって無効になります。これらのオペレーションで KMS キーを識別するには、そのキー ID またはキー ARN を使用します。デフォルトでは、カスタマーマネージドキーのキーローテーションは無効になっています。

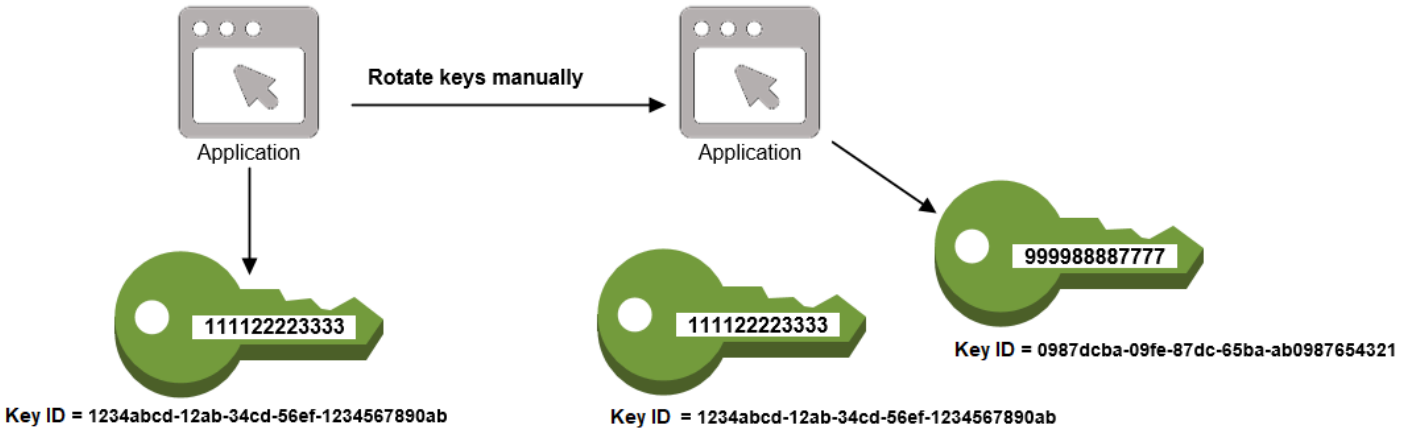
次の例では、指定された対称暗号化 KMS キーのキーローテーションを有効にし、GetKeyRotationStatusオペレーションを使用して結果を表示します。それから、キーローテーションを無効にし、再度、GetKeyRotationStatus を使用して変更を確認します。

```
$ aws kms enable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
```

```
    "KeyRotationEnabled": true
}

$ aws kms disable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyRotationEnabled": false
}
```

手動でのキーローテーション

自動キーローテーションを有効にする代わりに、新しい KMS キーを作成して、現在の KMS キーの代わりに使用することができます。新しい KMS キーが現在の KMS キーと異なる暗号化マテリアルを持つ場合、新しい KMS キーを使用すると、既存の KMS キーでキーマテリアルを変更するのと同じ効果があります。1 つの KMS キーを別のものと置き換えるプロセスは、手動キーローテーションと呼ばれます。



ローテーションの頻度をコントロールできるように、手動でキーをローテーションする方がよい場合があります。これは、非対称 KMS キー、HMAC KMS キー、カスタムキーストア内の KMS キー、およびインポートされたキーマテリアルを持つ KMS キーなど、自動キーローテーションの対象ではない KMS キーに適したソリューションでもあります。

> **Note**
>
> 新しい KMS キーの使用を開始する際は、AWS KMS が元の KMS キーによって暗号化されたデータを復号できるよう、必ず元の KMS キーを有効なままにしてください。

KMS キーを手動で更新すると、アプリケーションの KMS キー ID または キーの ARN へのリファレンスも更新する必要があります。KMS キーにわかりやすい名前を関連付けられるエイリアスが、このプロセスを容易にします。エイリアスを使用して、アプリケーションの KMS キーを参照します。続いて、アプリケーションが使用する KMS キーを変更するには、アプリケーションのコードを編集する代わりに、エイリアスのターゲット KMS キーを変更します。詳細については、「アプリケーションでのエイリアスの使用」を参照してください。

> **Note**
>
> 手動でローテーションされた KMS キーの最新バージョンを指すエイリアスはDescribeKey、、Encrypt 、、、GenerateMacおよび Sign GenerateDataKey GenerateDataKeyPairオペレーションに適しています。エイリアスは、 DisableKeyや などの KMS キーを管理するオペレーションでは許可されませんScheduleKeyDeletion。
> 手動で更新された対称暗号化 KMS キーに対する Decrypt オペレーションを呼び出すときは、コマンドの `KeyId` パラメータを省略します。暗号文を暗号化した KMS キーを、AWS KMS が自動的に使用します。
> 非対称 KMS キーを使用して `Decrypt`または Verify を呼び出す場合、または HMAC KMS キーVerifyMacを使用して を呼び出す場合は、 `KeyId`パラメータが必要です。キーが手動で更新された場合など、暗号化オペレーションを実行する KMS キーをもはや指していないエイリアスが、`KeyId` パラメータの値になっている場合には、これらのリクエストは失敗します。このエラーを回避するには、オペレーションごとに正しい KMS キーを追跡して、指定する必要があります。

エイリアスのターゲット KMS キーを変更するには、 AWS KMS API で UpdateAliasオペレーションを使用します。例えば、このコマンドでは、新しい KMS キーを指すように `alias/TestKey` エイリアスを更新します。オペレーションは出力を返さないため、この例では ListAliasesオペレーションを使用して、エイリアスが別の KMS キーに関連付けられ、 `LastUpdatedDate`フィールドが更新されたことを示します。 ListAliases コマンドは、 `alias/TestKey` の queryパラメータを使用して AWS CLIエイリアスのみを取得します。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1521097200.123
        },
    ]
}


$ aws kms update-alias --alias-name alias/TestKey --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321

$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1604958290.722
        },
    ]
}
```

AWS KMS keys のモニタリング

モニタリングは、AWS KMS の AWS KMS keys の可用性、ステータス、使用状況を理解し、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うえで重要な要素です。AWS ソリューションのすべてのパートからモニタリングデータを収集すると、マルチポイント障害が発生した場合にそれをデバッグするのに役立ちます。ただし、KMS キーのモニタリングをスタートする前に、以下の質問に対する回答を反映したモニタリング計画を作成する必要があります。

- モニタリングの目的は何ですか?
- どのリソースをモニタリングしますか?
- どのくらいの頻度でこれらのリソースをモニタリングしますか?
- 使用するモニタリングツールは ?
- 誰がモニタリングタスクを実行しますか?
- 問題が発生したときに誰が通知を受け取りますか ?

次のステップでは、KMS キーを経時的にモニタリングして、環境内での通常の AWS KMS の使用および想定のベースラインを確立します。KMS キーをモニタリングする際に、過去のモニタリングデータを保存し、現在のデータと比較することで、通常パターンと異常パターンを識別して、問題に対処する方法を考案できるようにします。

例えば、KMS キーに影響を与える AWS KMS API アクティビティとイベントをモニタリングできます。データが、確立基準を上回ったり、下回ったりする場合、修正作業を調査、または実行する必要が生じる場合があります。

通常のパターンのベースラインを確立するには、次の項目をモニタリングします。

- データプレーンオペレーション用の AWS KMS API アクティビティ。これらは、Decrypt 、Encrypt 、、 ReEncryptなどの KMS キーを使用する暗号化オペレーションですGenerateDataKey。
- 重要なコントロールプレーンオペレーションのための AWS KMS API アクティビティ。これらのオペレーションは KMS キーを管理し、KMS キーの可用性 (ScheduleKeyDeletion、、CancelKeyDeletion、DisableKey、 などDeleteImportedKeyMaterial) を変更したりImportKeyMaterial、KMS キーのアクセスコントロール (PutKeyPolicyや EnableKeyなど) を変更したりするオペレーションをモニタリングできますRevokeGrant。
- 他の AWS KMS メトリクス (インポートされたキーマテリアルが有効期限切れになるまでの残り時間など) およびイベント (インポートされたキーマテリアルの有効期限切れ、または KMS キーの削除やキーのローテーションなど)。

モニタリングツール

AWS は、KMS キーのモニタリングに使用できる各種のツールを提供します。これらのツールの中には、自動モニタリングを設定できるものもあれば、手オペレーションを必要とするものもあります。モニタリングタスクをできるだけ自動化することをお勧めします。

自動モニタリングツール

次の自動化されたモニタリングツールを使用して KMS キーを監視し、変更が生じたときに報告させることができます。

- AWS CloudTrail ログのモニタリング – アカウント間でログファイルを共有し、 CloudTrail ログを CloudWatch ログに送信してリアルタイムでモニタリングし、CloudTrail Processing Library を使用してログ処理アプリケーションを書き込み、 による配信後にログファイルが変更されていない

ことを確認します CloudTrail。詳細については、「 ユーザーガイド」の CloudTrail 「ログファイルの操作AWS CloudTrail」を参照してください。

- Amazon CloudWatch アラーム – 指定した期間にわたって単一のメトリクスを監視し、複数の期間にわたる特定のしきい値に対するメトリクスの値に基づいて 1 つ以上のアクションを実行します。アクションは、Amazon Simple Notification Service (Amazon SNS) トピックまたは Amazon EC2 Auto Scaling ポリシーに送信される通知です。 CloudWatch alarms は、単に特定の状態にあるというだけではアクションを呼び出しません。状態が変わり、指定された期間にわたって持続している必要があります。詳細については、「Amazon によるモニタリング CloudWatch」を参照してください。
- Amazon EventBridge - イベントをマッチングし、1 つ以上のターゲット関数またはストリームにルーティングして、状態情報をキャプチャし、必要に応じて変更を加えるか、修正作業を行います。詳細については、Amazon によるモニタリング EventBridge「」および「Amazon ユーザーガイド EventBridge 」を参照してください。
- Amazon CloudWatch Logs – AWS CloudTrailまたはその他のソースからのログファイルをモニタリング、保存、およびアクセスします。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。

手動モニタリングツール

KMS キーのモニタリングでもう 1 つ重要な点は、 CloudWatch アラームやイベントでカバーされていない項目を手動でモニタリングすることです。AWS KMS、AWS Trusted Advisor、 CloudWatch、およびその他の AWS ダッシュボードには、AWS環境の状態 at-a-glance が表示されます。

[AWS マネージドキー] および AWS KMS コンソールの [Customer managed keys] (カスタマーマネージドキー) ページをカスタマイズして、各 KMS キーに関する次の情報を表示できます。

- キー ID
- ステータス
- 作成日
- 有効期限 (インポートされたキーマテリアルを持つ KMS キーの場合)
- オリジン
- カスタムキーストア ID (カスタムキーストアのKMS キーの場合)

CloudWatch コンソールダッシュボードは、以下を示します。

- 現在のアラームとステータス

- アラームとリソースのグラフ
- サービスのヘルスステータス

さらに、 CloudWatch を使用して次の操作を実行できます。

- 重視するサービスをモニタリングするためのカスタマイズしたダッシュボードを作成します
- メトリクスデータをグラフ化して、問題のトラブルシューティングを行い、傾向を確認する
- AWS リソースのすべてのメトリクスを検索して、参照する
- 問題があることを通知するアラームを作成/編集する

AWS Trusted Advisor は、AWS リソースのパフォーマンス、信頼性、セキュリティ、費用効率を向上するためのモニタリングに役立ちます。すべてのユーザーは、4 つの Trusted Advisor; チェックを利用できます。ビジネスまたはエンタープライズサポートプランのユーザーは、50 以上のチェックを利用できます。詳細については、「AWS Trusted Advisor」を参照してください。

AWS KMS による AWS CloudTrail API コールのログ記録

AWS KMS は、 ユーザーAWS CloudTrail、ロール、およびその他の のサービスAWS KMSによる へのすべての呼び出しを記録するAWSサービスである と統合されています。 は、 AWS KMSコンソール、API、 AWS CloudFormation テンプレート、 AWS Command Line Interface (AWS CLI) 、 からの呼び出しを含む、 へのすべての AWS KMS API 呼び出しをイベントAWS KMSとして CloudTrail キャプチャしますAWS Tools for PowerShell。 APIs

CloudTrail はAWS KMS、 ListAliasesや などの読み取り専用オペレーション、GetKeyRotationStatusや などの KMS キーを管理するオペレーションCreateKeyPutKeyPolicy、 やGenerateDataKey Decrypt などの暗号化オペレーションを含むすべてのオペレーションを記録します。また、、、DeleteExpiredKeyMaterial、 などDeleteKey、 がAWS KMS呼び出す内部オペレーションも口グに記録されますSynchronizeMultiRegionKeyRotateKey。

CloudTrail は、呼び出し元がリソースへのアクセスを拒否された場合など、成功したオペレーションと失敗した試行された呼び出しを記録します。KMS キーに対するアカウントを横断したオペレーションは、発信者のアカウントと KMS キー所有者のアカウントの両方に記録されます。ただし、アクセスが拒否されたために拒否されたクロスアカウント AWS KMS リクエストは、呼び出し元のアカウントにのみ記録されます。

セキュリティ上の理由から、Encrypt リクエストの `Plaintext`パラメータや、 GetKeyPolicyまたは任意の暗号化オペレーションへの応答など、一部のフィールドはAWS KMSログエントリから省略さ

れます。特定の KMS キーの CloudTrail ログエントリを検索しやすくするために、 は、API オペレーションがキー ARN を返さない場合でも、一部のAWS KMSキー管理オペレーションのログエントリの `responseElements`フィールドに、影響を受ける KMS キーのキー ARN AWS KMSを追加します。

デフォルトでは、すべてのAWS KMSアクションが CloudTrail イベントとして記録されますが、証 CloudTrail 跡からAWS KMSアクションを除外できます。詳細については、「証跡からの AWS KMS イベントの除外」を参照してください。

詳細はこちら:

- AWS Nitro Enclave のAWS KMSオペレーションの CloudTrail ログの例については、「」を参照してくださいNitro Enclaves に対するリクエストの監視。

トピック

- でのイベントのログ記録 CloudTrail
- でのイベントの検索 CloudTrail
- 証跡からの AWS KMS イベントの除外
- AWS KMS ログエントリの例

でのイベントのログ記録 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。でアクティビティが発生するとAWS KMS、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。 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、「イベント履歴 での CloudTrail イベントの表示」を参照してください。

AWS KMS 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対して、追跡を作成します。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AWS のサービスするように他の を設定できます。詳細については、以下をご覧ください。

- 証跡を作成するための概要

- CloudTrail 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ログファイルの受信 CloudTrail

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。KMS キーの使用をモニタリングするその他の方法については、AWS KMS keys のモニタリング を参照してください。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するために役立ちます。

- リクエストが、ルート認証情報または 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一時的なセキュリティ認証情報を使用して送信されたか。
- リクエストが、別の AWS のサービス によって行われたか。

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

でのイベントの検索 CloudTrail

CloudTrail ログエントリを検索するには、 CloudTrail コンソールまたは CloudTrail LookupEventsオペレーションを使用します。 は、イベント名、ユーザー名、イベントソースなど、検索をフィルタリングするための多数の属性値 CloudTrail をサポートします。

でAWS KMSログエントリを検索しやすくするために CloudTrail、 AWS KMS は次の CloudTrail ログエントリフィールドに入力します。

> **Note**
>
> 2022 年 12 月以降、AWS KMS は、特定の KMS キーを変更するすべての管理オペレーションでリソースタイプとリソース名の各属性を入力しています。これらの属性値は、、、CreateAlias、CreateGrant、、、、DeleteAlias、、UpdateAlias、、および の各オペレーションの古い CloudTrail エントリでは null DeleteImportedKeyMaterial ImportKeyMaterial ReplicateKey RetireGrant RevokeGrantになる場合がありますUpdatePrimaryRegion。

属性	値	ログエントリ
イベントソース (EventSource)	kms.amazonaws.com	すべてのオペレーション
リソースタイプ (ResourceType)	AWS::KMS::Key	特定の KMS キーを変更する管理オペレーション (CreateKey や EnableKey など。ListKeys は除く)。
リソース名 (ResourceName)	キー ARN (またはキー ID およびキー ARN)	特定の KMS キーを変更する管理オペレーション (CreateKey や EnableKey など。ListKeys は除く)。

特定の KMS キーの、管理オペレーションのログエントリを容易に検索するために、AWS KMS は、AWS KMS API オペレーションがキー ARN を返さない場合でも、ログエントリの responseElements.keyId エレメントに、影響を受けた KMS キーのキー ARN を記録します。

例えば、 DisableKeyオペレーションを正常に呼び出すと、レスポンスの値は返されませんが、NULL 値の代わりに、DisableKey ログエントリresponseElements.keyIdの値には無効になっている KMS キーのキー ARN が含まれます。

この機能は 2022 年 12 月に追加され、次の CloudTrail ログエントリに影響します: CreateAlias、CreateGrant、DeleteAlias、DeleteKey、、、DisableKeyEnableKeyEnableKeyRotation、ImpoUpdatePrimaryRegion

証跡からの AWS KMS イベントの除外

AWS KMS リソースの使用と管理を記録するため、ほとんどのAWS KMSユーザーは CloudTrail 証跡内のイベントに依存しています。追跡は、AWS KMS keys の作成、無効化、削除、およびキーポリシーの変更、ユーザーに代わって AWS のサービスが行う KMS キーの使用など、重要なイベントを監査するための貴重なデータソースになります。場合によっては、暗号化オペレーションの暗号化コンテキストなど、 CloudTrail ログエントリのメタデータがエラーを回避または解決するのに役立ちます。

ただし、AWS KMS では多数のイベントを生成できるため、AWS CloudTrail では証跡から AWS KMS イベントを除外できます。この証跡単位の設定では、すべての AWS KMS イベントが除外されます。特定の AWS KMS イベントを除外することはできません。

> **Warning**
>
> CloudTrail ログからAWS KMSイベントを除外すると、KMS キーを使用するアクションが隠されることがあります。このオペレーションを実行するために必要な `cloudtrail:PutEventSelectors` アクセス許可をプリンシパルに与えるときは注意してください。

証跡から AWS KMS イベントを除外するには:

- CloudTrail コンソールで、証跡の作成時または証跡の更新時に、 ログキー管理サービスのイベント設定を使用します。手順については、AWS CloudTrail ユーザーガイドの Logging Management Events with the AWS Management Console を参照してください。
- CloudTrail API では、 PutEventSelectorsオペレーションを使用します。`ExcludeManagementEventSources` 属性を `kms.amazonaws.com` の値でイベントセレクタに追加します。例については、AWS CloudTrail ユーザーガイドのExample: A trail that does not log AWS Key Management Service events を参照してください。

この除外は、コンソール設定または証跡のイベントセレクタを変更することでいつでも無効にできます。その後、証跡は AWS KMS イベントの記録を開始します。ただし、除外が有効である間に発生した AWS KMS イベントはリカバリできません。

コンソールまたは API を使用してAWS KMSイベントを除外すると、結果 CloudTrailの `PutEventSelectors` API オペレーションも CloudTrail Logs に記録されます。AWS KMS イベントが CloudTrail ログに表示されない場合は、 `ExcludeManagementEventSources` 属性が に設定されている`PutEventSelectors`イベントを探します`kms.amazonaws.com`。

AWS KMS ログエントリの例

AWS KMS は、 AWS KMSオペレーションを呼び出すとき、および AWSサービスがユーザーに代わって オペレーションを呼び出すときに CloudTrail 、ログにエントリを書き込みます。AWS KMS また、 は、 オペレーションを呼び出すときにもエントリを書き込みます。例えば、削除をスケジュールした KMS キーを削除すると、エントリが書き込まれます。

以下のトピックでは、 AWS KMSオペレーションの CloudTrail ログエントリの例を示します。

AWS Nitro Enclaves AWS KMSから へのリクエストの CloudTrail ログエントリの例については、「」を参照してくださいNitro Enclaves に対するリクエストの監視。

トピック

- CancelKeyDeletion
- ConnectCustomKeyStore
- CreateAlias
- CreateCustomKeyStore
- CreateGrant
- CreateKey
- Decrypt
- DeleteAlias
- DeleteCustomKeyStore
- DeleteExpiredKeyMaterial
- DeleteImportedKeyMaterial
- DeleteKey
- DescribeCustomKeyStores
- DescribeKey
- DisableKey
- DisableKeyRotation
- DisconnectCustomKeyStore
- EnableKey
- EnableKeyRotation
- 暗号化
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- GenerateRandom

- GetKeyPolicy
- GetKeyRotationStatus
- GetParametersForImport
- ImportKeyMaterial
- ListAliases
- ListGrants
- PutKeyPolicy
- ReEncrypt
- ReplicateKey
- RetireGrant
- RevokeGrant
- RotateKey
- ScheduleKeyDeletion
- Sign
- SynchronizeMultiRegionKey
- TagResource
- UntagResource
- UpdateAlias
- UpdateCustomKeyStore
- UpdateKeyDescription
- UpdatePrimaryRegion
- VerifyMac
- Verify
- Amazon EC2 の例1
- Amazon EC2 の例 2

CancelKeyDeletion

次の例は、AWS CloudTrail オペレーションを呼び出して生成された CancelKeyDeletion ログエントリを示しています。AWS KMS keys の削除の詳細については、「AWS KMS keys を削除する」を参照してください。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T21:53:17Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CancelKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "e3452e68-d4b0-4ec7-a768-7ae96c23764f",
    "eventID": "d818bf03-6655-48e9-8b26-f279a07075fd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ConnectCustomKeyStore

次の例は、AWS CloudTrail オペレーションを呼び出して生成された ConnectCustomKeyStore ログエントリを示しています。カスタムキーストアの接続に関する詳細については、「AWS CloudHSM キーストアの接続と切断」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ConnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```


CreateAlias

次の例は、 CreateAliasオペレーションの AWS CloudTrailログエントリを示しています。resources 要素には、エイリアスと KMS キーリソースのフィールドが含まれます。AWS KMS でのエイリアスの作成については、エイリアスの作成 を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-14T23:08:31Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/ExampleAlias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "caec1e0c-ce03-419e-bdab-6ab1f7c57c01",
    "eventID": "2dd6e784-8286-46a6-befd-d64e5a02fb28",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

CreateCustomKeyStore

次の例は、CreateCustomKeyStore オペレーションを AWS CloudHSM キーストアで呼び出して生成された AWS CloudTrail ログエントリを示しています。カスタムキーストアの作成に関する詳細については、「AWS CloudHSM キーストアの作成」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

CreateGrant

次の例は、 CreateGrantオペレーションの AWS CloudTrailログエントリを示しています。AWS KMS での許可の作成の詳細については、「AWS KMS でのグラント」を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。


```
{
  "eventVersion": "1.02",
  "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "AWS Internal",
  "requestParameters": {
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "constraints": {
          "encryptionContextSubset": {
              "ContextKey1": "Value1"
          }
      },
      "operations": ["Encrypt",
      "RetireGrant"],
      "granteePrincipal": "EX_PRINCIPAL_ID"
  },
  "responseElements": {
      "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
      "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
```

```
  "eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
  "readOnly": false,
  "resources": [{
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

CreateKey

これらの例は、 CreateKeyオペレーションのAWS CloudTrailログエントリを示しています。

CreateKey ログエントリは、 CreateKey リクエストまたは ReplicateKeyリクエストの CreateKeyオペレーションの結果です。

次の例は、対称暗号化 KMS キー を作成するCreateKeyオペレーションの CloudTrail ログエントリを示しています。 ???KMS キー作成の詳細については、キーの作成 を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-10T22:38:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "description": "",
        "origin": "EXTERNAL",
        "bypassPolicyLockoutSafetyCheck": false,
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "keySpec": "SYMMETRIC_DEFAULT",
```

```
        "keyUsage": "ENCRYPT_DECRYPT"
    },
    "responseElements": {
        "keyMetadata": {
            "AWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Aug 10, 2022, 10:38:27 PM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "PendingImport",
            "origin": "EXTERNAL",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "requestID": "1aef6713-0223-4ff7-9a6d-781360521930",
    "eventID": "36327b37-f4f6-40a9-92ab-48064ec905a2",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


次の例は、 キーAWS CloudHSMストアに対称暗号化 KMS キーを作成する `CreateKey`オペレーションの CloudTrail ログを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyUsage": "ENCRYPT_DECRYPT",
        "bypassPolicyLockoutSafetyCheck": false,
        "origin": "AWS_CLOUDHSM",
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "customKeyStoreId": "cks-1234567890abcdef0",
        "description": ""
    },
    "responseElements": {
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "creationDate": "Oct 14, 2021, 5:39:50 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "AWS_CLOUDHSM",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "cloudHsmClusterId": "cluster-1a23b4cdefg",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
```

```
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "additionalEventData": {
        "backingKey": "{\"keyHandle\":\"19\",\"backingKeyId\":\"backing-key-id\"}"
    },
    "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
    "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

次の例は、外部キーストア に対称暗号化 KMS キーを作成するCreateKeyオペレーションの CloudTrail ログを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2022-09-07T22:37:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "AWS Internal",
    "requestParameters": {
        "tags": [],
        "keyUsage": "ENCRYPT_DECRYPT",
        "description": "",
        "origin": "EXTERNAL_KEY_STORE",
        "multiRegion": false,
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "bypassPolicyLockoutSafetyCheck": false,
        "customKeyStoreId": "cks-1234567890abcdef0",
        "xksKeyId": "bb8562717f809024"
    },
    "responseElements": {
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Dec 7, 2022, 10:37:45 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "EXTERNAL_KEY_STORE",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false,
            "xksKeyConfiguration": {
                "id": "bb8562717f809024"
            }
        }
    },
    "requestID": "ba197c82-3ac7-487a-8ff4-7736bbeb1316",
    "eventID": "838ad5f4-5fdd-4044-afd7-4dbd88c6af56",
    "readOnly": false,
    "resources": [
        {
            "accountId": "227179770375",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:227179770375:key/39c5eb22-
f37c-4956-92ca-89e8f8b57ab2"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

これらの例は、Decrypt オペレーションの AWS CloudTrail ログエントリを示しています。

オペレーションの CloudTrail ログエントリには、リクエストencryptionAlgorithmで暗号化アルゴリズムが指定されていない場合requestParametersでも、Decrypt常に に が含まれます。リクエスト内の暗号化テキストとレスポンス内のプレーンテキストは省略されます。

トピック

- 標準の対称暗号化キーを使用した復号
- 標準の対称暗号化キーを使用した復号の失敗
- AWS CloudHSM キーストアの KMS キーによる復号
- 外部キーストアの KMS キーを使用した復号
- 外部キーストアの KMS キーを使用した復号の失敗

標準の対称暗号化キーを使用した復号

以下は、標準の対称暗号化キーを使用した Decryptオペレーションの CloudTrail ログエントリの例です。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
```

```
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

標準の対称暗号化キーを使用した復号の失敗

次の CloudTrail ログエントリの例では、標準対称暗号化 KMS キーを使用して、失敗したDecryptオペレーションを記録します。例外 (errorCode) とエラーメッセージ (errorMessage) が記載されており、エラーの解決に役立ちます。

このケースでは、Decrypt リクエストで指定された対称暗号化 KMS キーは、データの暗号化に使用された対称暗号化 KMS キーではありませんでした。

```
{
    "eventVersion": "1.08",
```

```
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T18:57:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "IncorrectKeyException"
    "errorMessage": "The key ID in the request does not identify a CMK that can perform
 this operation.",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "22345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

AWS CloudHSM キーストアの KMS キーによる復号

次の CloudTrail ログエントリの例では、 キーAWS CloudHSMストア の KMS キーを使用して Decryptオペレーションを記録します。カスタムキーストアの KMS キーを使用した暗号化オペレーションのすべてのログエントリには、customKeyStoreId を持つ [additionalEventData] フィールドが含まれます。additionalEventData は、このリクエストでは指定されていません。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:41:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Development",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "e1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "a79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

外部キーストアの KMS キーを使用した復号

次の CloudTrail ログエントリの例では、外部キーストア の KMS キーを使用した Decryptオペレーションを記録します。additionalEventDataフィールドには、customKeyStoreId の他に外部キー ID (XksKeyId) が含まれます。additionalEventData は、このリクエストでは指定されていません。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
```

```
        "customKeyStoreId": "cks-9876543210fedcba9",
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

外部キーストアの KMS キーを使用した復号の失敗

次の CloudTrail ログエントリの例では、成功したリクエストに加えて、失敗した外部キーストアの .logs リクエストに KMS キーを含む Decryptオペレーションの失敗したリクエストを記録します。 CloudWatch 失敗を記録する場合、 CloudTrail ログエントリには例外 (errorCode) とそれに付随するエラーメッセージ (errorMessage) が含まれます。

こちらの例のように、失敗リクエストが外部キーストアのプロキシに到達した場合、requestId 値を使って失敗したリクエストを、外部キーストアプロキシが記録する、対応するリクエスト (プロキシが提供している場合) に関連付けることができます。

外部キーストアでの Decrypt リクエストに関するヘルプは、「復号エラー」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
```

```
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "errorCode": "KMSInvalidStateException",
    "errorMessage": "The external key store proxy rejected the request because the
 specified ciphertext or additional authenticated data is corrupted, missing, or
 otherwise invalid.",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-9876543210fedcba9",
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DeleteAlias

次の例は、 DeleteAliasオペレーションの AWS CloudTrailログエントリを示しています。エイリアスの削除については、を参照してください エイリアスの削除。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でも`responseElements.keyId`、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2014-11-04T00:52:27Z"
            }
        }
    },
    "eventTime": "2014-11-04T00:52:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteAlias",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9542792-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "12f48554-bb04-4991-9cfc-e7e85f68eda0",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-east-1:111122223333:alias/my_alias",
```

```
        "accountId": "111122223333"
    },
    {
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DeleteCustomKeyStore

次の例は、DeleteCustomKeyStore オペレーションを呼び出して生成された AWS CloudTrail ログエントリを示しています。カスタムキーストアの作成に関する詳細については、「AWS CloudHSM キーストアの削除」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
```

```
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

DeleteExpiredKeyMaterial

キーマテリアルを AWS KMS key (KMS キー) にインポートすると、そのキーマテリアルの有効期限日時を設定できます。 は、キーマテリアルをインポートするとき (有効期限設定を含む) と、 が期限切れのキーマテリアルAWS KMSを削除するときに CloudTrail 、ログにエントリAWS KMSを記録します。インポートされたキーマテリアルを使用する KMS キーの作成の詳細については、キーの AWS KMS キーマテリアルのインポート を参照してください。

次の例は、AWS KMS が期限切れのキーマテリアルを削除するときに生成される AWS CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-01T16:00:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteExpiredKeyMaterial",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
```

```
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    }
}
```

DeleteImportedKeyMaterial

KMS キーにキーマテリアルをインポートする場合、DeleteImportedKeyMaterialオペレーションを使用して、インポートされたキーマテリアルをいつでも削除できます。インポートしたキーマテリアルを KMS キーから削除すると、このKMS キーのキーの状態は PendingImport に変わり、暗号化オペレーションに使用できなくなります。詳細については、「インポートされたキーマテリアルの削除」を参照してください。

次の例は、DeleteImportedKeyMaterial オペレーションの AWS CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-10-04T21:43:33Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteImportedKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"&example-key-arn-1;"
    },
    "requestID": "dcf0e82f-dad0-4622-a378-a5b964ad42c1",
    "eventID": "2afbb991-c668-4641-8a00-67d62e1fecbd",
    "readOnly": false,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DeleteKey

次の例は、KMS キーが削除されたときに生成される AWS CloudTrail ログエントリを示しています。KMS キーを削除するには、 ScheduleKeyDeletionオペレーションを使用します。指定された待機期間が終了すると、 は KMS キーAWS KMSを削除し、次のようなエントリを CloudTrail ログに記録してそのイベントを記録します。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でも`responseElements.keyId`、影響を受ける KMS キーのキー ARN が値に含まれます。

`ScheduleKeyDeletion` オペレーションの CloudTrail ログエントリの例については、「」を参照してくださいScheduleKeyDeletion。KMS キーの削除の詳細については、AWS KMS keys を削除するを参照してください。

次の CloudTrail ログエントリの例では、 のキーマテリアルを持つ KMS キーの`DeleteKey`オペレーションを記録しますAWS KMS。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-07-31T00:07:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
```

```
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
     "managementEvent": true,
    "eventCategory": "Management"
}
```

次の CloudTrail ログエントリは、 AWS CloudHSMカスタムキーストアの KMS キーのDeleteKeyオペレーションを記録します。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-10-26T23:41:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
```

```
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"01\",\"backingKeyId\":
\"backing-key-id\",\"deletionStatus\":\"SUCCESS\"}]"
    },
    "eventID": "1234585c-4b0c-4340-ab11-662414b79239",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```


DescribeCustomKeyStores

次の例は、DescribeCustomKeyStores オペレーションを呼び出して生成された AWS CloudTrail ログエントリを示しています。カスタムキーストアの表示に関する詳細については、「AWS CloudHSM キーストアの表示」を参照してください。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeCustomKeyStores",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
```

```
    },
    "responseElements": null,
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

DescribeKey

次の例は、 DescribeKeyオペレーションの AWS CloudTrailログエントリを示しています。 AWS KMSコンソールで DescribeKeyオペレーションを呼び出すか、KMS キーを表示すると、 は次のようなエントリAWS KMSを記録します。この呼び出しは、AWS KMS 管理コンソールでキーを表示した結果です。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-26T18:01:36Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DisableKey

次の例は、 DisableKeyオペレーションの AWS CloudTrailログエントリを示しています。AWS KMS での AWS KMS keys の有効化と無効化の詳細については、「キーの有効化と無効化」を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でも`responseElements.keyId`、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
```

```
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DisableKeyRotation

次の例は、AWS CloudTrail オペレーションを呼び出して生成された DisableKeyRotation ログエントリを示しています。自動キーローテーションについては、「AWS KMS keys ローテーション」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:31:39Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "d6a9351a-ed6e-4581-88d1-2a9a8a538497",
    "eventID": "6313164c-83aa-4cc3-9e1a-b7c426f7a5b1",
    "readOnly": false,
    "resources": [
        {
```

```
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DisconnectCustomKeyStore

次の例は、DisconnectCustomKeyStore オペレーションを呼び出して生成された AWS CloudTrail ログエントリを示しています。カスタムキーストアの切断に関する詳細については、「AWS CloudHSM キーストアの接続と切断」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisconnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
```

```
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

EnableKey

次の例は、 EnableKeyオペレーションの AWS CloudTrailログエントリを示しています。AWS KMS での AWS KMS keys の有効化と無効化の詳細については、キーの有効化と無効化 を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でも`responseElements.keyId`、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:20Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d528a6fb-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "be393928-3629-4370-9634-567f9274d52e",
    "readOnly": false,
```

```
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

EnableKeyRotation

次の例は、 EnableKeyRotationオペレーションへの呼び出しの AWS CloudTrailログエントリを示しています。キーがローテーションされたときに書き込まれる CloudTrail ログエントリの例については、「」を参照してくださいRotateKey。AWS KMS keys のローテーションの詳細については、AWS KMS keys ローテーション を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:41:56Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
```

```
    "requestID": "81f5b794-452b-4d6a-932b-68c188165273",
    "eventID": "fefc43a7-8e06-419f-bcab-b3bf18d6a401",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

暗号化

次の例は、Encrypt オペレーションの AWS CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-07-14T20:17:42Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionContext": {
            "Department": "Engineering"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    },
    "responseElements": null,
```

```
    "requestID": "f3423043-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "91235988-eb87-476a-ac2c-0cdc244e6dca",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey

次の例は、 GenerateDataKeyオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
```

```
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair

次の例は、 GenerateDataKeyPairオペレーションの AWS CloudTrailログエントリを示しています。この例は、対称暗号化 AWS KMS keyで暗号化された RSA キーペアを生成するオペレーションを記録します。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPairWithoutPlaintext

次の例は、 GenerateDataKeyPairWithoutPlaintextオペレーションの AWS CloudTrailログエントリを示しています。この例は、対称暗号化 AWS KMS keyで暗号化された RSA キーペアを生成するオペレーションを記録します。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPairWithoutPlaintext",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_4096",
        "encryptionContext": {
            "Index": "5"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
```

```
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyWithoutPlaintext

次の例は、 GenerateDataKeyWithoutPlaintextオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "InvalidKeyUsageException",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "responseElements": null,
```

```
    "requestID": "d6b8e411-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f7734272-9ec5-4c80-9f36-528ebbe35e4a",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateMac

次の例は、 GenerateMacオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-12-23T19:26:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_512",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateRandom

次の例は、 GenerateRandomオペレーションの AWS CloudTrailログエントリを示しています。このオペレーションでは AWS KMS key を使用しないため、[resources] フィールドは空です。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GetKeyPolicy

次の例は、GetKeyPolicyオペレーションの AWS CloudTrailログエントリを示しています。KMS キーのキーポリシーの表示についての詳細は、キーポリシーの表示 を参照してください。


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:50:30Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default"
    },
    "responseElements": null,
    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


GetKeyRotationStatus

次の例は、GetKeyRotationStatusオペレーションの AWS CloudTrailログエントリを示しています。KMS キーのキーマテリアルの、自動ローテーションの詳細については、「AWS KMS keys ローテーション」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:32:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyRotationStatus",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12f9b7e8-49b9-4c1c-a7e3-34ac0cdf0467",
    "eventID": "3d082126-9e7d-4167-8372-a6cfcbed4be6",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


GetParametersForImport

次の例は、 GetParametersForImportオペレーションを使用するときに生成される AWS CloudTrail ログエントリを示しています。このオペレーションは、KMS キーにキーマテリアルをインポートするときに使用する公開キーとインポートトークンを返します。同じ CloudTrail エントリは、

GetParametersForImportオペレーションを使用するか、 AWS KMS コンソールを使用してパブリックキーとインポートトークンをダウンロードするときに記録されます。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:58:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetParametersForImport",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "wrappingAlgorithm": "RSAES_OAEP_SHA_256",
        "wrappingKeySpec": "RSA_2048"
    },
    "responseElements": null,
    "requestID": "b5786406-e3c7-43d6-8d3c-6d5ef96e2278",
    "eventID": "4023e622-0c3e-4324-bdef-7f58193bba87",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ImportKeyMaterial

次の例は、 ImportKeyMaterialオペレーションを使用するときに生成される AWS CloudTrailログエントリを示しています。ImportKeyMaterial オペレーションを使用するか、AWS KMSコンソールを使用してキーマテリアルを にインポートすると、同じ CloudTrail エントリが記録されますAWS KMS key。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-26T00:08:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "validTo": "Jan 1, 2021 8:00:00 PM",
        "expirationModel": "KEY_MATERIAL_EXPIRES"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ListAliases

次の例は、 ListAliasesオペレーションの AWS CloudTrailログエントリを示しています。このオペレーションでは特定のエイリアスまたは AWS KMS key を使用しないため、resources フィールドは空です。AWS KMS でのエイリアスの表示については、エイリアスの表示 を参照してください。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:51:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListAliases",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "limit": 5,
        "marker":
 "eyJiIjoiYWxpYXMvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwiYSI6ImFsaWFzL2U1NGNjMTkzL
    },
    "responseElements": null,
    "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ListGrants

次の例は、 ListGrantオペレーションの AWS CloudTrailログエントリを示しています。AWS KMS の許可の詳細については、「AWS KMS でのグラント」を参照してください。


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListGrants",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "marker":
 "eyJncmFudElkIjoiMWY4M2U2ZmM0YTY2NDgxYjQ2Yzc4MTdhM2Y4YmQwMDFkZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzc1N
\u003d\u003d",
        "limit": 10
    },
    "responseElements": null,
    "requestID": "e5c23960-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "d24380f5-1b20-4253-8e92-dd0492b3bd3d",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

PutKeyPolicy

次の例は、AWS CloudTrail オペレーションを呼び出して生成された PutKeyPolicy ログエントリを示しています。キーポリシー更新する方法の詳細については、「キーポリシーの変更」を参照してください。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T20:06:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "PutKeyPolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default",
        "policy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-default-1\",\n
 \"Statement\" : [ {\n    \"Sid\" : \"Enable IAM User Permissions\",\n    \"Effect\" :
 \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::111122223333:root
\"\n    },\n    \"Action\" : \"kms:*\",\n    \"Resource\" : \"*\"\n  } ]\n}",
        "bypassPolicyLockoutSafetyCheck": false
    },
    "responseElements": null,
    "requestID": "7bb906fa-dc21-4350-b65c-808ff0f72f55",
    "eventID": "c217db1f-903f-4a2f-8f88-9580182d6313",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

ReEncrypt

次の例は、 ReEncryptオペレーションの AWS CloudTrailログエントリを示しています。このログエントリの `resources` フィールドでは、ソース KMS キーと送信先 KMS キーの 2 つの AWS KMS keys を、この順序で指定します。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T23:09:13Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "sourceEncryptionContext": {
            "Project": "Alpha",
            "Department": "Engineering"
        },
        "destinationKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationEncryptionContext": {
            "Level": "3A"
        }
    },
    "responseElements": null,
    "requestID": "03769fd4-acf9-4b33-adf3-2ab8ca73aadf",
    "eventID": "542d9e04-0e8d-4e05-bf4b-4bdeb032e6ec",
```

```
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ReplicateKey

次の例は、AWS CloudTrail オペレーションを呼び出して生成された ReplicateKey ログエントリを示しています。ReplicateKey リクエストの結果、 ReplicateKeyオペレーションと CreateKeyオペレーションが発生します。

マルチリージョンキーのレプリケーションについては、マルチリージョンのレプリカキーを作成するを参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-11-18T01:29:18Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReplicateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "replicaRegion": "us-west-2",
        "bypassPolicyLockoutSafetyCheck": false,
        "description": ""
    },
    "responseElements": {
        "replicaKeyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Nov 18, 2020, 1:29:18 AM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Creating",
            "origin": "AWS_KMS",
            "keyManager": "CUSTOMER",
            "keySpec": "SYMMETRIC_DEFAULT",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": true,
            "multiRegionConfiguration": {
                "multiRegionKeyType": "REPLICA",
                "primaryKey": {
                    "arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                    "region": "us-east-1"
                },
                "replicaKeys": [
                    {
                        "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                        "region": "us-west-2"
                    }
                ]
            }
        },
        "replicaPolicy": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":[{\n
 \"Effect\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:user/
```

```
Alice\"},\n    \"Action\":\"kms:*\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect
\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Bob\"},\n
    \"Action\":\"kms:CreateGrant\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect\":
\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Charlie\"},\n
  \"Action\":\"kms:Encrypt\",\n    \"Resource\":\"*\"\n}]\n}",
    },
    "requestID": "abcdef68-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "fedcba44-6773-4f96-8763-1993aec9ae6a",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


RetireGrant

次の例は、AWS CloudTrail オペレーションを呼び出して生成された RetireGrant ログエントリを示しています。グラント廃止の詳細については、「グラントの使用停止と取り消し」を参照してください。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:39:33Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "requestID": "1d274d57-5697-462c-a004-f25fcc29fa26",
    "eventID": "0771bcfb-3e24-4332-9ac8-e1c06563eecf",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RevokeGrant

次の例は、AWS CloudTrail オペレーションを呼び出して生成された RevokeGrant ログエントリを示しています。グラント取り消しの詳細については、「グラントの使用停止と取り消し」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:35:17Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RevokeGrant",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "responseElements": null,
    "requestID": "59d94c03-c5b7-428d-ae6e-f2c4b47d2917",
    "eventID": "07a23a39-6526-4ae2-b31e-d35fbe9e24ee",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


RotateKey

次の例は、AWS KMS key をローテーションするオペレーションの AWS CloudTrail ログエントリを示しています。AWS KMS は、自動キーローテーションが有効化された KMS キーがローテーションされるタイミングでこのオペレーションを呼び出します。自動キーローテーション () を有効にするとEnableKeyRotation、 は 365 日後、その後は 365 日ごとに KMS キーをAWS KMSローテーションします。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

EnableKeyRotation オペレーションを記録する CloudTrail ログエントリの例については、「」を参照してくださいEnableKeyRotation。KMS キーのローテーションの詳細については、AWS KMS keys ローテーション を参照してください。


```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-01-14T01:41:59Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    }
}
```


ScheduleKeyDeletion

これらの例は、 ScheduleKeyDeletionオペレーションのAWS CloudTrailログエントリを示しています。

キーの削除時に書き込まれる CloudTrail ログエントリの例については、「」を参照してくださいDeleteKey。AWS KMS keys の削除の詳細については、「AWS KMS keys を削除する」を参照してください。

次の例では、単一リージョン KMS キーに対する ScheduleKeyDeletion リクエストが記録されています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-23T18:58:30Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "pendingWindowInDays": 20,
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyState": "PendingDeletion",
        "deletionDate": "Apr 12, 2021 18:58:30 PM"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

次の例では、レプリカキーを持つマルチリージョン KMS キーに対する ScheduleKeyDeletion リクエストが記録されています。

AWS KMS では、すべてのレプリカキーが削除されるまでマルチリージョンキーは削除されないため、responseElements フィールドでは、keyState が PendingReplicaDeletion になり、deletionDate フィールドは省略されます。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-28T17:59:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "pendingWindowInDays": 30,
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "keyState": "PendingReplicaDeletion",
        "pendingWindowInDays": 30
    },
    "requestID": "12341411-d846-42a6-a476-b1cbe3011f89",
    "eventID": "abcda5f-396d-494c-9380-0c47860df5f1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

次の例では、AWS CloudHSM カスタムキーストアの KMS キーに対する ScheduleKeyDeletion リクエストが記録されています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:25:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "pendingWindowInDays": 30
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "deletionDate": "Nov 2, 2021, 11:25:25 PM",
        "keyState": "PendingDeletion",
        "pendingWindowInDays": 30
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]"
    },
```

```
    "requestID": "abcd9f60-2c9c-4a0b-a456-d5d998f7f321",
    "eventID": "ca01996a-01b0-4edd-bbbb-25d7b6d1a6fa",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Sign

これらの例は、Sign オペレーションの AWS CloudTrail ログエントリを示しています。

次の例は、非対称 RSA KMS キーを使用してファイルのデジタル署名を生成する Sign オペレーションの CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:36:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Sign",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "messageType": "RAW",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
```

```
    },
    "responseElements": null,
    "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
    "eventID": "107b3cac-b125-4556-9702-12a2b9afcff7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

SynchronizeMultiRegionKey

次の例は、AWS KMS がマルチリージョンキーを同期するときに生成される AWS CloudTrail ログエントリを示しています。同期化には、マルチリージョンのプライマリキーの共有プロパティを、そのレプリカキーにコピーするための、クロスリージョンの呼び出しが含まれます。AWS KMS はマルチリージョンキーを定期的に同期し、関連するすべてのマルチリージョンキーが同じキーマテリアルを持つようにします。

CloudTrail ログエントリの `resources`要素には、 を含むマルチリージョンのプライマリキーのキー ARN が含まれますAWS リージョン。関連するマルチリージョンレプリカキーとそのリージョンは、このログエントリには一覧表示されません。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でも`responseElements.keyId`、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-11-18T02:04:37Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "SynchronizeMultiRegionKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345681-de97-42e9-bed0-b02ae1abd8dc",
    "eventID": "abcdec99-2b5c-4670-9521-ddb8f031e146",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

TagResource

次の例は、タグキーが でタグDepartment値が のタグを追加する TagResourceオペレーションへの呼び出しの AWS CloudTrailログエントリを示していますIT。

キーがローテーションされたときに書き込まれる UntagResource CloudTrail ログエントリの例については、「」を参照してくださいUntagResource。AWS KMS keys のタグ付けの詳細については、キーのタグ付け を参照してください。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
```

```
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tags": [
            {
                "tagKey": "Department",
                "tagValue": "IT"
            }
        ]
    },
    "responseElements": null,
    "requestID": "b942584a-f77d-4787-9feb-b9c5be6e746d",
    "eventID": "0a091b9b-0df5-4cf9-b667-6f2879532b8f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


UntagResource

次の例は、タグキーが のタグを削除する UntagResourceオペレーションの呼び出しの AWS CloudTrailログエントリを示していますDept。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

TagResource CloudTrail ログエントリの例については、「」を参照してくださいいTagResource。AWS KMS keys のタグ付けの詳細については、キーのタグ付け を参照してください。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tagKeys": [
            "Dept"
        ]
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "cb1d507b-6015-47f4-812b-179713af8068",
    "eventID": "0b00f4b0-036e-411d-aa75-87eb4a35a4b3",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
```

```
    "recipientAccountId": "111122223333"
}
```

UpdateAlias

次の例は、 UpdateAliasオペレーションの AWS CloudTrailログエントリを示しています。resources 要素には、エイリアスと KMS キーリソースのフィールドが含まれます。AWS KMS でのエイリアスの作成については、エイリアスの作成 を参照してください。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "EX_PRINCIPAL_ID",
         "arn": "arn:aws:iam::111122223333:user/Alice",
         "accountId": "111122223333",
         "accessKeyId": "EXAMPLE_KEY_ID",
         "userName": "Alice"
    },
    "eventTime": "2020-11-13T23:18:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9472f40-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f72d3993-864f-48d6-8f16-e26e1ae8dff0",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/my_alias"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UpdateCustomKeyStore

次の例は、カスタムキーストアのクラスター ID を更新する UpdateCustomKeyStore オペレーションの呼び出しによって生成された AWS CloudTrail ログエントリ示しています。カスタムキーストアの編集に関する詳細については、「AWS CloudHSM キーストア設定の編集」を参照してください。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": null,
    "additionalEventData": {
```

```
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

UpdateKeyDescription

次の例は、AWS CloudTrail オペレーションを呼び出して生成された UpdateKeyDescription ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:22:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateKeyDescription",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "description": "New key description"
    },
    "responseElements": null,
    "requestID": "8c3c1f8b-336d-4896-b034-4eb9916bc9b3",
    "eventID": "f5f3d548-2e9e-4658-8427-9dcb5b1ea791",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
```

```
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

UpdatePrimaryRegion

次の例は、マルチリージョンキー で UpdatePrimaryRegionオペレーションを呼び出して生成されるAWS CloudTrailログエントリを示しています。

UpdatePrimaryRegion オペレーションは 2 つの CloudTrail ログエントリを書き込みます。1 つはレプリカキーに変換されたマルチリージョンのプライマリキーを持つリージョンにあり、もう 1 つはプライマリキーに変換されたマルチリージョンのレプリカキーを持つリージョンにあります。

CloudTrail 2022 年 12 月以降に記録されたこのオペレーションの ログエントリには、このオペレーションがキー ARN を返さない場合でもresponseElements.keyId、影響を受ける KMS キーのキー ARN が値に含まれます。

次の例は、マルチリージョンキーがプライマリキーからレプリカキー (us-west-2) に変更されたリージョンUpdatePrimaryRegionの の CloudTrail ログエントリを示しています。primaryRegion フィールドは、プライマリキー (ap-northeast-1) をホストするリージョンを表示します。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

次の例は、マルチリージョンキーがレプリカキーからプライマリキー (ap-northeast-1) に変更されたリージョンUpdatePrimaryRegionの の CloudTrail ログエントリを示しています。このログエントリは、以前のプライマリリージョンを識別しません。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "invokedBy": "kms.amazonaws.com"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
```

```
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "ap-northeast-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "091e6be5-737f-43c6-8431-e3679d6d0619",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

VerifyMac

次の例は、 VerifyMacオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-03-31T19:25:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "VerifyMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_384",
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "f35da560-edff-4d6e-9b40-fb306fa9ef1e",
    "eventID": "6b464487-6dea-44cd-84ad-225d7450c975",
    "readOnly": true,
    "resources": [
        {
           "accountId": "111122223333",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Verify

これらの例は、Verify オペレーションの AWS CloudTrail ログエントリを示しています。

次の例は、非対称 RSA KMS キーを使用してデジタル署名を検証する Verify オペレーションの CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:50:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Verify",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256",
```

```
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "messageType": "RAW"
    },
    "responseElements": null,
    "requestID": "c73ab82a-af82-4750-ae2c-b6bb790e9c28",
    "eventID": "3b4331cd-5b7b-4de5-bf5f-82ec22f0dac0",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Amazon EC2 の例1

次の例では、Amazon EC2 マネジメントコンソールで、デフォルトのボリュームキーを使用して暗号化されたボリュームを作成する IAM プリンシパルが記録されています。

次の例は、ユーザー Alice が Amazon EC2 マネジメントコンソールでデフォルトのボリュームキーを使用して暗号化されたボリュームを作成する CloudTrail ログエントリを示しています。EC2 ログファイルレコードには、値が "vol-13439757" の volumeId フィールドが含まれます。AWS KMS レコードには、値が "aws:ebs:id": "vol-13439757" の encryptionContext フィールドが含まれます。同様に、2 つのレコード間の principalId と accountId が一致します。レコードは、暗号化されたボリュームを作成するとボリュームコンテンツの暗号化に使用されるデータキーが生成されるという事実を示しています。

```
{
  "Records": [
     {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
```

```
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
        "size": "10",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1415220618876,
        "volumeType": "gp2",
        "iops": 30,
        "encrypted": true
      },
      "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
      "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
      "eventType": "AwsApiCall",
      "recipientAccountId": "123456789012"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:19Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
```

```
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "&AWS; Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-13439757"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-123456789012-758241111-1415220618",
      "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```

Amazon EC2 の例 2

次の例では、Amazon EC2 インスタンスを実行する IAM プリンシパルにより、KMS キーで暗号化されたデータボリュームを作成してマウントします。このアクションにより、複数の CloudTrail ログレコードが生成されます。

ボリュームが作成されると、お客様の代わりに動作する Amazon EC2 は、AWS KMS (`GenerateDataKeyWithoutPlaintext`) から暗号化されたデータキーを取得します。次に、データキーの復号を可能にする許可 (`CreateGrant`) を作成します。ボリュームがマウントされると、Amazon EC2 は AWS KMS を呼び出してデータキー (`Decrypt`) を復号します。

Amazon EC2 インスタンスの `instanceId` である `"i-81e2f56c"` が `RunInstances` イベントに表示されます。同じインスタンス ID が、作成された許可 (`"111122223333:aws:ec2-infrastructure:i-81e2f56c"`) の `granteePrincipal` と、`Decrypt` 呼び出し

("arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c") のプリンシパルである引き受けたロールを修飾します。

データボリュームを保護する KMS キーの キー ARN、arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab は、3 つの AWS KMS すべてに表示され、(CreateGrant、GenerateDataKeyWithoutPlaintext、Decrypt) を呼び出します。


```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:27Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "RunInstances",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "instancesSet": {
          "items": [
            {
              "imageId": "ami-b66ed3de",
              "minCount": 1,
              "maxCount": 1
            }
          ]
        },
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2"
            }
          ]
        },
```

```
        "instanceType": "m3.medium",
        "blockDeviceMapping": {
          "items": [
            {
              "deviceName": "/dev/xvda",
              "ebs": {
                "volumeSize": 8,
                "deleteOnTermination": true,
                "volumeType": "gp2"
              }
            },
            {
              "deviceName": "/dev/sdb",
              "ebs": {
                "volumeSize": 8,
                "deleteOnTermination": false,
                "volumeType": "gp2",
                "encrypted": true
              }
            }
          ]
        },
        "monitoring": {
          "enabled": false
        },
        "disableApiTermination": false,
        "instanceInitiatedShutdownBehavior": "stop",
        "clientToken": "XdKUT141516171819",
        "ebsOptimized": false
      },
      "responseElements": {
        "reservationId": "r-5ebc9f74",
        "ownerId": "111122223333",
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2",
              "groupName": "launch-wizard-2"
            }
          ]
        },
        "instancesSet": {
          "items": [
            {
```

```
            "instanceId": "i-81e2f56c",
            "imageId": "ami-b66ed3de",
            "instanceState": {
              "code": 0,
              "name": "pending"
            },
            "amiLaunchIndex": 0,
            "productCodes": {

            },
            "instanceType": "m3.medium",
            "launchTime": 1415223328000,
            "placement": {
              "availabilityZone": "us-east-1a",
              "tenancy": "default"
            },
            "monitoring": {
              "state": "disabled"
            },
            "stateReason": {
              "code": "pending",
              "message": "pending"
            },
            "architecture": "x86_64",
            "rootDeviceType": "ebs",
            "rootDeviceName": "/dev/xvda",
            "blockDeviceMapping": {

            },
            "virtualizationType": "hvm",
            "hypervisor": "xen",
            "clientToken": "XdKUT1415223327917",
            "groupSet": {
              "items": [
                {
                  "groupId": "sg-98b6e0f2",
                  "groupName": "launch-wizard-2"
                }
              ]
            },
            "networkInterfaceSet": {

            },
            "ebsOptimized": false
```

```
            }
          ]
        }
      },
      "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
      "eventID": "cd75a605-2fee-4fda-b847-9c3d330ebaae",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:35Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "CreateGrant",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "constraints": {
          "encryptionContextSubset": {
            "aws:ebs:id": "vol-f67bafb2"
          }
        },
        "granteePrincipal": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },
      "responseElements": {
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
      },
      "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
      "eventID": "c1ad79e3-0d3f-402a-b119-d5c31d7c6a6c",
      "readOnly": false,
      "resources": [
        {
```

```
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "EX_PRINCIPAL_ID",
          "arn": "arn:aws:iam::111122223333:user/Alice",
          "accountId": "111122223333",
          "accessKeyId": "EXAMPLE_KEY_ID",
          "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:32Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "AWS Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-111122223333-758247346-1415223332",
      "eventID": "ac3cab10-ce93-4953-9d62-0b6e5cba651d",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
```

```
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "arn": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-81e2f56c",
        "accountId": "111122223333",
        "accessKeyId": "",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2014-11-05T21:35:38Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "111122223333:aws:ec2-infrastructure",
            "arn": "arn:aws:iam::111122223333:role/aws:ec2-infrastructure",
            "accountId": "111122223333",
            "userName": "aws:ec2-infrastructure"
          }
        }
      },
      "eventTime": "2014-11-05T21:35:47Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "Decrypt",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "responseElements": null,
      "requestID": "b4b27883-6533-11e4-b4d9-751f1761e9e5",
      "eventID": "edb65380-0a3e-4123-bbc8-3d1b7cff49b0",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
```

```
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```

Amazon によるモニタリング CloudWatch

Amazon AWS KMS keysを使用して をモニタリングできます。Amazon CloudWatchは、 から 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AWS KMSに加工する AWSサービスです。これらのデータは、履歴情報にアクセスして、時間の経過に伴う KMS キーの使用や変更に関する理解を深められるように 2 週間記録されます。

Amazon を使用して、次のような重要なイベントを CloudWatch アラートできます。

- KMS キーにインポートされたキーマテリアルの有効期限が近づいています。
- 削除保留中の KMS キーがまだ使用されています。
- KMS キーのキーマテリアルが自動的にローテーションされました。
- KMS キーが削除されました。

リクエストレートがクォータ値の特定のパーセンテージに達したときに警告する Amazon CloudWatch アラームを作成することもできます。詳細については、 AWS セキュリティブログの「Service Quotas と Amazon を使用して AWS KMS API リクエストレートを管理する CloudWatch」を参照してください。

トピック

- AWS KMS のメトリクスとディメンション
- AWS KMS メトリクスの表示
- KMS キーをモニタリングする CloudWatch アラームの作成

AWS KMS のメトリクスとディメンション

AWS KMS は Amazon CloudWatch メトリクスを事前に定義して、重要なデータのモニタリングとアラームの作成を容易にします。AWS Management Console および Amazon CloudWatch API を使用してAWS KMSメトリクスを表示できます。

このセクションでは、各AWS KMSメトリクスと各メトリクスのディメンションを一覧表示し、これらのメトリクスとディメンションに基づいて CloudWatch アラームを作成するための基本的なガイダンスを提供します。

> **Note**
>
> ディメンショングループ名:
> Amazon CloudWatch コンソールでメトリクスを表示するには、メトリクスセクションでディメンショングループ名を選択します。これにより、[Metric name] (メトリクス名) でフィルタリングできます。このトピックには、各 AWS KMS メトリクスのメトリクス名とディメンショングループ名が含まれています。

トピック

- SecondsUntilKeyMaterialExpiration
- ExternalKeyStoreThrottle
- XksProxyCertificateDaysToExpire
- XksProxyCredentialAge
- XksProxyErrors
- XksExternalKeyManagerStates
- XksProxyLatency

SecondsUntilKeyMaterialExpiration

KMS キーにインポートされたキーマテリアルの有効期限が切れるまでの残り秒数です。このメトリクスは、インポートされたキーマテリアル (EXTERNAL のキーマテリアルのオリジン) と有効期限を持つ KMS キーに対してのみ有効です。

このメトリクスを使用して、インポートしたキーマテリアルの有効期限までの残り時間を追跡します。残り秒数が定義したしきい値を下回った場合は、新しい有効期限を使用してキーマテリアルを再インポートできます。この `SecondsUntilKeyMaterialExpiration` メトリクスは KMS キーに固有です。このメトリクスを使用して、複数の KMS キーや将来作成する可能性のある KMS キーを監視することはできません。このメトリクスをモニタリングする CloudWatch アラームの作成については、「」を参照してくださいインポートされたキーマテリアルの有効期限切れの CloudWatch アラームの作成。

このメトリクスで最も有用な統計は Minimum で、指定された統計期間のすべてのデータポイントの最小残り時間を示します。このメトリクスの唯一の有効な単位は Seconds です。

ディメンショングループ名: [Per-Key Metrics] (キーごとのメトリクス)

SecondsUntilKeyMaterialExpiration のディメンション

ディメンション	説明: 関連する AWS
KeyId	各 KMS キーの値です。

ExternalKeyStoreThrottle

AWS KMS がスロットルする (ThrottlingException で応答する) 各外部キーストアの KMS キーに対する暗号化オペレーションのリクエスト数です。このメトリクスは、外部キーストアにのみ適用されます。

ExternalKeyStoreThrottle メトリクスは、外部キーストアの KMS キーと、暗号化オペレーションと DescribeKeyオペレーションのリクエストにのみ適用されます。 は、リクエストレートが外部キーストアのカスタムキーストアリクエストクォータを超えると、これらのリクエスト AWS KMSを調整します。 ???このメトリクスには、外部キーストアプロキシまたは外部キーマネージャーによるスロットリングは含まれていません。

このメトリクスを使用して、カスタムキーストアのリクエストクォータの値を確認および調整します。AWS KMS がこれらの KMS キーのリクエストを頻繁にスロットリングしていることをこのメトリクスが示している場合は、カスタムキーストアのリクエストクォータ値の引き上げをリクエストすることを検討してください。ヘルプについては、「Service Quotas ユーザーガイド」の「Requesting a quota increase」(クォータ引き上げのリクエスト) を参照してください。

「リクエストレートが極めて高いため」リクエストが拒否されたこと、または「外部キーストアプロキシが時間内に応答しなかったために」リクエストが拒否されたことを説明するメッセージで KMSInvalidStateException エラーが頻発する場合は、外部キーマネージャーまたは外部キーストアプロキシが現在のリクエストレートに対応していないことを示している可能性があります。可能な場合は、リクエスト率を下げます。また、カスタムキーストアのリクエストクォータ値の引き下げをリクエストすることも検討してください。このクォータ値を引き下げると、スロットリング (および ExternalKeyStoreThrottle メトリック値) が増加する可能性がありますが、AWS KMS は超過リクエストが外部キーストアプロキシまたは外部キーマネージャーに送信される前にすぐに拒否し

ます。クォータの削減をリクエストするには、AWS Support センターにアクセスしてケースを作成してください。

ディメンショングループ名: [Keystore Throttle Metrics] (キーストアスロットルメトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。
KmsOperation	各 AWS KMS API オペレーションの値です。このメトリクスは、暗号化オペレーションと外部キーストアの KMS キーに対する `DescribeKey` オペレーションにのみ適用されます。
KeySpec	KMS キーの各タイプの値です。外部キーストアの KMS キーでサポートされているキースペックは SYMMETRIC_DEFAULT のみです。

XksProxyCertificateDaysToExpire

外部キーストアプロキシエンドポイント (`XksProxyUriEndpoint`) の TLS 証明書の有効期限が切れるまでの日数です。このメトリクスは、外部キーストアにのみ適用されます。

このメトリクスを使用して、TLS 証明書の今後の有効期限を通知する CloudWatch アラームを作成します。証明書の有効期限が切れると、AWS KMS は外部キーストアプロキシと通信できなくなります。証明書が更新されるまで、外部キーストアの KMS キーで保護されているすべてのデータにアクセスできなくなります。

証明書アラームは、暗号化されたリソースへのアクセスを妨げる可能性のある証明書の有効期限切れを防ぎます。アラームを設定することで、組織は有効期限が切れる前に証明書を更新する時間をもつことができます。

ディメンショングループ名: [XKS Proxy Certificate Metrics] (XKS プロキシ証明書メトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。

ディメンション	説明
CertificateName	TLS 証明書のサブジェクト名 (CN) です。

XksProxyCredentialAge

現在の外部キーストアのプロキシ認証情報 (XksProxyAuthenticationCredential) が外部キーストアに関連付けられてからの日数です。このカウントは、外部キーストアの作成または更新の一環として、認証情報を入力した時点から開始されます。このメトリクスは、外部キーストアにのみ適用されます。

この値は、認証情報の有効期限を知らせるためのものです。ただし、外部キーストアプロキシで認証情報を作成した時点ではなく、認証情報を外部キーストアに関連付けた時点からカウントが開始されるため、プロキシでの認証情報の経過時間の正確なインジケータではない場合があります。

このメトリクスを使用して、外部キーストアプロキシ認証の認証情報をローテーションするよう通知する CloudWatch アラームを作成します。

ディメンショングループ名: [Per-Keystore Metrics] (キーストアごとのメトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。

XksProxyErrors

外部キーストアプロキシへの AWS KMS リクエストに関連する例外の数です。このカウントには、外部キーストアプロキシが AWS KMS に返す例外と、外部キーストアプロキシが 250 ミリ秒のタイムアウト間隔内に AWS KMS に応答しない場合に発生するタイムアウトエラーが含まれます。このメトリクスは、外部キーストアにのみ適用されます。

このメトリクスを使用して、外部キーストアの KMS キーのエラーレートを追跡します。最も頻発するエラーが明らかになるため、エンジニアリング作業に優先順位を付けることができます。例えば、

再試行不可能なエラーの発生率が高くなっている KMS キーは、外部キーストアの設定に問題があることを示している可能性があります。外部キーストア設定を確認するには、「外部キーストアを表示する」を参照してください。外部キーストア設定を編集するには、「外部キーストアのプロパティの編集」を参照してください。

ディメンショングループ名: [XKS Proxy Error Metrics] (XKS プロキシエラーメトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。
KmsOperation	XKS プロキシへのリクエストを生成した各 AWS KMS API オペレーションの値です。
XksOperation	各外部キーストアプロキシ API オペレーションの値です。
KeySpec	KMS キーの各タイプの値です。外部キーストアの KMS キーでサポートされているキースペックは SYMMETRIC_DEFAULT のみです。
ErrorType	値: • 再試行可能なエラー: ネットワークエラーなど、一時的なエラーである可能性があります。 • 再試行不可能なエラー: カスタムキーストア設定または外部コンポーネントの問題を示している可能性があります。 • N/A: リクエストに成功、エラーなし
ExceptionName	値: • 例外の名前 • なし: リクエストに成功、エラーなし

XksExternalKeyManagerStates

以下の各ヘルス状態 (Active、Degraded、Unavailable) における外部キーマネージャーインスタンス数のカウントです。このメトリクスの情報は、各外部キーストアに関連付けられた外部キーストアプロキシから取得されます。このメトリクスは、外部キーストアにのみ適用されます。

以下は、外部キーストアに関連付けられた外部キーマネージャーインスタンスのヘルス状態です。各外部キーストアプロキシは、外部キーマネージャーのヘルス状態を測定するために、異なるインジケータを使用する場合があります。詳細については、外部キーストアプロキシのドキュメントを参照してください。

- Active: 外部キーマネージャーは正常です。
- Degraded: 外部キーマネージャーに異常がありますが、トラフィックは処理できます。
- Unavailable: 外部キーマネージャーはトラフィックを処理できません。

このメトリクスを使用して、外部キーマネージャーインスタンスのパフォーマンスが低下し、使用できなくなった場合に警告する CloudWatch アラームを作成します。各状態の外部キーマネージャーインスタンスを判断するには、外部キーストアプロキシログを参照してください。

ディメンショングループ名: [XKS External Key Manager Metrics] (XKS 外部キーマネージャーメトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。
XksExternalKeyManagerState	各ヘルス状態の値です。

XksProxyLatency

外部キーストアプロキシが AWS KMS リクエストに応答するまでにかかるミリ秒数です。リクエストがタイムアウトした場合、記録される値は 250 ミリ秒のタイムアウト制限です。このメトリクスは、外部キーストアにのみ適用されます。

このメトリクスを使用して、外部キーストアプロキシと外部キーマネージャーのパフォーマンスを評価します。プロキシが暗号化オペレーションと復号オペレーションで頻繁にタイムアウトする場合は、外部プロキシ管理者に相談してください。

応答が遅い場合は、外部キーマネージャーが現在のリクエストトラフィックを処理できていない可能性もあります。AWS KMS では、外部キーマネージャーが 1 秒あたり最大 1,800 件の暗号化オペレーションリクエストを処理できることを推奨しています。外部キーマネージャーが 1 秒あたり 1,800 件のリクエストを処理できない場合は、カスタムキーストアの KMS キーリクエストクォータの引き下げをリクエストすることを検討してください。外部キーストアの KMS キーを使用した暗号化オペレーションのリクエストは、外部キーストアプロキシまたは外部キーマネージャーによって処理され、後で拒否されるのではなく、スロットリング例外でフェイルファストします。

ディメンショングループ名: [XKS Proxy Latency Metrics] (XKS プロキシレイテンシーメトリクス)

ディメンション	説明
CustomKeyStoreId	各外部キーストアの値です。
KmsOperation	XKS プロキシへのリクエストを生成した各 AWS KMS API オペレーションの値です。
XksOperation	各外部キーストアプロキシ API オペレーションの値です。
KeySpec	KMS キーの各タイプの値です。外部キーストアの KMS キーでサポートされているキースペックは SYMMETRIC_DEFAULT のみです。

AWS KMS メトリクスの表示

AWS Management Console および Amazon CloudWatch API を使用してAWS KMSメトリクスを表示できます。

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. 必要に応じてリージョンを変更します。ナビゲーションバーから、AWS リソースがあるリージョンを選択します。

3. ナビゲーションペインで、[Metrics]、[All metrics] を選択します。
4. [ブラウズ] タブで「KMS」を検索し、[KMS] を選択します。
5. 表示するメトリクスのディメンショングループ名を選択します。

   例えば、SecondsUntilKeyMaterialExpiration メトリクスには [Per-Key Metrics] (キーごとのメトリクス) を選択します。
6. メトリクス値のグラフを作成するには、メトリクス名を選択し、Add to graph を選択します。折れ線グラフを値に変換するには、[ライン] を選択し、次に [数値] を選択します。

Amazon CloudWatch API を使用してメトリクスを表示するには

CloudWatch API を使用してAWS KMSメトリクスを表示するには、 を Namespaceに設定して ListMetricsリクエストを送信しますAWS/KMS。次の例では、AWS Command Line Interface (AWS CLI) を使用してこのオペレーションを行う方法を示します。

```
$  aws cloudwatch list-metrics --namespace AWS/KMS

{
    "Metrics": [
        {
            "Namespace": "AWS/KMS",
            "MetricName": "SecondsUntilKeyMaterialExpiration",
            "Dimensions": [
                {
                    "Name": "KeyId",
                    "Value": "1234abcd-12ab-34cd-56ef-1234567890ab"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "ExtenalKeyStoreThrottle",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Encrypt"
                },
```

```
                {
                    "Name": "KeySpec",
                    "Value": "SYMMETRIC_DEFAULT"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCertificateDaysToExpire",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "CertificateName",
                    "Value": "myproxy.xks.example.com"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyCredentialAge",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyErrors",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
```

```
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                },
                {
                    "Name": "ErrorType",
                    "Value": "Retryable errors"
                },
                {
                     "Name": "ExceptionName",
                     "Value": "KMSInvalidStateException"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyHsmStates",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "XksProxyHsmState",
                    "Value": "Active"
                }
            ]
        },
        {
            "Namespace": "AWS/KMS",
            "MetricName": "XksProxyLatency",
            "Dimensions": [
                {
                    "Name": "CustomKeyStoreId",
                    "Value": "cks-1234567890abcdef0"
                },
                {
                    "Name": "KmsOperation",
                    "Value": "Decrypt"
                },
                {
                    "Name": "XksOperation",
```

```
                    "Value": "Decrypt"
                },
                {
                     "Name": "KeySpec",
                     "Value": "SYMMETRIC_DEFAULT"
                }
            ]
        }
    ]
}
```

KMS キーをモニタリングする CloudWatch アラームの作成

AWS KMS メトリクスに基づいて Amazon CloudWatch アラームを作成できます。メトリクス値がアラーム設定で指定されたしきい値を超えると、アラームは E メールメッセージを送信します。アラームは、Amazon Simple Notification Service (Amazon SNS) のトピックまたは Amazon EC2 Auto Scaling のポリシーに E メールメッセージを送信できます。 CloudWatch アラームの詳細については、「Amazon ユーザーガイド」の「Amazon CloudWatch アラームの使用 CloudWatch 」を参照してください。

インポートされたキーマテリアルの期限切れに関するアラームの作成

`SecondsUntilKeyMaterialExpiration` メトリクスを使用して、KMS キーにインポートされたキーマテリアルの有効期限が近づいたときに通知する CloudWatch アラームを作成できます。

キーマテリアルを KMS キーにインポートすると、キーマテリアルの有効期限の日時を任意で指定することができます。キーマテリアルが有効期限切れになると、AWS KMS はキーマテリアルを削除し、KMS キーは使用不可能になります。KMS キーを再度使用するには、キーマテリアルを再インポートする必要があります。

手順については、「インポートされたキーマテリアルの有効期限切れの CloudWatch アラームの作成」を参照してください。

削除保留中の KMS キーの使用状況に関するアラームを作成する

KMS キーのキー削除をスケジュールすると、AWS KMS は KMS キーを削除する前に待機時間を強制します。待機期間を設定することで、KMS キーが不要であり、今後も使用しないことを確認できます。また、待機期間中に暗号化オペレーションでユーザーまたはアプリケーションが KMS キーを使用しようとした場合に警告するようにアラームを設定 CloudWatchすることもできます。このようなアラームから通知を受け取った場合は、KMS キーの削除をキャンセルする必要がある可能性があります。

手順については、「削除保留中の KMS キーの使用を検出するアラームの作成」を参照してください。

外部キーストアをモニタリングするためのアラームを作成する

外部キーストアと外部キーストアの KMS キーのメトリクスに基づいて CloudWatch アラームを作成できます。

例えば、外部キーストアの TLS 証明書の有効期限が近づいたとき (XksProxyCertificateDaysToExpire)、 のとき、および外部キーストアプロキシが外部キーマネージャーインスタンスの状態が低下または使用不可であると報告したとき () に通知する CloudWatch アラームを設定することをお勧めしますXksProxyHsmStates。

手順については、「外部キーストアのモニタリング」を参照してください。

Amazon によるモニタリング EventBridge

Amazon EventBridge（ 以前の Amazon CloudWatch Events) を使用して、KMS キーのライフサイクルにおける以下の重要なイベントを警告できます。

- KMS キーのキーマテリアルが自動的にローテーションされました。
- KMS キーにインポートされたキーマテリアルの有効期限が切れました。
- 削除が予定されていた KMS キーが削除されました。

AWS KMS は Amazon と統合 EventBridge して、KMS キーに影響する重要なイベントを通知します。各イベントは JSON (JavaScript Object Notation) で表され、イベント名、イベントが発生した日時、および影響を受ける が含まれます。これらのイベントを収集し、AWS Lambda 関数、Amazon SNS トピック、Amazon SQS キュー、Amazon Kinesis Data Streams のストリーム、組み込みターゲットなどの 1 つ以上のターゲットにイベント送信のルールを確立できます。

読み取り/書き込み API リクエストを記録するAWS CloudTrailときに によって出力されるイベントなど、他の種類のイベント EventBridge で を使用する方法の詳細については、「Amazon EventBridge ユーザーガイド」を参照してください。

以下のトピックでは、 がAWS KMS生成する EventBridge イベントについて説明します。

KMS CMK ローテーション

AWS KMS は、対称暗号化 KMS キーのキーマテリアルの自動ローテーションをサポートします。カスタマーマネージドキーについては、キーマテリアルの年次ローテーションはオプションです。AWS マネージドキー のキーマテリアルは毎年自動的にローテーションされます。

がキーマテリアルをローAWS KMSテーションするたびに、 にKMS CMK Rotationイベントが送信されます EventBridge。 はベストエフォートベースでこのイベントAWS KMSを生成します。

このイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

KMS でインポートされたキーマテリアルの有効期限

キーマテリアルを KMS キーにインポートすると、キーマテリアルの有効期限を任意で指定することができます。キーマテリアルの有効期限が切れると、 はキーマテリアルAWS KMSを削除し、対応するKMS Imported Key Material Expirationイベントを に送信します EventBridge。 はベストエフォートベースでこのイベントAWS KMSを生成します。

このイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
```

```
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

KMS CMK 削除

KMS キーのキー削除をスケジュールすると、AWS KMS は KMS キーを削除する前に待機時間を強制します。待機期間が終了すると、 は KMS キーAWS KMSを削除し、 にKMS CMK Deletionイベントを送信します EventBridge。 はこの EventBridge イベントをAWS KMS保証します。再試行により、同じ KMS キーを削除する複数のイベントが数秒以内に生成される場合があります。

このイベントの例を以下に示します。

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2022-08-10T16:37:50Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

AWS CloudFormation での AWS KMS リソースの作成

AWS Key Management Service は、リソースとインフラストラクチャの作成と管理の所要時間を短縮できるように AWS リソースをモデル化して設定するためのサービスである AWS CloudFormation と統合されています。KMS キーとエイリアスを記述したテンプレートを作成すれば、AWS

CloudFormation がこれらのリソースのプロビジョニングや設定を処理します。AWS KMS のサポートについては CloudFormation、「 AWS CloudFormationユーザーガイド」の「KMS リソースタイプのリファレンス」を参照してください。

AWS CloudFormation を使用すると、テンプレートを再利用して AWS KMS リソースを同じように繰り返してセットアップできます。リソースを一度記述するだけで、同じリソースを複数の AWS アカウント とリージョンで何度でもプロビジョニングできます。

AWS KMS および他の AWS サービスのリソースをプロビジョニングして設定するには、AWS CloudFormation テンプレートについて理解しておく必要があります。テンプレートは、JSON や YAML でフォーマットされたテキストファイルです。これらのテンプレートには、AWS CloudFormation スタックにプロビジョニングしたいリソースを記述します。JSON や YAML に不慣れな方は、AWS CloudFormation Designer を使えば、AWS CloudFormation テンプレートを使いこなすことができます。詳細については、「AWS CloudFormation ユーザーガイド」の「AWS CloudFormation Designer とは」を参照してください。

リージョン

AWS KMS CloudFormation リソースは、 がサポートされているすべてのリージョンでサポートAWS CloudFormationされています。

AWS CloudFormation テンプレートの AWS KMS リソース

AWS KMS では、以下の AWS CloudFormation リソースがサポートされています。

- `AWS::KMS::Key` が、対称または非対称 KMS キーを作成します。このリソースを使用して、対称または非対称のマルチリージョンのプライマリ KMS キーを作成できます。マルチリージョンのレプリカキーを作成するには、`AWS::KMS::ReplicaKey` リソースを使用します。このリソースを使用して、インポートしたキーマテリアルの KMS キー、またはカスタムキーストアの KMS キーを作成することはできません。
- `AWS::KMS::Alias` がエイリアスを作成し、それを KMS キーに関連付けます。KMS キーは、テンプレートで定義することも、別のメカニズムで作成することもできます。
- `AWS::KMS::ReplicaKey` が、マルチリージョンレプリカキーを作成します。マルチリージョンのプライマリキーを作成するには、`AWS::KMS::Key` リソースを使用します。このリソースを使用して、インポートしたキーマテリアルでマルチリージョンキーをレプリケートすることはできません。マルチリージョンキーの詳細については、「AWS KMS のマルチリージョンキー」を参照してください。

> **Important**
>
> 既存の KMS キーの `KeyUsage`、`KeySpec`、`MultiRegion` プロパティを変更すると、既存の KMS キーの削除がスケジュールされ、指定された値で新しい KMS キーが作成されます。削除がスケジュールされている間、既存の KMS キーは使用できなくなります。AWS CloudFormation の外部で既存の KMS キーのスケジュールされた削除をキャンセルしない場合、既存の KMS キーで暗号化されたすべてのデータは、KMS キーが削除されると回復不能になります。

テンプレートによって作成される KMS キーは、AWS アカウント の実際のリソースです。認可されたプリンシパルは、テンプレート、AWS KMS コンソール、AWS KMS APIのいずれかを使用して、テンプレートで作成された KMS キーを使用および管理できます。テンプレートから KMS キーを削除すると、事前に指定した待機期間を使用して、KMS キーの削除がスケジュールされます。

例えば、AWS CloudFormation テンプレートを使用して、キーポリシー、キー仕様、キー使用法、エイリアス、タグを持つテスト KMS キーを作成できます。テストスイートで実行し、結果を確認してから、テンプレートを使用してテストキーの削除をスケジュールできます。その後、テンプレートを再度実行して、同じプロパティを持つテストキーを作成できます。

または、AWS CloudFormation テンプレートを使用して、ビジネスルールとセキュリティスタンダードを満たす特定の KMS キー設定を定義できます。その後、KMS キーを作成する必要があるときは、いつでもそのテンプレートを使用できます。設定ミスしたキーについて心配する必要もありません。希望する設定が変更された場合は、テンプレートを使用して KMS キーを更新できます。例えば、テンプレートを使用すると、テンプレートで定義されているすべての KMS キーの自動キーローテーションをプログラムによって簡単に有効化できます。

AWS KMS リソースの例を含む詳細については、「AWS CloudFormation ユーザーガイド」の「KMS リソースタイプリファレンス」を参照してください。

AWS CloudFormation の詳細情報

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

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

AWS KMS keys を削除する

AWS KMS key を削除することは破壊的であり、リスクが伴います。これは、キーマテリアルと KMS キーに関連付けられているすべてのメタデータを削除し、元に戻すことはできません。KMS キーを削除すると、その KMS キーで暗号化されたデータを復号できなくなります。これは、そのデータが回復不能になることを意味します。(唯一の例外は、マルチリージョンのレプリカキーと、キーマテリアルを含む非対称キーと HMAC KMS キーです。) 暗号化に使用される非対称 KMS キーの場合に重大なリスクとなります。この場合、ユーザーは、AWS KMS からプライベートキーが削除された後、警告やエラーが発生することなく、パブリックキーを使用して暗号文を生成し続けることができます。

KMS キーの削除は、そのキーをもう使用しないことが確実である場合にのみ行ってください。不明な場合は、削除するのではなく、KMS キーを無効化することを検討します。無効にした KMS キーを再度有効にしたり、KMS キーの削除のスケジュールをキャンセルしたりすることは可能ですが、すでに削除した KMS キーを復元することはできません。

スケジュールできるのは、カスタマーマネージドキーの削除のみです。AWS マネージドキー または AWS 所有のキー を削除することはできません。

KMS キーを削除する前に、その KMS キーで暗号化された暗号文の数の確認が必要な場合があります。AWS KMS は、この情報や暗号文を保存しません。この情報を取得するには、KMS キーの過去の使用状況を特定する必要があります。ヘルプについては、KMS キーの過去の使用状況を確認するを参照してください。

明示的に削除をスケジュールし、必須の待機期間が終了しない限り、AWS KMS は KMS キーを削除しません。

KMS キーを削除する理由には次の 1 つ以上のものが考えられます。

- 不要になった KMS キーのキーライフサイクルを完了する
- 使用しない KMS キーの維持に伴う管理オーバーヘッドとコストを回避する
- KMS キーリソースクォータに対してカウントされる KMS キーの数を減らすには

> **Note**
>
> AWS アカウント を終了すると、KMS キーにアクセスできなくなり、それらに対して課金されることはなくなります。

AWS KMS は、KMS キーの削除をスケジュールするとき、および KMS キーが実際に削除されたときに、AWS CloudTrail ログにエントリを記録します。

マルチリージョンのプライマリキーおよびレプリカキーの削除の詳細については、マルチリージョンキーを削除する を参照してください。

トピック

- 待機期間について
- 非対称 KMS キーの削除
- マルチリージョンキーを削除する
- インポートされたキーマテリアルを含む KMS キーの削除
- キー削除へのアクセスを制御する
- キーの削除のスケジュールとキャンセル
- 削除保留中の KMS キーの使用を検出するアラームの作成
- KMS キーの過去の使用状況を確認する

待機期間について

KMS キーを削除することは、破壊的でリスクを伴うため、AWS KMS で待機期間を 7 ~ 30 日に設定する必要があります。デフォルトの待機時間は、30 日です。

ただし、実際の待機期間は、スケジュールした待機期間よりも最大 24 時間長くなる場合があります。KMS キーが削除される実際の日時を取得するには、 DescribeKeyオペレーションを使用します。または、AWS KMS コンソール、KMS キーの詳細ページ、[General configuration] (一般的な設定) セクションで、スケジュールされた削除の日付を参照してください。必ずタイムゾーンをメモしておきます。

削除の待機期間中は、KMS キーステータスおよびキーの状態が削除保留中になります。

- 削除保留中の KMS キーを暗号化オペレーションで使用することはできません。
- AWS KMS は削除保留中の KMS キーのキーマテリアルをローテーションしません。

待機期間終了後、AWS KMS は KMS キー、そのエイリアス、関連するすべての AWS KMS メタデータを削除します。

KMS キーの削除をスケジュールしても、KMS キーで暗号化されたデータキーに、ただちに影響しない場合もあります。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

待機期間を設定することにより、KMS キーが不要であり、今後も使用することがないことを確認できます。待機期間中にユーザーまたはアプリケーションが KMS キーを使用しようとした場合に警告するように Amazon CloudWatch アラームを設定できます。KMS キーを復元するには、待機期間の終了前にキーの削除をキャンセルします。待機期間の終了後は、キーの削除はキャンセルできず、AWS KMS は KMS キーを削除します。

非対称 KMS キーの削除

認可されたユーザーは、対称または非対称 KMS キーを削除できます。これらの KMS キーの削除をスケジュールする手順は、どちらの種類のキーも同じです。ただし、非対称 KMS キーのパブリックキーは AWS KMS の外部でダウンロードして使用できるため、特に暗号化に使用される非対称 KMS キー (キーの使用法が ENCRYPT_DECRYPT) では、オペレーションに重大な追加リスクが生じます。

- KMS キーの削除をスケジュールすると、KMS キーのキーステータスが削除保留中に変わり、KMS キーを暗号化オペレーションで使用できなくなります。ただし、削除をスケジュールしても、AWS KMS の外部にあるパブリックキーには影響しません。パブリックキーを持つユーザーは、引き続きそのパブリックキーを使ってメッセージを暗号化できます。キーの状態が変更されたという通知は受信しません。削除がキャンセルされない限り、パブリックキーで作成された暗号文は復号できません。
- 削除保留中の KMS キーを使用する試みを検出するアラーム、ログ、その他の戦略では、AWS KMS の外部でのパブリックキーの使用は検出できません。
- KMS キーが削除されると、その KMS キーに関連するすべての AWS KMS アクションは失敗します。ただし、パブリックキーを持つユーザーは、引き続きそのパブリックキーを使ってメッセージを暗号化できます。これらの暗号文は復号できません。

キーの使用方法が である非対称 KMS キーを削除する必要がある場合はENCRYPT_DECRYPT、CloudTrail ログエントリを使用して、パブリックキーがダウンロードおよび共有されているかどうかを確認します。完了している場合は、パブリックキーが AWS KMS の外部で使用されていないことを確認します。次に、削除するのではなく、KMS キーを無効にすることを検討します。

非対称 KMS キーの削除によって生じるリスクは、インポートされたキーマテリアルを含む非対称 KMS キーであれば軽減されます。　詳細については、「キーマテリアルがインポートされた KMS キーの削除」を参照してください。

マルチリージョンキーを削除する

許可されているユーザーは、マルチリージョンのプライマリキーとレプリカキーの削除をスケジュールできます。ただし、AWS KMS では、レプリカキーを持つマルチリージョンのプライマリキーは削除されません。また、プライマリキーが存在する限り、削除されたマルチリージョンのレプリカキーを再作成することもできます。詳細については、「マルチリージョンキーを削除する」を参照してください。

インポートされたキーマテリアルを含む KMS キーの削除

承認されたユーザーは、インポートされたキーマテリアルを含む KMS キーの削除をスケジュールできます。このアクションにより、KMS キー、それのキーマテリアル、および KMS キーに関連するすべてのメタデータが完全に削除されます。

キーマテリアルのコピーがある場合でも、削除された対称暗号化キーの暗号文をインポートされたキーマテリアルで復号できる新しい対称暗号化 KMS キーを作成することはできません。ただし、キーマテリアルがあれば、インポートされたキーマテリアルを持つ非対称 KMS キーまたは HMAC KMS キーを効果的に再作成できます。詳細については、「キーマテリアルがインポートされた KMS キーの削除」を参照してください。

キー削除へのアクセスを制御する

IAM ポリシーを使用して AWS KMS のアクセス許可を許可している場合、AWS 管理者アクセス (`"Action": "*"`) または AWS KMS フルアクセス (`"Action": "kms:*"`) を持つ IAM アイデンティティは、KMS キーの削除をスケジュールおよびキャンセルすることをすでに許可されています。キー管理者に、キーポリシーでの、キー削除のスケジュールおよびキャンセルを許可するには、AWS KMS コンソールまたは AWS KMS API を使用します。

通常、キーの削除をスケジュールおよびキャンセルするための、アクセス許可を持っているのは、キー管理者のみです。ただし、これらのアクセス許可は、`kms:ScheduleKeyDeletion` と `kms:CancelKeyDeletion` のアクセス許可をキーポリシーまたは IAM ポリシーに追加すれば、他の IAM ID に付与することができます。`kms:ScheduleKeyDeletionPendingWindowInDays` 条件キーを使用して、プリンシパルがScheduleKeyDeletionリクエストの `PendingWindowInDays`パラメータで指定できる値をさらに制限することもできます。

キー管理者に、キーの削除のスケジュールおよびキャンセルを許可する (コンソール)

キー管理者に、キーの削除をスケジュールおよびキャンセルするためのアクセス許可を付与するには。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. アクセス許可を変更する KMS キーのエイリアスまたはキー ID を選択します。
5. [Key policy] (キーポリシー) タブを選択します。
6. 次のステップは、キーポリシーの既定のビューとポリシービューで異なります。既定のビューは、既定のコンソールキーポリシーを使用している場合のみ、使用できます。使用していない場合は、ポリシービューのみを使用できます。

   既定のビューを使用できる場合は、[Key policy] (キーポリシー) タブに [Switch to policy view] (ポリシービューに切り替え) または [Switch to default view] (既定のビューに切り替え) のいずれかのボタンが表示されます。

   - 既定のビュー:
     - [Key deletion] (キーの削除) で、[Allow key administrators to delete this key] (キー管理者にこのキーの削除を許可する) を選択します。
   - ポリシービュー:
     a. [編集] を選択します。
     b. キー管理者向けのポリシーステートメントで、`Action` 要素に `kms:ScheduleKeyDeletion` と `kms:CancelKeyDeletion` のアクセス許可を追加します。

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
```

```
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

c. [変更の保存] をクリックします。

キー管理者に、キーの削除をスケジュールおよびキャンセルするためのアクセスを許可する (AWS CLI)

AWS Command Line Interface を使用すると、キーの削除をスケジュールおよびキャンセルするアクセス許可を追加できます。

キーの削除をスケジュールおよびキャンセルするアクセス許可を追加するには

1. aws kms get-key-policy コマンドを使用して、既存のキーポリシーを取得し、ポリシードキュメントをファイルに保存します。
2. 任意のテキストエディタでポリシードキュメントを開きます。キー管理者向けのポリシーステートメントで、kms:ScheduleKeyDeletion と kms:CancelKeyDeletion のアクセス許可を追加します。次の例は、これらの 2 つのアクセス許可を持つポリシーステートメントを示しています。

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
```

```
  ],
  "Resource": "*"
}
```

3. aws kms put-key-policy コマンドを使用して、キーポリシーを KMS キーに適用します。

キーの削除のスケジュールとキャンセル

次の手順は、AWS Management Console、AWS CLI、AWS SDK for Java を使用して、AWS KMS で 単一リージョン AWS KMS keys のキーの削除をスケジュールおよびキャンセルする方法を説明しています。

マルチリージョンキーの削除のスケジューリングについては、マルチリージョンキーを削除する を参照してください。

> **Warning**
>
> KMS キーを削除することは破壊的であり、リスクを伴います。KMS キーが不要であり、今後も使用しないことが確実である場合にのみ実行してください。不明な場合は、削除するのではなく KMS キーを無効化するべきです。

KMS キーを削除する前に、削除するための許可を取得する必要があります。これらのアクセス許可をキー管理者に付与する方法の詳細は、「キー削除へのアクセスを制御する」を参照してください。kms:ScheduleKeyDeletionPendingWindowInDays条件キーを使用して、最小待機期間を設定するなど、待機期間をさらに制限することもできます。

AWS KMS は、KMS キーの削除をスケジュールするとき、および KMS キーが実際に削除されたときに、AWS CloudTrail ログにエントリを記録します。

キー削除のスケジュールとキャンセル (コンソール)

AWS Management Console では、複数の KMS キーの削除を一度にスケジューリングおよびキャンセルできます。

キーの削除をスケジュールするには

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。

2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。

   AWS マネージドキー または AWS 所有のキー の削除をスケジュールすることはできません。
4. 削除する KMS キーの横にあるチェックボックスをオンにします。
5. [Key actions] (キーのアクション)、[Schedule key deletion] (キーの削除をスケジュール) の順に選択します。
6. 待機中に、警告、および削除のキャンセルに関する情報を読み、検討してください。削除をキャンセルする場合はページ下部の [Cancel (キャンセル)] を選択します。
7. [Waiting period (in days)] (待機期間 (日数)) に、日数として 7 ~ 30 の値を入力します。
8. 削除する KMS キーを確認します。
9. [Confirm you want to schedule this key for deletion in ***<number of days>*** days] (このキーを <日数> 日後に削除するスケジュールを確定する) の横にあるチェックボックスをオンにします。
10. [Schedule deletion] (削除をスケジュールする) を選択します。

KMS キーのステータスが [Pending deletion] (削除保留中) に変わります。

キーの削除をキャンセルするには

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 復元する KMS キーの横にあるチェックボックスをオンにします。
5. [Key actions] (キーのアクション)、[Cancel key deletion] (キーの削除をキャンセル) の順に選択します。

KMS キーのステータスが [Pending deletion] (削除保留中) から [Disabled] (無効) に変わります。KMS キーを使用するには、有効化する必要があります。

キーの削除のスケジュールとキャンセル (AWS CLI)

次の例のように、`aws kms schedule-key-deletion` コマンドを使用して、カスタマーマネージドキーからキーの削除をスケジュールします。

AWS マネージドキー または AWS 所有のキー の削除をスケジュールすることはできません。

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --
pending-window-in-days 10
```

正しく使用すると、AWS CLI は以下の例に示すような出力を返します。

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": 1598304792.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 10
}
```

次の例のように、aws kms cancel-key-deletion コマンドを使用して AWS CLI からキーの削除をキャンセルします。

```
$ aws kms cancel-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

正しく使用すると、AWS CLI は以下の例に示すような出力を返します。

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

KMS キーのステータスが [Pending Deletion] (削除保留中) から [Disabled] (無効) に変わります。KMS キーを使用するには、有効化する必要があります。

キーの削除のスケジュールとキャンセル (AWS SDK for Java)

次の例では、AWS SDK for Java で KMS キーの削除をスケジュールする方法を示します。この例では、事前に AWSKMSClient を kms としてインスタンス化している必要があります。

```
String KeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

int PendingWindowInDays = 10;

ScheduleKeyDeletionRequest scheduleKeyDeletionRequest =
new
 ScheduleKeyDeletionRequest().withKeyId(KeyId).withPendingWindowInDays(PendingWindowInDays);
```

```
kms.scheduleKeyDeletion(scheduleKeyDeletionRequest);
```

次の例では、AWS SDK for Java でキーの削除をキャンセルする方法を示します。この例では、事前に AWSKMSClient を kms としてインスタンス化している必要があります。

```
String KeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CancelKeyDeletionRequest cancelKeyDeletionRequest =
new CancelKeyDeletionRequest().withKeyId(KeyId);
kms.cancelKeyDeletion(cancelKeyDeletionRequest);
```

KMS キーのステータスが [Pending Deletion] (削除保留中) から [Disabled] (無効) に変わります。KMS キーを使用するには、有効化する必要があります。

削除保留中の KMS キーの使用を検出するアラームの作成

AWS CloudTrail、Amazon CloudWatch Logs、Amazon Simple Notification Service (Amazon SNS) の機能を組み合わせて、アカウント内の誰かが削除保留中の KMS キーを使用しようとしたときに通知する Amazon CloudWatch アラームを作成できます。この通知を受け取った場合は、KMS キーの削除をキャンセルして、削除する決定を検討し直す必要があります。

次の手順では、*Key ARN* is pending deletion「」エラーメッセージが CloudTrail ログファイルに書き込まれるたびに通知するアラームを作成します。このエラーメッセージは、暗号化オペレーションで、ユーザーまたはアプリケーションが KMS キーの使用を試みたことを示します。この通知はエラーメッセージにリンクされているため、ListKeys、CancelKeyDeletion、PutKeyPolicy などの削除保留中の KMS キーで許可される API オペレーションを使用するときにはトリガーされません。このエラーメッセージを返す AWS KMS API オペレーションのリストを表示するには、「AWS KMS キーのキーステータス」を参照してください。

受信した E メール通知には、KMS キーや暗号化オペレーションは表示されません。その情報は CloudTrail ログで確認できます。その代わりに、アラームの状態が [OK] から [アラーム] に変わったことが E メールで報告されます。アラームと状態の変更の詳細については CloudWatch、「Amazon CloudWatch ユーザーガイド」の「Amazon アラームの使用 CloudWatch 」を参照してください。

> **Warning**
>
> この Amazon CloudWatch アラームは、 の外部での非対称 KMS キーのパブリックキーの使用を検出できませんAWS KMS。公開キーの暗号化に使用される非対称 KMS キーを削除する

際の特別なリスク (復号できない暗号テキストの作成など) の詳細については、非対称 KMS キーの削除 を参照してください。

トピック

- CloudWatch アラームの要件
- CloudWatch アラームの作成

CloudWatch アラームの要件

CloudWatch アラームを作成する前に、 AWS CloudTrail証跡を作成し、 CloudTrail ログファイルを Amazon CloudWatch Logs に配信 CloudTrail するように を設定する必要があります。アラームの通知用の Amazon SNS トピックも必要です。

- CloudTrail 証跡を作成します。

  CloudTrail アカウントを作成するAWS アカウントと、 は で自動的に有効になります。ただし、AWS KMS のイベントなど、アカウントのイベントの継続的な記録を取得するには、証跡を作成します。

- ログファイル CloudWatch Logs を配信 CloudTrail するように を設定します。

  CloudTrail ログファイルの CloudWatch Logs への配信を設定します。これにより、 CloudWatch Logs は削除保留中の KMS キーを使用しようとする AWS KMS API リクエストのログをモニタリングできます。

- Amazon SNS トピックを作成します。

  アラームがトリガーされると、Amazon Simple Notification Service (Amazon SNS) トピックの E メールアドレスに E メールメッセージを送信して通知します。

CloudWatch アラームの作成

この手順では、保留中の削除例外のインスタンスを検索する CloudWatch ロググループメトリクスフィルターを作成します。次に、ロググループメトリクスに基づいて CloudWatch アラームを作成します。ロググループのメトリクスフィルターの詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「フィルターを使用したログイベントからのメトリクスの作成」を参照してください。

1. CloudTrail ログを解析する CloudWatch メトリクスフィルターを作成します。

   以下の必須値を使用して、「ロググループのメトリクススフィルターを作成する」の手順に従ってください。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
フィルターパターン	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
メトリクス値	1

2. ステップ 1 で作成したメトリクスフィルターに基づいて CloudWatch アラームを作成します。

   「以下の必須値を使用したロググループメトリクスフィルターに基づく CloudWatch アラームの作成」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスフィルター	ステップ 1 で作成したメトリクスフィルターの名前。
しきい値タイプ	静的
条件	######が 1 より大きい場合は必ず
アラームへのデータポイント	1 のうち 1
欠損データ処理	欠損データを良好 (しきい値に違反していない) として扱う

この手順を完了すると、新しい CloudWatchアラームが ALARM状態になるたびに通知が届きます。このアラームの通知を受信した場合は、データの暗号化または復号化に削除が予定されている KMS

キーがまだ必要であることを意味します。その場合は、KMS キーの削除をキャンセルし、削除する決定を検討し直す必要があります。

KMS キーの過去の使用状況を確認する

KMS キーを削除する前に、そのキーで暗号化された暗号化テキストの数を確認する必要がある場合があります。AWS KMS は、この情報や暗号化テキストを保存しません。KMS キーの過去の使用状況を把握することで、後で必要になるかどうかを判断できる場合があります。このトピックでは、KMS キーの過去の使用状況の確認に役立つ複数の戦略を示します。

> **Warning**
>
> 過去と実際の使用方法を決定するためのこれらの戦略は、AWS ユーザーと AWS KMS オペレーションに対してのみ有効です。AWS KMS の外部で非対称 KMS キーの公開キーの使用を検出することはできません。公開キーの暗号化に使用される非対称 KMS キーを削除する際の特別なリスク (復号できない暗号テキストの作成など) の詳細については、非対称 KMS キーの削除 を参照してください。

トピック

- KMS キーのアクセス許可を確認し、潜在的な使用の範囲を判断する
- AWS CloudTrail ログを確認して実際の使用状況を判断する

KMS キーのアクセス許可を確認し、潜在的な使用の範囲を判断する

現在、KMS キーにアクセスできるユーザーやアプリケーションを明らかにすることで、KMS キーが使用されている範囲や、今後必要かどうかを判断できる場合があります。現在、KMS キーにアクセスできるユーザーやアプリケーションを明らかにする方法については、AWS KMS keys へのアクセスを特定する を参照してください。

AWS CloudTrail ログを確認して実際の使用状況を判断する

KMS キーの使用履歴を使用して、特定の KMS キーで暗号化された暗号化テキストがあるかどうかを判断できます。

すべての AWS KMS API アクティビティは AWS CloudTrail ログファイルに記録されます。KMS キーが配置されているリージョンに CloudTrail 証跡を作成した場合は、 CloudTrail ログファイルを

調べて、特定の KMS キーのすべての AWS KMS API アクティビティの履歴を表示できます。証跡がない場合でも、最近のイベントを CloudTrail イベント履歴に表示できます。が AWS KMSを使用する方法の詳細については CloudTrail、「」を参照してくださいAWS KMS による AWS CloudTrail API コールのログ記録。

次の例は、KMS キーを使用して Amazon Simple Storage Service (Amazon S3) に保存されているオブジェクトを保護するときに生成される CloudTrail ログエントリを示しています。この例では、KMS キー (SSE-KMS) を使用したサーバー側の暗号化を使用してデータを保護し、オブジェクトを Amazon S3 にアップロードします。SSE-KMS を使用して Amazon S3 にオブジェクトをアップロードする場合は、オブジェクトの保護に使用する KMS キーを指定します。Amazon S3 は AWS KMSGenerateDataKeyオペレーションを使用してオブジェクトの一意のデータキーをリクエストし、このリクエストイベントは次のようなエントリ CloudTrail でログインします。


```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:18Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
```

```
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"},
    "keySpec": "AES_256",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "cea04450-5817-11e5-85aa-97ce46071236",
  "eventID": "80721262-21a5-49b9-8b63-28740e7ce9c9",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

後で Amazon S3 からこのオブジェクトをダウンロードすると、Amazon S3 は AWS KMS に Decrypt リクエストを送信し、指定された KMS キーを使用してオブジェクトのデータキーを復号します。これを行うと、 CloudTrail ログファイルに次のようなエントリが含まれます。

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
```

```
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:39Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"}},
  "responseElements": null,
  "requestID": "db750745-5817-11e5-93a6-5b87e27d91a0",
  "eventID": "ae551b19-8a09-4cfc-a249-205ddba330e3",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

AWS KMS API アクティビティは、すべて CloudTrail によって記録されます。これらのログエントリを評価すると、特定の KMS キーの過去の使用状況を明らかにし、その KMS キーを削除する必要があるかどうかを判断できる場合があります。

AWS KMS API アクティビティが CloudTrail ログファイルにどのように表示されるかの例については、「」を参照してくださいAWS KMS による AWS CloudTrail API コールのログ記録。の詳細については、「 AWS CloudTrailユーザーガイド CloudTrail 」を参照してください。

AWS KMS キーのキーステータス

AWS KMS key は必ずキーステータスを持っています。KMS キーとその環境に対するオペレーションは、一時的に、または別のオペレーションによって変更されるまで、そのキーステータスを変更できます。

このセクションの表は、キーステータスが AWS KMS API オペレーションへの呼び出しにどのように影響するかを示しています。キーステータスの結果として、KMS キーでのオペレーションは成

功 (#)、失敗 (X)、または特定の条件でのみ成功する (?) と予想されます。多くの場合、この結果は KMS キーのインポートされたキーマテリアルによって異なります。

この表には、既存の KMS キーを使用する API オペレーションのみが含まれています。CreateKey やなどの他のオペレーションListKeysは省略されます。

トピック

- キーステータスと KMS キーの種類
- キーステータスの表

キーステータスと KMS キーの種類

KMS キーの種類によって、それが持つキーステータスが決まります。

- すべての KMS キーは、`Enabled`、`Disabled`、および `PendingDeletion` 状態になり得ます。
- ほとんどの KMS キーは `Enabled` ステータスで作成されます。インポートされたキーマテリアルを持つキーは `PendingImport` ステータスで作成されます。
- `PendingImport` ステータスは、インポートされたキーマテリアルを持つ KMS キーにのみ適用されます。
- `Unavailable` ステータスは、カスタムキーストア内の KMS キーにのみ適用されます。カスタムキーストアがその AWS CloudHSM クラスターから意図的に切断された場合、AWS CloudHSM キーストアの KMS キーは `Unavailable` です。カスタムキーストアがその外部キーストアプロキシから意図的に切断された場合、外部キーストアの KMS キーは `Unavailable` です。使用できない KMS キーを表示および管理することはできますが、暗号化オペレーションで使用することはできません。

  カスタムキーストア内の KMS キーのキーの状態は、そのバッキングキーが変更されても影響を受けません。AWS CloudHSM キーストアの KMSキーは、AWS CloudHSM クラスター内の関連するキーマテリアルが変更されても影響を受けません。外部キーストアの KMS キーは、外部のキーマネージャーの外部キーが変更されても影響を受けません。バッキングキーが無効化されるか削除されると、KMS キーの状態は変わりませんが、KMS キーを使用した暗号化オペレーションは失敗します。
- `Creating`、`Updating`、`PendingReplicaDeletion` のキーステータスは、マルチリージョンキーにのみ適用されます。
  - マルチリージョンのレプリカキーは、作成中の一時的な `Creating` キーステータスです。ReplicateKey オペレーションが完了すると、このプロセスがまだ進行中の可能性がありま

す。レプリケートプロセスが完了すると、レプリカキーは Enabled または PendingImport ステータスになります。

- プライマリリージョンの更新中、マルチリージョンキーは一時的に Updating キーステータスになります。UpdatePrimaryRegion オペレーションが完了すると、このプロセスがまだ進行中の可能性があります。更新プロセスが完了すると、プライマリキーとレプリカキーは、Enabled キーステータスを再開します。
- レプリカキーを持つマルチリージョンのプライマリキーの削除をスケジュールすると、プライマリキーはそのレプリカキーがすべて削除されるまで、PendingReplicaDeletion ステータスを保持します。その後、キーステータスが PendingDeletion に変わります。詳細については、「マルチリージョンキーを削除する」を参照してください。

キーステータスの表

次の表に、KMS キーのキーステータスが、AWS KMS オペレーションにどのような影響を与えるかを示します。

番号付きの脚注の説明 ([n]) は、このトピックの最後にあります。

> **Note**
>
> この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
CancelKeyDeletion	✗ [4]	✗ [4]	✓	✗ [4]	✗ [4], [13]	✗ [4]	✗ [4]
CreateAlias	✓	✓	✗	✓	✓	✓	✓

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
			[3]				
CreateGrant	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	✓	⊗ [14]	✓
Decrypt	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
DeleteAlias	✓	✓	✓	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓ (影響なし)	該当なし	⊗ [14]	⊗ [15]
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
DisableKeyRotation	 [7]	 [1] または [7]	 [3] または [7]	 [6]	 [7]	 [14]	 [7]
EnableKey			 [3]	 [5]	 [12]	 [14]	 [15]
EnableKeyRotation	 [7]	 [1] または [7]	 [3] または [7]	 [6]	 [7]	 [14]	 [7]
暗号化		 [1]	 [2] または [3]	 [5]	 [11]	 [14]	
GenerateDataKey		 [1]	 [2] または [3]	 [5]	 [11]	 [14]	
GenerateDataKeyPair		 [1]	 [2] または [3]	 [5]	 [11]	[14]	

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
GenerateDataKeyPairWithoutPlaintext	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateMac	✓	⊗ [1]	⊗ [2] または [3]	該当なし	該当なし	⊗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓
GetKeyRotationStatus	? [7]	? [7]	? [7]	⊗ [6]	⊗ [7]	? [7]	? [7]
GetParametersForImport	? [9]	? [9]	⊗ [8] または [9]	✓	⊗ [9]	⊗ [14]	⊗ [15]

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
GetPublicKey	✓	⊗ [1]	⊗ [2] または [3]	該当なし	該当なし	⊗ [14]	✓
ImportKeyMaterial	? [9]	? [9]	⊗ [8] または [9]	✓	⊗ [9]	⊗ [14]	✓
ListAliases	✓	✓	✓	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓	✓	✓	✓
ListResourceTags	✓	✓	✓	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
ReplicateKey	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	該当なし	⊗ [14]	⊗ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	⊗ [3]	✓	✓	✓	⊗ [15]
Sign	✓	⊗ [1]	⊗ [2] または [3]	該当なし	該当なし	⊗ [14]	✓
TagResource	✓	✓	⊗ [3]	✓	✓	✓	✓
UntagResource	✓	✓	⊗ [3]	✓	✓	✓	✓

API	有効	無効	削除保留中 レプリカの削除保留中	インポートの保留中	使用不可	[作成中]	[更新中]
UpdateAlias	✓	✓	? [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊗ [1]	⊗ [2] または [3]	⊗ [5]	該当なし	⊗ [14]	✓
検証	✓	⊗ [1]	⊗ [2] または [3]	該当なし	該当なし	⊗ [14]	✓
VerifyMac	✓	⊗ [1]	⊗ [2] または [3]	該当なし	該当なし	⊗ [14]	✓

テーブルの詳細

- [1] `DisabledException: <key ARN> is disabled.`
- [2] `DisabledException: <key ARN> is pending deletion (or pending replica deletion).`

- [3] KMSInvalidStateException: *<key ARN>* is pending deletion (or pending replica deletion).
- [4] KMSInvalidStateException: *<key ARN>* is not pending deletion (or pending replica deletion).
- [5] KMSInvalidStateException: *<key ARN>* is pending import.
- [6] UnsupportedOperationException: *<key ARN>* origin is EXTERNAL which is not valid for this operation.
- [7] KMS キーがインポートされたキーマテリアルを持つ場合、または KMS キーがカスタムキーストアにある場合: UnsupportedOperationException。
- [8] KMS キーがインポートされたキーマテリアルを持つ場合: KMSInvalidStateException
- [9] KMS キーがインポートされたキーマテリアルを持たない場合、または持てない場合: UnsupportedOperationException。
- [10] ソース KMS キーが削除保留中の場合、コマンドは成功します。送信先の KMS キーが削除保留中の場合、コマンドは次のエラーで失敗します: KMSInvalidStateException : *<key ARN>* is pending deletion.
- [11] KMSInvalidStateException: *<key ARN>* is unavailable. 使用不可能な KMS キーでこのオペレーションを実行することはできません。
- [12] オペレーションは成功しますが、KMS キーのキーステータスは、使用可能になるまで変更されません。
- [13] カスタムキーストアの KMS キーが削除保留中の場合、KMS キーが使用不可能になっても、そのキーステータスは PendingDeletion のままになります。これにより、待機期間中はいつでも KMS キーの削除をキャンセルできます。
- [14] KMSInvalidStateException: *<key ARN>* is creating. AWS KMS は、マルチリージョンキー (ReplicateKey) のレプリケーション中にこの例外をスローします。
- [15] KMSInvalidStateException: *<key ARN>* is updating. AWS KMS は、マルチリージョンキー (UpdatePrimaryRegion) のプライマリリージョンの更新中にこの例外をスローします。

AWS KMS の認証とアクセスコントロール

AWS KMS を使用する場合は、AWS によってリクエストの認証に使用される認証情報が必要です。認証情報には、AWS リソースへのアクセス、AWS KMS keys およびエイリアスを含める必要があります。AWS プリンシパルは、そのアクセス許可が明示的に提供され、拒否されていない限り、KMS キーに対して何もアクセス許可はありません。KMS キーを使用または管理するための暗黙的または自動的な権限はありません。

AWS KMS リソースへのアクセスを管理する主な手段は、ポリシーです。ポリシーは、どのプリンシパルがどのリソースにアクセスできるかを記述するドキュメントです。IAM アイデンティティにアタッチされたポリシーはアイデンティティベースのポリシー (または IAM ポリシー) と呼ばれ、他の種類のリソースにアタッチされたポリシーはリソースポリシーと呼ばれます。KMS キーの AWS KMS リソースポリシーはキーポリシーと呼ばれます。すべての KMS キーにはキーポリシーがあります。

AWS KMS エイリアスへのアクセスを管理するには、IAM ポリシーを使用します。プリンシパルにエイリアスの作成を許可するには、IAM ポリシーのエイリアスに対する許可と、キーポリシーのキーに対する許可を提供する必要があります。詳細については、「エイリアスへのアクセスの制御」を参照してください。

KMS キーへのアクセスを制御するには、次のポリシーメカニズムを使用できます。

- キーポリシー — すべての KMS キーにはキーポリシーがあります。これは、KMS キーへのアクセスを制御するための主要メカニズムです。キーポリシーを単独で使用してアクセスを制御できます。この場合、KMS キーへのアクセスの完全な範囲が 1 つのドキュメント (キーポリシー) で定義されます。キーポリシーの使用の詳細については、「キーポリシー」を参照してください。
- IAM ポリシー — IAM ポリシーをキーポリシーと組み合わせて使用し、KMS キーへのアクセスを制御できるように許可を付与します。この方法でアクセスを制御すると、IAM の IAM ID に対するすべてのアクセス許可を管理できます。IAM ポリシーを使用して KMS キーへのアクセスを許可するには、キーポリシーで明示的に許可する必要があります。IAM ポリシーの使用の詳細については、「IAM ポリシー」を参照してください。
- グラント — グラントをキーポリシーおよび IAM ポリシーと組み合わせて使用し、KMS キーへのアクセスを許可します。この方法でアクセスを制御すると、キーポリシーで KMS キーへのアクセスを許可し、アイデンティティが他のユーザーにアクセスを委任できるようになります。グラントの使用の詳細については、「AWS KMS でのグラント」を参照してください。

KMS キーは、作成された AWS アカウントに属します。ただし、アクセスグラントがキーポリシー、IAM ポリシー、またはグラントで明示的に提供されていない限り、アイデンティティやプリンシパルは、AWS アカウントのルートユーザーも含め、KMS キーを使用または管理するアクセス許可はありません。KMS キーを作成する IAM アイデンティティは、キー所有者とはみなされないため、作成した KMS キーを使用または管理するためのアクセス許可が自動的に付与されません。他のアイデンティティと同様に、キー作成者は、キーポリシー、IAM ポリシー、またはグラントを使用してアクセス許可を取得する必要があります。ただし、`kms:CreateKey` アクセス許可を持つアイデンティティは、初期キーポリシーを設定し、キーを使用または管理するためのアクセス許可を自分自身に付与できます。

次のトピックでは、AWS Identity and Access Management (IAM) と AWS KMS アクセス許可を使用して、リソースにアクセスできるユーザーを制御することでリソースをセキュア化する方法の詳細について説明します。

トピック

AWS KMS アクセスコントロールの概念

AWS KMS でのアクセスコントロールに関する説明で使用した概念について説明します。

トピック

- 認証
- 認証
- アイデンティティによる認証
- ポリシーを使用したアクセス権の管理
- AWS KMS リソース

認証

認証とは、アイデンティティを検証するプロセスです。AWS KMS にリクエストを送信するには、AWS 認証情報を使用して AWS にサインインする必要があります。

認証

認証により、AWS KMS リソースの作成、管理、使用といったリクエストを送信する権限が与えられます。例えば、暗号化オペレーションで KMS キーを使用するには、権限が必要です。

キーポリシー、IAM ポリシー、グラントを使用して、AWS KMS リソースへのアクセスを制御します。すべての KMS キーにはキーポリシーが必要です。キーポリシーで許可されている場合は、IAM ポリシーとグラントを使用して、プリンシパルに KMS キーへのアクセス権を付与することもできます。権限を絞り込むために、条件キーを使用して、リクエストまたはリソースが指定した条件を満たす場合に限り、アクセスを許可または拒否できます。また、他の AWS アカウント で、信頼するプリンシパルへのアクセスを許可できます。

アイデンティティによる認証

認証とは、アイデンティティ認証情報を使用して AWSにサインインする方法です。ユーザーは、AWS アカウントのルートユーザーもしくは IAM ユーザーとして、または IAM ロールを引き受けることによって、認証を受ける (AWS にサインインする) 必要があります。

ID ソースから提供された認証情報を使用して、フェデレーティッドアイデンティティとして AWS にサインインできます。 AWS IAM Identity Centerフェデレーティッドアイデンティティの例としては、(IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報などがあります。フェデレーティッドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーションを使用して AWSにアクセスする場合、間接的にロールを引き受けることになります。

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

プログラムで AWS にアクセスする場合、AWS は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) を提供し、認証情報でリクエストに暗号で署名します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。リクエストに署名する推奨方法の使用については、『IAM ユーザーガイド』の「AWS API リクエストの署名」を参照してください。

使用する認証方法を問わず、追加のセキュリティ情報の提供が求められる場合もあります。例えば、AWS では、アカウントのセキュリティ強化のために多要素認証 (MFA) の使用をお勧めしています。詳細については、「AWS IAM Identity Center ユーザーガイド」の「Multi-factor authentication (多要素認証)」 および「IAM ユーザーガイド」の「AWS での多要素認証 (MFA) の使用」を参照してください。

AWS アカウントのルートユーザー

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

フェデレーティッド ID

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

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

アクセスを一元管理する場合は、AWS IAM Identity Center を使用することをお勧めします。IAM アイデンティティセンター でユーザーとグループを作成するか、すべての AWS アカウント とアプリ

ケーションで使用するために、独自の ID ソースで一連のユーザーとグループに接続して同期することもできます。IAM アイデンティティセンター の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM アイデンティティセンター?」(IAM アイデンティティセンター とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、1 人のユーザーまたは 1 つのアプリケーションに対して特定の権限を持つ AWS アカウント 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

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

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

IAM ロール

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

一時的な認証情報を持った IAM ロールは、以下の状況で役立ちます。

- フェデレーションユーザーユーザーアクセス – フェデレーションアイデンティティに権限を割り当てるには、ロールを作成してそのロールの権限を定義します。フェデレーティッドアイデンティ

ティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている権限が付与されます。フェデレーションの詳細については、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー向けロールの作成」を参照してください。IAM アイデンティティセンターを使用する場合、権限セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。権限セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「権限セット」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の AWS のサービスでは、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス - 一部の AWS のサービスでは、他の AWS のサービスの機能を使用します。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの権限、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、AWS のサービスを呼び出すプリンシパルの権限を、AWS のサービスのリクエストと合わせて使用し、ダウンストリームのサービスに対してリクエストを行います。FAS リクエストは、サービスが、完了するために他の AWS のサービス またはリソースとのやりとりを必要とするリクエストを受け取ったときにのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスにアクセス許可を委任するロールの作成」を参照してください。

- サービスリンクロール - サービスリンクロールは、AWS のサービス にリンクされた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。
- Amazon EC2 で実行されているアプリケーション - EC2 インスタンスで実行され、AWS CLI または AWS API 要求を行っているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスに添付されたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用してアクセス許可を付与する」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、「IAM ユーザーガイド」の「(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?」を参照してください。

ポリシーを使用したアクセス権の管理

AWS でアクセス権を管理するには、ポリシーを作成して AWS アイデンティティまたはリソースにアタッチします。ポリシーは AWS のオブジェクトであり、アイデンティティやリソースに関連付けて、これらの権限を定義します。AWS は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うと、これらのポリシーを評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWSに保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は AWSJSON ポリシーを使用して、だれが何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

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

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

アイデンティティベースポリシー

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

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

リソースベースのポリシー

AWS KMS キーポリシーは、KMS キーへのアクセスを制御するリソースベースのポリシーです。すべての KMS キーにはキーポリシーが必要です。他の認可メカニズムを使用して、KMS キーへのアクセスを許可できますが、キーポリシーで許可されている場合に限ります (キーポリシーで明示的に許可されていなくても、IAM ポリシーを使用して KMS キーへのアクセスを「拒否」できます)。

リソースベースのポリシーとは、JSON ポリシードキュメントです。KMS キーなどのリソースにアタッチして、特定のリソースへのアクセスを制御します。リソースベースのポリシーは、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件を定義します。リソースベースのポリシーではリソースを指定しませんが、アカウント、ユーザー、ロール、フェデレーションユーザー、AWS のサービス などのプリンシパルを指定する必要があります。リソースベースのポリシーは、リソースを管理するそのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシー (`AWSKeyManagementServicePowerUser` マネージドポリシーなど) を使用することはできません。

アクセスコントロールリスト (ACL)

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

Simple Storage Service (Amazon S3)、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

AWS KMS では ACL はサポートされません。

他のポリシータイプ

AWS では、他の一般的ではないポリシータイプをサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- 権限の境界 - 権限の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる許可の上限を設定する高度な機能です。エンティティに権限の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとその権限の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、権限の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。権限の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティの権限の境界」を参照してください。
- サービスコントロールポリシー (SCP) - SCP は、AWS Organizations で組織や組織単位 (OU) の最大権限を指定する JSON ポリシーです。AWS Organizations は、顧客のビジネスが所有する複数の AWS アカウント をグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP はメンバーアカウントのエンティティに対する権限を制限します (各 AWS アカウントのルートユーザー など)。Organizations と SCP の詳細については、『AWS Organizations ユーザーガイド』の「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限の範囲は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」をご参照ください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、『IAM ユーザーガイド』の「Policy evaluation logic (ポリシーの評価ロジック)を参照してください。

AWS KMS リソース

AWS KMS では、プライマリリソースは AWS KMS key です。AWS KMS は、KMS キーのわかりやすい名前を提供する独立したリソースであるエイリアスもサポートします。一部の AWS KMS オペレーションでは、エイリアスを使用して KMS 許可キーを識別できます。

KMS キーまたはエイリアスの各インスタンスには、標準形式の一意の Amazon リソースネーム (ARN) があります。AWS KMS リソースの場合、AWS のサービス名は kms です。

- AWS KMS key

  ARN 形式:

  ```
  arn:AWS partition name:AWS service name:AWS #####:AWS ##### ID:key/key ID
  ```

  ARN の例:

  ```
  arn:aws:kms:us-
  west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
  ```

- エイリアス

  ARN 形式:

  ```
  arn:AWS partition name:AWS service name:AWS #####:AWS #####
  ID:alias/alias name
  ```

  ARN の例:

  ```
  arn:aws:kms:us-west-2:111122223333:alias/example-alias
  ```

AWS KMS には、AWS KMS リソースをオペレーションするための一連の API オペレーションが用意されています。AWS Management Console および AWS KMS API オペレーションでの KMS キーの識別の詳細については、キー識別子 (KeyId) を参照してください。AWS KMS オペレーションのリストについては、「AWS Key Management Service API リファレンス」を参照してください。

AWS KMS のキーポリシー

キーポリシーは、AWS KMS keyのためのリソースポリシーです。キーポリシーは、KMS キーへのアクセスを制御するための主要な方法です。すべての KMS キーには、厳密に 1 つのキーポリシーが必

要です。キーポリシーのステートメントでは、KMS キーの使用が許可されるユーザーとその使用方法を決定します。IAM ポリシーとグラントを使用して KMS キーへのアクセスを制御することもできますが、すべての KMS キーにはキーポリシーが必要です。

キーポリシー、IAM ポリシー、またはグラントで明示的に許可され、拒否されない限り、ルートユーザーやキー作成者を含むすべての AWS プリンシパルは、KMS キーに対するアクセス許可を持ちません。

キーポリシーで明示的に許可されていない限り、IAM ポリシーを使用して KMS キーへのアクセスを許可することはできません。キーポリシーからの許可がない場合、許可を許可する IAM ポリシーは効力を持ちません。(IAM ポリシーを使用して、キーポリシーからの許可を得ることなく KMS キーに対する許可を拒否することができます。) デフォルトのキーポリシーでは IAM ポリシーが有効になっています。キーポリシーで IAM ポリシーを有効にするには、AWS アカウント へのアクセスを許可し、IAM ポリシーを有効にする で説明されているポリシーステートメントを追加します。

グローバルな IAM ポリシーとは異なり、キーポリシーはリージョナルです。キーポリシーは、同じリージョン内の KMS キーへのアクセスのみを制御します。他のリージョンの KMS キーには影響しません。

トピック

キーポリシーを作成する

AWS KMS コンソールで、、、 などの AWS KMS API オペレーションを使用するか、 AWS CloudFormation テンプレートを使用してCreateKeyReplicateKeyPutKeyPolicy、キーポリシーを作成および管理できます。

AWS KMS コンソールで KMS キーを作成すると、コンソールのデフォルトのキーポリシーに基づいてキーポリシーを作成するためのステップがコンソールに表示されます。`CreateKey` または `ReplicateKey` API を使用するときにキーポリシーを指定しない場合、これらの API はプログラム的に作成されたキーのデフォルトキーポリシーを適用します。`PutKeyPolicy` API を使用するときは、キーポリシーを指定する必要があります。

各ポリシードキュメントには、1 つ以上のポリシーステートメントを指定できます。以下は、1 つのポリシーステートメントがある有効なキーポリシードキュメントの例です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Describe the policy statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/Alice"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeySpec": "SYMMETRIC_DEFAULT"
        }
      }
    }
  ]
}
```

トピック

- キーポリシー形式
- キーポリシー内の要素
- キーポリシーの例

キーポリシー形式

キーポリシードキュメントは、以下のルールに従う必要があります。

- 最大 32 キロバイト (32,768 バイト)
- キーポリシーステートメントの Sid 要素には、スペースを含めることができます。(IAM ポリシードキュメントの Sid 要素では、スペースが禁止されています。)

キーポリシードキュメントには、以下の文字のみを含めることができます。

- 印字可能な ASCII 文字

- 基本ラテンおよびラテン 1 補助文字セットの印字可能な文字
- タブ (\u0009)、ラインフィード (\u000A)、およびキャリッジリターン (\u000D) の特殊文字

キーポリシー内の要素

キーポリシードキュメントには、次の要素が必要です。

Version

キーポリシードキュメントのバージョンを指定します。バージョンを 2012-10-17 (最新バージョン) に設定します。

Statement

ポリシーステートメントを囲みます。キーポリシードキュメントには、少なくとも 1 つのステートメントが必要です。

各キーポリシーステートメントは、最大 6 個の要素で構成されます。Effect、Principal、Action、Resource の要素は必須です。

Sid

(オプション) ステートメント識別子 (Sid)。ステートメントを記述するために使用できる任意の文字列です。キーポリシーの Sid には、スペースを含めることができます。(IAM ポリシーの Sid 要素にスペースを含めることはできません。)

Effect

(必須) ポリシーステートメント内の許可を許容するか拒否するかを決定します。有効な値は Allow または Deny です。KMS キーへのアクセスを明示的に許可しない場合、アクセスは暗黙的に拒否されます。KMS キーへのアクセスを明示的に拒否することもできます。これは、別のポリシーがアクセスを許可している場合でも、ユーザーがアクセスできないようにするために行います。

Principal

(必須) プリンシパルは、ポリシーステートメントで指定されている許可を取得するアイデンティティです。キーポリシーのプリンシパルとして、AWS アカウント、IAM ユーザー、IAM ロール、および一部の AWS サービスを指定できます。IAM ユーザーグループは、どのポリシータイプにおいても有効なプリンシパルではありません。

"AWS": "*" などのアスタリスク値は、すべてのアカウントのすべての AWS ID を表します。

> **Important**
>
> 条件を使用してキーポリシーを制限しない限り、アクセス許可を付与するキーポリシーステートメントで、プリンシパルをアスタリスク (*) に設定しないでください。アスタリスクは、別のポリシーステートメントが明示的に拒否しない限り、すべての AWS アカウント のすべてのアイデンティティに、KMS キーを使用するアクセス許可を付与します。他の AWS アカウント のユーザーは、各自のアカウントに対応するアクセス権限があるときにはいつでも KMS キーを使用できます。

> **Note**
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

キーポリシーステートメントのプリンシパルが `arn:aws:iam::111122223333:root"` 形式の AWS アカウント プリンシパルである場合、ポリシーステートメントはいかなる IAM プリンシパルにも許可を付与しません。代わりに AWS アカウント 許可を付与し、IAM ポリシーを使用して、キーポリシーで指定された許可を委任します。(`arn:aws:iam::111122223333:root"` 形式のプリンシパルは、アカウント識別子に「root」が使用されていても AWS アカウントのルートユーザーを示すものではありません。ただし、アカウントプリンシパルは、アカウントのルートユーザーを含むアカウントとその管理者を表します。)

プリンシパルが別の AWS アカウント またはそのプリンシパルの場合、KMS キーとキーポリシーでリージョンのアカウントが有効になっている場合にのみ、アクセス許可が有効になります。デフォルトで有効になっていないリージョン (「オプトインリージョン」) については、「AWS 全般のリファレンス」の「AWS リージョン の管理」を参照してください。

別の AWS アカウント またはそのプリンシパルに KMS キーの使用を許可するには、キーポリシーと他のアカウントの IAM ポリシーで許可を提供する必要があります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

Action

(必須) 許可または拒否する API オペレーションを指定します。例えば、`kms:Encrypt` アクションは AWS KMS の暗号化 オペレーションに対応します。ポリシーステートメントに複数のアクションを一覧表示できます。詳細については、「アクセス許可に関するリファレンス」を参照してください。

Resource

(必須) キーポリシーでは、Resource 要素の値が "*" になります。これは「この KMS キー」を意味します。アスタリスク ("*") は、キーポリシーがアタッチされた KMS キーを識別します。

> **Note**
>
> 必要な `Resource` 要素がキーポリシーステートメントにない場合、ポリシーステートメントによる影響はありません。`Resource` 要素のないキーポリシーステートメントは、どの KMS キーにも適用されません。
> キーポリシーステートメントに `Resource`要素がない場合、AWS KMSコンソールはエラーを正しく報告しますが、ポリシーステートメントが無効な場合でも、CreateKeyおよび PutKeyPolicy APIsは成功します。

Condition

(オプション) 条件は、キーポリシーを有効にするために満たす必要がある要件を指定します。条件により、AWS は、API リクエストのコンテキストを評価し、ポリシーステートメントが適用されるかどうかを判断できます。

条件を指定するには、事前定義された条件キーを使用します。AWS KMS は AWS グローバル条件キーと AWS KMS 条件キーをサポートします。属性ベースのアクセスコントロール (ABAC) をサポートするために、AWS KMS は、タグとエイリアスに基づいて KMS キーへのアクセスを制御する条件キーを提供します。詳細については、「AWS KMS の ABAC」を参照してください。

条件の形式は次のとおりです。

```
"Condition": {"condition operator": {"condition key": "condition value"}}
```

例：

```
"Condition": {"StringEquals": {"kms:CallerAccount": "111122223333"}}
```

AWS ポリシー構文の詳細については、「IAM ユーザーガイド」の「AWS IAM ポリシーリファレンス」を参照してください。

キーポリシーの例

以下は、対称暗号化 KMS キーの完全なキーポリシーの例です。この章の主なポリシー概念について読むと、参考に使うことができます。このキーポリシーでは、先ほどの「デフォルトキーポリシー」のセクションからのポリシーステートメント例と、以下を実行するための 1 つのキーポリシーを組み合わせています。

- 例 AWS アカウント、111122223333 に、KMS キーへのフルアクセスを許可します。これにより、アカウントのルートユーザー (緊急用) を含む、アカウントとその管理者は、アカウントの IAM ポリシーを使用して KMS キーへのアクセスを許可できます。
- `ExampleAdminRole` IAM ロールが KMS キーを管理できるようにします。
- `ExampleUserRole` IAM ロールが KMS キーを使用できるようにします。

```
{
    "Id": "key-consolepolicy",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleAdminRole"
            },
            "Action": [
                "kms:Create*",
```

```
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Allow attachment of persistent resources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
    ]
}
```

デフォルトのキーポリシー

KMS キーを作成するときに、新しい KMS キーのキーポリシーを指定できます。指定しない場合、このポリシーは AWS KMS によって作成されます。AWS KMS が使用するデフォルトのキーポリシーは、キーが AWS KMS コンソールで作成されたかどうか、または AWS KMS API を使用したかどうかによって異なります。

KMS キーをプログラムにより作成したときのデフォルトキーポリシー

AWS KMS API を使用して KMS キーをプログラムで作成し (AWS SDK、AWS Command Line Interface、または AWS Tools for PowerShell の使用を含む)、キーポリシーを指定しなかった場合、AWS KMS は非常に単純なデフォルトキーポリシーを適用します。このデフォルトのキーポリシーには、KMS キー許可を有する AWS アカウント に対して、IAM ポリシーを使用して KMS キー上のすべての AWS KMS オペレーションに対するアクセスを許可する権限を付与するポリシーステートメントが 1 つ含まれます。このポリシーステートメントの詳細については、「AWS アカウント へのアクセスを許可し、IAM ポリシーを有効にする」を参照してください。

AWS Management Console を使用して KMS キーを作成した場合のデフォルトのキーポリシー

AWS Management Console を使用して KMS キーを作成した場合、キーポリシーは、AWS アカウント へのアクセスを許可し、IAM ポリシーを有効にするポリシーステートメントで始まります。次に、コンソールはキー管理者ステートメント、キーユーザーステートメント、および (ほとんどのキータイプで) プリンシパルに他の AWS サービスで KMS キーを使用する許可を与えるステートメントを追加します。AWS KMS コンソールの機能を使用して IAM ユーザー、IAM ロール、キー管理者およびキーユーザー (またはその両方) である AWS アカウント を指定できます。

アクセス許可

- AWS アカウント へのアクセスを許可し、IAM ポリシーを有効にする
- KMS キーの管理をキー管理者に許可する
- KMS キーの使用をキーユーザーに許可する

- 暗号化オペレーションで KMS キーを使用することをキーユーザーに許可する
- AWS サービスで KMS キーを使用することをキーユーザーに許可する

AWS アカウント へのアクセスを許可し、IAM ポリシーを有効にする

次のデフォルトのキーポリシーステートメントは重要です。

- このステートメントは、KMS キーを所有する AWS アカウント に KMS キーへのフルアクセスを付与しています。

  他の AWS リソースポリシーとは異なり、AWS KMS キーポリシーは、アカウントまたはそのアイデンティティに対して許可を自動的に付与しません。アカウント管理者に許可を付与するには、キーポリシーに、このような許可を提供する明示的なステートメントを含める必要があります。
- これにより、アカウントは IAM ポリシーを使用して、キーポリシーに加えて KMS キーへのアクセスを許可できるようになります。

  この許可がないと、キーへのアクセスを許可する IAM ポリシーは無効になりますが、キーへのアクセスを拒否する IAM ポリシーは依然として有効です。
- 削除できないアカウントのルートユーザーを含むアカウント管理者にアクセスコントロールの許可を付与することで、キーが管理不能になるリスクが軽減されます。

次のキーポリシーステートメントは、プログラムで作成された KMS キーのデフォルトキーポリシーの全文です。これは、AWS KMS コンソールで作成された KMS キーのデフォルトキーポリシーの最初のポリシーステートメントです。

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
   },
  "Action": "kms:*",
  "Resource": "*"
}
```

IAM ポリシーが KMS キーへのアクセスを許可できるようにします。

上記のキーポリシーステートメントは、キーの許可を有する AWS アカウント に IAM ポリシーとキーポリシーを使用して、KMS キーに対するすべてのアクション (kms:*) を許可する権限を付与しています。

このキーポリシーステートメントのプリンシパルはアカウントプリンシパルであり、これはこの形式にある ARN で表されています: arn:aws:iam::*account-id*:root。アカウントプリンシパルは、AWS アカウントとその管理者を表します。

キーポリシーステートメントのプリンシパルがアカウントプリンシパルである場合、ポリシーステートメントはどの IAM プリンシパルに対しても、KMS キーを使用する許可を付与しません。代わりにアカウントに IAM ポリシーを使用して、ポリシーステートメントで指定された許可を委譲することができます。このデフォルトのキーポリシーステートメントにより、アカウントは IAM ポリシーを使用して、KMS キーに対するすべてのアクション (kms:*) の許可を委譲することができます。

KMS キーが管理不能になるリスクを減らします。

他の AWS リソースポリシーとは異なり、AWS KMS キーポリシーは、アカウントまたはそのプリンシパルに対して許可を自動的に付与しません。プリンシパルに許可を与えるには (アカウントプリンシパルを含む)、許可を明示的に提供するキーポリシーステートメントを使用する必要があります。アカウントプリンシパルまたはプリンシパルに、KMS キーへのアクセス権を付与する必要はありません。ただし、アカウントプリンシパルにアクセス権を付与することは、キーが管理不能になるのを防ぐために役立ちます。

たとえば、KMS キーへのアクセスを 1 人のユーザーだけに付与するキーポリシーを作成したとします。その後、そのユーザーを削除すると、キーは管理不能になり、KMS キーへのアクセスを取り戻すためにAWS サポートに連絡しなければなりません。

上記のキーポリシーステートメントは、キーを制御する許可をアカウントプリンシパルに付与します。アカウントプリンシパルは、アカウントのルートユーザーを含む AWS アカウント およびその管理者を表します。アカウントのルートユーザーは、AWS アカウント を削除しない限り削除できない唯一のプリンシパルです。IAM のベストプラクティスでは、緊急時を除き、アカウントのルートユーザーの代わりとして行動することを推奨していません。ただし、KMS キーへのアクセス権を持つ他のすべてのユーザーおよびロールを削除した場合には、アカウントのルートユーザーとして行動する必要が生じる場合があります。

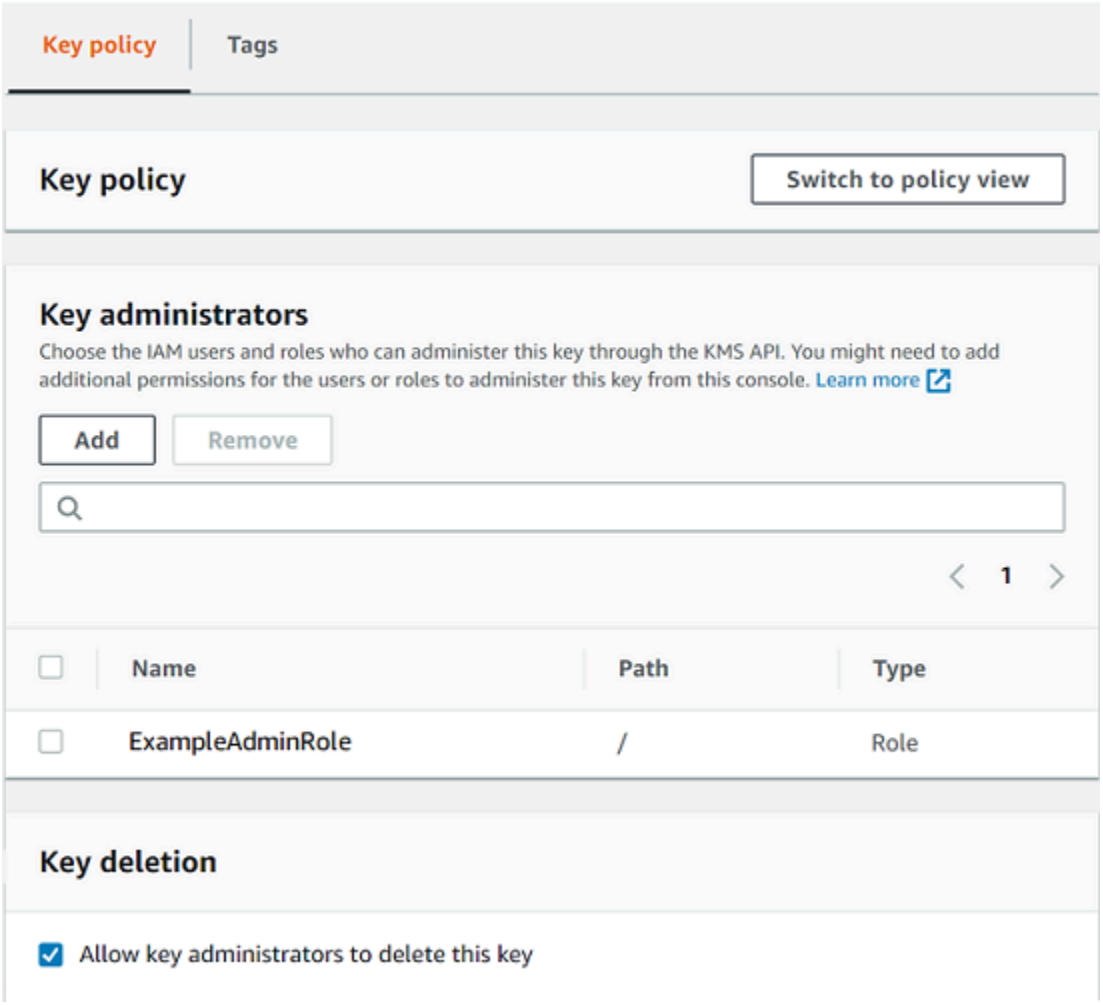
KMS キーの管理をキー管理者に許可する

コンソールによって作成されるデフォルトのキーポリシーでは、アカウントで IAM ユーザーとロールを選択し、それらをキー管理者にすることができます。このステートメントは、キー管理者ステートメントと呼ばれます。キー管理者には、KMS キーを管理するためのアクセス許可はありますが、暗号化オペレーションで KMS キーを使用するためのアクセス許可はありません。デフォルトビューやポリシービューで KMS キーを作成するときに、IAM ユーザーとロールをキー管理者のリストに追加できます。

> **Warning**
>
> キー管理者は、キーポリシーを変更してグラントを作成するグラントを持っているため、このポリシーで指定されていない AWS KMS 許可を自分自身や他者に付与することができます。
> タグとエイリアスを管理するアクセス権限を持つプリンシパルも、KMS キーへのアクセスを制御することができます。詳細については、「AWS KMS の ABAC」を参照してください。

> **Note**
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

次の例は、AWS KMS コンソールのデフォルトビューでのキー管理者ステートメントを示しています。



AWS KMS コンソールのポリシービューでのキー管理者ステートメントの例を次に示します。このキー管理者ステートメントは、単一リージョンの対称暗号化 KMS キー向けのものです。

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS":"arn:aws:iam::111122223333:role/ExampleAdminRole"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
```

```
    "kms:Delete*",
    "kms:TagResource",
    "kms:UntagResource",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

最も一般的な KMS キーである単一リージョンの対称暗号化 KMS キーのデフォルトキー管理者ステートメントは、以下の許可を付与します。各アクセス許可の詳細については、「AWS KMS アクセス許可」を参照してください。

AWS KMS コンソールを使用して KMS キーを作成する場合、コンソールでは、指定したユーザーとロールをキー管理者ステートメントの `Principal` 要素に追加します。

これらのアクセス許可の多くには、指定した動詞で始まるアクセス許可すべてを許可するワイルドカード文字 (*) が含まれています。その結果、AWS KMS で新しい API オペレーションが追加されると、キー管理者が自動的にそれらを使用できるようになります。新しいオペレーションを含めるために、キーポリシーを更新する必要はありません。キー管理者を固定された一連の API オペレーションに限定する場合は、キーポリシーを変更することができます。

`kms:Create*`

`kms:CreateAlias` および `kms:CreateGrant` を許可します。(`kms:CreateKey` のアクセス許可は IAM ポリシーでのみ有効です)

`kms:Describe*`

`kms:DescribeKey` を許可します。– AWS Management Console で KMS キーのキーの詳細ページを表示するには、`kms:DescribeKey` のアクセス許可が必要です。

`kms:Enable*`

`kms:EnableKey` を許可します。対称暗号化 KMS キーについては、`kms:EnableKeyRotation` も許可します。

`kms:List*`

`kms:ListGrants`、`kms:ListKeyPolicies`、および `kms:ListResourceTags` を許可します。(KMS キーを AWS Management Console で表示するのに必要な `kms:ListAliases` および `kms:ListKeys` のアクセス許可は、IAM ポリシーでのみ有効です)

kms:Put*

kms:PutKeyPolicy を許可します。このアクセス許可により、キー管理者が、この KMS キーのキーポリシーを変更できるようになります。

kms:Update*

kms:UpdateAlias および kms:UpdateKeyDescription を許可します。マルチリージョンキーの場合、この KMS キーで kms:UpdatePrimaryRegion が許可されます。

kms:Revoke*

kms:RevokeGrant を許可します。これにより、キー管理者は自分がグラントの無効なプリンシパルでなくてもグラントを削除できるようになります。

kms:Disable*

kms:DisableKey を許可します。対称暗号化 KMS キーについては、kms:DisableKeyRotation も許可します。

kms:Get*

kms:GetKeyPolicy および kms:GetKeyRotationStatus を許可します。インポートしたキーマテリアルを含む KMS キーの場合、kms:GetParametersForImport を許可します。非対称 KMS キーの場合、kms:GetPublicKey を許可します。AWS Management Console で KMS キーのキーポリシーを表示するには、kms:GetKeyPolicy のアクセス許可が必要です。

kms:Delete*

kms:DeleteAlias を許可します。インポートしたキーマテリアルを含むキーの場合、kms:DeleteImportedKeyMaterial を許可します。kms:Delete* のアクセス許可では、キー管理者が KMS キー (ScheduleKeyDeletion) を削除できません。

kms:TagResource

kms:TagResource を許可します。これにより、キー管理者は KMS キーにタグを追加できるようになります。タグは KMS キーへのアクセス制御にも使用できるため、このアクセス許可により、管理者が KMS キーへのアクセスを許可または拒否できるようになります。詳細については、「AWS KMS の ABAC」を参照してください。

kms:UntagResource

kms:UntagResource を許可します。これにより、キー管理者は KMS キーからタグを削除できるようになります。タグはキーへのアクセス制御に使用できるため、このアクセス許可により、

管理者は KMS キーへのアクセスを許可または拒否できるようになります。詳細については、「AWS KMS の ABAC」を参照してください。

kms:ScheduleKeyDeletion

kms:ScheduleKeyDeletion を許可します。これにより、キー管理者はこの KMS キーを削除できるようになります。このアクセス許可を削除するには、[Allow key administrators to delete this key] (キーの管理者がこのキーを削除できるようにします) オプションをオフにします。

kms:CancelKeyDeletion

kms:CancelKeyDeletion を許可します。これにより、キー管理者はこの KMS キーの削除をキャンセルできるようになります。このアクセス許可を削除するには、[Allow key administrators to delete this key] (キーの管理者がこのキーを削除できるようにします) オプションをオフにします。

特定用途のキーを作成するときに、AWS KMS がデフォルトのキー管理者ステートメントに次のアクセス許可を追加します。

kms:ImportKeyMaterial

kms:ImportKeyMaterial のアクセス許可により、キー管理者が KMS キーにキーマテリアルをインポートできるようになります。このアクセス許可は、キーマテリアルのない KMS キーを作成する場合にのみキーポリシーに含まれます。

kms:ReplicateKey

kms:ReplicateKey のアクセス許可により、キー管理者が、異なる AWS リージョンでマルチリージョンプライマリキーのレプリカを作成できるようになります。このアクセス許可は、マルチリージョンプライマリキーまたはマルチリージョンレプリカキーを作成する場合にのみ、キーポリシーに含まれます。

kms:UpdatePrimaryRegion

kms:UpdatePrimaryRegion のアクセス許可により、キー管理者が、マルチリージョンレプリカキーをマルチリージョンプライマリキーに変更できるようになります。このアクセス許可は、マルチリージョンプライマリキーまたはマルチリージョンレプリカキーを作成する場合にのみ、キーポリシーに含まれます。

KMS キーの使用をキーユーザーに許可する

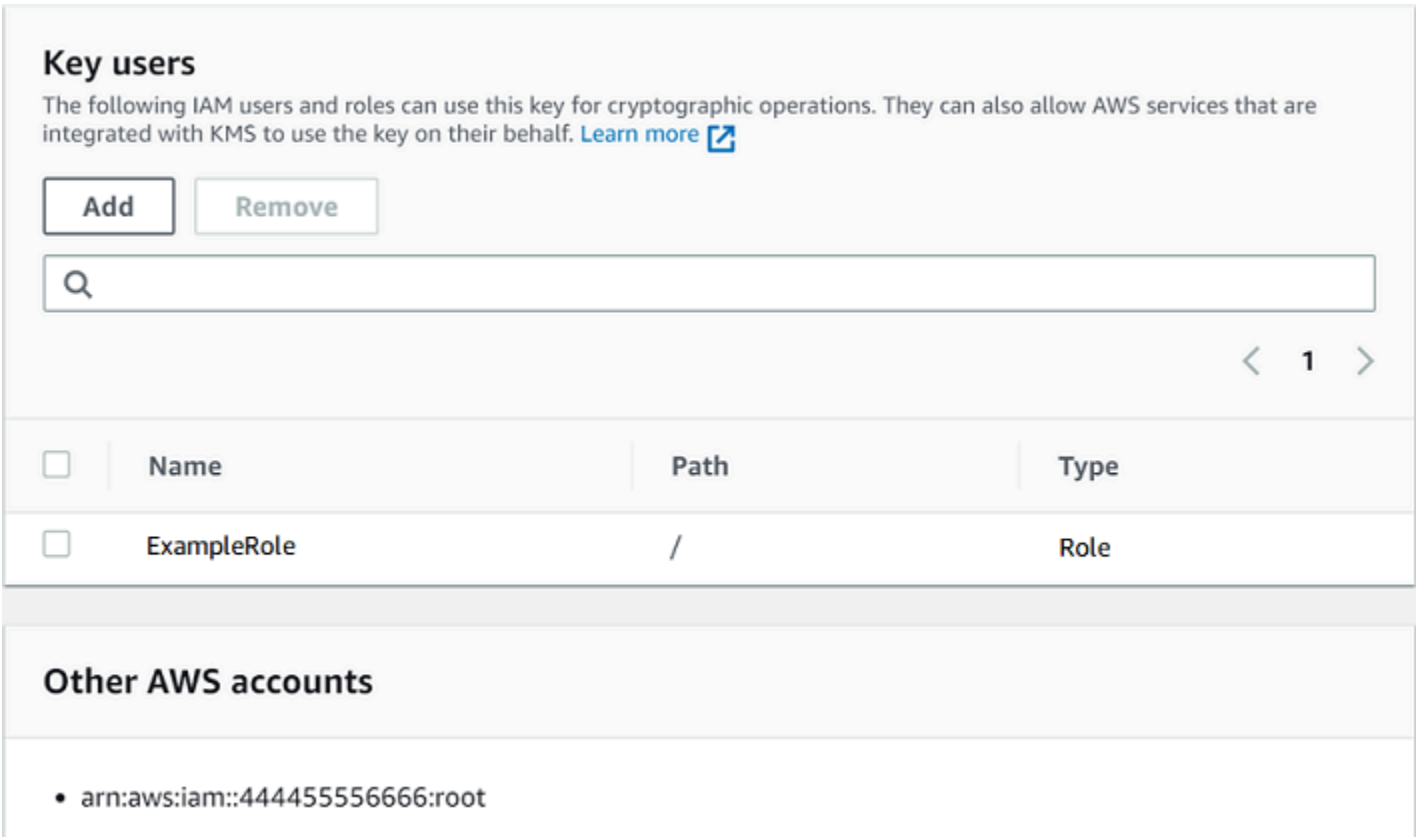
コンソールが KMS キー用に作成するデフォルトキーポリシーでは、アカウント内の IAM ユーザーと IAM ロール、および外部 AWS アカウントを選択し、それらをキーユーザーにすることができます。

コンソールは、キーユーザーのキーポリシーに 2 つのポリシーステートメントを追加します。

- KMS キーの直接使用 — 最初のキーポリシーステートメントは、そのタイプの KMS キーでサポートされるすべての暗号化オペレーションに対して KMS キーを直接使用するアクセス許可をキーユーザーに付与します。
- AWS サービスで KMS キーを使用する — 2 番目のポリシーステートメントでは、キーユーザーにアクセス許可を付与し、AWS KMS と統合された AWS サービスが KMS キーを使用して、Amazon S3 バケットや Amazon DynamoDB テーブルなどのリソースを保護できるようにします。

KMS キーの作成時に、IAM ユーザー、IAM ロール、その他の AWS アカウント をキーユーザーのリストに追加できます。次の図に示すように、キーポリシーのコンソールのデフォルトビューを使用してリストを編集することもできます。キーポリシー用デフォルトビューは、キーの詳細ページにあります。他の AWS アカウント のユーザーに KMS キーの使用を許可する方法の詳細については、他のアカウントのユーザーに KMS キーの使用を許可する を参照してください 。

> **Note**
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。



デフォルトの単一リージョンの対称用キーユーザーステートメントにより、次のアクセス許可が付与されます。各アクセス許可の詳細については、AWS KMS アクセス許可 を参照してください。

AWS KMS コンソールを使用して KMS キーを作成する場合、コンソールでは、指定したユーザーとロールを各キーユーザーステートメントの `Principal` 要素に追加します。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
```

```
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

暗号化オペレーションで KMS キーを使用することをキーユーザーに許可する

キーユーザーには、KMS キーでサポートされるすべての暗号化オペレーションで KMS キー を直接使用するためのアクセス許可が付与されています。また、 DescribeKey オペレーションを使用して、AWS KMSコンソールで、または AWS KMS API オペレーションを使用して、KMS キーに関する詳細情報を取得することもできます。

デフォルトでは、AWS KMS コンソールはデフォルトのキーポリシーに次の例のようなキーユーザーのステートメントを追加します。これらは異なる API オペレーションをサポートするため、対称暗号化 KMS キー、HMAC KMS キー、パブリックキー暗号化用の非対称 KMS キー、および署名と検証用の非対称 KMS キー向けのポリシーステートメント内のアクションは、それぞれわずかに異なります。

対称暗号化 KMS キー

コンソールは、対称暗号化 KMS キーのキーポリシーに以下のステートメントを追加します。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:Encrypt",
```

```
    "kms:GenerateDataKey*",
    "kms:ReEncrypt*"
  ],
  "Resource": "*"
}
```

HMAC KMS キー

コンソールは、HMAC KMS キーのキーポリシーに以下のステートメントを追加します。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateMac",
    "kms:VerifyMac"
  ],
  "Resource": "*"
}
```

非対称 KMS キーのパブリックキー暗号化

コンソールでは、暗号化と復号のキーを使用して、非対称 KMS キーのキーポリシーに次のステートメントを追加します。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
    "kms:GetPublicKey"
  ],
  "Resource": "*"
}
```

非対称 KMS キーの署名および検証

コンソールは、署名と検証のキーを使用して、非対称 KMS キーのキーポリシーに次のステートメントを追加します。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:Sign",
    "kms:Verify"
  ],
  "Resource": "*"
}
```

これらのステートメントのアクションは、キーユーザーに次のアクセス許可を付与します。

`kms:Encrypt`

キーユーザーがこの KMS キーでデータを暗号化できるようにします。

`kms:Decrypt`

キーユーザーがこの KMS キーでデータを復号できるようにします。

`kms:DescribeKey`

この KMS キーに関する詳細情報 (その識別子、作成日、キーステータスなど) を、キーユーザーが取得できるようにします。キーユーザーは KMS キーの詳細を AWS KMS コンソールに表示することもできます。

`kms:GenerateDataKey*`

クライアント側の暗号化オペレーションのために、対称データキーまたは非対称データキーペアをキーユーザーがリクエストできるようにします。コンソールは * ワイルドカード文字を使用して、、GenerateDataKey、、GenerateDataKeyWithoutPlaintextGenerateDataKeyPairおよび の API オペレーションのアクセス許可を表しますGenerateDataKeyPairWithoutPlaintext。これらのアクセス許可は、データキーを暗号化する対称 KMS キーでのみ有効です。

kms:GenerateMac

キーユーザーが HMAC KMS キーを使用して HMAC タグを生成できるようにします。

kms:GetPublicKey

非対称 KMS キーのパブリックキーをキーユーザーがダウンロードできるようにします。このパブリックキーを共有する当事者は、AWS KMS の外部のデータを暗号化できます。ただし、これらの暗号テキストは、AWS KMS で Decrypt オペレーションを呼び出すことによってのみ復号できます。

kms:ReEncrypt*

この KMS キーで最初に暗号化されたデータの再暗号化、またはこの KMS キーを使用して以前に暗号化されたデータの再暗号化をキーユーザーが行えるようにします。ReEncrypt オペレーションでは、送信元と送信先の KMS キーの両方にアクセスする必要があります。これを行うには、ソース KMS キーの `kms:ReEncryptFrom` アクセス許可と宛先 KMS キーの `kms:ReEncryptTo` アクセス許可を許可します。ただし、わかりやすいようにコンソールでは、両方の KMS キーで `kms:ReEncrypt*` を (* をワイルドカード文字で) 許可します。

kms:Sign

この KMS キーでメッセージにキーユーザーが署名できるようにします。

kms:Verify

この KMS キーで署名をキーユーザーが検証できるようにします。

kms:VerifyMac

キーユーザーが HMAC KMS キーを使用して HMAC タグを検証できるようにします。

AWS サービスで KMS キーを使用することをキーユーザーに許可する

コンソールのデフォルトキーポリシーでは、グラントを使用する AWS サービス内のデータを保護するために必要となるグラント許可もキーユーザーに付与されます。AWS サービスでは、KMS キーを使用するための特定の限定された許可を取得するために、しばしばグラントを使用します。

このキーポリシーステートメントでは、キーユーザーは KMS キーでグラントを作成、表示、取り消すことを許可しますが、グラントオペレーションのリクエストが AWS KMS と統合された AWS サービスからのものである場合に限ります。kms:GrantIsForAWSResource ポリシー条件では、ユーザーはこれらの許可オペレーションを直接呼び出すことはできません。キーユーザーが許可した場

合、AWS サービスはユーザーに代わってグラントを作成し、サービスが KMS キーを使用してユーザーのデータを保護できるようにします。

キーユーザーが統合サービスで KMS キーを使用するには、これらのグラント許可が必要になりますが、これらの許可だけでは不十分です。キーユーザーには、統合されたサービスを使用するアクセス許可も必要です。AWS KMS と統合される AWS サービスへのアクセスをユーザーに許可する方法の詳細については、統合されるサービスのドキュメントを参照してください。

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

例えば、キーユーザーは以下の方法により、KMS キーでこれらのアクセス許可を使用できます。

- この KMS キーを Amazon Elastic Block Store (Amazon EBS) および Amazon Elastic Compute Cloud (Amazon EC2) とともに使用して、暗号化された EBS ボリュームを EC2 インスタンスにアタッチします。キーユーザーは、KMS キーを使用して暗号化されたボリュームをインスタンスにアタッチするアクセス許可を、Amazon EC2 に暗黙的に付与します。詳細については、「Amazon Elastic Block Store (Amazon EBS) が AWS KMS を使用する方法」を参照してください。
- この KMS キーを Amazon Redshift とともに使用して、暗号化されたクラスターを起動します。キーユーザーは、KMS キーを使用して暗号化されたクラスターを起動し、暗号化されたスナップショットを作成するアクセス許可を、Amazon Redshift に暗黙的に付与します。詳細については、「Amazon Redshift が AWS KMS を使用する方法」を参照してください。
- この KMS キーを、AWS KMS と統合された他の AWS サービスで使用します。これらのサービスは、暗号化されたリソースを作成、管理したり、これらのサービスで使用する際にグラントを使用します。

デフォルトキーポリシーにより、キーユーザーは、自身のグラント許可をグラントを使用するすべての統合サービスに付与することができます。ただし、特定の AWS サービスへの許可を制限するカス

タムキーポリシーを作成することができます。詳細については、「kms:ViaService」の条件キーを参照してください。

キーポリシーの表示

アカウントAWS マネージドキー内の AWS KMSカスタマーマネージドキーまたは のキーポリシーを表示するには、 AWS Management Consoleまたは AWS KMS API の GetKeyPolicyオペレーションを使用します。これらの手法を使用して別の AWS アカウント にある KMS キーのキーポリシーを表示することはできません。

AWS KMS キーポリシーの詳細については、「AWS KMS のキーポリシー」を参照してください。KMS キーにアクセスできるユーザーとロールを特定する方法については、the section called “アクセスの確認” を参照してください。

トピック

- キーポリシーの表示 (コンソール)
- キーポリシーの表示 (AWS KMS API)

キーポリシーの表示 (コンソール)

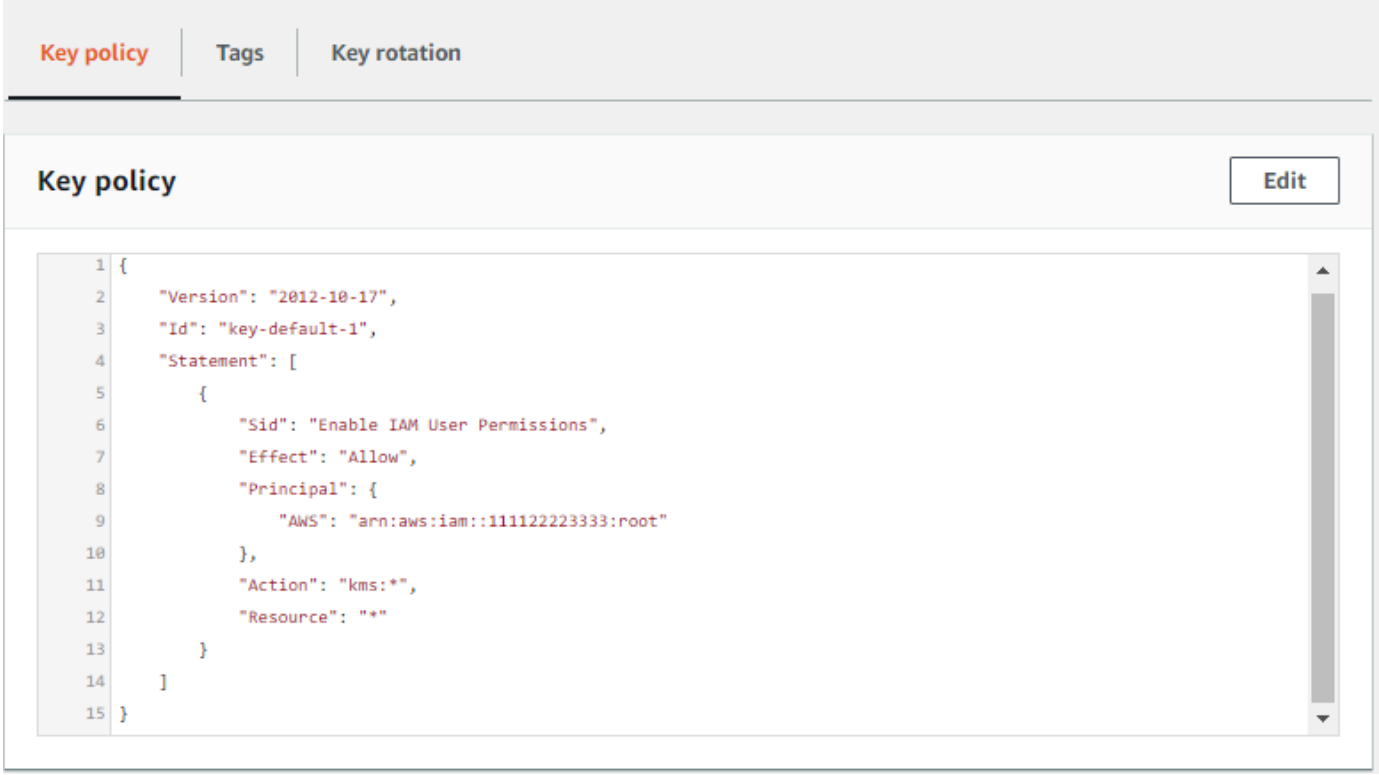
認可されたユーザーは、AWS Management Console の [Key policy] (キーポリシー) タブで、AWS マネージドキー またはカスタマーマネージドキーのキーポリシーを表示できます。

で KMS キーのキーポリシーを表示するにはAWS Management Console、kms:ListAliases、kms:DescribeKey、および kms:GetKeyPolicy アクセス許可が必要です。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。
4. KMS キーのリストで、確認する KMS キーのエイリアスまたはキー ID を選択します。
5. [Key policy] (キーポリシー) タブを選択します。

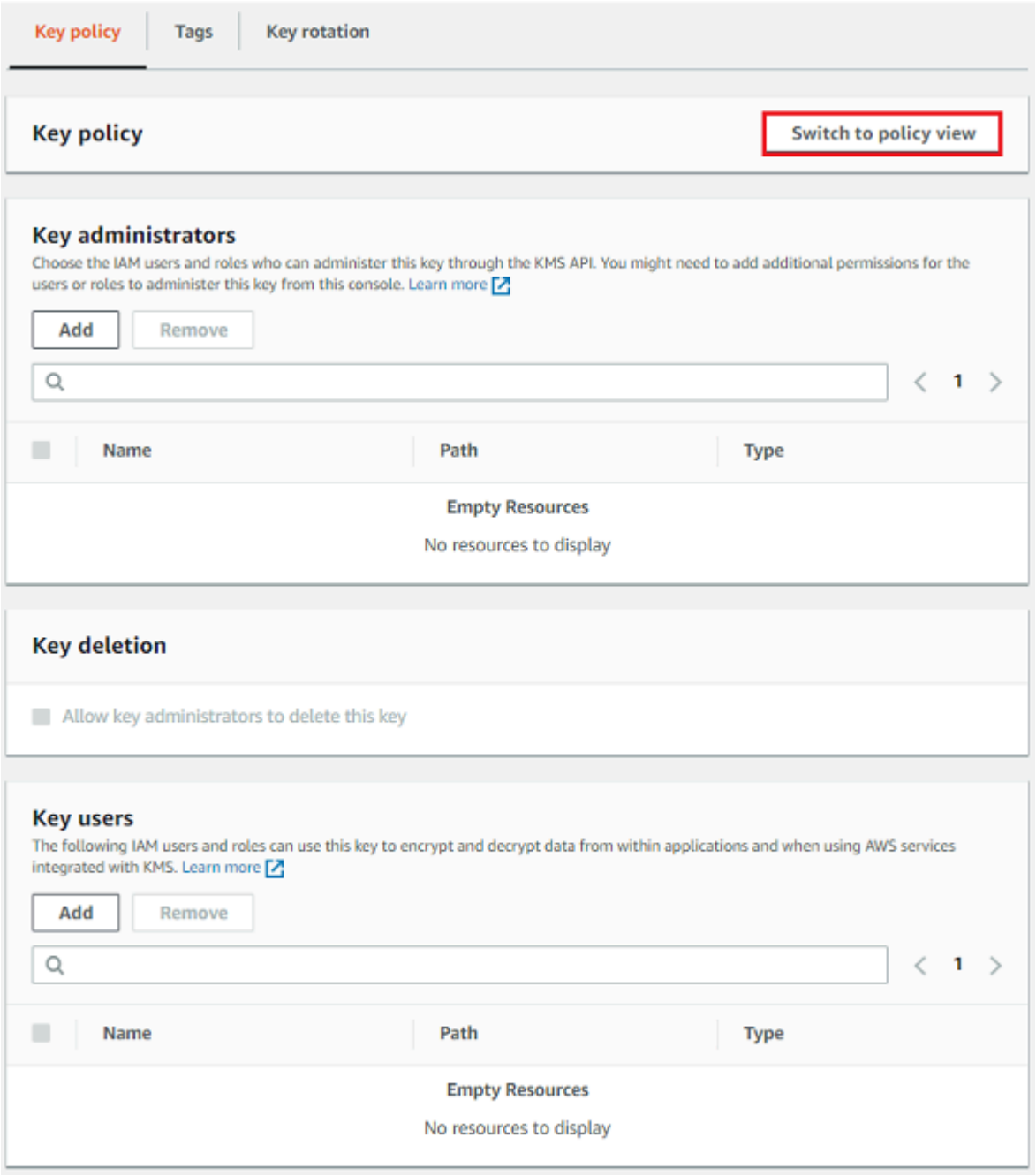
[キーポリシー] タブに、キーポリシードキュメントが表示されることがあります。これはポリシービューです。キーポリシーステートメントでは、キーポリシーによって KMS キーへのアクセス許可を付与されたプリンシパルを表示して、それらが実行できるアクションを確認できます。

次の例は、デフォルトのキーポリシーのポリシービューを示しています。



または、AWS Management Console で KMS キーを作成した場合、[Key administrators] (キー管理者)、[Key deletion] (キーの削除)、[Key Users] (キーユーザー) のセクションで、デフォルトビューを表示します。キーポリシードキュメントを表示するには、[ポリシービューに切り替える] を選択します。

次の例は、デフォルトのキーポリシーのデフォルトのビューを示しています。



キーポリシーの表示 (AWS KMS API)

で KMS キーのキーポリシーを取得するにはAWS アカウント、 AWS KMS API の GetKeyPolicyオペレーションを使用します。このオペレーションを使用して、別のアカウントのキーポリシーを表示することはできません。

次の例では、 AWS Command Line Interface (AWS CLI) の get-key-policy コマンドを使用していますが、任意の AWS SDK を使用してこのリクエストを行うことができます。

default が唯一の有効な値であっても、PolicyName パラメータ は必須であることに注意してください。また、このコマンドは、表示を容易にするために、JSON ではなくテキストでの出力を要求します。

このコマンドを実行する前に、サンプルキー ID をアカウントの有効なキー ID に置き換えます。

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
```

レスポンスは、デフォルトのキーポリシーを返す、次のようなものである必要があります。

```
{
  "Version" : "2012-10-17",
  "Id" : "key-consolepolicy-3",
  "Statement" : [ {
    "Sid" : "Enable IAM User Permissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

キーポリシーの変更

AWS Management Console または PutKeyPolicyオペレーションAWS アカウントを使用して、 の KMS キーのキーポリシーを変更できます。これらの手法を使用して別の AWS アカウント にある KMS キーのキーポリシーを変更することはできません。

キーポリシーを変更する場合は、以下のルールに注意してください。

- AWS マネージドキー または カスタマーマネージドキーのキーポリシーを表示できますが、変更できるのは、カスタマーマネージドキーのキーポリシーのみです。AWS マネージドキー のポリシーは、アカウントで KMS キーを作成した AWS のサービスによって作成および管理されます。AWS 所有のキー のキーポリシーは表示または変更できません。
- キーポリシーで IAM ユーザー、IAM ロール、AWS アカウント を追加または削除し、これらのプリンシパルに対して許可または拒否されるアクションを変更できます。キーポリシーでプリンシパルとアクセス権限を指定する方法については、「キーポリシー」を参照してください。

- IAM グループをキーポリシーに追加することはできませんが、複数の IAM ユーザーおよび IAM ロールを追加できます。詳細については、「複数の IAM プリンシパルが KMS キーにアクセスできるようにする」を参照してください。
- 外部 AWS アカウント をキーポリシーに追加する場合は、外部アカウントの IAM ポリシーを使用して、それらのアカウントの IAM ユーザー、グループ、ロールにアクセス許可を付与する必要があります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。
- 結果として得られるキーポリシードキュメントは 32 KB (32,768 バイト) を超えることはできません。

トピック

- キーポリシーを変更する方法
- 複数の IAM プリンシパルが KMS キーにアクセスできるようにする

キーポリシーを変更する方法

キーポリシーは、以下のセクションで説明している 3 つの異なる方法で変更できます。

トピック

- AWS Management Console のデフォルトビューの使用
- AWS Management Console のポリシービューの使用
- AWS KMS API を使用する場合

AWS Management Console のデフォルトビューの使用

コンソールを使用して、デフォルトビューと呼ばれるグラフィカルインターフェイスで、キーポリシーを変更できます。

以下のステップがコンソールの表示と一致しない場合、このキーポリシーはコンソールで作成されなかった可能性があります。または、コンソールのデフォルトビューがサポートしていない方法でキーポリシーが変更されている可能性があります。その場合は、「AWS Management Console のポリシービューの使用」または「AWS KMS API を使用する場合」の手順に従ってください。

1. キーポリシーの表示 (コンソール) の説明に従って、カスタマーマネージドキーのキーポリシーを表示します。(AWS マネージドキー のキーポリシーの変更はできません)。
2. 変更する対象を決定します。

- キー管理者を追加または削除し、キー管理者による KMS キーの削除を許可または拒否するには、ページの [Key administrators] (キー管理者)セクションのコントロールを使用します。キー管理者は、KMS キーの有効化と無効化、キーポリシーの設定、キーローテーションの有効化などを含む KMS キーの管理を行います。
- キーユーザーを追加または削除し、外部 AWS アカウント による KMS キーの使用を許可または拒否するには、本ページの [Key users] (キーユーザー) セクションのコントロールを使用します。キーユーザーは、データキーの暗号化、復号、再暗号化、生成などの暗号化オペレーションで KMS キーを使用できます。

AWS Management Console のポリシービューの使用

コンソールのポリシービューで、キーポリシードキュメントを変更できます。

1. キーポリシーの表示 (コンソール) の説明に従って、カスタマーマネージドキーのキーポリシーを表示します。(AWS マネージドキー のキーポリシーの変更はできません)。
2. [Key Policy (キーポリシー)] セクションで、[Switch to policy view (ポリシービューへの切り替え)] を選択します。
3. キーポリシードキュメントを編集し、[変更の保存] を選択します。

AWS KMS API を使用する場合

PutKeyPolicy オペレーションを使用して、 の KMS キーのキーポリシーを変更できますAWS アカウント。この API を別の AWS アカウント の KMS キーで使用することはできません。

1. GetKeyPolicy オペレーションを使用して既存のキーポリシードキュメントを取得し、キーポリシードキュメントをファイルに保存します。複数のプログラミング言語のサンプルコードについては、「キーポリシーの取得」を参照してください。
2. 任意のテキストエディタでキーポリシードキュメントを開き、キーポリシードキュメントを編集してファイルを保存します。
3. PutKeyPolicy オペレーションを使用して、更新されたキーポリシードキュメントを KMS キーに適用します。複数のプログラミング言語のサンプルコードについては、「キーポリシーの設定」を参照してください。

ある KMS キーから別の KMS キーにキーポリシーをコピーする例については、 AWS CLI コマンドリファレンスGetKeyPolicy の例を参照してください。

複数の IAM プリンシパルが KMS キーにアクセスできるようにする

IAM グループは、キーポリシー内の有効なプリンシパルではありません。複数のユーザーおよびロールが KMS キーにアクセスできるようにするには、次のいずれかを行います。

- IAM ロールをキーポリシーのプリンシパルとして使用します。必要に応じて、複数の権限を持つユーザーがそのロールを引き受けることができます。詳細については、「IAM ユーザーガイド」の「IAM ロール」を参照してください。

  複数の IAM ユーザーをキーポリシーにリストすることは可能ですが、許可されたユーザーのリストが変更されるたびにキーポリシーを更新する必要があるため、この方法は推奨されません。また、IAM のベストプラクティスでは、長期的な認証情報を持つ IAM ユーザーの使用は推奨されていません。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。
- IAM ポリシーを使用して IAM グループに許可を付与します。そのためには、キーポリシーに、IAM ポリシーでKMS キーへのアクセスを許可するステートメントが含まれていることを確認します。次に、KMS キーへのアクセスを許可する IAM ポリシーを作成し、そのポリシーを IAM グループ (許可された IAM ユーザーを含む) にアタッチします。このアプローチを使用すると、承認されたユーザーのリストが変更されたときにポリシーを変更する必要はありません。代わりに、適切な IAM グループに対してそれらのユーザーを追加または削除するだけで済みます。詳細については、「IAM ユーザーガイド」の「IAM ユーザーグループ」を参照してください。

AWS KMS キーポリシーと IAM ポリシーがどのように連携するかについての詳細は、キーアクセスのトラブルシューティング を参照してください。

キーポリシーにおける AWS サービスのアクセス許可

多くの AWS のサービスでは、管理するリソースを保護するために AWS KMS keys が使用されます。あるサービスで AWS 所有のキー または AWS マネージドキー が使用される場合、そのサービスではこれらの KMS キーのキーポリシーが確立されて管理されます。

ただし、お客様が AWS のサービスでカスタマーマネージドキーを使用する場合は、ご自身でキーポリシーを設定して管理します。ユーザーに代わってリソースを保護するのに必要な最小限のアクセス許可が、そのキーポリシーによってサービスに付与される必要があります。最小特権の原則 (サービスで必要なアクセス許可のみを付与) に従うことをお勧めします。どのアクセス許可がそのサービスで必要かを把握し、AWS グローバル条件キーおよび AWS KMS 条件キーを使用してアクセス許可を絞り込むことで、これを効果的に行うことができます。

カスタマーマネージドキーに対してサービスで必要になるアクセス許可を調べるには、そのサービスの暗号化に関するドキュメントを参照してください。例えば、Amazon Elastic Block Store (Amazon EBS) で必要なアクセス許可については、Linux インスタンス用 Amazon EC2 ユーザーガイドおよび Windows インスタンス用 Amazon EC2 ユーザーガイドの Permissions for IAM users を参照してください。Secrets Manager で必要なアクセス許可については、AWS Secrets Manager ユーザーガイドの Authorizing use of the KMS key を参照してください。

最小特権のアクセス許可の実装

KMS キーを使用するためのアクセス許可を AWS のサービスに付与する場合、そのサービスでお客様に代わってアクセスが行われる必要があるリソースに対してのみ、アクセス許可が有効であることを確認します。この最小特権戦略は、AWS のサービス間でリクエストが渡されたときに KMS キーが不正に使用されるのを防ぐのに役立ちます。

最小特権戦略を実装する場合、AWS KMS 暗号化コンテキスト条件キー、およびグローバルソース ARN またはソースアカウントの条件キーを使用することをお勧めします。

暗号化コンテキスト条件キーの使用

AWS KMS リソースを使用するときに最小特権の許可を実装する最も効果的な方法は、プリンシパルが AWS KMS 暗号化オペレーションを呼び出すことを許可するポリシーに kms:EncryptionContext:context-key または kms:EncryptionContextKeys 条件キーを含めることです。これらの条件キーは、リソースが暗号化されるときに暗号文にバインドされる暗号化コンテキストにアクセス許可を関連付けるため、特に効果的です。

暗号化コンテキスト条件キーは、ポリシーステートメントのアクションが AWS KMS である場合、CreateGrantまたは GenerateDataKeyや Decrypt などのオペレーションなど、`EncryptionContext`パラメータを受け取る対称暗号化オペレーションの場合にのみ使用します。(サポートされているオペレーションのリストについては、kms:EncryptionContext:context-key または kms:EncryptionContextKeys を参照してください) これらの条件キーを使用して などの他のオペレーションを許可するとDescribeKey、アクセス許可は拒否されます。

サービスがリソースを暗号化するときに使用する暗号化コンテキストに値を設定します。この情報は通常、サービスのドキュメントのセキュリティに関する章で確認できます。例えば、AWS Proton の暗号化コンテキストが、AWS Proton リソースとそれに関連するテンプレートを識別します。AWS Secrets Manager 暗号化コンテキストが、シークレットとそのバージョンを識別します。Amazon Location の暗号化コンテキストが、トラッカーやコレクションを識別します。

次のキーポリシーステートメントの例では、承認されたユーザーの代わりに Amazon Location Service がグラントの作成を許可しています。このポリシーステートメント

は、kms:ViaService、kms:CallerAccount、および kms:EncryptionContext:*context-key*条件キーを使用してアクセス許可を制限し、アクセス許可を特定のトラッカーリソースに結び付けます。

```
{
  "Sid": "Allow Amazon Location to create grants on behalf of authorized users",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/LocationTeam"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "geo.us-west-2.amazonaws.com",
      "kms:CallerAccount": "111122223333",
      "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:tracker/
SAMPLE-Tracker"
    }
  }
}
```

aws:SourceArn または aws:SourceAccount 条件キーの使用

キーポリシーステートメントのプリンシパルが AWS のサービスプリンシパルになる場合は、kms:EncryptionContext:*context-key* 条件キーに加えて aws:SourceArn または aws:SourceAccount グローバル条件キーの使用を強くお勧めします。ARN と アカウントの値は、リクエストが別の AWS KMS のサービスから AWS に送信された場合にのみ、認可コンテキストに含まれます。この条件の組み合わせにより、最小特権のアクセス許可が実装され、可能性のある「混乱した代理」シナリオが回避されます。サービスプリンシパルは通常、キーポリシーではプリンシパルとして使用されませんが、AWS CloudTrail などの 一部の AWS のサービスでは必要になります。

aws:SourceArn または aws:SourceAccount グローバル条件キーを使用する場合、暗号化されているリソースの Amazon リソースネーム (ARN) またはアカウントに値を設定します。例えば、証跡を暗号化するための AWS CloudTrail アクセス許可を付与するキーポリシーステートメントでは、その証跡の ARN に aws:SourceArn の値を設定します。可能な限り、より具体的な aws:SourceArn を使用してください。ARN またはワイルドカード文字を使用した ARN パターンに値を設定します。リソースの ARN が不明の場合は、代わりに aws:SourceAccount を使用してください。

> **Note**
>
> AWS KMS キーポリシーで許可されていない文字がリソース ARN に含まれている場合、そのリソース ARN を aws:SourceArn 条件キーの値に使用することはできません。その代わりに aws:SourceAccount 条件キーを使用してください。キーポリシードキュメントのルールに関する詳細については、「キーポリシー形式」を参照してください。

次のキーポリシーの例では、アクセス許可を取得するプリンシパルは AWS CloudTrail サービスプリンシパル、cloudtrail.amazonaws.com になります。最小特権を実装するために、このポリシーでは aws:SourceArn および kms:EncryptionContext:*context-key* 条件キーが使用されます。ポリシーステートメントでは CloudTrail 、 は KMS キーを使用して、証跡の暗号化に使用するデータキーを生成できます。aws:SourceArn および kms:EncryptionContext:*context-key* の条件は個別に評価されます。指定されたオペレーションの KMS キーを使用するリクエストでは、両方の条件が満たされている必要があります。

サンプルアカウント (111122223333) および us-west-2 リージョンの finance 証跡に対するサービスのアクセス許可を制限する場合、このポリシーステートメントで aws:SourceArn 条件キーを特定の証跡の ARN に設定します。条件ステートメントは、 ArnEquals演算子を使用して、ARN 内のすべての要素が一致するときに個別に評価されるようにします。この例では、特定のアカウントおよびリージョンの証跡に対するアクセス許可を制限するために、kms:EncryptionContext:*context-key* 条件キーも使用されています。

このキーポリシーを使用する前に、サンプルアカウント ID、リージョン、および証跡名をアカウントの有効な値に置き換えてください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow CloudTrail to encrypt logs",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
```

```
            "arn:aws:cloudtrail:us-west-2:111122223333:trail/finance"
          ]
        },
        "StringLike": {
          "kms:EncryptionContext:aws:cloudtrail:arn": [
            "arn:aws:cloudtrail:*:111122223333:trail/*"
          ]
        }
      }
    }
  ]
}
```

AWS KMS で IAM ポリシーを使用する

IAM ポリシーをキーポリシー、権限、VPC エンドポイントポリシーとともに使用して、AWS KMS の AWS KMS keys へのアクセスを制御できます。

> **Note**
>
> IAM ポリシーを使用して KMS キーへのアクセスを制御するには、KMS キーのキーポリシーが IAM ポリシーを使用するアクセス許可をアカウントに付与する必要があります。具体的には、キーポリシーには IAM ポリシーを有効にするポリシーステートメントを含める必要があります。
>
> このセクションでは、IAM ポリシーを使用して、AWS KMS オペレーションへのアクセスを制御する方法について説明します。IAM の一般的な情報については、「IAM ユーザーガイド」を参照してください。

すべての KMS キーはキーポリシーを持つ必要があります。IAM ポリシーはオプションです。IAM ポリシーを使用して KMS キーへのアクセスを制御するには、KMS キーのキーポリシーが IAM ポリシーを使用するアクセス許可をアカウントに付与する必要があります。具体的には、キーポリシーには IAM ポリシーを有効にするポリシーステートメントを含める必要があります。

IAM ポリシーは、任意の AWS KMS オペレーションへのアクセスを制御できます。キーポリシーとは異なり、IAM ポリシーは複数の KMS キーへのアクセスを制御し、複数の関連 AWS サービスのオペレーションに対するアクセス許可を付与できます。ただし、IAM ポリシーは、 などのオペレーションへのアクセスを制御するのに特に便利です。特定の KMS キーが関連しないためCreateKey、キーポリシーでは制御できません。

Amazon Virtual Private Cloud (Amazon VPC) エンドポイントを介して AWS KMS にアクセスする場合、VPC エンドポイントポリシーを使用して、エンドポイントの使用時に AWS KMS リソースへのアクセスを制限することもできます。例えば、VPC エンドポイントを使用する場合、AWS アカウント のプリンシパルのみにカスタマーマネージドキーへのアクセスを許可できます。詳細については、「VPC エンドポイントへのアクセスの制御」を参照してください。

JSON ポリシードキュメントの記述と書式設定については、『IAM ユーザーガイド』の「IAM JSON ポリシーリファレンス」を参照してください。

トピック

- IAM ポリシーの概要
- IAM ポリシーのベストプラクティス
- IAM ポリシーステートメントで KMS キーを指定する
- AWS KMS コンソールの使用に必要な許可
- パワーユーザー用向けの AWS 管理ポリシー
- IAM ポリシーの例

IAM ポリシーの概要

IAM ポリシーは、次の方法で使用できます。

- フェデレーションまたはクロスアカウントアクセス権限のロールにアクセス許可ポリシーをアタッチ する — IAM ロールに IAM ポリシーをアタッチして、ID フェデレーションを有効にしたり、クロスアカウントアクセス権限を許可したり、EC2 インスタンスで実行されているアプリケーションにアクセス許可を付与したりできます。IAM ロールのさまざまなユースケースの詳細については、IAM ユーザーガイドの IAM ロールを参照してください。
- ユーザーまたはグループにアクセス許可ポリシーをアタッチする — ユーザーまたはユーザーのグループ に AWS KMS オペレーションの呼び出しを許可するポリシーをアタッチできます。ただし、IAM ベストプラクティスでは、可能な限り IAM ロールなどの一時的な認証情報を持つアイデンティティを使用することが推奨されています。

次の例は、AWS KMS のアクセス許可を使用する IAM ポリシーを示しています。このポリシーは、アタッチされた IAM アイデンティティで、すべての KMS キーとエイリアスを一覧表示できるようにします。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

すべての IAM ポリシーと同様に、このポリシーには Principal 要素がありません。IAM アイデンティティ に IAM ポリシーをアタッチすると、そのアイデンティティは、ポリシーで指定されたアクセス権限を取得します。

すべての AWS KMS API アクションとそれらが適用されるリソースの表については、「アクセス許可に関するリファレンス」を参照してください。

IAM ポリシーのベストプラクティス

AWS KMS keys へのアクセスを保護することは、すべての AWS リソースのセキュリティにとって不可欠です。KMS キーは、AWS アカウント で最も機密性の高いリソースの多くを保護するために使用されます。KMS キーへのアクセスを制御するキーポリシー、IAM ポリシー、権限、VPC エンドポイントポリシーを設計します。

KMS キーへのアクセスを制御する IAM ポリシーステートメントでは、最小権限の原則を使用します。IAM プリンシパルで、使用または管理する必要がある KMS キーのみに対して、必要なアクセス許可のみを付与します。

次のベストプラクティスは、AWS KMS キーとエイリアスへのアクセスを制御する IAM ポリシーに適用されます。IAM ポリシーの一般的なベストプラクティスのガイダンスについては、「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。

キーポリシーを使用する

可能であれば、他の AWS アカウント を含む多くの KMS キーに適用できる IAM ポリシーではなく、1 つの KMS キーに影響するキーポリシーでアクセス許可を付与します。これは、kms:PutKeyPolicy や kms:ScheduleKeyDeletion などの機密性の高いアクセス許可だけでなく、データの保護方法を決定する暗号化オペレーションでも特に重要です。

アクセス CreateKey 許可を制限する

キー (kms:CreateKey) を作成する許可を、それを必要とするプリンシパルにのみ付与します。KMS キーを作成するプリンシパルは、そのキーポリシーも設定するため、自分自身や他のユーザーに、作成した KMS キーを使用および管理するためのアクセス許可を付与できます。このアクセス許可を許可する場合は、 ポリシー条件を使用して制限することを検討してください。例えば、kms:KeySpec 条件を使用して、アクセス許可を対称暗号化 KMS キーに制限できます。

IAM ポリシーで KMS キーを指定する

ベストプラクティスとして、ポリシーステートメントの Resource 要素でアクセス許可が適用する各 KMS キーの キー ARN を指定します。この方法は、プリンシパルが必要とする KMS キーへのアクセス許可を制限します。例えば、この Resource 要素は、プリンシパルが使用する必要がある KMS キーのみを一覧表示します。

```
"Resource": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
]
```

KMS キーの指定が実用的でない場合は、arn:aws:kms:*region*:*account*:key/* のような信頼できる AWS アカウント およびリージョン内の KMS キーへのアクセスを制限する Resource 値を使用します。または、arn:aws:kms:*:*account*:key/* など、信頼できる AWS アカウント のすべてのリージョン (*) で、KMS キーへのアクセスを制限します。

キー ID、エイリアス名、エイリアス ARN を使用して、IAM ポリシーの Resource フィールド内の KMS キーを表すことはできません。エイリアス ARN を指定する場合、ポリシーは KMS キーではなく、エイリアスに適用されます。エイリアスの IAM ポリシーの詳細については、エイリアスへのアクセスの制御 を参照してください。

IAM ポリシーの「リソース」:「*」を避ける

ワイルドカード文字 (*) を慎重に使用してください。キーポリシーでは、Resource 要素のワイルドカード文字は、キーポリシーがアタッチされている KMS キーを表します。ただし、IAM ポリシーでは、Resource 要素 ("Resource": "*") 内のワイルドカード文字のみが、プリンシパルのアカウントが使用許可を持つすべての AWS アカウント で、すべての KMS キーにアクセス許可を適用します。これには、他の AWS アカウント の KMS キーと、プリンシパルのアカウントの KMS キーが含まれます。

例えば、別の AWS アカウント で KMS キーを使用するには、プリンシパルは、外部アカウントの KMS キーのキーポリシーのアクセス許可、および自分のアカウントの IAM ポリシーのアクセ

ス許可が必要です。任意のアカウントが自分の KMS キーに対する AWS アカウント kms:Decrypt アクセス権限を付与したとします。この場合、すべての KMS キー ("Resource": "*") に対するロール kms:Decrypt のアクセス許可を付与するアカウントの IAM ポリシーは、要件の IAM パートを満たします。その結果、そのロールを引き受けることができるプリンシパルは、信頼されていないアカウントの KMS キーを使用して、暗号テキストを復号できるようになります。オペレーションのエントリは、両方のアカウントの CloudTrail ログに表示されます。

特に、次の API オペレーションを許可するポリシーステートメントではを使用しないでください。 "Resource": "*"これらのオペレーションは、他の AWS アカウント のKMS キーで呼び出すことができます。

- DescribeKey
- GetKeyRotationStatus
- 暗号化オペレーション (暗号化 、復号 、、GenerateDataKey、GenerateDataKeyPair、GenerateDataKeyWithoutPlaintext、GenerateDataKe 名 、検証)
- CreateGrant, ListGrants, ListRetirableGrants, RetireGrant, RevokeGrant

「リソース」を使用する場合:「*」

IAM ポリシーでは、 Resource 要素でワイルドカード文字は、それを必要とするアクセス権限に対してのみ使用します。"Resource": "*" 要素が必要なのは、次の権限のみです。

- kms:CreateKey
- kms:GenerateRandom
- kms:ListAliases
- kms:ListKeys
- kms: CreateCustomKeyStoreや kms:ConnectCustomKeyStore などのカスタムキーストアのアクセス許可。

> **Note**
>
> エイリアスオペレーション (kms:CreateAlias、kms:UpdateAlias、kms:DeleteAlias) のアクセス許可は、エイリアスと KMS キーにアタッチする必要があります。IAM ポリシーで "Resource": "*" を使用し、エイリアスと KMS キーを表すことも、Resource エレメントでエイリアスと KMS キーを指定することもできます。例については、「エイリアスへのアクセスの制御」を参照してください。

このトピックの例では、KMS キーの IAM ポリシーを設計するための詳細情報とガイダンスを示します。一般的な AWS KMS のベストプラクティスガイダンスについては、AWS Key Management Service のベストプラクティス (PDF) を参照してください。すべての AWS リソースの IAM ベストプラクティスについては、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。

IAM ポリシーステートメントで KMS キーを指定する

IAM ポリシーを使用して、プリンシパルに KMS キーの使用または管理を許可できます。KMS キーは、ポリシーステートメントの `Resource` 要素で指定されます。

- IAM ポリシーステートメントで KMS キーを指定するには、キー ARN を使用する必要があります。キー ID、エイリアス名、エイリアス ARN を使用して IAM ポリシーステートメントの KMS キーを識別することはできません。

  例:「`Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab`」

  エイリアスに基づいて KMS キーへのアクセスを制御するには、kms:RequestAlias または kms:ResourceAliases 条件キーを使用します。詳細については、「AWS KMS の ABAC」を参照してください。

  エイリアス ARN は、、、CreateAliasUpdateAliasまたは などのエイリアスオペレーションへのアクセスを制御するポリシーステートメントでのみリソースとして使用しますDeleteAlias。詳細については、「エイリアスへのアクセスの制御」を参照してください。
- アカウントとリージョンで複数のKMS キーを指定するには、キー ARN のリージョンまたはリソース ID の位置にワイルドカード文字 (*) を使用します。

  例えば、アカウントの米国西部 (オレゴン) リージョンのすべての KMS キーを指定するには、「`Resource": "arn:aws:kms:us-west-2:111122223333:key/*`」を使用します。アカウントのすべてのリージョンですべての KMS キーを指定するには、「`Resource": "arn:aws:kms:*:111122223333:key/*`」を使用します。
- すべての KMS キーを表すには、ワイルドカード文字のみ ("*") を使用します。この形式は、特定の KMS キー、つまり CreateKey、、ListAliases、および GenerateRandomを使用しないオペレーションに使用しますListKeys。

ポリシーステートメントを書き込むときは、すべての KMS キーへのアクセス許可を付与するのではなく、プリンシパルが使用する必要がある KMS キーだけを指定するのがベストプラクティスです。

例えば、次の IAM ポリシーステートメントではDescribeKey、ポリシーステートメントのResource要素にリストされている KMS キーに対してのみGenerateDataKey、、、Decrypt オペレーションを呼び出すことをプリンシパルに許可します。キー ARN による KMS キーの指定は、アクセス許可を指定した KMS キーのみに制限するベストプラクティスです。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

特定の信頼できる AWS アカウント のすべての KMS キーにアクセス許可を適用するには、リージョンとキー ID の位置にワイルドカード文字 (*) を使用します。例えば、次のポリシーステートメントでは、プリンシパルが 2 つの信頼できるサンプルアカウントのすべての KMS キーで、指定されたオペレーションを呼び出すことができます。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyPair"
    ],
    "Resource": [
      "arn:aws:kms:*:111122223333:key/*",
      "arn:aws:kms:*:444455556666:key/*"
    ]
```

```
  }
}
```

Resource 要素内でワイルドカード文字 ("*") を単独で使用することもできます。アカウントが使用を許可されているすべての KMS キーへのアクセスを許可するため、主に、特定の KMS キーおよび Deny ステートメントを含まないオペレーションに推奨されます。また、機密性の低い読み取り専用オペレーションのみを許可するポリシーステートメントで使用することもできます。AWS KMS オペレーションが特定の KMS キーを含むかどうかを判断するには、the section called “アクセス許可に関するリファレンス” のテーブルの[Resources] (リソース) 列の [KMS key] (KMS キー) の値を探します。

例えば、次のポリシーステートメントでは、Deny エフェクトを使用して、プリンシパルが任意の KMS キーに対して指定されたオペレーションを使用できないようにします。Resource 要素でワイルドカード文字を使用して、すべての KMS キーを表します。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:CreateKey",
      "kms:PutKeyPolicy",
      "kms:CreateGrant",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

次のポリシーステートメントでは、ワイルドカード文字だけを使用してすべての KMS キーを表します。ただし、機密性の低い読み取り専用オペレーションと、特定の KMS キーに適用されないオペレーションのみを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:CreateKey",
      "kms:ListKeys",
      "kms:ListAliases",
```

```
      "kms:ListResourceTags"
    ],
    "Resource": "*"
  }
}
```

AWS KMS コンソールの使用に必要な許可

AWS KMS コンソールを使用して作業するユーザーには、AWS アカウント の AWS KMS リソースの使用を許可する一連の最小限のアクセス許可が必要です。これらの AWS KMS アクセス許可に加えて、ユーザーには、IAM ユーザーおよび IAM ロールを一覧表示するためのアクセス許可も必要です。これらの最小限必要なアクセス許可よりも制限された IAM ポリシーを作成した場合、AWS KMS コンソールは、その IAM ポリシーを使用するユーザーの意図したとおりには機能しません。

AWS KMS コンソールへの読み取り専用アクセスをユーザーに許可するために必要な最小限のアクセス権限については、「AWS KMS コンソールでの KMS キーの表示をユーザーに許可する」を参照してください。

ユーザーがコンソールを使用して KMS キーAWS KMSを作成および管理できるようにするには、次のセクションで説明するように、 AWSKeyManagementServicePowerUserマネージドポリシーをユーザーにアタッチします。

AWS SDK、AWS Command Line Interface、AWS Tools for PowerShell を使用して AWS KMS API を使用するユーザーに対して、最小限のコンソールアクセス許可を許可する必要はありません。ただし、API を使用する権限をこれらのユーザーに付与する必要があります。詳細については、「アクセス許可に関するリファレンス」を参照してください。

パワーユーザー用向けの AWS 管理ポリシー

`AWSKeyManagementServicePowerUser` マネージドポリシーを使用して、アカウントの IAM プリンシパルにパワーユーザーの許可を付与できます。パワーユーザーは KMS キーを作成し、作成した KMS キーを使用および管理し、すべての KMS キーと IAM アイデンティティを表示できます。`AWSKeyManagementServicePowerUser` マネージドポリシーを持つプリンシパルは、キーポリシー、他の IAM ポリシー、許可など、他のソースから許可を取得することもできます。

`AWSKeyManagementServicePowerUser` は AWS マネージド IAM ポリシーです。AWS マネージドポリシーの詳細については、IAM ユーザーガイドの「AWS マネージドポリシー」を参照してください。

> **Note**
>
> KMS キーに固有のこのポリシー内の許可は (`kms:TagResource` および `kms:GetKeyRotationStatus` など)、その KMS キーのキーポリシーが、キーへのアクセスを制御するために IAM ポリシーを使用することを AWS アカウント に対して明示的に許可している場合にのみ有効です。許可が KMS キーに固有のものであるかどうかを判断するには、AWS KMS アクセス許可 を表示して、[Resources] (リソース) 列にある [KMS key] (KMS キー) の値を確認します。
> このポリシーは、オペレーションを許可するキーポリシーを持つすべての KMS キーに対して、パワーユーザーの許可を付与します。クロスアカウントの許可の場合 (`kms:DescribeKey` および `kms:ListGrants` など)、これにより、信頼されていない AWS アカウント の KMS キーが含まれる可能性があります。詳細については、「IAM ポリシーのベストプラクティス」および「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。許可が他のアカウントの KMS キーに対して有効かどうかを判断するには、AWS KMS アクセス許可 を表示して、[Cross-account use] (クロスアカウントの使用) 列にある [Yes] (はい) の値を確認します。
> プリンシパルがエラーなしでAWS KMSコンソールを表示できるようにするには、プリンシパルにタグ：GetResources アクセス許可が必要ですが、これは`AWSKeyManagementServicePowerUser`ポリシーに含まれていません。この許可は、別の IAM ポリシーで許可できます。

AWSKeyManagementServicePower 管理 IAM ポリシーには以下のアクセス許可が含まれています。

- プリンシパルが KMS キーを作成できるようにします。このプロセスにはキーポリシーの設定が含まれるため、パワーユーザーは自分や他者に、自分が作成した KMS キーを使用および管理するためのアクセス許可を付与することができます。
- プリンシパルは、すべての KMS キーのエイリアスとタグを作成および削除できます。タグまたはエイリアスを変更すると、KMS キーを使用および管理するためのアクセス許可が、許可または拒否される場合があります。詳細については、「AWS KMS の ABAC」を参照してください。
- プリンシパルは、キー ARN、暗号化設定、キーポリシー、エイリアス、タグ、ローテーション状態など、すべての KMS キーに関する詳細情報を取得できます。
- プリンシパルが IAM ユーザー、グループ、ロールを一覧表示できるようにします。
- このポリシーでは、プリンシパルが自分で作成していない KMS キーを使用または管理することは許可できません。ただし、すべての KMS キーのエイリアスとタグを変更することはできるため、KMS キーを使用または管理する許可を許可または拒否できる可能性があります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:CreateAlias",
                "kms:CreateKey",
                "kms:DeleteAlias",
                "kms:Describe*",
                "kms:GenerateRandom",
                "kms:Get*",
                "kms:List*",
                "kms:TagResource",
                "kms:UntagResource",
                "iam:ListGroups",
                "iam:ListRoles",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM ポリシーの例

このセクションでは、さまざまな AWS KMS アクションのためのアクセス許可を付与する IAM ポリシーの例を示しています。

> **Important**
>
> 以下のポリシーのアクセス許可の一部は、KMS キーのキーポリシーも同じアクセス許可を付与する場合にのみ付与されます。詳細については、「アクセス許可に関するリファレンス」を参照してください。

JSON ポリシードキュメントの記述と書式設定については、『IAM ユーザーガイド』の「IAM JSON ポリシーリファレンス」を参照してください。

例

- AWS KMS コンソールでの KMS キーの表示をユーザーに許可する

- KMS キーの作成をユーザーに許可する
- 特定の AWS アカウント で KMS キーにより暗号化および復号することをユーザーに許可する
- 特定の AWS アカウント およびリージョンで KMS キーにより暗号化および復号することをユーザーに許可する
- 特定の KMS キーにより暗号化および復号することをユーザーに許可する
- KMS キーの無効化または削除を禁止する

AWS KMS コンソールでの KMS キーの表示をユーザーに許可する

以下の IAM ポリシーでは、ユーザーに AWS KMS コンソールへの読み取り専用アクセスを許可します。これらのアクセス許可を持つユーザーは、AWS アカウント 内のすべての KMS キーを表示できますが、KMS キーを作成　または変更することはできません。

AWS マネージドキー およびカスタマーマネージドキーページで KMS キーを表示するには、キーにタグやエイリアスがない場合でも、プリンシパルに kms:ListKeys、kms:ListAliases、および tag:GetResources のアクセス許可が必要です。KMS キーの詳細ページでオプションの KMS キーテーブル列とデータを表示するには、残りのアクセス許可、特に kms:DescribeKey が必要です。iam:ListUsers および iam:ListRoles アクセス許可は、キーポリシーをエラーなくデフォルトビューに表示するために必要です。カスタムキーストアページのデータとカスタムキーストアの KMS キーの詳細を表示するには、プリンシパルに kms:DescribeCustomKeyStores アクセス許可も必要です。

ユーザーのコンソールアクセスを特定の KMS キーに制限すると、コンソールは、表示されない各 KMS キーにエラーを表示します。

このポリシーには 2 つのポリシーステートメントが含まれます。最初のポリシーステートメントの `Resource` 要素は、例 AWS アカウント のすべてのリージョンのすべての KMS キーで指定されたアクセス許可を付与します。AWS KMS コンソールはプリンシパルのアカウントの KMS キーのみを表示するため、コンソールビューワーは追加のアクセスを必要としません。これは、他の AWS アカウント で KMS キーを表示するアクセス許可がある場合にも当てはまります。特定の KMS キーには `"Resource": "*"` 要素が適用されないため、残りの AWS KMS と IAM のアクセス許可には、 要素が必要です。

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

```
      "Sid": "ReadOnlyAccessForAllKMSKeysInAccount",
      "Effect": "Allow",
      "Action": [
        "kms:GetPublicKey",
        "kms:GetKeyRotationStatus",
        "kms:GetKeyPolicy",
        "kms:DescribeKey",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "tag:GetResources"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "ReadOnlyAccessForOperationsWithNoKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases",
        "iam:ListRoles",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

KMS キーの作成をユーザーに許可する

以下の IAM ポリシーは、ユーザーがすべてのタイプの KMS キーを作成することを許可します。`CreateKey` オペレーションは特定の AWS KMS リソース (KMS キーまたはエイリアス) を使用することがないため、`Resource` 要素の値は `*` になります。

ユーザーを特定のタイプの KMS キーに制限するには、kms:KeySpec、kms:KeyUsage、および kms:KeyOrigin 条件キーを使用します。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "kms:CreateKey",
    "Resource": "*"
  }
```

```
}
```

キーを作成するプリンシパルには、関連するパーミッションが必要な場合があります。

- kms:PutKeyPolicy — アクセスkms:CreateKey許可を持つプリンシパルは、KMS キーの初期キーポリシーを設定できます。ただし、CreateKey呼び出し元には KMS キーポリシーを変更できる kms:PutKeyPolicy アクセス許可が必要です。または、 の BypassPolicyLockoutSafetyCheckパラメータを指定する必要があります。CreateKeyこれは推奨されません。CreateKey 発信者は、IAM ポリシーから KMS キーの kms:PutKeyPolicy アクセス許可を取得することも、作成している KMS キーのキーポリシーにこのアクセス許可を含めることもできます。
- kms:TagResource — CreateKeyオペレーション中に KMS キーにタグを追加するには、CreateKey呼び出し元が IAM ポリシーで kms:TagResource アクセス許可を持っている必要があります。新しい KMS キーのキーポリシーにこのアクセス許可を含めるだけでは不十分です。ただし、CreateKey 発信者が初期キーポリシーの kms:TagResource を含む場合は、KMS キーの作成後に、個別の呼び出しでタグを追加できます。
- kms:CreateAlias — AWS KMSコンソールで KMS キーを作成するプリンシパルには、KMS キーとエイリアスに対する kms:CreateAlias アクセス許可が必要です。(コンソールは、へのコール CreateKey とへのコールを 2 つ行います CreateAlias)。IAM ポリシーでエイリアスアクセス許可を指定する必要があります。キーポリシーまたは IAM ポリシーで、KMS キーアクセス許可を付与できます。詳細については、「エイリアスへのアクセスの制御」を参照してください。

kms:CreateKey に加えて、次の IAM ポリシーでは AWS アカウント のすべての KMS キーに kms:TagResource アクセス許可を付与し、アカウントのすべてのエイリアスに kms:CreateAlias アクセス許可を付与します。また、IAM ポリシーでのみ提供できる便利な読み取り専用アクセス許可もいくつか含まれています。

この IAM ポリシーには、キーポリシーで設定できる kms:PutKeyPolicy アクセス権限やその他のアクセス権限は含まれていません。これらのアクセス許可は、1 つの KMS キーにのみ適用されるキーポリシーで設定するのがベストプラクティスです。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPermissionsForParticularKMSKeys",
      "Effect": "Allow",
      "Action": "kms:TagResource",
```

```
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPermissionsForParticularAliases",
      "Effect": "Allow",
      "Action": "kms:CreateAlias",
      "Resource": "arn:aws:kms:*:111122223333:alias/*"
    },
    {
      "Sid": "IAMPermissionsForAllKMSKeys",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey",
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```

特定の AWS アカウント で KMS キーにより暗号化および復号することをユーザーに許可する

次の IAM ポリシーでは、AWS アカウント 111122223333 の任意の KMS キーを使用して、ユーザーがデータを暗号化および復号できます。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/*"
  }
}
```

特定の AWS アカウント およびリージョンで KMS キーにより暗号化および復号することをユーザーに許可する

次の IAM ポリシーでは、米国西部 (オレゴン) リージョンの AWS アカウント 111122223333 の任意の KMS キーを使用して、ユーザーがデータを暗号化および復号できます。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/*"
    ]
  }
}
```

特定の KMS キーにより暗号化および復号することをユーザーに許可する

次の IAM ポリシーでは、 `Resource` エレメントで指定された 2 つの KMS キーを使用して、ユーザーがデータを暗号化および復号できます。IAM ポリシーステートメントで KMS キーを指定するときは、KMS キーの キー ARN を使用する必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
    ]
  }
}
```

KMS キーの無効化または削除を禁止する

次の IAM ポリシーでは、別の IAM ポリシーまたはキーポリシーで許可されている場合でも、ユーザーが KMS キーを無効化、削除できないようにします。アクセス権限を明示的に拒否するポリシーは、同じアクセス権限を明示的に付与するポリシーを含め、他のすべてのポリシーを上書きします。詳細については、「キーアクセスのトラブルシューティング」を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:DisableKey",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

AWS KMS でのグラント

グラントはポリシーツールであり、AWS プリンシパルに暗号化オペレーションでの KMS キーの使用を許可します。また、KMS キー (DescribeKey) を表示して、グラントの作成、管理をできるようにします。KMS キーへのアクセスを認可する際、グラントはキーポリシーおよび IAM ポリシーと共に考慮されます。グラントは、作成してそのアクセス許可を使用し、キーポリシーまたは IAM ポリシーを変更することなく削除できるため、一時的なアクセス許可としてよく使用されます。

グラントは、一般的に、AWS KMS と統合された AWS サービスによって使用され、保管中のデータを暗号化します。サービスは、アカウント内のユーザーの代わりにグラントを作成し、そのアクセス許可を使用して、タスクが完了するとすぐにグラント廃止にします。AWS のサービス方法、グラントの使用方法の詳細については、AWS のサービスで AWS KMS を使用する方法 またはサービスのユーザーガイドあるいはデベロッパーガイドの保管時の暗号化トピックを参照してください。

複数のプログラミング言語による、グラントのオペレーション方法を示すコード例については、「許可の使用」を参照してください。

トピック

- グラントについて
- グラントの概念

- AWS KMS グラントのベストプラクティス
- グラントの作成
- グラントの管理

グラントについて

権限は、きわめて柔軟で便利なアクセス制御メカニズムです。KMS キーのグラントを作成すると、このグラントは、グラントで指定されたすべての条件が満たされている場合に限り、被付与者プリンシパルが指定されたグラントオペレーションを KMS キーで呼び出すことを許可します。

- 各グラントでは、正確に 1 つの KMS キーにアクセスできます。異なる AWS アカウント でKMS キーのグラントを作成できます。
- グラントは KMS キーへのアクセスを許可できますが、アクセスを拒否することはできません。
- 各グラントは、それぞれ 1 つの被付与者プリンシパルを持っています。被付与者プリンシパルは、KMS キーと同じ AWS アカウント か異なるアカウントの、1 つまたは複数の ID を代表することができます。
- グラントは、グラントオペレーションのみを許可することができます。グラントオペレーションは、グラントの KMS キーによってサポートされている必要があります。サポートされていないオペレーションを指定すると、CreateGrantリクエストは `ValidationError` 例外で失敗します。
- 被付与者プリンシパルは、アクセス許可がキーポリシーまたは IAM ポリシーから付与された場合と同様に、グラントを指定せずに付与されたアクセス許可を使用できます。ただし、AWS KMS API は結果整合性モデルに従うので、グラントの作成、廃止、または取り消しを行うと、変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。権限でアクセス許可をすぐに使用するには、権限トークンを使用します。
- 認可されたプリンシパルはグラントを削除できます (グラントの廃止または失効)。グラントを削除すると、グラントが許可したすべてのアクセス許可が削除されます。グラントを取り消すために追加または削除するポリシーを特定する必要はありません。
- AWS KMS は、各 KMS キーのグラント数を制限します。詳細については、「KMS キーあたりのグラント: 50,000」を参照してください。

グラントを作成するとき、およびグラントを作成する許可を他のユーザーに付与するときは注意が必要です。許可を作成するアクセス許可にはセキュリティ上の影響があります。これは、kms:PutKeyPolicy ポリシーを設定するアクセス許可を許可するのと似ています。

- KMS キー (kms:CreateGrant) のグラントを作成するためのアクセス許可を持つユーザーは、グラントを使用してユーザーとロール (AWS のサービスを含む) に KMS キーの使用を許可します。プリンシパルは独自の AWS アカウント、または異なるアカウントや組織のアイデンティティにすることができます。
- グラントは、AWS KMS オペレーションのサブセットのみを許可します。グラントを使用して、プリンシパルに KMS キーの表示、暗号化オペレーションでの使用、グラントの作成、グラントの廃止を許可できます。詳細については、「グラントオペレーション」を参照してください。グラントの制約を使用して、対称暗号化キーに対するグラントにある許可を制限することもできます。
- プリンシパルはアクセス許可を取得して、キーポリシーまたは IAM ポリシーからグラントを作成できます。ポリシーから kms:CreateGrant アクセス許可を取得したプリンシパルは、KMS キーの任意の付与オペレーションの許可を作成できます。これらのプリンシパルは、キーに対して付与している許可を持っている必要はありません。ポリシーで kms:CreateGrant アクセス許可を許可する場合は、ポリシー条件を使用してこの許可を制限します。
- プリンシパルは、グラントからグラントを作成する許可を取得することもできます。これらのプリンシパルは、ポリシーからの他のアクセス許可を持っている場合でも、付与されたアクセス許可のみを委任できます。詳細については、「アクセス CreateGrant 許可の付与」を参照してください。

グラントに関する概念については、Grant terminology を参照してください。

グラントの概念

グラントを効果的に使用するには、AWS KMS が使用する用語と概念を理解する必要があります。

グラントの制約

グラントのアクセス許可を制限する条件。現在、AWS KMS は、暗号化オペレーションのリクエストで、暗号化コンテキストに基づいてグラントの制約をサポートしています。詳細については、「グラントの制約の使用」を参照してください。

グラント ID

KMS キーのグラントの一意の識別子。許可 ID とキー識別子 を使用して、 RetireGrantまたは RevokeGrantリクエストで許可を識別できます。

グラントオペレーション

グラントで許可できる AWS KMS オペレーションです。他のオペレーションを指定すると、CreateGrantリクエストは ValidationError 例外で失敗します。これらは、グラントトー

クンを承認するオペレーションでもあります。これらのアクセス許可の詳細については、AWS KMS アクセス許可 を参照してください。

これらのグラントオペレーションは、オペレーションを使用するアクセス許可を表します。したがって、ReEncrypt オペレーションの場合、ReEncryptFrom、ReEncryptTo、または両方の ReEncrypt* を指定できます。

グラントオペレーション:

- 暗号化オペレーション
  - Decrypt
  - 暗号化
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyPairWithoutPlaintext
  - GenerateDataKeyWithoutPlaintext
  - GenerateMac
  - ReEncryptFrom
  - ReEncryptTo
  - Sign
  - 検証
  - VerifyMac
- その他のオペレーション
  - CreateGrant
  - DescribeKey
  - GetPublicKey
  - RetireGrant

許可するグラントオペレーションは、許可の KMS キーでサポートされている必要があります。サポートされていないオペレーションを指定すると、CreateGrantリクエストは ValidationError 例外で失敗します。例えば、対称暗号化 KMS キーのグラントは、Sign、Verify、GenerateMac または VerifyMac オペレーションを許可できません。非対称 KMS キーのグラントは、データキーまたはデータキーペアを生成するいかなるオペレーションも許可できません。

グラントトークン

AWS KMS API は結果整合性モデルに従います。グラントを作成すると、変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。通常、変更がシステム全体に反映されるまでに数秒もかかりませんが、場合によっては数分かかることがあります。グラントがシステム全体に完全に伝播される前に使用しようとすると、アクセス拒否エラーが発生することがあります。グラントトークンを使用すると、グラントを参照し、グラントのアクセス許可をすぐに使用できます。

グラントトークンは、一意、非シークレット、可変長、base64 エンコードの、グラントを表す文字列です。グラントトークンを使用して、任意のグラントオペレーションでグラントを識別できます。ただし、トークン値はハッシュダイジェストであるため、グラントの詳細は明らかになりません。

グラントトークンは、グラントが AWS KMS 全体に完全に伝播されるまでにのみ使用されるように設計されています。その後、被付与者プリンシパルは、グラントトークンやその他のグラント限の証拠を提供することなく、グラントでアクセス許可を使用することができます。グラントトークンは常時使用できますが、グラントが最終的な整合性を取得した時点で、AWS KMS はグラントトークンではなく、グラントを使用してアクセス許可を決定します。

例えば、次のコマンドは GenerateDataKeyオペレーションを呼び出します。これは、グラントトークンを使用して、発信者 (被付与者プリンシパル) に、指定した KMS キーで `GenerateDataKey` を呼び出す許可を付与します。

```
$ aws kms generate-data-key \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --key-spec AES_256 \
        --grant-token $token
```

また、グラントトークンを使用して、グラントを管理するオペレーションでグラントを識別することもできます。例えば、使用停止プリンシパルは、 RetireGrantオペレーションの呼び出しでグラントトークンを使用できます。

```
$ aws kms retire-grant \
        --grant-token $token
```

`CreateGrant` は、グラントトークンを返す唯一のオペレーションです。グラントトークンは、他の AWS KMSオペレーションまたは CreateGrant オペレーションのCloudTrail ログイベントから取得することはできません。ListGrants および ListRetirableGrantsオペレーションはグラント ID を返しますが、グラントトークンは返しません。

詳細については、「グラントトークンを使用する」を参照してください。

被付与者プリンシパル

グラントで指定されたアクセス許可を取得する ID。各グラントはそれぞれ 1 つの被付与者プリンシパルを持っていますが、この被付与者プリンシパルは複数の ID を代表することができます。

被付与者プリンシパルは、AWS アカウント (ルート)、IAM ユーザー、IAM ロール、フェデレーティッドロールまたはユーザー、引き受けたロールユーザーなどの、任意の AWS プリンシパルとすることができます。被付与者プリンシパルは、KMS キーと同じアカウントか、別のアカウントにすることができます。ただし、被付与者プリンシパルを、サービスプリンシパル、IAM グループ、AWS 組織にすることはできません。

> **Note**
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

(グラント) を廃止にする

グラントを終了します。アクセス許可の使用が終了したら、グラントを廃止にします。

グラントの取り消しと使用停止のどちらも、グラント限を削除します。ただし、使用停止はグラントで指定されたプリンシパルによって行われます。通常、取り消しはキー管理者が行います。詳細については、「グラントの使用停止と取り消し」を参照してください。

プリンシパルを使用停止にする

グラントを廃止にするプリンシパル。グラントで使用停止プリンシパルを指定できますが、必須ではありません。使用停止プリンシパルは、AWS アカウント、IAM ユーザー、IAM ロール、フェデレーティッドユーザー、引き受けたロールユーザーを含む AWS プリンシパルのいずれかにすることができます。使用停止プリンシパルは、KMS キーと同じアカウントか、別のアカウントにすることができます。

> **Note**
>
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してくださ

い。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。

グラントで指定された廃止プリンシパルに加えて、グラントが作成された AWS アカウント によっても、グラントを廃止にできます。グラントで RetireGrant オペレーションが許可されている場合は、被付与者プリンシパルがグラントを廃止にできます。また、廃止プリンシパルの AWS アカウント または AWS アカウント は、グラントを廃止にする許可を、同じ AWS アカウント で IAM プリンシパルに委任できます。詳細については、「グラントの使用停止と取り消し」を参照してください。

(グラント) を取り消す

グラントを終了します。グラントを取り消して、グラントが許可するアクセス許可をアクティブに拒否します。

権限の取り消しと使用停止のどちらも、権限を削除します。ただし、使用停止はグラントで指定されたプリンシパルによって行われます。通常、取り消しはキー管理者が行います。詳細については、「グラントの使用停止と取り消し」を参照してください。

結果整合性 (グラント用)

AWS KMS API は結果整合性モデルに従います。グラントの作成、廃止、または取り消しを行うと、変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。通常、変更がシステム全体に反映されるまでに数秒もかかりませんが、場合によっては数分かかることがあります。

想定外のエラーが発生する場合は、この短い遅延に注意する必要があります。例えば、新しいグラントを管理したり、グラントが AWS KMS 全体で認識される前に新しいグラントを使用したりすると、アクセス拒否エラーが表示される可能性があります。グラントを廃止にするか取り消す場合でも、被付与者プリンシパルは、グラントが完全に削除されるまで、そのアクセス許可を短い期間使用できる可能性があります。典型的な戦略は、リクエスト、および自動バックオフと再試行ロジックを含む一部の AWS SDK を再試行することです。

AWS KMS には、この短い遅延を軽減する機能があります。

- 新しいグラントでアクセス許可をすぐに使用するには、グラントトークンを使用します。グラントトークンを使用して、任意のグラントのオペレーションのグラントを参照できます。手順については、「グラントトークンを使用する」を参照してください。
- CreateGrant オペレーションには、再試行オペレーションで重複する許可が作成されないようにする Nameパラメータがあります。

> **Note**
>
> サービスのすべてのエンドポイントが新しいグラント状態で更新されるまで、グラントトークンはグラントの有効性を優先します。ほとんどの場合、結果整合性は 5 分以内に取得されます。

詳細については、「AWS KMS の結果整合性」を参照してください。

AWS KMS グラントのベストプラクティス

AWS KMS では、グラントの作成、使用、管理を行う際に、以下のベストプラクティスを推奨します。

- グラントのアクセス許可を、被付与者プリンシパルに必要なアクセス許可に制限します。最小限の特権アクセスの原則を使用します。
- IAM ロールなどの特定の被付与者プリンシパルを使用し、被付与者プリンシパルに、必要な API オペレーションのみを使用するアクセス許可を付与します。
- 暗号化コンテキストのグラントの制約を使用して、発信者が意図した目的のために KMS キーを使用していることを保証します。リクエストで暗号化コンテキストを使用してデータを保護する方法の詳細については、 AWS セキュリティブログの「 AWS Key Management Serviceと を使用して暗号化されたデータの整合性を保護する方法 EncryptionContext」を参照してください。

> **Tip**
>
> 可能な限り、EncryptionContextEqualグラントの制約を使用します。EncryptionContextSubset グラントの制約は、正しく使用することがより困難です。使用する必要がある場合は、ドキュメントをよく読み、グラントの制約をテストして、意図したとおりに動作することを確認してください。

- 重複するグラントを削除します。重複するグラントには、同じキー ARN、API アクション、被付与者プリンシパル、暗号化コンテキスト、名前などがあります。元のグラントを廃止にしたか取り消したにも関わらず、重複したグラントが残った場合、残った重複グラントは意図しない特権エスカレーションを構成します。`CreateGrant` リクエストの再試行時にグラントが重複しないようにするには、`Name` パラメータを使用します。重複する権限を検出するには、 ListGrantsオペレーションを使用します。誤って重複するグラントを作成した場合は、できるだけ早く廃止にするか、取り消します。

> **Note**
>
> AWS マネージドキーのグラントは重複しているようにみえますが、異なる被付与者プリンシパルを持っています。
> 通常、ListGrants レスポンスの GranteePrincipal フィールドには、グラントの被付与者プリンシパルが含まれます。ただし、グラントの被付与者プリンシパルが AWS のサービスの場合、GranteePrincipal フィールドにはサービスプリンシパルが含まれます。これは、複数の異なる被付与者プリンシパルを表す場合があります。

- グラントは、自動的に期限切れにならないことに注意してください。アクセス許可が不要になったら、すぐにグラントの廃止または取り消しをします。削除されないグラントは、暗号化されたリソースに対してセキュリティ上のリスクを引き起こす可能性があります。

グラントの作成

グラントを作成する前に、グラントをカスタマイズするためのオプションを確認します。グラントの制約を使用して、グラントのアクセス許可を制限することができます。CreateGrant アクセス許可の付与も参照してください。グラントからグラントを作成する許可を取得したプリンシパルは、作成できるグラントが制限されます。

トピック

- グラントの作成
- グラントの制約の使用
- アクセス CreateGrant 許可の付与

グラントの作成

許可を作成するには、 CreateGrantオペレーションを呼び出します。KMS キー、被付与者プリンシパル、許可されたグラントオペレーションのリストを指定します。オプションの使用停止プリンシパルを指定することもできます。許可をカスタマイズするには、オプションの Constraints パラメータを使用して許可の制約を定義します。

グラントの作成、廃止、取り消しの際、変更が AWS KMS 全体で利用可能になるまで短い遅延が発生する場合があります (通常は 5 分未満)。詳細については、「結果整合性 (グラント用)」を参照してください。

例えば、次の CreateGrant コマンドは、keyUserRole ロールを引き受ける権限を持つユーザーに、指定された対称 KMS キーで Decrypt オペレーションを呼び出すことを許可するグラントを作成します。グラントでは、RetiringPrincipal パラメータを使用して、グラントを廃止できるプリンシパルを指定します。また、リクエスト内の暗号化コンテキストに "Department": "IT" が含まれている場合にのみアクセス許可を付与する許可制約も含まれます。

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

コードが CreateGrant オペレーションを再試行するか、AWS リクエストを自動的に再試行する SDK を使用する場合は、オプションの名前パラメータを使用して、重複グラントの作成を防止します。AWS KMS が同じプロパティのグラントの CreateGrant リクエストを、名前を含む既存のグラントとして取得する場合、リクエストは再試行として認識され、新しいグラントは作成されません。Name 値を使用して、任意の AWS KMS オペレーションでグラントを識別することはできません。

> **Important**
>
> グラント名には、機密情報や重要情報を含めないでください。 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

```
$ aws kms create-grant \
    --name IT-1234abcd-keyUserRole-decrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

複数のプログラミング言語による、権限のオペレーション方法を示すコード例については、許可の使用 を参照してください。

グラントの制約の使用

グラントの制約は、グラントが被付与者プリンシパルに付与するアクセス許可の条件を設定します。グラントの制約は、キーポリシーまたは IAM ポリシーで、条件キーの代わりになります。各グラントの制約値には、最大 8 個の暗号化コンテキストペアを含めることができます。各グラントの制約の暗号化コンテキスト値は、384 文字を超えることはできません。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

AWS KMS では、`EncryptionContextEquals` および `EncryptionContextSubset` の 2 つのグラントの制約をサポートします。どちらの制約も暗号化オペレーションのリクエストで、暗号化コンテキストの要件を確立します。

暗号化コンテキストのグラントの制約は、暗号化コンテキストパラメータを持つグラントオペレーションで使用されるように設計されています。

- 暗号化コンテキストの制約は、対称暗号化 KMS キーのグラントのみで有効です。他の KMS キーを使用する暗号化オペレーションは、暗号化コンテキストをサポートしません。
- 暗号化コンテキストの制約は、`DescribeKey` および `RetireGrant` オペレーションでは無視されます。`DescribeKey` および `RetireGrant` には暗号化コンテキストパラメータはありませんが、暗号化コンテキストの制約を持つグラントにこれらのオペレーションを含めることができます。
- `CreateGrant` オペレーションのグラントで、暗号化コンテキストの制約を使用することができます。暗号化コンテキストの制約では、`CreateGrant` 許可で作成された任意のグラントが、同様に厳密またはより厳密な暗号化コンテキストの制約を持っている必要があります。

AWS KMS は、次の暗号化コンテキストのグラントの制約をサポートします。

EncryptionContextEquals

`EncryptionContextEquals` を使用して、許可されたリクエストの正確な暗号化コンテキストを指定します。

EncryptionContextEquals では、リクエストの暗号化コンテキストペアが、グラントの制約の暗号化コンテキストペアと、大文字と小文字の区別で完全に一致することを要求します。このペアは任意の順序で表示できますが、各ペアのキーと値を変更することはできません。

例えば、EncryptionContextEquals のグラントの制約が "Department": "IT" 暗号化コンテキストペアを要求する場合、グラントはリクエストの暗号化コンテキストが完全に "Department": "IT" である場合にのみ、指定されたタイプのリクエストを許可します。

EncryptionContextSubset

EncryptionContextSubset を使用して、リクエストに特定の暗号化コンテキストペアを含めるように要求します。

EncryptionContextSubset では、リクエストにグラントの制約 (完全な大文字と小文字を区別する一致) のすべての暗号化コンテキストペアを含むことを要求しますが、リクエストが追加の暗号化コンテキストペアを持っている可能性もあります。このペアは任意の順序で表示できますが、各ペアのキーと値を変更することはできません。

例えば、EncryptionContextSubset のグラントの制約が、Department=IT の暗号化コンテキストペアを要求する場合、グラントはリクエストの暗号化コンテキストが "Department": "IT" の場合、またはリクエストが "Department": "IT","Purpose": "Test" のような他の暗号化コンテキストペアと共に "Department": "IT" を含む場合に、指定されたタイプのリクエストを許可します。

対称暗号化 KMS キーのグラントで暗号化コンテキストの制約を指定するには、 CreateGrantオペレーションで Constraintsパラメータを使用します。このコマンドが作成する許可では、keyUserRole ロールを引き受ける権限を持つユーザーに、Decrypt オペレーションを呼び出すためのアクセス許可を付与します。ただし、そのアクセス許可は Decrypt リクエストの暗号化コンテキストが、"Department": "IT" 暗号化コンテキストペアである場合にのみ有効です。

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextEquals={Department=IT}
```

したがって、グラントは以下のようになります。keyUserRole ロールに付与されるアクセス許可は、Decrypt リクエストがグラントの制約で指定された同じ暗号化コンテキストペアを使用する場合にのみ有効です。KMS キーの許可を検索するには、 ListGrantsオペレーションを使用します。

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "Decrypt"
            ],
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextEquals": {
                    "Department": "IT"
                }
            },
            "CreationDate": 1568565290.0,
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole"
        }
    ]
}
```

EncryptionContextEquals のグラントの制約を満たすために、Decrypt オペレーションのリクエストの暗号化コンテキストは、"Department": "IT" ペアである必要があります。被付与者プリンシパルからの次のようなリクエストは、EncryptionContextEquals のグラントの制約を満たします。

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab\
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT
```

グラントの制約が EncryptionContextSubset の場合、リクエストの暗号化コンテキストペアは、グラントの制約の暗号化コンテキストペアを含む必要がありますが、リクエストは他の暗号化コンテキストペアを含んでいる可能性もあります。次のグラントの制約は、リクエスト内の暗号化コンテキストペアの 1 つが "Deparment": "IT" であることを要求します。

```
"Constraints": {
   "EncryptionContextSubset": {
       "Department": "IT"
   }
}
```

被付与者プリンシパルからの次のリクエストは、この例の EncryptionContextEqual および EncryptionContextSubset のグラントの制約両方を満たします

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT
```

ただし、被付与者プリンシパルからの次のようなリクエストは、EncryptionContextSubset のグラントの制約を満たしますが、EncryptionContextEquals のグラントの制約は満たしません。

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT,Purpose=Test
```

AWS のサービスは、多くの場合、AWS アカウント で KMS キーを使用するためのアクセス許可を付与するグラントに、暗号化コンテキストの制約を使用します。例えば、Amazon DynamoDB では、次のようなグラントを使用して、アカウントで DynamoDB の AWS マネージドキー を使用するアクセス許可を取得します。このグラント内の EncryptionContextSubset のグラントの制約により、要求内の暗号化コンテキストに "subscriberID": "111122223333" と "tableName": "Services" ペアが含まれる場合にのみ、グラントの許可が有効になります。このグラントの制約は、DynamoDB が AWS アカウント の特定のテーブルに対してのみ、指定された KMS キーを使用することをグラントが許可することを意味します。

この出力を取得するには、アカウントの DynamoDB AWS マネージドキー用 で ListGrantsオペレーションを実行します。

```
$ aws kms list-grants --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
```

```
{
    "Grants": [
        {
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:tableName": "Services",
                    "aws:dynamodb:subscriberId": "111122223333"
                }
            },
            "CreationDate": 1518567315.0,
            "KeyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59"
        }
    ]
}
```

アクセス CreateGrant 許可の付与

グラントには、CreateGrant オペレーションを呼び出す許可を含めることができます。ただし、被付与者プリンシパルが CreateGrant を呼び出す許可をポリシーからではなくグラントから取得すると、その許可は制限されます。

- 被付与者プリンシパルは、親グラントで一部またはすべてのオペレーションを許可するグラントのみを作成できます。
- 作成されたグラントのグラントの制約は、少なくとも親グラントの制約と同じくらい厳密である必要があります。

これらの制限は、CreateGrant 許可をポリシーから取得するプリンシパルには適用されませんが、プリンシパルの許可はポリシー条件によって制限されます。

例えば、被付与者プリンシパルが GenerateDataKey、Decrypt、および CreateGrant オペレーションを呼び出せるようにする許可について考えてみます。CreateGrant 許可、親グラントを許可するグラントを呼び出します。

```
# The original grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "GenerateDataKey",
                "Decrypt",
                "CreateGrant
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

被付与者プリンシパル、exampleUser はこのアクセス許可を使用して、CreateGrant や Decrypt などの元のグラントで、指定されたオペレーションのサブセットを含むグラントを作成できます。子グラントに ScheduleKeyDeletion または ReEncrypt などの他のオペレーションを含めることはできません。

また、子グラントのグラントの制約は、親グラントの制約と同じか、より厳密である必要があります。例えば、子グラントは親グラントの EncryptionContextSubset 制約にペアを追加

できますが、削除することはできません。子グラントは EncryptionContextSubset 制約を EncryptionContextEquals 制約に変更することはできますが、その逆はできません。

例えば、被付与者プリンシパルは、親グラントから取得した CreateGrant 許可を使用して、次の子グラントを作成します。子グラントのオペレーションは、親グラントのオペレーションのサブセットであり、グラントの制約がより限定的です。

```
# The child grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572249600.0,
            "GrantId":
 "fedcba9999c1e2e9876abcde6e9d6c9b6a1987650000abcee009abcdef40183f",
            "Operations": [
                "CreateGrant"
                "Decrypt"
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/anotherUser",
            "Constraints": {
IAM best practices discourage the use of IAM users with long-term credentials. Whenever
                      possible, use IAM roles, which provide temporary credentials. For
 details,
                      see Security best practices in IAM in the IAM User Guide.
        "EncryptionContextEquals": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

子グラントの被付与者プリンシパル、anotherUser は CreateGrant 許可を使用してグラントを作成できます。ただし、anotherUser が作成したグラントは親グラントまたはサブセット内のオペレーションを含める必要があり、グラントの制約は同じか、より厳密である必要があります。

グラントの管理

必要な許可を持つプリンシパルは、グラントを表示、使用、削除 (廃止または取り消し) できます。グラントを作成および管理するための許可を絞り込むために、AWS KMS では、キーポリシーと IAM ポリシーで使用できる複数のポリシー条件をサポートします。

トピック

グラントへのアクセスを制御する

キーポリシー、IAM ポリシー、グラントで、グラントを作成および管理するオペレーションへのアクセスを制御できます。グラントから `CreateGrant` 許可を取得したプリンシパルは、より限定的な許可の付与を行います。

API オペレーション	キーポリシーまたは IAM ポリシー	権限
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
許可を使用停止にする	(制限あり。「グラントの使用停止と取り消し」を参照してください)	✓
RevokeGrant	✓	-

キーポリシーまたは IAM ポリシーを使用して、グラントを作成および管理するオペレーションへのアクセスを制御する際は、次の 1 つ以上のポリシー条件を使用して、アクセス許可を制限できます。AWS KMS は、グラントに関連する次のすべての条件キーをサポートします。詳細と例については、「AWS KMS 条件キー」を参照してください。

kms:GrantConstraintType

グラントに指定されたグラントの制約が含まれている場合にのみ、プリンシパルにグラントの作成を許可します。

kms:GrantIsForAWSResource

AWS KMS と統合された AWS のサービスがプリンシパルの代わりにリクエストを送信する場合にのみ、CreateGrant、ListGrants、RevokeGrant の呼び出しをプリンシパルに許可します。

kms:GrantOperations

プリンシパルにグラントの作成を許可しますが、グラントを指定されたオペレーションに制限します。

kms:GranteePrincipal

指定された被付与者プリンシパルに対してのみ、グラントの作成を許可します。

kms:RetiringPrincipal

グラントが特定の使用停止プリンシパルを指定する場合にのみ、プリンシパルにグラントの作成を許可します。

グラントの表示

許可を表示するには、 ListGrantsオペレーションを使用します。グラントを適用する KMS キーを指定する必要があります。グラント ID または被付与者プリンシパルにより、グラントリストをフィルタリングすることもできます。その他の例については、「許可の表示」を参照してください。

特定の使用停止プリンシパル を持つ AWS アカウントおよびリージョンのすべての許可を表示するには、 を使用しますListRetirableGrants。レスポンスには、各グラントの詳細が含まれます。

> **Note**
>
> 通常、ListGrants レスポンスの GranteePrincipal フィールドには、グラントの被付与者プリンシパルが含まれます。ただし、グラントの被付与者プリンシパルが AWS のサービスの場合、GranteePrincipal フィールドにはサービスプリンシパルが含まれます。これは、複数の異なる被付与者プリンシパルを表す場合があります。

例えば、次のコマンドは KMS キーのすべてのグラントを一覧表示します。

```
$  aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Operations": [
                "Decrypt"
            ]
        }
    ]
}
```

グラントトークンを使用する

AWS KMS API は結果整合性モデルに従います。グラントの作成時、グラントがすぐに有効にならないことがあります。変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。通常、変更がシステム全体に反映されるまでに数秒もかかりませんが、場合によっては数分かかることがあります。システム全体で変更が完全に伝播されると、被付与者プリンシパルはグラントトークンやグラントの証拠を指定せずに、グラントのアクセス許可を使用できます。ただし、グラントが新しく、一部の AWS KMS で認識されていない場合、リクエストは `AccessDeniedException` のエラーにより失敗する可能性があります。

新しいグラントでアクセス許可をすぐに使用するには、グラントのグラントトークンを使用します。CreateGrant オペレーションが返すグラントトークンを保存します。次に、AWS KMS オペレーションのリクエストでグラントトークンを送信します。グラントトークンを任意の AWS KMS グラントオペレーションへ送信できます。また、同じリクエストで複数のグラントトークンを送信できます。

次の例では、 CreateGrantオペレーションを使用して、 GenerateDataKeyおよび Decrypt オペレーションを許可する許可を作成します。これは、CreateGrant が token 変数で返すグラントトークンを保存します。次に、GenerateDataKey オペレーションへの呼び出しで、token 変数のグラントトークンを使用します。

```
# Create a grant; save the grant token
$ token=$(aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/appUser \
    --retiring-principal arn:aws:iam::111122223333:user/acctAdmin \
    --operations GenerateDataKey Decrypt \
    --query GrantToken \
    --output text)

# Use the grant token in a request
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --grant-tokens $token
```

アクセス許可を持つプリンシパルは、グラントが AWS KMS 全体で使用可能になる前であっても、グラントトークンを使用して新しいグラントを廃止にすることができます。(RevokeGrant オペレーションはグラントトークンを承認しません)。詳細については、「グラントの使用停止と取り消し」を参照してください。

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

グラントの使用停止と取り消し

グラントを削除するには、グラントの廃止または取り消しをします。

RetireGrant および RevokeGrantオペレーションは互いに非常によく似ています。どちらのオペレーションもグラントを削除します。これにより、グラントが許可しているアクセス許可が削除されます。これらのオペレーションの主な違いは、オペレーションの認可方法です。

RevokeGrant

ほとんどの AWS KMS オペレーションと同様に、RevokeGrant オペレーションへのアクセスは、キーポリシーおよび IAM ポリシーによって制御されます。RevokeGrant API は、 アクセスkms:RevokeGrant許可を持つ任意のプリンシパルによって呼び出すことができます。このア

クセス許可は、キー管理者に付与される標準のアクセス許可に含まれています。通常、管理者はグラントを取り消して、グラントが許可するアクセス許可を拒否します。

RetireGrant

グラントでは、グラントを廃止にできる管理者を決定できます。この設計により、キーポリシーや IAM ポリシーを変更することなく、グラントのライフサイクルを制御できます。通常、許可の使用を終了したら、グラントを廃止にします。

グラントは、グラントで指定されたオプションの廃止プリンシパルにより廃止にできます。被付与者プリンシパルもグラントを廃止にできますが、RetireGrant オペレーションを含むプリンシパルまたはグラントも同時に廃止にする場合に限られます。バックアップとして、グラントが作成された AWS アカウント もグラントを廃止にできます。

IAM ポリシーで使用できる kms:RetireGrant アクセス許可がありますが、ユーティリティは限られています。グラントで指定されたプリンシパルは、kms:RetireGrant アクセス許可なしでグラントを廃止にできます。kms:RetireGrant アクセス許可だけでは、プリンシパルにグラントの廃止を許可できません。kms:RetireGrant アクセス許可はキーポリシーでは無効です。

- グラントを廃止にするアクセス許可を拒否するには、kms:RetireGrant アクセス許可で Deny アクションを使用します。
- KMS キーを所有する AWS アカウント は、kms:RetireGrant 許可をアカウントの IAM プリンシパルに委任できます。
- 廃止プリンシパルが異なる AWS アカウント の場合、他のアカウントの管理者は kms:RetireGrant を使用して、そのアカウントの IAM プリンシパルに、グラントを廃止にするアクセス許可を委任できます。

AWS KMS API は結果整合性モデルに従います。グラントの作成、廃止、または取り消しを行うと、変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。通常、変更がシステム全体に反映されるまでに数秒もかかりませんが、場合によっては数分かかることがあります。新しいグラントをすぐに削除する必要がある場合は、グラントが AWS KMS 全体で利用可能になる前に、グラントトークンを使用してグラントを廃止にします。グラントトークンを使用してグラントを取り消すことはできません。

VPC エンドポイントを介した AWS KMS への接続

仮想プライベートクラウド (VPC) 内のプライベートインターフェイスエンドポイント経由で AWS KMS に直接接続することができます。インターフェイス VPC エンドポイントを使用する場合、VPC と AWS KMS 間の通信は完全に AWS ネットワーク内で行われます。

AWS KMS は、AWS PrivateLink を利用する Amazon Virtual Private Cloud (Amazon VPC) エンドポイントをサポートしています。各 VPC エンドポイントは、VPC サブネット内のプライベート IP アドレスを持つ 1 つ以上の Elastic Network Interfaces (ENI) で表されます。

インターフェイス VPC エンドポイントは VPC を AWS KMS に直接接続します。その際、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 接続を使用しません。VPC のインスタンスは、パブリック IP アドレスがなくても AWS KMS と通信できます。

リージョン

AWS KMS は、AWS KMS がサポートされているすべての AWS リージョン で、VPC エンドポイントと VPC エンドポイントポリシーをサポートしています。

トピック

- AWS KMS VPC エンドポイントに関する考慮事項
- AWS KMS 用の VPC エンドポイントの作成
- AWS KMS VPC エンドポイントへの接続
- VPC エンドポイントへのアクセスの制御
- ポリシーステートメントでの VPC エンドポイントの使用
- VPC エンドポイントのログ記録

AWS KMS VPC エンドポイントに関する考慮事項

AWS KMS 用のインターフェイス VPC エンドポイントをセットアップする前に、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントのプロパティと制限」のトピックを確認してください。

AWS KMS には、VPC エンドポイントをサポートするために以下の機能が用意されています。

- VPC エンドポイントを使用して、VPC からすべての AWS KMS API オペレーションを呼び出すことができます。
- AWS KMS リージョンエンドポイントまたは AWS KMS FIPS エンドポイントに接続するインターフェイス VPC エンドポイントを作成できます。
- AWS CloudTrail ログを使用して、VPC エンドポイントを介した KMS キーの使用を監査することができます。詳細については、「VPC エンドポイントのログ記録」を参照してください。

AWS KMS 用の VPC エンドポイントの作成

Amazon VPC コンソールまたは Amazon VPC API を使用して、AWS KMS 用の 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「インターフェイスエンドポイントを作成」を参照してください。

- AWS KMS 用の VPC エンドポイントを作成するには、次のサービス名を使用します。

  ```
  com.amazonaws.region.kms
  ```

  例えば、米国西部 (オレゴン) リージョン (us-west-2) では、サービス名は次のようになります。

  ```
  com.amazonaws.us-west-2.kms
  ```

- AWS KMS FIPS エンドポイントに接続する VPC エンドポイントを作成するには、次のサービス名を使用します。

  ```
  com.amazonaws.region.kms-fips
  ```

  例えば、米国西部 (オレゴン) リージョン (us-west-2) では、サービス名は次のようになります。

  ```
  com.amazonaws.us-west-2.kms-fips
  ```

VPC エンドポイントを使いやすくするために、VPC エンドポイントに対してプライベート DNS 名を有効にすることができます。[Enable DNS Name] (DNS 名を有効にする) オプションを選択すると、標準の AWS KMS DNS ホスト名が VPC エンドポイントに解決されます。例えば、`https://kms.us-west-2.amazonaws.com` はサービス名 `com.amazonaws.us-west-2.kms` に接続された VPC エンドポイントに解決されます。

このオプションにより VPC エンドポイントが使いやすくなります。AWS SDK および AWS CLI はデフォルトで標準の AWS KMS DNS ホスト名を使用するため、アプリケーションおよびコマンドで VPC エンドポイント URL を指定する必要はありません。

詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントを介したサービスへアクセスする」を参照してください。

AWS KMS VPC エンドポイントへの接続

AWS SDK、AWS CLI、または AWS Tools for PowerShell を使用して VPC エンドポイント経由で AWS KMS に接続できます。VPC エンドポイントを指定するには、DNS 名を使用します。

例えば、この list-keys コマンドは、 `endpoint-url` パラメータを使用して VPC エンドポイントを指定します。こうしたコマンドを使用するには、サンプルの VPC エンドポイント ID を、ご自身のアカウントのものに置き換えてください。

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcdf5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

VPC エンドポイントの作成時にプライベートホスト名を有効にした場合は、CLI コマンドまたはアプリケーションの設定で VPC エンドポイント URL を指定する必要はありません。標準の AWS KMS DNS ホスト名は VPC エンドポイントに解決されます。AWS CLI と SDK はデフォルトでこのホスト名を使用します。このため、VPC エンドポイントの使用を開始して、スクリプトやアプリケーションで何も変更することなく AWS KMS リージョンエンドポイントに接続できます。

プライベートホスト名を使用するには、VPC の `enableDnsHostnames` 属性と `enableDnsSupport` 属性を `true` に設定する必要があります。これらの属性を設定するには、ModifyVpcAttributeオペレーションを使用します。詳細については、「Amazon VPC ユーザーガイド」の「VPC の DNS 属性の表示と更新」を参照してください。

VPC エンドポイントへのアクセスの制御

AWS KMS の VPC エンドポイントへのアクセスを制御するには、VPC エンドポイントポリシーを VPC エンドポイントにアタッチします。エンドポイントポリシーは、プリンシパルが VPC エンドポイントを使用して AWS KMS リソースに対する AWS KMS オペレーションを呼び出すことができるかどうかを決定します。

エンドポイントの作成時に VPC エンドポイントポリシーを作成できます。また、VPC エンドポイントポリシーはいつでも変更できます。VPC マネジメントコンソール、または CreateVpcEndpointまたは ModifyVpcEndpointオペレーションを使用します。AWS CloudFormation テンプレートを使用して VPC エンドポイントポリシーを作成および変更することもできます。VPC マネジメントコンソールの使用方法については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントの作成」および「インターフェイスエンドポイントの変更」を参照してください。

> **Note**
>
> AWS KMS は、2020 年 7 月以降、VPC エンドポイントポリシーをサポートします。その日付以前に作成された AWS KMS の VPC エンドポイントには、デフォルトの VPC エンドポイントポリシーが設定されていますが、いつでも変更できます。

JSON ポリシードキュメントの記述と書式設定については、『IAM ユーザーガイド』の「IAM JSON ポリシーリファレンス」を参照してください。

トピック

- VPC エンドポイントポリシーについて
- デフォルトの VPC エンドポイントポリシー
- VPCエンドポイントポリシーの作成
- VPC エンドポイントポリシーの表示

VPC エンドポイントポリシーについて

VPC エンドポイントを使用する AWS KMS リクエストが成功するには、プリンシパルに 2 つのソースからのアクセス許可が必要です。

- キーポリシー、IAM ポリシー、権限が リソース (KMS キーまたはエイリアス) でオペレーションを呼び出すために、プリンシパルにアクセス許可を付与する必要があります。
- VPC エンドポイントポリシーは、エンドポイントを使用してリクエストを実行するためのアクセス権限をプリンシパルに付与する必要があります。

例えば、キーポリシーがプリンシパルに、特定の KMS キーで Decrypt を呼び出すためのアクセス許可を付与します。ただし、VPC エンドポイントポリシーは、プリンシパルがエンドポイントを使用して、その KMS キー で `Decrypt` を呼び出すことを許可しない場合があります。

または、VPC エンドポイントポリシーは、プリンシパルがエンドポイントを使用して特定の KMS キーDisableKeyで を呼び出すことを許可する場合があります。ただし、プリンシパルにキーポリシー、IAM ポリシー、または付与からのアクセス権限がない場合、リクエストは失敗します。

デフォルトの VPC エンドポイントポリシー

すべての VPC エンドポイントには VPC エンドポイントポリシーがありますが、ポリシーを指定する必要はありません。ポリシーを指定しない場合、デフォルトのエンドポイントポリシーでは、エンドポイント上のすべてのリソースのすべてのプリンシパルによるすべてのオペレーションが許可されます。

ただし、AWS KMS リソースでは、プリンシパルがキーポリシー、IAM ポリシー、権限からオペレーションを呼び出すアクセス許可も必要です。したがって、実際には、デフォルトポリシーでは、プリンシパルがリソースに対してオペレーションを呼び出す権限を持っている場合、エンドポイントを使用してオペレーションを呼び出すこともできます。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

許可されたオペレーションのサブセットのみに VPC エンドポイントを使用することをプリンシパルに許可するには、VPC エンドポイントポリシーを作成または変更します。

VPCエンドポイントポリシーの作成

VPC エンドポイントポリシーは、プリンシパルに VPC エンドポイントを使用してリソースに対してオペレーションを実行するアクセス許可があるかどうかを決定します。AWS KMS リソースでは、プリンシパルがキーポリシー、IAM ポリシー、権限からオペレーションを実行するアクセス許可も必要です。

各 VPC エンドポイントポリシーステートメントには、次の要素が必要です。

- アクションを実行できるプリンシパル
- 実行可能なアクション
- アクションを実行できるリソース

ポリシーステートメントは VPC エンドポイントを指定しません。代わりに、ポリシーがアタッチされているすべての VPC エンドポイントに適用されます。詳細については、「Amazon VPC ユーザーガイド」の「VPC エンドポイントでサービスへのアクセスを制御する」を参照してください。

AWS KMS の VPC エンドポイントポリシーの例を以下に示します。VPC エンドポイントにアタッチされると、このポリシーは ExampleUser に、VPC エンドポイントを使用して指定された KMS キーで指定されたオペレーションを呼び出すことを許可します。このようなポリシーを使用する前に、プリンシパルと キー ARN の例をアカウントの有効な値に置き換えてください。

```
{
   "Statement":[
      {
         "Sid": "AllowDecryptAndView",
         "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
         "Effect":"Allow",
         "Action": [
             "kms:Decrypt",
             "kms:DescribeKey",
             "kms:ListAliases",
             "kms:ListKeys"
          ],
         "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
   ]
}
```

AWS CloudTrail は、VPC エンドポイントを使用するすべてのオペレーションを記録します。ただし、 CloudTrail ログには、他のアカウントのプリンシパルによって要求されたオペレーションや、他のアカウントの KMS キーのオペレーションは含まれません。

そのため、外部アカウントのプリンシパルが VPC エンドポイントを使用してローカルアカウントの任意のキーで AWS KMS オペレーションを呼び出すことを阻止する、VPC エンドポイントポリシーを作成することもできます。

次の例では、aws:PrincipalAccount グローバル条件キーを使用して、プリンシパルがローカルアカウントに存在しない限り、すべての KMS キーに対するすべてのオペレーションのすべてのプリンシパルへのアクセスを拒否します。このようなポリシーを使用する前に、サンプルアカウント ID を有効なものに置き換えてください。

```
{
```

```
  "Statement": [
    {
      "Sid": "AccessForASpecificAccount",
      "Principal": {"AWS": "*"},
      "Action": "kms:*",
      "Effect": "Deny",
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": "111122223333"
        }
      }
    }
  ]
}
```

VPC エンドポイントポリシーの表示

エンドポイントの VPC エンドポイントポリシーを表示するには、VPC マネジメントコンソールまたは DescribeVpcEndpointsオペレーションを使用します。

次の AWS CLI コマンドは、指定した VPC エンドポイント ID を持つエンドポイントのポリシーを取得します。

このコマンドを使用する前に、サンプルのエンドポイント ID をアカウントの有効なものに置き換えてください。

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

ポリシーステートメントでの VPC エンドポイントの使用

リクエストが VPC から送信されたとき、または VPC エンドポイントを使用するときに、AWS KMS リソースとオペレーションへのアクセスを制御できます。このためには、キーポリシーまたは IAM ポリシーで、次のいずれかのグローバル条件キーを使用します。

- aws:sourceVpce 条件キーを使用して、VPC エンドポイントに基づいてアクセスを許可または制限します。
- aws:sourceVpc 条件キーを使用して、プライベートエンドポイントをホストする VPC に基づいてアクセスを許可または制限します。

> **Note**
>
> VPC エンドポイントに基づいてキーポリシーと IAM ポリシーを作成する場合は、注意が必要です。ポリシーステートメントによって、特定の VPC または VPC エンドポイントからリクエストが送信されるように要求されている場合は、ユーザーに代わって AWS KMS リソースを使用する統合 AWS のサービスからのリクエストが失敗する可能性があります。ヘルプについては、「AWS KMS アクセス許可を持つポリシーでの VPC エンドポイント条件の使用」を参照してください。
> また、リクエストが Amazon VPC エンドポイントから送信される場合、`aws:sourceIP` 条件キーは無効です。リクエストを VPC エンドポイントに制限するには、`aws:sourceVpce` または `aws:sourceVpc` 条件キーを使用します。詳細については、「AWS PrivateLink ガイド」の「VPC エンドポイントおよび VPC エンドポイントサービスの Identity and Access Management」を参照してください。

これらのグローバル条件キーを使用して、 AWS KMS keys (KMS キー) 、エイリアス、および特定のリソースに依存しCreateKeyない などのオペレーションへのアクセスを制御できます。

例えば、次のサンプルキーポリシーでは、リクエストが指定された VPC エンドポイントを使用する場合にのみ、KMS キーを使用してユーザーが一部の暗号化オペレーションを実行できます。ユーザーが AWS KMS へのリクエストを行うと、リクエストの VPC エンドポイント ID が、ポリシーの `aws:sourceVpce` 条件キーの値と比較されます。一致しない場合、要求は拒否されます。

このようなポリシーを使用するには、AWS アカウント ID と VPC エンドポイント ID のプレースホルダーを、アカウントの有効な値で置き換えます。

```
{
    "Id": "example-key-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM policies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["kms:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
            "Effect": "Deny",
```

```
            "Principal": "*",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": "vpce-1234abcdf5678c90a"
                }
            }
        }

    ]
}
```

また、aws:sourceVpc 条件キーを使用して、VPC エンドポイントが存在する VPC に基づいて、KMS キーへのアクセスを制限することもできます。

次のサンプルキーポリシーでは、コマンドが vpc-12345678 から送信される場合にのみ、KMS キーを管理するコマンドが許可されます。また、コマンドが vpc-2b2b2b2b から送信される場合にのみ、暗号化オペレーションで KMS キーを使用するコマンドが許可されます。ある VPC でアプリケーションが実行されていれば、このようなポリシーを使用できますが、管理機能のために 2 番目の切り離された VPC を使用します。

このようなポリシーを使用するには、AWS アカウント ID と VPC エンドポイント ID のプレースホルダーを、アカウントの有効な値で置き換えます。

```
{
    "Id": "example-key-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Allow administrative actions from vpc-12345678",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Create*","kms:Enable*","kms:Put*","kms:Update*",
                "kms:Revoke*","kms:Disable*","kms:Delete*",
                "kms:TagResource", "kms:UntagResource"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-12345678"
                }
            }
        },
        {
            "Sid": "Allow key usage from vpc-2b2b2b2b",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Encrypt","kms:Decrypt","kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "Allow read actions from everywhere",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "kms:Describe*","kms:List*","kms:Get*"
            ],
            "Resource": "*",
        }
    ]
}
```

VPC エンドポイントのログ記録

AWS CloudTrail は、VPC エンドポイントを使用するすべてのオペレーションを記録します。AWS KMS へのリクエストで VPC エンドポイントが使用されている場合、VPC エンドポイント ID は、そのリクエストが記録されている AWS CloudTrail ログのエントリに表示されます。このエンドポイント ID を使用して、AWS KMS VPC エンドポイントの使用状況を監査できます。

ただし、 CloudTrail ログには、他のアカウントのプリンシパルによって要求されたオペレーションや、他のアカウントの KMS キーおよびエイリアスに対するAWS KMSオペレーションのリクエスト

は含まれません。また、VPC を保護するために VPC エンドポイントポリシーによって拒否されたが、それ以外の場合は許可されるリクエストは AWS CloudTrail に記録されません。

例えば、次のサンプルログエントリには、VPC エンドポイントを使用した GenerateDataKey リクエストが記録されます。vpcEndpointId フィールドは、ログエントリの最後に表示されます。

```
{
  "eventVersion":"1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime":"2018-01-16T05:46:57Z",
  "eventSource":"kms.amazonaws.com",
  "eventName":"GenerateDataKey",
  "awsRegion":"eu-west-1",
  "sourceIPAddress":"172.01.01.001",
  "userAgent":"aws-cli/1.14.23 Python/2.7.12 Linux/4.9.75-25.55.amzn1.x86_64
 botocore/1.8.27",
  "requestParameters":{
    "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab",
    "numberOfBytes":128
  },
  "responseElements":null,
  "requestID":"a9fff0bf-fa80-11e7-a13c-afcabff2f04c",
  "eventID":"77274901-88bc-4e3f-9bb6-acf1c16f6a7c",
  "readOnly":true,
  "resources":[{
    "ARN":"arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId":"111122223333",
    "type":"AWS::KMS::Key"
  }],
  "eventType":"AwsApiCall",
  "recipientAccountId":"111122223333",
  "vpcEndpointId": "vpce-1234abcdf5678c90a"
}
```

の条件キー AWS KMS

AWS KMS リソースへのアクセスを制御するキーポリシーと IAM ポリシーで条件を指定できます。ポリシーステートメントは、条件が true の場合にのみ有効です。例えば、特定の日付の後にのみ適用されるポリシーステートメントが必要になる場合があります。また、特定の値が API リクエストに表示されているときにのみ、ポリシーステートメントによるアクセスコントロールが必要になる場合もあります。

条件を指定するには、IAM 条件演算子を指定して、ポリシーステートメントの Condition 要素で条件キーを使用します。一部の条件キーは一般的に に適用 AWSされ、その他の条件キーは に固有です AWS KMS。

条件キーの値は、 AWS KMS キーポリシーと IAM ポリシーの文字ルールとエンコーディングルールに従う必要があります。キーポリシードキュメントのルールに関する詳細については、「キーポリシー形式」を参照してください。IAM ポリシードキュメントのルールに関する詳細については、「IAM ユーザーガイド」の「IAM 名前の要件」を参照してください。

トピック

- AWS グローバル条件キー
- AWS KMS 条件キー
- AWS KMSAWS Nitro Enclaves の 条件キー

AWS グローバル条件キー

AWS は、アクセスコントロールに IAM を使用するすべての AWS サービスのポリシー条件キーのセットであるグローバル条件キーを定義します。 は、すべてのグローバル条件キー AWS KMS をサポートします。 AWS KMS キーポリシーと IAM ポリシーで使用できます。

例えば、aws:PrincipalArn グローバル条件キーを使用して、 AWS KMS key リクエストのプリンシパルが条件キーの値で Amazon リソースネーム (ARN) で表される場合にのみ、 (KMS キー) へのアクセスを許可できます。で属性ベースのアクセスコントロール (ABAC) をサポートするには AWS KMS、IAM ポリシーで aws:ResourceTag/tag-key グローバル条件キーを使用して、特定のタグを持つ KMS キーへのアクセスを許可できます。

プリンシパルが AWS サービスAWS プリンシパルであるポリシーで サービスが混乱した代理として使用されないようにするには、 aws:SourceArnまたは aws:SourceAccount グローバル条件キーを使用できます。詳細については、「aws:SourceArn または aws:SourceAccount 条件キーの使用」を参照してください。

使用可能なリクエストのタイプなど、 AWS グローバル条件キーの詳細については、IAM AWS ユーザーガイドの「 グローバル条件コンテキストキー」を参照してください。IAM ポリシーでグローバル条件キーを使用する例については、IAM ユーザーガイドのリクエストへのアクセスの制御およびタグキーの制御を参照してください。

以下のトピックでは、IP アドレスと VPC エンドポイントに基づく条件キーを使用するための特別なガイダンスを提供します。

トピック

- AWS KMS アクセス許可を持つポリシーでの IP アドレス条件の使用
- AWS KMS アクセス許可を持つポリシーでの VPC エンドポイント条件の使用

AWS KMS アクセス許可を持つポリシーでの IP アドレス条件の使用

を使用して AWS KMS 、統合 AWS サービス 内のデータを保護できます。ただし、 へのアクセスを許可または拒否する同じポリシーステートメントで IP アドレス条件演算子または `aws:SourceIp`条件キーを指定する場合は注意が必要です AWS KMS。例えば、AWS: 送信元 IP AWS に基づいて へのアクセスを拒否するのポリシーでは、指定された IP 範囲からのリクエストに AWS アクションを制限します。

次のシナリオを考えてみます。

1. AWS「: 送信元 IP AWS に基づいて へのアクセスを拒否する」に示すようなポリシーを IAM アイデンティティにアタッチします。`aws:SourceIp` 条件キーの値は、ユーザーの会社の IP アドレス範囲に設定します。この IAM アイデンティティには、Amazon EBS、Amazon EC2、 AWS KMSの使用を許可する他のポリシーがアタッチされています。
2. アイデンティティは、暗号化された EBS ボリュームを EC2 インスタンスにアタッチしようとします。このアクションは、関連するすべてのサービスを使用するためのアクセス権限がユーザーに付与されているにもかかわらず、承認エラーが発生して失敗します。

ボリュームの暗号化されたデータキーを復号 AWS KMS する へのリクエストが Amazon EC2 インフラストラクチャに関連付けられている IP アドレスから送信されるため、ステップ 2は失敗します。成功させるには、リクエストは、元のユーザーの IP アドレスからリクエストが送られてこなければなりません。ステップ 1 のポリシーでは、指定されたもの以外の IP アドレスからのすべてのリクエストが明示的に拒否されるため、Amazon EC2 は EBS ボリュームの暗号化されたデータキーを復号化する権限を拒否されます。

また、リクエストが Amazon VPC エンドポイントから送信される場合、aws:sourceIP 条件キーは無効です。リクエストを AWS KMS VPC エンドポイントなどの VPC エンドポイントに制限するには、aws:sourceVpce または aws:sourceVpc 条件キーを使用します。詳細については、Amazon VPC ユーザーガイドの「VPC エンドポイント - エンドポイントの使用の管理」を参照してください。

AWS KMS アクセス許可を持つポリシーでの VPC エンドポイント条件の使用

は、 を使用するAWS KMS Amazon Virtual Private Cloud (Amazon VPC) エンドポイントをサポートします。 AWS PrivateLinkキーポリシーと IAM ポリシーで次のグローバル条件キーを使用して、リクエストが VPC から送信されたとき、または VPC エンドポイントを使用するときに AWS KMS リソースへのアクセスを制御できます。詳細については、「ポリシーステートメントでの VPC エンドポイントの使用」を参照してください。

- aws:SourceVpc は、指定した VPC からのリクエストにアクセスを制限します。
- aws:SourceVpce は、指定した VPC エンドポイントからのリクエストにアクセスを制限します。

これらの条件キーを使用して KMS キーへのアクセスを制御すると、 が AWS KMS ユーザーに代わって使用する AWS サービスへのアクセスを誤って拒否する可能性があります。

IP アドレス条件キーの例のような状況にならないように注意してください。KMS キーのリクエストを VPC または VPC エンドポイントに制限すると、Amazon S3 や Amazon EBS などの統合サービス AWS KMS からの の呼び出しが失敗する可能性があります。ソースリクエストの最初の送信元が VPC 内または VPC エンドポイントであっても、これは発生することがあります。

AWS KMS 条件キー

AWS KMS には、キーポリシーと IAM ポリシーで使用できる一連の条件キーが用意されています。これらの条件キーは に固有です AWS KMS。例えば、kms:EncryptionContext:*context-key* 条件キーは、対称暗号化 KMS キーへのアクセスを制御するときに特定の暗号化コンテキストを要求するために使用できます。

API オペレーションリクエストの条件

多くの AWS KMS 条件キーは、 AWS KMS オペレーションのリクエスト内のパラメータの値に基づいて KMS キーへのアクセスを制御します。例えば、IAM ポリシーで kms:KeySpec 条件キーを使用して、CreateKeyリクエストの KeySpecパラメータの値が の場合にのみ CreateKeyオペレーションの使用を許可できますRSA_4096。

このタイプの条件は、パラメータのデフォルト値を使用する場合など、リクエストにパラメータが表示されない場合でも機能します。たとえば、kms:KeySpec 条件キーを使用すると、KeySpec パラメータの値が SYMMETRIC_DEFAULT (デフォルト値) の場合にのみ CreateKey オペレーションを使用できるようになります。この条件では、SYMMETRIC_DEFAULT 値を持つ KeySpec パラメータを持つリクエストと、KeySpec パラメータを持たないリクエストが許可されます。

API オペレーションで使用される KMS キーの条件

一部の AWS KMS 条件キーは、オペレーションで使用される KMS キーのプロパティに基づいて、オペレーションへのアクセスを制御できます。例えば、kms:KeyOrigin 条件を使用して、KMS キーOriginの が である場合にのみ、プリンシパルが KMS キーGenerateDataKeyで を呼び出せるようにしますAWS_KMS。この方法で条件キーを使用できるかどうかを確認するには、条件キーの説明を参照してください。

このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。などの特定の KMS キーリソースに対して許可されていないオペレーションでこのタイプの条件キーを使用する場合、条件が満たされないためListKeys、アクセス許可は有効ではありません。ListKeys オペレーションの認可に関与する KMS キーリソースおよび KeySpec プロパティはありません。

以下のトピックでは、各 AWS KMS 条件キーについて説明し、ポリシー構文を示すポリシーステートメントの例を示します。

条件キーで集合演算子を使用する

ポリシー条件が、リクエスト内のタグのセットやポリシー内のタグのセットなど、2 つの値のセットを比較する場合、セットを比較する AWS 方法を に指示する必要があります。この目的のために、IAM は、2 つの集合演算子、ForAnyValue および ForAllValues を定義します。集合演算子は、それらを必要とする複数値を持つ条件キーでのみ使用します。単一値の条件キーで集合演算子を使用しないでください。ポリシーステートメントは必ず、本稼働環境での使用前に完全にテストしてください。

条件キーは単一値または複数値です。 AWS KMS 条件キーが単一値か複数値かを判断するには、条件キーの説明の「値のタイプ」列を参照してください。

- 単一値の条件キーは認可コンテキスト (リクエストまたはリソース) に、最大で 1 つの値を持ちます。例えば、各 API コールは 1 つの からしか発生できないため AWS アカウント、kms:CallerAccount は単一の値条件キーです。単一値の条件キーで集合演算子を使用しないでください。

- 複数値の条件キーでは、認可コンテキスト (リクエストまたはリソース) に複数の値があります。例えば、各 KMS キーは複数のエイリアスを持つことができるため、kms:ResourceAliases は複数の値を持つことができます。複数値の条件キーには集合演算子が必要です。

単一値と複数値の条件キーの違いは、ポリシー条件の値の数ではなく、認可コンテキストの値の数であることに注意してください。

> **Warning**
>
> 単一値の条件キーで集合演算子を使用すると、過度に許可される (または過度に制限される) ポリシーステートメントが作成される可能性があります。集合演算子は、複数値の条件キーでのみ使用してください。
>
> kms::EncryptionContextcontext-key または `aws:RequestTag/tag-key`条件キーを持つ`ForAllValues`集合演算子を含むポリシーを作成または更新すると、 は次のエラーメッセージ AWS KMS を返します。
>
> ```
> OverlyPermissiveCondition: Using the ForAllValues set operator with
> a single-valued condition key matches requests without the specified
> [encryption context or tag] or with an unspecified [encryption
> context or tag]. To fix, remove ForAllValues.
> ```

`ForAnyValue` および `ForAllValues` 集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。`ForAllValues` 集合演算子を単一値条件で使用するリスクについては、IAM ユーザーガイドの「単一値キー ForAllValues によるセキュリティ警告」を参照してください。

トピック

- kms:BypassPolicyLockoutSafetyCheck
- kms:CallerAccount
- kms:CustomerMasterKeySpec (非推奨)
- kms:CustomerMasterKeyUsage (非推奨)
- kms:DataKeyPairSpec
- kms:EncryptionAlgorithm
- kms:EncryptionContext：context-key
- kms:EncryptionContextKeys

- kms:ExpirationModel
- kms:GrantConstraintType
- kms:GrantIsForAWSResource
- kms:GrantOperations
- kms:GranteePrincipal
- kms:KeyOrigin
- kms:KeySpec
- kms:KeyUsage
- kms:MacAlgorithm
- kms:MessageType
- kms:MultiRegion
- kms:MultiRegionKeyType
- kms:PrimaryRegion
- kms:ReEncryptOnSameKey
- kms:RequestAlias
- kms:ResourceAliases
- kms:ReplicaRegion
- kms:RetiringPrincipal
- kms:ScheduleKeyDeletionPendingWindowInDays
- kms:SigningAlgorithm
- kms:ValidTo
- kms:ViaService
- kms:WrappingAlgorithm
- kms:WrappingKeySpec

kms:BypassPolicyLockoutSafetyCheck

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
`kms:BypassPolicyLo`	ブール値	単一値	`CreateKey`	IAM ポリシーのみ

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
ckoutSafetyCheck			PutKeyPolicy	キーポリシーと IAM ポリシー

kms:BypassPolicyLockoutSafetyCheck 条件キーは、リクエスト内の BypassPolicyLockoutSafetyCheckパラメータの値に基づいて、 CreateKeyおよび PutKeyPolicyオペレーションへのアクセスを制御します。

次の IAM ポリシーステートメントの例では、CreateKey リクエストの BypassPolicyLockoutSafetyCheck パラメータ値が true. の場合に、KMS キーを作成するアクセス許可を拒否することで、ユーザーがポリシーのロックアウト安全チェックを回避できないようにします。

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

IAM ポリシーまたはキーポリシーで kms:BypassPolicyLockoutSafetyCheck 条件キーを使用して、 PutKeyPolicy オペレーションへのアクセスを制御することもできます。次のキーポリシーのポリシーステートメントの例では、KMS キーのポリシーを変更する際に、ユーザーがポリシーのロックアウト安全チェックを回避できないようにします。

このポリシーステートメントでは、明示的に Deny を使用せずに、Null 条件演算子とともに Allow を使用し、リクエストに BypassPolicyLockoutSafetyCheck パラメータが含まれていない場合にのみアクセスを許可します。パラメータが使用されていない場合、デフォルト値は false です。この弱いポリシーステートメントは、バイパスが必要な限定された状況では上書きされる場合があります。

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

以下の資料も参照してください。

- kms:KeySpec
- kms:KeyOrigin
- kms:KeyUsage

kms:CallerAccount

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:CallerAccount	文字列	単一値	KMS キーリソースのオペレーション カスタムキーストアのオペレーション	キーポリシーと IAM ポリシー

この条件キーを使用して、 AWS アカウントのすべてのアイデンティティ (ユーザーおよびロール) へのアクセスを許可または拒否できます。キーポリシーでは、Principal 要素を使って、ポリシーステートメントが適用される ID を指定できます。Principal 要素の構文では、 AWS アカウントのすべてのアイデンティティを指定することはできません。ただし、この条件キーをすべての AWS ID を指定する Principal要素と組み合わせることで、この効果を実現できます。

これを使用して、任意の KMS キーリソースオペレーション 、つまり特定の KMS キーを使用する任意の AWS KMS オペレーションへのアクセスを制御できます。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。また、カスタムキーストアを管理するオペレーションにも有効です。

例えば、次のポリシーステートメントは、kms:CallerAccount 条件キーを使用する方法を示します。このポリシーステートメントは、Amazon EBS の AWS マネージドキー のキーポリシーにあります。すべての AWS ID を指定する Principal要素と kms:CallerAccount条件キーを組み合わせて、 AWS アカウント 111122223333 のすべての ID へのアクセスを効果的に許可します。これには、Amazon EBS を通過するリクエストのみを許可することで、アクセス許可をさらに制限するための追加の AWS KMS 条件キー (kms:ViaService) が含まれています。詳細については、「kms:ViaService」を参照してください。

```
{
  "Sid": "Allow access through EBS for all principals in the account that are
 authorized to use EBS",
  "Effect": "Allow",
  "Principal": {"AWS": "*"},
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "111122223333",
      "kms:ViaService": "ec2.us-west-2.amazonaws.com"
    }
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

kms:CustomerMasterKeySpec (非推奨)

kms:CustomerMasterKeySpec 条件キーは非推奨です。代わりに、kms:KeySpec 条件キーを使用します。

kms:CustomerMasterKeySpec および kms:KeySpec 条件キーは同じように機能します。名前だけが異なります。kms:KeySpec を使用することをお勧めします。ただし、重大な変更を避けるために、 は両方の条件キー AWS KMS をサポートします。

kms:CustomerMasterKeyUsage (非推奨)

kms:CustomerMasterKeyUsage 条件キーは非推奨です。代わりに、kms:KeyUsage 条件キーを使用します。

kms:CustomerMasterKeyUsage および kms:KeyUsage 条件キーは同じように機能します。名前だけが異なります。kms:KeyUsage を使用することをお勧めします。ただし、重大な変更を避けるために、 は両方の条件キー AWS KMS をサポートします。

kms:DataKeyPairSpec

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:DataKeyPairSpec	文字列	単一値	GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエストの KeyPairSpecパラメータの値に基づいて GenerateDataKeyPairおよび GenerateDataKeyPairWithoutPlaintextオペレーションへのアクセスを制御できます。例えば、特定のタイプのデータキーペアのみを生成することをユーザーに許可できます。

次のキーポリシーステートメントの例では、kms:DataKeyPairSpec 条件キーを使用して、ユーザーが KMS キーを使用して RSA データキーペアのみを生成できるようにします。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
```

```
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

以下の資料も参照してください。

- kms:KeySpec
- the section called “kms:EncryptionAlgorithm”
- the section called “kms:EncryptionContext：context-key”
- the section called “kms:EncryptionContextKeys”

kms:EncryptionAlgorithm

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:EncryptionAlgorithm	文字列	単一値	Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext	キーポリシーと IAM ポリシー

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
			GenerateDataKeyWithoutPlaintext ReEncrypt	

kms:EncryptionAlgorithm 条件キーを使用して、オペレーションで使用される暗号化アルゴリズムに基づいて暗号化オペレーションへのアクセスを制御できます。Encrypt 、Decrypt 、および ReEncryptオペレーションでは、リクエスト内の EncryptionAlgorithmパラメータの値に基づいてアクセスを制御します。データキーとデータキーペアを生成するオペレーションでは、データキーの暗号化に使用される暗号化アルゴリズムに基づいてアクセスを制御します。

この条件キーは AWS KMS、 の外部の非対称 KMS キーペアのパブリックキーによる暗号化など、の外部で実行されるオペレーションには影響しません AWS KMS。

EncryptionAlgorithm リクエスト内の パラメータ

ユーザーが KMS キーで特定の暗号化アルゴリズムのみを使用できるようにするには、Deny 効果と StringNotEquals 条件演算子を含むポリシーステートメントを使用します。例えば、以下のキーポリシーステートメント例は、リクエスト内の暗号化アルゴリズムが RSAES_OAEP_SHA_256 (RSA KMS キーで使用される非対称暗号化アルゴリズム) 以外の場合に、ExampleRole ロールを引き受けることができるプリンシパルが、指定された暗号化オペレーションでこの KMS キーを使用することを禁止します。

ユーザーが特定の暗号化アルゴリズムを使用できるようにするポリシーステートメントとは異なり、このような二重否定を持つポリシーステートメントは、この KMS キーに対する他のポリシーおよび権限によって、このロールが他の暗号化アルゴリズムを使用することを防止します。このポリシーステートメントの Deny は、 Allow 効果を持つキーポリシーまたは IAM ポリシーよりも優先され、この KMS キーおよびそのプリンシパルのすべての権限よりも優先されます。

```
{
  "Sid": "Allow only one encryption algorithm with this asymmetric KMS key",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
```

```
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "RSAES_OAEP_SHA_256"
    }
  }
}
```

オペレーションに使用される暗号化アルゴリズム

kms:EncryptionAlgorithm 条件キーを使用して、リクエストでアルゴリズムが指定されていない場合でも、オペレーションで使用される暗号化アルゴリズムに基づいてオペレーションへのアクセスを制御することもできます。これにより、デフォルト値のためリクエストで指定されない可能性のある SYMMETRIC_DEFAULT アルゴリズムを要求または禁止することができます。

この機能により、kms:EncryptionAlgorithm 条件キーを使用して、データキーとデータキーペアを生成するオペレーションへのアクセスを制御することができます。これらのオペレーションは、対称暗号化 KMS キーと SYMMETRIC_DEFAULT アルゴリズムのみを使用します。

例えば、この IAM ポリシーは、プリンシパルを対称暗号化に制限します。リクエストで指定された、またはオペレーションで使用される暗号化アルゴリズムが SYMMETRIC_DEFAULT でない限り、暗号化オペレーションのサンプルアカウントにある KMS キーへのアクセスを拒否します。を含めるとGenerateDataKey、GenerateDataKeyWithoutPlaintext、、GenerateDataKeyPair、およびGenerateDataKeyPairWithoutPlaintextがアクセス許可GenerateDataKey*に追加されます。この条件は、常に対称暗号化アルゴリズムを使用するため、これらのオペレーションには影響しません。

```
{
  "Sid": "AllowOnlySymmetricAlgorithm",
  "Effect": "Deny",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
  ],
```

```
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
    }
  }
}
```

以下の資料も参照してください。

- the section called “kms:MacAlgorithm”
- kms:SigningAlgorithm

kms:EncryptionContext：context-key

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ

`kms:EncryptionContext: context-key`	文字列	単一値	`CreateGrant`
			`Encrypt`
			`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateDataKeyPairWithoutPlaintext`
			`GenerateDataKeyWithoutPlaintext`
				キーポリシーと IAM ポリシー

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
			ReEncrypt	

kms:EncryptionContext:*context-key* 条件キーを使用して、対称暗号化 KMS キーへのアクセスを、暗号化オペレーションに対するリクエスト内の暗号化コンテキストに基づいて制御することができます。暗号化コンテキストペアのキーと値の両方を評価するには、この条件キーを使用します。暗号化コンテキストキーのみを評価するか、キーや値に関係なく暗号化コンテキストを要求するには、kms:EncryptionContextKeys 条件キーを使用します。

> **Note**
>
> 条件キーの値は、 キーポリシーと IAM ポリシーに関する文字ルールに従う必要があります。暗号化コンテキストで有効な文字の中には、ポリシーでは有効にならないものもあります。この条件キーは、すべての有効な暗号化コンテキスト値を表現するために使用できない場合があります。キーポリシードキュメントのルールに関する詳細については、「キーポリシー形式」を参照してください。IAM ポリシードキュメントのルールに関する詳細については、「IAM ユーザーガイド」の「IAM 名前の要件」を参照してください。

非対称 KMS キーまたは HMAC KMS キーを使用する暗号化オペレーションで暗号化コンテキストを指定することはできません。非対称アルゴリズムと MAC アルゴリズムは、暗号化コンテキストをサポートしません。

kms:EncryptionContext：context-key 条件キーを使用するには、*context-key* プレースホルダーを暗号化コンテキストキーに置き換えます。*#######* プレースホルダーを暗号化コンテキスト値と置き換えます。

```
"kms:EncryptionContext:context-key": "context-value"
```

例えば、次の条件キーでは、キーが AppName、値が ExampleApp (AppName = ExampleApp) の暗号化コンテキストを指定します。

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

これは単一値の条件キーです。条件キーのキーは、特定の暗号化コンテキストキー (context-key) を指定します。各 API リクエストに複数の暗号化コンテキストペアを含めることができますが、指

定されたコンテキストキーを持つ暗号化コンテキストペアが設定できる値は 1 つだけです。例えば、kms:EncryptionContext:Department 条件キーは、Department キーを持つ暗号化コンテキストペアにのみ適用されます。Department キーを持つ任意の暗号化コンテキストペアには 1 つの値しか設定できません。

kms:EncryptionContext:*context-key* 条件キーで集合演算子を使用しないでください。ポリシーステートメントを Allow アクションで作成する場合、kms:EncryptionContext:*context-key* 条件キー、ForAllValues 集合演算子、条件は、暗号化コンテキストのないリクエストと、ポリシー条件で指定されていない暗号化コンテキストペアを持つリクエストを許可します。

> **Warning**
>
> この単一値の条件キーで、ForAnyValue または ForAllValues の集合演算子を使用しないでください。これらの集合演算子は、要求する値を必要としないポリシー条件を作成し、禁止する値を許可する可能性があります。
> kms::EncryptionContextcontext-key を持つForAllValues集合演算子を含むポリシーを作成または更新すると、 は次のエラーメッセージ AWS KMS を返します。
> `OverlyPermissiveCondition:EncryptionContext: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified encryption context or with an unspecified encryption context. To fix, remove ForAllValues.`

特定の暗号化コンテキストペアを要求するには、StringEquals 演算子を持つ kms:EncryptionContext:*context-key* 条件キーを使用します。

次の例に示されるキーポリシーステートメントでは、リクエスト内の暗号化コンテキストに AppName:ExampleApp ペアが含まれる場合に限り、ロールを引き受けることができるプリンシパルが GenerateDataKey リクエストのKMS キーを使用できるようにします。他の暗号化コンテキストのペアも許可されます。

キースペース名では、大文字と小文字は区別されません。値の大文字と小文字の区別は、条件演算子 (StringEquals など) によって決定されます。詳細については、「暗号化コンテキスト条件での大文字と小文字の区別」を参照してください。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

暗号化コンテキストペアを要求し、他のすべての暗号化コンテキストペアを禁止するには、ポリシーステートメント`kms:EncryptionContextKeys`で kms:EncryptionContext：context-key との両方を使用します。次のポリシーステートメントでは、`kms:EncryptionContext:AppName` 条件キーを使用してリクエストの AppName=ExampleApp 暗号化コンテキストペアを要求します。`ForAllValues` 集合演算子を持つ `kms:EncryptionContextKeys` 条件キーを使用して、AppName 暗号化コンテキストキーのみを許可することもできます。

`ForAllValues` 集合演算子は、リクエストの暗号化コンテキストキーを AppName に制限します。ポリシーステートメントに、`ForAllValues` 集合演算子を持つ `kms:EncryptionContextKeys` 条件が単独で使用された場合、この集合演算子は、暗号化コンテキストのないリクエストを許可します。ただし、リクエストに暗号化コンテキストがない場合、`kms:EncryptionContext:AppName` 条件は失敗します。`ForAllValues` 集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KeyUsers"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "AppName"
      ]
    }
  }
```

```
}
```

この条件キーを使用して、特定のオペレーションの KMS キーへのアクセスを拒否することもできます。次の例のキーポリシーステートメントでは、Deny 効果を使用して、リクエストの暗号化コンテキストに Stage=Restricted 暗号化コンテキストペアが含まれる場合に、プリンシパルがKMS キーを使用することを禁止します。この条件により、他の暗号化コンテキストペアを含むリクエストが許可されます。これには、Stage キーおよびその他の値 (Stage=Test など) を持つ暗号化コンテキストペアが含まれます。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

複数の暗号化コンテキストペアを使用する

複数の暗号化コンテキストペアを要求または禁止できます。また、複数の暗号化コンテキストペアのうちの 1 つを要求することもできます。これらの条件を解釈するために使用されるロジックの詳細については、IAM ユーザーガイドの複数のキーまたは値を持つ条件の作成を参照してください。

> **Note**
>
> このトピックの以前のバージョンでは、kms::EncryptionContextcontext-key 条件キーで ForAnyValueおよび ForAllValues集合演算子を使用するポリシーステートメントが表示されていました。単一値の条件キーを持つ集合演算子を使用すると、暗号化コンテキストのないリクエストおよび暗号化コンテキストペアが指定されていないリクエストを許可するポリシーとなる可能性があります。
> 例えば、Allow 効果を持つポリシー条件の場合、ForAllValues 集合演算子および "kms:EncryptionContext:Department": "IT" 条件キーは、暗号化コンテキストを「Department=IT」ペアに制限しません。これは、暗号化コンテキストのないリク

エストとおよび暗号化コンテキストのペアが指定されていないリクエストを許可します (Stage=Restricted など)。
ポリシーを確認し、kms::EncryptionContextcontext-key を含む条件から集合演算子を削除してください。この形式のポリシーを作成または更新しようとすると、OverlyPermissiveCondition の例外を含むエラーが発生します。このエラーを解決するには、集合演算子を削除します。

複数の暗号化コンテキストのペアを要求するには、同じ条件でペアを一覧表示します。次の例のキーポリシーステートメントでは、2 つの暗号化コンテキストペア (Department=IT および Project=Alpha) を要求します。条件には異なるキー (kms:EncryptionContext:Departmentおよびkms:EncryptionContext:Project) が含まれるため、それらは暗黙的に AND 演算子によって接続されます。他の暗号化コンテキストペアは許可されますが、必須ではありません。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    }
  }
}
```

1 つの暗号化コンテキストペア、または別のペアを要求するには、各条件キーを個別のポリシーステートメントに配置します。次のキーポリシーの例では、Department=IT または Project=Alpha ペア、またはその両方を要求します。他の暗号化コンテキストペアは許可されますが、必須ではありません。

```
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
 },
```

```
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Department": "IT"
  }
 }
},
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
 },
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Project": "Alpha"
  }
 }
}
```

特定の暗号化ペアを要求し、他のすべての暗号化コンテキストペアを除外するには、ポリシーステートメントkms:EncryptionContextKeysで kms:EncryptionContext：context-key と の両方を使用します。次のキーポリシーステートメントでは、kms:EncryptionContext：context-key 条件を使用して、 Department=ITと の両方のProject=Alphaペアを持つ暗号化コンテキストを要求します。このコンテキストでは、ForAllValues 集合演算子を持つ kms:EncryptionContextKeys 条件キーを使用して、Department および Project 暗号化コンテキストキーのみを許可します。

ForAllValues 集合演算子は、リクエストの暗号化コンテキストキーを Department Project に制限します。条件内で単独で使用した場合、この集合演算子は暗号化コンテキストのないリクエストを許可しますが、この設定では、この条件の kms:EncryptionContext：context-key は失敗します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
```

```
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "Department",
        "Project"
      ]
    }
  }
}
```

複数の暗号化コンテキストペアを禁止することもできます。次の例のキーポリシーステートメントでは、Deny 効果を使用して、リクエストの暗号化コンテキストに Stage=Restricted または Stage=Production ペアが含まれる場合に、プリンシパルがKMS キーを使用することを禁止します。

同じキー (kms:EncryptionContext:Stage) の複数の値 (RestrictedおよびProduction) は、暗黙的に OR によって接続されます。詳細については、IAM ユーザーガイドの Evaluation logic for conditions with multiple keys or values を参照してください。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
         "Restricted",
         "Production"
      ]
    }
  }
}
```

暗号化コンテキスト条件での大文字と小文字の区別

復号オペレーションで指定される暗号化コンテキストは、暗号化オペレーションで指定される暗号化コンテキストに大文字と小文字を区別して完全に一致する必要があります。複数のペアの暗号化コンテキストのペアの順序のみを変更できます。

ただし、ポリシー条件では、条件キーの大文字と小文字は区別されません。条件値の大文字と小文字の区別は、使用する ポリシー条件演算子 (StringEquals や StringEqualsIgnoreCase など) によって決まります。

したがって、kms:EncryptionContext: プレフィックスと *context-key* の置換で構成される条件キーでは、大文字と小文字は区別されません。この条件を使用するポリシーでは、条件キーのいずれの要素もチェックされません。値の大文字と小文字の区別 (*context-value* の置換) は、ポリシー条件演算子によって決まります。

例えば、次のポリシーステートメントでは、大文字と小文字に関係なく暗号化コンテキストに Appname キーが含まれている場合にオペレーションが許可されます。この StringEquals 条件では、指定時に ExampleApp を大文字にする必要があります。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Appname": "ExampleApp"
    }
  }
}
```

大文字と小文字を区別する暗号化コンテキストキーを要求するには、kms:EncryptionContextKeys policy 条件と、 などの大文字と小文字を区別する条件演算子を使用しますStringEquals。このポリシー条件では、暗号化コンテキストキーがポリシー条件値であるため、大文字と小文字の区別は条件演算子によって決定されます。

```
{
  "Effect": "Allow",
```

```
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

暗号化コンテキストキーと値の両方について大文字と小文字を区別する評価を要求するには、同じポリシーステートメントで kms:EncryptionContextKeysと kms:EncryptionContext：context-key ポリシー条件を一緒に使用します。大文字と小文字を区別する条件演算子 (StringEquals など) は、常に条件の値に適用されます。暗号化コンテキストキー (AppName など) は kms:EncryptionContextKeys 条件の値です。暗号化コンテキスト値 (などExampleApp) は、kms:EncryptionContext：context-key 条件の値です。

例えば、次のキーポリシーステートメントでは、StringEquals 演算子で大文字と小文字が区別されるため、暗号化コンテキストキーと暗号化コンテキスト値の両方で大文字と小文字が区別されます。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    },
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

暗号化コンテキスト条件での変数の使用

暗号化コンテキストペアのキーと値はシンプルなリテラル文字列であることが必要です。整数やオブジェクトなど完全に解決されない型のキーと値は使用できません。整数や浮動小数点数など、別の型を使用する場合、 はそれをリテラル文字列として AWS KMS 解釈します。

```
"encryptionContext": {
    "department": "10103.0"
}
```

ただし、kms:EncryptionContext:*context-key* 条件キーの値は IAM ポリシー変数となる可能性があります。これらのポリシー変数はリクエストの値に基づいて実行時に解決されます。例えば、aws:CurrentTime はリクエストの時間に解決され、aws:username は呼び出し元のフレンドリ名に解決されます。

これらのポリシー変数を使用し、ポリシーステートメントを作成して、暗号化コンテキストに限定的な情報 (呼び出し元のユーザー名など) を必要とする条件を指定できます。ポリシーステートメントに変数を含めるため、ロールを引き受けることができるすべてのユーザーに同じポリシーステートメントを使用できます。ユーザー別にポリシーステートメントを記述する必要はありません。

ロールを引き受けることができるすべてのユーザーが同じ KMS キーを使用して、データを暗号化および復号する状況を考慮します。ただし、それらのユーザーに自分が暗号化したデータのみの復号を許可するとします。まず、すべてのリクエストに、キーが でuser値が呼び出し元の AWS ユーザー名である暗号化コンテキスト AWS KMS を含めるように要求します。次に例を示します。

```
"encryptionContext": {
    "user": "bob"
}
```

次に、この要件を定義するために、以下の例のようなポリシーステートメントを使用できます。このポリシーステートメントでは、TestTeam ロールに、KMS キーを使用してデータを暗号化および復号するためのアクセス許可を付与します。ただし、そのアクセス許可は、"user": "*<username>*" ペアがリクエストの暗号化コンテキストに含まれる場合にのみ有効です。条件では、ユーザー名を表すために aws:username policy 変数を使用します。

リクエストが評価されるとき、条件の変数が呼び出し元のユーザー名に置き換えられます。そのため条件では、「bob」には "user": "bob"、「alice」には "user": "alice" の暗号化コンテキストを必須とします。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

IAM ポリシー変数は、kms:EncryptionContext:*context-key* 条件キーの値でのみ使用できます。キーで変数を使用することはできません。

変数に プロバイダ固有のコンテキストキー を使用することもできます。これらのコンテキストキー AWS は、ウェブ ID フェデレーションを使用して にログインしたユーザーを一意に識別します。

すべての変数と同様に、これらの変数は、実際の暗号化コンテキストではなく kms:EncryptionContext:*context-key* ポリシー条件でのみ使用できます。また、条件のキーではなく値でのみ使用できます。

例えば、以下のキーポリシーステートメントは前のものと似ています。ただし、この条件には、Amazon Cognito ユーザープールにログインしたユーザーを値が一意に識別する、キーが sub の暗号化コンテキストが必要です。Amazon Cognito でのユーザーおよびロールの識別についての詳細は、Amazon Cognito デベロッパーガイドの IAM ロールを参照してください。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
```

```
    "StringEquals": {
       "kms:EncryptionContext:sub": "${cognito-identity.amazonaws.com:sub}"
    }
  }
}
```

以下の資料も参照してください。

- the section called “kms:EncryptionContextKeys”
- the section called “kms:GrantConstraintType”

kms:EncryptionContextKeys

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:EncryptionContextKeys	文字列 (リスト)	複数値	CreateGrant Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt	キーポリシーと IAM ポリシー

kms:EncryptionContextKeys 条件キーを使用することで、暗号化オペレーションに対するリクエスト内の暗号化コンテキストに基づいて対称暗号化 KMS キーへのアクセスを制御することができます。各暗号化コンテキストペアのキーのみを評価するには、この条件キーを使用します。暗号化コンテキストのキーと値の両方を評価するには、kms:EncryptionContext:*context-key* 条件キーを使用します。

非対称 KMS キーまたは HMAC KMS キーを使用する暗号化オペレーションで暗号化コンテキストを指定することはできません。非対称アルゴリズムと MAC アルゴリズムは、暗号化コンテキストをサポートしません。

> **Note**
>
> 暗号化コンテキストキーを含む条件キー値は、 AWS KMS キーポリシーの文字とエンコーディングルールに従う必要があります。この条件キーは、すべての有効な暗号化コンテキストキーを表現するために使用できない場合があります。キーポリシードキュメントのルールに関する詳細については、「キーポリシー形式」を参照してください。IAM ポリシードキュメントのルールに関する詳細については、「IAM ユーザーガイド」の「IAM 名前の要件」を参照してください。

これは複数値の条件キーです。各 API リクエストで複数の暗号化コンテキストペアを指定できます。kms:EncryptionContextKeys は、リクエストの暗号化コンテキストキーとポリシーの暗号化コンテキストキーのセットを比較します。これらのセットを比較する方法を決定するには、ForAnyValue または ForAllValues 集合演算子をポリシー条件で使用します。集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。

- ForAnyValue: リクエスト内の 1 つ以上の暗号化コンテキストキーがポリシー条件の暗号化コンテキストキーと一致する必要があります。その他の暗号化コンテキストキーも許可されます。リクエストに暗号化コンテキストがない場合、条件は満たされません。
- ForAllValues: リクエスト内のすべての暗号化コンテキストキーがポリシー条件の暗号化コンテキストキーと一致する必要があります。この集合演算子は、暗号化コンテキストキーをポリシー条件内のキーに制限します。暗号化コンテキストキーは必要ありませんが、指定されていない暗号化コンテキストキーは禁止されています。

次の例のキーポリシーステートメントでは、ForAnyValue 集合演算子で kms:EncryptionContextKeys 条件キーを使用します。このポリシーステートメントでは KMS

キーを使用して、リクエストの暗号化コンテキストペアの 1 つ以上に、値にかかわらず AppName キーが含まれる場合にのみ、指定されたオペレーションで KMS キーの使用を許可します。

例えば、このキーポリシーステートメントでは、2 つの暗号化コンテキストペア AppName=Helper および Project=Alpha を持つ GenerateDataKey リクエストを許可します。これは、最初の暗号化コンテキストペアが条件を満たすためです。Project=Alpha のみを持つリクエスト、または暗号化コンテキストがないリクエストは失敗します。

StringEquals 条件オペレーションでは大文字と小文字が区別されるため、このポリシーステートメントでは暗号化コンテキストキーのスペルと大文字と小文字が区別されます。ただし、キーの大文字と小文字の区別を無視する条件演算子 (StringEqualsIgnoreCase など) を使用できます。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

kms:EncryptionContextKeys 条件キーを使用して、KMS キーを使用する暗号化オペレーションで、暗号化テキスト (任意の暗号化コンテキスト) を要求することもできます。

次の例のキーポリシーステートメントでは、kms:EncryptionContextKeys 条件キーを Null 条件演算子とともに使用して、API リクエストに暗号化コンテキストが存在する (null ではない) 場合にのみ、KMS キーへのアクセスを許可します。この条件では、暗号化コンテキストのキーまたは値をチェックしません。暗号化コンテキストが存在することだけを検証します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

以下の資料も参照してください。

- kms:EncryptionContext：context-key
- kms:GrantConstraintType

kms:ExpirationModel

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ExpirationModel	文字列	単一値	ImportKeyMaterial	キーポリシーと IAM ポリシー

kms:ExpirationModel 条件キーは、リクエスト内の ExpirationModelパラメータの値に基づいて、 ImportKeyMaterialオペレーションへのアクセスを制御します。

ExpirationModel は、インポートされたキーマテリアルの有効期限が切れているかどうかを判断するオプションのパラメータです。有効な値は、KEY_MATERIAL_EXPIRES および KEY_MATERIAL_DOES_NOT_EXPIRE です。KEY_MATERIAL_EXPIRES はデフォルト値です。

有効期限日時は、 ValidToパラメータの値によって決まります。ValidTo パラメータの値が ExpirationModel である場合を除き、KEY_MATERIAL_DOES_NOT_EXPIRE パラメータが必要です。kms:ValidTo 条件キーを使用して、アクセスの条件として特定の有効期限を要求することもできます。

次のポリシーステートメントの例では、kms:ExpirationModel 条件キーを使用して、リクエストに ExpirationModel パラメータが含まれていて、その値が KEY_MATERIAL_DOES_NOT_EXPIRE の場合にのみ、ユーザーが KMS キーにキーマテリアルをインポートできるようにします。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

また、kms:ExpirationModel 条件キーを使用して、キーマテリアルの有効期限が切れている場合にのみ、ユーザーがキーマテリアルをインポートできるようにもします。次のポリシーステートメントの例では、Null 条件演算子とともに kms:ExpirationModel 条件キーを使用して、リクエストに ExpirationModel パラメータが含まれていない場合にのみ、ユーザーがキーマテリアルをインポートできるようにします。のデフォルト値は ExpirationModel ですKEY_MATERIAL_EXPIRES。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

以下の資料も参照してください。

- kms:ValidTo

- kms:WrappingAlgorithm
- kms:WrappingKeySpec

kms:GrantConstraintType

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:GrantConstraintType	文字列	単一値	CreateGrant	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエスト内のグラント制約のタイプに基づいてCreateGrantオペレーションへのアクセスを制御できます。

許可の作成では、オプションで許可の制約を指定して、特定の暗号化コンテキストが存在する場合のみ、許可によってオペレーションを実行できます。許可の制約には、EncryptionContextEqualsまたはEncryptionContextSubset の 2 つのタイプがあります。この条件キーを使用して、リクエストにどちらのタイプが含まれているか確認できます。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

次のキーポリシーステートメントの例では、kms:GrantConstraintType 条件キーを使用して、リクエストに EncryptionContextEquals 権限の制約が含まれている場合にのみ、ユーザーが権限を作成できるようにします。この例は、キーポリシーのポリシーステートメントを示しています。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
```

```
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

以下の資料も参照してください。

- kms:EncryptionContext：context-key
- kms:EncryptionContextKeys
- kms:GrantIsForAWSResource
- kms:GrantOperations
- kms:GranteePrincipal
- kms:RetiringPrincipal

kms:GrantIsForAWSResource

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ

`kms:GrantIsForAWSResource`	ブール値	単一値	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				キーポリシーと IAM ポリシー

AWS と統合された のサービス AWS KMSがユーザーに代わって RevokeGrantオペレーションを呼び出す場合にのみCreateGrant、ListGrants、、または オペレーションのアクセス許可を許可または拒否します。このポリシー条件では、ユーザーがこれらの許可オペレーションを直接呼び出すことはできません。

次のキーポリシーステートメント例では、 `kms:GrantIsForAWSResource` 条件キーを使用しています。これにより AWS KMS、Amazon EBS などの と統合された AWS サービスが、指定されたプリンシパルに代わってこの KMS キーで許可を作成できます。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

以下の資料も参照してください。

- kms:GrantConstraintType
- kms:GrantOperations
- kms:GranteePrincipal
- kms:RetiringPrincipal

kms:GrantOperations

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:GrantOperations	文字列	複数値	CreateGrant	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエストの権限CreateGrantオペレーションに基づいて オペレーションへのアクセスを制御できます。 ???例えば、暗号化へのアクセス権限を委任し、復号化へのアクセス権限を委任しないという許可をユーザーが作成するようにできます。権限の詳細については、権限の使用を参照してください。

これは複数値を持つ条件キーです。kms:GrantOperations は CreateGrant リクエストの権限オペレーションのセットをポリシーの権限オペレーションのセットと比較します。これらのセットを比較する方法を決定するには、ForAnyValue または ForAllValues 集合演算子をポリシー条件で使用します。集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。

- ForAnyValue: リクエストの 1 つ以上の権限オペレーションが、ポリシー条件の権限オペレーションのうちの 1 つと一致する必要があります。その他の権限オペレーションは許可されます。
- ForAllValues: リクエスト内のすべてのグラントオペレーションは、ポリシー条件のグラントオペレーションと一致する必要があります。この集合演算子は、権限オペレーションをポリシー条件で指定されたオペレーションに制限します。この集合演算子は権限オペレーションを必要としませんが、不特定の権限オペレーションを禁止します。

  ForAllValues また、 は、リクエストに許可オペレーションがない場合にも true を返しますが、CreateGrantはそれを許可しません。Operations パラメータが欠落しているか、null 値を持っている場合、CreateGrant リクエストは失敗します。

次のキーポリシーステートメントの例では、kms:GrantOperations 条件キーを使用して、権限オペレーションが Encrypt、ReEncryptTo、または両方の場合にのみ、権限を作成します。権限に他のオペレーションが含まれている場合、CreateGrant リクエストは失敗します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

ポリシー条件で集合演算子を ForAnyValue に変更した場合、ポリシーステートメントでは、権限の権限オペレーションの 1 つ以上が Encrypt または ReEncryptTo であっても、Decrypt または ReEncryptFrom のような他の権限オペレーションを許可する必要があります。

以下の資料も参照してください。

- kms:GrantConstraintType
- kms:GrantIsForAWSResource

- kms:GranteePrincipal
- kms:RetiringPrincipal

kms:GranteePrincipal

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:GranteePrincipal	文字列	単一値	CreateGrant	IAM とキーポリシー

この条件キーを使用して、リクエストの GranteePrincipalパラメータの値に基づいてCreateGrantオペレーションへのアクセスを制御できます。例えば、CreateGrant リクエストの被付与者プリンシパルが条件ステートメントで指定されたプリンシパルと一致した場合にのみ、KMS キーを使用する権限を作成できます。

被付与者プリンシパルを指定するには、 プリンシパルの Amazon リソースネーム (ARN) AWS を使用します。有効なプリンシパルには AWS アカウント、、IAM ユーザー、IAM ロール、フェデレーティッドユーザー、引き受けたロールユーザーが含まれます。プリンシパルの ARN 構文については、「IAM ユーザーガイド」のARNs」を参照してください。

次のキーポリシーステートメントの例では、kms:GranteePrincipal 条件キーを使用して、権限の被付与者プリンシパルが LimitedAdminRole の場合にのみ、KMS キーの権限を作成します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

以下の資料も参照してください。

- kms:GrantConstraintType
- kms:GrantIsForAWSResource
- kms:GrantOperations
- kms:RetiringPrincipal

kms:KeyOrigin

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:KeyOrigin	文字列	単一値	CreateKey KMS キーリソースのオペレーション	IAM ポリシー キーポリシーと IAM ポリシー

kms:KeyOrigin 条件キーは、オペレーションによって作成される、またはオペレーションで使用される KMS キーの Origin プロパティの値に基づいて、オペレーションへのアクセスを制御します。これは、リソース条件または要求条件として機能します。

この条件キーを使用して、リクエストの Origin パラメータの値に基づいてCreateKeyオペレーションへのアクセスを制御できます。Origin の有効値は、AWS_KMS、AWS_CLOUDHSM、および EXTERNAL です。

例えば、KMS キーを作成できるのは、キーマテリアルが AWS KMS （ AWS_KMS) で生成されたときか、カスタムキーストア () に関連付けられた AWS CloudHSM クラスターでキーマテリアルが生成されたときか、キーマテリアルが外部ソース (AWS_CLOUDHSM) からインポートされたときだけですEXTERNAL。 ???

次のキーポリシーステートメントの例では、 kms:KeyOrigin条件キーを使用して、 がキーマテリアルを作成する場合にのみ KMS キー AWS KMS を作成します。

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": "kms:CreateKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_KMS"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:GenerateDataKeyPair",
        "kms:GenerateDataKeyPairWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_CLOUDHSM"
        }
      }
    }
  ]
}
```

kms:KeyOrigin 条件キーを使用し、オペレーションに使用される KMS キーの Origin プロパティに基づいて、KMS キーを使用または管理するオペレーションへのアクセスを制御することもできます。このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。

例えば次の IAM ポリシーでは、カスタムキーストアで作成されたアカウントの KMS キーのみを使用して、プリンシパルが、指定された KMS キーリソースのオペレーションを実行できるようにします。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext",
    "kms:ReEncrypt*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_CLOUDHSM"
    }
  }
}
```

以下の資料も参照してください。

- kms:BypassPolicyLockoutSafetyCheck
- kms:KeySpec
- kms:KeyUsage

kms:KeySpec

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:KeySpec	文字列	単一値	CreateKey KMS キーリソースのオペレーション	IAM ポリシー キーポリシーと IAM ポリシー

kms:KeySpec 条件キーは、オペレーションによって作成される、またはオペレーションで使用される KMS キーの KeySpec プロパティの値に基づいて、オペレーションへのアクセスを制御します。

IAM ポリシーでこの条件キーを使用して、CreateKeyリクエストの KeySpecパラメータの値に基づいてCreateKeyオペレーションへのアクセスを制御できます。例えば、この条件を使用して、対称暗号化 KMS キーのみの作成、または HMAC KMS キーのみの作成をユーザーに許可することができます。

以下の IAM ポリシーステートメント例は、kms:KeySpec 条件キーを使用して、RSA 非対称 KMS キーのみの作成をプリンシパルに許可します。この許可は、リクエスト内の KeySpec が RSA_ で始まる場合に限り、有効です。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

kms:KeySpec 条件キーを使用し、オペレーションに使用される KMS キーの KeySpec プロパティに基づいて、KMS キーを使用または管理するオペレーションへのアクセスを制御することもできます。このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。

例えば、以下の IAM ポリシーは、指定された KMS キーリソースオペレーションの実行をプリンシパルに許可しますが、許可されるのはアカウント内の対称暗号化 KMS キーを使用する場合のみになります。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey"
  ],
```

```
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeySpec": "SYMMETRIC_DEFAULT"
    }
  }
}
```

以下の資料も参照してください。

- kms:BypassPolicyLockoutSafetyCheck
- kms:CustomerMasterKeySpec (非推奨)
- kms:DataKeyPairSpec
- kms:KeyOrigin
- kms:KeyUsage

kms:KeyUsage

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:KeyUsage	文字列	単一値	CreateKey	IAM ポリシー
			KMS キーリソースのオペレーション	キーポリシーと IAM ポリシー

kms:KeyUsage 条件キーは、オペレーションによって作成される、またはオペレーションで使用される KMS キーの KeyUsage プロパティの値に基づいて、オペレーションへのアクセスを制御します。

この条件キーを使用して、リクエストの KeyUsageパラメータの値に基づいてCreateKeyオペレーションへのアクセスを制御できます。KeyUsage の有効値は、ENCRYPT_DECRYPT、SIGN_VERIFY、および GENERATE_VERIFY_MAC です。

例えば、KeyUsage が ENCRYPT_DECRYPT である場合にのみ KMS キーを作成し、KeyUsage が SIGN_VERIFY の場合はユーザーのアクセス許可を拒否できます。

次の IAM ポリシーステートメントの例では、kms:KeyUsage 条件キーを使用して、KeyUsage が ENCRYPT_DECRYPT の場合にのみ KMS キーを作成します。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "ENCRYPT_DECRYPT"
    }
  }
}
```

kms:KeyUsage 条件キーを使用し、オペレーションの KMS キーの KeyUsage プロパティに基づいて、KMS キーを使用または管理するオペレーションへのアクセスを制御することもできます。このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。

例えば次の IAM ポリシーでは、署名と検証に使用されるアカウント内の KMS キーのみを使用して、指定された KMS キーリソースのオペレーションをプリンシパルが実行できるようにします。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:ScheduleKeyDeletion"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "SIGN_VERIFY"
    }
  }
}
```

以下の資料も参照してください。

- kms:BypassPolicyLockoutSafetyCheck

- kms:CustomerMasterKeyUsage (非推奨)
- kms:KeyOrigin
- kms:KeySpec

kms:MacAlgorithm

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:MacAlgorithm	文字列	単一値	GenerateMac VerifyMac	キーポリシーと IAM ポリシー

kms:MacAlgorithm 条件キーを使用して、リクエスト内の MacAlgorithmパラメータの値に基づいて GenerateMacおよび VerifyMacオペレーションへのアクセスを制御できます。

以下のキーポリシー例は、リクエスト内の MAC アルゴリズムが HMAC_SHA_384 または HMAC_SHA_512 である場合に限り、HMAC KMS キーを使用して HMAC タグを生成し、検証することを、testers ロールを引き受けることができるユーザーに許可します。このポリシーは、それぞれが独自の条件を持つ 2 つの個別のポリシーステートメントを使用します。単一の条件ステートメント内で複数の MAC アルゴリズムを指定する場合、条件はアルゴリズムのどちらか一方ではなく、両方を必須とします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```
          "kms:MacAlgorithm": "HMAC_SHA_384"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_512"
        }
      }
    }
  ]
}
```

以下の資料も参照してください。

- the section called “kms:EncryptionAlgorithm”
- kms:SigningAlgorithm

kms:MessageType

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:MessageType	文字列	単一値	Sign Verify	キーポリシーと IAM ポリシー

kms:MessageType 条件キーは、リクエストの MessageType パラメータの値に基づいて、Sign および Verify オペレーションへのアクセスを制御します。MessageType の有効値は、RAW と DIGEST です。

例えば、次のキーポリシーステートメントでは kms:MessageType 条件キーを使用して、非対称 KMS キーを使用してメッセージに署名することを許可し、メッセージダイジェストの使用は拒否しません。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:MessageType": "RAW"
    }
  }
}
```

以下の資料も参照してください。

- the section called “kms:SigningAlgorithm”

kms:MultiRegion

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:MultiRegion	ブール値	単一値	CreateKey KMS キーリソースのオペレーション	キーポリシーと IAM ポリシー

この条件キーを使用すると、単一リージョンキー、またはマルチリージョンキーのみのオペレーションを許可できます。kms:MultiRegion 条件キーは、KMS キーの MultiRegionプロパティの値に基づいて、KMS キーに対する AWS KMS オペレーションおよび CreateKeyオペレーションへのアクセスを制御します。有効な値は、true (マルチリージョン)、および false (単一リージョン) です。すべての KMS キーには MultiRegion プロパティが含まれます。

例えば、次の IAM ポリシーステートメントでは kms:MultiRegion 条件キーを使用して、プリンシパルが単一リージョンキーを作成できるようにします。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}
```

kms:MultiRegionKeyType

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:MultiRegionKeyType	文字列	単一値	CreateKey KMS キーリソースのオペレーション	キーポリシーと IAM ポリシー

この条件キーを使用して、マルチリージョンプライマリキーまたはマルチリージョンレプリカキーのみのオペレーションを許可できます。kms:MultiRegionKeyType 条件キーは、KMS キーの MultiRegionKeyTypeプロパティに基づいて、KMS キーの AWS KMS オペレーションと CreateKeyオペレーションへのアクセスを制御します。有効な値は PRIMARY および REPLICA です。マルチリージョンキーのみに MultiRegionKeyType プロパティがあります

通常は、IAM ポリシーの kms:MultiRegionKeyType 条件キーを使用して、複数の KMS キーへのアクセスを制御します。ただし、特定のマルチリージョンキーがプライマリまたはレプリカに変更されることがあるため、キーポリシーでこの条件を使用して、特定のマルチリージョンキーがプライマリキーまたはレプリカキーである場合にのみ、オペレーションを許可します。

例えば、次の IAM ポリシーステートメントでは kms:MultiRegionKeyType 条件キーを使用して、指定した AWS アカウントのマルチリージョンレプリカキーのみでキー削除のスケジュールおよびキャンセルを実行することをプリンシパルに許可します。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}
```

すべてのマルチリージョンキーへのアクセスを許可または拒否するには、両方の値または kms:MultiRegionKeyType の null 値を使用します。ただし、その目的には kms:MultiRegion 条件キーを使用することをお勧めします。

kms:PrimaryRegion

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:PrimaryRegion	文字列 (リスト)	単一値	UpdatePrimaryRegion	キーポリシーと IAM ポリシー

この条件キーを使用して、UpdatePrimaryRegionオペレーションの送信先リージョンを制限できます。これらは AWS リージョン 、マルチリージョンのプライマリキーをホストできる です。

kms:PrimaryRegion 条件キーは、 PrimaryRegionパラメータの値に基づいてUpdatePrimaryRegionオペレーションへのアクセスを制御します。PrimaryRegion パラメータは、プライマリに昇格するマルチリージョンレプリカキー AWS リージョン の を指定します。???条件の値は、 や などの 1 us-east-1 つ以上の AWS リージョン 名前ap-southeast-2、または などのリージョン名パターンです。 eu-*

例えば、次のキーポリシーステートメントでは kms:PrimaryRegion 条件キーを使用して、プリンシパルがマルチリージョンキーのプライマリリージョンを、指定した 4 つのリージョンのうちの 1 つに更新できるようにします。

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Developer"
  },
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-east-1",
         "us-west-2",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

kms:ReEncryptOnSameKey

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ReEncryptOnSameKey	ブール値	単一値	ReEncrypt	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエストが元の暗号化に使用したものと同じ送信先 KMS キーを指定するかどうかに基づいて、 ReEncryptオペレーションへのアクセスを制御できます。

例えば、次のポリシーステートメントでは kms:ReEncryptOnSameKey 条件キーを使用して、対象の KMS キーが元の暗号化に使用されたものと同じである場合にのみ再暗号化します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
```

```
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}
```

kms:RequestAlias

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:RequestAlias	文字列 (リスト)	単一値	暗号化オペレーション DescribeKey GetPublicKey	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエストが KMS キーを識別するために特定のエイリアスを使用する場合にのみ、オペレーションを許可します。kms:RequestAlias 条件キーでは、リクエスト内の KMS キーを識別するエイリアスに基づいて、暗号化オペレーション (GetPublicKey または DescribeKey) で使用される KMS キーへのアクセスを制御します。(このポリシー条件は、GenerateRandomオペレーションが KMS キーまたはエイリアスを使用しないため、オペレーションには影響しません) 。

この条件は、 の属性ベースのアクセスコントロール (ABAC) をサポートします。これにより AWS KMS、KMS キーのタグとエイリアスに基づいて KMS キーへのアクセスを制御できます。ポリシーや権限を変更せずに、タグとエイリアスを使用して KMS キーへのアクセスを許可または拒否できます。詳細については、「AWS KMS の ABAC」を参照してください。

このポリシー条件でエイリアスを指定するには、エイリアス名 (alias/project-alpha など)、またはエイリアス名パターン (alias/*test* など) を使用します。この条件キーの値にエイリアス ARN を指定することはできません。

この条件を満たすには、リクエストの KeyId パラメータの値が、一致するエイリアス名またはエイリアス ARN である必要があります。リクエストが別のキー識別子を使用する場合、同じ KMS キーを識別しても条件を満たしません。

例えば、次のキーポリシーステートメントでは、プリンシパルが KMS キーで GenerateDataKeyオペレーションを呼び出すことができます。ただし、これは、リクエスト内の KeyId パラメータ値が alias/finance-key、またはその エイリアス名を持つエイリアス ARN (arn:aws:kms:us-west-2:111122223333:alias/finance-key など)である場合にのみ許可されます。

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/developer"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RequestAlias": "alias/finance-key"
    }
  }
}
```

この条件キーを使用して、 CreateAliasや などのエイリアスオペレーションへのアクセスを制御することはできませんDeleteAlias。エイリアスオペレーションへのアクセスの制御については、エイリアスへのアクセスの制御 を参照してください。

kms:ResourceAliases

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ResourceAliases	文字列 (リスト)	複数値	KMS キーリソースのオペレーション	IAM ポリシーのみ

この条件キーを使用し、KMS キーに関連付けられたエイリアスに基づいて、KMS キーへのアクセスを制御します。このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。

この条件では、 AWS KMSの属性ベースのアクセスコントロール (ABAC) をサポートします。ABAC を使用すると、KMS キーに割り当てられたタグと KMS キーに関連付けられたエイリアスに基づいて、KMS キーへのアクセスを制御できます。ポリシーや権限を変更せずに、タグとエイリアスを使用して KMS キーへのアクセスを許可または拒否できます。詳細については、「AWS KMS の ABAC」を参照してください。

エイリアスは AWS アカウント およびリージョンで一意である必要がありますが、この条件により、同じリージョン内の複数の KMS キー (`StringLike` 比較演算子を使用) または AWS リージョン 各アカウントが異なる の複数の KMS キーへのアクセスを制御できます。

> **Note**
>
> kms:ResourceAliases 条件は、KMS キーが KMS キークォータあたりのエイリアスに準拠している場合にのみ有効です。KMS キーがこのクォータを超えると、KMS キーを `kms:ResourceAliases` 条件で使用するよう認可されたプリンシパルは、KMS キーへのアクセスを拒否されます。

このポリシー条件でエイリアスを指定するには、エイリアス名 (`alias/project-alpha` など)、またはエイリアス名パターン (`alias/*test*` など) を使用します。この条件キーの値にエイリアス ARN を指定することはできません。条件を満たすには、オペレーションで使用する KMS キーが指定されたエイリアスを持っている必要があります。オペレーションのリクエストで KMS キーが識別されるかどうか、またはどのように識別されるかは関係ありません。

これは、KMS キーに関連付けられたエイリアスのセットとポリシー内のエイリアスのセットを比較する、複数値を持つ条件キーです。これらのセットを比較する方法を決定するには、`ForAnyValue` または `ForAllValues` 集合演算子をポリシー条件で使用します。集合演算子の詳細については、IAM ユーザーガイドの複数のキーと値の使用を参照してください。

- ForAnyValue: KMS キーに関連付けられた少なくとも 1 つのエイリアスがポリシー条件のエイリアスと一致する必要があります。その他のエイリアスは許可されます。KMS キーにエイリアスがない場合、条件は満たされません。
- ForAllValues: KMS キーに関連付けられているすべてのエイリアスは、ポリシーのエイリアスと一致する必要があります。この集合演算子は、KMS キーと関連付けられるエイリアスを、ポリシー条件内のエイリアスに制限します。エイリアスを必要としませんが、不特定のエイリアスを禁止します。

例えば、次の IAM ポリシーステートメントでは、エイリアスに関連付けられている指定された 内の任意の KMS finance-key キーでプリンシパル AWS アカウント が GenerateDataKeyオペレーションを呼び出すことができます。(影響を受ける KMS キーのキーポリシーでは、プリンシパルのアカウントに、このオペレーションでキーを使用することも許可する必要があります)。KMS キーに関連付けられる可能性がある多数のエイリアスのいずれかが alias/finance-key である場合に、条件が満たされていることを示すには条件に ForAnyValue 集合演算子を使用します。

kms:ResourceAliases 条件はリクエストではなくリソースに基づいているため、GenerateDataKey に対する呼び出しは、リクエストが KMS キーを識別するためにキー ID またはキー ARN を使用している場合でも、finance-key エイリアスに関連付けられているすべての KMS キーに対して成功します。

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": "kms:GenerateDataKey",
  "Resource": [
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:ResourceAliases": "alias/finance-key"
    }
  }
}
```

次の IAM ポリシーステートメントの例では、KMS キーのすべてのエイリアスに「Test」が含まれる場合にのみ、プリンシパルが KMS キーを有効または無効にすることを許可します。このポリシーステートメントは 2 つの条件を使用します。ForAllValues 集合演算子を持つ条件では、KMS キーに関連付けられたすべてのエイリアスに「Test」が含まれている必要があります。ForAnyValue 集合演算子を持つ条件では、KMS キーに 1 つ以上の「Test」を持つエイリアスが含まれている必要があります。ForAnyValue 条件なしの場合、このポリシーステートメントはプリンシパルに、エイリアスがない KMS キーの使用を許可する可能性があります。

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
```

```
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    }
  }
}
```

kms:ReplicaRegion

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ReplicaRegion	文字列 (リスト)	単一値	ReplicateKey	キーポリシーと IAM ポリシー

この条件キーを使用して、プリンシパル AWS リージョン がマルチリージョンキー をレプリケートできる を制限できます。kms:ReplicaRegion 条件キーは、リクエスト内の ReplicaRegionパラメータの値に基づいて、 ReplicateKeyオペレーションへのアクセスを制御します。このパラメータは、 AWS リージョン の新しいレプリカキーを指定します。

条件の値は、 や などの 1 us-east-1 つ以上の AWS リージョン 名前ap-southeast-2、またはなどの名前パターンですeu-*。が AWS KMS サポート AWS リージョン する の名前のリストについては、「」のAWS Key Management 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。

例えば、次のキーポリシーステートメントでは、 kms:ReplicaRegion条件キーを使用して、ReplicaRegionパラメータの値が指定されたリージョンのいずれかである場合にのみ、プリンシパルが ReplicateKeyオペレーションを呼び出すことを許可します。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey"
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

この条件キーは、 ReplicateKeyオペレーションへのアクセスのみを制御します。UpdatePrimaryRegion オペレーションへのアクセスを制御するには、kms:PrimaryRegion 条件キーを使用します。

kms:RetiringPrincipal

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:RetiringPrincipal	文字列 (リスト)	単一値	CreateGrant	キーポリシーと IAM ポリシー

この条件キーを使用して、リクエストの RetiringPrincipalパラメータの値に基づいてCreateGrantオペレーションへのアクセスを制御できます。例えば、CreateGrant リクエストの RetiringPrincipal が条件ステートメントの RetiringPrincipal と一致した場合にのみ、KMS キーの使用権限を作成します。

廃止プリンシパルを指定するには、 プリンシパルの Amazon リソースネーム (ARN) AWS を使用します。有効なプリンシパルには AWS アカウント、、IAM ユーザー、IAM ロール、フェデレーティッドユーザー、引き受けたロールユーザーが含まれます。プリンシパルの ARN 構文については、「IAM ユーザーガイド」のARNs」を参照してください。

次のキーポリシーステートメントの例では、ユーザーが KMS キーの許可を作成することを許可します。kms:RetiringPrincipal 条件キーは、グラントの廃止プリンシパルが であるCreateGrantリクエストへのアクセス許可を制限しますLimitedAdminRole。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RetiringPrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

以下の資料も参照してください。

- kms:GrantConstraintType
- kms:GrantIsForAWSResource
- kms:GrantOperations
- kms:GranteePrincipal

kms:ScheduleKeyDeletionPendingWindowInDays

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ScheduleKeyDeletionPendingWindowInDays	数値	単一値	ScheduleKeyDeletion	キーポリシーと IAM ポリシー

この条件キーを使用して、プリンシパルがScheduleKeyDeletionリクエストの PendingWindowInDaysパラメータで指定できる値を制限できます。

AWS KMS は、キーを削除する前に が待機する日数PendingWindowInDaysを 指定します。 AWS KMS では、7 ~ 30 日間の待機期間を指定できますが、 kms:ScheduleKeyDeletionPendingWindowInDays条件キーを使用して、有効な範囲内の最小待機期間を強制するなど、待機期間をさらに制限できます。

例えば、次のキーポリシーステートメントで は、kms:ScheduleKeyDeletionPendingWindowInDays 条件キーを使用して、待機期間が 21 日以内の場合にプリンシパルがキーの削除をスケジュールできないようにしています。

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
      }
  }
}
```

kms:SigningAlgorithm

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:SigningAlgorithm	文字列	単一値	Sign Verify	キーポリシーと IAM ポリシー

kms:SigningAlgorithm 条件キーを使用して、リクエストの SigningAlgorithmパラメータの値に基づいて、Sign および Verify オペレーションへのアクセスを制御できます。この条件キーは AWS KMS、 の外部の非対称 KMS キーペアでパブリックキーを使用して署名を検証するなど、 の外部で実行されるオペレーションには影響しません AWS KMS。

次のキーポリシーの例では、リクエストに使用される署名アルゴリズムが RSASSA_PSS アルゴリズム (RSASSA_PSS_SHA512 など) である場合にのみ、testers ロールを引き受けることができるユーザーが KMS キーを使用してメッセージに署名できるようにします。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
  }
}
```

以下の資料も参照してください。

- kms:EncryptionAlgorithm
- the section called “kms:MacAlgorithm”
- the section called “kms:MessageType”

kms:ValidTo

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ValidTo	タイムスタンプ	単一値	ImportKeyMaterial	キーポリシーと IAM ポリシー

kms:ValidTo 条件キーは、インポートされたキーマテリアルの有効期限を決定するリクエスト内の ValidToパラメータの値に基づいて、 ImportKeyMaterialオペレーションへのアクセスを制御します。この値は、Unix 時間で表現されます。

デフォルトでは、ValidTo パラメータは ImportKeyMaterial リクエストで必要です。ただし、ExpirationModelパラメータの値が の場合KEY_MATERIAL_DOES_NOT_EXPIRE、 ValidToパラメー

タは無効です。kms:ExpirationModel 条件キーを使用して、 ExpirationModelパラメータまたは特定のパラメータ値を要求することもできます。

次のポリシーステートメントの例では、キーマテリアルの KMS キーへのインポートをユーザーに許可します。kms:ValidTo 条件キーは、ImportKeyMaterial リクエストへのアクセス権限を制限します。ここで、ValidTo の値は 1546257599.0 (2018 年 12 月 31 日 午後 11:59:59) 以下となります。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
    }
  }
}
```

以下の資料も参照してください。

- kms:ExpirationModel
- kms:WrappingAlgorithm
- kms:WrappingKeySpec

kms:ViaService

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:ViaService	文字列	単一値	KMS キーリソースのオペレーション	キーポリシーと IAM ポリシー

kms:ViaService 条件キーは、KMS キーの使用を、指定された AWS サービスからのリクエストに制限します。各 kms:ViaService 条件キーに 1 つ以上のサービスを指定できます。このオペレーションは KMS キーリソースオペレーションである必要があります。つまり、特定の KMS キーに認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、アクションとリソースの表で、オペレーションの Resources 列の KMS key の値を探します。

例えば、次のキーポリシーステートメントでは kms:ViaService 条件キーを使用して、リクエストが ExampleRole に代わって米国西部 (オレゴン) リージョンの Amazon EC2 または Amazon RDS から送信された場合にのみ、カスタマーマネージドキーを指定されたアクションで使用できるようにします 。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "ec2.us-west-2.amazonaws.com",
        "rds.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

kms:ViaService 条件キーを使用して、特定のサービスからリクエストが送信された場合に KMS キーの使用許可を拒否することもできます。例えば、次のキーポリシーからのポリシーステートメントでは kms:ViaService 条件キーを使用して、ExampleRole の代わりに AWS Lambda からリクエストが送信された場合に、カスタマーマネージドキーが Encrypt オペレーションに使用されるのを防ぎます。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
          "lambda.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Important

`kms:ViaService` 条件キーを使用する場合は、サービスが AWS アカウントのプリンシパルの代わりにリクエストを行います。これらのプリンシパルは、次のアクセス許可が必要です。

- KMS キーを使用するアクセス許可。プリンシパルの代わりにサービスがカスタマーマネージドキーを使用できるようにするために、プリンシパルは統合されたサービスにこれらのアクセス許可を付与する必要があります。詳細については、「AWS のサービスで AWS KMS を使用する方法」を参照してください。
- 統合されたサービスを使用するアクセス権限。と統合する AWS サービスへのアクセスをユーザーに許可する方法の詳細については AWS KMS、統合サービスのドキュメントを参照してください。

すべての AWS マネージドキー は、キーポリシードキュメントで `kms:ViaService` 条件キーを使用します。この条件では、KMS キーを作成したサービスからリクエストされた場合にのみ、KMS キーの使用を許可します。のキーポリシーを表示するには AWS マネージドキー、 GetKeyPolicyオペレーションを使用します。

kms:ViaService 条件キーは IAM とキーポリシーのステートメントで有効です。指定するサービスは、AWS KMSと統合されていて、kms:ViaService 条件キーをサポートしている必要があります。

kms:ViaService 条件キーをサポートするサービス

次の表は、 と統合 AWS KMS され、カスタマーマネージドキーでの kms:ViaService条件キーの使用をサポートする AWS サービスの一覧です。この表のサービスは、一部のリージョンで利用できない場合があります。すべての AWS パーティションで AWS KMS ViaService 名前の .amazonaws.comサフィックスを使用します。

> **Note**
>
> この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

サービス名	AWS KMS ViaService 名前
AWS App Runner	apprunner. *AWS_region* .amazonaws.com
AWS AppFabric	appfabric. *AWS_region* .amazonaws.com
Amazon AppFlow	appflow.*AWS_region* .amazonaws.com
AWS Application Migration Service	mgn.*AWS_region* .amazonaws.com
Amazon Athena	athena.*AWS_region* .amazonaws.com
AWS Audit Manager	auditmanager. *AWS_region* .amazonaws.com
Amazon Aurora	rds.*AWS_region* .amazonaws.com
AWS Backup	backup.*AWS_region* .amazonaws.com
AWS Backup ゲートウェイ	backup-gateway. *AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Chime SDK	chimevoiceconnector. *AWS_region* .amazonaws.com
AWS CodeArtifact	codeartifact. *AWS_region* .amazonaws.com
Amazon CodeGuru Reviewer	codeguru-reviewer. *AWS_region* .amazonaws.com
Amazon Comprehend	comprehend. *AWS_region* .amazonaws.com
Amazon Connect	connect.*AWS_region* .amazonaws.com
Amazon Connect Customer Profiles	profile.*AWS_region* .amazonaws.com
Amazon Q in Connect	wisdom.*AWS_region* .amazonaws.com
AWS Database Migration Service (AWS DMS)	dms.*AWS_region* .amazonaws.com
AWS Directory Service	directoryservice. *AWS_region* .amazonaws.com
Amazon DynamoDB	dynamodb. *AWS_region* .amazonaws.com
Amazon DocumentDB	docdb-elastic. *AWS_region* .amazonaws.com
Amazon EC2 Systems Manager (SSM)	ssm.*AWS_region* .amazonaws.com
Amazon Elastic Block Store (Amazon EBS)	ec2.*AWS_region* .amazonaws.com (EBS のみ)
Amazon Elastic Container Registry (Amazon ECR)	ecr.*AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Elastic File System (Amazon EFS)	elasticfilesystem. *AWS_region* .amazonaws.com
Amazon ElastiCache	条件キーの値に両方の ViaService 名前を含めます。 • elasticache. *AWS_region* .amazonaws.com • dax.*AWS_region* .amazonaws.com
AWS Elemental MediaTailor	mediatailor. *AWS_region* .amazonaws.com
AWS エンティティ解決	entityresolution. *AWS_region* .amazonaws.com
Amazon FinSpace	finspace. *AWS_region* .amazonaws.com
Amazon Forecast	forecast. *AWS_region* .amazonaws.com
Amazon FSx	fsx.*AWS_region* .amazonaws.com
AWS Glue	glue.*AWS_region* .amazonaws.com
AWS Ground Station	groundstation. *AWS_region* .amazonaws.com
Amazon GuardDuty	malware-protection. *AWS_region* .amazonaws.com
AWS HealthLake	healthlake. *AWS_region* .amazonaws.com
AWS IoT SiteWise	iotsitewise. *AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Kendra	kendra.*AWS_region* .amazonaws.com
Amazon Keyspaces (Apache Cassandra 向け)	cassandra. *AWS_region* .amazonaws.com
Amazon Kinesis	kinesis.*AWS_region* .amazonaws.com
Amazon Data Firehose	firehose. *AWS_region* .amazonaws.com
Amazon Kinesis Video Streams	kinesisvideo. *AWS_region* .amazonaws.com
AWS Lambda	lambda.*AWS_region* .amazonaws.com
Amazon Lex	lex.*AWS_region* .amazonaws.com
AWS License Manager	license-manager. *AWS_region* .amazonaws.com
Amazon Location Service	geo.*AWS_region* .amazonaws.com
Amazon Lookout for Equipment	lookoutequipment. *AWS_region* .amazonaws.com
Amazon Lookout for Metrics	lookoutmetrics. *AWS_region* .amazonaws.com
Amazon Lookout for Vision	lookoutvision. *AWS_region* .amazonaws.com
Amazon Macie	macie.*AWS_region* .amazonaws.com
AWS Mainframe Modernization	m2.*AWS_region* .amazonaws.com
Amazon Managed Blockchain	managedblockchain. *AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	kafka.*AWS_region* .amazonaws.com
Amazon Managed Workflows for Apache Airflow (MWAA)	airflow.*AWS_region* .amazonaws.com
Amazon MemoryDB for Redis	memorydb. *AWS_region* .amazonaws.com
Amazon Monitron	monitron. *AWS_region* .amazonaws.com
Amazon MQ	mq.*AWS_region* .amazonaws.com
Amazon Neptune	rds.*AWS_region* .amazonaws.com
Amazon Nimble Studio	nimble.*AWS_region* .amazonaws.com
AWS HealthOmics	omics.*AWS_region* .amazonaws.com
Amazon OpenSearch サービス	es.*AWS_region* .amazonaws.com , aoss.*AWS_region* .amazonaws.com
AWS Proton	proton.*AWS_region* .amazonaws.com
Amazon Quantum Ledger Database (Amazon QLDB)	qldb.*AWS_region* .amazonaws.com
「Amazon RDS Performance Insights」	rds.*AWS_region* .amazonaws.com
Amazon Redshift	redshift. *AWS_region* .amazonaws.com
Amazon Redshift クエリエディタ V2	sqlworkbench. *AWS_region* .amazonaws.com
Amazon Redshift Serverless	redshift-serverless. *AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Rekognition	rekognition. *AWS_region* .amazonaws.com
Amazon Relational Database Service (Amazon RDS)	rds.*AWS_region* .amazonaws.com
Amazon 複製データストア	ards.*AWS_region* .amazonaws.com
Amazon SageMaker	sagemaker. *AWS_region* .amazonaws.com
AWS Secrets Manager	secretsmanager. *AWS_region* .amazonaws.com
Amazon Security Lake	securitylake. *AWS_region* .amazonaws.com
Amazon Simple Email Service (Amazon SES)	ses.*AWS_region* .amazonaws.com
Amazon Simple Notification Service (Amazon SNS)	sns.*AWS_region* .amazonaws.com
Amazon Simple Queue Service (Amazon SQS)	sqs.*AWS_region* .amazonaws.com
Amazon Simple Storage Service (Amazon S3)	s3.*AWS_region* .amazonaws.com
AWS Snowball	importexport. *AWS_region* .amazonaws.com
AWS Storage Gateway	storagegateway. *AWS_region* .amazonaws.com
AWS Systems Manager Incident Manager	ssm-incidents. *AWS_region* .amazonaws.com
AWS Systems Manager Incident Manager 連絡先	ssm-contacts. *AWS_region* .amazonaws.com

サービス名	AWS KMS ViaService 名前
Amazon Timestream	`timestream.` *`AWS_region`* `.amazonaws.com`
Amazon Translate	`translate.` *`AWS_region`* `.amazonaws.com`
AWS Verified Access	`verified-access.` *`AWS_region`* `.amazonaws.com`
Amazon WorkMail	`workmail.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces	`workspaces.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces シンクライアント	`thinclient.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces Web	`workspaces-web.` *`AWS_region`* `.amazonaws.com`
AWS X-Ray	`xray.`*`AWS_region`* `.amazonaws.com`

kms:WrappingAlgorithm

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
`kms:WrappingAlgorithm`	文字列	単一値	`GetParametersForImport`	キーポリシーと IAM ポリシー

この条件キーは、リクエスト内の WrappingAlgorithmパラメータの値に基づいて、GetParametersForImportオペレーションへのアクセスを制御します。この条件を使用して、インポートプロセス時にプリンシパルが特定のアルゴリズムを使用してキーマテリアルを暗号化するよう

要求できます。異なるラップアルゴリズムを指定すると必要なパブリックキーとインポートトークンのリクエストが失敗します。

次のキーポリシーステートメントの例では、kms:WrappingAlgorithm 条件キーを使用して、GetParametersForImport オペレーションを呼び出すアクセス許可をサンプルユーザーに付与しますが、RSAES_OAEP_SHA_1 ラップアルゴリズムの使用を阻止します。WrappingAlgorithm リクエストの GetParametersForImport が RSAES_OAEP_SHA_1 の場合、オペレーションは失敗します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

以下の資料も参照してください。

- kms:ExpirationModel
- kms:ValidTo
- kms:WrappingKeySpec

kms:WrappingKeySpec

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:WrappingKeySpec	文字列	単一値	GetParametersForImport	キーポリシーと IAM ポリシー

この条件キーは、リクエスト内の WrappingKeySpecパラメータの値に基づいて、GetParametersForImportオペレーションへのアクセスを制御します。この条件を使用して、インポートプロセス時にプリンシパルが特定のタイプのパブリックキーを使用するよう要求できます。リクエストで別のキータイプを指定すると、エラーになります。

WrappingKeySpec パラメータ値の有効な値は RSA_2048 のみであるため、ユーザーによるこの値の使用を無効にすることで、GetParametersForImport オペレーションを効率的に無効にすることができます。

次のポリシーステートメントの例では、kms:WrappingAlgorithm 条件キーを使用してリクエストの WrappingKeySpec が RSA_4096 になるようにします。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

以下の資料も参照してください。

- kms:ExpirationModel
- kms:ValidTo
- kms:WrappingAlgorithm

AWS KMSAWS Nitro Enclaves の 条件キー

AWS Nitro Enclaves は、機密性の高いデータを保護および処理するために、エンクレーブと呼ばれる分離されたコンピューティング環境を作成できる Amazon EC2 の機能です。 は、 AWS Nitro Enclaves をサポートする条件キー AWS KMS を提供します。これらの条件キーは、Nitro Enclave AWS KMS の へのリクエストに対してのみ有効です。

エンクレーブから署名付きアテステーションドキュメントを使用して Decrypt、GenerateDataKey、GenerateDataKeyPair、または GenerateRandom API オペレーションを呼び出すと、これらの APIsアテステーションドキュメントからのパブリックキーのレスポンスでプレーンテキストを暗号化し、プレーンテキストではなく暗号文を返します。この暗号文は、Enclave のプライベートキーを使用してのみ復号できます。詳細については、「AWS Nitro Enclaves が AWS KMS を使用する方法」を参照してください。

次の条件キーを使用すると、署名付きアテステーションドキュメントの内容に基づいて、これらのオペレーションのアクセス許可を制限できます。オペレーションを許可する前に、 はエンクレーブからのアテステーションドキュメントをこれらの AWS KMS 条件キーの値 AWS KMS と比較します。

kms:RecipientAttestation：ImageSha384

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ

`kms:RecipientAttestation:ImageSha384`	文字列	単一値	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				キーポリシーと IAM ポリシー

`kms:RecipientAttestation:ImageSha384` 条件キーは、リクエストの署名付きアテステーションドキュメントからのイメージダイジェストが条件キーの値と一致する場合、KMS キーを使用して `Decrypt`、`GenerateDataKey`、`GenerateDataKeyPair`、および `GenerateRandom` へのアクセスを制御します。`ImageSha384` 値は、アテステーションドキュメントの PCR0 に対応します。この条件キーは、リクエストの `Recipient`パラメータが AWS Nitro Enclave の署名付きアテステーションドキュメントを指定する場合にのみ有効です。

この値は、Nitro Enclaves の へのリクエスト AWS KMS のCloudTrailイベントにも含まれます。

> **Note**
>
> この条件キーは、IAM コンソールまたは IAM サービス認証リファレンスに表示されない場合でも、キーポリシーステートメントおよび IAM ポリシーステートメントで有効です。

例えば、次のキーポリシーステートメントでは、data-processingロールが Decrypt、、、GenerateDataKeyGenerateDataKeyPairおよび GenerateRandomオペレーションに KMS キーを使用することを許可します。kms:RecipientAttestation:ImageSha384 条件キーでは、リクエスト内のアテステーションドキュメントのイメージダイジェスト値 (PCR0) が条件内のイメージダイジェスト値と一致する場合にのみ、オペレーションを許可します。この条件キーは、リクエストのRecipientパラメータが AWS Nitro Enclave の署名付きアテステーションドキュメントを指定する場合にのみ有効です。

リクエストに AWS Nitro Enclave からの有効なアテステーションドキュメントが含まれていない場合、この条件が満たされないため、アクセス許可は拒否されます。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms:RecipientAttestation:PCR<PCR_ID>

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:RecipientAttestation:PCR<PCR_ID>	文字列	単一値	Decrypt GenerateDataKey GenerateDataKeyPair GenerateRandom	キーポリシーと IAM ポリシー

kms:RecipientAttestation:PCR<PCR_ID> 条件キーは、リクエスト内の署名付きアテステーションドキュメントからのプラットフォーム設定登録 (PCR) が条件キーの PCR と一致する場合にのみ、KMS を使用して Decrypt、GenerateDataKey、GenerateDataKeyPair、および GenerateRandom へのアクセスを制御します。この条件キーは、リクエストの Recipientパラメータが AWS Nitro Enclave からの署名付きアテステーションドキュメントを指定する場合にのみ有効です。

この値は、Nitro Enclaves AWS KMS の へのリクエストを表すCloudTrailイベントにも含まれます。

> **Note**
>
> この条件キーは、IAM コンソールまたは IAM サービス認証リファレンスに表示されない場合でも、キーポリシーステートメントおよび IAM ポリシーステートメントで有効です。

PCR 値を指定するには、次の形式を使用します。PCR ID を条件キー名に連結します。PCR 値は、最大 96 バイトの小文字の 16 進文字列である必要があります。

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

たとえば、次の条件キーは PCR1 の特定の値を指定します。これは、エンクレーブとブートストラッププロセスに使用されるカーネルのハッシュに対応します。

```
kms:RecipientAttestation:PCR1:
 "0x1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcdef
```

次のキーポリシーステートメントの例では、data-processing ロールに Decrypt オペレーションでの KMS キーの使用を許可します。

このステートメントの kms:RecipientAttestation:PCR 条件キーでは、リクエスト内の署名付きアテステーションドキュメントの PCR1 値が条件の kms:RecipientAttestation:PCR1 値と一致した場合にのみ、オペレーションを許可します。StringEqualsIgnoreCase ポリシー演算子を使用して、PCR 値の大文字と小文字を区別しない比較を要求します。

リクエストにアテステーションドキュメントが含まれない場合は、この条件が満たされないため、アクセス許可は拒否されます。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f15
    }
  }
}
```

AWS KMS の ABAC

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可の方法です。AWS KMS は、KMS キーに関連付けられたタグとエイリアスに基づいてカスタマーマネージドキーへのアクセスを制御することで、ABAC をサポートします。AWS KMS の ABAC を有効にするタグおよびエイリアスの条件キーは、ポリシーを編集したりグラントを管理したりすることなく、プリンシパルに KMS キーの使用を許可する強力で柔軟な方法を提供します。ただし、プリンシパルが誤ってアクセスを許可されたり拒否されないように、注意してこれらの機能を使用する必要があります。

ABAC を使用する場合は、タグとエイリアスを管理する許可が、アクセス制御許可になることに注意してください。タグまたはエイリアスに依存するポリシーをデプロイする前に、すべての KMS キーの既存のタグとエイリアスを把握していることを確認してください。エイリアスの追加、削除、更新時、およびキーのタグ付けおよびタグ解除時には、妥当な予防措置を講じます。タグとエイリアスを必要とするプリンシパルにのみ管理する許可を付与し、管理できるタグとエイリアスを制限します。

> **メモ**
>
> ABACを AWS KMS で使用する際は、タグとエイリアスを管理する許可をプリンシパルに付与することに注意してください。タグまたはエイリアスを変更すると、KMS キーに対するアクセス許可を、許可または拒否する可能性があります。キーポリシーを変更したり、グラントを作成したりする許可を持たないキー管理者も、タグやエイリアスを管理する許可があれば、KMS キーへのアクセスを制御できます。
>
> タグとエイリアスの変更が KMS キーの認可に影響を及ぼすまでに最長 5 分かかることがあります。最近の変更は、認可に影響を与える前に API オペレーションで表示される場合があります。
>
> エイリアスに基づいて KMS キーへのアクセスを制御するには、条件キーを使用する必要があります。ポリシーステートメントの `Resource` 要素のエイリアスを使用して KMS キーを表すことはできません。エイリアスが `Resource` 要素で表示される場合、ポリシーステートメントは、関連付けられた KMS キーではなく、エイリアスに適用されます。

詳細はこちら

- 例を含む、ABAC の AWS KMS サポートについての詳細は、エイリアスを使用して KMS キーへのアクセスを制御する および タグを使用してKMS キーへのアクセスを制御する を参照してください。
- AWS リソースへのアクセスを制御するタグの使用に関する一般情報の詳細は、IAM ユーザーガイドの What is ABAC for AWS? および Controlling Access to AWS Resources Using Resource Tags を参照してください。

AWS KMS のABAC 条件キー

タグとエイリアスに基づいて KMS キーへのアクセスを認可するには、キーポリシーまたは IAM ポリシーで次の条件キーを使用します。

ABAC 条件キー	説明	ポリシータイプ	AWS KMS オペレーション
aws:ResourceTag	KMS キーのタグ (キーと値) が、ポリシーのタグ (キーと値) またはタグパターンと一致する	IAM ポリシーのみ	KMS キーリソースのオペレーション [2]
aws:RequestTag/tag-key	リクエスト内のタグ (キーと値) が、ポリシー内のタグ (キーと値) またはタグパターンと一致する	キーポリシーと IAM ポリシー [1]	TagResource, UntagResource
aws:TagKeys	リクエスト内のタグキーが、ポリシーのタグキーと一致する	キーポリシーと IAM ポリシー [1]	TagResource, UntagResource
kms:ResourceAliases	KMS キーに関連付けられたエイリアスが、ポリシーのエイリアスまたはエイリアスパターンと一致する	IAM ポリシーのみ	KMS キーリソースのオペレーション [2]
kms:RequestAlias	リクエスト内の KMS キーを表すエイリアスが、ポリシーのエイリアスまたはエイリアスパターンと一致する。	キーポリシーと IAM ポリシー [1]	暗号化オペレーション 、DescribeKey、GetPublicKey

[1] キーポリシーで使用できる条件キーは、IAM ポリシーでも使用できます。ただし、キーポリシーによって許可される場合に限ります。

[2] KMS キーのリソースオペレーションは、特定の KMS キーに対して認可されるオペレーションです。KMS キーリソースオペレーションを識別するには、AWS KMS アクセス許可の表で、オペレーションの Resources 列の KMS キー値を探します。

例えば、これらの条件キーを使用して、次のポリシーを作成できます。

- 特定のエイリアスまたはエイリアスパターンを持つ KMS キーを使用するアクセス許可を付与する kms:ResourceAliases を備えた IAM ポリシー。これは、タグに依存するポリシーとは少し異なります。ポリシーでエイリアスパターンを使用できますが、各エイリアスは AWS アカウント とリージョンで、一意である必要があります。これにより、KMS キーのキー ARN をポリシーステートメントに表示せずに、選択した KMS キーのセットにポリシーを適用できます。セットの KMS キーを追加または削除するには、KMS キーのエイリアスを変更します。
- Encrypt リクエストがエイリアスを使用して KMS キーを識別する場合にのみ、Encrypt オペレーションでプリンシパルの KMS キー使用を許可する kms:RequestAlias を備えたキーポリシー。
- 特定のタグキーとタグ値を持つ KMS キーを使用するアクセス許可を拒否する aws:ResourceTag/*tag-key* を備えた IAMポリシー。これにより、KMS キーのキー ARN をポリシーステートメントに表示せずに、選択した KMS キーのセットにポリシーを適用できます。セットの KMS キーを追加または削除するには、KMS キーをタグ付けまたはタグ解除します。
- プリンシパルが "Purpose"="Test" KMS キータグのみを削除することを許可する aws:RequestTag/*tag-key* を備えた IAMポリシー。
- Restricted タグキーで KMS キーをタグ付けまたはタグ解除する許可を拒否する aws:TagKeys を備えた IAMポリシー。

ABAC は、アクセス管理を柔軟かつスケーラブルにします。例えば、aws:ResourceTag/*tag-key* 条件キーを使用して、KMS キーが Purpose=Test タグを持つ場合にのみ、プリンシパルが指定されたオペレーションに KMS キーを使用することを許可する IAM ポリシーを作成します。このポリシーは、AWS アカウント のすべてのリージョンの KMS キーに適用されます。

ユーザーまたはロールにアタッチされると、以下の IAM ポリシーにより、プリンシパルは指定されたオペレーションの Purpose=Test タグで既存のKMS キーを使用できるようになります。新規または既存の KMS キーにこのアクセスを提供するためにポリシーを変更する必要はありません。単に、KMS キーに Purpose=Test タグをアタッチします。同様に、Purpose=Test タグでこのアクセスを KMS キーから削除するには、タグを編集または削除します。

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

```
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

ただし、この機能を使用する場合は、タグとエイリアスを管理する際に注意が必要です。タグまたはエイリアスを追加、変更、削除する際に、KMS キーへのアクセスを誤って許可または拒否する可能性があります。キーポリシーを変更したり、グラントを作成したりする許可を持たないキー管理者も、タグとエイリアスを管理する許可があれば、KMS キーへのアクセスを制御できます。このリスクを軽減するには、タグおよびエイリアスを管理する許可の制限を検討します。例えば、選択したプリンシパルのみが Purpose=Test タグを管理できるようにします。詳細については、「エイリアスを使用して KMS キーへのアクセスを制御する」および「タグを使用してKMS キーへのアクセスを制御する」を参照してください。

タグまたはエイリアス

AWS KMS は、タグとエイリアスで ABAC をサポートします。どちらのオプションも、柔軟でスケーラブルなアクセス制御方法を提供しますが、互いに若干異なります。

タグを使用するか、特定の AWS 使用パターンに基づいてエイリアスを使用するかを決定します。例えば、すでにほとんどの管理者にタグ付け許可を付与している場合は、エイリアスに基づいて認可の方法を制御する方が簡単です。または、KMS キーごとのエイリアスのクォータに近づいている場合、タグに基づく認可の方法の方が得策かもしれません。

以下は、一般的な利点です。

タグベースのアクセスコントロールの利点

- 異なるタイプの AWS リソースで同じ認可メカニズム。

  同じタグまたはタグキーを使用して、Amazon Relational Database Service (Amazon RDS) クラスター、Amazon Elastic Block Store (Amazon EBS) ボリューム、KMS キーなど、複数のリソースタイプへのアクセスを制御できます。この機能により、従来のロールベースのアクセス制御よりも柔軟性が高い、複数の異なる認可モデルが可能になります。

- KMS キーグループへのアクセスを認可します。

  タグを使用すると、同じ AWS アカウント およびリージョンの KMS キーグループへのアクセスを管理できます。選択した KMS キーに同じタグまたはタグキーを割り当てます。次に、タグまたはタグキーに基づくシンプルな easy-to-maintain ポリシーステートメントを作成します。認可グループの KMS キーを追加または削除するには、タグを追加または削除します。ポリシーを編集する必要はありません。

エイリアスベースのアクセス制御の利点

- エイリアスに基づいて暗号化オペレーションへのアクセスを認可します。

  aws:RequestTag/tag-key を含む属性のリクエストベースのポリシー条件のほとんどは、属性を追加、編集、削除するオペレーションにのみ影響します。ただし、kms:RequestAlias 条件キーは、リクエスト内の KMS キーを識別するために使用されるエイリアスに基づいて、暗号化オペレーションへのアクセスを制御します。例えば、`Encrypt` オペレーションで KMS キーを使用するプリンシパルに、`KeyId` パラメータ値が `alias/restricted-key-1` の場合にのみ、アクセス許可を付与できます。この条件を満たすには、次のすべてが必要です。

  - KMS キーがそのエイリアスに関連付けられている必要があります。
  - リクエストで、KMS キーを識別するためにエイリアスを使用する必要があります。
  - `kms:RequestAlias` を条件として、プリンシパルが KMS キーを使用するアクセス許可が必要です。

  これは、アプリケーションが一般的にエイリアス名またはエイリアス ARN を使用して KMS キーを参照する場合に特に便利です。

- きわめて限定されたアクセス許可を付与します。

  エイリアスは AWS アカウント とリージョン内で、一意である必要があります 結果として、プリンシパルに、エイリアスに基づいて KMS キーへのアクセスを許可することは、タグに基づいてプリンシパルにアクセスを許可するよりもはるかに制限が厳しくなります。エイリアスとは異なり、タグは同じアカウントとリージョン内の複数の KMS キーに割り当てることができます。選択す

ると、エイリアスパターン (alias/test* など) を使用して、プリンシパルに同じアカウントとリージョン内の KMS キーグループへのアクセスを許可します。ただし、特定のエイリアスへのアクセスを許可または拒否すると、KMS キーのきわめて厳密な制御が可能になります。

AWS KMS の ABAC トラブルシューティング

タグとエイリアスに基づいて KMS キーへのアクセスを制御することは、便利で強力です。ただし、いくつかの予測可能なエラーが発生しやすいため、予防する必要があります。

タグの変更によりアクセスが変更される

タグが削除されるかその値が変更されると、そのタグのみに基づいて KMS キーにアクセスするプリンシパルは、KMS キーへのアクセスを拒否されます。これは、拒否ポリシーステートメントに含まれるタグが KMS キーに追加された場合にも発生します。ポリシー関連のタグを KMS キーに追加すると、KMS キーへのアクセスを拒否されるプリンシパルにアクセスを許可できます。

例えば、プリンシパルに、Project=Alpha タグに基づく KMS キーへのアクセス許可 (以下の例の、IAM ポリシーステートメントによって付与されるアクセス許可など) があるとします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

タグがその KMS キーから削除されるか、タグの値が変更されると、プリンシパルには指定されたオペレーションで KMS キーを使用するアクセス許可がなくなります。これは、プリンシパルがカス

タマーマネージドキーを使用する AWSサービスのデータを読み書きしようとすると明らかになります。タグの変更を追跡するには、ログで CloudTrail TagResourceまたは UntagResource エントリを確認します。

ポリシーを更新せずにアクセスを復元するには、KMS キーのタグを変更します。このアクションは、AWS KMS 全体で有効な期間中、短い期間を除いて影響は最小限です。このようなエラーを防ぐには、タグ付けとタグ解除の許可を必要なプリンシパルのみに付与し、タグ付け許可を、管理する必要があるタグに制限します。タグを変更する前にポリシーを検索して、タグに依存するアクセスを検出し、タグを持つすべてのリージョンで KMS キーを取得します。特定のタグが変更された場合は、Amazon CloudWatch アラームの作成を検討してください。

エイリアスの変更によるアクセス変更

エイリアスが削除されたり、別の KMS キーに関連付けられている場合、そのエイリアスのみに基づいて KMS キーにアクセスするプリンシパルは、KMS キーへのアクセスを拒否されます。これは、KMS キー に関連付けられたエイリアスが拒否ポリシーステートメントに含まれる場合にも発生します。ポリシー関連のエイリアスを KMS キーに追加すると、KMS キーへのアクセスを拒否されるプリンシパルへのアクセスを許可できます。

例えば、次の IAM ポリシーステートメントでは、kms:ResourceAliases 条件キーを使用して、指定されたエイリアスのいずれかを持つアカウントの異なるリージョンの KMS キーへのアクセスを許可します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "kms:ResourceAliases": [
            "alias/ProjectAlpha",
            "alias/ProjectAlpha_Test",
            "alias/ProjectAlpha_Dev"
```

```
            ]
          }
        }
      }
    ]
}
```

エイリアスの変更を追跡するには、 CloudTrail ログで CreateAlias、、UpdateAliasおよびDeleteAliasエントリを確認します。

ポリシーを更新せずにアクセスを復元するには、KMS キーに関連付けられたエイリアスを変更します。各エイリアスは、アカウントおよびリージョン内の 1 つの KMS キーのみにしか関連付けできないため、エイリアスの管理は、タグの管理よりも若干難しくなります。ある KMS キーの一部のプリンシパルへのアクセスを復元すると、異なる KMS キーの同じプリンシパルまたは他のプリンシパルへのアクセスが拒否されることがあります。

このエラーを防ぐには、エイリアス管理許可を必要なプリンシパルのみに付与し、エイリアス管理許可の制限を管理する必要があるエイリアスに制限します。エイリアスを更新または削除する前に、ポリシーを検索してエイリアスに依存するアクセスを検出し、エイリアスに関連付けられているすべてのリージョンで KMS キーを検索します。

エイリアスクォータによりアクセスが拒否された

kms:ResourceAliases 条件によって KMS キーの使用が許可されているユーザーは、KMS キーがそのアカウントとリージョンの KMS キーあたりのデフォルトエイリアスを超えるとAccessDenied、例外を受け取ります。

アクセスを復元するには、KMS キーに関連付けられたエイリアスを削除して、クォータに適合させます。または、別のメカニズムを使用して、ユーザーに KMS キーへのアクセスを許可します。

認可変更の遅延

タグとエイリアスを変更すると、KMS キーの認可に影響を及ぼすまでに最大 5 分かかる場合があります。結果として、タグやエイリアスの変更が認可に影響を与える前に、API オペレーションからのレスポンスに反映される可能性があります。この遅延はほとんどの場合、結果整合性の短い遅延よりも長くなる可能性があり、AWS KMS オペレーションに最も大きく影響します。

例えば、特定のプリンシパルに "Purpose"="Test" タグで KMS キーの使用を許可する IAM ポリシーなどです。次に、"Purpose"="Test"タグを KMS キーに追加します。TagResource オペレーションが完了し、ListResourceTagsレスポンスによってタグが KMS キーに割り当てられていることが確認されますが、プリンシパルは最大 5 分間 KMS キーにアクセスできない場合があります。

エラーを防ぐには、この予想される遅延をコードに組み込みます。

エイリアスの更新によるリクエストの失敗

エイリアスを更新するときは、既存のエイリアスを別の KMS キーに関連付けます。

エイリアス名またはエイリアス ARN を指定する復号および ReEncryptリクエストは、エイリアスが暗号文を暗号化しなかった KMS キーに関連付けられているため、失敗する可能性があります。この状況は通常、IncorrectKeyException または NotFoundException を返します。または、リクエストに KeyId または DestinationKeyId パラメータがない場合、発信者が暗号文を暗号化した KMS キーにアクセスできなくなったため、AccessDenied の例外により、オペレーションは失敗する可能性があります。

CreateAlias、、および CloudTrail ログエントリのDeleteAliasログを確認することでUpdateAlias、変更を追跡できます。ListAliases レスポンスの LastUpdatedDateフィールドの値を使用して変更を検出することもできます。

例えば、次のレスポンスListAliases例は、 ProjectAlpha_Test kms:ResourceAliases条件のエイリアスが更新されたことを示しています。この結果、エイリアスに基づくアクセス許可を持つプリンシパルは、以前に関連付けられた KMS キーにアクセスできなくなります。代わりに、新しく関連付けられた KMS キーにアクセスできます。

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/ProjectAlpha`)]'

{
    "Aliases": [
        {
            "AliasName": "alias/ProjectAlpha_Test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ProjectAlpha_Test",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1566518783.394,
            "LastUpdatedDate": 1605308931.903
        },
        {
            "AliasName": "alias/ProjectAlpha_Restricted",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/
ProjectAlpha_Restricted",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1553410800.010,
            "LastUpdatedDate": 1553410800.010
        }
```

```
    ]
}
```

この変更を回避するのは簡単ではありません。エイリアスを再度更新して、元の KMS キーに関連付けることができます。ただし、オペレーションを行う前に、その変更が現在関連付けられている KMS キーに及ぼす影響を考慮する必要があります。プリンシパルが暗号化オペレーションで後者の KMS キーを使用した場合は、引き続きそのキーにアクセスする必要がある可能性もあります。この場合はポリシーを更新し、プリンシパルに両方の KMS キーを使用するアクセス許可があることを確認します。

エイリアスを更新する前にポリシーを検索して、エイリアスに依存するアクセスを検出することで、このようなエラーを防ぐことができます。次に、エイリアスに関連付けられているすべてのリージョンで KMS キーを取得します。エイリアス管理許可を、必要とするプリンシパルにのみ付与し、エイリアス管理許可の制限を、管理する必要があるエイリアスに設定します。

他のアカウントのユーザーに KMS キーの使用を許可する

異なる AWS アカウント で、ユーザーまたはロールに、アカウントの KMS キーの使用を許可できます。クロスアカウントアクセスには、KMS キーのキーポリシーと、外部ユーザーアカウントの IAM ポリシーのアクセス許可が必要です。

クロスアカウント許可は、以下のオペレーションに対してのみ有効です。

- 暗号化オペレーション
- CreateGrant
- DescribeKey
- GetKeyRotationStatus
- GetPublicKey
- ListGrants
- RetireGrant
- RevokeGrant

別のアカウントのユーザーに他のオペレーションのアクセス許可を付与しても、それらのアクセス許可には効果がありません。例えば、別のアカウントのプリンシパルに kms:ListKeys アクセス許可を IAM ポリシーで付与した場合、または kms:ScheduleKeyDeletion アクセス許可をキーポリシーで

KMS キーに付与した場合、リソースでこれらのオペレーションを呼び出そうとした場合も失敗します。

別のアカウントの AWS KMS オペレーションで KMS キーを使用する方法の詳細については、AWS KMS アクセス許可 および 他のアカウントで KMS キーを使用する のクロスアカウント使用の記事を参照してください。また、AWS Key Management Service API リファレンスの各 API の説明に、クロスアカウント使用のセクションがあります。

> **Warning**
>
> KMS キー使用のアクセス許可をプリンシパルに付与する際は注意してください。可能な限り、 最小権限 の原則に従ってください。オペレーションで必要な KMS キーのみに、アクセスを許可します。
> また、使い慣れていない KMS キー、特に別のアカウントの KMS キーを使用する際にも注意してください。悪意のあるユーザーは、ユーザーまたはユーザーのアカウントに関する情報を取得するために、KMS キーの使用許可を付与する可能性があります。
> ポリシーを使用してアカウント内のリソースを保護する方法については、「IAM ポリシーのベストプラクティス」を参照してください。

別のアカウントのユーザーとロールに KMS キーを使用するアクセス許可を付与するには、2 つの異なるタイプのポリシーを使用する必要があります。

- KMS キーのキーポリシーでは、KMS キーを使用するアクセス許可を、外部アカウント (または外部アカウントのユーザーとロール) に付与する必要があります。キーポリシーは、KMS キーを所有するアカウントにあります。
- 外部アカウントの IAM ポリシーは 、キーポリシーのアクセス権限をそのユーザーとロールに委任する必要があります。これらのポリシーは外部アカウントで設定され、そのアカウントのユーザーとロールにアクセス許可を与えます。

キーポリシーによって、KMS キーにアクセスできるユーザーが決定されます。IAM ポリシーによって、KMS キーにアクセスするユーザーが決定されます。キーポリシーも IAM ポリシーも十分ではありません。両方を変更する必要があります。

キーポリシーを編集するには、 のポリシービューを使用するAWS Management Consoleか、CreateKey または PutKeyPolicyオペレーションを使用します。KMS キーの作成時にキーポリシーを設定する方法については、他のアカウントで使用できる KMS キーを作成する を参照してください。

IAM ポリシーの編集については、「AWS KMS で IAM ポリシーを使用する」を参照してください。

キーポリシーと IAM ポリシーが連携して、別のアカウントで KMS キーの使用を許可する方法を示す例については、例 2: ユーザーが別の AWS アカウント の KMS キーを使用するためのアクセス許可を持つロールを引き受ける を参照してください。

KMS キーのクロスアカウント AWS KMS オペレーションの結果は、AWS CloudTrail ログに表示されます。他のアカウントで KMS キーを使用するオペレーションは、発信者のアカウントと KMS キー所有者のアカウントの両方に記録されます。

トピック

- ステップ 1: ローカルアカウントにキーポリシーステートメントを追加する
- ステップ 2: 外部アカウントに IAM ポリシーを追加する
- 他のアカウントで使用できる KMS キーを作成する
- 外部 KMS キーの使用を許可する AWS のサービス
- 他のアカウントで KMS キーを使用する

> **Note**
>
> このトピックの例は、キーポリシーと IAM ポリシーを併用して、KMS キーへのアクセス権を付与し、それらを制限する方法を示しています。これらの一般的な例は、特定の AWS のサービスが必要とする KMS キーの許可を表すものではありません。AWS のサービスが必要とする許可の詳細については、サービスドキュメントの暗号化トピックを参照してください。

ステップ 1: ローカルアカウントにキーポリシーステートメントを追加する

KMS キーのキーポリシーは、KMS キーにアクセスできるユーザーと、実行できるオペレーションの主要な決定要因です。キーポリシーは常に、KMS キーを所有するアカウントにあります。IAM ポリシーとは異なり、キーポリシーはリソースを指定しません。リソースは、キーポリシーに関連付けられている KMS キーです。クロスアカウント許可を付与する場合、KMS キーのキーポリシーでは、KMS キーを使用するアクセス許可を、外部アカウント (または外部アカウントのユーザーとロール) に付与する必要があります。

KMS キーを使用する許可を外部アカウントに付与するには、外部アカウントを指定するステートメントをキーポリシーに追加します。キーポリシーの Principal 要素に、外部アカウントの Amazon リソースネーム (ARN) を入力します。

キーポリシーで外部アカウントを指定すると、外部アカウントの IAM 管理者は IAM ポリシーを使用して、外部アカウントのすべてのユーザーおよびロールにこれらのアクセス権限を委任できます。また、ユーザーおよびロールが実行できるキーポリシーで指定されたアクションを決定することもできます。

外部アカウントとそのプリンシパルに付与されたアクセス許可は、KMS キーとそのキーポリシーをホストするリージョンで外部アカウントが有効になっている場合にのみ有効です。デフォルトで有効になっていないリージョン (「オプトインリージョン」) については、「AWS 全般のリファレンス」の「AWS リージョン の管理」を参照してください。

例えば、アカウント 444455556666 にアカウント 111122223333 の対称暗号化 KMS キーの使用を許可するとします。これを行うには、次の例のようなポリシーステートメントを、アカウント 111122223333 の KMS キーのキーポリシーに追加します。このポリシーステートメントは、対称暗号化 KMS キーの暗号化オペレーションで KMS キーを使用する許可を外部アカウントである 444455556666 に付与します。

> **Note**
>
> 次の例は、KMS キーを別のアカウントと共有するためのサンプルキーポリシーを示しています。例の Sid、Principal、Action の値を、KMS キーの使用目的に合った有効な値に置き換えます。

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:root"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
```

```
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

外部アカウントにアクセス許可を付与する代わりに、キーポリシーで特定の外部ユーザーとロールを指定できます。ただし、外部アカウントの IAM 管理者が適切な IAM ポリシーを IAM 管理者のアイデンティティにアタッチするまで、これらのユーザーとロールは KMS キーを使用できません。IAM ポリシーは、キーポリシーで指定されている外部ユーザーとロールのすべてまたはサブセットにアクセス許可を与えることができます。また、キーポリシーで指定されたアクションのすべてまたはサブセットを許可できます。

キーポリシーで ID を指定すると、外部アカウントの IAM 管理者が提供できるアクセス権限が制限されます。ただし、2 つのアカウントを使用したポリシー管理はより複雑になります。例えば、ユーザーまたはロールを追加する必要があるとします。IAM 管理者のアイデンティティを、KMS キーを所有するアカウントのキーポリシーに追加し、アイデンティティのアカウントに IAM ポリシーを作成する必要があります。

キーポリシーで特定の外部ユーザーまたはロールを指定するには、`Principal` 要素に、外部アカウントのユーザーまたはロールの Amazon リソースネーム (ARN) を入力します。

例えば、次のキーポリシーステートメントの例では、アカウント `444455556666` の `ExampleRole` に、アカウント `111122223333` の KMS キーの使用を許可します。このキーポリシーステートメントは、対称暗号化 KMS キーの暗号化オペレーションで KMS キーを使用する許可を外部アカウントである `444455556666` に付与します。

> **Note**
>
> 次の例は、KMS キーを別のアカウントと共有するためのサンプルキーポリシーを示しています。例の `Sid`、`Principal`、`Action` の値を、KMS キーの使用目的に合った有効な値に置き換えます。

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
    },
    "Action": [
```

```
            "kms:Encrypt",
            "kms:Decrypt",
            "kms:ReEncrypt*",
            "kms:GenerateDataKey*",
            "kms:DescribeKey"
        ],
        "Resource": "*"
}
```

> **Note**
>
> 条件を使用してキーポリシーを制限しない限り、アクセス許可を付与するキーポリシーステートメントで、プリンシパルをアスタリスク (*) に設定しないでください。アスタリスクは、別のポリシーステートメントが明示的に拒否しない限り、すべての AWS アカウント のすべてのアイデンティティに、KMS キーを使用するアクセス許可を付与します。他の AWS アカウント のユーザーは、各自のアカウントに対応するアクセス権限があるときにはいつでも KMS キーを使用できます。

また、外部アカウントに付与するアクセス許可を決定する必要があります。KMS キーに対するアクセス許可のリストについては、AWS KMS アクセス許可 を参照してください 。

暗号化オペレーションで KMS キーを使用したり、AWS KMS と統合された AWS のサービスで KMS キーを使用したりするためのアクセス許可を外部アカウントに付与できます。これを行うには、AWS Management Console の Key Users セクションを使用します。詳細については、「他のアカウントで使用できる KMS キーを作成する」を参照してください。

キーポリシーで他のアクセス許可を指定するには、キーポリシードキュメントを編集します。例えば、復号するが暗号化しないアクセス許可をユーザーに付与したり、KMS キーが表示されても使用できないアクセス許可を付与したりできます。キーポリシードキュメントを編集するには、 のポリシービュー、AWS Management Consoleまたは CreateKeyまたは PutKeyPolicyオペレーションを使用します。

ステップ 2: 外部アカウントに IAM ポリシーを追加する

KMS キーを所有するアカウントのキーポリシーは、アクセス許可の有効範囲を設定します。ただし、外部アカウントのユーザーとロールは、それらのアクセス許可を委任する IAM ポリシーをアタッチするか、権限を使用して KMS キーへのアクセスを管理するまで、KMS キーを使用できません。IAM ポリシーは外部アカウントで設定されます。

キーポリシーが外部アカウントにアクセス許可を与える場合は、アカウント内の任意のユーザーまたはロールに IAM ポリシーをアタッチできます。ただし、キーポリシーが指定したユーザーまたはロールにアクセス許可を付与する場合、IAM ポリシーでは、指定したユーザーとロールのすべてまたはサブセットにのみそれらのアクセス許可を付与できます。IAM ポリシーが他の外部ユーザーまたはロールにKMS キーへのアクセスを許可しても、影響はありません。

キーポリシーは、IAM ポリシー内のアクションも制限します。IAM ポリシーは、キーポリシーで指定されたアクションのすべてまたはサブセットを委任できます。IAM ポリシーに、キーポリシーで指定されていないアクションがリストされている場合、それらのアクセス権限は有効ではありません。

以下の IAM ポリシー例では、プリンシパルがアカウント 111122223333 の KMS キーを暗号化オペレーションに使用することを許可します。アカウント内のユーザーとロールにこの権限を付与するには 444455556666, 、アカウント内のユーザーまたはロールにポリシーをアタッチ444455556666します。

> **Note**
>
> 次の例は、KMS キーを別のアカウントと共有するための IAM ポリシーのサンプルを示しています。例の Sid、Resource、Action の値を、KMS キーの使用目的に合った有効な値に置き換えます。

```
{
    "Sid": "AllowUseOfKeyInAccount111122223333",
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

このポリシーに関する以下の詳細情報に注意してください。

- キーポリシーとは異なり、IAM ポリシーステートメントには Principal 要素が含まれていません。IAM ポリシーでは、プリンシパルはポリシーがアタッチされている ID です。
- IAM ポリシーの Resource エレメントは、プリンシパルが使用できる KMS キーを識別します。KMS キーを指定するには、そのキー ARN を Resource 要素に追加します。
- Resource 要素には複数の KMS キーを指定できます。ただし、Resource 要素で特定の KMS キーを指定しないと、意図したよりも多くの KMS キーへのアクセス許可を、誤って付与する可能性があります。
- 外部ユーザーが AWS KMS と統合された AWS サービスで KMS キーを使用できるようにするには、キーポリシーまたは IAM ポリシーにアクセス許可を追加する必要があります。詳細については、「外部 KMS キーの使用を許可する AWS のサービス」を参照してください。

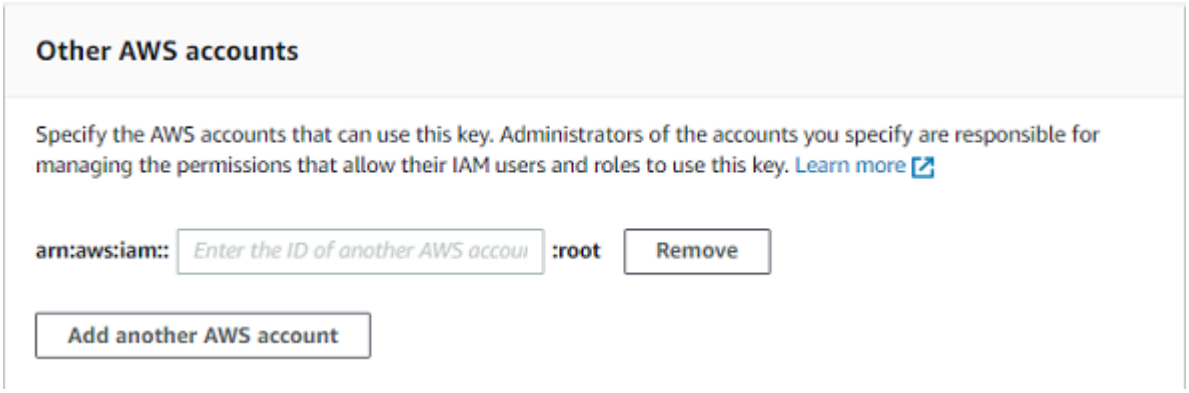
IAM ポリシーのオペレーションの詳細については、「IAM ポリシー」を参照してください。

他のアカウントで使用できる KMS キーを作成する

CreateKey オペレーションを使用して KMS キーを作成する場合、その Policyパラメータを使用して、KMS キーを使用するアクセス許可を外部アカウントまたは外部ユーザーとロールに付与するキーポリシーを指定できます。また、キーとロールがキー ポリシーで指定されている場合でも、これらのアクセス権限をアカウントのユーザーとロールに委任する IAM ポリシーを外部アカウントに追加する必要があります。キーポリシーは、 PutKeyPolicyオペレーションを使用していつでも変更できます。

AWS Management Console で KMS キーを作成するときは、そのキーポリシーも作成します。キー管理者セクションとキーユーザーセクションでアイデンティティを選択すると、AWS KMS はこれらのアイデンティティのポリシーステートメントを KMS キーのキーポリシーに追加します。

Key Users セクションでは、外部アカウントをキーユーザーとして追加することもできます。



外部アカウントのアカウント ID を入力すると、 AWS KMS はキーポリシーに 2 つのステートメントを追加します。このアクションは、キーポリシーにのみ影響します。外部アカウントのユーザーとロールは、IAM ポリシーをアタッチしてこれらのアクセス許可の一部またはすべてを付与するまで、KMS キーを使用できません。

最初のポリシーステートメントでは、暗号化オペレーションで KMS キーを使用するアクセス許可を外部アカウントに付与します。

> **Note**
>
> 次の例は、KMS キーを別のアカウントと共有するためのサンプルキーポリシーを示しています。例の `Sid`、`Principal`、`Action` の値を、KMS キーの使用目的に合った有効な値に置き換えます。

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

2 番目のポリシーステートメントでは、外部アカウントが KMS キーで権限を作成、表示、取り消すことを許可します。ただし、リクエストが AWS KMS と統合された AWS サービスからのものである場合に限ります。これらのアクセス許可は、KMS キーを使用するためにユーザーデータを暗号化するなどの、他の AWS のサービスを許可します。

これらのアクセス許可は、Amazon WorkMail などの AWSのサービスでユーザーデータを暗号化する KMS キー用に設計されています。これらのサービスでは、通常、ユーザーに代わって KMS キーを使用するために必要なアクセス許可を取得するために権限を使用します。詳細については、「外部 KMS キーの使用を許可する AWS のサービス」を参照してください。

```
{
    "Sid": "Allow attachment of persistent resources",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:root"
    },
    "Action": [
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": "true"
        }
    }
}
```

これらのアクセス許可がニーズに合わない場合は、コンソールのポリシービューまたは PutKeyPolicyオペレーションを使用して編集できます。外部アカウントにアクセス許可を付与する代わりに、特定の外部ユーザーとロールを指定できます。ポリシーで指定するアクションを変更できます。また、グローバル条件と AWS KMS ポリシー条件を使用して、アクセス許可を絞り込むことができます。

外部 KMS キーの使用を許可する AWS のサービス

別のアカウントのユーザーに、AWS KMS と統合されたサービスで KMS キーを使用するアクセス許可を付与できます。例えば、外部アカウントのユーザーは KMS キーを使用して、Amazon S3 バケット内のオブジェクトを暗号化したり、AWS Secrets Manager に保存されたシークレットを暗号化したりできます。

キーポリシーでは、KMS キーを使用するための許可を外部ユーザーまたは外部ユーザーのアカウントに付与する必要があります。さらに、AWS のサービス を使用する許可をユーザーに付与するアイデンティティに、IAM ポリシーをアタッチする必要があります。また、このサービスでは、ユーザーがキーポリシーまたは IAM ポリシーで、追加のアクセス許可を持っている必要があります。カスタマーマネージドキーに対して AWS のサービス で必要になる許可のリストについては、サービスのユーザーガイドまたはデベロッパーガイドの「セキュリティ」トピックにある「Data Protection topic」(データ保護トピック) を参照してください。

他のアカウントで KMS キーを使用する

KMS キーを使用するアクセス許可を別の AWS アカウント で持っている場合、AWS Management Console、AWS SDK、AWS CLI、AWS Tools for PowerShell で KMS キーを使用できます。

シェルコマンドまたは API リクエストで別のアカウントの KMS キーを識別するには、次のキー識別子を使用します。

- 暗号化オペレーション 、DescribeKey、および ではGetPublicKey、KMS キーのキー ARN またはエイリアス ARN を使用します。
- CreateGrant、GetKeyRotationStatus、ListGrants、および にはRevokeGrant、KMS キーのキー ARN を使用します。

キー ID またはエイリアス名のみを入力する場合、AWS では、KMS キーがアカウントにあることを前提とします。

AWS KMS コンソールでは、他のアカウントの KMS キーを使用するためのアクセス許可がある場合でも、KMS キーは表示されません。他の AWS サービスのコンソールに表示される KMS キーのリストには、他のアカウントの KMS キーは含まれません。

AWS サービスのコンソールで異なるアカウントの KMS キーを指定するには、KMS キーのキー ARN またはエイリアス ARN を入力する必要があります。必要なキー識別子はサービスによって異なり、サービスコンソールとその API オペレーションとの間でも異なる場合があります。詳細については、サービスのドキュメントを参照してください。

AWS KMS のサービスにリンクされたロールの使用

AWS Key Management Service では AWS Identity and Access Management (IAM) のサービスリンクロールを使用します。サービスリンクロールは、AWS KMS 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、AWS KMS によって定義されたロールであり、ユーザーに代わってサービスから AWS の他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を備えています。

サービスリンク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、AWS KMS の設定が簡単になります。このサービスリンクロールのアクセス許可は AWS KMS で定義します。特に定義されている場合を除き、AWS KMS のみがそのロールを引き受けることができます。定義されるアクセス許可には、信頼ポリシーと許可ポリシーが含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスにリンクされたロールを削除するには、まずその関連リソースを削除します。これにより、リソースにアクセスするための許可を意図せず削除することが防止され、AWS KMS リソースが保護されます。

サービスリンクロールをサポートする他のサービスについては、「IAM と連携する AWS のサービス」を参照して、[サービスリンクロール] 列が [はい] のサービスを探してください。[はい] のリンクを選択すると、該当するサービスのサービスリンクロールに関するドキュメントが表示されます。

AWS KMS カスタムキーストア用のサービスにリンクされたロールのアクセス許可

AWS KMS は、 という名前のサービスにリンクされたロールAWSServiceRoleForKeyManagementServiceCustomKeyStoresを使用して、カスタムキーストアをサポートします。このサービスリンクロールは、AWS KMS に、AWS CloudHSM クラスターを表示して、カスタムキーストアとその AWS CloudHSM クラスターをサポートするネットワークインフラストラクチャを作成する許可を付与します。AWS KMS はこのロールを、カスタムキーストアを作成する場合にのみ作成します。このサービスにリンクされたロールを直接作成することはできません。

サービスにリンクされたロール AWSServiceRoleForKeyManagementServiceCustomKeyStores は、このロールを引き受けるために `cks.kms.amazonaws.com` を信頼します。その結果、AWS KMS だけがこのサービスにリンクされたロールを引き受けることができます。

ロールのアクセス許可は、カスタムキーストアを AWS CloudHSM クラスターに接続するために AWS KMS が実行するアクションに限定されます。AWS KMS に対して追加のアクセス許可は付与されません。たとえば、AWS KMS に、AWS CloudHSM クラスター、HSM、またはバックアップを作成、管理、または削除するためのアクセス許可はありません。

AWSServiceRoleForKeyManagementServiceCustomKeyStores ロールの詳細とアクセス許可のリスト、およびロールの表示、ロールの説明の編集、ロールの削除、AWS KMS によるロールの再作成の手順については、「AWS CloudHSM および Amazon EC2 リソースの管理を AWS KMS に認可する」を参照してください。

AWS KMS マルチリージョンキーのサービスリンクロールの許可

AWS KMS は、 という名前のサービスにリンクされたロールAWSServiceRoleForKeyManagementServiceMultiRegionKeysを使用して、マルチリージョンキーをサポートします。このサービスリンクロールは、AWS KMS にアクセス許可を付与して、マルチ

リージョンのプライマリキーのキーマテリアルに対する変更をレプリカキーと同期します。AWS KMS はこのロールを、マルチリージョンプライマリキーを作成する場合にのみ作成します。このサービスにリンクされたロールを直接作成することはできません。

サービスにリンクされたロール AWSServiceRoleForKeyManagementServiceMultiRegionKeys は、このロールを引き受けるために mrk.kms.amazonaws.com を信頼します。その結果、AWS KMS だけがこのサービスにリンクされたロールを引き受けることができます。ロールのアクセス許可は、AWS KMS が実行するアクションに制限され、キーマテリアルと関連するマルチリージョンキーの同期を維持します。AWS KMS に対して追加のアクセス許可は付与されません。

AWSServiceRoleForKeyManagementServiceMultiRegionKeys ロールの詳細とアクセス許可のリスト、およびロールの表示、ロールの説明の編集、ロールの削除、AWS KMS によるロールの再作成の手順については、「マルチリージョンキーの同期を AWS KMS に認可する」を参照してください。

AWS マネージドポリシーの AWS KMS 更新

このサービスがこれらの変更の追跡を開始してからの、AWS KMS の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動通知については、[AWS KMS ドキュメント履歴] ページの RSS フィードを購読してください。

変更	説明	日付
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy – 既存のポリシーの更新	AWS KMS は、AWS CloudHSM クラスターが含まれる VPC 内の変更を監視するための ec2:DescribeVpcs 、ec2:DescribeNetworkAcls 、および ec2:DescribeNetworkInterfaces 許可を追加して、障害が発生した場合に AWS KMS が明瞭なエラーメッセージを提供できるようにしました。	2023 年 11 月 10 日
AWS KMS は変更の追跡を開始しました	AWS KMS が AWS マネージドポリシーの変更の追跡を開始しました。	2023 年 11 月 10 日

AWS KMS でハイブリッドポスト量子 TLS を使用する

AWS Key Management Service (AWS KMS) は、Transport Layer Security (TLS) ネットワーク暗号化プロトコル用のハイブリッドポスト量子キー交換オプションをサポートします。この TLS オプションは、AWS KMS API エンドポイントに接続するときに使用できます。この機能はポスト量子アルゴリズムが標準化される前に提供されているため、これらのキー交換プロトコルの AWS KMS コールへの影響のテストを開始できます。これらのオプションのハイブリッドポスト量子キー交換機能は、現在使用している TLS 暗号化と同等以上に安全であり、セキュリティ上のさらなる長期的な利点をもたらす可能性があります。ただし、現在使用されている従来のキー交換プロトコルと比較して、レイテンシーとスループットに影響します。

AWS Key Management Service (AWS KMS) に送信されたデータは、Transport Layer Security (TLS) 接続が提供する暗号化によって転送中に保護されます。AWS KMS が TLS セッションでサポートしている従来の暗号スイートにより、現在のテクノロジーではキー交換メカニズムに対するブルートフォース攻撃は実行不可能です。しかし、大規模な量子コンピューティングが将来実用的になると、TLS 鍵交換メカニズムで使用される従来の暗号スイートは、これらの攻撃の影響を受けやすくなります。TLS 接続を介して渡されるデータの長期的な機密性に依存するアプリケーションを開発している場合は、大規模な量子コンピュータが使用できるようになる前に、ポスト量子暗号化に移行する計画を検討する必要があります。AWS は、この未来に備えるために取り組んでいます。また、皆さんにも十分に備えてほしいと思っています。

潜在的な将来の攻撃から現在暗号化されたデータを保護するために、AWS は量子耐性またはポスト量子アルゴリズムを開発する暗号化コミュニティに参加しています。従来の要素とポスト量子要素を組み合わせたハイブリッドポスト量子キー交換暗号スイートを AWS KMS に実装したことで、TLS 接続が従来の暗号スイートと同等以上の強度になりました。

これらのハイブリッド暗号スイートは、ほとんどの AWS リージョン の本稼働用ワークロードで使用できます。ただし、ハイブリッド暗号スイートのパフォーマンス特性と帯域幅要件は従来のキー交換メカニズムのものとは異なるため、異なる条件下での AWS KMS API コールでテストすることをお勧めします。

フィードバック

これまでと同様、皆様のフィードバックや私たちのオープンソースリポジトリへの参加はいつでも歓迎です。特に、この新しい種類の TLS トラフィックが皆様のインフラストラクチャとどのように相互作用するかをぜひお聞かせください。

- このトピックに関するフィードバックを提供するには、このページの右下隅にある [フィードバック] リンクを使用してください。

- これらのハイブリッド暗号スイートは、 の s2n-tlsリポジトリのオープンソースで開発されています GitHub。暗号スイートのユーザビリティに関するフィードバックを提供したり、新しいテスト条件や結果を共有したりするには、s2n-tls リポジトリで課題を作成してください。
- aws-kms-pq-tls-example GitHub リポジトリAWS KMSでハイブリッドポスト量子 TLS を で使用するためのコードサンプルを記述しています。ハイブリッド暗号スイートを使用するための HTTP クライアントまたは AWS KMS クライアントの設定に関して質問したり、アイデアを共有するには、aws-kms-pq-tls-example リポジトリで課題を作成します。

サポートされる AWS リージョン

AWS KMS 用のポスト量子 TLS は、中国 (北京) と 中国 (寧夏) を除き、AWS KMS がサポートされているすべての AWS リージョンで使用できます。

> **Note**
>
> AWS KMS は、AWS GovCloud (US) の FIPS エンドポイントのハイブリッドポスト量子 TLS をサポートしていません。

AWS リージョン それぞれの AWS KMS エンドポイントのリストについては、「Amazon Web Services 全般のリファレンス」の「AWS Key Management Service エンドポイントとクォータ」を参照してください。FIPS エンドポイントの詳細については、「Amazon Web Services 全般のリファレンス」の「FIPS エンドポイント」を参照してください。

TLS におけるハイブリッドポスト量子キー交換について

AWS KMS は、ハイブリッドポスト量子キー交換暗号スイートをサポートします。Linux システムに AWS SDK for Java 2.x および AWS 共通ランタイムを使用して、これらの暗号スイートを使用する HTTP クライアントを設定できます。その後、HTTP クライアントを使用して AWS KMS エンドポイントに接続するたびに、ハイブリッド暗号スイートが使用されます。

この HTTP クライアントは、TLS プロトコルのオープンソース実装である s2n-tls を使用します。s2n-tls が使用するハイブリッド暗号スイートは、直接データ暗号化ではなく、キー交換専用に実装されています。キー交換中、クライアントとサーバーは、転送中にデータの暗号化と復号に使用するキーを計算します。

s2n-tls が使用しているアルゴリズムは、Elliptic Curve Diffie-Hellman (ECDH)、TLS で現在使用されている従来の鍵交換アルゴリズムを Kyber を組み合わせたハイブリッドで、アメリカ国立標準技術研究所 (NIST) が最初の標準ポスト量子キー合意アルゴリズとして指定した公開鍵暗号化および

キー確立アルゴリズムです。このメカニズムは、各アルゴリズムを独立して使用してキーを生成します。次に、2つのキーを暗号的に組み合わせます。s2n-tls では、プリファレンスリストで ECDH と Kyber を最初に配置するポスト量子 TLS を優先して、HTTP クライアントを設定できます。互換性を確保するために、従来のキー交換アルゴリズムがプリファレンスリストに含まれていますが、プリファレンスの順序では低くなっています。

進行中の研究で、Kyber アルゴリズムには予想されるポスト量子強度が欠けていることが明らかになった場合でも、ハイブリッドキーは現在使用されている単独の ECDH キーと少なくとも同じ強度です。このプロセスが完了するまで、ポスト量子アルゴリズムを単独で使用するのではなく、ハイブリッドアルゴリズムを使用することをお勧めします。

AWS KMS でハイブリッドポスト量子 TLS を使用する

AWS KMS へのコールには、ハイブリッドポスト量子 TLS を使用できます。HTTP クライアントのテスト環境を設定するときは、次の点に注意してください。

転送時の暗号化

s2n-tls のハイブリッド暗号スイートは、転送中の暗号化のみに使用されます。これらはクライアントから AWS KMS エンドポイントへ移動中のデータを保護します。AWS KMS はこれらの暗号スイートを使用して AWS KMS keys のデータを暗号化しません。

代わりに、AWS KMS が KMS キーでデータを暗号化する際は、256 ビットキーによる対称暗号化と、既に量子耐性を備えている Advanced Encryption Standard in Galois Counter Mode (AES-GCM) アルゴリズムを使用します。理論上の将来、256 ビット AES-GCM キーで作成された暗号文に対する大規模な量子コンピューティング攻撃は、キーの効果的なセキュリティを 128 ビットに低下させます。このセキュリティレベルは、AWS KMS 暗号文に対するブルートフォース攻撃を実行不可能にするのに十分です。

サポートされているシステム

s2n-tls のハイブリッド暗号スイートの使用は、現在 Linux システムでのみサポートされています。加えて、これらの暗号スイートは、AWS SDK for Java 2.x などの AWS 共通ランタイムをサポートする SDK でのみサポートされます。例については「ハイブリッドポスト量子 TLS の設定方法」を参照してください。

AWS KMS エンドポイント

ハイブリッド暗号スイートを使用する場合は、標準 AWS KMS エンドポイントを使用します。s2n-tls のハイブリッド暗号スイートは、AWS KMS の FIPS 140-2 検証済みエンドポイントと互換性がありません。

s2n-tls でポスト量子 TLS 接続を優先して HTTP クライアントを設定すると、ポスト量子暗号は暗号プリファレンスリストの先頭になります。ただし、互換性を保つため、プリファレンスリストには優先順で低い従来の非ハイブリッド暗号が含まれています。AWS KMS FIPS 140-2 検証済みエンドポイントを使用してポスト量子 TLS を優先するように HTTP クライアントを設定すると、s2n-tls は従来の非ハイブリッドキー交換暗号をネゴシエートします。

AWS リージョン それぞれの AWS KMS エンドポイントのリストについては、「Amazon Web Services 全般のリファレンス」の「AWS Key Management Service エンドポイントとクォータ」を参照してください。FIPS エンドポイントの詳細については、「Amazon Web Services 全般のリファレンス」の「FIPS エンドポイント」を参照してください。

期待されるパフォーマンス

初期のベンチマークテストでは、s2n-tls のハイブリッド暗号スイートは従来の TLS 暗号スイートよりも遅いことが示されています。この効果は、ネットワークプロファイル、CPU 速度、コア数、コールレートによって異なります。パフォーマンステストの結果については、「How to tune TLS for hybrid post-quantum cryptography with Kyber」(Kyber でハイブリッドポスト量子暗号用に TLS を調整する方法) を参照してください。

ハイブリッドポスト量子 TLS の設定方法

この手順では、AWS 共通ランタイム HTTP クライアントの Maven 依存関係を追加します。次に、ポスト量子 TLS を優先する HTTP クライアントを設定します。次に、HTTP クライアントを使用する AWS KMS クライアントを作成します。

AWS KMS でのハイブリッドポスト量子 TLS の設定と使用の完全な実例については、aws-kms-pq-tls-example リポジトリを参照してください。

> **Note**
>
> プレビューとして公開された AWS 共通ランタイム HTTP クライアントは、2023 年 2 月に一般公開されました。このリリースでは、`tlsCipherPreference` クラスと `tlsCipherPreference()` メソッドパラメータが、`postQuantumTlsEnabled()` メソッドパラメータに置き換えられました。プレビュー中にこの例を使用していた場合は、コードを更新する必要があります。

1. Maven 依存関係に AWS 共通ランタイムクライアントを追加します。利用可能な最新バージョンを使用することをお勧めします。

例えば、このステートメントは Maven の依存関係に、AWS 共通ランタイムクライアントのバージョン 2.20.0 を追加します。

```
<dependency>
    <groupId>software.amazon.awssdk</groupId>
    <artifactId>aws-crt-client</artifactId>
    <version>2.20.0</version>
</dependency>
```

2. ハイブリッドポスト量子暗号スイートを有効にするには、AWS SDK for Java 2.x をプロジェクトに追加して初期化します。続いて、次の例に示すように、HTTP クライアントでハイブリッドポスト量子暗号スイートを有効にします。

   このコードでは、postQuantumTlsEnabled() メソッドパラメータを使用して、推奨されるハイブリッドポスト量子暗号スイートである ECDH with Kyber を優先する AWS 共通ランタイム HTTP クライアントを設定します。次に、設定済み HTTP クライアントを使用して、AWS KMS 非同期クライアントのインスタンスである KmsAsyncClient を構築します。このコードが完了すると、KmsAsyncClient インスタンス上のすべての AWS KMS API リクエストは、ハイブリッドポスト量子 TLS を使用します。

   ```
   // Configure HTTP client
   SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
              .postQuantumTlsEnabled(true)
              .build();

   // Create the AWS KMS async client
   KmsAsyncClient kmsAsync = KmsAsyncClient.builder()
             .httpClient(awsCrtHttpClient)
             .build();
   ```

3. ハイブリッドポスト量子 TLS を使用して AWS KMS の呼び出しをテストします。

   構成された AWS KMS クライアントで AWS KMS API オペレーションを呼び出すと、コールはハイブリッドポスト量子 TLS を使用して AWS KMS エンドポイントに送信されます。設定をテストするには、ListKeys などの AWS KMS API を呼び出します。

   ```
   ListKeysReponse keys = kmsAsync.listKeys().get();
   ```

AWS KMS でハイブリッドポスト量子 TLS をテストする

AWS KMS を呼び出すアプリケーションで、ハイブリッド暗号スイートを使用して次のテストを実行することを検討してください。

- 負荷テストとベンチマークを実行します。ハイブリッド暗号スイートの動作は、従来のキー交換アルゴリズムとは異なります。ハンドシェイク時間が長くなるように、接続のタイムアウトを調整する必要がある場合があります。AWS Lambda 関数内で実行している場合は、実行タイムアウト設定を拡張します。
- 別の場所からの接続を試します。要求が通過するネットワークパスによっては、ディープパケットインスペクション (DPI) を持つ中間ホスト、プロキシ、またはファイアウォールが要求をブロックしていることが検出されることがあります。これは、TLS ハンドシェイクの ClientHello部分で新しい暗号スイートを使用した場合や、キー交換メッセージが大きい場合に発生する可能性があります。これらの問題を解決できない場合は、セキュリティチームまたは IT 管理者と協力して、関連する構成を更新し、新しい TLS 暗号スイートのブロックを解除してください。

AWS KMS でのポスト量子 TLS について

AWS KMS でのハイブリッドポスト量子 TLS の使用の詳細については、次のリソースを参照してください。

- ブログ記事や研究論文へのリンクなど、AWS でのポスト量子暗号の詳細については、ポスト量子暗号化を参照してください。
- s2n-tls の詳細については、「新しいオープンソース TLS 実装である s2n-tls の導入」と「s2n-tls の使用」を参照してください。
- AWS 共通ランタイム HTTP クライアントについては、「AWS SDK for Java 2.x デベロッパーガイド」の「AWS CRT ベースの HTTP クライアントの設定」」を参照してください。
- 米国国立標準技術研究所 (NIST) のポスト量子暗号プロジェクトの詳細については、「Post-Quantum Cryptography」(ポスト量子暗号化) を参照してください。
- NIST ポスト量子暗号標準化については、「Post-Quantum Cryptography Standardization」(ポスト量子暗号標準化) を参照してください。

AWS KMS keys へのアクセスを特定する

現在、AWS KMS key へアクセスできるユーザーやアプリケーションの範囲を明らかにするには、KMS キーのキーポリシーと KMS キーに適用されるすべての権限を確認する必要があります。

また、すべての AWS Identity and Access Management (IAM) ポリシーの確認が必要な可能性もあります。これらを確認することで、潜在的な KMS キーの使用範囲を明らかにしたり、コンプライアンスや監査の要件を満たしたりできます。次のトピックは、現在 KMS キーにアクセスできる AWS プリンシパル (アイデンティティ) の完全なリストを作成するのに役立ちます。

トピック

キーポリシーを確認する

キーポリシーは、KMS キーへのアクセスを制御するための主要な方法です。すべての KMS キーには、厳密に 1 つのキーポリシーが必要です。

キーポリシーがデフォルトのキーポリシーで設定されるか、含まれている場合、キーポリシーによって、アカウントの IAM 管理者が IAM ポリシーを使用し、KMS キーへのアクセスを制御できるようになります。また、キーポリシーによって KMS キーを使用するアクセス許可が別の AWS アカウント に付与されている場合、外部アカウントの IAM 管理者は、IAM ポリシーを使用してこれらのアクセス許可を委任できます。KMS キーにアクセスできるプリンシパルの完全なリストを確認するには、IAM ポリシーを調べます。

アカウントAWS マネージドキー内の AWS KMSカスタマーマネージドキー または のキーポリシーを表示するには、 AWS KMS API で AWS Management Consoleまたは GetKeyPolicyオペレーションを使用します。キーポリシーを表示するには、KMS キーの `kms:GetKeyPolicy` アクセス許可が必要です。KMS キーのキーポリシーを表示する手順については、the section called “キーポリシーの表示” を参照してください。

キーポリシードキュメントを確認し、各ポリシーステートメントの `Principal` 要素で指定されているすべてのプリンシパルを書き留めます。`Allow` 効果を持つポリシーステートメントでは、`Principal` 要素内の IAM ユーザー、IAM ロール、AWS アカウント は、この KMS キーにアクセスできます。

> **Note**
>
> 条件を使用してキーポリシーを制限しない限り、アクセス許可を付与するキーポリシーステートメントで、プリンシパルをアスタリスク (*) に設定しないでください。アスタリスク

は、別のポリシーステートメントが明示的に拒否しない限り、すべての AWS アカウント のすべてのアイデンティティに、KMS キーを使用するアクセス許可を付与します。他の AWS アカウント のユーザーは、各自のアカウントに対応するアクセス権限があるときにはいつでも KMS キーを使用できます。

次の例では、デフォルトのキーポリシーで見つかったポリシーステートメントを使用してこれを行う方法を示します。

Example ポリシーステートメント 1

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
  "Resource": "*"
}
```

ポリシーステートメント 1 では、arn:aws:iam::111122223333:root は、AWS アカウント 111122223333 を参照する AWS プリンシパルアカウントです (アカウントのルートユーザーではありません)。デフォルトでは、このようなポリシーステートメントは、AWS Management Console で新しい KMS キーを作成するときや、新しい KMS キーをプログラムで作成してもキーポリシーを提供しないときに、キーポリシードキュメントに含まれます。

AWS アカウント へのアクセスを許可するステートメントを持つキーポリシードキュメントでは、アカウントの IAM ポリシーで KMS キーへのアクセスが有効化されます。つまり、アカウントのユーザーとロールは、キーポリシードキュメントにプリンシパルとして明示的にリストされていない場合でも、KMS キーにアクセスできる可能性があります。プリンシパルとしてリストされているすべての AWS アカウント のすべての IAM ポリシーを調べて、この KMS キーへのアクセスを許可しているかどうかを判断します。

Example ポリシーステートメント 2

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/KMSKeyAdmins"},
  "Action": [
```

```
    "kms:Describe*",
    "kms:Put*",
    "kms:Create*",
    "kms:Update*",
    "kms:Enable*",
    "kms:Revoke*",
    "kms:List*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

ポリシーステートメント 2 では、 は 111122223333 の KMS という名前の IAM AWS アカウント ロールarn:aws:iam::111122223333:role/KMSKeyAdminsを参照KeyAdmins します。このロールを引き受ける権限を持つユーザーは、KMS キーを管理するための管理アクションである、ポリシーステートメントにリストされたアクションを実行できます。

Example ポリシーステートメント 3

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

ポリシーステートメント 3 では、 は 111122223333 の という名前の IAM AWS アカウント ロールarn:aws:iam::111122223333:role/EncryptionAppを参照 EncryptionApp します。このロールを引き受ける権限を持つプリンシパルは、対称暗号化 KMS キーの暗号化オペレーションを含む、ポリシーステートメントにリストされたアクションを実行できます。

Example ポリシーステートメント 4

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:ListGrants",
    "kms:CreateGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

ポリシーステートメント 4 では、 は 111122223333 の という名前の IAM AWS アカウント ロールarn:aws:iam::111122223333:role/EncryptionAppを参照 EncryptionApp します。このロールを引き受ける権限を持つプリンシパルは、ポリシーステートメントにリストされたアクションを実行できます。これらのアクションは、ポリシーステートメント 3 の例で許可されたアクションと連動する際に、KMS キーの使用を、AWS KMS と統合されたほとんどの AWS のサービス (特に、権限を使用するサービス) に委任するために必要なアクションになります。Condition 要素の kms:GrantIsForAWSResource 値により、受任者が と統合AWS KMSされ、承認に許可を使用するAWSサービスである場合にのみ、委任が許可されます。

キーポリシードキュメントでプリンシパルを指定する各種方法をすべて確認するには、IAM ユーザーガイドのプリンシパルの指定を参照してください。

AWS KMS キーポリシーの詳細については、「AWS KMS のキーポリシー」を参照してください。

IAM ポリシーの確認

キーポリシーと許可に加え、IAM ポリシーを使用して KMS キーへのアクセスを許可することもできます。IAM ポリシーとキーポリシーがどのように連携するかについては、「キーアクセスのトラブルシューティング」を参照してください。

IAM ポリシーを使用して KMS キーに現在アクセスできるプリンシパルを特定するには、ブラウザベースの IAM Policy Simulator ツールを使用するか、IAM API にリクエストします。

IAM ポリシーを調べる方法

- IAM ポリシーシミュレーターを使用した IAM ポリシーの確認

- IAM API を使用した IAM ポリシーの確認

IAM ポリシーシミュレーターを使用した IAM ポリシーの確認

IAM Policy Simulator は、IAM ポリシーを介して KMS キーにアクセスできるプリンシパルを学習するのに役立ちます。

IAM Policy Simulator を使用して KMS キーへのアクセスを特定するには

1. AWS Management Console にサインインし、https://policysim.aws.amazon.com/ で IAM Policy Simulator を開きます。
2. [Users, Groups, and Roles] ペインで、ポリシーをシミュレートするユーザー、グループ、またはロールを選択します。
3. (オプション) シミュレーションから除外するポリシーの横のチェックボックスをオフにします。すべてのポリシーをシミュレートする場合は、すべてのポリシーが選択された状態にします。
4. [Policy Simulator] ペインで、以下のオペレーションを行います。
   a. [Select service] で、[Key Management Service] を選択します。
   b. 特定の AWS KMS アクションをシミュレートするには、[Select actions] でシミュレートするアクションを選択します。すべての AWS KMS アクションをシミュレートする場合は、[Select All] を選択します。
5. (オプション) Policy Simulator が、デフォルトですべての KMS キーへのアクセスをシミュレートします。　特定の KMS キーへのアクセスをシミュレートするには、[Simulation Settings] (シミュレーション設定) を選択し、シミュレートする KMS キーの Amazon リソースネーム (ARN) を入力します。
6. [Run Simulation (シミュレーションの実行)] を選択します。

シミュレーションの結果は、[Results] セクションに表示されます。AWS アカウント のすべてのユーザー、グループ、ロールについて、ステップ 2 ~ 6 を繰り返します。

IAM API を使用した IAM ポリシーの確認

IAM API を使用して、IAM ポリシーをプログラムで調べることができます。次のステップは、API でユーザーベースのポリシーを確認する方法の概要を示します。

1. キーポリシーでプリンシパルとしてAWS アカウントリストされている各 (つまり、次の形式で指定された各AWSアカウントプリンシパル"Principal": {"AWS":

"arn:aws:iam::111122223333:root"}) について、IAM API の ListUsersおよび ListRolesオペレーションを使用して、アカウントのすべてのユーザーとロールを取得します。

2. リスト内のユーザーとロールごとに、IAM API の SimulatePrincipalPolicyオペレーションを使用して、以下のパラメータを渡します。
   - PolicySourceArn について、リストのユーザーやロールから Amazon リソースネーム (ARN) を指定します。PolicySourceArn は、各 SimulatePrincipalPolicy リクエストで 1 つしか指定できないため、このオペレーションは複数回 (リスト内のユーザーおよびロールごとに 1 回) 呼び出す必要があります。
   - ActionNames のリストについて、シミュレートするすべての AWS KMS API アクションを指定します。すべての AWS KMS API アクションをシミュレートするには、kms:* を使用します。AWS KMS API アクションを個別にテストするには、各 API アクションを「kms:", for example "kms:ListKeys」で実行します。AWS KMS API アクションの完全なリストについては、「AWS Key Management Service API リファレンス」の「Actions」(アクション) を参照してください。
   - (オプション) ユーザーやロールが特定の KMS キーにアクセスできるかどうかを特定するために、ResourceArns パラメータを使用して KMS キーの Amazon リソースネーム (ARN) のリストを指定します。ユーザーまたはロールが KMS キーにアクセスできるかどうかを特定するために、ResourceArns パラメータを省略してください。

IAM は各 SimulatePrincipalPolicy リクエストに対して、、 allowed, explicitDeny, 、またはの評価決定で応答 implicitDenyします。allowed の評価決定を含む各応答には、許可された特定の AWS KMS API オペレーションの名前が含まれます。評価で使用した KMS キーの ARN がある場合は、これも含まれます。

許可の確認

権限は、ユーザー、または AWS KMS と統合された AWS サービスが、KMS キーをいつどのように使用するかを指定できるアクセス許可を指定する、アドバンストメカニズムです。権限は KMS キーにアタッチされ、各権限には KMS キーを使用するためのアクセス許可を受け取るプリンシパルと、許可されるオペレーションのリストが含まれます。許可は、キーポリシーに代わる手段であり、特定のユースケースで役立ちます。詳細については、「AWS KMS でのグラント」を参照してください。

KMS キーの許可のリストを取得するには、 AWS KMSListGrantsオペレーションを使用します。KMS キーの権限を確認することで、これらの権限で KMS キーを使用するアクセス許可を現在持っているユーザーやアプリケーションを特定できます。例えば、AWS CLI の list-grants コマンドから取得した権限の JSON 表現を次に示します。

```
{"Grants": [{
  "Operations": ["Decrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "0d8aa621-43ef-4657-b29c-3752c41dc132",
  "RetiringPrincipal": "arn:aws:iam::123456789012:root",
  "GranteePrincipal": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-5d476fab",
  "GrantId": "dc716f53c93acacf291b1540de3e5a232b76256c83b2ecb22cdefa26576a2d3e",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151834E9,
  "Constraints": {"EncryptionContextSubset": {"aws:ebs:id": "vol-5cccfb4e"}}
}]}
```

KMS キーにアクセスできるユーザーやアプリケーションを見つけるには、"GranteePrincipal" 要素を確認します。前述の例では、被付与者のプリンシパルは、EC2 インスタンス i-5d476fab に関連して割り当てられたロールユーザーです。EC2 インフラストラクチャはこのロールを使用して、暗号化された EBS ボリューム vol-5cccfb4e をインスタンスにアタッチします。この場合、EC2 インフラストラクチャロールは、KMS キーを使用するアクセス許可を持っています。これは、この KMS キーで保護された、暗号化された EBS ボリュームを以前に作成したためです。その後、ボリュームを EC2 インスタンスにアタッチしました。

以下は、AWS CLI の list-grants コマンドから取得した権限の JSON 表現の別の例です。次の例では、被付与者プリンシパルは別の AWS アカウント アカウントです。

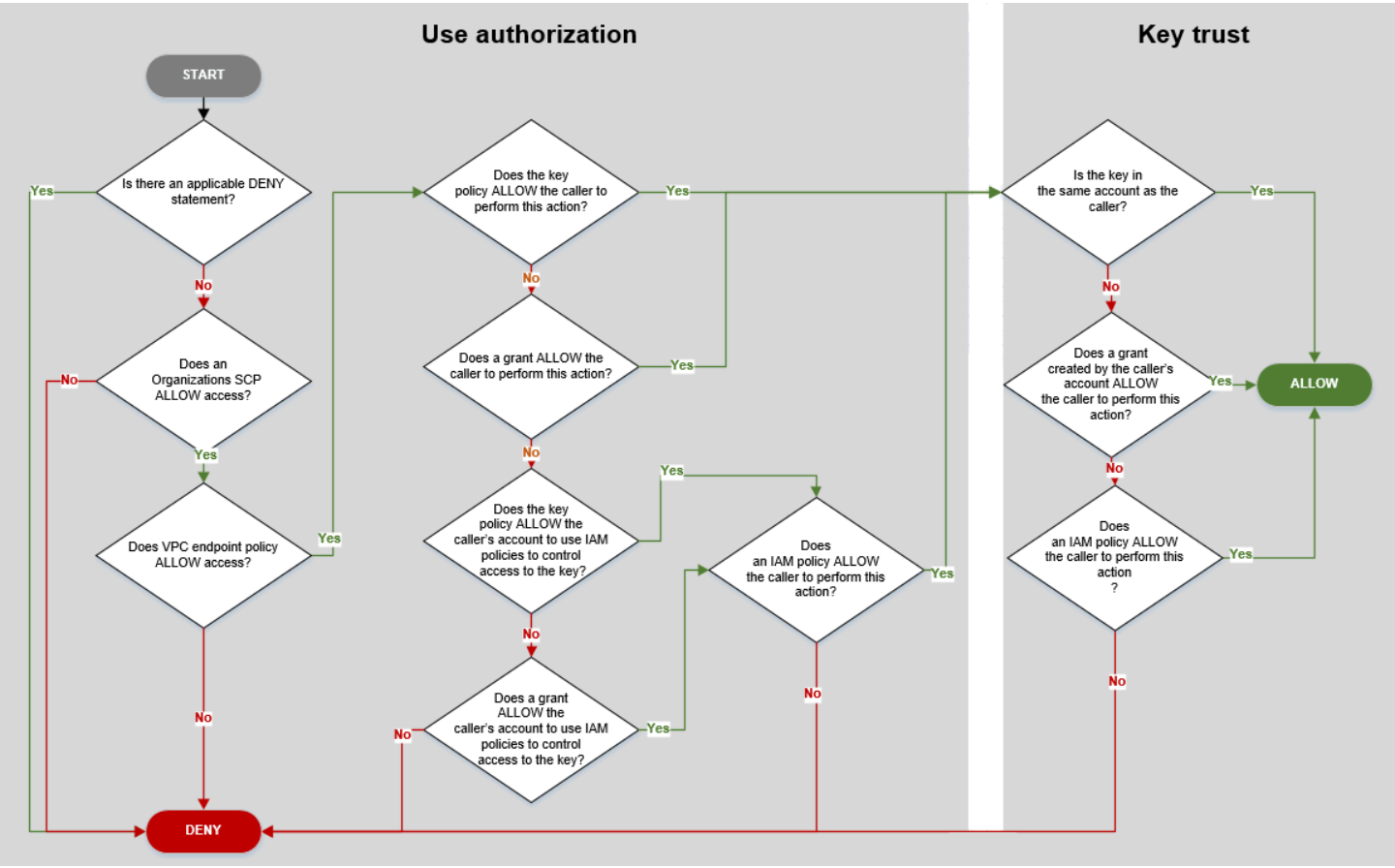
```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

キーアクセスのトラブルシューティング

KMS キーへのアクセスを許可する際、AWS KMS は以下を評価します。

- KMS キーにアタッチされているキーポリシー。キーポリシーは常に、AWS アカウント および KMS キーを所有するリージョンで定義されます。
- リクエストを行うユーザーまたはロールにアタッチされているすべての IAM ポリシー。プリンシパルの KMS キーの使用を管理する IAM ポリシーは、常にプリンシパルの AWS アカウント で定義されます。
- KMS キーに適用されるすべての権限。
- AWS Organizations のサービスコントロールポリシーや VPC エンドポイントポリシーなど、KMS キーを使用するリクエストに適用できるその他の種類のポリシー。これらのポリシーはオプションであり、デフォルトですべてのアクションを許可しますが、プリンシパルに付与される権限を制限するために使用できます。

AWS KMS は、これらのポリシーメカニズムも合わせて評価し、KMS キーへのアクセスの許可または拒否を決定します。そのために、AWS KMS は以下のフローチャートに示されているようなプロセスを使用します。以下のフローチャートは、ポリシーの評価プロセスを視覚的に表したものです。



このフローチャートは 2 つの部分に分かれています。これらの部分には順序があるように見えますが、一般的に同時に評価されます。

- 認可の使用は、キーポリシー、IAM ポリシー、権限、およびその他の適用可能なポリシーに基づいて、KMS キーの使用を許可するかどうかを判断します。
- キーの信頼は、使用が許可されている KMS キーを信頼すべきかどうかを判断します。通常、ユーザーは自分の AWS アカウント のリソースを信頼します。ただし、アカウントの権限または IAM ポリシーで KMS キーの使用が許可される場合は、別の AWS アカウント で KMS キーの使用を信頼できます。

このフローチャートを使用すると、発信者が KMS キーの使用許可を許可または拒否された理由がわかります。また、ポリシーと許可を評価することもできます。例えば、フローチャートは、キーポリシー、IAM ポリシー、または許可で、明示的な DENY ステートメントまたは明示的な ALLOW ステートメントがないことによって、呼び出し元がアクセスを拒否できることを示しています。

フローチャートでは、いくつかの一般的なアクセス許可のシナリオについて説明しています。

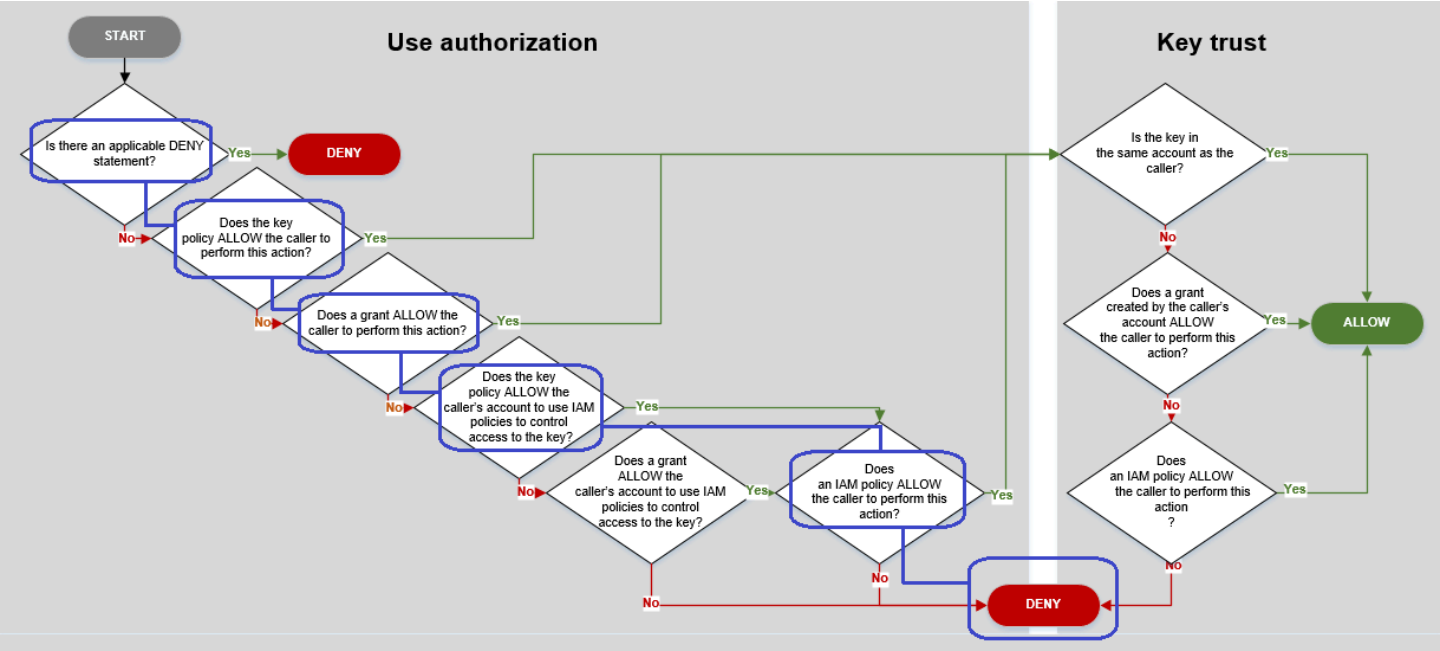
アクセス許可の例

- 例 1: ユーザーが自分の AWS アカウント で KMS キーへのアクセスを拒否された
- 例 2: ユーザーが別の AWS アカウント の KMS キーを使用するためのアクセス許可を持つロールを引き受ける

例 1: ユーザーが自分の AWS アカウント で KMS キーへのアクセスを拒否された

Alice は、111122223333 AWS アカウント の IAM ユーザーです。Alice は、同じ AWS アカウントの KMS キーへのアクセスを拒否されました。Alice が KMS キーを使用できないのはなぜでしょうか。

この場合、Alice が KMS キーへのアクセスを拒否されたのは、必要なアクセス許可を付与するキーポリシー、IAM ポリシー、権限がないためです。KMS キーのキーポリシーでは、AWS アカウントが IAM ポリシーを使用して KMS キーへのアクセスを制御することができますが、KMS キーを使用する許可を Alice に付与する IAM ポリシーはありません。



この例に関連するポリシーを考えてみます。

- Alice が使用する KMS キーには、デフォルトキーポリシーがあります。このポリシーは、KMS キーを所有する AWS アカウント に、IAM ポリシーを使用して KMS キーへのアクセスを制御することを許可します。このキーポリシーは、フローチャートのキーポリシーが、発信者のアカウントが IAM ポリシーを使用してキーへのアクセスをコントロールすることを許可しているか?という条件を満たしています。

```
{
  "Version" : "2012-10-17",
  "Id" : "key-test-1",
  "Statement" : [ {
    "Sid" : "Delegate to IAM policies",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

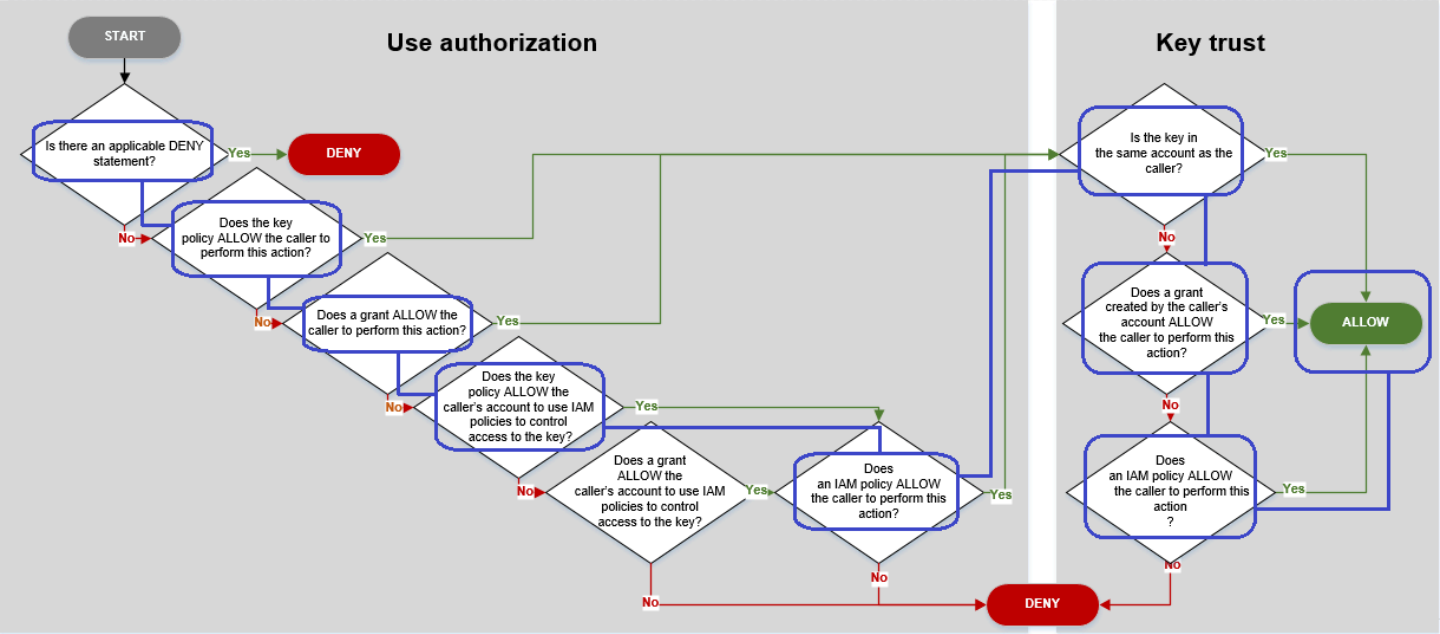
- ただし、Alice に KMS キーの使用許可を付与するキーポリシー、IAM ポリシー、権限はありません。このため、Alice は KMS キーの使用許可を拒否されます。

例 2: ユーザーが別の AWS アカウント の KMS キーを使用するためのアクセス許可を持つロールを引き受ける

Bob はアカウント 1 (111122223333) のユーザーです。彼は、暗号化オペレーションでアカウント 2 (444455556666) の KMS キーの使用を許可されています。これはどのようにすれば可能になるのでしょうか。

> **Tip**
>
> クロスアカウントのアクセス許可を評価するときは、キーポリシーが KMS キーのアカウントで指定されていることに注意してください。IAM ポリシーは、発信者が別のアカウントにいる場合でも、発信者のアカウントで指定されます。KMS キーへのクロスアカウントアクセス提供の詳細については、他のアカウントのユーザーに KMS キーの使用を許可する を参照してください。

- アカウント 2 の KMS キーのキーポリシーにより、アカウント 2 は IAM ポリシーを使用して KMS キーへのアクセスを制御できます。
- アカウント 2 の KMS キーのキーポリシーは、アカウント 1 が暗号化オペレーションで KMS キーを使用することを許可します。ただし、アカウント 1 は IAM ポリシーを使用して、プリンシパルに KMS キーへのアクセスを許可する必要があります。
- アカウント 1 の IAM ポリシーでは、`Engineering` ロールがアカウント 2 の KMS キーを暗号化オペレーションに使用することを許可します。
- アカウント 1 のユーザーである Bob には、`Engineering` ロールを引き受けるアクセス権限があります。
- Bob はこの KMS キーを信頼できます。この KMS キーは Bob のアカウントではありませんが、アカウントの IAM ポリシーにより、この KMS キーを使用する明示的なアクセス許可が Bob に付与されるためです。



アカウント 1 のユーザーである Bob が、アカウント 2 の KMS キーを使用することを許可するポリシーを考えてみます。

- KMS キーのキーポリシーにより、アカウント 2 (444455556666、KMS キーを所有するアカウント) は IAM ポリシーを使用して、KMS キーへのアクセスを制御できます。このキーポリシーでは、アカウント 1 (111122223333) に、KMS キーを暗号化オペレーション (ポリシーステートメントの `Action` 要素で指定) で使用することも許可します。ただし、プリンシパルに KMS キーへのアクセスを許可する IAM ポリシーがアカウント 1 で定義されるまでは、アカウント 1 のユーザーはアカウント 2 の KMS キーを使用できません。

  フローチャートでは、アカウント 2 のこのキーポリシーは、キーポリシーは、呼び出し元のアカウントに IAM ポリシーを使用してキーへのアクセスを制御することを許可しますか? という条件を満たしています。

```
{
    "Id": "key-policy-acct-2",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Permission to use IAM policies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Action": "kms:*",
```

```
            "Resource": "*"
        },
        {
            "Sid": "Allow account 1 to use this KMS key",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- 発信者の AWS アカウント (アカウント 1、111122223333) の IAM ポリシーでは、アカウント 2 (444455556666) の KMS キーを使用して、暗号化オペレーションを実行するためのアクセス許可をプリンシパルに付与します。Action 要素は、アカウント 2 のキーポリシーがアカウント 1 に付与したのと同じアクセス許可をプリンシパルに委任します。これらのアクセス許可をアカウント 1 の Engineering のロールに付与するために、[このインラインポリシーは Engineering のロールに埋め込まれています]。

  このようなクロスアカウント IAM ポリシーは、アカウント 2 の KMS キーのキーポリシーが、KMS キーの使用許可をアカウント 1 に付与している場合にのみ有効です。また、アカウント 1 がプリンシパルに付与できるのは、キーポリシーがそのアカウントに付与しているアクションの実行アクセス許可のみです。

  フローチャートでは、これは IAM ポリシーが呼び出し元にこのアクションを実行することを許可していますか? という条件を満たしています。

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

```
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": [
                "arn:aws:kms:us-
west-2:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            ]
        }
    ]
}
```

- 最後に必要な要素は、アカウント 1 での Engineering ロールの定義です。ロールの AssumeRolePolicyDocument により、Bob は Engineering ロールを引き受けることができます。

```
{
    "Role": {
        "Arn": "arn:aws:iam::111122223333:role/Engineering",
        "CreateDate": "2019-05-16T00:09:25Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": {
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:user/bob"
                },
                "Effect": "Allow",
                "Action": "sts:AssumeRole"
            }
        },
        "Path": "/",
        "RoleName": "Engineering",
        "RoleId": "AROA4KJY2TU23Y7NK62MV"
    }
}
```

AWS KMS アクセス許可

この表は、 AWS KMS リソースへのアクセスを制御するためのアクセス AWS KMS 許可を理解するのに役立つように設計されています。列見出しの定義は、表の下に表示されます。

アクセス AWS KMS 許可については、「サービス認証リファレンス」の「 のアクション、リソース、および条件キー AWS Key Management Service」トピックでも説明されています。ただし、このトピックには、各許可の絞り込みに使用できる条件キーがすべて一覧表示されているわけではありません。

> **Note**
>
> テーブル内のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
CancelKeyDeletion kms:CancelKeyDeletion	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー)

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
				kms:ViaService
ConnectCustomKeyStore kms:ConnectCustomKeyStore	IAM ポリシー	いいえ	*	kms:CallerAccount
CreateAlias kms:CreateAlias	IAM ポリシー (エイリアス用)	いいえ	エイリアス	なし (エイリアスへのアクセスを制御する場合)
このオペレーションを使用するには、発信者には 2 つのリソースでの kms:CreateAlias 許可が必要です。 • エイリアス (IAM ポリシーにおける) • KMS キー (キーポリシー内) 詳細については、「エイリアスへのアクセスの制御」を参照してください。	キーポリシー (KMS キー用)	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
CreateCustomKeyStore kms:CreateCustomKeyStore	IAM ポリシー	いいえ	*	kms:CallerAccount

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
CreateGrant kms:CreateGrant	キーポリシー	はい	KMS キー	暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys 権限条件: kms:GrantConstraintType kms:GranteePrincipal kms:GrantIsForAWSResource kms:GrantOperations kms:RetiringPrincipal KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
				aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
CreateKey kms:CreateKey	IAM ポリシー	いいえ	*	kms:BypassPolicyLockoutSafetyCheck kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ViaService aws:RequestTag/tag-key (AWS グローバル条件キー) aws:ResourceTag/tag-key (AWS グローバル条件キー) aws:TagKeys (AWS グローバル条件キー)

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
Decrypt kms:Decrypt	キーポリシー	はい	KMS キー	暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
DeleteAlias kms:DeleteAlias このオペレーションを使用するには、発信者には 2 つのリソースでの kms:DeleteAlias 許可が必要です。 • エイリアス (IAM ポリシーにおける) • KMS キー (キーポリシー内) 詳細については、「エイリアスへのアクセスの制御」を参照してください。	IAM ポリシー (エイリアス用) キーポリシー (KMS キー用)	いいえ いいえ	エイリアス KMS キー	なし (エイリアスへのアクセスを制御する場合) KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
DeleteCustomKeyStore kms:DeleteCustomKeyStore	IAM ポリシー	いいえ	*	kms:CallerAccount

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
DeleteImportedKeyMaterial kms:DeleteImportedKeyMaterial	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
DescribeCustomKeyStores kms:DescribeCustomKeyStores	IAM ポリシー	いいえ	*	kms:CallerAccount

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
DescribeKey kms:DescribeKey	キーポリシー	はい	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:RequestAlias

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
DisableKey kms:DisableKey	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
DisableKeyRotation kms:DisableKeyRotation	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
DisconnectCustomKeyStore kms:DisconnectCustomKeyStore	IAM ポリシー	いいえ	*	kms:CallerAccount

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
EnableKey kms:EnableKey	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
EnableKeyRotation kms:EnableKeyRotation	キーポリシー	いいえ	KMS キー (対称のみ)	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
暗号化 kms:Encrypt	キーポリシー	はい	KMS キー	暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GenerateDataKey kms:GenerateDataKey	キーポリシー	はい	KMS キー (対称のみ)	暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GenerateDataKeyPair `kms:GenerateDataKeyPair`	キーポリシー	はい	KMS キー (対称のみ) 対称暗号化 KMS キーによって保護される非対称データキーペアを生成します。	データキーペアの条件: kms:DataKeyPairSpec 暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
				aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GenerateDataKeyPairWithoutPlaintext kms:GenerateDataKeyPairWithoutPlaintext	キーポリシー	はい	KMS キー (対称のみ) 対称暗号化 KMS キーによって保護される非対称データキーペアを生成します。	データキーペアの条件: kms:DataKeyPairSpec 暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
				aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GenerateDataKeyWithoutPlaintext `kms:GenerateDataKeyWithoutPlaintext`	キーポリシー	はい	KMS キー (対称のみ)	暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GenerateMac kms:GenerateMac	キーポリシー	はい	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService 暗号化オペレーションの条件: kms:MacAlgorithm kms:RequestAlias
GenerateRandom kms:GenerateRandom	IAM ポリシー	該当なし	*	なし

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GetKeyPolicy kms:GetKeyPolicy	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GetKeyRotationStatus `kms:GetKeyRotationStatus`	キーポリシー	はい	KMS キー (対称のみ)	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GetParametersForImport kms:GetParametersForImport	キーポリシー	いいえ	KMS キー	kms:WrappingAlgorithm kms:WrappingKeySpec KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
GetPublicKey kms:GetPublicKey	キーポリシー	はい	KMS キー (非対称のみ)	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:RequestAlias

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ImportKeyMaterial kms:ImportKeyMaterial	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:ExpirationModel kms:ValidTo
ListAliases kms:ListAliases	IAM ポリシー	いいえ	*	なし

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ListGrants kms:ListGrants	キーポリシー	はい	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:GrantIsForAWSResource

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ListKeyPolicies kms:ListKeyPolicies	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
ListKeys kms:ListKeys	IAM ポリシー	いいえ	*	なし

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ListResourceTags kms:ListResourceTags	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
ListRetirableGrants kms:ListRetirableGrants	IAM ポリシー	指定されたプリンシパルがローカルアカウントにある必要があります。オペレーションはすべてのアカウントで権限を返します。	*	なし

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
PutKeyPolicy kms:PutKeyPolicy	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:BypassPolicyLockoutSafetyCheck

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ReEncrypt kms:ReEncryptFrom kms:ReEncryptTo このオペレーションを使用するには、発信者に 2 つの KMS キーでのアクセス許可が必要です。 • 復号に使用される KMS キーの kms:ReEncryptFrom • 暗号化に使用される KMS キーの kms:ReEncryptTo	キーポリシー	はい	KMS キー	暗号化オペレーションの条件 kms:EncryptionAlgorithm kms:RequestAlias 暗号化コンテキスト条件: kms:EncryptionContext：context-key kms:EncryptionContextKeys KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
				その他の条件: kms:ReEncryptOnSameKey
ReplicateKey kms:ReplicateKey このオペレーションを使用するには、発信者に次のアクセス許可が必要です。 • マルチリージョンプライマリキーの kms:ReplicateKey • レプリカリージョンの IAM ポリシーの kms:CreateKey	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:ReplicaRegion

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
RetireGrant kms:RetireGrant 権限を使用停止にするアクセス許可は、主に権限によって決定されます。ポリシーだけでは、このオペレーションへのアクセスを許可することはできません。詳細については、「グラントの使用停止と取り消し」を参照してください。	IAM ポリシー (このアクセス許可はキーポリシーでは無効です)。	はい	KMS キー	kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー)

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
RevokeGrant kms:RevokeGrant	キーポリシー	はい	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件: kms:GrantIsForAWSResource

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
ScheduleKeyDeletion kms:ScheduleKeyDeletion	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
Sign kms:Sign	キーポリシー	はい	KMS キー (非対称のみ)	署名および検証の条件: kms:MessageType kms:RequestAlias kms:SigningAlgorithm KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
TagResource kms:TagResource	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService タグ付けの条件: aws:RequestTag/tag-key (AWS グローバル条件キー) aws:TagKeys (AWS グローバル条件キー)

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
UntagResource kms:UntagResource	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService タグ付けの条件: aws:RequestTag/tag-key (AWS グローバル条件キー) aws:TagKeys (AWS グローバル条件キー)

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
UpdateAlias kms:UpdateAlias このオペレーションを使用するには、発信者には 3 つのリソースでの kms:UpdateAlias 許可が必要です。 • エイリアス • 現在関連付けられている KMS キー • 新しく関連付けられた KMS キー 詳細については、「エイリアスへのアクセスの制御」を参照してください。	IAM ポリシー (エイリアス用) キーポリシー (KMS キー用)	いいえ	エイリアス KMS キー	なし (エイリアスへのアクセスを制御する場合) KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService
UpdateCustomKeyStore kms:UpdateCustomKeyStore	IAM ポリシー	いいえ	*	kms:CallerAccount

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
UpdateKeyDescription `kms:UpdateKeyDescription`	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
UpdatePrimaryRegion kms:UpdatePrimaryRegion 発信者がこのオペレーションを使用するには、レプリカキーとなるマルチリージョンプライマリキーと、プライマリキーとなるマルチリージョンレプリカキーの両方に対する kms:UpdatePrimaryRegion のアクセス許可が必要です。	キーポリシー	いいえ	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService その他の条件 kms:PrimaryRegion

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
Verify kms:Verify	キーポリシー	はい	KMS キー (非対称のみ)	署名および検証の条件: kms:MessageType kms:RequestAlias kms:SigningAlgorithm KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService

アクションおよびアクセス許可	ポリシータイプ	クロスアカウントの使用	リソース (IAM ポリシー用)	AWS KMS 条件キー
VerifyMac kms:VerifyMac	キーポリシー	はい	KMS キー	KMS キーオペレーションの条件: kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key (AWS グローバル条件キー) kms:ViaService 暗号化オペレーションの条件: kms:MacAlgorithm kms:RequestAlias

列の説明

このテーブルの列には、以下の情報が表示されます:

- アクションとアクセス許可には、各 AWS KMS API オペレーションと、オペレーションを許可するアクセス許可が一覧表示されます。ポリシーステートメントの `Action` 要素でオペレーションを指定します。

- ポリシータイプは、アクセス許可がキーポリシーまたは IAM ポリシーで使用できるかどうかを表示します。

  キーポリシーは、キーポリシーでアクセス許可を指定できることを意味します。キーポリシーに IAM ポリシーを有効にするポリシーステートメントが含まれている場合には、IAM ポリシーで許可を指定できます。

  IAM ポリシーは、IAM ポリシーでのみアクセス許可を指定できることを意味します。
- クロスアカウント使用は、別の AWS アカウントで、認可されたユーザーが実行できるオペレーションを表示します。

  はいの値は、別の AWS アカウントで、プリンシパルがリソースに対してオペレーションを実行できることを意味します。

  いいえの値は、自分の AWS アカウントで、プリンシパルがリソースに対してのみオペレーションを実行できることを意味します。

  別のアカウントのプリンシパルにクロスアカウントリソースで使用できないアクセス許可を付与した場合、そのアクセス許可は無効になります。例えば、別のアカウントのプリンシパルに kms:TagResource アカウントの KMS キーへのアクセス許可を付与すると、アカウントの KMS キーにタグを付ける試みは失敗します。
- リソースには、アクセス許可が適用される AWS KMS リソースが一覧表示されます。 は、KMS キーとエイリアスの 2 つのリソースタイプ AWS KMS をサポートします。キーポリシーでは、Resource 要素の値は常に * であり、キーポリシーがアタッチされている KMS キーを示します。

  IAM ポリシーの AWS KMS リソースを表すには、次の値を使用します。

  KMS キー

  リソースが KMS キーの場合は、そのキー ARN を使用します。ヘルプについては、「the section called “キー ID とキー ARN を検索する”」を参照してください。

  `arn:AWS_partition_name:kms:AWS_Region:AWS_account_ID:key/key_ID`

  例:

  arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

エイリアス

リソースがエイリアスの場合は、そのエイリアス ARN を使用します。ヘルプについては、「the section called “エイリアス名とエイリアス ARN を見つける”」を参照してください。

arn:*AWS_partition_name*:kms:*AWS_region*:*AWS_account_ID*:alias/*alias_name*

例:

arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias

***** (アスタリスク)

アクセス許可が特定のリソース (KMS キーまたはエイリアス) に適用されない場合は、アスタリスク (*) を使用します。

アクセス AWS KMS 許可の IAM ポリシーでは、 Resource要素のアスタリスクはすべての AWS KMS リソース (KMS キーとエイリアス) を示します。アクセス AWS KMS 許可が特定の KMS キーまたはエイリアスに適用されない場合は、 Resource要素でアスタリスクを使用することもできます。例えば、kms:CreateKey 権限または kms:ListKeys 権限を許可または禁止する場合、Resource 要素を * に設定したり、アカウント固有のバリエーション (arn:*AWS_partition_name*:kms:*AWS_region*:*AWS_account_ID*:* など) に設定したりできます。

- AWS KMS 条件キーには、オペレーションへのアクセスを制御するために使用できる AWS KMS 条件キーが一覧表示されます。ポリシーの Condition 要素で条件を指定します。詳細については、「AWS KMS 条件キー」を参照してください。この列には、 でサポートされているAWS グローバル条件キーも含まれていますが AWS KMS、すべての AWS サービスでサポートされているわけではありません。

アクセス許可をテストする

AWS KMS を使用する場合は、AWS が API リクエストの認証に使用する認証情報が必要です。認証情報には、 KMS キーとエイリアスにアクセスするためのアクセス許可を含める必要があります。アクセス許可は、キーポリシー、IAM ポリシー、グラント、およびクロスアカウントアクセス制御によって決定されます。KMS キーへのアクセス制御に加えて、CloudHSM やカスタムキーストアへのアクセスを制御できます。

DryRun API パラメータを指定して、AWS KMS キーを使用するために必要なアクセス許可があることを確認できます。DryRun を使用して、AWS KMS API 呼び出しのリクエストパラメータが正しく指定されていることを確認することもできます。

トピック

DryRun パラメータとは

DryRun は、AWS KMS API 呼び出しが成功することを確認するために指定する API パラメータ (オプション) です。実際に AWS KMS を呼び出す前に、DryRun を使用して API 呼び出しをテストします。次のことを確認できます。

- AWS KMS キーを使用するために必要なアクセス許可があること。
- 呼び出しのパラメータが正しく指定されていること。

AWS KMS は 特定の API アクションでの DryRun パラメータの使用をサポートします。

- CreateGrant
- Decrypt
- 暗号化
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign
- 検証
- VerifyMac

DryRun パラメータを使用すると料金が発生し、標準の API リクエストとして課金されます。AWS KMS の料金の詳細については、「AWS Key Management Service の料金」を参照してください。

DryRun パラメータを使用するすべての API リクエストは API のリクエストクォータに適用され、API リクエストクォータを超えた場合にスロットリング例外が発生する可能性があります。例えば、Decrypt を呼び出す際に DryRun を使用する場合でも DryRun を使用しない場合でも、同じ暗号化オペレーションクォータに対してカウントされます。詳細については、「AWS KMS リクエストのスロットリング」を参照してください。

AWS KMS API オペレーションへのすべての呼び出しは、AWS CloudTrail ログにイベントとしてキャプチャおよび記録されます。DryRun パラメータを指定するオペレーションの出力が CloudTrail ログに表示されます。詳細については、「AWS KMS による AWS CloudTrail API コールのログ記録」を参照してください。

API DryRun を使用した の指定

DryRun を使用するには、–dry-run パラメータをサポートする AWS CLI コマンドと AWS KMS API 呼び出しでパラメータを指定します。実行すると、呼び出しが成功するかどうかが AWS KMS によって検証されます。 DryRun を使用する AWS KMS 呼び出しは常に失敗し、呼び出しが失敗した理由に関する情報を含むメッセージが返されます。メッセージには次の例外が含まれます。

- DryRunOperationException - DryRun が指定されていなければリクエストは成功します。
- ValidationException - 間違った API パラメータが指定されたためリクエストが失敗しました。
- AccessDeniedException - KMS リソースで指定された API アクションを実行するアクセス許可がありません。

例えば、次のコマンドは CreateGrantオペレーションを使用し、keyUserRoleロールを引き受ける権限を持つユーザーが、指定された対称 KMS キー で Decrypt オペレーションを呼び出すことを許可する権限を作成します。DryRun パラメータが指定されます。

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

特定用途のキー

AWS Key Management Service (AWS KMS) は、さまざまな用途に対応し、複数の異なるタイプのキーをサポートします。

AWS KMS key を作成するときは、デフォルトで対称暗号化用の KMS キーが作成されます。AWS KMS では、対称暗号化 KMS キーは、暗号化と復号に使用される 256 ビット AES-GCM キーを表します。ただし、中国リージョンでは SM4 暗号化を使用する 128 ビットの対称キーを表します。対称キーマテリアルが、暗号化されずに AWS KMS 外で使用されることは一切ありません。タスクが非対称暗号化キーまたは HMAC キーを明示的に要求する場合以外は、暗号化されずに AWS KMS 外で使用されることがない対称暗号化 KMS キーが良い選択肢になります。また、AWS KMS と統合された AWS のサービスでも、データの暗号化には対称暗号化 KMS キーのみが使用されます。これらのサービスは、非対称 KMS キーを使用する暗号化をサポートしません。

AWS KMS で対称暗号化 KMS キーを使用して、データの暗号化、復号、再暗号化、データキーとデータキーペアの生成、およびランダムバイト文字列の生成を行うことができます。対称暗号化 KMS キーに独自のキーマテリアルをインポートし、カスタムキーストアで対称暗号化 KMS キーを作成することができます。対称 KMS キーと非対称 KMS キーで実行できるオペレーションを比較した表については、「キータイプリファレンス」を参照してください。

また、AWS KMS は、次の専用 KMS キータイプもサポートします。

- パブリックキー暗号化用の非対称 RSA キー
- 署名および検証用の非対称 RSA キーおよび ECC キー
- 非対称 SM2 キー (中国リージョンのみ) パブリックキー暗号または署名と検証用
- Hash-based Message Authentication Code を生成して検証するための HMAC キー
- 異なる AWS リージョン で同じキーのコピーのように機能するマルチリージョンキー (対称および非対称)
- 指定されたインポートしたキーマテリアルを含むキー
- AWS CloudHSM クラスターか、AWS の外側にある外部キーマネージャーによりバックアップされた、カスタムキーストアにあるキー。

KMS キータイプの選択

AWS KMS は複数タイプの KMS キーをサポートしており、これには対称暗号化キー、対称 HMAC キー、非対称暗号化キー、および非対称署名キーがあります。

KMS キーに違いがあるのは、異なる暗号化キーマテリアルが含まれているからです。

- 対称暗号化 KMS キー: 単一の 256 ビット AES-GCM 暗号化キーを表します。ただし、中国リージョンでは 128 ビット SM4 暗号化キーを表します。対称キーマテリアルが、暗号化されずに AWS KMS 外で使用されることは一切ありません。対称暗号化 KMS キーを使用するには、AWS KMS を呼び出す必要があります。

  デフォルト KMS キーである対称暗号化キーは、ほとんどの用途に最適です。AWS のサービス内のデータを保護するために KMS キーを使用する必要がある場合は、別のタイプのキーを使用するように指示される場合を除き、対称暗号化キーを使用します。
- 非対称 KMS キー: 暗号化および復号、または検証に使用できますが、両方には使用できない、数学的に関連するパブリックキーとプライベートキーペアを表します。プライベートキーが暗号化されないまま AWS KMS から出ていくことはありません。AWS KMS API オペレーションを呼び出すことによって、AWS KMS 内でパブリックキーを使用することも、パブリックキーをダウンロードして AWS KMS の外部で使用することもできます。
- HMAC KMS キー (対称): Hash-based Message Authentication Code の生成と検証に使用される、さまざまな長さの対称キーを表します。HMAC KMS キーのキーマテリアルが、暗号化されずに AWS KMS 外で使用されることはありません。HMAC KMS キーを使用するには、AWS KMS を呼び出す必要があります。

作成する KMS キーのタイプは、KMS キーの使用計画、セキュリティ要件、認可要件に大きく依存します。KMS キーの作成時は、KMS キーの暗号化設定 (キー仕様やキーの使用方法を含む) は、KMS キーの作成時に設定され、変更できないことに注意してください。

以下のガイダンスを使用し、ユースケースに基づいて必要なタイプの KMS キーを決定します。

データの暗号化と復号

データの暗号化と復号を必要とするほとんどのユースケースには、対称 KMS キーを使用します。AWS KMS が使用する対称暗号化アルゴリズムは、高速かつ効率的で、データの機密性と信頼性を保証します。これは、暗号化コンテキストとして定義された追加認証データ (AAD) による認証された暗号化をサポートします。このタイプの KMS キーでは、暗号化されたデータの送信者と受信者の両方が AWS KMS を呼び出す有効な AWS 認証情報を持っている必要があります。

AWS KMS を呼び出しできないユーザーが AWS 外部で暗号化を必要とするユースケースの場合は、非対称 KMS キーが適しています。非対称 KMS キーのパブリックキーを配信して、これらのユーザーがデータを暗号化できるようにします。また、そのデータを復号する必要があるアプリケーションは、AWS KMS 内で非対称 KMS キーのプライベートキーを使用できます。

メッセージの署名および署名の検証

メッセージに署名して、署名を検証するには、非対称 KMS キーを使用する必要があります。RSA キーペアまたは楕円曲線 (ECC) キーペア、または SM2 キーペア (中国リージョンのみ) を表すキー仕様で KMS キーを使用できます。選択するキー仕様は、使用する署名アルゴリズムによって決まります。ECC キーペアがサポートする ECDSA 署名アルゴリズムは、RSA 署名アルゴリズムよりも推奨されます。ただし、AWS 外部で署名を検証するユーザーをサポートするには、特定のキー仕様と署名アルゴリズムを使用することが必要になる場合があります。

パブリックキー暗号化の実行

パブリックキーの暗号化を実行するには、RSA キー仕様、または SM2 キーペア仕様 (中国リージョンのみ) で、非対称 KMS キーを使用する必要があります。KMS キーペアのパブリックキーを使用して、AWS KMS でデータを暗号化するには、Encrypt オペレーションを使用します。パブリックキーをダウンロードして、AWS KMS の外部でデータを暗号化する必要がある当事者と共有することもできます。

非対称 KMS キーのパブリックキーをダウンロードする場合、AWS KMS の外部で使用できます。ただし、AWS KMS の KMS キーを保護するセキュリティ管理の対象ではなくなります。例えば、AWS KMS キーポリシーまたは許可を使用して、パブリックキーの使用を制御することはできません。または、AWS KMS がサポートする暗号化アルゴリズムを使用して、キーを暗号化および復号のみに使用するかどうかを制御することもできません。詳細については、「パブリックキーのダウンロードに関する特別な考慮事項」を参照してください。

AWS KMS の外部のパブリックキーで暗号化されたデータを復号するには、Decrypt オペレーションを呼び出します。`SIGN_VERIFY` のキーの用途で KMS キーからのパブリックキーでデータが暗号化された場合、`Decrypt` オペレーションは失敗します。また、AWS KMS が、選択されたキー仕様をサポートしないアルゴリズムを使用して暗号化されている場合にも失敗します。主な仕様とサポートされているアルゴリズムの詳細については、「非対称キーの仕様」を参照してください。

これらのエラーを回避するには、AWS KMS の外部でパブリックキーを使用するすべてのユーザーがキー設定を保存する必要があります。AWS KMS コンソールとGetPublicKeyレスポンスは、パブリックキーを共有するときに含める必要がある情報を提供します。

HMAC コードを生成して検証する

Hash-based Message Authentication Code を生成して検証するには、HMAC KMS キーを使用します。AWS KMS で HMAC キーを作成すると、AWS KMS がキーマテリアルを作成して保護し、キーに正しい MAC アルゴリズムが使用されることを確実にします。HMAC コードは、擬似乱数としての使用、および対称署名とトークン化のための特定のシナリオでの使用も可能です。

HMAC KMS キーは対称キーです。AWS KMS コンソールで HMAC KMS キーを作成するときは、`Symmetric` キータイプを選択します。

AWS のサービスと使用する

AWS KMS と統合されている AWS のサービスでの使用のために KMS キーを作成するには、そのサービスのドキュメントを参照してください。データを暗号化する AWS のサービスには対称暗号化 KMS キーが必要です。

これらの考慮事項に加えて、キー仕様が異なる KMS キーの暗号化オペレーションでは、料金とリクエストクォータも異なります。AWS KMS の料金については、AWS Key Management Service の料金を参照してください。リクエストクォータの詳細については、「クォータのリクエスト」を参照してください 。

キーの用途の選択

KMS キーのキーの用途は、KMS キーが暗号化と復号、署名と署名の検証、または HMAC タグの生成と検証のどれに使用されのるかを決定します。キーの用途は、各 KMS キーに 1 つしかありません。KMS キーを複数タイプのオペレーションに使用すると、すべてのオペレーションの成果が攻撃に対してより脆弱になります。

以下の表にあるように、対称暗号化 KMS キーは暗号化と復号のみに使用できます。HMAC KMS キーは、HMAC コードの生成と検証のみに使用できます。楕円曲線 (ECC) KMS キーは、署名と検証にのみ使用できます。キーの用途を決定する必要があるのは、RSA KMS キーのみです。

KMS キータイプの有効なキーの用途

KMS キータイプ	暗号化と復号化 ENCRYPT_DECRYPT	署名と検証 SIGN_VERIFY	HMAC を生成して検証する GENERATE_VERIFY_MAC
対称暗号化 KMS キー	✓	⊗	⊗
HMAC KMS キー (対称)	⊗	⊗	✓

KMS キータイプ	暗号化と復号化 ENCRYPT_DECRYPT	署名と検証 SIGN_VERIFY	HMAC を生成して検証する GENERATE_VERIFY_MAC
RSA キーペアを使用した非対称 KMS キー	✓	✓	⊗
ECC キーペアを使用した非対称 KMSキー	⊗	✓	⊗
SM2 キーペアを持つ非対称 KMS キー (中国リージョンのみ)	✓	✓	⊗

AWS KMS コンソールでは、まずキータイプ (対称か非対称) を選択してから、キーの用途を選択します。選択するキーのタイプによって、表示されるキーの用途が決まります。選択するキーの用途によって、表示されるキーの仕様 (存在する場合) が決まります。

AWS KMS コンソールでキーの用途を選択するには:

- 対称暗号化 KMS キー (デフォルト) の場合は、[Encrypt and decrypt] (暗号化および復号) を選択します。
- HMAC KMS キーの場合は、[Generate and verify MAC] (MAC の生成と検証) を選択します。
- 楕円曲線 (ECC) キーマテリアルを持つ非対称 KMS キーの場合は、[Sign and verify] (署名および検証) を選択します。
- RSA キーマテリアルを持つ非対称 KMS キーの場合は、[Encrypt and decrypt] (暗号化および復号) または [Sign and verify] (署名および検証) を選択します。
- SM2 キーマテリアルを持つ非対称 KMS キーの場合は、[Encrypt and decrypt] (暗号化および復号) または [Sign and verify] (署名および検証) を選択します。SM2 キー仕様は、中国リージョンでのみ利用可能です。

プリンシパルが特定のキー使用に対してのみ KMS キーを作成できるようにするには、kms:KeyUsage 条件キーを使用します。kms:KeyUsage 条件キーを使用して、プリンシパルがキーの用途に基づいて KMS キーの API オペレーションを呼び出せるようにすることもできます。例

えば、キーの用途が SIGN_VERIFY である場合にのみ、KMS キーを無効にするアクセス許可を許可できます。

キー仕様の選択

非対称 KMS キーまたは HMAC KMS キーを作成するときは、そのキーの仕様を選択します。キー仕様は、すべての AWS KMS key のプロパティであり、KMS キーの暗号化設定を表します。KMS キーの作成時に選択したキー仕様を変更することはできません。間違ったキー仕様を選択した場合は、KMS キーを削除し、新しいキー仕様を作成します。

> **Note**
>
> KMS キーのキー仕様は、「カスタマーマスターキー仕様」として知られていました。CreateKey オペレーションの `CustomerMasterKeySpec`パラメータは廃止されました。代わりに、`KeySpec` パラメータを使用します。`CreateKey` および DescribeKeyオペレーションのレスポンスには、同じ値を持つ `KeySpec`および `CustomerMasterKeySpec` メンバーが含まれます。

キーの仕様によって、KMS キーのタイプ (対称か非対称)、KMS キーのキーマテリアルのタイプ、および AWS KMS が KMS キーに対してサポートする暗号化アルゴリズム、署名アルゴリズム、またはメッセージ認証コード (MAC) アルゴリズムが決まります。選択するキー仕様は、通常、ユースケースと規制要件によって決まります。ただし、キー仕様の異なる KMS キーの暗号化オペレーションは料金が異なるため、クォータも異なる場合があります。料金の詳細については、「AWS Key Management Service の料金」を参照してください。リクエストクォータの詳細については、「クォータのリクエスト」を参照してください 。

アカウントのプリンシパルが KMS キーに使用できるキー仕様を決定するには、kms:KeySpec 条件キーを使用します。

AWS KMS は、KMS キーの次のキー仕様をサポートします。

対称暗号化キーの仕様 (デフォルト)

- SYMMETRIC_DEFAULT

HMAC キーの仕様

- HMAC_224
- HMAC_256

- HMAC_384
- HMAC_512

RSA キー仕様 (暗号化と復号、または署名と検証)

- RSA_2048
- RSA_3072
- RSA_4096

楕円曲線のキー仕様

- 非対称 NIST 推奨楕円曲線キーペア (署名と検証)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- その他の非対称楕円曲線キーペア (署名と検証)
  - ECC_SECG_P256K1 (secp256k1)、暗号化に一般的に使用されます。

SM2 キー仕様 (暗号化と復号、または署名と検証)

- SM2 (中国リージョンのみ)

AWS KMS の非対称キー

AWS KMS は、数学的に関連する RSA、楕円曲線 (ECC)、SM2 (中国リージョンのみ) のパブリックキーとプライベートキーのペアを表す非対称 KMS キーをサポートします。これらのキーペアは、中国 (北京) および中国 (寧夏) リージョンを除き、FIPS 140-2 暗号化モジュール検証プログラムで認定された AWS KMS ハードウェアセキュリティモジュールで生成されます。プライベートキーにより、AWS KMS HSM が暗号化されないままになることはありません。ディストリビューション用にパブリックキーをダウンロードして AWS の外部で使用できます。暗号化および復号用、または署名および検証用に非対称 KMS キーを作成できますが、両方には作成できません。

AWS アカウント で、非対称 KMS キーの作成と管理が可能です。キーポリシー、IAM ポリシー、およびキーへのアクセスを制御するグラントの設定、KMS キーの有効化と無効化、タグとエイリアスの作成、KMS キーの削除などが含まれます。AWS CloudTrail ログで AWS 内の非対称 KMS キーを使用または管理する、すべてのオペレーションを監査できます。

AWS KMS は、AWS KMS 外部のクライアント側で暗号化に使用するように設計された非対称データキーペアも提供しています。非対称データキーペアのプライベートキーは、AWS KMS の対称暗号化 KMS キーによって保護されます。

このトピックでは、非対称 KMS キーの仕組み、他の KMS キーとの違い、およびデータを保護するために必要な KMS キーのタイプを判断する方法について説明します。また、非対称データキーペアの仕組み、および AWS KMS の外部でそれらを使用する方法についても説明します。

リージョン

非対称 KMS キーと非対称データキーペアは、AWS KMS がサポートするすべての AWS リージョンでサポートされます。

詳細はこちら

- 非対称 KMS キーを作成するには、「非対称 KMS キーを作成する」を参照してください。対称暗号化 KMS キーを作成するには、「キーの作成」を参照してください。
- マルチリージョンの非対称 KMS キーを作成するには、「マルチリージョンキーを作成する」を参照してください。
- KMS キーが対称か非対称かを調べるには、「非対称 KMS キーの識別」を参照してください。
- 各タイプの KMS キーに適用される AWS KMS API オペレーションを比較する表については、「the section called “キータイプリファレンス”」を参照してください。
- アカウントのプリンシパルが KMS キーおよびデータキーに使用できるキー仕様、キーの用途、暗号化アルゴリズム、署名アルゴリズムへのアクセスを制御するには、「the section called “AWS KMS 条件キー”」を参照してください。
- 各種タイプの KMS キーに適用されるリクエストクォータについては、the section called “クォータのリクエスト” を参照してください。
- 非対称 KMS キーを使用してメッセージに署名し、署名を検証する方法については、AWS セキュリティブログの Digital signing with the new asymmetric keys feature of AWS KMS を参照してください。

トピック

非対称 KMS キー

AWS KMS で非対称 KMS キーを作成できます。非対称 KMS キーは、数学的に関連する公開キーとプライベートキーペアを表します。パブリックキーは、たとえ信頼されていなくても、誰にでも渡すことができますが、シークレットキーは秘密にしておく必要があります。

非対称 KMS キーでは、プライベートキーが AWS KMS で作成され、AWS KMS を暗号化されないままにしません。プライベートキーを使用するには、AWS KMS を呼び出す必要があります。AWS KMS API オペレーションを呼び出すことによって、AWS KMS 内でパブリックキーを使用できます。または、パブリックキーをダウンロードし、AWS KMS の外部で使用することもできます。

AWS KMS を呼び出しできないユーザーが AWS 外部で暗号化を必要とするユースケースの場合は、非対称 KMS キーが適しています。ただし、AWS のサービスで保存または管理するデータを暗号化するための KMS キーを作成している場合は、対称暗号化 KMS キーを使用してください。AWS KMS と統合された AWS のサービスは、データの暗号化に対称暗号化 KMS キーのみを使用します。これらのサービスは、非対称 KMS キーを使用する暗号化をサポートしません。

AWS KMS では、3 種類の非対称 KMS キーがサポートされます。

- RSA KMS キー: 暗号化と復号、または署名と検証用 (両方ではない) の RSA キーペアを持つ KMS キー。AWS KMS は、さまざまなセキュリティ要件に対応し、複数のキーの長さをサポートします。
- 楕円曲線 (ECC) KMS キー: 署名と検証のための楕円曲線キーペアを持つ KMS キー。AWS KMS は、一般的に使用される複数の曲線をサポートします。
- SM2 KMS キー (中国リージョンのみ): 暗号化と復号、または署名と検証用 (両方ではない) の SM2 キーペアを持つ KMS キー。

非対称キーの設定を選択する方法については、「KMS キータイプの選択」を参照してください。AWS KMS がサポートする RSA KMS キーの暗号化および署名アルゴリズムの技術的な詳細については、「RSA キー仕様」を参照してください。AWS KMS がサポートする ECC KMS キーの署名アルゴリズムの技術的な詳細については、「楕円曲線のキー仕様」を参照してください。AWS KMS がサポートする SM2 KMS キー (中国リージョンのみ) の暗号化および署名アルゴリズムの技術的な詳細については、「SM2 キー仕様」を参照してください。

対称および非対称 KMS キーで実行できるオペレーションを比較した表については、「対称および非対称 KMS キーの比較」を参照してください。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

リージョン

非対称 KMS キーと非対称データキーペアは、AWS KMS がサポートするすべての AWS リージョンでサポートされます。

非対称 KMS キーを作成する

非対称 KMS キーは、AWS KMSコンソール、 CreateKey API、または AWS CloudFormation テンプレート を使用して作成できます。非対称 KMS キーは、暗号化または署名に使用できる公開キーとプライベートキーのキーペアを表します。プライベートキーは AWS KMS の範囲内にあります。AWS KMS の外部で使用するために公開キーをダウンロードするには、パブリックキーのダウンロード を参照してください。

AWS のサービスで保存または管理するデータを暗号化するための KMS キーを作成するときは、対称暗号化 KMS キーを使用します。AWS KMS と統合された AWS のサービスは、非対称 KMS キーをサポートしません。対称または非対称のどちらの KMS キーを作成するかを決定する方法については、「KMS キータイプの選択」を参照してください。

KMS キーの作成に必要なアクセス許可については、KMS キーを作成するためのアクセス許可 を参照してください。

トピック

- 非対称 KMS キーを作成する (コンソール)
- 非対称 KMS キーを作成する (AWS KMS API)

非対称 KMS キーを作成する (コンソール)

AWS Management Console を使用して、非対称 AWS KMS keys (KMS キー) を作成できます。各非対称 KMS キーは、公開キーとプライベートキーのキーペアを表します。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。

3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. [Create key] (キーの作成) を選択します。
5. 非対称 KMS キーを作成するには、[Key type] (キータイプ) で [Asymmetric] (非対称) を選択します。

   AWS KMS コンソールで対称暗号化 KMS キーを作成する方法については、「対称暗号化 KMS キーの作成 (コンソール)」を参照してください。
6. 公開キー暗号化用の非対称 KMS キーを作成するには、[Key usage] (キーの使用方法) で [Encrypt and decrypt] (暗号化と復号) を選択します。メッセージに署名して署名を検証するための非対称 KMS キーを作成するには、[Key usage] (キーの使用方法) で [Sign and verify] (署名と検証) を選択します。

   キー使用法の値の選択については、を参照してください キーの用途の選択。
7. 非対称 KMS キーの仕様 ([Key spec] (キー仕様)) を選択します。

   選択するキー仕様は、多くの場合、規制、セキュリティ、ビジネス要件によって決定されます。また、暗号化または署名する必要があるメッセージのサイズによっても影響を受ける可能性があります。一般に、長い暗号化キーは、ブルートフォース攻撃に対してより耐性があります。

   主要な仕様の選択については、「キー仕様の選択」を参照してください。
8. [次へ] をクリックします。
9. KMS キーのエイリアスを入力します。エイリアス名の先頭を **aws/** にすることはできません。この **aws/** プレフィックスは、アカウント内の AWS マネージドキー を表すために、Amazon Web Services によって予約されます。

   エイリアスは、コンソールおよび一部の AWS KMS API で、KMS キーを識別するために使用されるわかりやすい名前です。。保護する予定のデータタイプ、または KMS キーで使用する予定のアプリケーションを示すエイリアスを選択することをお勧めします。

   エイリアスは AWS Management Console で KMS キーを作成するときに必要です。CreateKey オペレーションを使用する場合、エイリアスを指定することはできませんが、コンソールまたは CreateAliasオペレーションを使用して既存の KMS キーのエイリアスを作成できます。詳細については、「エイリアスの使用」を参照してください。
10. (オプション) KMS キーの説明を入力します。

    保護する予定のデータタイプ、または KMS キーで使用する予定のアプリケーションを表す説明を入力します。

今すぐ説明を追加するか、キーの状態が Pending Deletion または Pending Replica Deletion でない限り、後でいつでも更新できます。既存のカスタマーマネージドキーの説明を追加、変更、または削除するには、 で説明を編集するAWS Management Consoleか、UpdateKeyDescriptionオペレーションを使用します。

11. (オプション) タグキーとオプションのタグ値を入力します。KMS キーに複数のタグを追加するには、[Add tag] (タグを追加) を選択します。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

12. [次へ] をクリックします。

13. KMS キーを管理できる IAM ユーザーとロールを選択します。

> **Note**
>
> このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシパルに KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

14. (オプション) 選択した IAM ユーザーとロールがこの KMS キーを削除しないようにするには、ページの下部にある [Key deletion] (キーの削除) セクションで、[Allow key administrators to delete this key] (キー管理者にこのキーの削除を許可する) のチェックボックスをオフにします。

15. [次へ] をクリックします。

16. 暗号化オペレーションで KMS キーを使用できる IAM ユーザーとロールを選択します。

> **Note**
>
> このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシ

パルに暗号化オペレーションで KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

17. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID 番号を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

> **Note**
>
> 外部アカウントでプリンシパルが KMS キーを使用できるようにするには、外部アカウントの管理者が、これらの許可を付与する IAM ポリシーを作成する必要があります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

18. [次へ] を選択します。
19. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。
20. [Finish] (完了) を選択し、KMS キーを作成します。

非対称 KMS キーを作成する (AWS KMS API)

CreateKey オペレーションを使用して、非対称 を作成できますAWS KMS key。以下の例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

非対称 KMS キーを作成する場合は、作成するキーのタイプを決定する KeySpec パラメータを指定する必要があります。また、ENCRYPT_DECRYPT または SIGN_VERIFY の KeyUsage 値も指定する必要があります。KMS キー作成後にこれらのプロパティを変更することはできません。

CreateKey オペレーションではエイリアスを指定することはできませんが、 CreateAliasオペレーションを使用して新しい KMS キーのエイリアスを作成できます。

> **Important**
>
> Description フィールドまたは Tags フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

次の例では、CreateKey オペレーションを使用して、公開キーの暗号化用に設計された 4096 ビット RSA キーの非対称 KMS キーを作成します。

```
$ aws kms create-key --key-spec RSA_4096 --key-usage ENCRYPT_DECRYPT
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1569973196.214,
        "MultiRegion": false,
        "KeySpec": "RSA_4096",
        "CustomerMasterKeySpec": "RSA_4096",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

次のコマンド例では、署名と検証に使用される ECDSA キーのペアを表す非対称 KMS キーを作成します。暗号化と復号のために楕円曲線キーペアを作成することはできません。

```
$ aws kms create-key --key-spec ECC_NIST_P521 --key-usage SIGN_VERIFY
{
    "KeyMetadata": {
        "KeyState": "Enabled",
```

```
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1570824817.837,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ],
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "AWSAccountId": "111122223333",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Enabled": true,
        "MultiRegion": false,
        "KeyUsage": "SIGN_VERIFY"
    }
}
```

パブリックキーのダウンロード

AWS Management Console または AWS KMS API を使用して、非対称 KMS キーペアからパブリックキーを表示、コピー、ダウンロードできます。非対称 KMS キーに対する `kms:GetPublicKey` アクセス許可が必要です。

各非対称 KMS キーペアは、AWS KMS を暗号化されないままにしないプライベートキーと、ダウンロードして共有できるパブリックキーで構成されます。

パブリックキーを共有して、プライベートキーでのみ復号できる AWS KMS 外部のデータを他のユーザーが暗号化できるようにすることもできます。または、プライベートキーを使用して生成した AWS KMS の外部にあるデジタル署名を他のユーザーが確認できるようにすることができます。

AWS KMS 内の非対称 KMS キーでパブリックキーを使用すると、すべての AWS KMS オペレーションの一部である認証、認可、ロギングの利点を得ることができます。また、復号できないデータを暗号化するリスクも軽減します。これらの機能は、AWS KMS の外部では有効ではありません。詳細については、「パブリックキーのダウンロードに関する特別な考慮事項」を参照してください。

> **Tip**
>
> データキーまたは SSH キーをお探しですか。このトピックでは、プライベートキーをエクスポートすることができない AWS Key Management Service での非対称キーの管理方法を説明しています。プライベートキーが対称暗号化 KMS キーで保護されているエクスポート

可能なデータキーペアについては、「」を参照してくださいGenerateDataKeyPair。Amazon EC2 インスタンスに関連付けられたパブリックキーのダウンロード方法については、「Linux インスタンス用 Amazon EC2 ユーザーガイド」および「Windows インスタンス用 Amazon EC2 ユーザーガイド」の「パブリックキーの取得」を参照してください。

トピック

- パブリックキーのダウンロードに関する特別な考慮事項
- パブリックキーをダウンロードする (コンソール)
- パブリックキーをダウンロードする (AWS KMS API)

パブリックキーのダウンロードに関する特別な考慮事項

KMS キーを保護するために、AWS KMS は、アクセス制御、認証された暗号化、すべてのオペレーションの詳細なログを提供します。AWS KMS は、KMS キーの使用を一時的または永続的に阻止することもできます。最後に、AWS KMS オペレーションは、復号できないデータを暗号化するリスクを最小限に抑えるように設計されています。これらの機能は、ダウンロードしたパブリックキーを AWS KMS の外部で使用する場合には使用できません。

認証

AWS KMS 内の KMS キーへのアクセスを制御するキーポリシーおよび IAM ポリシーは、AWS の外部で実行されるオペレーションには影響しません。パブリックキーを取得できるユーザーは、KMS キーで、データを暗号化、または署名を検証する許可がない場合でも、AWS KMS の外部でパブリックキーを使用できます。

キーの用途の制限

キーの用途の制限は、AWS KMS の外部では有効ではありません。SIGN_VERIFY の KeyUsage を持つ KMS キーで Encrypt オペレーションを呼び出すと、AWS KMS オペレーションは失敗します。ただし、SIGN_VERIFY の KeyUsage で、KMS キーからパブリックキーを使用して AWS KMS の外部のデータを暗号化する場合、データを復号することはできません。

アルゴリズムの制限

AWS KMS がサポートする暗号化および署名アルゴリズムの制限は、AWS KMS の外部では有効ではありません。AWS KMS の外部の KMS キーからのパブリックキーを使用してデータを暗号化し、AWS KMS がサポートしていない暗号化アルゴリズムを使用すると、データを復号できません。

KMS キーの無効化と削除

AWS KMS 内の暗号化オペレーションで KMS キーの使用を阻止するために実行できるアクションは、AWS KMS の外部でパブリックキーを使用することを妨げません。例えば、KMS キーの無効化、KMS キーの削除のスケジューリング、KMS キーの削除、KMS キーからのキーマテリアルの削除は、AWS KMS の外部のパブリックキーには影響しません。非対称 KMS キーを削除、またはそのキーマテリアルを削除したり紛失したりすると、AWS KMS の外部にあるパブリックキーで暗号化したデータを回復できなくなります。

ログ記録

AWS CloudTrail ログは、リクエスト、レスポンス、日付、時刻、許可されたユーザーなど、すべての AWS KMS オペレーションを記録しますが、AWS KMS の外部でのパブリックキーの使用は記録されません。

SM2 キーペアによるオフライン検証 (中国リージョンのみ)

SM2 パブリックキーを使用して AWS KMS の外部で署名を検証するには、識別 ID を指定する必要があります。デフォルトでは、AWS KMS は識別 ID として 1234567812345678 を使用します。詳細については、「SM2 キーペアによるオフライン検証」(中国リージョンのみ) を参照してください。

パブリックキーをダウンロードする (コンソール)

AWS Management Console を使用して、AWS アカウント アカウントの非対称 KMS キーからパブリックキーを表示、コピー、ダウンロードできます。別の AWS アカウント で非対称 KMS キーからパブリックキーをダウンロードするには、AWS KMS API を使用します。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 非対称 KMS キーのエイリアスまたはキー ID を選択します。
5. [Cryptographic configuration] (暗号化の設定) タブを選択します。[Key spec] (キー仕様)、[Key usage] (キーの用途)、[Encryption algorithms] (暗号化アルゴリズム)、または [Signing Algorithms] (署名アルゴリズム) フィールドの値を記録します。AWS KMS の外部でパブリックキーを使用するには、これらの値を使用する必要があります。パブリックキーを共有するときは、必ずこの情報を共有してください。

6. [Public key] (パブリックキー) タブを選択します。
7. パブリックキーをクリップボードにコピーするには、[Copy] (コピー) を選択します。パブリックキーをファイルにダウンロードするには、[Download] (ダウンロード) を選択します。

パブリックキーをダウンロードする (AWS KMS API)

GetPublicKey オペレーションは、非対称 KMS キーでパブリックキーを返します。また、キーの用途や暗号化アルゴリズムなど、AWS KMS の外部でパブリックキーを正しく使用するために必要な重要な情報も返されます。これらの値を保存し、パブリックキーを共有する場合は必ず共有してください。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

KMS キーを指定するには、そのキー ID、キー ARN、エイリアス名、エイリアス ARN を使用します。エイリアス名を使用する場合は、接頭辞として alias/ を付けます。別の AWS アカウント で KMS キーを指定するには、そのキー ARN またはエイリアス ARN を使用する必要があります。

このコマンドを実行する前に、サンプルのエイリアス名を KMS キーの有効な識別子に置き換えます。このコマンドを実行するには、KMS キーに対する kms:GetPublicKey アクセス許可が必要です。

```
$ aws kms get-public-key --key-id alias/example_RSA_3072

{
    "KeySpec": "RSA_3072",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ],
    "PublicKey": "MIIBojANBgkqhkiG..."
}
```

非対称 KMS キーの識別

特定の KMS キーが非対称 KMS キーであるかどうかを判断するには、キーのタイプ、またはキーの仕様を見つけます。AWS KMS コンソールまたは AWS KMS API を使用できます。

これらの方法の一部では、キー使用法、および KMS キーがサポートする暗号化アルゴリズムまたは署名アルゴリズムなど、KMS キーの暗号化設定のその他の状況も表示されます。既存の KMS キーの暗号化設定を表示できますが、変更することはできません。

コンソール表示のソート、フィルタリング、列の選択など、KMS キーの表示に関する一般情報については、コンソールで KMS キーを表示する を参照してください。

トピック

- KMS キーテーブルでキータイプを検索する
- [詳細] ページでのキータイプの検索
- AWS KMS API を使用したキー仕様の検索

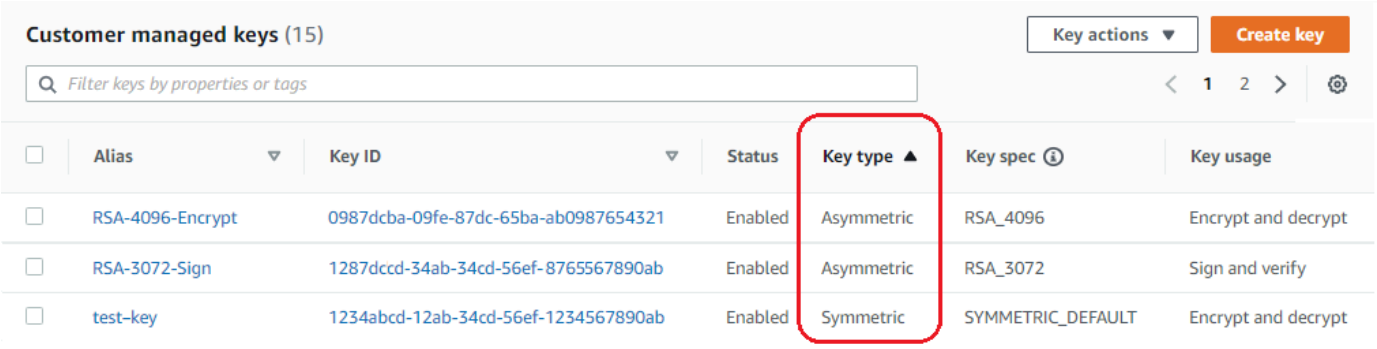
KMS キーテーブルでキータイプを検索する

AWS KMS コンソールのキータイプ列には、各 KMS キーが対称であるか非対称であるかが表示されます。コンソールの [Customer managed keys] (カスタマーマネージドキー) または [AWS マネージドキー] ページで、キータイプ列を KMS キーテーブルに追加できます。

KMS キーテーブルで対称および非対称 KMS キーを識別するには、以下の手順に従います。

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。
4. キータイプ列には、各 KMS キーが対称であるか非対称であるかが表示されます。キータイプにより、ソートおよびフィルタリングも実行できます。

   KMS キーテーブルにキータイプ列が表示されない場合は、ページの右上隅にある歯車アイコン、[Key type (キータイプ)]、[Confirm (確認)] の順に選択します。[Key spec (キー仕様)] 列と [Key usage (キー使用法)] 列を追加することもできます。



[詳細] ページでのキータイプの検索

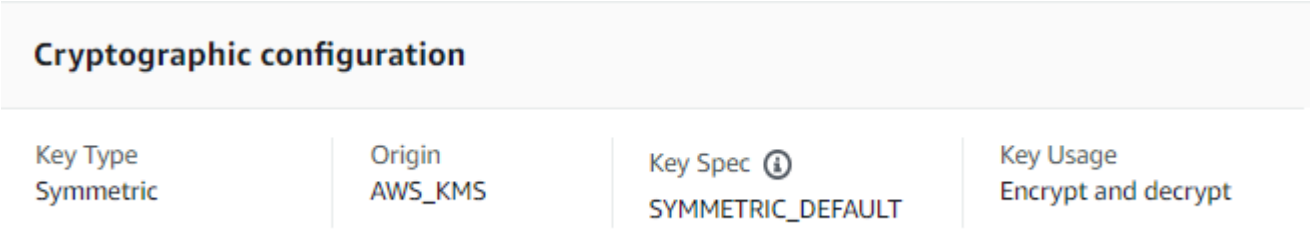
AWS KMS コンソールでは、各 KMS キーの詳細ページに Cryptographic Configuration タブがあります。このタブには、キータイプ (対称または非対称) と、KMS キーに関するその他の暗号化の詳細が表示されます。

KMS キーの詳細ページで対称および非対称 KMS キーを識別するには、以下の手順に従います。

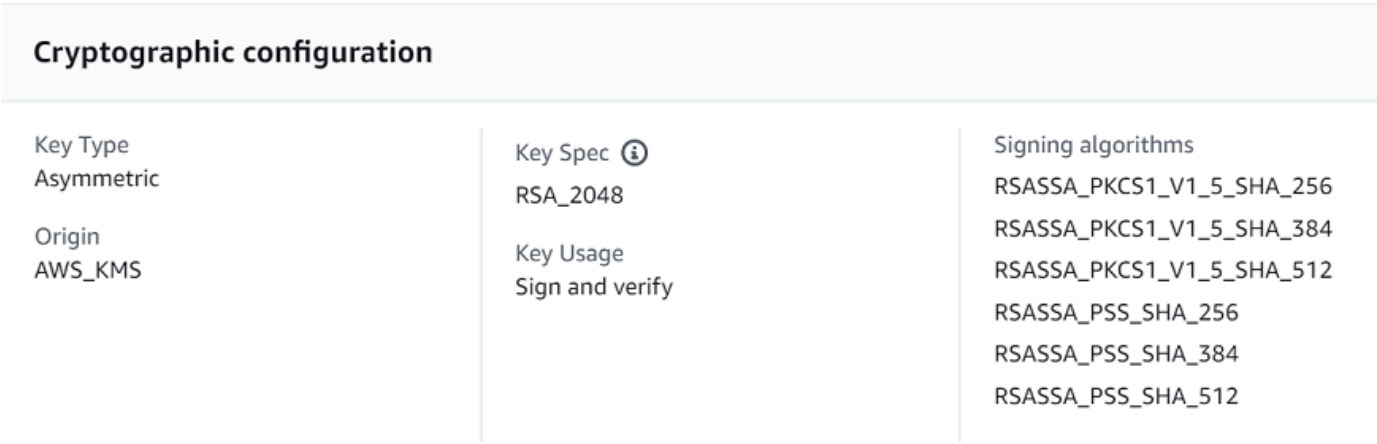
1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ユーザーが作成および管理するアカウント内のキーを表示するには、ナビゲーションペインで [Customer managed keys] (カスタマーマネージドキー) を選択します。AWS によって作成および管理されるアカウントのキーを表示するには、ナビゲーションペインで [AWS マネージドキー] を選択します。
4. KMS キーのエイリアスまたは キー ID を選択します。
5. [Cryptographic configuration] (暗号化の設定) タブを選択します。これらのタブは、[General configuration] (一般設定) セクションの下にあります。

   [暗号化設定] タブに [Key Type,] が表示され、対称か非対称かを示します。KMS キーが暗号化と復号、または署名と検証に使用できるかどうかを示すキー使用法など、KMS キーに関するその他の詳細も表示されます。非対称 KMS キーの場合は、KMS キーがサポートする暗号化アルゴリズムまたは署名アルゴリズムが表示されます。

   例えば、以下は対称暗号化 KMS キーの [Cryptographic configuration] (暗号化設定) タブの例です。



次の例は、署名と検証に使用される非対称 RSA KMS キーの暗号化設定タブです。



AWS KMS API を使用したキー仕様の検索

KMS キーが対称か非対称かを判断するには、 DescribeKeyオペレーションを使用します。レスポンスの KeySpec フィールドには、KMS キーのキー仕様が含まれます。対称暗号化 KMS キーの場合、KeySpec の値は SYMMETRIC_DEFAULT です。これ以外の値は、非対称 KMS キーまたは HMAC KMS キーを示しています。

> **Note**
>
> CustomerMasterKeySpec メンバーは非推奨です。代わりに KeySpec を使用してください。互換性を破る変更を防ぐために、DescribeKey レスポンスは同じ値を持つ KeySpec および CustomerMasterKeySpec メンバーを含みます。

例えば、DescribeKey は、対称暗号化 KMS キーについて以下のレスポンスを返します。KeySpec 値は SYMMETRIC_DEFAULT です。

```
{
  "KeyMetadata": {
```

```
    "AWSAccountId": "111122223333",
    "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
    "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
    "CreationDate": 1496966810.831,
    "Enabled": true,
    "Description": "",
    "KeyState": "Enabled",
    "Origin": "AWS_KMS",
    "KeyManager": "CUSTOMER",
    "MultiRegion": false,
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

署名および検証に使用される非対称 RSA KMS キーに対する DescribeKey のレスポンスは、この例のようになります。KeySpec 値は RSA_2048、KeyUsage は SIGN_VERIFY です。SigningAlgorithms 要素は KMS キーの有効な署名アルゴリズムを一覧表示します。

```
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1571767572.317,
    "CustomerMasterKeySpec": "RSA_2048",
    "Enabled": false,
    "Description": "",
    "KeyState": "Disabled",
    "Origin": "AWS_KMS",
    "MultiRegion": false,
    "KeyManager": "CUSTOMER",
    "KeySpec": "RSA_2048",
    "KeyUsage": "SIGN_VERIFY",
    "SigningAlgorithms": [
        "RSASSA_PKCS1_V1_5_SHA_256",
        "RSASSA_PKCS1_V1_5_SHA_384",
```

```
        "RSASSA_PKCS1_V1_5_SHA_512",
        "RSASSA_PSS_SHA_256",
        "RSASSA_PSS_SHA_384",
        "RSASSA_PSS_SHA_512"
    ]
  }
}
```

非対称キーの仕様

以下のトピックは、非対称 KMS キーについて AWS KMS がサポートするキー仕様に関する技術的な情報を提供します。比較できるように、対称暗号化キーの SYMMETRIC_DEFAULT キー仕様に関する情報が含まれています。

トピック

- RSA キー仕様
- 楕円曲線のキー仕様
- SM2 キー仕様 (中国リージョンのみ)
- SYMMETRIC_DEFAULT キー仕様

RSA キー仕様

RSA キー仕様を使用する際、AWS KMS は RSA キーペアを持つ非対称 KMS キーを作成します。プライベートキーが暗号化されないまま AWS KMS から出ていくことはありません。AWS KMS 内でパブリックキーを使用することも、AWS KMS の外部で使用するためにパブリックキーをダウンロードすることもできます。

> **Warning**
>
> AWS KMS の外部でデータを暗号化する場合は、暗号文を復号できることを確認してください。AWS KMS から削除された KMS キーのパブリックキー、署名と検証用に設定された KMS キーのパブリックキー、または KMS キーでサポートされていない暗号化アルゴリズムを使用する場合、データは回復できません。

AWS KMS では、非対称 KMS キーを RSA キーペアとともに暗号化と復号、または署名と検証に使用できますが、両方には使用できません。このプロパティは、キーの用途と呼ばれ、キー仕様とは別に決定されますが、キー仕様を選択する前に決定する必要があります。

AWS KMS は、暗号化と復号、署名と検証について、次の RSA キー仕様をサポートしています。

- RSA_2048
- RSA_3072
- RSA_4096

RSA キー仕様は、RSA キーの長さ（ビット単位）によって異なります。選択する RSA キー仕様は、セキュリティ標準またはタスクの要件によって決定される場合があります。一般的に、タスクに実用的で手頃な価格の、最大のキーを使用します。RSA キー仕様の異なる KMS キーの暗号化オペレーションは、料金が異なります。AWS KMS の価格設定については、「AWS Key Management Service の料金」を参照してください。リクエストクォータの詳細については、「クォータのリクエスト」を参照してください。

暗号化および復号の RSA キー仕様

RSA 非対称 KMS キーを暗号化および復号に使用する場合、パブリックキーで暗号化し、プライベートキーで復号します。RSA KMS キーの AWS KMS で `Encrypt` オペレーションを呼び出す場合、AWS KMS は RSA キーペアのパブリックキーと、指定された暗号化アルゴリズムを使用してデータを暗号化します。暗号テキストを復号するには、`Decrypt` オペレーションを呼び出し、同じ KMS キーと暗号化アルゴリズムを指定します。AWS KMS は次に、RSA キーペアのプライベートキーを使用してデータを復号します。

パブリックキーをダウンロードして、AWS KMS の外部のデータを暗号化するために使用することもできます。必ず、AWS KMS が RSA KMS キーでサポートする暗号化アルゴリズムを使用してください。暗号テキストを復号するには、同じ KMS キーと暗号化アルゴリズムを使用して `Decrypt` 関数を呼び出します。

AWS KMS は、RSA キー仕様を持つ KMS キーの 2 つの暗号化アルゴリズムをサポートします。PKCS #1 v2.2 で定義されるこれらのアルゴリズムは、内部的に使用するハッシュ関数によって異なります。AWS KMS で、RSAES_OAEP アルゴリズムは、ハッシュ目的とマスク生成関数 (MGF1) の両方に常に同じハッシュ関数を使用します。Encrypt および Decrypt オペレーションを呼び出すときは、暗号化アルゴリズムを指定する必要があります。リクエストごとに異なるアルゴリズムを選択できます。

RSA キー仕様にサポートされる暗号化アルゴリズム

暗号化アルゴリズム	アルゴリズムの説明
RSAES_OAEP_SHA_1	PKCS #1 v2.2、セクション 7.1。ハッシュと MGF1 マスク生成機能の両方に空のラベルとともに SHA-1 を使用した OAEP パディングによるRSA暗号化。
RSAES_OAEP_SHA_256	PKCS #1、セクション 7.1. ハッシュと MGF1 マスク生成機能の両方に空のラベルとともに SHA-256 を使用した OAEP パディングによる RSA暗号化。

特定の暗号化アルゴリズムを使用するように KMS キーを設定することはできません。ただし、kms:EncryptionAlgorithm policy 条件を使用して、プリンシパルが KMS キーで使用できる暗号化アルゴリズムを指定できます。

KMS キーの暗号化アルゴリズムを取得するには、 AWS KMSコンソールで KMS キーの暗号化設定を表示するか、 DescribeKeyオペレーションを使用します。 は、 AWS KMSコンソールまたは GetPublicKeyオペレーションを使用してパブリックキーをダウンロードするときに、キー仕様と暗号化アルゴリズムAWS KMSも提供します。

各リクエストで暗号化できるプレーンテキストデータの長さに基づいて、RSA キー仕様を選択できます。次の表に、Encrypt オペレーションに対する 1 回の呼び出しで暗号化できるプレーンテキストの最大サイズをバイト単位で示します。値は、キー仕様と暗号化アルゴリズムによって異なります。比較すると、対称暗号化 KMS キーは一度に最大 4,096 バイトを暗号化するために使用できます。

これらのアルゴリズムのプレーンテキストの最大長 (バイト単位) を計算するには、次の式を使用します。(*key_size_in_bits* / 8) - (2 * *hash_length_in_bits*/8) - 2。例えば、RSA_2048 と SHA-256 の場合、バイト単位のプレーンテキストの最大サイズは、(2048/8) - (2 * 256/8) -2 = 190 です。

暗号化オペレーションの最大プレーンテキストサイズ (バイト単位)

	暗号化アルゴリズム	
キー仕様	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

署名および検証用の RSA キー仕様

署名と検証に RSA 非対称 KMS キーを使用する場合、プライベートキーを持つメッセージの署名を生成し、パブリックキーで署名を検証します。

非対称 KMS キーの AWS KMS で `Sign` オペレーションを呼び出すと、AWS KMS は RSA キーペアのプライベートキー、メッセージ、指定した署名アルゴリズムを使用して、署名を生成します。署名を検証するには、Verify オペレーションを呼び出します。署名に加えて、同じ KMS キー、メッセージ、署名アルゴリズムを指定します。AWS KMS は、RSA キーペアのパブリックキーを使用して、署名を検証します。また、パブリックキーをダウンロードして、AWS KMS の外部で署名を検証するために使用することもできます。

AWS KMS は、RSA キー仕様のすべての KMS キーに対して、次の署名アルゴリズムをサポートします。Sign オペレーションと Verify オペレーションを呼び出すときは、署名アルゴリズムを指定する必要があります。リクエストごとに異なるアルゴリズムを選択できます。RSA キーペアで署名する場合は、RSASSA-PSS アルゴリズムが推奨されます。既存のアプリケーションとの互換性を保つため、RSASSA-PKCS1-v1_5 アルゴリズムが採用されています。

RSA キー仕様でサポートされる署名アルゴリズム

署名アルゴリズム	アルゴリズムの説明
RSASSA_PSS_SHA_256	PKCS #1 v2.2、セクション 8.1、メッセージダイジェストと MGF1 マスク生成機能の両方に SHA-256 とともに 256 ビットソルトを使用する PSS パディング付きの RSA 署名

署名アルゴリズム	アルゴリズムの説明
RSASSA_PSS_SHA_384	PKCS #1 v2.2、セクション 8.1、メッセージダイジェストと MGF1 マスク生成機能の両方に SHA-384 とともに 384 ビットソルトを使用する PSS パディング付きの RSA 署名
RSASSA_PSS_SHA_512	PKCS #1 v2.2、セクション 8.1、メッセージダイジェストと MGF1 マスク生成機能の両方に SHA-512 とともに 512 ビットソルトを使用する PSS パディング付きの RSA 署名
RSASSA_PKCS1_V1_5_SHA_256	PKCS #1 v2.2、セクション 8.2、PKCS #1v1.5 パディングおよび SHA-256 を使用した RSA 署名
RSASSA_PKCS1_V1_5_SHA_384	PKCS #1 v2.2、セクション 8.2、PKCS #1v1.5 パディングおよび SHA-384 を使用した RSA 署名
RSASSA_PKCS1_V1_5_SHA_512	PKCS #1 v2.2、セクション 8.2、PKCS #1v1.5 パディングおよび SHA-512 を使用した RSA 署名

特定の署名アルゴリズムを使用するように KMS キーを設定することはできません。ただし、kms:SigningAlgorithm policy 条件を使用して、プリンシパルが KMS キーで使用できる署名アルゴリズムを指定できます。

KMS キーの署名アルゴリズムを取得するには、AWS KMSコンソールまたは DescribeKeyオペレーションを使用して KMS キーの暗号化設定を表示します。 は、AWS KMSコンソールまたは GetPublicKeyオペレーションを使用してパブリックキーをダウンロードするときに、キー仕様と署名アルゴリズムAWS KMSも提供します。

楕円曲線のキー仕様

楕円曲線 (ECC) キー仕様を使用すると、AWS KMS は署名と検証のために ECC キーペアを持つ非対称 KMS キーを作成します。署名を生成するプライベートキーでは、AWS KMS が暗号化されない

ままになることはありません。パブリックキーを使用して AWS KMS 内の署名を検証したり、AWS KMS の外部で使用するためにパブリックキーをダウンロードしたりできます。

AWS KMS は、非対称 KMS キーの、次の ECC キー仕様をサポートします。

- 非対称 NIST 推奨楕円曲線キーペア (署名と検証)
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
- その他の非対称楕円曲線キーペア (署名と検証)
  - ECC_SECG_P256K1 (secp256k1)。一般に暗号通貨に用いられる。

選択する ECC キー仕様は、セキュリティ標準またはタスクの要件によって決定される場合があります。一般的に、タスクに実用的で手頃な価格の、最も多くのポイントがある曲線を使用します。

暗号通貨で使用する非対称 KMS キーを作成する場合は、ECC_SECG_P256K1 キー仕様を使用します。このキー仕様を他の目的に使用することもできますが、Bitcoin やその他の暗号化通貨には必要です。

ECC キー仕様が異なる KMS キー仕様は料金が異なるため、リクエストクォータも異なる場合があります。AWS KMS の料金については、AWS Key Management Service の料金を参照してください。リクエストクォータの詳細については、「クォータのリクエスト」を参照してください。

次の表に、ECC キー仕様ごとに AWS KMS がサポートする署名アルゴリズムを示します。特定の署名アルゴリズムを使用するように KMS キーを設定することはできません。ただし、kms:SigningAlgorithm policy 条件を使用して、プリンシパルが KMS キーで使用できる署名アルゴリズムを指定できます。

ECC キー仕様でサポートされる署名アルゴリズム

キー仕様	署名アルゴリズム	アルゴリズムの説明
ECC_NIST_P256	ECDSA_SHA_256	メッセージダイジェストのためにキーおよび SHA-256 で指定された曲線を使用する、NIST FIPS 186-4、セクション 6.4、ECDSA 署名。

キー仕様	署名アルゴリズム	アルゴリズムの説明
ECC_NIST_P384	ECDSA_SHA_384	メッセージダイジェストのためにキーおよび SHA-384 で指定された曲線を使用する、NIST FIPS 186-4、セクション 6.4、ECDSA 署名。
ECC_NIST_P521	ECDSA_SHA_512	メッセージダイジェストのためにキーおよび SHA-512 で指定された曲線を使用する、NIST FIPS 186-4、セクション 6.4、ECDSA 署名。
ECC_SECG_P256K1	ECDSA_SHA_256	メッセージダイジェストのためにキーおよび SHA-256 で指定された曲線を使用する、NIST FIPS 186-4、セクション 6.4、ECDSA 署名。

SM2 キー仕様 (中国リージョンのみ)

SM2 キー仕様は、中国国家商業暗号局 (OSCCA) によって公開されている GM/T シリーズの仕様で定義されている楕円曲線のキー仕様です。SM2 キー仕様は、中国リージョンでのみ利用可能です。SM2 キー仕様を使用する際、AWS KMS は SM2 キーペアを持つ非対称 KMS キーを作成します。AWS KMS 内でSM2 キーを使用することも、AWS KMS の外部で使用するためにパブリックキーをダウンロードすることもできます。

ECC キースペックとは異なり、署名と検証または暗号化と復号に SM2 KMS キーを使用できます。KMS キーの作成時にキーの用途を指定する必要があります。キーの作成後に変更することはできません。

AWS KMS は、以下の SM2 暗号化および署名アルゴリズムをサポートしています。

- SM2PKE 暗号化アルゴリズム

  SM2PKE は、GM/T 0003.4-2012 で OSCCA によって定義された楕円曲線ベースの暗号化アルゴリズムです。

- SM2DSA 署名アルゴリズム

  SM2DSA は、GM/T 0003.2-2012 で OSCCA によって定義された楕円曲線ベースの暗号化アルゴリズムです。SM2DSA には、SM3 ハッシュアルゴリズムでハッシュ化され、AWS KMS に渡されたメッセージまたはメッセージダイジェストと組み合わされた識別 ID が必要です。この連結された値は、AWS KMS によりハッシュ化されて署名されます。

SM2 によるオフライン運用 (中国リージョンのみ)

オフラインオペレーションで SM2 キーペアのパブリックキーをダウンロードし、AWS KMS の外部でのオペレーションを行えます。ただし、SM2 パブリックキーをオフラインで使用する場合、追加の変換と計算を手動で実行する必要がある場合があります。SM2DSA 操作では、識別 ID の提供またはメッセージダイジェストの計算が必要になる場合があります。SM2PKE 暗号化オペレーションでは、生の暗号文出力を AWS KMS が受け入れるフォーマットに変換する必要のある場合があります。

これらの操作を支援するために、Java の `SM2OfflineOperationHelper` クラスには、タスクを実行するメソッドがあります。このヘルパークラスは、他の暗号化プロバイダのモデルとして使用できます。

> **Important**
>
> `SM2OfflineOperationHelper` リファレンスコードは Bouncy Castle バージョン 1.68 と互換性があるように設計されています。他のバージョンに関するヘルプについては、bouncycastle.org にアクセスしてください。

SM2 キーペアによるオフライン検証 (中国リージョンのみ)

SM2 パブリックキーを使用して AWS KMS の外部で署名を検証するには、識別 ID を指定する必要があります。生のメッセージ `MessageType:RAW` を Sign API に渡すとき、AWS KMS は GM/T 0009-2012 で OSCCA によって定義されているデフォルトの識別 ID `1234567812345678` を使用します。独自の識別 ID を AWS KMS で指定することはできません。

ただし、AWS メッセージダイジェストを外部で生成する場合、独自の識別 ID を指定して、メッセージダイジェスト `MessageType:DIGEST` を AWS KMS に渡し、署名することができます。これを行うには、`SM2OfflineOperationHelper` クラスの `DEFAULT_DISTINGUISHING_ID` 値を変更します。指定する識別 ID は、最大 8,192 文字の任意の文字列です。AWS KMS がメッセージダイ

ジェストに署名した後、メッセージダイジェストまたはメッセージと、ダイジェストを計算してオフラインで検証するために使用される識別 ID のいずれかが必要です。

`SM2OfflineOperationHelper` クラス

AWS KMS 内では、生の暗号文変換と SM2DSA メッセージダイジェストの計算が自動的に行われます。どの暗号化プロバイダーも同じ方法で SM2 を実装しているとは限りません。OpenSSL バージョン 1.1.1 以降など、一部のライブラリはこれらのアクションを自動的に実行します。AWS KMS は OpenSSL バージョン 3.0 でのテストでこの動作を確認しました。変換と計算を手動で実行するには、Bouncy Castle などのライブラリを持つ以下の `SM2OfflineOperationHelper` クラスを使用します。

`SM2OfflineOperationHelper` クラスは、次のオフラインオペレーションのためのメソッドを提供します。

- メッセージダイジェストの計算

  オフライン検証に使用できる、または AWS KMS に渡して署名することができるメッセージダイジェストをオフラインで生成するには、`calculateSM2Digest` メソッドを使用します。`calculateSM2Digest` メソッドは SM3 ハッシュアルゴリズムでメッセージダイジェストを生成します。GetPublicKey API は、パブリックキーをバイナリ形式で返します。バイナリキーを Java に解析する必要があります PublicKey。解析されたパブリックキーをメッセージとともに提供します。このメソッドは、メッセージをデフォルトの識別 ID、`1234567812345678` と自動的に組み合わせますが、`DEFAULT_DISTINGUISHING_ID` 値を変更して、独自の識別 ID を設定することもできます。

- 検証

  署名をオフラインで検証するには、`offlineSM2DSAVerify` メソッドを使用します。`offlineSM2DSAVerify` メソッドは、指定された識別 ID から計算されたメッセージダイジェストと、指定された元のメッセージを使用してデジタル署名を検証します。GetPublicKey API は、パブリックキーをバイナリ形式で返します。バイナリキーを Java に解析する必要があります PublicKey。解析されたパブリックキーに、元のメッセージと検証する署名を指定します。詳細については、「SM2 キーペアによるオフライン検証」を参照してください。

- 暗号化

  プレーンテキストをオフラインで暗号化するには、`offlineSM2PKEEncrypt` メソッドを使用します。この方法により、AWS KMS が復号できる形式の暗号文であることが保証されます。`offlineSM2PKEEncrypt` メソッドは、プレーンテキストを暗号化し、生成された生の暗号文を SM2PKE によって ASN.1 形式に変換します。GetPublicKey API は、パブリックキーを

バイナリ形式で返します。バイナリキーを Java に解析する必要があります PublicKey。解析したパブリックキーに、暗号化するプレーンテキストを指定します。

変換を実行する必要があるかどうかわからない場合は、次の OpenSSL オペレーション使用して暗号文の形式をテストします。オペレーションが失敗した場合は、暗号文を ASN.1 形式に変換する必要があります。

```
openssl asn1parse -inform DER -in ciphertext.der
```

デフォルトでは、SM2DSA オペレーションのメッセージダイジェストを生成するとき、SM2OfflineOperationHelper クラスはデフォルトの識別 ID、1234567812345678 を使用します。

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
```

```
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
 distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
 "1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
 GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
 message) throws
            NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;

        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
 0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(entla.length +
 DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
                        xa.length +
 ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
                        .array());

        // Combine hashed distinguishing ID with original message to generate final
 digest
        return MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(za.length +
 message.length).put(za).put(message)
```

```
                        .array());
    }

    // ***offlineSM2DSAVerify***
    // Verify digital signature with SM2 public key
    public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
            final byte [] signature) throws InvalidKeyException {
        final SM2Signer signer = new SM2Signer();
        CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
        cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
        signer.init(false, cipherParameters);
        signer.update(message, 0, message.length);
        return signer.verifySignature(signature);
    }

    // ***offlineSM2PKEEncrypt***
    // Encrypt data with SM2 public key
    public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
plaintext) throws
            NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException, IOException {
        final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
        sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);

        // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
        final byte [] cipherText = sm2Cipher.doFinal(plaintext);

        // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
        final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
        final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
7) / 8 * 2 + 1;
        final int sm3HashLength = 32;
        final int xCoordinateInCipherText = 33;
        final int yCoordinateInCipherText = 65;
        byte[] coords = new byte[coordinateLength];
        byte[] sm3Hash = new byte[sm3HashLength];
        byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
sm3HashLength];

        // Split components out of the ciphertext
```

```
        System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
        System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
 sm3HashLength);
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
 0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

SYMMETRIC_DEFAULT キー仕様

デフォルトのキー仕様である SYMMETRIC_DEFAULT は、対称暗号化 KMS キーのキー仕様です。AWS KMS コンソールでキーのタイプに [Symmetric] (対称)、キーの用途に [Encrypt and decrypt] (暗号化および復号) を選択すると、SYMMETRIC_DEFAULT キー仕様が選択されます。CreateKey オペレーションでは、KeySpec値を指定しない場合、SYMMETRIC_DEFAULT が選択されます。別のキー仕様を使用する理由がない場合は、SYMMETRIC_DEFAULT を選択することをお勧めします。

SYMMETRIC_DEFAULT は現在、AES-256-GCM を表しています。Galois Counter Mode (GCM) のアドバンスト暗号化スタンダード (AES) に基づく対称アルゴリズムは、安全な暗号化のための業界標準である 256 ビットキーを備えています。このアルゴリズムが生成する暗号文は、暗号化コンテキストなどの追加認証データ (AAD) をサポートし、GCM は暗号文での追加の整合性チェックを提供します。技術的な詳細については、「AWS Key Management Service 暗号化の詳細」を参照してください。

AES-256-GCM で暗号化されたデータは、現在も将来も保護されています。暗号作成者は、このアルゴリズムには量子耐性があると考えています。理論上の将来、256 ビット AES-GCM キーで作成された暗号文に対する大規模な量子コンピューティング攻撃は、キーの効果的なセキュリティを 128 ビットに低下させます。ただし、このセキュリティレベルは、AWS KMS 暗号文に対するブルートフォース攻撃を実行不可能にするのに十分です。

中国リージョンでは唯一の例外で、SYMMETRIC_DEFAULT は SM4 暗号化を使用する 128 ビットの対称キーを表します。128 ビット SM4 キーは、中国リージョンでのみ作成できます。中国リージョンでは、256 ビット AES-GCM KMS キーを作成することはできません。

AWS KMS で対称暗号化 KMS キーを使用して、データの暗号化、復号、再暗号化、および生成されたデータキーとデータキーペアの保護を行うことができます。AWS KMS と統合された AWS のサービスは、対称暗号化 KMS キーを使用して保管中のデータを暗号化します。対称暗号化 KMS キーに独自のキーマテリアルをインポートし、カスタムキーストアで対称暗号化 KMS キーを作成することができます。対称および非対称 KMS キーで実行できるオペレーションを比較した表については、「対称および非対称 KMS キーの比較」を参照してください。

AWS KMS と対称暗号化キーの技術的詳細については、「AWS Key Management Service の暗号化の詳細説明」を参照してください。

AWS KMS での HMAC キー

Hash-based Message Authentication Code (HMAC) KMS キーは、AWS KMS 内で HMAC を生成して検証するために使用される対称キーです。それぞれの HMAC KMS キーに関連する一意のキーマテリアルは、HMAC アルゴリズムが必要とするシークレットキーを提供します。HMAC KMS キーを `GenerateMac` および `VerifyMac` オペレーションで使用して、AWS KMS 内のデータの整合性と信頼性を検証することができます。

HMAC アルゴリズムは、暗号化ハッシュ関数と共有シークレットキーを組み合わせます。これらはメッセージとシークレットキー (HMAC KMS キーのキーマテリアルなど) を使用して、一意の固定サイズのコードまたはタグを返します。メッセージの文字が 1 字でも違う場合、またはシークレットキーが同一ではない場合、結果として得られるタグはまったく異なるものになります。HMAC は、シークレットキーをリクエストすることによって信頼性も提供します。シークレットキーがなければ、同一の HMAC タグを生成することは不可能です。HMAC は対称署名と呼ばれることもあります。これらはデジタル署名のように機能しますが、署名と検証の両方に単一のキーを使用するからです。

AWS KMS が使用する HMAC KMS キーと HMAC アルゴリズムは、RFC 2104 で定義されている業界標準に準拠しています。AWS KMS GenerateMac オペレーションは、標準の HMAC タグを生成します。HMAC KMS キーは、FIPS 140-2 Cryptographic Module Validation Program の認定を受けた AWS KMS ハードウェアセキュリティモジュールで生成されており (中国 (北京) および中国 (寧夏) リージョンを除く)、暗号化されずに AWS KMS 外で使用されることはありません。HMAC KMS キーを使用するには、AWS KMS を呼び出す必要があります。

HMAC を使用して、JSON Web トークン (JWT)、トークン化されたクレジットカード情報、または送信されたパスワードなどのメッセージの信頼性を判断することができます。決定論的なキーを必要とするアプリケーションでは特に、セキュアなキー導出関数 (KDF) としても使用することもできます。

HMAC KMS キーは、ユーザーがキーに設定するアクセスコントロールに従って、キーマテリアルの生成と使用のすべてが AWS KMS で行われることから、アプリケーションソフトウェアからの HMAC よりも多くのメリットを提供します。

> **Tip**
>
> ベストプラクティスでは、HMAC を含めたどの署名メカニズムについても、その有効時間を制限することが推奨されています。そうすることで、アクターが有効性を何度も確立したり、メッセージが置き換えらえた後も長期間有効性を確立したりする攻撃が阻止されます。HMAC タグにタイムスタンプは含まれませんが、トークンまたはメッセージにタイムスタンプを含めて、HMAC を更新するタイミングを検知できるようにすることが可能です。

承認されたユーザーは、AWS アカウント内でHMAC KMS キーを作成、管理、および使用できます。これには、キーの有効化と無効化、エイリアスとタグの設定と変更、および HMAC KMS キーの削除のスケジュールが含まれます。キーポリシー、IAM ポリシー、およびグラントを使用して、KMS キーへのアクセスを制御することもできます。AWS CloudTrail ログで、AWS 内で HMAC KMS キーを使用または管理するすべてのオペレーションを監査することができます。インポートされたキーマテリアルで HMAC KMS キーを作成できます。また、複数のAWS リージョンで同じ HMAC KMS キーのコピーのように機能する HMAC マルチリージョン KMS キーを作成することも可能です。

HMAC KMS キーは、`GenerateMac` および `VerifyMac` 暗号化オペレーションのみをサポートします。HMAC KMS キーを使用してデータの暗号化やメッセージの署名を行ったり、HMAC オペレーションで他のタイプの KMS キーを使用したりすることはできません。`GenerateMac` オペレーションを使用するときは、ユーザーが最大 4,096 バイトのメッセージ、HMAC KMS キー、および HMAC キー仕様との互換性がある MAC アルゴリズムを指定して、`GenerateMac` が HMAC タグを計算します。HMAC タグを検証するには、HMAC タグ、同一のメッセージ、HMAC KMS キー、および 元の HMAC タグを計算するために `GenerateMac` が使用した MAC アルゴリズムを提供する必要があります。`VerifyMac` オペレーションは HMAC タグを計算し、それが提供された HMAC タグと同一であることを検証します。入力と計算された HMAC タグが同一ではない場合は、検証が失敗します。

HMAC KMS キーは自動キーローテーションをサポートしていないため、カスタムキーストアで HMAC KMS キーを作成することはできません。

AWS のサービス内のデータを暗号化するために KMS キーを作成している場合は、対称暗号化キーを使用します。HMAC KMS キーを使用することはできません。

リージョン

HMAC KMS キーは、AWS KMS がサポートするすべての AWS リージョン でサポートされています。

詳細はこちら

- KMS キーのタイプを選択する方法については、「KMS キータイプの選択」を参照してください。
- 各タイプの KMS キーでサポートされる AWS KMS API オペレーションを比較する表については、「キータイプリファレンス」を参照してください。
- マルチリージョン HMAC KMS キーの作成については、「AWS KMS のマルチリージョンキー」を参照してください。
- AWS KMS コンソールが HMAC KMS キーに設定するデフォルトのキーポリシーの違いを調べるには、「the section called “AWS サービスで KMS キーを使用することをキーユーザーに許可する”」を参照してください。
- HMAC KMS キーの料金については、「AWS Key Management Service の料金」を参照してください。
- HMAC KMS キーに適用されるクォータについては、「リソースクォータ」および「クォータのリクエスト」を参照してください。
- HMAC KMS キーの削除については、「AWS KMS keys を削除する」を参照してください。
- JSON Web トークンを作成するための HMAC の使用について学ぶには、AWS セキュリティブログの「How to protect HMACs inside AWS KMS」(内で HMAC を保護する方法) を参照してください。
- ポッドキャストを聴く: AWS 公式ポッドキャストにてAWS Key Management Service が HMAC を導入。

トピック

- HMAC KMS キーの表示

HMAC KMS キーの主な仕様

AWS KMS は、さまざまな長さの対称 HMAC キーをサポートします。選択するキー仕様は、セキュリティ、規制、またはビジネス要件に応じて異なります。キーの長さによって、 GenerateMacおよび VerifyMacオペレーションで使用される MAC アルゴリズムが決まります。一般に、キーが長いほど安全性が高くなります。ユースケースにとって実用的な最も長いキーを使用してください。

HMAC キーの仕様	MAC アルゴリズム
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

HMAC KMS キーの作成

HMAC KMS キーは、AWS KMS コンソール、CreateKey API、または AWS CloudFormation テンプレートを使用して作成することができます。

AWS KMS は、HMAC KMS キーに対して複数のキー仕様をサポートします。ユーザーが選択するキー仕様は、規制、セキュリティ、またはビジネス要件に応じて決定される場合があります。一般に、長いキーはブルートフォース攻撃に対する耐性が高くなります。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

AWS のサービス内のデータを暗号化するために KMS キーを作成している場合は、対称暗号化 KMS キーを使用します。AWS KMS と統合された AWS のサービスは、非対称 KMS キー、または HMAC

KMS キーをサポートしません。対称暗号化 KMS キーの作成方法については、「キーの作成」を参照してください。

詳細はこちら

- 作成する KMS キーの種類を判断するには、「KMS キータイプの選択」を参照してください。
- このトピックで説明されている手順を使用して、マルチリージョンのプライマリ HMAC KMS キーを作成できます。マルチリージョン HMAC キーをレプリケートするには、「the section called “レプリカキーを作成する”」を参照してください。
- KMS キーの作成に必要なアクセス許可については、KMS キーを作成するためのアクセス許可 を参照してください。
- AWS CloudFormation テンプレートを使用して HMAC KMS キーを作成する方法については、AWS CloudFormation「 ユーザーガイド」のAWS::KMS::Key「」を参照してください。

トピック

- HMAC KMS キーの作成 (コンソール)
- HMAC KMS キーの作成 (AWS KMS API)

HMAC KMS キーの作成 (コンソール)

AWS Management Console を使用して、HMAC KMS キー を作成することができます。HMAC KMS キーは、キーの用途が [Generate and verify MAC] (MAC の生成と検証) である対称キーです。マルチリージョンの HMAC キーを作成することもできます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. [Create key] (キーの作成) を選択します。
5. [キーの種類] で、[対称] を選択します。

   HMAC KMS キーは対称です。同じキーを使用して、HMAC タグの生成と検証を行います。
6. [Key usage] (キーの使用) には、[Generate and verify MAC] (MAC の生成と検証) を選択します。

   MAC の生成と検証は、HMAC KMS キーに対して唯一有効なキーの用途です。

> **Note**
>
> 対称キーに対する [Key usage] (キーの使用) は、選択されたリージョンで HMAC KMS キーがサポートされている場合にのみ表示されます。

7. HMAC KMS キーの仕様 ([Key spec] (キーの仕様)) を選択します。

   選択するキーの仕様は、規制、セキュリティ、またはビジネス要件に応じて決定できます。一般に、キーが長いほど安全性が高くなります。

8. マルチリージョンのプライマリ HMAC キーを作成するには、[Advanced options] (詳細オプション) で [Multi-Region key] (マルチリージョンキー) を選択します。この KMS キーに定義する共有プロパティ (キーのタイプとキーの用途など) は、そのレプリカキーと共有されます。詳細については、「マルチリージョンキーを作成する」を参照してください。

   この手順を使用してレプリカキーを作成することはできません。マルチリージョンのレプリカ HMAC キーを作成するには、レプリカキーを作成するための手順に従ってください。

9. [次へ] をクリックします。

10. KMS キーのエイリアスを入力します。エイリアス名の先頭を **aws/** にすることはできません。この **aws/** プレフィックスは、アカウント内の AWS マネージドキー を表すために、Amazon Web Services によって予約されます。

    KMS キーを HMAC キーとして識別するエイリアス (`HMAC/test-key` など) の使用をお勧めします。これは、タグとエイリアスによるキーのソートとフィルタリングは可能でも、キーの仕様や用途によるキーのソートとフィルタリングは可能ではない AWS KMS コンソールでの HMAC キーの識別を容易にします。

    エイリアスは AWS Management Console で KMS キーを作成するときに必要です。CreateKey オペレーションを使用する場合、エイリアスを指定することはできませんが、コンソールまたは CreateAlias オペレーションを使用して既存の KMS キーのエイリアスを作成できます。詳細については、「エイリアスの使用」を参照してください。

11. (オプション) KMS キーの説明を入力します。

    保護する予定のデータタイプ、または KMS キーで使用する予定のアプリケーションを表す説明を入力します。

    今すぐ説明を追加するか、キーの状態が `Pending Deletion` または `Pending Replica Deletion` でない限り、後でいつでも更新できます。既存のカスタマーマネージドキーの説

明を追加、変更、または削除するには、 で説明を編集するAWS Management Consoleか、UpdateKeyDescriptionオペレーションを使用します。

12. (オプション) タグキーとオプションのタグ値を入力します。KMS キーに複数のタグを追加するには、[Add tag] (タグを追加) を選択します。

    `Type=HMAC` など、キーを HMAC キーとして識別するタグの追加を検討してください。これは、タグとエイリアスによるキーのソートとフィルタリングは可能でも、キーの仕様や用途によるキーのソートとフィルタリングは可能ではない AWS KMS コンソールでの HMAC キーの識別を容易にします。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

13. [次へ] をクリックします。

14. KMS キーを管理できる IAM ユーザーとロールを選択します。

> **Note**
>
> このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシパルに KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

15. (オプション) 選択した IAM ユーザーとロールがこの KMS キーを削除しないようにするには、ページの下部にある [Key deletion] (キーの削除) セクションで、[Allow key administrators to delete this key] (キー管理者にこのキーの削除を許可する) のチェックボックスをオフにします。

16. [次へ] をクリックします。

17. 暗号化オペレーションで KMS キーを使用できる IAM ユーザーとロールを選択します。

> **Note**
>
> このキーポリシーにより、AWS アカウント はこの KMS キーを完全に制御できるようになります。これにより、アカウント管理者は IAM ポリシーを使用して、他のプリンシパルに暗号化オペレーションで KMS キーを管理する許可を付与できます。詳細については、「the section called “デフォルトのキーポリシー”」を参照してください。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

18. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID 番号を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

> **Note**
>
> 外部アカウントでプリンシパルが KMS キーを使用できるようにするには、外部アカウントの管理者が、これらのアクセス許可を付与する IAM ポリシーを作成する必要があります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

19. [次へ] を選択します。
20. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。
21. [Finish] (完了) を選択して HMAC KMS キーを作成します。

HMAC KMS キーの作成 (AWS KMS API)

CreateKey オペレーションを使用して HMAC KMS キーを作成できます。以下の例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

HMAC KMS キーを作成するときは、KMS キーのタイプを決定する KeySpec パラメータを指定する必要があります。また、GENERATE_VERIFY_MAC が HMAC キーに唯一有効なキーの用途であっても、GENERATE_VERIFY_MAC の KeyUsage 値を指定する必要があります。マルチリージョンの

HMAC KMS キーを作成するには、値が `true` の `MultiRegion` パラメータを追加します。KMS キー作成後にこれらのプロパティを変更することはできません。

`CreateKey` オペレーションではエイリアスを指定することはできませんが、 CreateAliasオペレーションを使用して新しい KMS キーのエイリアスを作成できます。KMS キーを HMAC キーとして識別するエイリアス (`HMAC/test-key` など) の使用をお勧めします。これは、エイリアスによるキーのソートとフィルタリングは可能でも、キーの仕様や用途によるキーのソートとフィルタリングは可能ではない AWS KMS コンソールでの HMAC キーの識別を容易にします。

HMAC キーがサポートされていていない AWS リージョン で HMAC KMS キーを作成しようとすると、`CreateKey` オペレーションが `UnsupportedOperationException` を返します。

以下の例では、`CreateKey` オペレーションを使用して 512 ビットの HMAC KMS キーを作成します。

```
$ aws kms create-key --key-spec HMAC_512 --key-usage GENERATE_VERIFY_MAC
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1669973196.214,
        "MultiRegion": false,
        "KeySpec": "HMAC_512",
        "CustomerMasterKeySpec": "HMAC_512",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_512"
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

HMAC KMS キーへのアクセスの制御

HMAC KMS キーへのアクセスを制御するには、すべての KMS キーに必要とされるキーポリシーを使用します。IAM ポリシーと権限を使用することも可能です。

AWS KMS コンソールで作成された HMAC キー用のデフォルトキーポリシーは、GenerateMac および VerifyMac オペレーションを呼び出す許可をキーのユーザーに付与します。ただし、これには AWS のサービスでの権限の使用向けに設計されたキーポリシーステートメントは含まれていません。CreateKey オペレーションを使用して HMAC キーを作成する場合は、キーポリシーまたは IAM ポリシーでこれらの許可を指定する必要があります。

AWS グローバル条件キー、および AWS KMS 条件キーを使用して、HMAC キーに対する許可を絞り込み、制限することができます。例えば、kms:ResourceAliases 条件キーを使用して、AWS KMS オペレーションへのアクセスを HMAC キーに関連付けられたエイリアスに基づいて制御することができます。以下の AWS KMS ポリシー条件は、HMAC キーに対するポリシーに役立ちます。

- kms:MacAlgorithm 条件キーを使用して、プリンシパルが GenerateMac および VerifyMac オペレーションを呼び出すときにリクエストできるアルゴリズムを制限します。例えば、リクエスト内の MAC アルゴリズムが HMAC_SHA_384 の場合にのみ、プリンシパルが GenerateMac オペレーションの呼び出せるようにすることができます。
- kms:KeySpec 条件キーを使用して、プリンシパルによる特定タイプの HMAC キーの作成を許可または拒否します。例えば、プリンシパルが HMAC キーのみを作成できるようにするには、 CreateKeyオペレーションを許可しますが、 kms:KeySpec条件を使用して、 キー仕様のHMAC_384キーのみを許可します。

  kms:KeySpec 条件キーを使用することで、キーのキー仕様に基づいて、KMS キーでのその他オペレーションへのアクセスを制御することもできます。例えば、プリンシパルが HMAC_256 キー仕様を持つ KMS キーのみでキーの削除のスケジュールとキャンセルを実行できるようにすることが可能です。
- kms:KeyUsage 条件キーを使用して、プリンシパルによる HMAC キーの作成を許可または拒否します。例えば、プリンシパルが HMAC キーのみを作成できるようにするには、CreateKeyオペレーションを許可しますが、 kms:KeyUsage条件を使用して、キーの使用を持つGENERATE_VERIFY_MACキーのみを許可します。

  kms:KeyUsage 条件キーを使用して、キーの用途に基づいて KMS キーでのその他オペレーションへのアクセスを制御することもできます。例えば、用途が GENERATE_VERIFY_MAC である KMS キーのみでの有効化と無効化をプリンシパルに許可することができます。

権限オペレーションである GenerateMac および VerifyMac オペレーションの権限を作成することもできます。ただし、HMAC キーの権限で暗号化コンテキストの権限の制約を使用することはできません。HMAC タグ形式は、暗号化コンテキスト値をサポートしません。

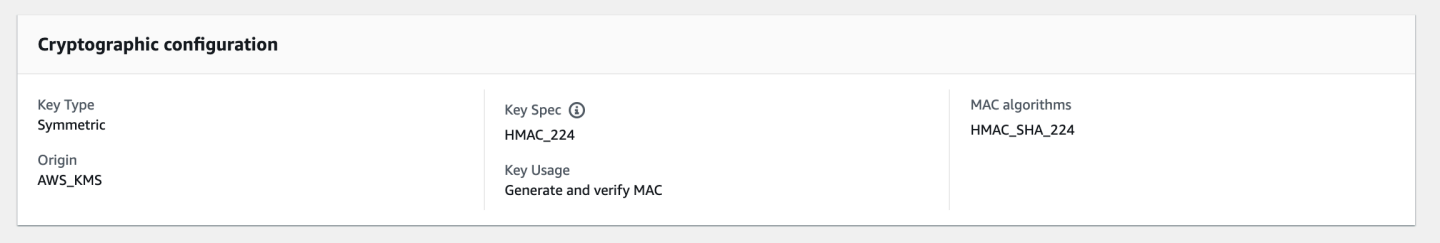
HMAC KMS キーの表示

HMAC KMS キーは、AWS KMS コンソール、または DescribeKey API を使用して表示できます。HMAC KMS キーの使用は、 AWS CloudTrailログおよび Amazon CloudWatchでモニタリングできます。KMS キーの表示に関する基本的な手順については、「キーの表示」を参照してください。

HMAC KMS キーは、HMAC で始まるキーの仕様、または常に [Generate and verify MAC] (MAC の生成と検証)(GENERATE_VERIFY_MAC) であるキーの用途によって、他のタイプの KMS キーと区別することができます。

HMAC KMS キーは、AWS KMS コンソールの [Customer managed keys] (カスタマー管理型のキー) ページにある表に記載されています。ただし、キーの仕様や用途で KMS キーをソートまたはフィルタリングすることはできません。HMAC キーを見つけやすくするには、それらに固有のエイリアスまたはタグを割り当てます。そうすることで、エイリアスまたはタグによるソートまたはフィルタリングが可能になります。

HMAC KMS キーに関するキーの詳細ページの [Cryptographic configuration] (暗号化設定) タブで、設定の詳細を確認できます。



AWS KMS のマルチリージョンキー

AWS KMS では、複数のリージョンで同じキーのように相互使用ができる、異なる AWS リージョン の AWS KMS keys であるマルチリージョンキーをサポートします。関連するマルチリージョンキーセットごとに、同じキーマテリアルおよびキー ID があるため、1つの AWS リージョン でデータを暗号化し、再暗号化や AWS KMS へのクロスリージョン呼び出しを行うことなく、異なる AWS リージョン で復号できます。

すべての KMS キーと同様に、マルチリージョンキーは AWS KMS を暗号化されないままにしません。暗号化または署名用の対称または非対称のマルチリージョンキーを作成する、HMAC タグの生成と検証用の HMAC マルチリージョンキーを作成する、および AWS KMS が生成するキーマテリアル、またはインポートされたキーマテリアルを持つマルチリージョンキーを作成することができます。エイリアスおよびタグの作成、キーポリシーとグラントの設定、有効化/無効化の選択など、各

マルチリージョンキーを個別に管理する必要があります。単一リージョンキーで実行できるすべての暗号化オペレーションで、マルチリージョンキーを使用できます。

マルチリージョンキーは、多くの一般的なデータセキュリティシナリオに対応する、柔軟で強力なソリューションです。

ディザスタリカバリ

バックアップおよびリカバリのアーキテクチャでは、マルチリージョンキーを使用することで、AWS リージョン 停止のイベント時でも、暗号化されたデータを中断することなく処理できます。バックアップリージョンで保持されるデータはバックアップリージョンで復号し、バックアップリージョンで新たに暗号化されたデータは、そのリージョンの復元時にプライマリリージョンで復号することができます。

グローバルなデータ管理

グローバルに展開されるビジネスには、グローバルに配信され、AWS リージョン 全体で一貫して利用可能なデータが必要です。データが存在するすべてのリージョンでマルチリージョンキーを作成し、クロスリージョン呼び出しのレイテンシーや、各リージョンで異なるキーのデータの再暗号化に掛かるコストなしで、単一リージョンキーであるかのようにキーを使用できます。

配信署名アプリケーション

クロスリージョン署名機能を必要とするアプリケーションでは、マルチリージョンの非対称署名キーを使用して、異なる AWS リージョン で同一のデジタル署名を、一貫して繰り返し生成することができます。

単一のグローバルトラストストア (単一のルート認証機関 (CA))、およびルート CA によって署名されたリージョンの中間 CA で証明書チェーンを使用する場合、マルチリージョンキーは不要です。ただし、アプリケーション署名などの中間 CA がシステムでサポートされない場合は、マルチリージョンキーを使用して、リージョンの認定に一貫性を持たせることができます。

複数のリージョンにまたがるアクティブ-アクティブアプリケーション

一部のワークロードとアプリケーションは、アクティブ-アクティブアーキテクチャで複数のリージョンにまたがることができます。これらのアプリケーションでは、マルチリージョンキーを使用して、リージョンの境界を越えて移動する可能性のあるデータに対する暗号化と復号の同時オペレーションに同じキーマテリアルを提供し、複雑さを軽減できます。

マルチリージョンキーは、クライアント側の暗号化ライブラリ (AWS Encryption SDK、DynamoDB 暗号化クライアント、Amazon S3 クライアント側暗号化) などで使用できます。Amazon DynamoDB グローバルテーブルおよび DynamoDB 暗号化クライアントでマルチリージョンキーを

使用する例については、AWS セキュリティブログの Encrypt global data client-side with AWS KMS multi-Region keys を参照してください。

保管時の暗号化またはデジタル署名用の AWS KMS と統合された AWS のサービスでは、現在、マルチリージョンキーを単一リージョンキーのように扱っています。リージョン間で移動されたデータを再ラップまたは再暗号化する場合があります。例えば、Amazon S3 クロスリージョンレプリケーションでは、マルチリージョンキーで保護されたオブジェクトをレプリケートする場合でも、コピー先リージョンの KMS キーでデータを復号および再暗号化します。

マルチリージョンキーはグローバルではありません。マルチリージョンのプライマリキーを作成し、そのキーを AWS パーティション内で選択するリージョンにレプリケートします。次に、各リージョンでマルチリージョンキーを個別に管理します。AWS または AWS KMS のどちらも、ユーザーの代わりにマルチリージョンキーを任意のリージョンに自動的に作成、またはレプリケートしません。AWS マネージドキー、アカウントで AWS サービスが作成する KMS キーは、常に単一リージョンキーです。

既存の単一リージョンキーをマルチリージョンキーに変換することはできません。この設計により、既存の単一リージョンキーで保護されているすべてのデータが、同じデータ常駐プロパティとデータ主権プロパティを維持できます。

ほとんどのデータセキュリティニーズに対して、リージョナルリソースのリージョン分離と耐障害性により、スタンダードな AWS KMS 単一リージョンキーは最適なソリューションです。ただし、複数のリージョンにまたがるクライアント側のアプリケーションでデータを暗号化または署名する必要がある場合は、マルチリージョンキーがソリューションとなることがあります。

リージョン

マルチリージョンキーは、中国 (北京) および中国 (寧夏) を除く、AWS KMS がサポートしているすべての AWS リージョン でサポートされます。

料金とクォータ

関連するマルチリージョンキーのセットに含まれるすべてのキーは、料金およびクォータに関して、1 つの KMS キーとしてカウントされます。AWS KMS クォータは、アカウントのリージョンごとに個別に計算されます。各リージョンのマルチリージョンキーの使用と管理は、そのリージョンのクォータでカウントされます。

サポートされる KMS キータイプ

次の種類のマルチリージョン KMS キーを作成できます。

- 対称暗号化 KMS キー
- 非対称 KMS キー
- HMAC KMS キー
- インポートされたキーマテリアルを持つ KMS キー

カスタムキーストアでマルチリージョンキーを作成することはできません。

トピック

- マルチリージョンキーへのアクセスを制御する
- マルチリージョンキーを作成する
- マルチリージョンキーを表示する
- マルチリージョンのキーを管理する
- キーマテリアルをマルチリージョンキーにインポートする
- マルチリージョンキーを削除する

マルチリージョンキーのセキュリティに関する考慮事項

AWS KMS マルチリージョンキーは、必要なときにだけ使用します。マルチリージョンキーは、暗号化されたデータを AWS リージョン の間で移動するワークロード、またはクロスリージョンアクセスが必要なワークロードに柔軟でスケーラブルなソリューションを提供します。保護されたデータを、リージョンを超えて共有、移動、バックアップする必要がある場合、または異なるリージョンで同一のデジタル署名を作成する必要がある場合は、マルチリージョンキーを検討します。

ただし、マルチリージョンキーを作成するプロセスでは、キーマテリアルは AWS KMS 内の AWS リージョン の境界を越えて移動します。マルチリージョンキーによって生成される暗号文は、複数の地理的位置にある複数の関連キーによって復号される可能性があります。地域的に孤立したサービスやリソースにも大きな利点があります。各 AWS リージョン は他のリージョンから分離され、独立しています。リージョンでは耐障害性や安定性が提供され、レイテンシーを低減することもできます。これにより、別のリージョンの障害の影響を受けずに利用できる冗長リソースを作成できます。AWS KMS では、すべての暗号文を 1 つのキーのみで復号できます。

マルチリージョンキーは、セキュリティに関する新しい考慮事項も提起します。

- マルチリージョンキーでは、アクセス制御とデータセキュリティポリシーの適用がより複雑です。複数の独立したリージョン全体で、ポリシーがキーで一貫して監査されることを確認する必要があ

ります。また、個別のキーに依存する代わりに、ポリシーを使用して境界を適用する必要があります。

例えば、あるリージョンの給与チームが別のリージョンの給与データを読み取れないようにするために、データにポリシー条件を設定する必要があります。また、あるリージョンのマルチリージョンキーが 1 つのテナントのデータを保護し、別のリージョンの関連するマルチリージョンキーが別のテナントのデータを保護するというシナリオを防ぐには、アクセス制御を使用する必要があります。

- リージョン全体のキーの監査もより複雑です。マルチリージョンキーでは、複数のリージョン全体の監査アクティビティを調べて調整し、保護されるデータのキーアクティビティを完全に理解する必要があります。
- データ常駐に関するコンプライアンスは、より複雑になる可能性があります。孤立したリージョンでは、データ常駐とデータ主権のコンプライアンスを確保できます。特定のリージョンの KMS キーは、そのリージョン内の機密データのみを復号できます。あるリージョンで暗号化されたデータは完全に保護され、他のリージョンではアクセスできません。

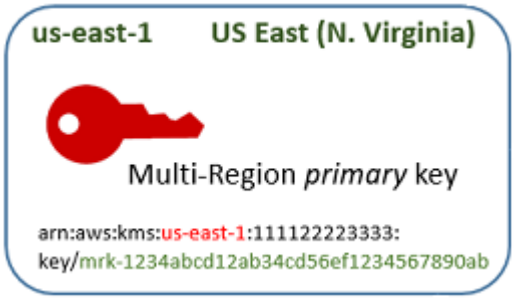
  マルチリージョンキーを使用してデータの常駐性とデータ主権を検証するには、アクセスポリシーを実装し、複数のリージョンに AWS CloudTrail イベントをコンパイルします。

マルチリージョンキーのアクセスコントロールを管理しやすくするために、マルチリージョンキー (kms:ReplicateKey) をレプリケートするアクセス許可は、キーを作成する標準アクセス許可 (kms:CreateKey) とは別のものです。また、AWS KMS では、マルチリージョンキーに関する複数のポリシー条件 (kms:MultiRegion: マルチリージョンキーを作成、使用、管理するアクセス許可を許可または拒否します。kms:ReplicaRegion: マルチリージョンキーのレプリケーション先となるリージョンを制限します) をサポートします。詳細については、「マルチリージョンキーへのアクセスを制御する」を参照してください。

マルチリージョンキーの仕組み

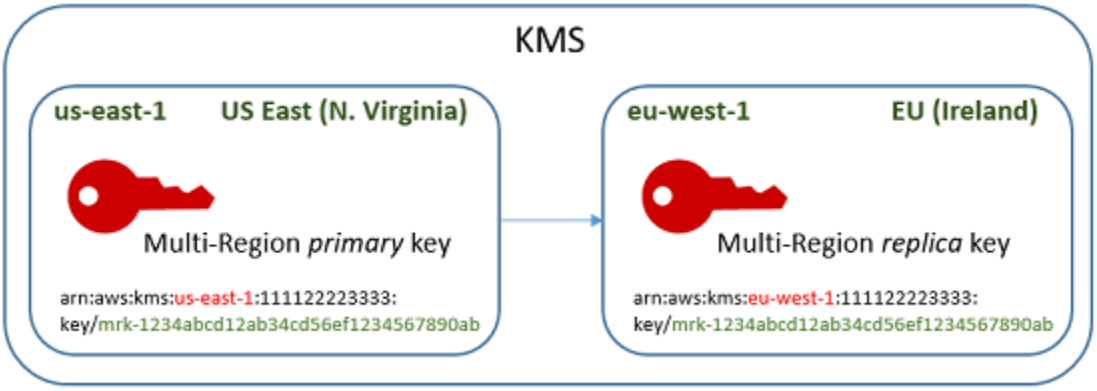
まず、AWS KMS がサポートする AWS リージョン (米国東部 (バージニア北部) など) で、対称または非対称のマルチリージョンのプライマリキーを作成します。キーを単一リージョンにするか、マルチリージョンにするかは、作成時にのみ決定できます。このプロパティは後で変更できません。KMS キーと同様に、マルチリージョンキーのキーポリシーを設定することで、グラントを作成したり、分類と認可用のエイリアスとタグを追加したりできます。(これらは、他のキーと共有または同期されない独立したプロパティです)。暗号化または署名の暗号化オペレーションで、マルチリージョンのプライマリキーを使用できます。

マルチリージョンのプライマリキーは、コンソールで作成するか、 パラメータを に設定して API を使用して作成できますtrue。 AWS KMS CreateKey MultiRegionマルチリージョンキーには mrk- で始まる固有のキー ID があります。mrk- プレフィックスを使用して、プログラムで MRK を識別できます。



選択すると、マルチリージョンのプライマリキーを同じ AWS パーティション (欧州 (アイルランド) など) の 1 つ以上の異なる AWS リージョン にレプリケートできます。これにより、AWS KMS は指定されたリージョンで、同じキー ID と他の共有プロパティを使用して、プライマリキーとしてレプリカキーを作成します。次に、キーマテリアルをリージョンの境界を越えて安全に転送し、すべて AWS KMS 内で、コピー先リージョンの新しい KMS キーに関連付けます。結果として、2 つの関連するマルチリージョンキーが作成されます (プライマリキーとレプリカキー)。これらは相互に使用することができます。

マルチリージョンのレプリカキーは、コンソールで作成することも、 API を使用して作成することもできます。 AWS KMS ReplicateKey



作成されたマルチリージョンレプリカキーは、完全に機能する KMS キーで、プライマリキーと同じ共有プロパティを備えています。それ以外の点では、独自の説明、キーポリシー、グラント、エイリアス、タグを持つ独立した KMS キーです。マルチリージョンキーを有効または無効にしても、関連するマルチリージョンキーには影響しません。プライマリキーとレプリカキーは、暗号化オペレーションで個別に使用することも、連携させて使用することもできます。例えば、米国東部 (バージニ

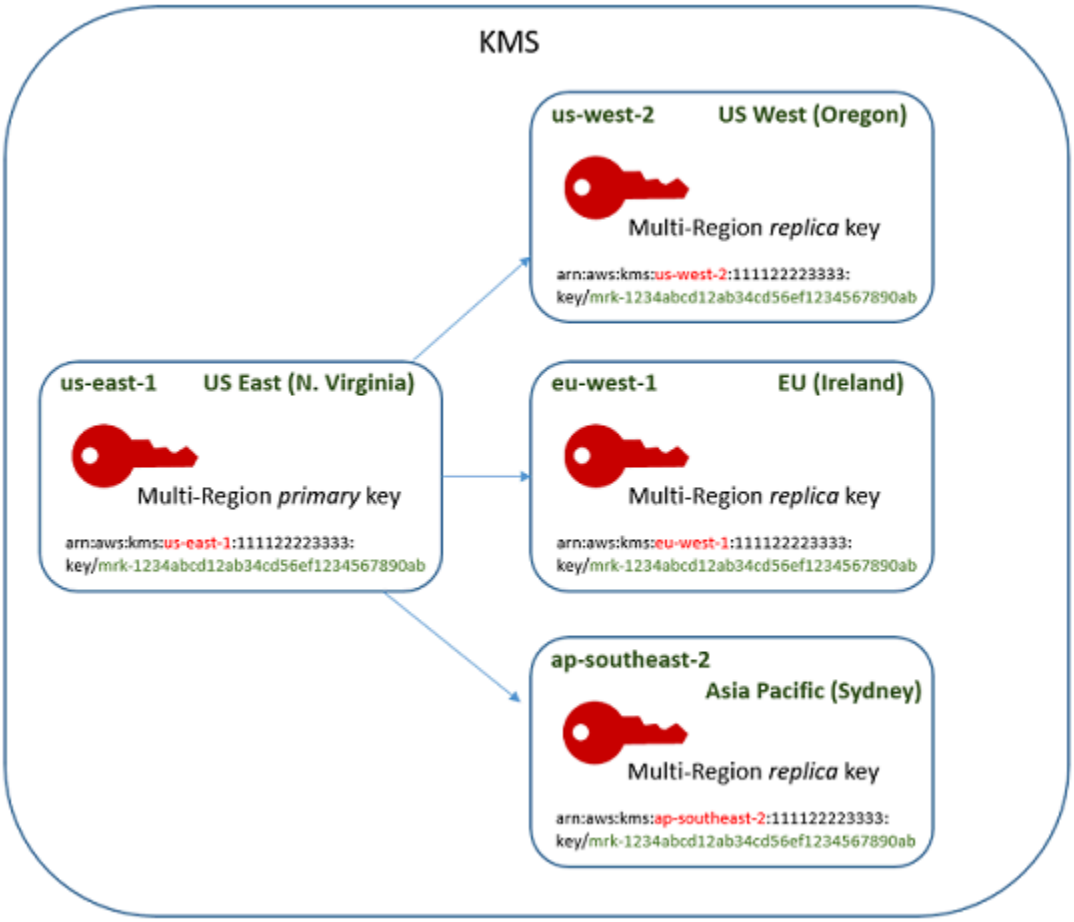
ア北部) リージョンのプライマリキーを使用してデータを暗号化し、欧州 (アイルランド) リージョンにデータを移動し、レプリカキーを使用してデータを復号できます。

関連するマルチリージョンキーは同じキー ID を持ちます。キー ARN (Amazon リソースネーム) は、リージョンフィールドでのみ異なります。例えば、マルチリージョンのプライマリキーとレプリカキーには、次のキー ARN があります。キー ID (キー ARN の最後の要素) は同一です。両方のキーには、mrk- で始まる、マルチリージョンキー固有のキー ID があります。

```
Primary key: arn:aws:kms:us-
east-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
Replica key: arn:aws:kms:eu-
west-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
```

相互運用性のために、同じキー ID が必要です。暗号化する場合、AWS KMS は、KMS キーのキー ID を暗号文にバインドします。そのため、暗号文は当該 KMS キーまたは同じキー ID を持つ KMS キーでのみ解読できます。この機能により、関連するマルチリージョンキーが認識しやすくなり、相互に使用しやすくなります。例えば、アプリケーションで使用する場合は、共有キー ID で関連するマルチリージョンキーを参照できます。次に、必要に応じて、リージョンまたは ARN を指定して、それらを区別します。

データのニーズの変化に応じて、プライマリキーを米国西部 (オレゴン) およびアジアパシフィック (シドニー)など、同じパーティションの他の AWS リージョン にレプリケートできます。結果は、次の図表に示された、同じキーマテリアルとキー ID を持つ 4 つの関連するマルチリージョンキーとなります。キーは個別に管理します。キーは独立して使用することも、連携させて使用することもできます。例えば、アジアパシフィック (シドニー) でレプリカキーを使用してデータを暗号化し、データを米国西部 (オレゴン) に移動して、米国西部 (オレゴン) でレプリカキーを使用して復号できます。



マルチリージョンキーに関するその他の考慮事項は次のとおりです。

共有プロパティの同期 — マルチリージョンキーの共有プロパティが変更されると、AWS KMS は自動的にプライマリキーからすべてのレプリカキーに変更を同期させます。共有プロパティの同期をリクエストまたは強制することはできません。AWS KMS はすべての変更を検出して同期します。ただし、 CloudTrail ログの SynchronizeMultiRegionKeyイベントを使用して同期を監査できます。

例えば、対称マルチリージョンのプライマリキーで自動キーローテーションを有効にすると、AWS KMS は、その設定をすべてのレプリカキーにコピーします。キーマテリアルをローテーションすると、関連するすべてのマルチリージョンキー間でローテーションが同期されます。これにより、マルチリージョンキーは引き続き現在と同じキーマテリアルを持ち、古いバージョンのキーマテリアルすべてにアクセスできます。新しいレプリカキーを作成すると、そのキーマテリアルは、関連するすべてのマルチリージョンキーの現在のキーマテリアルと同じになり、以前のバージョンのキーマテリアルすべてにアクセスできます。詳細については、「マルチリージョンキーをローテーションする」を参照してください。

プライマリキーの変更 — マルチリージョンキーのすべてのセットには、プライマリキーが 1 つだけ必要です。プライマリキーはレプリケートできる唯一のキーです。また、レプリカキーの共有プロパティのソースでもあります。ただし、プライマリキーをレプリカに変更し、レプリカキーの 1 つをプライマリに昇格させることができます。これにより、特定のリージョンから複数リージョンのプライマリキーを削除したり、プロジェクト管理者の近くにあるリージョンでプライマリキーを検索したりできます。詳細については、「プライマリリージョンを更新する」を参照してください。

マルチリージョンキーの削除 — すべての KMS キーと同様に、AWS KMS がマルチリージョンキーを削除する前に、削除をスケジュールする必要があります。キーが削除保留中の間は、暗号化オペレーションでキーを使用することはできません。ただし、すべてのレプリカキーが削除されるまで、AWS KMS はマルチリージョンのプライマリキーを削除しません。詳細については、「マルチリージョンキーを削除する」を参照してください。

概念

マルチリージョンキーでは、次の条件と概念を使用します。

マルチリージョンキー

マルチリージョンキーは、異なる AWS リージョン で同じキー ID とキーマテリアル (およびその他の共有プロパティ) を持つ KMS キーのセットの 1 つです。各マルチリージョンキーは、完全に機能する KMS キーで、関連するマルチリージョンキーとは完全に独立して使用できます。すべての関連するマルチリージョンキーは同じキー ID とキーマテリアルを持ち、相互運用可能です。つまり、すべての AWS リージョン のすべての関連するマルチリージョンキーは、他の関連するすべてのマルチリージョンキーによって暗号化された暗号文を復号できます。

KMS キーの作成時に、KMS キーのマルチリージョンのプロパティを設定します。既存のキーでマルチリージョンプロパティを変更することはできません。単一リージョンキーをマルチリージョンキーに変換したり、マルチリージョンキーを単一リージョンキーに変換したりすることはできません。既存のワークロードをマルチリージョンシナリオに移動するには、データを再暗号化するか、新しいマルチリージョンキーを使用して新しい署名を作成する必要があります。

マルチリージョンキーは、対称または非対称で、AWS KMS キーマテリアルまたはインポートされたキーマテリアルを使用できます。カスタムキーストアでマルチリージョンキーを作成することはできません。

関連するマルチリージョンキーのセットには、常に 1 つだけプライマリキーがあります。他の AWS リージョン で、そのプライマリキーのレプリカキーを作成できます。プライマリリージョンを更新すると、プライマリキーがレプリカキーに変更され、指定されたレプリカキーがプライマリキーに変

更されます。ただし、AWS リージョン ごとに保持できるプライマリキーまたはレプリカキーは 1 つだけです。リージョンはすべて、同じ AWS パーティションである必要があります。

関連するマルチリージョンキーの複数のセットを、同じまたは異なる AWS リージョン で持つことができます。関連するマルチリージョンキーは相互運用可能ですが、関連しないマルチリージョンキーは相互運用できません。

プライマリキー

マルチリージョンのプライマリキーは KMS キーであり、同じパーティションの他の AWS リージョン にレプリケートできます。マルチリージョンキーの各セットには、プライマリキーが 1 つしかありません。

プライマリキーは、次の点でレプリカキーとは異なります。

- プライマリキーのみがレプリケーション可能です。
- プライマリキーは、レプリカキー (キーマテリアルとキー ID を含む) の共有プロパティのソースです。
- 自動キーローテーションは、プライマリキーでのみ有効または無効にできます。
- プライマリキーの削除をいつでもスケジュールすることができます。ただし、すべてのレプリカキーが削除されるまで、AWS KMS はプライマリキーを削除しません。

プライマリキーとレプリカキーは、暗号化プロパティにおいて違いはありません。プライマリキーとそのレプリカキーは、同じ意味で使用することができます。

プライマリキーをレプリケートする必要はありません。プライマリキーは、KMS キーと同じように使用し、有用であればレプリケートすることができます。ただし、マルチリージョンキーには単一リージョンキーとは異なるセキュリティプロパティがあるため、プライマリキーをコレプリケートする場合にのみ、マルチリージョンキーを作成することをお勧めします。

レプリカキー

マルチリージョンのレプリカキーは、プライマリキーおよび関連するレプリカキーと同じキー IDとキーマテリアルを持ち、異なる AWS リージョン に存在する KMS キーです。

レプリカキーは、固有のキーポリシー、グラント、エイリアス、タグ、およびその他のプロパティを持つ、完全に機能する KMS キーです。レプリカキーは、プライマリキーまたは他のキーのコピーまたはポインタではありません。プライマリキーと関連するすべてのレプリカキーが無効になっている

場合でも、レプリカキーを使用できます。また、レプリカキーをプライマリキーに変換し、プライマリキーをレプリカキーに変換することもできます。レプリカキーが作成されると、レプリカキーはそのプライマリキーにキーローテーションおよびプライマリリージョンの更新のみを依存します。

プライマリキーとレプリカキーは、暗号化プロパティにおいて違いはありません。プライマリキーとそのレプリカキーは、同じ意味で使用することができます。プライマリキーまたはレプリカキーで暗号化されたデータは、同じキー、または関連する任意のプライマリキーまたはレプリカキーで復号できます。

レプリケーション

マルチリージョンのプライマリキーを、同じパーティションの別の AWS リージョン にレプリケートできます これにより、AWS KMS は指定されたリージョンで、プライマリキーと同一のキー ID およびその他の共有プロパティを持つマルチリージョンレプリカキーを作成します。次に、キーマテリアルをリージョンの境界を越えて安全に転送し、すべて AWS KMS 内で、新しいレプリカキーに関連付けます。

共有プロパティ

共有プロパティは、レプリカキーと共有するマルチリージョンのプライマリキーのプロパティです。AWS KMS は、プライマリキーと同じ共有プロパティ値を持つレプリカキーを作成します。次に、プライマリキーの共有プロパティ値をレプリカキーに定期的に同期します。レプリカキーでは、これらのプロパティを設定できません。

以下は、マルチリージョンキーの共有プロパティです。

- キー ID — (キー ARN の Region 要素が異なります)。
- キーマテリアル
- キーマテリアルのオリジン
- キー仕様および暗号化アルゴリズム
- キーの用途
- 自動キーローテーション — 自動キーローテーションは、プライマリキーでのみ有効または無効にできます。新しいレプリカキーは、共有キーマテリアルのすべてのバージョンで作成されます。詳細については、「マルチリージョンキーをローテーションする」を参照してください。

関連するマルチリージョンキーのプライマリおよびレプリカの指定は、共有プロパティと考えることもできます。新しいレプリカキーの作成時またはプライマリキーの更新時、AWS KMS は、関連する

すべてのマルチリージョンキーに変更を同期します。これらの変更が完了すると、関連するすべてのマルチリージョンキーが、プライマリキーとレプリカキーを正確に一覧表示します。

マルチリージョンキーのその他のプロパティはすべて、独立したプロパティです (説明、キーポリシー、グラント、有効および無効キーステータス、エイリアス、タグを含む)。関連するすべてのマルチリージョンキーでこれらのプロパティに同じ値を設定できますが、独立したプロパティの値を変更すると、AWS KMS は同期しません。

マルチリージョンキーの共有プロパティの同期を追跡できます。AWS CloudTrail ログで、SynchronizeMultiRegionKeyイベントを探します。

マルチリージョンキーへのアクセスを制御する

マルチリージョンキーは、単一リージョンキーを使用するとより複雑な、コンプライアンス、災害対策、バックアップのシナリオで使用できます。ただし、マルチリージョンキーのセキュリティプロパティは単一リージョンキーのセキュリティプロパティとは大きく異なるため、マルチリージョンキーの作成、管理、使用の認可には注意が必要です。

> **Note**
>
> `Resource` フィールドのワイルドカード文字を含む既存の IAM ポリシーステートメントが、単一リージョンキーおよびマルチリージョンキーの両方に適用されるようになりました。単一リージョン KMS キーまたはマルチリージョンキーに制限するには、kms:MultiRegion 条件キーを使用します。

認可ツールを使用して、単一リージョンで十分なシナリオでのマルチリージョンキーの作成および使用を阻止します。プリンシパルが、必要とする AWS リージョン のみにマルチリージョンキーをコピーできるようにします。マルチリージョンキーのアクセス許可を、それらを必要とするプリンシパルおよびタスクに対してのみ付与します。

キーポリシー、IAM ポリシー、権限を使用して、IAM プリンシパルが AWS アカウント でマルチリージョンキーを管理および使用できるようにします。各マルチリージョンキーは、一意のキー ARN とキーポリシーを持つ独立したリソースです。各キーのキーポリシーを確立して維持し、新規および既存の IAM ポリシーが認可戦略を実装していることを確認する必要があります。

トピック

- マルチリージョンキーの認可の原則

- マルチリージョンキー管理者およびユーザーを認可する
- マルチリージョンキーの同期を AWS KMS に認可する

マルチリージョンキーの認可の原則

マルチリージョンキーのキーポリシーと IAM ポリシーを設計するときは、次の原則を考慮します。

- キーポリシー — 各マルチリージョンキーは、固有のキーポリシーを持つ独立した KMS キーリソースです。関連するマルチリージョンキーセットの各キーに、同じ、または異なるキーポリシーを適用できます。キーポリシーはマルチリージョンキーの共有プロパティではありません。AWS KMS は関連するマルチリージョンキー間のキーポリシーをコピーまたは同期しません。

  AWS KMS コンソールでレプリカキーを作成するときに、コンソールは便宜上、プライマリキーの現在のキーポリシーを表示します。このキーポリシーの使用、編集、削除、置き換えを行うことができます。ただし、プライマリキーポリシーを変更せずに受け入れる場合でも、AWS KMS はポリシーを同期しません。例えば、プライマリキーのキーポリシーを変更しても、レプリカキーのキーポリシーは変わりません。

- デフォルトキーポリシー — CreateKeyおよび `ReplicateKey`オペレーションを使用してマルチリージョンキーを作成する場合、リクエストでキーポリシーを指定しない限り、デフォルトキーポリシーが適用されます。これは、単一リージョンキーに適用されるのと同じデフォルトのキーポリシーです。

- IAM ポリシー — すべての KMS キーと同様に、キーポリシーによって許可される場合にのみ、IAM ポリシーを使用してマルチリージョンキーへのアクセスを制御できます。IAM ポリシーは、すべての AWS リージョン にデフォルトで適用されます。ただし、aws:RequestedRegion などの条件キーを使用して、特定のリージョンへのアクセス許可を制限できます。

  プライマリキーおよびレプリカキーを作成するには、キーが作成されたリージョンに適用される、IAM ポリシーの `kms:CreateKey` アクセス許可をプリンシパルに付与する必要があります。

- 権限 — AWS KMS 権限はリージョナルです。各権限は、1 つの KMS キーにアクセス許可を付与します。権限を使用して、マルチリージョンのプライマリキーまたはレプリカキーへのアクセス許可を付与できます。ただし、マルチリージョンキーが関連付けられている場合でも、単一の権限を使用して複数の KMS キーにアクセス許可を付与することはできません。

- キー ARN — 各マルチリージョンキーは一意のキー ARN を持ちます。関連するマルチリージョンキーのキー ARN は、同じパーティション、アカウント、キー ID を持ちますが、リージョンが異なります。

IAM ポリシーステートメントを特定のマルチリージョンキーに適用するには、そのキー ARN またはリージョンを含むキー ARN パターンを使用します。関連するすべてのマルチリージョンキーに IAM ポリシーステートメントを適用するには、次の例に示すように、ARN のリージョン要素でワイルドカード文字 (*) を使用します。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Describe*",
    "kms:List*"
  ],
  "Resource": {
      "arn:aws:kms:*::111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
  }
}
```

ポリシーステートメントを 内のすべてのマルチリージョンキーに適用するにはAWS アカウント、kms:MultiRegion policy 条件または固有のmrk-プレフィックスを含むキー ID パターンを使用できます。

- サービスにリンクされたロール — マルチリージョンのプライマリキーを作成するプリンシパルには、iam:CreateServiceLinkedRole アクセス許可が必要です。

  関連するマルチリージョンキーの共有プロパティを同期するために、AWS KMS は IAM サービスリンクロールを引き受けます。AWS KMS はマルチリージョンのプライマリキーを作成するたびに、AWS アカウント でサービスリンクロールを作成します。(ロールが存在する場合は、AWS KMS は悪影響のないロールを再作成します)。ロールはすべてのリージョンで有効です。AWS KMS がサービスにリンクされたロールを作成 (または再作成) できるようにするには、マルチリージョンのプライマリキーを作成するプリンシパルに iam:CreateServiceLinkedRole アクセス許可が必要です。

マルチリージョンキー管理者およびユーザーを認可する

マルチリージョンキーを作成および管理するプリンシパルには、プライマリリージョンとレプリカリージョンで次のアクセス許可が必要です。

- kms:CreateKey
- kms:ReplicateKey

- kms:UpdatePrimaryRegion
- iam:CreateServiceLinkedRole

プライマリキーを作成する

マルチリージョンのプライマリキーを作成するには、プリンシパルにプライマリキーのリージョンで有効な IAM ポリシーの kms:CreateKey および iam:CreateServiceLinkedRole アクセス許可が必要です。これらのアクセス許可を持つプリンシパルは、アクセス許可が制限されない限り　単一リージョンキーおよびマルチリージョンキーを作成できます。

アクセスiam:CreateServiceLinkedRole許可によりAWS KMS、はAWSServiceRoleForKeyManagementServiceMultiRegionKeysロールを作成して、関連するマルチリージョンキーの共有プロパティを同期できます。

例えば、この IAM ポリシーはプリンシパルに、任意のタイプの KMS キーの作成を許可します。

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action": [
        "kms:CreateKey",
        "iam:CreateServiceLinkedRole"
      ],
      "Effect":"Allow",
      "Resource":"*"
  }
}
```

マルチリージョンのプライマリキーを作成するアクセス許可を許可または拒否するには、kms:MultiRegion 条件キーを使用します。有効な値は、true (マルチリージョンキー) または false (単一リージョンキー) です。例えば、次の IAM ポリシーステートメントでは、kms:MultiRegion 条件キーを持つ Deny アクションを使用して、プリンシパルがマルチリージョンキーを作成しないようにします。

```
{
  "Version": "2012-10-17",
  "Statement":{
      "Action":"kms:CreateKey",
      "Effect":"Deny",
      "Resource":"*",
```

```
      "Condition": {
          "Bool": "kms:MultiRegion": true
      }
  }
}
```

キーをレプリケートする

マルチリージョンのレプリカキーを作成するには、プリンシパルに次のアクセス許可が必要です。

- kms:ReplicateKey プライマリキーのキーポリシーの アクセス許可。
- kms:CreateKey レプリカキーリージョンで有効な IAM ポリシーの アクセス許可。

これらのアクセス許可を許可する場合は注意が必要です。これにより、プリンシパルは KMS キーと、その使用を認可するキーポリシーを作成できます。kms:ReplicateKey アクセス許可は AWS KMS 内のリージョンの境界を越えるキーマテリアルの転送も認可します。

マルチリージョンキーAWS リージョンをレプリケートできる を制限するには、kms:ReplicaRegion 条件キーを使用します。これは、kms:ReplicateKey アクセス許可のみを制限します。それ以外には影響を与えません。例えば、次のキーポリシーは、指定されたリージョンでのみプリンシパルにこのプライマリキーのレプリケーションを許可します。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

プライマリリージョンを更新する

認可されたプリンシパルは、レプリカキーをプライマリキーに変換し、プライマリキーは以前のプライマリキーをレプリカキーに変更します。このアクションはプライマリリージョンの更新として知られています。。プライマリリージョンを更新するには、プリンシパルに両方のリージョンの kms:UpdatePrimaryRegion アクセス許可が必要です。キーポリシーまたは IAM ポリシーでこれらのアクセス許可を付与できます。

- プライマリキーの kms:UpdatePrimaryRegion。このアクセス許可は、プライマリキーリージョンで有効である必要があります。
- レプリカキーの kms:UpdatePrimaryRegion。このアクセス許可は、レプリカキーリージョンで有効である必要があります。

例えば、次のキーポリシーは、KMS キーのプライマリリージョンを更新する管理者のロールを引き受けることができるユーザーに付与します。この KMS キーを、このオペレーションでプライマリキーまたはレプリカキーにすることができます。

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

プライマリキーをホストAWS リージョンできる を制限するには、kms:PrimaryRegion 条件キーを使用します。例えば、次の IAM ポリシーステートメントでは、新しいプライマリリージョンが、指定されたリージョンの 1 つである場合にのみ、プリンシパルは AWS アカウント でマルチリージョンキーのプライマリリージョンを更新できます。

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
      "arn:aws:kms:*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
```

```
            "us-west-2",
            "sa-east-1",
            "ap-southeast-1"
        ]
      }
    }
}
```

マルチリージョンキーを使用および管理する

デフォルトでは、AWS アカウント でKMS キーを使用および管理するためのアクセス許可を持つプリンシパルは、マルチリージョンキーを使用および管理するためのアクセス許可も持っています。ただし、kms:MultiRegion 条件キーを使用して、単一リージョンキーのみ、またはマルチリージョンキーのみを許可できます。または、kms:MultiRegionKeyType 条件キーを使用して、マルチリージョンのプライマリキーのみ、またはレプリカキーのみを許可します。どちらの条件キーも、 CreateKey オペレーション、および Encrypt や などの既存の KMS キーを使用するオペレーションへのアクセスを制御しますEnableKey。

以下の IAM ポリシーステートメントの例では、kms:MultiRegion 条件キーを使用して、プリンシパルがマルチリージョンキーを使用または管理できないようにします。

```
{
  "Effect": "Deny",
  "Action": "kms:*",
  "Resource": "*",
  "Condition": {
    "Bool": "kms:MultiRegion": true
  }
}
```

この IAM ポリシーステートメントの例では、kms:MultiRegionKeyType 条件を使用して、マルチリージョンレプリカキーのみでキー削除のスケジュールおよびキャンセルを実行することをプリンシパルに許可します。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
```

```
      "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

マルチリージョンキーの同期を AWS KMS に認可する

マルチリージョンキーをサポートするために、AWS KMS は IAM サービスリンクロールを使用します。このロールは、AWS KMS に共有プロパティの同期に必要なアクセス許可を付与します。AWS CloudTrail ログで共有プロパティのAWS KMS同期を記録するSynchronizeMultiRegionKey CloudTrail イベントを表示できます。

マルチリージョンキーのサービスリンクロールについて

サービスリンクロールは、ユーザーの代わりに他の AWS サービスを呼び出す 1 つの AWS サービスアクセス許可を付与する IAM ロールです。これは、複数の統合された AWS サービスの機能を、複雑な IAM ポリシーを作成したり維持したりせずに簡単に使用できるように設計されました。

マルチリージョンキーの場合、 はAWSKeyManagementServiceMultiRegionKeysServiceRolePolicyポリシーを使用してAWSServiceRoleForKeyManagementServiceMultiRegionKeysサービスにリンクされたロールAWS KMSを作成します。このポリシーは、ロールに `kms:SynchronizeMultiRegionKey` アクセス許可を付与します。これにより、マルチリージョンキーの共有プロパティを同期できます。

AWSServiceRoleForKeyManagementServiceMultiRegionKeys サービスにリンクされたロールは のみを信頼するため`mrk.kms.amazonaws.com`、 のみがこのサービスにリンクされたロールを引き受けAWS KMSることができます。このロールは、AWS KMS がマルチリージョンの共有プロパティを同期するために必要なオペレーションを制限します。AWS KMS に対して追加のアクセス許可は付与されません。例えば、AWS KMS に KMS キーを作成、レプリケート、削除するためのアクセス許可はありません。

AWS のサービスでサービスリンクロールを使用する方法の詳細については、IAM ユーザーガイドのサービスリンクロールの使用を参照してください。

サービスにリンクされたロールの作成

AWS KMS マルチリージョンキーを作成するAWS アカウントときに、ロールがまだ存在しない場合、 でAWSServiceRoleForKeyManagementServiceMultiRegionKeys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。このサービスにリンクされたロールを直接作成または再作成することはできません。

サービスにリンクされたロールの説明を編集する

AWSServiceRoleForKeyManagementServiceMultiRegionKeys サービスにリンクされたロールでは、ロール名またはポリシーステートメントを編集することはできませんが、ロールの説明を編集することはできます。手順については、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。

サービスにリンクされたロールを削除する

AWS KMS はAWSServiceRoleForKeyManagementServiceMultiRegionKeys、サービスにリンクされたロールを から削除せずAWS アカウント、削除することもできません。ただし、 AWS KMS は、 AWS アカウントおよびリージョンにマルチリージョンキーがない限り、AWSServiceRoleForKeyManagementServiceMultiRegionKeysロールを引き受けたり、そのアクセス許可を使用したりしません。

マルチリージョンキーを作成する

マルチリージョンキーは、コンソールで、または AWS KMS API を使用して作成できます。

この手順で設定するマルチリージョンプロパティはイミュータブルです。単一リージョンキーをマルチリージョンキーに変換したり、マルチリージョンキーを単一リージョンキーに変換したりすることはできません。

トピック

- マルチリージョンのプライマリキーを作成する
- マルチリージョンのレプリカキーを作成する

マルチリージョンのプライマリキーを作成する

マルチリージョンのプライマリキーは、AWS KMS コンソールで、または AWS KMS API を使用して作成できます。プライマリキーは、AWS KMS がマルチリージョンキーをサポートする任意の AWS リージョン で作成できます。

マルチリージョンのプライマリキーを作成するには、プリンシパルには、IAM ポリシーの kms:CreateKey アクセス許可など、KMS キーを作成するために必要なのと同じアクセス

許可が必要です。プリンシパルには iam:CreateServiceLinkedRole アクセス許可も必要です。kms:MultiRegionKeyType 条件キーを使用して、マルチリージョンのプライマリキーを作成するアクセス許可を許可または拒否できます。

これらの手順により、AWS KMS を生成するキーマテリアルを持つ、マルチリージョンのプライマリキーが作成されます。インポートされたキーマテリアルを持つマルチリージョンのプライマリキーを作成するには、キーマテリアルがインポートされたプライマリキーを作成する を参照してください。

トピック

- マルチリージョンのプライマリキーを作成する (コンソール)
- マルチリージョンのプライマリキーを作成する (AWS KMS API)

マルチリージョンのプライマリキーを作成する (コンソール)

マルチリージョンのプライマリキーを AWS KMS コンソールで作成するには、KMS キーを作成するのと同じプロセスを使用します。[Advanced options (アドバンストオプション)] でマルチリージョンキーを選択します。詳細な手順については、「キーの作成」を参照してください。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. [Create key] (キーの作成) を選択します。
5. 対称または非対称のキータイプを選択します。デフォルトは [Symmetric] (対称) キーです。

   マルチリージョンの対称キーと非対称キーを作成できます。これには、対称であるマルチリージョン HMAC KMS キーが含まれます。
6. キーの用途を選択します。デフォルトは [Encrypt and decrypt] (暗号化および復号化) です。

ヘルプについては、「the section called “キーの作成”」、「the section called “非対称 KMS キーを作成する”」、または「the section called “HMAC キーの作成”」を参照してください。

7. [詳細オプション] を展開します。
8. [Key material origin] (キーマテリアルのオリジン) で、プライマリキーとレプリカキーが共有するキーマテリアルを AWS KMS に生成させるには、[KMS] を選択します。キーマテリアルをプライマリキーとレプリカキーにインポートするには、[External (Import key material)] (外部 (キーマテリアルのインポート)) を選択します。
9. [Multi-Region replication] (マルチリージョンレプリケーション) で、[Allow this key to be replicated into other Regions] (このキーを他のリージョンにレプリケートすることを許可する) を選択します。

   KMS キーの作成後は、この設定を変更できません。
10. プライマリキーのエイリアスを入力します。

    エイリアスは、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じエイリアスまたは別のエイリアスを割り当てることができます。AWS KMS は、マルチリージョンキーのエイリアスを同期しません。

    > **Note**
    >
    > エイリアスを追加、削除、更新すると、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「エイリアスを使用して KMS キーへのアクセスを制御する」を参照してください。
11. (オプション) プライマリキーの説明を入力します。

    説明は、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じ説明または別の説明を割り当てることができます。AWS KMS は、マルチリージョンキーのキーの説明を同期しません。
12. (オプション) タグキーとオプションのタグ値を入力します。プライマリキーに複数のタグを割り当てるには、[Add tag] (タグを追加する) を選択します。

    タグは、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じタグまたは別のタグを割り当てることができます。AWS KMS は、マルチリージョンキーのタグを同期しません。KMS キーのタグはいつでも変更できます。

> **Note**
>
> KMS キーのタグ付けまたはタグ解除により、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「タグを使用してKMS キーへのアクセスを制御する」を参照してください。

13. プライマリキーを管理できる IAM ユーザーとロールを選択します。

> **Note**
>
> IAM ポリシーでは、他の IAM ユーザーおよびロールに、KMS キーを管理するアクセス許可を付与できます。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

このステップでは、キーポリシーをプライマリキー用に作成するプロセスをスタートします。キーポリシーは、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じキーポリシーまたは別キーポリシーを割り当てることができます。AWS KMS は、マルチリージョンキーのキーポリシーを同期しません。KMS キーのキーポリシーは、いつでも変更できます。

14. キーユーザーの選択など、キーポリシーを作成するステップを完了します。キーポリシーを確認したら、[Finish] (完了) を選択して KMS キーを作成します。

マルチリージョンのプライマリキーを作成する (AWS KMS API)

マルチリージョンのプライマリキーを作成するには、 CreateKeyオペレーションを使用します。値が `True` の `MultiRegion` パラメータを使用します。

例えば、次のコマンドでは、発信者の AWS リージョン (us-east-1) でマルチリージョンのプライマリキーを作成します。キーポリシーを含む、他のすべてのプロパティはデフォルト値を受け入れます。マルチリージョンのプライマリキーのデフォルト値は、他のすべての KMS キーのデフォルト値と同じです (デフォルトのキーポリシーを含む)。この手順は、対称暗号化キー、デフォルト KMS キーを作成します。

レスポンスには、MultiRegion 要素と典型的なサブ要素を持つ MultiRegionConfiguration 要素、およびレプリカキーを持たないマルチリージョンのプライマリキーの値が含まれます。マルチリージョンキーのキー ID は、必ず mrk- で始まります。

> **Important**
>
> Description フィールドまたは Tags フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

```
$ aws kms create-key --multi-region
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1606329032.475,
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [ ]
      }
    }
```

```
}
```

マルチリージョンのレプリカキーを作成する

マルチリージョンのレプリカキーは、コンソールで、 ReplicateKeyオペレーションを使用するか、AWS CloudFormationテンプレート を使用して作成できます。 AWS KMSCreateKey オペレーションを使用してレプリカキーを作成することはできません。

これらの手順を使用して、対称暗号化 KMS キー、非対称 KMS キー、または HMAC KMS キーなどの任意のマルチリージョンプライマリキーをレプリケートすることができます。

このオペレーションが完了すると、新しいレプリカキーは一時的に `Creating` のキーステータスを持ちます。このキーステータスは、新しいレプリカキーの作成プロセスが完了すると、数秒後に `Enabled` (または PendingImport) に変わります。キーステータスが `Creating` の間、キーを管理することはできますが、暗号化オペレーションで使用することはできません。レプリカキーをプログラムで作成して使用している場合は、使用する前に を再試行`KMSInvalidStateException`するか、DescribeKeyを呼び出してその`KeyState`値を確認します。

レプリカキーを誤って削除した場合は、この手順を使用してレプリカキーを再度作成できます。同じリージョンで同じプライマリキーをレプリケートする場合、作成する新しいレプリカキーは、元のレプリカキーと同じ共有プロパティを有します。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

詳細はこちら

- インポートされたキーマテリアルを持つマルチリージョンのレプリカキーを作成するには、キーマテリアルがインポートされたレプリカキーを作成する を参照してください。
- AWS CloudFormation テンプレートを使用してレプリカキーを作成するには、「 AWS CloudFormationユーザーガイドAWS::KMS::ReplicaKey」の「」を参照してください。

トピック

- レプリカリージョン

- レプリカキーを作成する (コンソール)
- レプリカキーを作成する (AWS KMS API)

レプリカリージョン

ビジネスモデルと規制要件に基づいて、通常は、マルチリージョンキーを AWS リージョン にレプリケートするよう選択します。例えば、リソースを保管するリージョンにキーをレプリケートできます。または、災害対策の要件に準拠するために、地理的に離れたリージョンにキーをレプリケートすることもできます。

以下は、レプリカリージョンの AWS KMS 要件です。選択したリージョンがこれらの要件を満たしていない場合、キーレプリケーションの試行は失敗します。

- リージョンごとに 1 つの関連するマルチリージョンキー — プライマリキーと同じリージョンにレプリカキーを作成したり、プライマリキーの別のレプリカと同じリージョンにレプリカキーを作成することはできません。

  すでにそのプライマリキーのレプリカがあるリージョンでプライマリキーをレプリケートしようとすると、試行は失敗します。リージョンの現在のレプリカキーが `PendingDeletion` キーステータスにある場合は、レプリカキーの削除をキャンセルするが、レプリカキーが削除されるまで待機します。
- 同じリージョン内の複数の関連しないマルチリージョンキー — 同じリージョン内に、複数の関連しないマルチリージョンキーを持つことができます。例えば、`us-east-1` リージョンで 2 つのマルチリージョンのプライマリキーを持つことができます。プライマリキーごとに、`us-west-2` リージョンでレプリカキーを持つことができます。
- 同じパーティション内のリージョン — レプリカキーリージョンは、プライマリキーリージョンと同じ AWS パーティションである必要があります。
- リージョンの有効化 — リージョンがデフォルトで無効に設定されている場合、AWS アカウントで有効化されるまで、そのリージョンでリソースを作成することはできません。

レプリカキーを作成する (コンソール)

AWS KMS コンソールでは、マルチリージョンのプライマリキーのレプリカを、同じオペレーションで 1 つ以上作成できます。

この手順は、コンソールでスタンダードの単一リージョン KMS キーを作成する場合と似ています。ただし、レプリカキーはプライマリキーに基づいているため、キー仕様 (対称または非対称)、キー使用法、キーオリジンなどの共有プロパティの値は選択しません。

エイリアス、タグ、説明、キーポリシーなど、共有されないプロパティを指定します。便宜上、コンソールにプライマリキーの現在のプロパティ値が表示されますが、変更することもできます。プライマリキー値を保持しても、AWS KMS はこれらの値の同期を維持しません。

> **Important**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. マルチリージョンのプライマリキーのキー ID またはエイリアスを選択します。KMS キーのキーの詳細ページが開きます。

   マルチリージョンのプライマリキーを識別するには、右上隅にあるツールアイコンを使用して [Regionality] (リージョナリティー) 列をテーブルに追加します。
5. [Regionality] (リージョナリティー) タブを選択します。
6. [Related multi-Region keys] (関連するマルチリージョンキー) のセクションで、[Create new replica keys] (新しいレプリカキーの作成) を選択します。

   [Related multi-Region keys] (関連するマルチリージョンキー) のセクションには、プライマリキーとそのレプリカキーのリージョンが表示されます。この表示を使用して、新しいレプリカキーのリージョンを選択できます。
7. 1 つ以上の AWS リージョン を選択します。この手順では、選択したリージョンごとにレプリカキーが作成されます。

   メニューには、プライマリキーと同じ AWS パーティションのプライマリキーのみが含まれます。関連するマルチリージョンキーがすでに存在するリージョンが表示されますが、選択することはできません。メニューのすべてのリージョンに対しては、キーをレプリケートする許可がない場合があります。

リージョンの選択が完了したら、メニューを閉じます。選択したリージョンが表示されます。リージョンへのレプリケーションをキャンセルするには、リージョン名の横にある [X] を選択します。

8. レプリカキーのエイリアスを入力します。

   コンソールには、プライマリキーの現在のエイリアスの 1 つが表示されますが、変更することもできます。マルチリージョンのプライマリキーとそのレプリカに同じエイリアスまたは別のエイリアスを割り当てることができます。エイリアスはマルチリージョンキーの共有プロパティではありません。AWS KMS は、マルチリージョンキーのエイリアスを同期しません。

   エイリアスを追加、削除、更新すると、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「エイリアスを使用して KMS キーへのアクセスを制御する」を参照してください。

9. (オプション) レプリカキーの説明を入力します。

   コンソールには、プライマリキーの現在の説明が表示されますが、変更することもできます。説明は、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じ説明または別の説明を割り当てることができます。AWS KMS は、マルチリージョンキーのキーの説明を同期しません。

10. (オプション) タグキーとオプションのタグ値を入力します。レプリカキーに複数のタグを割り当てるには、[Add tag] (タグを追加する) を選択します。

    コンソールには、プライマリキーに現在アタッチされているタグが表示されますが、変更することもできます。タグは、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じタグまたは別のタグを割り当てることができます。AWS KMS は、マルチリージョンキーのタグを同期しません。

    KMS キーのタグ付けまたはタグ解除により、KMS キーに対するアクセス許可が許可または拒否される可能性があります。詳細については、「AWS KMS の ABAC」および「タグを使用して KMS キーへのアクセスを制御する」を参照してください。

11. レプリカキーを管理できる IAM ユーザーとロールを選択します。

> **Note**
>
> IAM ポリシーは、他の IAM ユーザーおよびロールに レプリカキーを管理するアクセス許可を付与できます。

IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

このステップでは、キーポリシーをレプリカキー用に作成するプロセスをスタートします。コンソールには、プライマリキーの現在のキーポリシーが表示されますが、変更することもできます。キーポリシーは、マルチリージョンキーの共有プロパティではありません。マルチリージョンのプライマリキーとそのレプリカに同じキーポリシーまたは別のキーポリシーを割り当てることができます。AWS KMS は、キーポリシーを同期しません。KMS キーのキーポリシーは、いつでも変更できます。

12. キーユーザーの選択など、キーポリシーを作成するステップを完了します。キーポリシーを確認したら、[Finish] (完了) を選択して レプリカキーを作成します。

レプリカキーを作成する (AWS KMS API)

マルチリージョンのレプリカキーを作成するには、 ReplicateKeyオペレーションを使用します。CreateKey オペレーションを使用してレプリカキーを作成することはできません。このオペレーションでは、一度に 1 つのレプリカキーが作成されます。指定するリージョンは、レプリカキーのリージョンの要件に準拠している必要があります。

`ReplicateKey` オペレーションを使用する際、マルチリージョンキーの任意の共有プロパティの値は指定しません。共有プロパティ値はプライマリキーからコピーされ、同期が維持されます。ただし、共有されないプロパティには値を指定できます。それ以外は、AWS KMS は、プライマリキーの値ではなく、KMS キーのスタンダードのデフォルト値を適用します。

> **Note**
>
> `Description`、`KeyPolicy`、`Tags` パラメータ値を指定しない場合、AWS KMS は、空の文字列説明を含むレプリカキーおよびデフォルトのキーポリシーを作成し、タグを作成しません。
>
> `Description` フィールドまたは `Tags` フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

例えば、次のコマンドでは、アジアパシフィック (シドニー) リージョン (ap-southeast-2) にマルチリージョンのレプリカキーを作成します。このレプリカキーは、米国東部 (バージニア北部) リージョン (us-east-1) のプライマリキーをモデルにしています。これは、KeyId パラメータの値で識別されます。この例では、キーポリシーを含む、他のすべてのプロパティのデフォルト値を受け入れます。

レスポンスは新しいレプリカキーを示します。これには、共有プロパティのフィールド (KeyId、KeySpec、KeyUsage、およびキーマテリアルのオリジン (Origin) など) が含まれます。また、プライマリキーとは独立したプロパティも含まれます (Description、キーポリシー (ReplicaKeyPolicy)、タグ (ReplicaTags) など)。

レスポンスには、プライマリキーのキー ARN とリージョン、およびそのすべてのレプリカキー (ap-southeast-2 リージョンで作成されたものを含む) も含まれます。この例では、このプライマリキーが既に欧州 (アイルランド) リージョン (eu-west-1) でレプリケートされていることを、ReplicaKey 要素が示しています。

```
$ aws kms replicate-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
    --replica-region ap-southeast-2
{
    "ReplicaKeyMetadata": {
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "REPLICA",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                },
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                }
            ]
```

```
        },
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1607472987.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    },
    "ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-
default-1\",...,
    "ReplicaTags": []
}
```

マルチリージョンキーを表示する

単一リージョンキーおよびマルチリージョンキーは AWS KMS コンソールで、および AWS KMS API オペレーションを使用して表示できます。

トピック

- コンソールでマルチリージョンを表示する
- API でマルチリージョンキーを表示する

コンソールでマルチリージョンを表示する

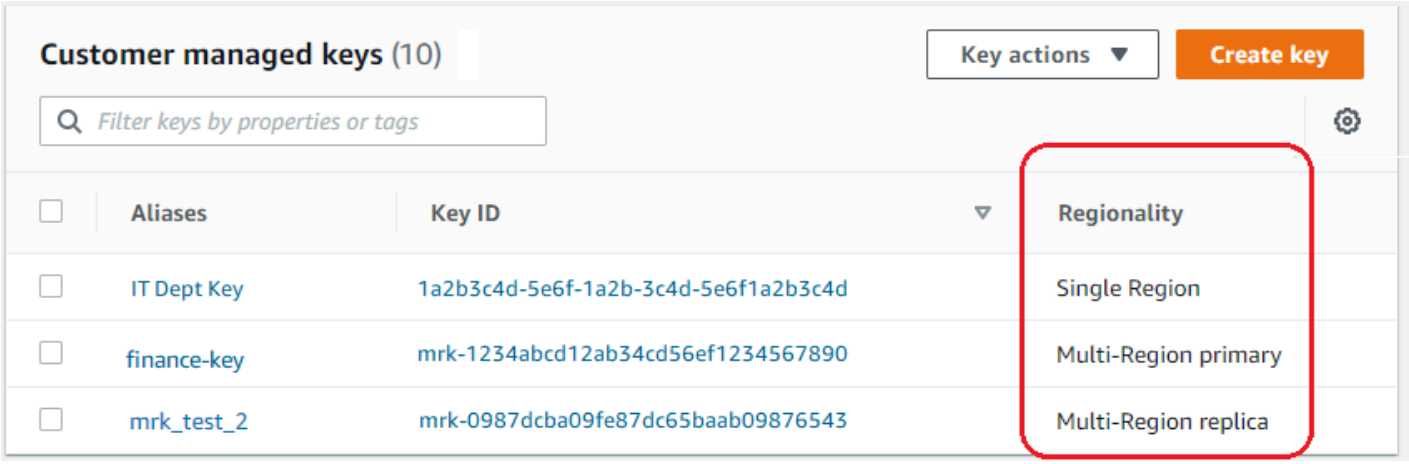
AWS KMS コンソールでは、選択したリージョンで KMS キーを表示できます。マルチリージョンキーを持っている場合、他の AWS リージョン で関連するマルチリージョンキーを表示することもできます。

AWS KMS コンソールのカスタマーマネージドキーテーブルでは、選択したリージョンで KMS キーのみが表示されます。選択したリージョンでマルチリージョンのプライマリキーおよびレプリカキー

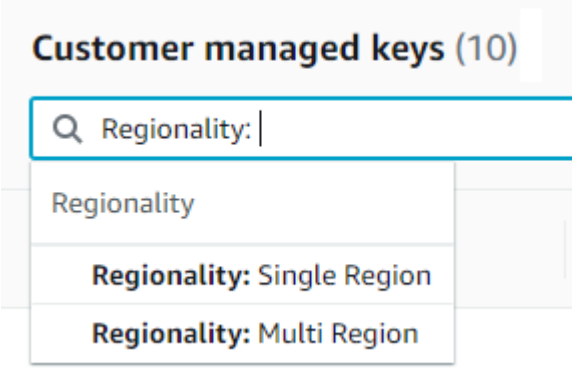
を表示できます。AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。

AWS マネージドキー は常に単一リージョンキーのため、AWS マネージドキー テーブルにリージョナリティ機能はありません。

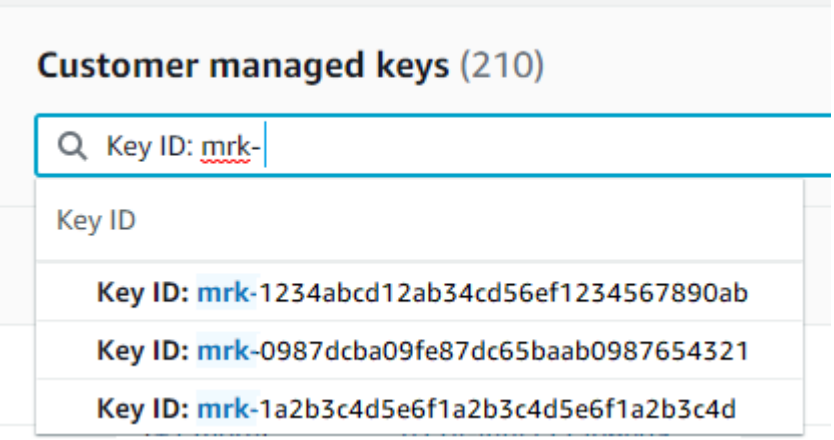
- マルチリージョンキーを簡単に識別するには、キーテーブルにリージョナリティー列を追加します。ヘルプについては、「KMS キーテーブルをカスタマイズする」を参照してください。



- キーテーブルに単一リージョンキーのみ、またはマルチリージョンキーのみを表示するには、各キーのリージョナリティープロパティでキーをフィルタリングします。ヘルプについては、「KMS キーをソートおよびフィルタリングする」を参照してください。



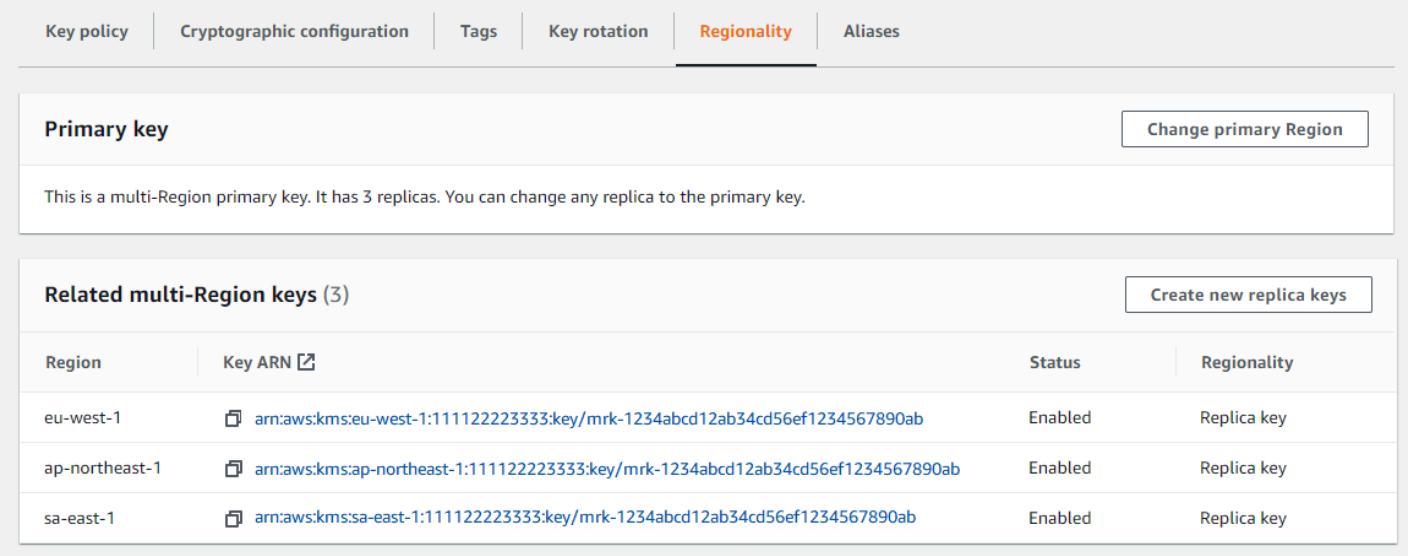
- 固有の mrk- キー ID プレフィックスのカスタマーマネージドキーテーブルをソートおよびフィルタリングすることもできます。



- マルチリージョンのプライマリキーまたはレプリカキーの詳細については、キーの詳細ページに移動して、[Regionality (リージョナリティー)] タブを選択します。

  プライマリキーの [Regionality (リージョナリティー)] タブには、プライマリリージョンの変更ボタンと新しいレプリカキーの作成ボタンがあります。(レプリカキーのリージョナリティータブには、どちらのボタンもありません)。関連するマルチリージョンキーセクションには、現在のキーに関連するすべてのマルチリージョンキーが一覧表示されます。現在のキーがレプリカキーの場合、このリストにはプライマリキーが含まれます。

  関連するマルチリージョンキーを関連するマルチリージョンキーテーブルから選択すると、AWS KMS コンソールが選択したキーのリージョンに変わり、キーの詳細ページが開きます。例えば、以下の [Related multi-Region keys] (関連するマルチリージョンキー) セクション例の `sa-east-1` リージョンでレプリカキーを選択した場合、AWS KMS コンソールが `sa-east-1` リージョンに変わり、そのレプリカキーの詳細ページが表示されます。レプリカキーのエイリアスまたはキーポリシーを表示するにはこのオペレーションを行います。リージョンを再度変更するには、ページの右上隅にあるリージョンセレクターを使用します。



API でマルチリージョンキーを表示する

AWS KMS API でマルチリージョンキーを表示するには、 DescribeKeyオペレーションを使用します。指定したキーとそれに関連するすべてのマルチリージョンキーが表示されます。

AWS KMS コンソール同様、AWS KMS API オペレーションはリージョナルです。例えば、ListKeysまたは ListAliasesオペレーションを呼び出すと、現在または指定されたリージョンのリソースのみが返されます。ただし、マルチリージョンキーで DescribeKey オペレーションを呼び出すと、レスポンスには他の AWS リージョン のすべての関連するマルチリージョンキーが含まれます。

例えば、次の例の DescribeKey リクエストでは、アジアパシフィック (東京) (ap-northeast-1) リージョンのマルチリージョンのサンプルレプリカキーの詳細を取得します。

```
$ aws kms describe-key \
        --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
        --region ap-northeast-1
```

レスポンスのほとんどの KeyMetadata は、リクエストの対象となるアジアパシフィック (東京) リージョンのレプリカキーを記述します。ただし、MultiRegionConfiguration 要素は、米国西部 (オレゴン) (us-west-2) リージョンのプライマリキー、およびアジアパシフィック (東京) リージョンのレプリカを含む、他の AWS リージョン のレプリカキーを記述します。DescribeKey はすべての関連するマルチリージョンキーの同じ MultiRegionConfiguration 値を返します

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1586329200.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-northeast-1"
                },
                {
                    "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "sa-east-1"
                }
            ]
        }
```

```
    }
}
```

マルチリージョンのキーを管理する

ほとんどのアクションでは、単一リージョンキーを使用および管理するのと同じ方法で、マルチリージョンキーを管理します。キーを有効または無効にしたり、エイリアス、キーポリシー、権限、タグを設定したり、更新したりできます。ただし、マルチリージョンキーの管理は、次の点で異なります。

- プライマリリージョンを更新できます。これにより、レプリカキーの 1 つがプライマリキーに変更され、現在のプライマリキーがレプリカに変更されます。
- 自動キーローテーションはプライマリキーでのみ管理します。
- 関連する任意のプライマリキーまたはレプリカキーから非対称マルチリージョンキーの公開キーを取得できます。

KMS キーの作成時に設定するマルチリージョンのプロパティはイミュータブルです。単一リージョンキーをマルチリージョンキーに変換したり、マルチリージョンキーを単一リージョンキーに変換したりすることはできません。

プライマリリージョンを更新する

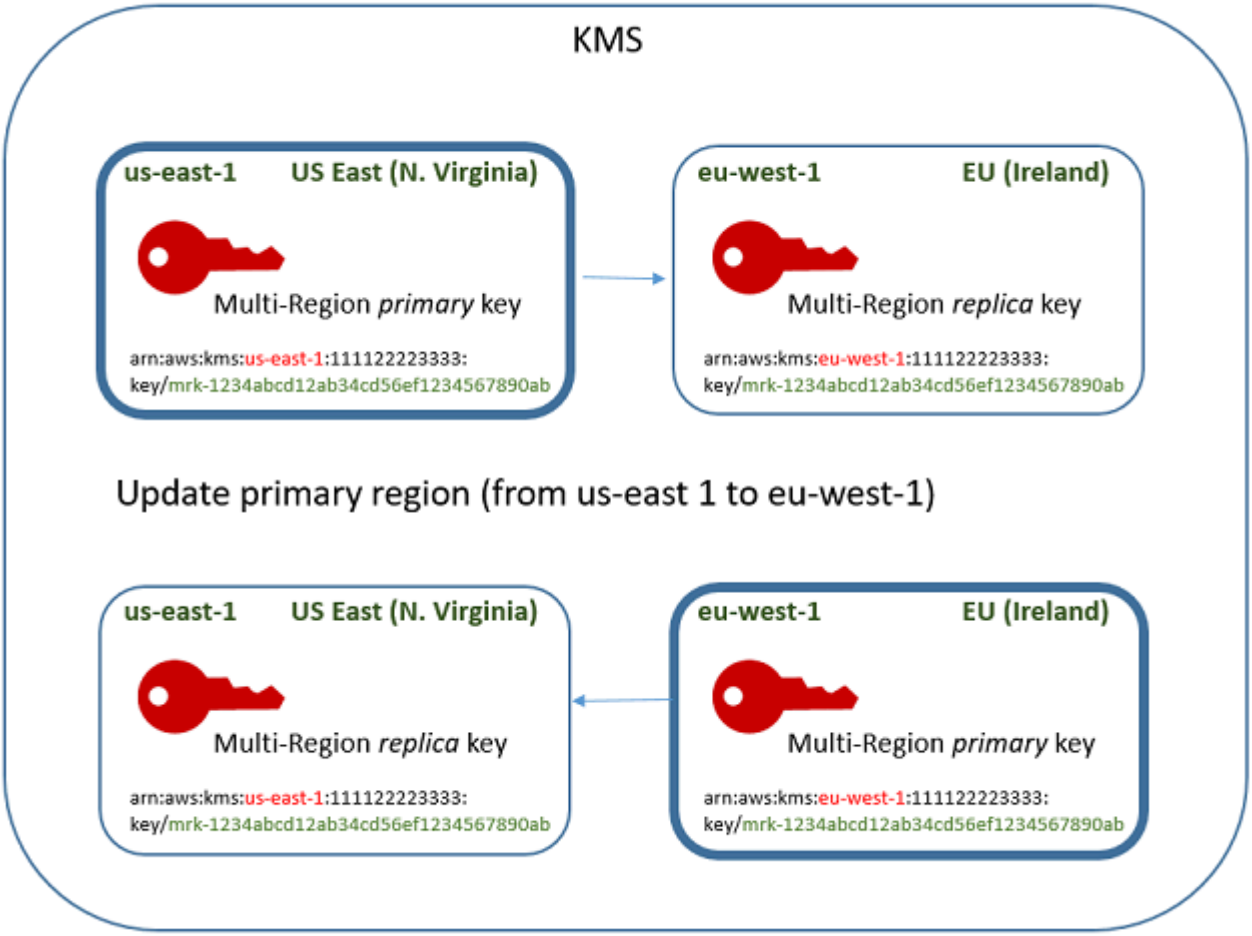
関連するマルチリージョンのキーのセットはすべて、プライマリキーを持つ必要があります。ただし、プライマリキーは変更できます。このアクションは、プライマリリージョンの更新と呼ばれ、現在のプライマリキーをレプリカキーに変換し、関連するレプリカキーの 1 つをプライマリキーに変換します。これは、レプリカキーを維持しながら現在のプライマリキーを削除する必要がある場合、またはキー管理者と同じリージョンでプライマリキーを検索する必要がある場合に実行できます。

関連する任意のレプリカキーを選択して、新しいプライマリキーにすることができます。オペレーションのスタート時に、プライマリキーとレプリカキーの両方が `Enabled` キーステータスである必要があります。

このオペレーション完了後も、プライマリリージョンの更新プロセスは、さらに数秒間進行中である可能性があります。この間、新旧プライマリキーのキーステータスは、一時的に更新中となります。キーステータスが `Updating` の間も暗号化オペレーションでキーを使用できますが、新しいプライマリキーをレプリケートしたり、これらのキーを有効または無効にするなどの、特定の管理オペレーションを実行することはできません。などのオペレーションでは、古いプライマリキーと新し

いプライマリキーの両方がレプリカとして表示DescribeKeyされる場合があります。更新が完了すると、Enabled キーステータスは復元されます。

米国東部 (バージニア北部) (us-east-1) にプライマリキーがあり、欧州 (アイルランド) (eu-west-1) にレプリカキーがあるとします。更新機能を使用して、米国東部 (バージニア北部) (us-east-1) のプライマリキーをレプリカキーに変更し、欧州 (アイルランド) (eu-west-1) のレプリカキーをプライマリキーに変更できます。



更新プロセスが完了すると、欧州 (アイルランド) (eu-west-1) リージョンのマルチリージョンキーがマルチリージョンのプライマリキーになり、米国東部 (バージニア北部) (us-east-1) リージョンのキーがそのレプリカキーになります。関連するレプリカキーが他に存在する場合、それらは新しいプライマリキーのレプリカになります。次回、AWS KMS はマルチリージョンキーの共有プロパティを同期し、共有プロパティを新しいプライマリキーからコピーして、それらを以前のプライマリキーを含むレプリカキーにコピーします。

更新オペレーションは、任意のマルチリージョンキーのキー ARN には影響を及ぼしません。また、キーマテリアルなどの共有プロパティや、キーポリシーなどの独立したプロパティにも影響を及ぼしません。ただし、新しいプライマリキーのキーポリシーの更新が必要になります。例え

ば、kms:ReplicateKey 信頼されたプリンシパルの アクセス許可を新しいプライマリキーに追加し、新しいレプリカキーから削除できます。

`Updating` キーステータス

プライマリリージョンの更新プロセスには、多くの AWS KMS オペレーションに影響を及ぼす結果整合性の短い遅延よりも、若干長い時間がかかります。`UpdatePrimaryRegion` オペレーションが復帰した後、またはコンソールで更新手順が完了した後も、プロセスがまだ進行中である可能性があります。などのオペレーションでは、プロセスが完了するまで、古いプライマリキーと新しいプライマリキーの両方がレプリカとして表示DescribeKeyされることがあります。

プライマリリージョンの更新プロセス中、新旧のプライマリキーは、`Updating` キーステータスとなります。更新プロセスが正常に完了すると、両方のキーは `Enabled` キーステータスに戻ります。`Updating` ステータスの間、キーの有効化や無効化などの、一部の管理オペレーションは使用できません。ただし、暗号化オペレーションでは、両方のキーを中断なしで使用し続けることができます。`Updating` キーステータスの影響の詳細については、AWS KMS キーのキーステータス を参照してください。

プライマリリージョンを更新する (コンソール)

プライマリキーは AWS KMS コンソールで更新できます。現在のプライマリキーのキーの詳細ページをスタートします。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. マルチリージョンのプライマリキーのキー ID またはエイリアスを選択します。これにより、プライマリキーのキーの詳細ページが開きます。

   マルチリージョンのプライマリキーを識別するには、右上隅にあるツールアイコンを使用して [Regionality] (リージョナリティー) 列をテーブルに追加します。
5. [Regionality] (リージョナリティー) タブを選択します。
6. [Primary key (プライマリキー)] セクションで、[Change primary Region (プライマリリージョンの変更)] を選択します
7. 新しいプライマリキーのリージョンを選択します。メニューから選択できるリージョンは 1 つだけです。

[Change primary Region (プライマリリージョンの変更)] メニューには、関連するマルチリージョンキーを持つリージョンのみが含まれます。メニュー上のすべてのリージョンに対するプライマリリージョンを更新するアクセス許可は持っていない可能性があります。

8. [Change primary Region (プライマリリージョンの変更)] を選択します。

プライマリリージョンを更新する (AWS KMS API)

関連するマルチリージョンキーのセットのプライマリキーを変更するには、UpdatePrimaryRegionオペレーションを使用します。

KeyId パラメータを使用して、現在のプライマリキーを識別します。PrimaryRegion パラメータを使用して、新しいプライマリキーの AWS リージョン を表示します。プライマリキーに新しいプライマリリージョンのレプリカがまだない場合、オペレーションは失敗します。

次の例では、プライマリキーを us-west-2 リージョンのマルチリージョンキーから eu-west-1 リージョンのレプリカに変更します。KeyId パラメータは、現在のプライマリキーを us-west-2 リージョンで識別します。PrimaryRegion パラメータは、新しいプライマリキー、eu-west-1 の AWS リージョン を指定します。

```
$ aws kms update-primary-region \
      --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
      --primary-region eu-west-1
```

成功すると、このオペレーションは出力を返さず、HTTP ステータスコードのみを返します。効果を確認するには、マルチリージョンキーのいずれかで DescribeKeyオペレーションを呼び出します。キーステータスが Enabled に戻るまで待機する必要がある場合があります。キーステータスが更新中の間は、キー値がまだ流動的である可能性があります。

例えば、次の DescribeKey 呼び出しにより、eu-west-1 リージョンでマルチリージョンキーの詳細を取得します。出力は、eu-west-1 リージョンのマルチリージョンキーが現在、プライマリキーであることを示します。関連する us-west-2 リージョンのマルチリージョンキー (同じキー ID) は現在、レプリカキーです。

```
$ aws kms describe-key \
      --key-id arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
```

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1609193147.831,
        "Enabled": true,
        "Description": "multi-region-key",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
           "MultiRegionKeyType": "PRIMARY",
           "PrimaryKey": {
              "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
              "Region": "eu-west-1"
           },
           "ReplicaKeys": [
              {
                 "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                 "Region": "us-west-2"
              }
           ]
        }
    }
}
```

マルチリージョンキーをローテーションする

マルチリージョンキーでキーマテリアルの自動ローテーションを、有効または無効にすることができます 自動キーローテーションは、マルチリージョンキーの共有プロパティです。

自動キーローテーションはプライマリキーでのみ、有効または無効にできます。

- AWS KMS がマルチリージョンキーを同期すると、キーローテーションプロパティの設定がプライマリキーから関連するすべてのレプリカキーにコピーされます。
- AWS KMS がキーマテリアルをローテーションすると、プライマリキーの新しいキーマテリアルが作成され、リージョンの境界を越えて、新しいキーマテリアルが関連するすべてのレプリカキーにコピーされます。キーマテリアルが暗号化されないまま AWS KMS から出ていくことはありません。このステップは、暗号化オペレーションでキーが使用される前にキーマテリアルが完全に同期されるよう、慎重に制御されます。
- AWS KMS では、プライマリキーとそのすべてのレプリカキーでキーマテリアルが使用可能になるまで、新しいキーマテリアルでデータを暗号化しません。
- ローテーションされたプライマリキーをレプリケートすると、新しいレプリカキーに、現在のキーマテリアルと関連するマルチリージョンキーのキーマテリアルの以前のすべてのバージョンが含まれます。

このパターンにより、関連するマルチリージョンキーが完全に相互運用可能であることが保証されます。すべてのマルチリージョンキーは、キーが作成される前に暗号化テキストが暗号化されていても、関連するマルチリージョンキーによって暗号化された暗号化テキストを復号できます。

自動キーローテーションは、非対称 KMS キーまたはインポートされたキーマテリアルを持つ KMS キーではサポートされません。自動キーローテーションの詳細と、キーローテーションを有効または無効にする手順については、AWS KMS keys ローテーション を参照してください。

パブリックキーのダウンロード

マルチリージョンの作成時に、非対称 KMS キー、AWS KMS はプライマリキーの RSA または楕円曲線 (ECC) キーペアを作成します。次に、そのキーペアをプライマリキーのすべてのレプリカにコピーします。その結果、プライマリキーまたはそのいずれかのレプリカキーから公開キーをダウンロードできます。ユーザーは常に同じキーマテリアルを取得します。

AWS KMS の外部で公開キーをダウンロードして使用する方法については、パブリックキーのダウンロードに関する特別な考慮事項 を参照してください。手順については、「パブリックキーのダウンロード」を参照してください。

キーマテリアルをマルチリージョンキーにインポートする

独自のキーマテリアルをマルチリージョン KMS キーにインポートできます。独自のキーマテリアルを使用して作成するマルチリージョンキーは相互運用可能です。関連するマルチリージョンキーを使用してあるリージョンでデータを暗号化し、別のリージョンでデータを復号できます。

ただし、キーマテリアルを管理する必要があります。

- AWS KMS は、キーマテリアルがインポートされたプライマリキーからそのレプリカキーに、キーマテリアルをコピーまたは同期しません。同じキーマテリアルを、関連するプライマリキーとレプリカキーにインポートする必要があります。
- キーマテリアルをインポートするときに、各キーの有効期限モデルと有効期限を個別に設定します。関連するマルチリージョンキーに対して、同じ、または異なる有効期限モデルと有効期限を設定できます。キーマテリアルが有効期限に近づいた場合は、影響を受けるマルチリージョンキーにそのキーマテリアルを再インポートする必要があります。

  関連するマルチリージョンキーのキーステータスは、お互いに独立しています。例えば、プライマリキーのキーマテリアルが有効期限切れになっても、そのレプリカキーは影響を受けません。

同じレプリカキーのリージョン要件が、インポートされたキーマテリアルを持つマルチリージョンキーに適用されます。同じキーマテリアルを単一リージョンキーまたは関連のないマルチリージョンキーにインポートした場合、これらの KMS キーは相互運用できません。

インポートされた対称キーマテリアル、非対称キーマテリアル、または HMAC キーマテリアルを含むマルチリージョンキーを作成できます。 AWS KMS は カスタムキーストアでインポートされたキー マテリアルをサポートしていません。キーマテリアルがインポートされた KMS キーの自動キーローテーションを有効にすることもできません。

マルチリージョンの機能を除いて、インポートされたキーマテリアルを持つマルチリージョンキーは、インポートされたキーマテリアルを持つ他の KMS キーと同じです。インポートされたキーマテリアルを持つ単一リージョンキーを作成し、設定する方法の詳細については、インポートしたキーマテリアルについて を参照してください。

トピック

- インポートされたキーマテリアルを持つすべての KMS キーが相互運用できない理由
- キーマテリアルがインポートされたプライマリキーを作成する
- キーマテリアルがインポートされたレプリカキーを作成する

インポートされたキーマテリアルを持つすべての KMS キーが相互運用できない理由

インポートされたキーマテリアルを持つ単一リージョン KMS キーは、同じキーマテリアルを持つ場合でも相互運用できません。AWS KMS が KMS キーを使用してデータを暗号化する際は、キーメタ

データの一部を暗号化テキストに暗号でバインドします。これにより、暗号化テキストが保護され、データを暗号化した KMS キーのみがそのデータを復号できます。

マルチリージョンキーは相互運用できるように設計されています。キーマテリアルが同じであるだけでなく、キー ID と他のメタデータも同じです。したがって、それらが生成する暗号化テキストは、関連する任意のマルチリージョンキーで復号できます。結果として、マルチリージョンキーの信頼プロパティは、単一リージョンキーの信頼プロパティとは異なります。ただし、一部のお客様にとって、複数のリージョンで復号する利点は、単一 AWS リージョン の 1 つの KMS キーの暗号化テキストによるセキュリティバリューを上回ります。

キーマテリアルがインポートされたプライマリキーを作成する

キーマテリアルがインポートされたプライマリキーを作成するには、最初にキーマテリアルなしで KMS キーを作成します。キーマテリアルなしでプライマリキーを作成する場合は、インポートする予定のキーマテリアルのタイプを反映したキー仕様を指定する必要があります。次に、キーマテリアルをプライマリキーにインポートします。

キーマテリアルを持たないマルチリージョンのプライマリキーを作成する手順は、キーマテリアルを持たない単一リージョンキーを作成する手順とほぼ同じです。唯一の違いは、キーがマルチリージョンキーであることを指定することです。

インポートされたキーマテリアルを使用してマルチリージョンのプライマリキーを作成するアクセス許可は、IAM ポリシーの kms:CreateKey および iam:CreateServiceLinkedRole アクセス許可など、AWS KMSキーマテリアルを使用してマルチリージョンのプライマリキーを作成するために必要なアクセス許可と同じです。kms:MultiRegionKeyType および kms:KeyOrigin 条件キーを使用して、インポートされたキーマテリアルを持つマルチリージョンのプライマリキーを作成するアクセス許可を許可または拒否できます。

AWS KMS コンソールで、キーマテリアルがインポートされたプライマリキーを作成するには、[Advanced options] (アドバンストオプション) セクションの設定を使用します。KMS キー作成後にこれらのプロパティを変更することはできません。

- キーマテリアルのオリジンを EXTERNAL (キーマテリアルのインポート) に設定します。
- [Multi-Region replication (マルチリージョンレプリケーション)] を [Allow this key to be replicated into other Regions (このキーを他のリージョンにレプリケートすることを許可する)] に設定します。

CreateKey オペレーションを使用して、インポートされたキーマテリアルを持つプライマリキーを作成する場合は、 `Origin` および `MultiRegion`パラメータを使用し、 `KeySpec`および を指定しま

すKeyUsage。次の例では、ECC_NIST_P384 キーマテリアルをインポートできる EXTERNAL KMS キーを作成します。

```
$ aws kms create-key --origin EXTERNAL --key-spec ECC_NIST_P384 --key-usage SIGN_VERIFY
 --multi-region
```

結果として、キーマテリアルを持たない、キーステータスが PendingImport のマルチリージョンのプライマリキーが作成されます。

この KMS キーを有効にするには、公開キーとインポートトークンをダウンロードし、公開キーを使用してキーマテリアルを暗号化してから、キーマテリアルをインポートする必要があります。手順については、「キーの AWS KMS キーマテリアルのインポート」を参照してください。

キーマテリアルがインポートされたレプリカキーを作成する

AWS KMS コンソールで、または AWS KMS API オペレーションを使用して、マルチリージョンのレプリカキーを作成できます。キーマテリアルがインポートされたマルチリージョンのプライマリキーをレプリケートするには、AWS KMS キーマテリアルを持つレプリカキーを作成するのと同じ手順を使用します。ただし、結果は異なります。レプリケーションプロセスは、プライマリキーと同じキーマテリアルを持つレプリカキーを返す代わりに、キーステータスが PendingImport の、キーマテリアルを持たないレプリカキーを返します。　レプリカキーを有効にするには、プライマリキーにインポートしたのと同じキーマテリアルをレプリカキーにインポートする必要があります。

キーマテリアルはレプリケートされませんが、AWS KMS は、プライマリキーと同じキー ID、キー仕様、キーの用途、キーマテリアルのオリジンを持つレプリカキーを作成します。また、レプリカキーにインポートするキーマテリアルが、プライマリキーにインポートしたキーマテリアルと同一であることも保証されます。

キーマテリアルがインポートされたレプリカキーを作成するには

1. キーマテリアルがインポートされたマルチリージョンのプライマリキーを作成します。
2. 以下のいずれかを行ってください。

   AWS KMS コンソールで、キーマテリアルがインポートされたマルチリージョンのプライマリキーを選択します。次に、その [Regionality (リージョナリティー)] タブで、[Create new replica keys (新しいレプリカキーの作成)] を選択します。。手順については、「レプリカキーを作成する (コンソール)」を参照してください。

または、 ReplicateKeyオペレーションを使用します。KeyId パラメータに、キーマテリアルがインポートされたマルチリージョンのプライマリキーのキー ID またはキー ARN を入力します。手順については、「レプリカキーを作成する (AWS KMS API)」を参照してください。

3. 新しいレプリカキーごとに、公開キーとインポートトークンをダウンロードするステップを実行します。公開キーを使用してプライマリキーのキーマテリアルを暗号化し、プライマリキーのキーマテリアルをレプリカキーにインポートします。レプリカキーごとに、異なる公開キーとインポートトークンが必要です。

   レプリカキーにインポートしようとするキーマテリアルがプライマリキーと異なる場合、オペレーションは失敗します。AWS KMS では、有効期限モデルと有効期限の連携は不要ですが、マルチリージョンキーのビジネスルールを確立する必要があります。手順については、「キーの AWS KMS キーマテリアルのインポート」を参照してください。

インポートされたキーマテリアルを持つキーをレプリケートする許可

キーマテリアルがインポートされたレプリカキーを作成するには、次のアクセス許可が必要です。

プライマリキーリージョンで:

- kms:ReplicateKey プライマリキー (プライマリキーのリージョン内) 。プライマリキーのキーポリシーまたは IAM ポリシーにこのアクセス許可を含めます。

レプリカキーリージョンで:

- kms:CreateKey IAM ポリシーの 。
- kms:GetParametersForImport レプリカキーのキーポリシーまたは IAM ポリシーにこのアクセス許可を含めることができます。
- kms:ImportKeyMaterial レプリカキーのキーポリシーまたは IAM ポリシーにこのアクセス許可を含めることができます。
- kms:TagResource レプリケート時にタグを割り当てるには が必要です。レプリカリージョンの IAM ポリシーにこのアクセス許可を含めます。
- kms:CreateAlias AWS KMSコンソールでキーをレプリケートするには が必要です。詳細については、「エイリアスへのアクセスの制御」を参照してください。

マルチリージョンキーを削除する

マルチリージョンのプライマリキーまたはレプリカキーを使用しなくなった場合は、その削除をスケジュールできます。

KMS キーの削除は常に慎重に行う必要がありますが、プライマリキーがまだ AWS KMS に存在している場合は、マルチリージョンキーのレプリカを削除する際のリスクは比較的低いといえます。レプリカキーをそのリージョンから削除しても、削除されたキーで暗号化された暗号化テキストを発見した場合、関連するマルチリージョンキーを使用してその暗号化テキストを復号できます。また、プライマリキーをレプリカキーリージョンに再度レプリケートして、レプリカキーを再作成することもできます。

ただし、プライマリキーとそのすべてのレプリカキーを削除することは、単一リージョンキーを削除することと同様に、非常に危険なオペレーションです。

> **Warning**
>
> KMS キーを削除することは破壊的であり、リスクを伴います。KMS キーが不要であり、今後も使用しないことが確実である場合にのみ実行してください。不明な場合は、削除するのではなく KMS キーを無効化するべきです。

プライマリキーを削除するには、まずそのプライマリキーのレプリカキーをすべて削除する必要があります。レプリカキーを削除せずに特定のリージョンからプライマリキーを削除する必要がある場合は、プライマリリージョンの更新を使用して、プライマリキーをレプリカキーに変更します。

KMS キーの削除をスケジュールする前に、 AWS KMS keys を削除する トピックの注意点と、KMS キーの過去の使用状況を確認する方法と、待機期間中に KMS キーの使用を警告する CloudWatch アラームを設定する方法について説明するトピックを確認してください。非対称マルチリージョンキーのプライマリキーを削除する前に、非対称キーの削除のトピックを確認してください。

トピック

- マルチリージョンキーを削除するためのアクセス許可
- レプリカキーを削除するには
- プライマリキーを削除するには

マルチリージョンキーを削除するためのアクセス許可

マルチリージョンキーの削除をスケジュールするには、次のアクセス許可のみが必要です。

- kms:ScheduleKeyDeletion — マルチリージョンキーの削除をスケジュールし、待機期間を設定します。

次の関連するアクセス許可を持つことを強くお勧めします。

- kms:CancelKeyDeletion — マルチリージョンキーのスケジュールされた削除をキャンセルします。
- kms:DescribeKey — マルチリージョンキーのキーステータスと関連するマルチリージョンキーのリストを表示します。
- kms:DisableKey — マルチリージョンキーを削除する代わりに無効にするオプションを提供します。
- kms:EnableKey — 削除をキャンセルした後にマルチリージョンキーの機能を復元します。

プライマリキーのレプリケーションおよび変更のアクセス許可を含めることもできます。

- kms:ReplicateKey
- kms:UpdateReplicaRegion

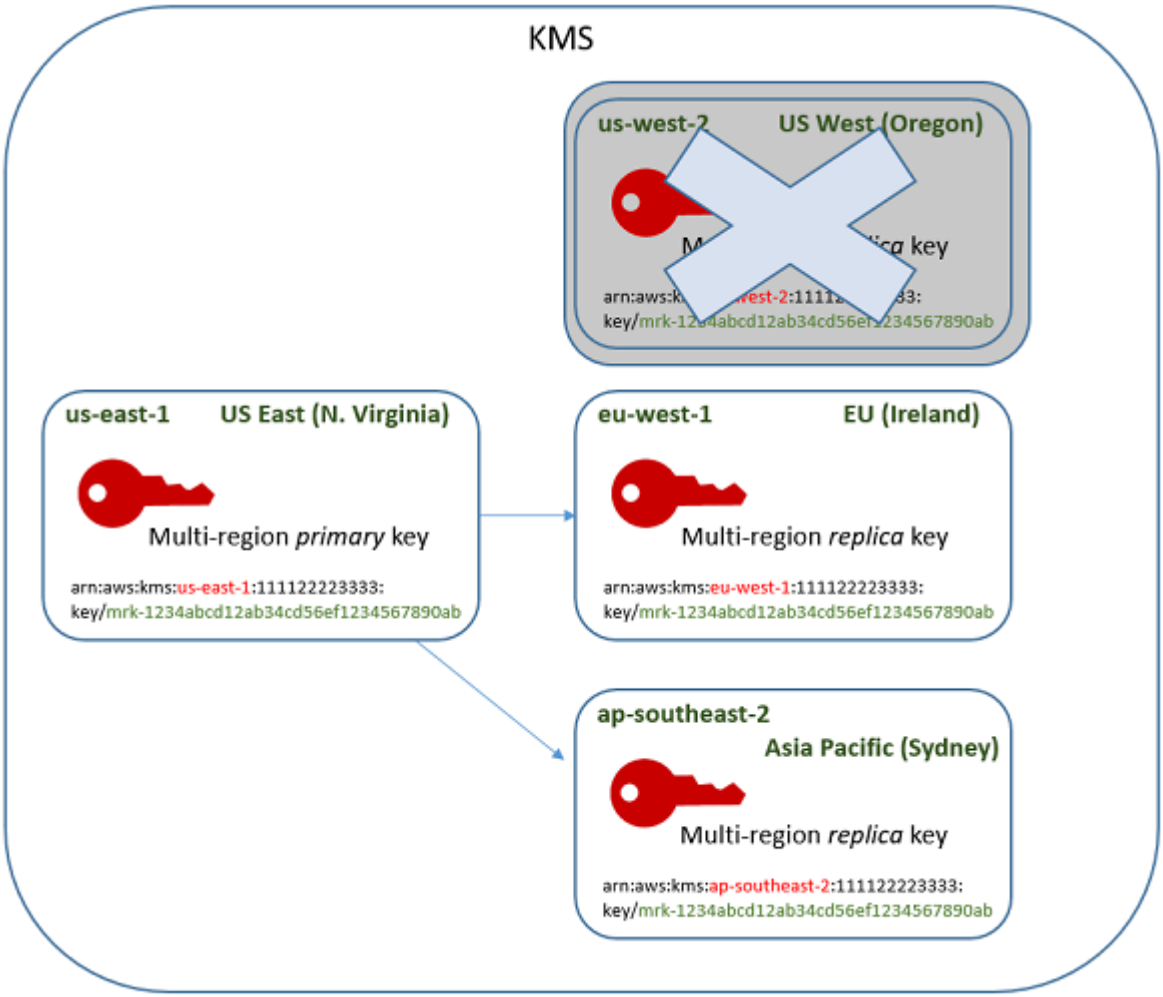
これらのアクセス許可は IAM ポリシーに含めることができますが、管理する必要がある KMS キーにのみ適用されるキーポリシーに付与することがベストプラクティスです。

レプリカキーを削除するには

AWS KMS コンソールまたは AWS KMS API を使用して、レプリカキーを削除できます。レプリカキーはいつでも削除することができます。これは、他の KMS キーのキーステータスに依存しません。

誤ってレプリカキーを削除した場合は、同じリージョンで同じプライマリキーをレプリケートすることで再度作成できます。作成した新しいレプリカキーは、元のレプリカキーと同じ共有プロパティを有します。

マルチリージョンのレプリカキーを削除する手順は、単一リージョンキーを削除する手順と同じです。



1. レプリカキーの削除をスケジュールします。7 ~ 30 日間の待機期間を選択します。デフォルトの待機時間は、30 日です。
2. 待機期間中はレプリカキーのキーステータスが `Pending deletion` (`PendingDeletion`) に変更され、暗号化オペレーションで使用することはできません。
3. レプリカキーのスケジュールされた削除は、待機期間中の任意の時点でキャンセルできます。キーステータスが `Disabled` に変更され、KMS キーを再度有効にできます。
4. 待機期間が終了すると、AWS KMS はレプリカキーを削除します。

アクションのレコードは AWS CloudTrail ログで表示できます。AWS KMS は、KMS キーの削除をスケジュールするオペレーションおよび KMS キーの削除アクションを記録します。

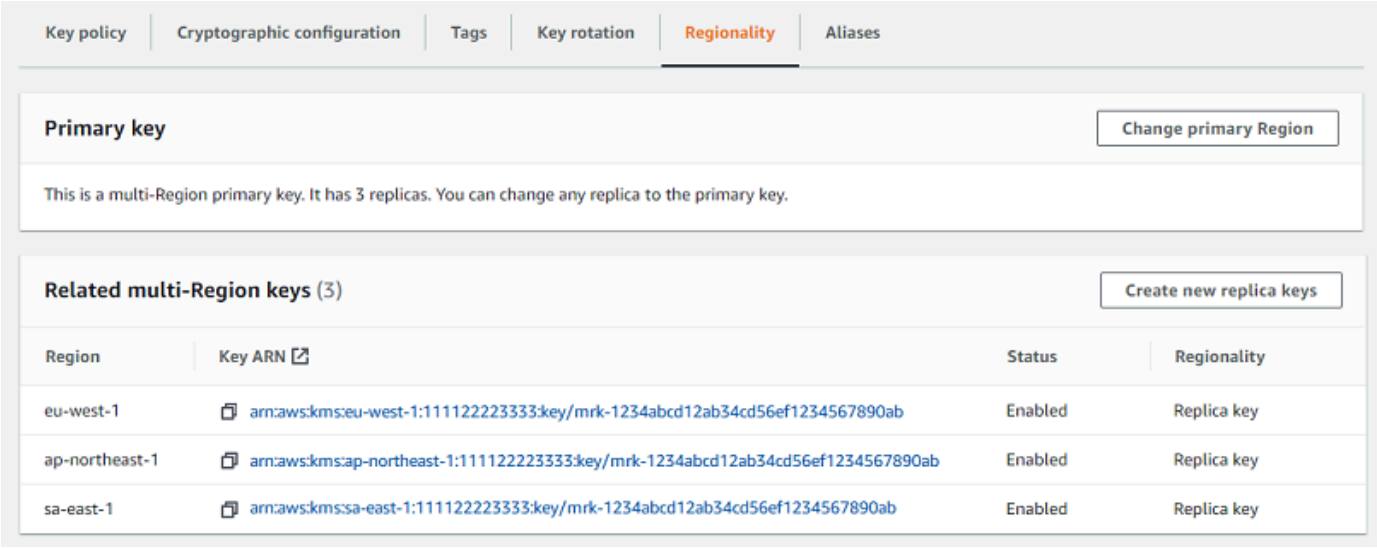
レプリカキーを削除する (コンソール)

マルチリージョンのレプリカキーの削除をスケジュールするには、単一リージョンキーの削除をスケジュールのと同じ手順を使用します。

関連するレプリカキーが別の AWS リージョン に設定されている場合、一度に複数のレプリカキーの削除をスケジュールすることはできません。関連するレプリカキーを削除するには、次のようなパターンを使用します。

関連するすべてのレプリカキーの削除をスケジュールするには

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
3. 右上隅にあるリージョンセレクターを使用して、マルチリージョンのプライマリキーのリージョンを選択します。
4. プライマリキーのエイリアスまたは キー ID を選択します。
5. [Regionality] (リージョナリティー) タブを選択します。



6. [Related multi-Region keys] (関連するマルチリージョンキー) のセクションで、レプリカキーのキー ARN を選択します。

   このアクションにより、新しいブラウザタブでレプリカキーのキーの詳細ページが開きます。コンソールはレプリカキーリージョンに設定されます。
7. [Key actions] (キーのアクション) メニューから、[Schedule key deletion] (キーの削除をスケジュール) を選択します。

このアクションにより、キーの削除をスケジュールするプロセスがスタートします。キーの削除をスケジュールするプロセスを完了します。詳細については、「キー削除のスケジュールとキャンセル (コンソール)」を参照してください。

8. プライマリキーの [Regionality] (リージョナリティー) タブを表示するブラウザタブに戻ります。(レプリカキーの更新ステータスを表示するには、ページを更新する必要がある場合があります)。別のレプリカキーのキー ARN を選択し、レプリカキーの削除をスケジュールするプロセスを繰り返します。

レプリカキーを削除する (AWS KMS API)

マルチリージョンのレプリカキーの削除をスケジュールするには、 ScheduleKeyDeletionオペレーションを使用します。KMS キーを指定するには、キー ID またはキー ARN を使用します。マルチリージョンキーを使用する場合は、明示的なリージョン値を持つキー ARN を使用することでエラーの発生率を減らすことができます。

例えば、このコマンドは us-west-2 米国西部 (オレゴン) リージョンからレプリカキーを削除します。コマンドでは待機期間が指定されていないため、待機期間はデフォルトの 30 日間に設定されます。

```
$ aws kms schedule-key-deletion \
    --region us-west-2 \
    --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

コマンドが成功すると、キー ARN (KeyId)、待機期間 (PendingWindowInDays)、削除日 (DeletionDate)、想定では PendingDeletion となる現在のキーステータス (KeyState) が返されます。

マルチリージョンのレプリカキーを削除するときは、キー ARN のキー ID とリージョン値が想定どおりの値であることを確認してください。

```
{
    "KeyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "DeletionDate": 1599523200.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 30
}
```

マルチリージョンのプライマリキーのすべてのレプリカをプログラムで削除するには、レプリカキーを含むリージョンのリストを作成します。次に、リスト内の各リージョンに対して、上図のように ScheduleKeyDeletion オペレーションを呼び出します。

完全に削除される単一リージョンキーとは異なり、レプリカキーは、削除したレプリカキーがあったリージョンでプライマリキーのレプリケーションを実行することで復元できます。

レプリカキーのステータスを確認し、マルチリージョンキーのプライマリキーとレプリカキーを表示するには、 DescribeKeyオペレーションを使用します。

プライマリキーを削除するには

マルチリージョンのプライマリキーの削除はいつでもスケジュールすることができます。ただし、AWS KMS では、削除がスケジュールされていても、レプリカキーを持つマルチリージョンのプライマリキーは削除されません。

プライマリキーを削除するには、すべてのレプリカキーの削除をスケジュールし、レプリカキーが削除されるまで待機する必要があります。プライマリキーの削除に必要な待機時間は、最後のレプリカキーが削除された時点から始まります。レプリカキーを削除せずに特定のリージョンからプライマリキーを削除する必要がある場合は、プライマリリージョンの更新を使用して、プライマリキーをレプリカキーに変更します。

プライマリキーにレプリカキーがない場合、プロセスはレプリカキーの削除またはリージョナル KMS キーの削除と同じです。

プライマリキーの削除がスケジュールされている間は、プライマリキーを暗号化オペレーションで使用したり、レプリケートすることはできません。ただし、削除がスケジュールされていない限り、レプリカキーは影響を受けません。

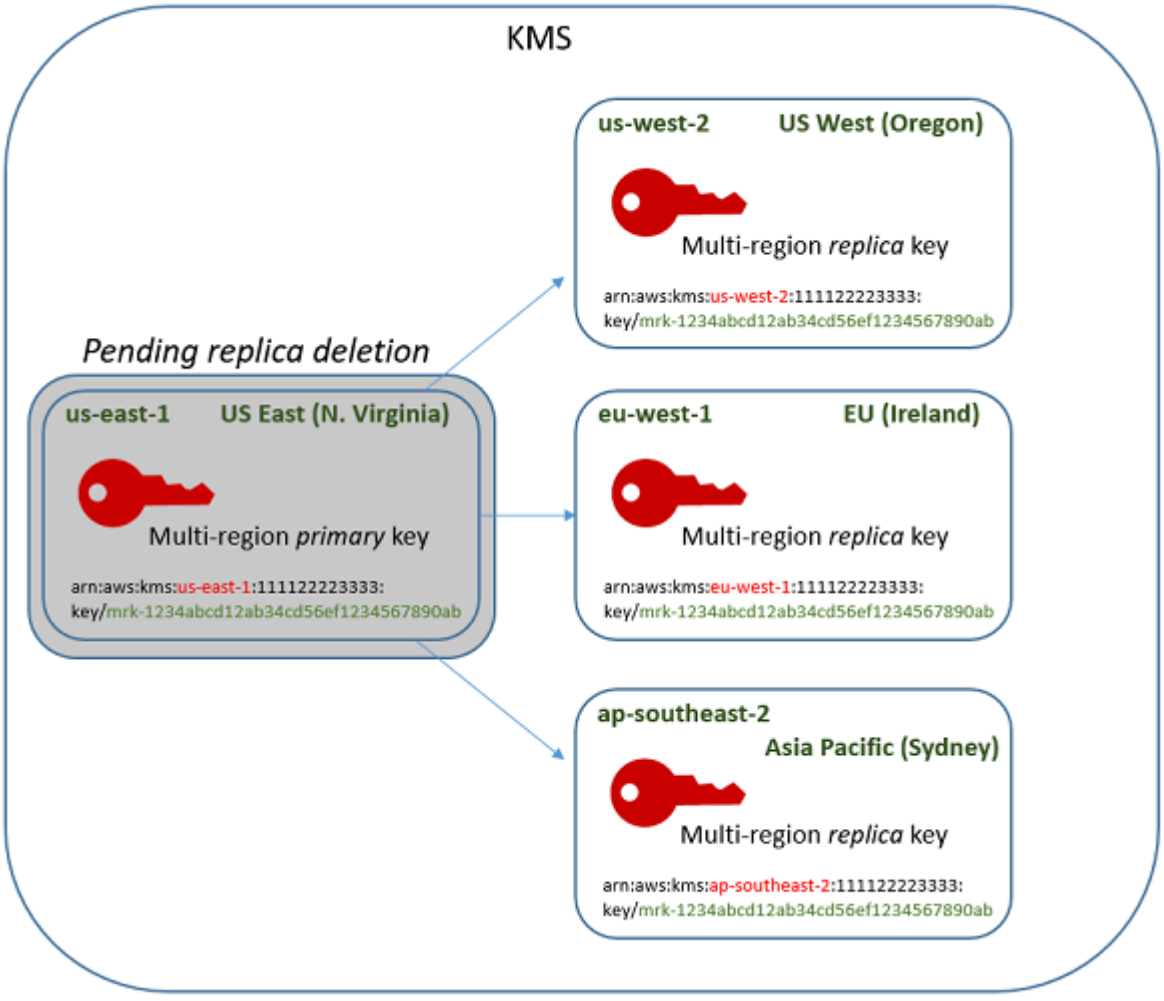
AWS KMS コンソールまたは AWS KMS API を使用して、プライマリキーとレプリカキーの削除をスケジュールできます。プライマリキーの削除スケジュールは、レプリカキーの削除スケジュールの前、後、または同時に設定できます。プロセスは、次のようになります。

1. プライマリキーの削除をスケジュールします。7 ~ 30 日間の待機期間を選択します。デフォルトの待機時間は、30 日です。ただし、プライマリキーの待機期間は、すべてのレプリカキーが削除されるまで開始されません。

   レプリカキーがまだ存在する場合は、プライマリキーのキーステータスが Pending replica deletion (PendingReplicaDeletion) に変更されます。それ以外の場合は、Pending deletion (PendingDeletion) に変更されます。いずれの場合も、プライマリキーを暗号化オペレーションで使用したり、レプリケートしたりすることはできません。

プライマリキーの削除をスケジュールしても、レプリカキーには影響しません。キーステータスは有効なままであり、暗号化オペレーションで使用することができます。レプリカキーが削除されない場合、プライマリキーの `Pending replica deletion` ステータスは無期限に持続する可能性があります。

```
KMS key:                    Key state:
Primary (us-east-1)           Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Enabled
Replica (eu-west-1)           Enabled
Replica (ap-southeast-2)      Enabled
```



2. 各レプリカキーの削除をスケジュールします。7 ~ 30 日間の待機期間を選択します。デフォルトの待機時間は、30 日です。複数のレプリカキーを同時に削除できます。待機期間は同時に実行されます。待機期間中はレプリカキーのキーステータスが `Pending deletion`

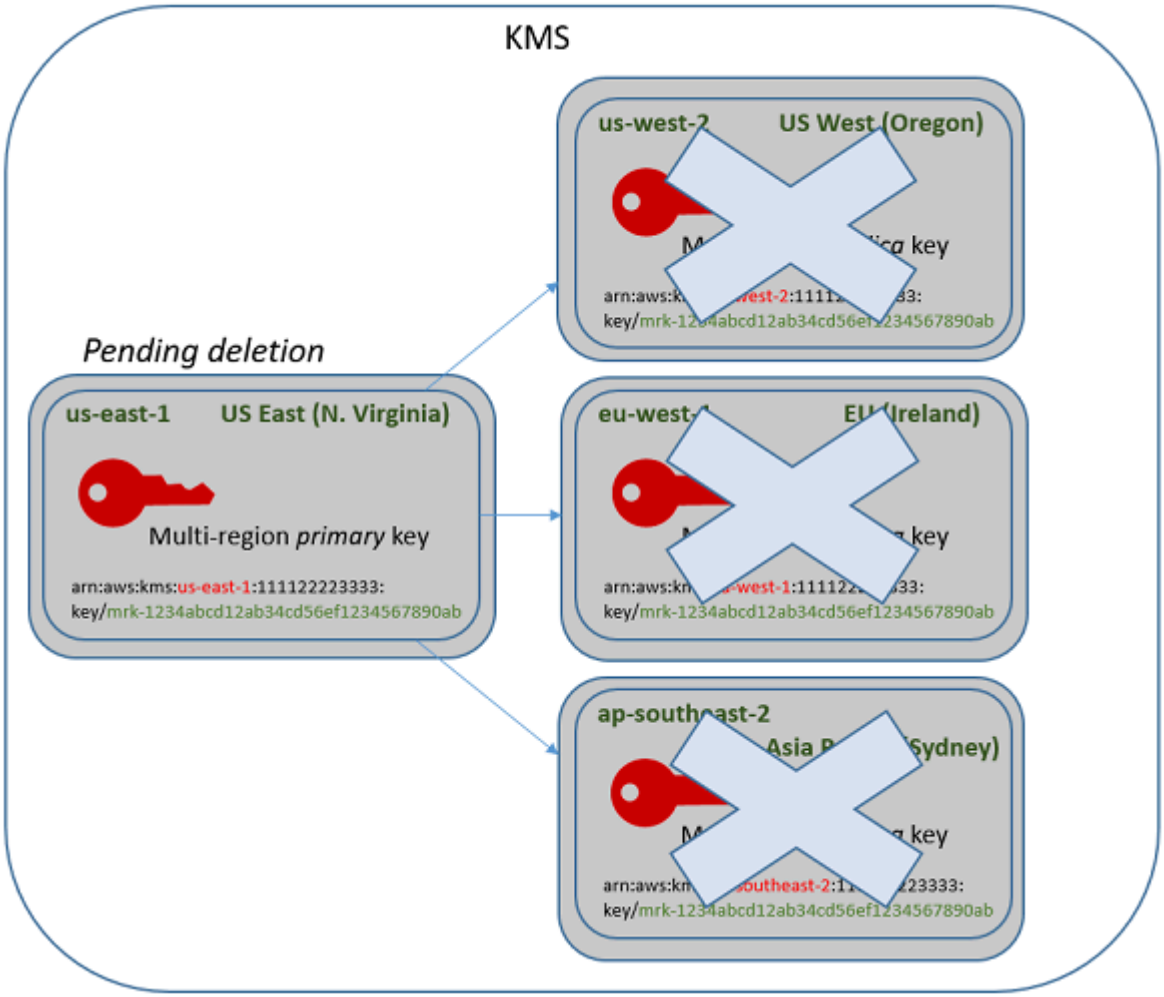
(PendingDeletion) に変化し、これらの KMS キーを暗号化オペレーションで使用することはできません。

例えば、3 つのレプリカキーがある場合、3 つのレプリカキーをすべて同時に削除するようにスケジュールできます。レプリカキーには、同じ、または異なる待機期間を設定することができます。プライマリキーの待機期間がまだ始まっていないことに注意してください。レプリカキーが存在するため、キーステータスは PendingReplicaDeletion です。

```
KMS key:                    Key state:
Primary key (us-east-1)       Pending replica deletion (waiting period 30 days -- not
 started)
Replica (us-west-2)           Pending deletion (7 days)
Replica (eu-west-1)           Pending deletion (7 days)
Replica (ap-southeast-2)      Pending deletion (30 days)
```

3. プライマリキーまたはレプリカキーのスケジュールされた削除は、キーが削除されるまでキャンセルできます。キーステータスが Disabled に変更され、KMS キーを再度有効にできます。
4. 最後のレプリカキーの待機期間が終了すると、AWS KMS は最後のレプリカキーを削除します。プライマリキーのキーステータスが Pending replica deletion (PendingReplicaDeletion) から Pending deletion (PendingDeletion) に変更され、プライマリキーの 7 ～ 30 日の待機期間が開始します。

```
KMS key:                    Key state:
Primary key (us-east-1)       Pending deletion (waiting period 30 days)
```



5. 待機期間が終了すると、AWS KMS はプライマリキーを削除します。

[The minimum time to delete a primary key with replicas is 14 days.] (レプリカを持つプライマリキーを削除する最小期間は 14 日間です。)

プライマリキーとすべてのレプリカキーの削除の待機期間を 7 日間にスケジュールすると、レプリカキーは 7 日後に削除されます。プライマリキーは 14 日後に削除されます。

- 1日目: 最小待機期間が 7 日間のプライマリキーとレプリカキーの削除をスケジュールします。レプリカキーの 7 日間の削除待機期間がスタートします。プライマリキーの削除待機期間はまだスタートしていません。
- 7日目: レプリカキーの削除待機期間が終了します。AWS KMS は、すべてのレプリカキーを削除します。最後のレプリカキーが削除されると、プライマリキーの 7 日間の削除待機期間がスタートします。

- 14日目: プライマリキーの削除待機期間が終了します。AWS KMS はプライマリキーを削除します。

アクションのレコードは AWS CloudTrail ログで表示できます。AWS KMS は、各 KMS キーの削除をスケジュールするオペレーションおよび KMS キーの削除アクションを記録します。

プライマリキーを削除する (コンソール)

マルチリージョンのプライマリキーを削除するには、以下の手順に従います。

キーの削除をスケジュールするには

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 削除するプライマリキーの横にあるチェックボックスをオンにします。このプライマリキーのレプリカを含む 1 つ以上の KMS キーを選択することもできます。
5. [Key actions] (キーのアクション)、[Schedule key deletion] (キーの削除をスケジュール) の順に選択します。
6. 待機中に、警告、および削除のキャンセルに関する情報を読み、検討してください。削除をキャンセルする場合は、[キャンセル] を選択します。
7. [Waiting period (in days)] (待機期間 (日数)) に、日数として 7 ~ 30 の値を入力します。複数の KMS キーを選択した場合、選択した待機期間は、選択したすべての KMS キーに適用されます。レプリカキーの待機期間は同時に実行されますが、プライマリキーの待機期間は、AWS KMS が最後のレプリカキーを削除するまで始まりません。
8. [Confirm that you want to delete this key in ***<number of days>*** days] (<日数> 日後に、このキーを削除することを確認する) の横にあるチェックボックスをオンにします。
9. [Schedule deletion] (削除をスケジュールする) を選択します。

KMS キーの削除ステータスをチェックするには詳細ページ、プライマリキーについては一般設定セクションを参照してください。キーステータスがステータスフィールドに表示されます。プライマリキーのキーステータスが `Pending deletion` に変化すると、スケジュールされた削除日が表示されます。

すべてのプライマリキーおよびレプリカキーのキーステータス (ステータス) は、任意のマルチリージョンキー詳細ページのリージョナリティータブでチェックすることもできます。詳細については、「マルチリージョンキーを表示する」を参照してください。

プライマリキーを削除する (AWS KMS API)

マルチリージョンのレプリカキーを削除するには、 ScheduleKeyDeletionオペレーションを使用します。KMS キーを指定するには、キー ID またはキー ARN を使用します。マルチリージョンキーを使用する場合は、明示的なリージョン値を持つキー ARN を使用することでエラーの発生率を減らすことができます。

例えば、このコマンドは、us-east-1 (米国東部 (バージニア北部)) リージョンからプライマリキーを削除します。コマンドでは待機期間が指定されていないため、待機期間はデフォルトの 30 日間に設定されます。

```
$ aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

コマンドが成功すると、キー ARN、結果のキーステータス、待機期間 (PendingWindowInDays) が返されます。

プライマリキーにレプリカがない場合、プライマリキーのキーステータスは PendingDeletion になり、出力には DeletionDate フィールドが含まれます。レプリカキーが残っている場合、プライマリキーのキーステータスは PendingReplicaDeletion になり、DeletionDate は不確実なため省略されます。レプリカキーの削除がスケジュールされている場合でも、スケジュールされた削除をキャンセルすることができます。

マルチリージョンのプライマリキーを削除するときは、キー ARN のキー ID とリージョン値が想定どおりの値であることを確認してください。

```
{
    "KeyId": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
    "KeyState": "PendingReplicaDeletion",
    "PendingWindowInDays": 30
}
```

KMS キーの削除ステータスを確認するには、プライマリキーまたは残りのレプリカキーで DescribeKeyオペレーションを使用します。プライマリキーの待機期間クロックは、最後のレプリカが削除され、キーステータスが PendingDeletion に変化するまでスタートしません。

プライマリキーの予定削除日を計算するには、レスポンスでレプリカキー ARN をループスルーし、それぞれに DescribeKey を実行して最新の DeletionDate 値を取得し、プライマリキーの PendingDeletionWindowInDays 値を追加します。レプリカキーの待機期間は同時に実行されます。

次の例では、KMS キーは、既存のレプリカキーを持つマルチリージョンのプライマリキーです。キーステータスが PendingReplicaDeletion のため、レスポンスには DeletionDate ではなく待機期間 (PendingWindowInDays) が含まれます。プライマリキーの実際の削除日は、レプリカキーがいつ削除されるかによって異なります。

```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1597902361.481,
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingReplicaDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
```

```
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "us-west-2"
                },
{
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                }
            ]
        },
        "PendingDeletionWindowInDays": 30
    }
}
```

すべてのレプリカが削除されると、DescribeKey 出力は、残っている、キーステータス PendingDeletion のプライマリキーを表示します。キーステータスが PendingDeletion の間、PendingWindowInDays フィールドの代わりに DeletionDate フィールドが表示されます。

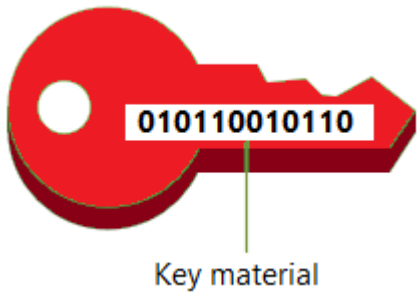
```
$ aws kms describe-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "",
        "CreationDate": 1597902361.481,
        "Enabled": false,
        "Description": "",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "PendingDeletion",
        "KeyUsage": "ENCRYPT_DECRYPT",
```

```
        "DeletionDate": 1597968000.0,
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": []
        }
    }
}
```


キーの AWS KMS キーマテリアルのインポート

自分で用意したキーマテリアルで AWS KMS keys (KMS キー) を作成できます。

KMS キーは、暗号化キーの論理表現です。KMS キーのメタデータには、データを暗号化および復号するために使用されるキーマテリアルの ID が含まれます。KMS キーを作成すると、デフォルトで AWS KMS がその KMS キーのキーマテリアルを生成します。ただし、キーマテリアルを使用せずに KMS キーを作成し、独自のキーマテリアルをその KMS キーにインポートできます。この機能は、「キーの持ち込み」(BYOK) とも呼ばれます。



> **Note**
>
> AWS KMS は AWS KMS、 AWS KMS 暗号化テキストがインポートされたキーマテリアルを持つ KMS キーで暗号化された場合でも、 の外部の暗号文の復号をサポートしていません。 AWS KMS は、このタスクに必要な暗号文形式を公開せず、形式が予告なく変更される場合があります。

インポートされたキーマテリアルは、カスタムキーストアの KMS キーを除くすべてのタイプの KMS キーでサポートされています。ただし、中国リージョンでは、KMS キーにインポートできるのは対称暗号化キーマテリアルのみです。

インポートされたキーマテリアルを使用する場合、 がキーマテリアルのコピーを使用 AWS KMS できるようにしながら、キーマテリアルに対する責任は引き続き負います。そうする場合の理由として次のものが考えられます (複数が組み合わさる場合もあります)。

- 要件を満たすエントロピーのソースを使用してキーマテリアルが生成されたことを証明するため。
- AWS サービスで独自のインフラストラクチャのキーマテリアルを使用し、 を使用して AWS KMS 内のそのキーマテリアルのライフサイクルを管理する AWS。
- コード署名、PKI 証明書署名 AWS KMS、証明書ピン留めアプリケーションのキーなど、 で既存の十分に確立されたキーを使用するには
- でキーマテリアルの有効期限を設定し、 AWS 手動で を削除するには、ただし、後で再び使用できるようにするため。これに対して、キー削除のスケジュールは、7 日から 30 日間の待機時間が必要となり、その後、削除された KMS キーは復元できません。
- キーマテリアルの元のコピーを所有し、キーマテリアルのライフサイクル全体における耐久性とディザスタリカバリを強化 AWS するために の外部に保持するため。
- 非対称キーと HMAC キーの場合、インポートすると、 の内部と外部で動作する互換性のある相互運用可能なキーが作成されます AWS。

インポートされたキーマテリアルを持つ KMS キーの使用と管理を監査およびモニタリングできます。 AWS KMS は、KMS キー の作成、ラップパブリックキーとインポートトークン のダウンロード、およびキーマテリアル のインポート時にイベントを AWS CloudTrail ログに記録します。 AWS KMS は、インポートされたキーマテリアルを手動で削除するか、 が AWS KMS 期限切れのキーマテリアル を削除するときにもイベントを記録します。

インポートされたキーマテリアルを持つ KMS キーと、 によって生成されたキーマテリアルを持つ KMS キーの重要な違いについては AWS KMS、「」を参照してくださいインポートしたキーマテリアルについて。

サポートされている KMS キー

AWS KMS は、次のタイプの KMS キーのインポートされたキーマテリアルをサポートします。カスタムキーストア内の KMS キーにキーマテリアルをインポートすることはできません。中国リージョンでは、キーマテリアルを対称暗号化キーにのみインポートできます。

- 対称暗号化 KMS キー
- 非対称 RSA KMS キー (暗号化用または署名用、両方は不可)
- 非対称楕円曲線 (ECC) KMS キー (署名のみ)
- HMAC KMS キー
- サポートされているすべてのタイプのマルチリージョンキー。

リージョン

インポートされたキーマテリアルは、 がサポート AWS リージョン する AWS KMS すべての でサポートされています。

中国リージョンでは、キーマテリアルを対称暗号化 KMS キーにのみインポートできます。また、キーマテリアルの要件は他のリージョンとは異なります。詳細については、「キーマテリアルのインポート ステップ 3: キーマテリアルを暗号化する」を参照してください。

トピック

- キーマテリアルのインポートを計画する
- インポートされたキーマテリアルの管理
- キーマテリアルをインポートするステップ 1: キーマテリアルなしで AWS KMS key を作成する
- キーマテリアルのインポート ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード
- キーマテリアルのインポート ステップ 3: キーマテリアルを暗号化する
- キーマテリアルのインポート ステップ 4: キーマテリアルのインポート

キーマテリアルのインポートを計画する

インポートされたキーマテリアルを使用すると、生成した暗号化キーで AWS リソースを保護できます。インポートするキーマテリアルは、特定の KMS キーに関連付けられます。同じキーマテリアルを同じ KMS キーに再インポートすることはできますが、異なるキーマテリアルを KMS キーにインポートすることはできず、インポートされたキーマテリアル用に設計された KMS キーをキーマテリアルを持つ KMS AWS KMS キーに変換することはできません。

詳細はこちら:

- the section called “ラップパブリックキーの仕様を選択”
- the section called “ラップアルゴリズムの選択”

トピック

- インポートしたキーマテリアルについて
- インポートされたキーマテリアルの保護
- キーマテリアルをインポートするためのアクセス許可
- インポートされたキーマテリアルの要件

インポートしたキーマテリアルについて

キーマテリアルを にインポートする前に AWS KMS、インポートされたキーマテリアルの次の特性を理解する必要があります。

キーのマテリアルを生成する

お客様の責任において、セキュリティ要件を満たすランダムのソースを使用して、キーマテリアルを生成する必要があります。

キーマテリアルを削除できます。

KMS キーから、インポートしたキーマテリアルを削除すると、KMS キーはただちに使用できなくなります。また、KMS キーにキーマテリアルをインポートするときに、キーの有効期間を確認して有効期限を設定することができます。有効期限に達すると、 AWS KMS はキーマテリアルを削除します。キーマテリアルがない場合、KMS キーは暗号化オペレーションで使用することはできません。キーを復元するには、キーに同じキーマテリアルを再インポートする必要があります。

キーマテリアルは変更できません

KMS キーにキーマテリアルをインポートすると、KMS キーはキーマテリアルに永続的に関連付けられます。同じキーマテリアルを再インポートすることはできますが、別のキーマテリアルをその KMS キーにインポートすることはできません。また、キーマテリアルがインポートされた KMS キーに対して、自動キーローテーションを有効にすることはできません。ただし、キーマテリアルがインポートされた KMS キーを手動でローテーションすることはできます。

キーマテリアルのオリジンは変更できません

インポートされたキーマテリアルのために設計されている KMS キーには、変更することができない EXTERNAL のオリジンの値があります。インポートされたキーマテリアルの KMS キーを変換して、 を含む他のソースのキーマテリアルを使用することはできません AWS KMS。同様に、キーマテリアルを含む KMS AWS KMS キーを、インポートされたキーマテリアル用に設計された KMS キーに変換することはできません。

キーマテリアルはエクスポートできません

インポートしたキーマテリアルをエクスポートすることはできません。インポートしたキーマテリアルをどのような形式でも返す AWS KMS ことはできません。インポートしたキーマテリアルのコピーは AWS、 の外部、できればハードウェアセキュリティモジュール (HSM) などのキーマネージャーに保持する必要があります。そうすれば、キーマテリアルを削除した場合や有効期限が切れた場合に再インポートできます。

インポートしたキーマテリアルを含むマルチリージョンキーを作成できます

インポートされたキーマテリアルを含むマルチリージョンには、インポートされたキーマテリアルの KMS キーと同様の機能があり、 AWS リージョン間の相互運用が可能です。キーマテリアルがインポートされたマルチリージョンキーを作成するには、同じキーマテリアルをプライマリ KMS キーと各レプリカキーにインポートする必要があります。詳細については、「キーマテリアルをマルチリージョンキーにインポートする」を参照してください。

非対称キーと HMAC キーは移植可能で相互運用可能です

非対称キーマテリアルと HMAC キーマテリアルを の外部で使用 AWS して、同じインポートされたキーマテリアルを持つ AWS KMS キーと相互運用できます。

アルゴリズムで使用される KMS キーに密接にバインドされている AWS KMS 対称暗号文とは異なり、 は暗号化、署名、MAC 生成に標準の HMAC 形式と非対称形式 AWS KMS を使用します。そのため、キーは移植可能で、従来のエスクローキーシナリオにも対応しています。

KMS キーにインポートされたキーマテリアルがある場合、 の外部 AWS でインポートされたキーマテリアルを使用して、次のオペレーションを実行できます。

- HMAC キー – キーマテリアルがインポートされた HMAC KMS キーによって生成された HMAC タグを検証できます。インポートされたキーマテリアルで HMAC KMS キーを使用して、 の外部でキーマテリアルによって生成された HMAC タグを検証することもできます AWS。
- 非対称暗号化キー — の外部でプライベート非対称暗号化キーを使用して AWS 、対応するパブリックキーを持つ KMS キーで暗号化された暗号文を復号できます。非対称 KMS キーを使用して、 の外部で生成された非対称暗号文を復号することもできます AWS。
- 非対称署名キー — インポートされたキーマテリアルで非対称署名 KMS キーを使用して、 の外部でプライベート署名キーによって生成されたデジタル署名を検証できます AWS。の外部で非対称パブリック署名キーを使用して AWS 、非対称 KMS キーによって生成された署名を検証することもできます。

同じキーマテリアルを同じ AWS リージョン内の別の KMS キーにインポートすると、それらのキーも相互運用できます。異なる で相互運用可能な KMS キーを作成するには AWS リージョン、インポートされたキーマテリアルを使用してマルチリージョンキーを作成します。

対称暗号化キーは移植も相互運用もできません

が AWS KMS 生成する対称暗号文は移植も相互運用もできません。 AWS KMS は移植に必要な対称暗号文形式を公開せず、形式は予告なく変更される場合があります。

- AWS KMS は、インポートしたキーマテリアルを使用する場合でも AWS、 の外部で暗号化した対称暗号文を復号できません。
- AWS KMS は、暗号化テキストがインポートされたキーマテリアルを持つ KMS キーで暗号化された場合でも AWS KMS、 の外部での AWS KMS 対称暗号文の復号をサポートしていません。
- インポートされた同じキーマテリアルの KMS キーは相互運用できません。が AWS KMS 生成する対称暗号文は、各 KMS キーに固有の暗号文です。この暗号文形式により、データを暗号化した KMS キーのみが復号できることが保証されます。

また、 AWS Encryption SDKや Amazon S3 クライアント側の暗号化 などの AWS ツールを使用して、 AWS KMS 対称暗号文を復号することはできません。

その結果、インポートされたキーマテリアルを持つキーを使用して、キーマテリアルへの条件付きアクセス権を持つ許可された第三者が の外部で特定の暗号文を復号できるキーエスクローの配置をサポートすることはできません AWS KMS。キーエスクローをサポートするには、AWS Encryption SDK を使用して、 AWS KMSに依存しないキーでメッセージを暗号化します。

可用性と耐久性に責任があります

AWS KMS は、インポートされたキーマテリアルの可用性を高めるように設計されています。ただし、 AWS KMS はインポートされたキーマテリアルの耐久性を、 が AWS KMS 生成するキーマテリアルと同じレベルで維持しません。詳細については、「インポートされたキーマテリアルの保護」を参照してください。

インポートされたキーマテリアルの保護

インポートしたキーマテリアルは、転送中も保管中も保護されます。キーマテリアルをインポートする前に、FIPS 140-2 暗号化モジュール検証プログラム で検証された AWS KMS ハードウェアセキュリティモジュール (HSMs「ラップ」) します。キーマテリアルをラッピングパブリックキーで直接暗号化することも、キーマテリアルを AES 対称キーで暗号化してから、AES 対称キーを RSA パブリックキーで暗号化することもできます。

受信時に、 は AWS KMS HSM 内の対応するプライベートキーを使用してキーマテリアルを復 AWS KMS 号し、HSM の揮発性メモリにのみ存在する AES 対称キーで再暗号化します。キーマテリアルがプレーンテキストで HSM 外に出ることはありません。これは、使用中および AWS KMS HSMs。

キーマテリアルがインポートされた KMS キーの使用は、KMS キーに設定したアクセス制御ポリシーによってのみ特定されます。さらに、エイリアスとタグを使用して、KMS キーを識別し、KMS キーへのアクセスを制御できます。キーを有効または無効にしたり、プロパティを表示および編集したり、 AWS CloudTrailのようなサービスを使用してキーをモニタリングしたりできます。

ただし、キーマテリアルのフェイルセーフコピーはお客様のみが管理します。この追加の制御基準とは対照的に、インポートされたキーマテリアルの耐久性と全体的な可用性に対する責任はお客様にあります。 AWS KMS は、インポートされたキーマテリアルの可用性を高めるように設計されています。ただし、 AWS KMS はインポートされたキーマテリアルの耐久性を、 が AWS KMS 生成するキーマテリアルと同じレベルで維持しません。

耐久性におけるこの相違は、次の場合に有意義です。

- インポートしたキーマテリアルの有効期限を設定すると、 は有効期限が切れた後にキーマテリアル AWS KMS を削除します。 AWS KMS は KMS キーまたはそのメタデータを削除しません。インポートされたキーマテリアルの有効期限が近づいたときに通知する Amazon CloudWatch アラームを作成できます。

が KMS キーに対して AWS KMS 生成するキーマテリアルを削除することはできません。また、AWS KMS キーマテリアルを期限切れに設定することはできません。ただし、 をローテーションすることはできます。

- インポートされたキーマテリアル を手動で削除すると、 はキーマテリアル AWS KMS を削除しますが、KMS キーまたはそのメタデータは削除しません。一方、キー削除のスケジュールは、7 日から 30 日間の待機期間を必要とし、その後、 AWS KMS は KMS キー、そのメタデータおよびキーマテリアルをすべて削除します。
- に影響する特定のリージョン全体の障害 AWS KMS （電力の損失など) が発生する可能性が低い場合 AWS KMS 、インポートしたキーマテリアルを自動的に復元することはできません。ただし、KMS キーとそのメタデータを復元 AWS KMS することはできます。

インポートしたキーマテリアルのコピーは、管理するシステムの の外部 AWS に保持する必要があります。インポートされたキーマテリアルのエクスポート可能なコピーを、HSM などのキーマネジメントシステムに保存しておくことをお勧めします。インポートしたキーマテリアルが削除されるか期限切れになった場合、同じキーマテリアルを再インポートするまで、関連付けられた KMS キーは使用できなくなります。インポートしたキーマテリアルが完全に失われた場合、KMS キーで暗号化された暗号文は回復できません。

キーマテリアルをインポートするためのアクセス許可

インポートされたキーマテリアルを使用して KMS キーを作成および管理するには、このプロセスのオペレーションに対するアクセス許可が必要です。KMS キーの作成時、キーポリシーに `kms:GetParametersForImport`、`kms:ImportKeyMaterial`、`kms:DeleteImportedKeyMaterial` のアクセス許可を付与できます。 AWS KMS コンソールでは、外部キーマテリアルオリジンを使用してキーを作成すると、キー管理者にこれらのアクセス許可が自動的に追加されます。

インポートされたキーマテリアルを持つ KMS キーを作成するには、プリンシパルに次の許可が必要です。

- kms:CreateKey (IAM ポリシー)
  - インポートされたキーマテリアルを持つ KMS キーにこのアクセス許可を制限するには、kms:KeyOrigin ポリシー条件を の値で使用しますEXTERNAL。

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
```

```
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
}
```

- kms:GetParametersForImport (キーポリシーまたは IAM ポリシー)
  - 特定のラッピングアルゴリズムとラッピングキー仕様を使用するリクエストにこのアクセス許可を制限するには、kms:WrappingAlgorithm および kms:WrappingKeySpec ポリシー条件を使用します。
- kms:ImportKeyMaterial (キーポリシーまたは IAM ポリシー)
  - 有効期限が切れるキーマテリアルを許可または禁止し、有効期限を制御するには、kms:ExpirationModel および kms:ValidTo ポリシー条件を使用します。

インポートされたキーマテリアルを再インポートするには、プリンシパルに kms:GetParametersForImport および kms:ImportKeyMaterial アクセス許可が必要です。

インポートされたキーマテリアルを削除するには、プリンシパルに kms:DeleteImportedKeyMaterial アクセス許可が必要です。

例えば、キーマテリアルがインポートされた KMS キーのすべての側面を管理する KMSAdminRole 許可をサンプルに付与するには、KMS キーのキーポリシーに次のようなキーポリシーステートメントを含めてください。

```
{
  "Sid": "Manage KMS keys with imported key material",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KMSAdminRole"
  },
  "Action": [
    "kms:GetParametersForImport",
    "kms:ImportKeyMaterial",
    "kms:DeleteImportedKeyMaterial"
  ]
}
```

インポートされたキーマテリアルの要件

インポートするキーマテリアルには、関連付けられた KMS キーのキー仕様との互換性が必要です。非対称キーペアの場合は、ペアのプライベートキーのみをインポートします。 AWS KMS はプライベートキーからパブリックキーを取得します。

AWS KMS は、インポートされたキーマテリアルを持つ KMS キーの次のキー仕様をサポートします。中国リージョンでは、インポートされたキーマテリアルは SYMMETRIC_DEFAULT キー仕様でのみサポートされます。

KMS キーのキー仕様	キーマテリアルの要件
対称暗号化キー SYMMETRIC_DEFAULT	256 ビット (32 バイト) のバイナリデータ 中国リージョンでは、128 ビット (16 バイト) のバイナリデータである必要があります。
HMAC キー HMAC_224 HMAC_256 HMAC_384 HMAC_512	HMAC キーマテリアルは RFC 2104 に準拠している必要があります。 キーの長さは、キー仕様で指定された長さと一致する必要があります。
RSA 非対称プライベートキー RSA_2048 RSA_3072 RSA_4096	インポートする RSA 非対称プライベートキーは、RFC 3447 に準拠したキーペアの一部である必要があります。 モジュラス: 2048 ビット、3072 ビット、または 4096 ビット 素数の数: 2 (マルチプライム RSA キーはサポートされていません) 非対称キーマテリアルは、RFC 5208 に準拠したパブリックキー暗号標準 (PKCS) #8 形式で

KMS キーのキー仕様	キーマテリアルの要件
	BER エンコードまたは DER エンコードする必要があります。
楕円曲線非対称プライベートキー ECC_NIST_P256 (secp256r1) ECC_NIST_P384 (secp384r1) ECC_NIST_P521 (secp521r1) ECC_SECG_P256K1 (secp256k1)	インポートする ECC 非対称プライベートキーは、RFC 5915 に準拠したキーペアの一部である必要があります。 曲線: NIST P-256、NIST P-384、NIST P-521、または Secp256k1 パラメータ: 名前付き曲線のみ (パラメータが明示された ECC キーは拒否されます) パブリックポイント座標: 圧縮、非圧縮、射影のいずれでも可 非対称キーマテリアルは、RFC 5208 に準拠したパブリックキー暗号標準 (PKCS) #8 形式で BER エンコードまたは DER エンコードする必要があります。

インポートされたキーマテリアルの管理

以下のトピックでは、キーマテリアルを KMS キーにインポートおよび再インポートする方法と、インポートされたキーマテリアルを自動的に有効期限が切れるように作成する方法について説明します。

トピック

- キーマテリアルのインポートの概要
- キーマテリアルの再インポート
- インポートされたキーマテリアルを含む KMS キーの識別
- インポートされたキーマテリアルの有効期限切れの CloudWatch アラームの作成
- インポートされたキーマテリアルの削除
- キーマテリアルがインポートされた KMS キーの削除

キーマテリアルのインポートの概要

次に、 AWS KMSにキーマテリアルをインポートする手順の概要を説明します。この手順の各ステップの詳細については、該当するトピックを参照してください。

1. キーマテリアルなしで KMS キーを作成 – オリジンは EXTERNAL である必要があります。のキーオリジンは、キーがインポートされたキーマテリアル用に設計されていることEXTERNALを示し、AWS KMS が KMS キーのキーマテリアルを生成できないようにします。後のステップで、この KMS キーに独自のキーマテリアルをインポートします。

   インポートするキーマテリアルは、関連付けられたキーの AWS KMS キー仕様と互換性がある必要があります。互換性の詳細については、「the section called “インポートされたキーマテリアルの要件”」を参照してください。
2. ラッピングパブリックキーとインポートトークンをダウンロード – ステップ 1 を完了した後、ラッピングパブリックキーとインポートトークンをダウンロードします。これらの項目は、 にインポートされている間、キーマテリアルを保護します AWS KMS。

   このステップでは、RSA ラッピングキーのタイプ (「キー仕様」) と、 AWS KMSへの転送中のデータの暗号化に使用するラッピングアルゴリズムを選択します。同じキーマテリアルをインポートまたは再インポートするたびに、異なるラッピングキー仕様とラッピングキーアルゴリズムを選択できます。
3. キーマテリアルを暗号化 – ステップ 2 でダウンロードしたラッピングパブリックキーを使用して、独自のシステムで作成したキーマテリアルを暗号化します。
4. キーマテリアルのインポート — 手順 3 で作成した暗号化されたキーマテリアルと、手順 2 でダウンロードしたインポートトークンをアップロードします。

   この段階で、オプションの有効期限を設定できます。インポートされたキーマテリアルの有効期限が切れると、 はそれ AWS KMS を削除し、KMS キーは使用できなくなります。この KMS キーを再度使えるようにするには、同じキーマテリアルを再インポートする必要があります。

   インポートオペレーションが正常に完了すると、KMS キーのキーステータスは `PendingImport` から `Enabled` に変化します。これで、KMS キーを暗号化オペレーションで使用できます。

AWS KMS は、KMS キー を作成し、ラップパブリックキー とインポートトークン をダウンロードし、キーマテリアル をインポートするときに、 AWS CloudTrail ログにエントリを記録します。は、インポートされたキーマテリアルを削除するとき、または が AWS KMS 期限切れのキーマテリアル を削除するときに AWS KMS もエントリを記録します。

キーマテリアルの再インポート

キーマテリアルがインポートされた KMS キーを管理する場合、キーマテリアルを再インポートしなければならない場合があります。キーマテリアルを再インポートして、有効期限の切れるキーマテリアルまたは削除されたキーマテリアルを置き換えるか、キーマテリアルの有効期限モデルまたは有効期限を変更することもできます。

KMS キーにキーマテリアルをインポートすると、KMS キーはキーマテリアルに永続的に関連付けられます。同じキーマテリアルを再インポートすることはできますが、別のキーマテリアルをその KMS キーにインポートすることはできません。キーマテリアルのローテーションはできません。AWS KMS は、キーマテリアルがインポートされた KMS キー用のキーマテリアルを作成できません。

キーマテリアルは、スケジュールがセキュリティ要件を満たしていれば、いつでも再インポートできます。キーマテリアルが有効期限に達するかそれに近づくのを待つ必要はありません。

キーマテリアルを再インポートするには、次のような例外を除いて、最初にキーマテリアルのインポートに使用したのと同じ手順を使用します。

- 新しい KMS キーを作成する代わりに、既存の KMS キーを使用します。インポートの手順のステップ 1 はスキップできます。
- キーマテリアルを再インポートするときに、有効期限モデルと有効期限を変更することができます。

KMS キーにキーマテリアルをインポートするたびに、KMS キーの新しいラップキーおよびインポートトークンをダウンロードして使用する必要があります。このラッピングの手順は、キーマテリアルの内容には影響しません。そのため、異なるラッピングキーおよび異なるラッピングアルゴリズムを使用して同じキーマテリアルをインポートできます。

インポートされたキーマテリアルを含む KMS キーの識別

キーマテリアルなしで KMS キーを作成する場合、KMS キーの `Origin` プロパティの値は `EXTERNAL` であり、変更することはできません。キーステータスと違って、`Origin` の値は、キーマテリアルの有無に依存しません。

`EXTERNAL` オリジン値を使用して、インポートされたキーマテリアル用に設計された KMS キーを識別することができます。キーオリジンは、 AWS KMS コンソールまたは DescribeKeyオペレーションを使用して見つけることができます。コンソールまたは API を使用して、有効期限が切れるかどうか、いつ失効するかなどの、キーマテリアルのプロパティを表示することもできます。

インポートされたキーマテリアルで KMS キーを識別するには (コンソール)

1. https://console.aws.amazon.com/kms で AWS KMS コンソールを開きます。
2. を変更するには AWS リージョン、ページの右上隅にあるリージョンセレクターを使用します。
3. 次のいずれかの方法を使用して、KMS キーの `Origin` プロパティを表示します。

   - KMS キーテーブルに [Origin] (オリジン) 列を追加するには、右上隅の、[Settings] (設定) アイコンを選択します。[Origin] (オリジン)、[Confirm] (確認) の順に選択します。[オリジン] 列によって、EXTERNAL (キーマテリアルのインポート) のオリジンプロパティ値を持つ KMS キーを簡単に識別できます。
   - 特定の KMS キーの `Origin` プロパティ値を検索するには、KMS キーのキー ID またはエイリアスを選択します。次に、[Cryptographic configuration] (暗号の設定) タブを選択します。これらのタブは、[General configuration] (一般設定) セクションの下にあります。
4. キーマテリアルに関する詳細情報を表示するには、[Key material] (キーマテリアル) タブを選択します。このタブは、インポートされたキーマテリアルを持つ KMS キーのみの詳細ページに表示されます。

インポートされたキーマテリアルを持つ KMS キーを識別するには (AWS KMS API)

DescribeKey 操作を使用します。レスポンスには、次の例に示すように KMS キーの `Origin` プロパティ、有効期限モデル、有効期限日が含まれます。

```
$  aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Origin": "EXTERNAL",
        "ExpirationModel": "KEY_MATERIAL_EXPIRES"
        "ValidTo": 2023-06-05T12:00:00+00:00,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": 2018-06-09T00:06:50.831000+00:00,
        "Enabled": false,
        "MultiRegion": false,
        "Description": "",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
```

```
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

インポートされたキーマテリアルの有効期限切れの CloudWatch アラームの作成

KMS キーにインポートされたキーマテリアルの有効期限が近づいたときに通知する CloudWatch アラームを作成できます。例えば、有効期限が 30 日未満になるとアラームが通知します。

キーマテリアルを KMS キーにインポートすると、キーマテリアルの有効期限の日時を任意で指定することができます。キーマテリアルの有効期限が切れると、 はキーマテリアル AWS KMS を削除し、KMS キーは使用できなくなります。KMS キーを再度使用するには、キーマテリアルを再インポートする必要があります。ただし、有効期限が切れる前にキーマテリアルを再インポートすると、その KMS キーを使用するプロセスが中断されるのを防ぐことができます。

このアラームは、インポートされたキーマテリアルの有効期限が切れる KMS キー CloudWatch に対して が に AWS KMS 発行する SecondsUntilKeyMaterialExpiresメトリクスを使用します。各アラームはこのメトリクスを使用して、特定の KMS キーのインポートされたキーマテリアルを監視します。キーマテリアルの有効期限が近づいているすべての KMS キーに対して単一のアラームを作成したり、将来作成する可能性のある KMS キーに対してアラームを作成したりすることはできません。

要件

インポートされたキーマテリアルの有効期限を監視する CloudWatch アラームには、次のリソースが必要です。

- 有効期限が切れる、インポートされたキーマテリアルを含む KMS キー。ヘルプについては、「インポートされたキーマテリアルを含む KMS キーの識別」を参照してください。
- Amazon SNS トピック 詳細については、「Amazon ユーザーガイド」の「Amazon SNS トピックの作成 CloudWatch 」を参照してください。

アラームの作成

「以下の必須値を使用して静的しきい値に基づいて CloudWatch アラームを作成する」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスの選択	[KMS] を選択し、次に [キーごとのメトリクス] を選択します。 KMS キーと SecondsUntilKeyMaterialExpires メトリクスを含む行を選択します。次に [Select metric] (メトリクスの選択) を選択します。 メトリクスリストには、インポートされたキーマテリアルが有効期限切れになっている KMS キーの SecondsUntilKeyMaterialExpires メトリクスのみが表示されます。アカウントとリージョンにこれらのプロパティを持つ KMS キーがない場合、このリストは空になります。
統計)	最小値
[Period] (期間)	1 分
しきい値タイプ	静的
Whenever ..。	######が 1 より大きい場合は必ず

インポートされたキーマテリアルの削除

インポートされたキーマテリアルは、KMS キーからいつでも削除できます。また、有効期限切れのインポートされたキーマテリアルの有効期限が切れると、 はキーマテリアル AWS KMS を削除します。どちらの場合も、キーマテリアルが削除されると、KMS キーのキーステータスがインポート保留中に変わり、同じキーマテリアルを再インポートするまで、KMS キーを暗号化オペレーションで使用することはできません。(別のキーマテリアルを KMS キーにインポートすることはできません)。

KMS キーの無効化とアクセス許可の取り消しに加えて、キーマテリアルの削除は、KMS キーの使用をすぐに、しかし一時的に停止する戦略として使用できます。これとは対照的に、キーマテリアルがインポートされた KMS キーの削除をスケジュールすることでも、KMS キーの使用をすぐに停止できます。ただし、待機期間中に削除をキャンセルしないと、KMS キー、キーマテリアル、およびすべてのキーメタデータは完全に削除されます。詳細については、「the section called “キーマテリアルがインポートされた KMS キーの削除”」を参照してください。

キーマテリアルを削除するには、 AWS KMS コンソールまたは DeleteImportedKeyMaterial API オペレーションを使用できます。インポートされたキーマテリアルを削除するとき AWS KMS 、および

AWS KMS が期限切れのキーマテリアルを削除するときに、 AWS CloudTrail ログにエントリを記録します。

トピック

- キーマテリアルの削除が AWS サービスに与える影響
- キー資料の削除 (コンソール)
- キーマテリアルの削除 (AWS KMS API)

キーマテリアルの削除が AWS サービスに与える影響

キーマテリアルを削除すると、キーマテリアルを持たない KMS キーはただちに使用できなくなります (結果整合性の影響を受ける)。ただし、KMS キーで保護されたデータキーにより暗号化されたリソースは、KMS キーがデータキーの復号化などで再び使用されるまでは影響を受けません。この問題は に影響します。その多くは AWS のサービス、データキーを使用して リソースを保護します。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

キー資料の削除 (コンソール)

を使用してキーマテリアル AWS Management Console を削除できます。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/kms で AWS Key Management Service （ AWS KMS) コンソールを開きます。
2. を変更するには AWS リージョン、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 次のいずれかを行います。
   - キーマテリアルがインポートされた KMS キーのチェックボックスをオンにします。[キーのアクション]、[キーマテリアルの削除] を選択します。
   - キーマテリアルがインポートされた KMS キーのエイリアスまたはキー ID を選択します。[キーマテリアル] タブを選択し、[キーマテリアルを削除] を選択します。
5. キーマテリアルを削除することを確認してから、[キーマテリアルの削除] を選択します。KMS キーのステータスに対応するそのキーステータスは、インポート保留中に変わります。

キーマテリアルの削除 (AWS KMS API)

AWS KMS API を使用してキーマテリアルを削除するには、 DeleteImportedKeyMaterialリクエストを送信します。次の例では、AWS CLI を使用してこのオペレーションを行う方法を示します。

1234abcd-12ab-34cd-56ef-1234567890ab を、削除する予定のキーマテリアルを持つ KMS キーのキー ID と置き換えます。KMS キーのキー ID または ARN を使用できますが、このオペレーションにエイリアスを使用することはできません。

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

キーマテリアルがインポートされた KMS キーの削除

キーマテリアルがインポートされた KMS キーのキーマテリアルの削除は一時的で、元に戻すことができます。キーを復元するには、キーマテリアルを再インポートします。

一方、KMS キーの削除は破棄できません。キーの削除をスケジュールし、必要な待機期間が終了すると、 は KMS キー、そのキーマテリアル、および KMS キーに関連付けられたすべてのメタデータ AWS KMS を完全に削除し、元に戻すことはできません。

ただし、キーマテリアルがインポートされた KMS キーを削除した場合のリスクと結果は、KMS キーのタイプ (「キー仕様」) によって異なります。

- 対称暗号化キー – 対称暗号化 KMS キーを削除すると、そのキーで暗号化された残りの暗号文はすべて回復できなくなります。同じキーマテリアルを使用しても、削除された対称暗号化 KMS キーの暗号文を復号できる新しい対称暗号化 KMS キーを作成することはできません。各 KMS キーに固有のメタデータは、各対称暗号文に暗号的にバインドされます。このセキュリティ機能により、対称暗号文を暗号化した KMS キーのみで復号できることが保証されますが、同等の KMS キーを再作成することができなくなっています。
- 非対称キーと HMAC キー — 元のキーマテリアルがある場合は、削除された非対称 KMS キーまたは HMAC KMS キーと同じ暗号化プロパティを持つ新しい KMS キーを作成できます。 は、一意のセキュリティ機能を含まない標準の RSA 暗号文と署名、ECC 署名、および HMAC タグ AWS KMS を生成します。また、HMAC キーまたは非対称キーペアのプライベートキーを AWSの外部で使用できます。

  同じ非対称キーマテリアルまたは HMAC キーマテリアルを使用して作成した新しい KMS キーには、異なるキー識別子が割り当てられます。新しいキーポリシーを作成し、エイリアスをすべて再作成し、新しいキーを参照するように既存の IAM ポリシーとアクセス許可を更新する必要があります。

キーマテリアルをインポートするステップ 1: キーマテリアルなしで AWS KMS key を作成する

デフォルトでは、KMS キーの作成時に AWS KMS がキーマテリアルを作成します。代わりに独自のキーマテリアルをインポートするには、最初にキーマテリアルなしで KMS キーを作成します。そして、キーマテリアルをインポートします。キーマテリアルなしで KMS キーを作成するには、 AWS KMS コンソールまたは CreateKeyオペレーションを使用します。

キーマテリアルなしでキーを作成するには、オリジンとして EXTERNAL を指定します。KMS キーのオリジンプロパティは変更不可です。いったん作成すると、インポートしたキーマテリアル用に設計された KMS キーを、AWS KMS または他のソースからキーマテリアルを含む KMS キーに変換することはできません。

EXTERNAL オリジンを持つ KMS キーのキーステータスであり、PendingImport であるキーマテリアルはありません。KMS キーは、無期限に PendingImport 状態を維持できます。ただし、暗号化オペレーションでは PendingImport 状態で KMS キーを使用することはできません。キーマテリアルをインポートすると、KMS キーのキーステータスが Enabled に変わり、暗号オペレーションで使用できるようになります。

AWS KMS は、KMS キー を作成し、パブリックキー とインポートトークン をダウンロードし、キーマテリアル をインポートするときに、AWS CloudTrailログにイベントを記録します。は、インポートされたキーマテリアルを削除するとき、または が期限切れのキーマテリアル AWS KMS を削除するときにAWS KMSも CloudTrail イベントを記録します。 ???

インポートしたキーマテリアルでマルチリージョンキーを作成する方法の詳細については、キーマテリアルをマルチリージョンキーにインポートする を参照してください。

トピック

- キーマテリアルなしで KMS キーを作成する (コンソール)
- キーマテリアルなしで KMS キーを作成する (AWS KMS API)

キーマテリアルなしで KMS キーを作成する (コンソール)

インポートされたキーマテリアルの KMS キーを 1 回作成する必要があるだけです。同じキーマテリアルを既存の KMS キーに必要な回数だけインポートおよび再インポートできますが、1 つの KMS キーに別のキーマテリアルをインポートすることはできません。詳細については、「ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード」を参照してください。

キーマテリアルがインポートされた既存の KMS キーを、[Customer managed keys] (カスタマーマネージドキー) テーブルで検索するには、右上隅にある歯車アイコンを使用して、KMS キーのリストの [Origin] (オリジン) 列を表示します。インポートされたキーの Origin 値は EXTERNAL (キーマテリアルのインポート) です。

キーマテリアルがインポートされた KMS キーを作成するには、まず基本的な手順に従って、目的のキータイプの KMS キーを作成します。ただし、次の例外があります。

キーの使用方法を選択したら、次の操作を行います。

1. [詳細オプション] を展開します。
2. [キーマテリアルのオリジン] で、[EXTERNAL (キーマテリアルのインポート)] を選択します。
3. インポートされたキーマテリアルの使用による影響について理解したことを示すため、[インポートされたキーの使用によるセキュリティと耐久性への影響について理解しました] の隣にあるチェックボックスをオンにします。これらの意味については、「インポートされたキーマテリアルの保護」を参照してください。
4. 基本的な手順に戻ります。基本的な手順の残りのステップは、そのタイプの KMS キーについてすべて同じです。

[完了] を選択すると、キーマテリアルがなく、ステータス (キーステータス) が [インポート保留中] の KMS キーが作成されたことになります。

ただし、[カスタマー管理型のキー] テーブルに戻るのではなく、コンソールには、キーマテリアルのインポートに必要なパブリックキーとインポートトークンをダウンロードできるページが表示されます。ここでダウンロードのステップを続行することも、[キャンセル] を選択してこの時点で停止することもできます。いつでもこのダウンロードのステップに戻ることができます。

次の手順: ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード

キーマテリアルなしで KMS キーを作成する (AWS KMS API)

AWS KMS API を使用して、キーマテリアルなしで対称暗号化 KMS キーを作成するには、`Origin`パラメータを に設定して CreateKeyリクエストを送信しますEXTERNAL。次の例では、AWS Command Line Interface (AWS CLI) を使用してこのオペレーションを行う方法を示します。

```
$ aws kms create-key --origin EXTERNAL
```

コマンドが成功した場合は、以下のような出力が表示されます。AWS KMS キーの `Origin` は `EXTERNAL`、その `KeyState` は `PendingImport` です。

> **Tip**
>
> コマンドが成功しない場合は、KMSInvalidStateException または NotFoundException が表示されることがあります。リクエストは再試行できます。

```
{
    "KeyMetadata": {
        "Origin": "EXTERNAL",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "Enabled": false,
        "MultiRegion": false,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "CreationDate": 1568289600.0,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

後のステップで使用するために、コマンド出力からの KeyId 値をコピーしてから、「ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード」に進みます。

> **Note**
>
> このコマンドは、SYMMETRIC_DEFAULT の KeySpec およびENCRYPT_DECRYPT の KeyUsage を含む対称暗号化 KMS キーを作成します。オプションのパラメータ --key-spec と --key-usage を使用して、非対称 KMS キーまたは HMAC KMS キーを作成できます。詳細については、CreateKey オペレーションを参照してください。

キーマテリアルのインポート ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード

キーマテリアル AWS KMS keyなしで を作成したら、AWS KMSコンソールまたは GetParametersForImport API を使用して、その KMS キーの RSA ラップパブリックキーとインポートトークンをダウンロードします。ラップパブリックキーとインポートトークンはセットであり分割できないので、組み合わせて使用する必要があります。

ラップパブリックキーは、転送用のキーマテリアルを暗号化するために使用します。ダウンロードする前に、RSA ラップキーペアの長さ (キー仕様) と、インポートしたキーマテリアルを暗号化してステップ 3 で転送するためのラップアルゴリズムを選択します。

各ラップパブリックキーとインポートトークンのセットは 24 時間有効です。それらを 24 時間以内にキーマテリアルのインポートに使用しなかった場合、新しいセットをダウンロードする必要があります。ラップパブリックキーとインポートトークンの新しいセットは、任意のタイミングでダウンロードが可能です。これにより、RSA ラップキーの長さ (キー仕様) を変更したり、紛失したセットを置き換えたりできます。

また、ラップパブリックキーとインポートトークンのセットをダウンロードして、KMS キーに同じキーマテリアルを 再インポートすることもできます。このオペレーションは、キーマテリアルの有効期限を設定もしくは変更したり、期限切れまたは削除済みのキーマテリアルを復元したりする場合に実行します。キーマテリアルは、AWS KMS にインポートするたびにダウンロードと暗号化を行なう必要があります。

ラップパブリックキーの使用

ダウンロードには、ラップパブリックキーとも呼ばれる、AWS アカウント に固有のパブリックキーも含まれます。

キーマテリアルをインポートする前に、ラップパブリックキーにより対象のキーマテリアルを暗号化してから、そのキーマテリアルを AWS KMS にアップロードします。暗号化されたキーマテリアルを受け取った AWS KMS では、対応するプライベートキーを使用してキーマテリアルを復号化した後に、AES対称キーによりそのキーマテリアルを再暗号化します。これらはすべて AWS KMS ハードウェアセキュリティモジュール (HSM) 内で実行されます。

インポートトークンの使用

ダウンロードには、キーマテリアルが正しくインポートされたことを保証するメタデータが付随する、インポートトークンが含まれています。AWS KMS に暗号化されたキーマテリアルをアッ

プロードする場合、このステップでダウンロードした同じインポートトークンをアップロードする必要があります。

ラップパブリックキーの仕様を選択

インポート中にキーマテリアルを保護するには、AWS KMS からダウンロードしたラップパブリックキーと、サポートされているラップアルゴリズムを使用してキーマテリアルを暗号化します。ラップパブリックキーとインポートトークンをダウンロードする前に、キー仕様を選択します。すべてのラップキーペアは、AWS KMS ハードウェアセキュリティモジュール (HSM) で生成されます。プレーンテキストのプライベートキーが、HSM の外部に出ることはありません。

ラップパブリックキーのキー仕様によって、AWS KMS への転送時にキーマテリアルを保護する RSA キーペアにおける、キーの長さが決まります。一般的には、実用的で最長のラップパブリックキーを使用することをお勧めします。各種の HSM やキーマネージャーをサポートするために、いくつかのラップパブリックキー仕様が提供されています。

AWS KMS では、特に明記されていない限り、すべてのタイプのキーマテリアルのインポートに使用される RSA ラップキーについて、以下の主要な仕様をサポートしています。

- RSA_4096 (推奨)
- RSA_3072
- RSA_2048

> **Note**
>
> ECC_NIST_P521 キーマテリアル、RSA_2048 パブリックラップキー仕様、および RSAES_OAEP_SHA_* ラップアルゴリズムの組み合わせはサポートされていません。RSA_2048 パブリックラップキーを使用して、ECC_NIST_P521 のキーマテリアルを直接ラップすることはできません。大きなラップキー、または RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムを使用してください。

ラップアルゴリズムの選択

インポート中にキーマテリアルを保護するには、ダウンロードしたラップパブリックキーと、サポートされているラップアルゴリズムを使用して、キーマテリアルを暗号化します。

AWS KMS は、複数の標準 RSA ラップアルゴリズムと、2 段階のハイブリッドラップアルゴリズムをサポートしています。基本的には、インポートしたキーマテリアルおよびラップキー仕様との互換性がある、最も安全なラップアルゴリズムを使用することをお勧めします。通常、ハードウェアセキュリティモジュール (HSM) がサポートするアルゴリズム、またはキーマテリアルを保護するキー管理システムを選択します。

次の表は、キーマテリアルと KMS キーの各タイプでサポートされているラップアルゴリズムを示しています。これらのアルゴリズムは、優先度順にリストされています。

キーマテリアル	サポートされるラップアルゴリズムと仕様
対称暗号化キー 256 ビット AES キー 128 ビット SM4 キー (中国リージョンのみ)	ラップアルゴリズム: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 非推奨となっているラップアルゴリズム: RSAES_PKCS1_V1 **Note** 2023 年 10 月 10 日現在、AWS KMS は RSAES_PKCS1_V1_5 ラップアルゴリズムをサポートしていません。 ラップキーの仕様: RSA_2048 RSA_3072 RSA_4096
非対称 RSA プライベートキー	ラップアルゴリズム: RSA_AES_KEY_WRAP_SHA_256

キーマテリアル	サポートされるラップアルゴリズムと仕様
	RSA_AES_KEY_WRAP_SHA_1 ラップキーの仕様: RSA_2048 RSA_3072 RSA_4096
非対称楕円曲線 (ECC) プライベートキー RSA_2048 ラップキー仕様の RSAES_OAEP_SHA_* ラップアルゴリズムを使用して、ECC_NIST_P521 のキーマテリアルをラップすることはできません。	ラップアルゴリズム: RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 ラップキーの仕様: RSA_2048 RSA_3072 RSA_4096
HMAC キー	ラップアルゴリズム: RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 ラップキーの仕様: RSA_2048 RSA_3072 RSA_4096

- RSA_AES_KEY_WRAP_SHA_256 – 生成した AES 対称キーでキーマテリアルを暗号化し、その後、ダウンロードした RSA パブリックラップキーと RSAES_OAEP_SHA_256 ラップアルゴリズムによりAES 対称キーを暗号化する、2 段階のハイブリッドラップアルゴリズム。

  RSA プライベートキーマテリアルをラップするには、RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムが必要です。
- RSA_AES_KEY_WRAP_SHA_1 – 生成した AES 対称キーでキーマテリアルを暗号化し、その後、ダウンロードした RSA ラップパブリックキーと RSAES_OAEP_SHA_1 ラップアルゴリズムにより AES 対称キーを暗号化する、2 段階のハイブリッドラップアルゴリズム。

  RSA プライベートキーマテリアルをラップするには、RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムが必要です。
- RSAES_OAEP_SHA_256 — SHA -256 ハッシュ関数を使用した最適な非対称暗号化パディング (OAEP) を使用する RSA 暗号化アルゴリズム。
- RSAES_OAEP_SHA_1 — SHA-1 ハッシュ関数を使用した最適な非対称暗号化パディング (OAEP) を使用する RSA 暗号化アルゴリズム。
- RSAES_PKCS1_V1_5 (廃止。2023 年 10 月 10 日現在、AWS KMS は RSAES_PKCS1_V1_5 ラップアルゴリズムをサポートしていません) – PKCS #1 バージョン 1.5 で定義されているパディング形式を使用した RSA 暗号化アルゴリズム。

トピック

- ラップパブリックキーとインポートトークンのダウンロード (コンソール)
- ラップパブリックキーとインポートトークンのダウンロード (AWS KMS API)

ラップパブリックキーとインポートトークンのダウンロード (コンソール)

AWS KMS コンソールを使用して、ラップパブリックキーとインポートトークンをダウンロードできます。

1. キーマテリアルなしで KMS キーを作成するステップを完了し、ラップキーとインポートトークンのダウンロードのページが開いている場合は、Step 9 に進みます。
2. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
3. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
4. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。

> **Tip**
>
> キーマテリアルは、EXTERNAL (キーマテリアルのインポート) の Origin を持つ KMS キーにのみインポートできます。これは、キーマテリアルなしで KMS キーが作成されたことを示します。テーブルに [オリジン] 列を追加するには、右上隅の設定アイコン ( ) を選択します。[オリジン] をオンにして、[確認] を選択します。

5. インポート保留中の KMS キーのエイリアスまたはキー ID を選択します。
6. [暗号構成] タブを選択し、その値を表示します。これらのタブは、[General configuration] (一般設定) セクションの下にあります。

   キーマテリアルは、EXTERNAL (キーマテリアルのインポート) の Origin として、KMS キーにのみインポートできます。インポートされたキーマテリアルで KMS キーを作成する方法の詳細については、キーの AWS KMS キーマテリアルのインポート を参照してください。
7. [キーマテリアル] タブを選択し、[キーマテリアルをインポート] を選択します。

   [キーマテリアル] タブは、EXTERNAL (キーマテリアルのインポート) のOrigin 値が指定された KMS キー に対してのみ表示されます。
8. [ラップキー仕様の選択] で、使用する KMS キーの設定を選択します。このキーの作成後は、キー仕様を変更することはできません。
9. [ラップアルゴリズムの選択] で、キーマテリアルの暗号化に使用するオプションを選択します。オプションの詳細については、「ラップアルゴリズムの選択」を参照してください。
10. [ラップパブリックキーとインポートトークンをダウンロード] を選択した後、ファイルを保存します。

    [次へ] オプションがある場合、今すぐプロセスを続行するには、[次へ] を選択します。後で続行するには、[キャンセル] を選択します。
11. 前のステップ (`Import_Parameters_<key_id>_<timestamp>`) で保存した `.zip` ファイルを解凍します。

    フォルダには以下のファイルが含まれています。

    - `WrappingPublicKey.bin` という名前のファイルにある、RSA ラップパブリックキー。
    - `ImportToken.bin` という名前のファイルにある、インポートトークン。

- README.txt という名前のテキストファイル。このファイルには、ラップパブリックキー、キーマテリアルの暗号化に使用するラップアルゴリズム、およびラップパブリックキーとインポートトークンの有効期限が切れる日時に関する情報が格納されています。

12. プロセスを続行する場合は、「キーマテリアルの暗号化」を参照してください。

ラップパブリックキーとインポートトークンのダウンロード (AWS KMS API)

パブリックキーとインポートトークンをダウンロードするには、 GetParametersForImport API を使用します。インポートされたキーマテリアルに関連付けられる KMS キーを指定します。この KMS キーには、EXTERNAL の Origin 値が必要です。

この例では、RSA_AES_KEY_WRAP_SHA_256 ラップアルゴリズム、RSA_3072 ラップパブリックキーの仕様、およびサンプルキー ID を規定します。これらのサンプル値は、実際のダウンロードのための有効な値に置き換えます。このオペレーションでは、キーの識別子としてキー ID またはキー ARN を使用できますが、エイリアス名やエイリアス ARN を使用することはできません。

```
$ aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSA_AES_KEY_WRAP_SHA_256 \
    --wrapping-key-spec RSA_3072
```

コマンドが成功した場合は、以下のような出力が表示されます。

```
{
    "ParametersValidTo": 1568290320.0,
    "PublicKey": "public key (base64 encoded)",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "ImportToken": "import token (base64 encoded)"
}
```

次のステップのためにデータを準備します。パブリックキーとインポートトークンを base64 でデコードし、デコードされた値をファイルに保存します。

パブリックキーとインポートトークンを base64 でデコードするには、次を実行します。

1. base64 でエンコードされたパブリックキー (この例の出力では *####### (base64 ######)*) をコピーして、新しいファイルに貼り付けて、ファイルを保存します。ファイルに PublicKey.b64 などの名前を付けます。

2. ファイル内容の base64 デコード、およびデコードされたデータの新しいファイルへの保存に、OpenSSL を使用します。次の例では、前のステップ (PublicKey.b64) で保存したファイルのデータをデコードして、WrappingPublicKey.bin という名前の新しいファイルに出力を保存します。

```
$ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
```

3. base64 でエンコードされたインポートトークン (この例の出力では *######### (base64 #######)*) をコピーして、新しいファイルに貼り付けて、ファイルを保存します。importtoken.b64 など、ファイルにわかりやすい名前を付けます。

4. ファイル内容の base64 デコード、およびデコードされたデータの新しいファイルへの保存に、OpenSSL を使用します。次の例では、前のステップ (ImportToken.b64) で保存したファイルのデータをデコードして、ImportToken.bin という名前の新しいファイルに出力を保存します。

```
$ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
```

ステップ 3: キーマテリアルを暗号化する に進みます。

キーマテリアルのインポート ステップ 3: キーマテリアルを暗号化する

パブリックキーとインポートトークンをダウンロードした後、ダウンロードしたパブリックキーと指定したラップアルゴリズムを使用してキーマテリアルを暗号化します。パブリックキーまたはインポートトークンを置き換える必要がある場合、あるいはラップアルゴリズムを変更する必要がある場合は、新しいパブリックキーとインポートトークンをダウンロードする必要があります。AWS KMS がサポートするパブリックキーとラップアルゴリズムについては、「ラップパブリックキーの仕様を選択」および「ラップアルゴリズムの選択」を参照してください。

キーマテリアルはバイナリ形式である必要があります。詳細については、「インポートされたキーマテリアルの要件」を参照してください。

> **Note**
>
> 非対称キーペアの場合は、プライベートキーのみを暗号化してインポートします。AWS KMS は、プライベートキーからパブリックキーを取得します。
> ECC_NIST_P521 キーマテリアル、RSA_2048 パブリックラップキー仕様、および RSAES_OAEP_SHA_* ラップアルゴリズムの組み合わせはサポートされていません。

RSA_2048 パブリックラップキーを使用して、ECC_NIST_P521 のキーマテリアルを直接ラップすることはできません。大きなラップキー、または RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムを使用してください。

通常、ハードウェアセキュリティモジュール (HSM) またはキー管理システムから エクスポートする場合、キーマテリアルを暗号化します。バイナリ形式でキーマテリアルをエクスポートする方法については、HSM またはキー管理システムに関するドキュメントを参照してください。OpenSSL を使用して、概念実証デモを提供する、次のセクションを参照できます。

キーマテリアルを暗号化する場合、パブリックキーとインポートトークンをダウンロードしたときに指定した、同じラップアルゴリズムを使用します。指定したラップアルゴリズムを確認するには、関連するGetParametersForImportリクエストの CloudTrail ログイベントを参照してください。

テスト用のキーマテリアルを生成

次の OpenSSL コマンドは、サポートされている各タイプのテスト用のキーマテリアルを生成します。これらの例は、テストと proof-of-concept デモンストレーションのみを目的としています。本稼働システムの場合、ハードウェアセキュリティモジュールやキー管理システムなど、より安全な方法を使用してキーマテリアルを生成します。

非対称キーペアのプライベートキーを DER でエンコードされた形式に変換するには、パイプを使ってキーマテリアル生成コマンドを次の `openssl pkcs8` コマンドに渡します。この `topk8` パラメータは、プライベートキーを入力として受け取り、PKCS#8 形式のキーを返すように OpenSSL に指示します。(デフォルトの動作は逆です)

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

次のコマンドは、サポートされている各キータイプのテストキーマテリアルを生成します。

- 対称暗号化キー (32 バイト)

  このコマンドは、256 ビット対称キー (32 バイトのランダム文字列) を生成し、`PlaintextKeyMaterial.bin` ファイルに保存します。このキーマテリアルをエンコードする必要はありません。

  ```
  openssl rand -out PlaintextKeyMaterial.bin 32
  ```

中国リージョンでのみ、128 ビット対称キー (16 バイトのランダム文字列) を生成する必要があります。

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- HMAC キー

  このコマンドは、指定したサイズのランダムバイト文字列を生成します。このキーマテリアルをエンコードする必要はありません。

  HMAC キーの長さは、KMS キーのキー仕様で定義されている長さと一致する必要があります。例えば、KMS キーが HMAC_384 の場合、384 ビット (48 バイト) キーをインポートする必要があります。

```
openssl rand -out HMAC_224_PlaintextKey.bin 28

openssl rand -out HMAC_256_PlaintextKey.bin 32

openssl rand -out HMAC_384_PlaintextKey.bin 48

openssl rand -out HMAC_512_PlaintextKey.bin 64
```

- RSA プライベートキー

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_2048_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:3072 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_3072_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_4096_PrivateKey.der
```

- ECC プライベートキー

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-256 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P256_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-384 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P384_PrivateKey.der
```

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-521 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P521_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp256k1 | openssl pkcs8 -
topk8 -outform der -nocrypt > ECC_SECG_P256K1_PrivateKey.der
```

OpenSSL によるキーマテリアルの暗号化の例

以下の例は、OpenSSL を使用して、ダウンロードしたパブリックキーでキーマテリアルを暗号化する方法を示しています。

> **Important**
>
> これらの例では、概念実証デモのみです。本稼働システムの場合、より安全な方法 (商用 HSM またはキー管理システムなど) を使用して、キーマテリアルを生成し、保存します。ECC_NIST_P521 キーマテリアル、RSA_2048 パブリックラップキー仕様、および RSAES_OAEP_SHA_* ラップアルゴリズムの組み合わせはサポートされていません。RSA_2048 パブリックラップキーを使用して、ECC_NIST_P521 のキーマテリアルを直接ラップすることはできません。大きなラップキー、または RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムを使用してください。

RSAES_OAEP_SHA_1

AWS KMS は、対称暗号化キー (SYMMETRIC_DEFAULT)、楕円曲線 (ECC) プライベートキー、および HMAC キー用の RSAES_OAEP_SHA_1 をサポートしています。

RSAES_OAEP_SHA_1 は RSA プライベートキーではサポートされていません。また、任意の RSAES_OAEP_SHA_* ラップアルゴリズムの RSA_2048 パブリックラップキーを使用して ECC_NIST_P521 (secp521r1) プライベートキーをラップすることはできません。より大きいサイズのパブリックラップキーまたは RSA_AES_KEY_WRAP ラップアルゴリズムを使用する必要があります。

次の例では、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_1 ラップアルゴリズムを使用してキーマテリアルを暗号化し、`EncryptedKeyMaterial.bin` ファイルに保存します。

この例では、以下のようになっています。

- *`WrappingPublicKey.bin`* は、ダウンロードしたラップパブリックキーを含むファイルです。

- *PlaintextKeyMaterial.bin* は、PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.b
  ど、暗号化するキーマテリアルを含むファイルです。

```
$ openssl pkeyutl \
    -encrypt \
    -in PlaintextKeyMaterial.bin \
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha1
```

RSAES_OAEP_SHA_256

AWS KMS は、対称暗号化キー (SYMMETRIC_DEFAULT)、楕円曲線 (ECC) プライベートキー、および HMAC キー用の RSAES_OAEP_SHA_256 をサポートしています。

RSAES_OAEP_SHA_256 は RSA プライベートキーではサポートされていません。また、任意の RSAES_OAEP_SHA_* ラップアルゴリズムの RSA_2048 パブリックラップキーを使用して ECC_NIST_P521 (secp521r1) プライベートキーをラップすることはできません。より大きいサイズのパブリックキーまたは RSA_AES_KEY_WRAP ラップアルゴリズムを使用する必要があります。

次の例では、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_256 ラップアルゴリズムを使用してキーマテリアルを暗号化し、EncryptedKeyMaterial.bin ファイルに保存します。

この例では、以下のようになっています。

- *WrappingPublicKey.bin* は、ダウンロードしたパブリックラップキーを含むファイルです。コンソールからパブリックキーをダウンロードした場合、このファイルの名前は wrappingKey_*KMS key_key_ID_timestamp* (例えば、wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909) となります。
- *PlaintextKeyMaterial.bin* は、PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.b
  ど、暗号化するキーマテリアルを含むファイルです。

```
$ openssl pkeyutl \
```

```
    -encrypt \
    -in PlaintextKeyMaterial.bin \
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

RSA_AES_KEY_WRAP_SHA_1

RSA_AES_KEY_WRAP_SHA_1 ラップアルゴリズムには 2 つの暗号化オペレーションオペレーションが含まれています。

1. 生成した AES 対称キーと AES 対称暗号化アルゴリズムを使用してキーマテリアルを暗号化します。
2. 使用した AES 対称キーを、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_1 ラップアルゴリズムで暗号化します。

AWS KMS は、サポートされているすべてのタイプのインポートされたキーマテリアルとサポートされているすべてのパブリックキー仕様の RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムをサポートします。RSA_AES_KEY_WRAP_SHA_* アルゴリズムは、RSA キーマテリアルのラップでサポートされている唯一のラップアルゴリズムです。

RSA_AES_KEY_WRAP_SHA_1 ラップアルゴリズムには OpenSSL バージョン 3.x 以降が必要です。

1. 256 ビット AES 対称暗号化キーを生成する

   このコマンドは、256 ランダムビットで構成される AES 対称暗号化キーを生成し、`aes-key.bin` ファイルに保存します

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. AES 対称暗号化キーを使用してキーマテリアルを暗号化する

   このコマンドは、AES 対称暗号化キーを使用してキーマテリアルを暗号化し、暗号化されたキーマテリアルを `key-material-wrapped.bin` ファイルに保存します。

このコマンド例では、

- *PlaintextKeyMaterial.bin* は、PlaintextKeyMaterial.bin、HMAC_384_PlaintextK ど、インポートするキーマテリアルを含むファイルです。
- *aes-key.bin* は、前のコマンドで生成した 256 ビット AES 対称暗号化キーを含むファイルです。

```
# Encrypt your key material with the AES symmetric encryption key
$ openssl enc -id-aes256-wrap-pad \
        -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
        -iv A65959A6 \
        -in PlaintextKeyMaterial.bin\
        -out key-material-wrapped.bin
```

3. AES 対称暗号化キーをパブリックキーで暗号化する

このコマンドは、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_1 ラップアルゴリズムで AES 対称暗号化キーを暗号化し、DER エンコードして aes-key-wrapped.bin ファイルに保存します。

このコマンド例では、

- *WrappingPublicKey.bin* は、ダウンロードしたパブリックラップキーを含むファイルです。コンソールからパブリックキーをダウンロードした場合、このファイルの名前は wrappingKey_*KMS key_key_ID_timestamp* (例えば、wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909) となります
- *aes-key.bin* は、このシーケンスの例の最初のコマンドで生成した 256 ビット AES 対称暗号化キーを含むファイルです。

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
    -encrypt \
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
```

```
    -pkeyopt rsa_oaep_md:sha1 \
    -pkeyopt rsa_mgf1_md:sha1
```

4. インポートするファイルを生成する

   暗号化されたキーマテリアルを含むファイルと暗号化された AES キーを含むファイルを連結します。これらを EncryptedKeyMaterial.bin ファイルに保存します。このファイルは、ステップ 4: キーマテリアルのインポート にインポートするファイルです。

   このコマンド例では、

   - *key-material-wrapped.bin* は、暗号化されたキーマテリアルを含むファイルです。
   - *aes-key-wrapped.bin* は、暗号化された AES 暗号化キーを含むファイルです。

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

RSA_AES_KEY_WRAP_SHA_256

RSA_AES_KEY_WRAP_SHA_256 ラップアルゴリズムには 2 つの暗号化手順が含まれています。

1. 生成した AES 対称キーと AES 対称暗号化アルゴリズムを使用してキーマテリアルを暗号化します。
2. 使用した AES 対称キーを、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_256 ラップアルゴリズムで暗号化します。

AWS KMS は、サポートされているすべてのタイプのインポートされたキーマテリアルとサポートされているすべてのパブリックキー仕様の RSA_AES_KEY_WRAP_SHA_* ラップアルゴリズムをサポートします。RSA_AES_KEY_WRAP_SHA_* アルゴリズムは、RSA キーマテリアルのラップでサポートされている唯一のラップアルゴリズムです。

RSA_AES_KEY_WRAP_SHA_256 ラップアルゴリズムには OpenSSL バージョン 3.x 以降が必要です。

1. 256 ビット AES 対称暗号化キーを生成する

   このコマンドは、256 ランダムビットで構成される AES 対称暗号化キーを生成し、aes-key.bin ファイルに保存します

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. AES 対称暗号化キーを使用してキーマテリアルを暗号化する

   このコマンドは、AES 対称暗号化キーを使用してキーマテリアルを暗号化し、暗号化されたキーマテリアルを key-material-wrapped.bin ファイルに保存します。

   このコマンド例では、

   - *PlaintextKeyMaterial.bin* は、PlaintextKeyMaterial.bin、HMAC_384_PlaintextK ど、インポートするキーマテリアルを含むファイルです。
   - *aes-key.bin* は、前のコマンドで生成した 256 ビット AES 対称暗号化キーを含むファイルです。

   ```
   # Encrypt your key material with the AES symmetric encryption key
   $ openssl enc -id-aes256-wrap-pad \
           -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
           -iv A65959A6 \
           -in PlaintextKeyMaterial.bin\
           -out key-material-wrapped.bin
   ```

3. AES 対称暗号化キーをパブリックキーで暗号化する

   このコマンドは、ダウンロードしたパブリックキーと RSAES_OAEP_SHA_256 ラップアルゴリズムで AES 対称暗号化キーを暗号化し、DER エンコードして aes-key-wrapped.bin ファイルに保存します。

   このコマンド例では、

   - *WrappingPublicKey.bin* は、ダウンロードしたパブリックラップキーを含むファイルです。コンソールからパブリックキーをダウンロードした場合、このファイルの名前は wrappingKey_*KMS key_key_ID_timestamp* (例えば、wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909) となります

- *aes-key.bin* は、このシーケンスの例の最初のコマンドで生成した 256 ビット AES 対称暗号化キーを含むファイルです。

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
    -encrypt \
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

4. インポートするファイルを生成する

   暗号化されたキーマテリアルを含むファイルと暗号化された AES キーを含むファイルを連結します。これらを EncryptedKeyMaterial.bin ファイルに保存します。このファイルは、ステップ 4: キーマテリアルのインポート にインポートするファイルです。

   このコマンド例では、

   - *key-material-wrapped.bin* は、暗号化されたキーマテリアルを含むファイルです。
   - *aes-key-wrapped.bin* は、暗号化された AES 暗号化キーを含むファイルです。

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

ステップ 4: キーマテリアルのインポート に進みます。

キーマテリアルのインポート ステップ 4: キーマテリアルのインポート

キーマテリアルを暗号化すると、キーマテリアルをインポートして AWS KMS key で使用できます。キーマテリアルをインポートするには、ステップ 3: キーマテリアルを暗号化する から暗号化されたキーマテリアルと、ステップ 2: ラップパブリックキーおよびインポートトークンのダウンロード でダウンロードしたインポートトークンをアップロードします。パブリックキーとインポートトークンをダウンロードしたときに指定したものと同じ KMS キーに、キーマテリアルをインポート

する必要があります。キーマテリアルが正常にインポートされると、KMS キーのキーステータスが Enabled に変化し、暗号化オペレーションで KMS キーを使用できるようになります。

キーマテリアルをインポートする場合、キーマテリアルの有効期限をオプションで設定できます。キーマテリアルが有効期限切れになると、AWS KMS はキーマテリアルを削除し、KMS キーは使用不可能になります。暗号化オペレーションで KMS キーを使用するには、同じキーマテリアルを再インポートする必要があります。キーマテリアルをインポートした後は、現在のインポートの有効期限を設定、変更、またはキャンセルできません。これらの値を変更するには、同じキーマテリアルを削除または再インポートする必要があります。

キーマテリアルをインポートするには、 AWS KMSコンソールまたは ImportKeyMaterial API を使用できます。HTTP リクエストを作成することにより、または AWS SDKs、AWS Command Line Interface または AWS Tools for PowerShell を使用することにより、直接 API を使用できます。

キーマテリアルをインポートすると、AWS CloudTrailログにImportKeyMaterialエントリが追加され、ImportKeyMaterialオペレーションが記録されます。 CloudTrail エントリは、AWS KMSコンソールと AWS KMS API のどちらを使用する場合でも同じです。

有効期限の設定 (オプション)

KMS キーのキーマテリアルをインポートするとき、オプションで、キーマテリアルの有効期限 (日付と時刻) をインポートした日から最大 365 日間までの範囲で設定できます。インポートされたキーマテリアルの有効期限が過ぎると、AWS KMS はこれを削除します。このアクションにより KMS キーのキーキステータスは PendingImport に変更され、暗号化オペレーションで使用できなくなります。KMS キーを使用する場合は、元のキーマテリアルを再度インポートする必要があります。

インポートされたキーマテリアルが頻繁に期限切れになるようにすれば規制要件を満たすのには役立ちますが、KMS キーで暗号化されたデータのリスクが高まります。元のキーマテリアルのコピーを再インポートするまでは、キーマテリアルの期限が切れた KMS キーは使用できず、KMS キーで暗号化されたデータにはアクセスできなくなります。元のキーマテリアルのコピーを紛失するなど何らかの理由でキーマテリアルを再インポートしなければ、その KMS キーは永久に使用できなくなり、その KMS キーで暗号化されたデータは回復不能となります。

こうしたリスクを取り除くには、インポートされたキーマテリアルのコピーにアクセスできることを確認してから、キーマテリアルの有効期限が切れて AWS ワークロードが中断されることになる前に、キーマテリアルを削除し再インポートするシステムを設計します。インポートされたキーマテリアルの、有効期限のアラームを設定し、期限切れになる前に、キーマテリアルを再インポートする十分な時間を確保しておくことが推奨されます。 CloudTrail ログを使用して、キーマテリアル をインポート (および再インポート) してインポートされたキーマテリアル を削除する監査オペレーショ

ン、および期限切れのキーマテリアル を削除する AWS KMSオペレーションを実行することもできます。

KMS キーに異なるキーマテリアルをインポートすることはできません。また AWS KMS は、削除されたキーマテリアルを復元、回復、再作成することもできません。有効期限を設定する代わりに、インポートしたキーマテリアルをプログラムで定期的に削除し再インポートすることができますが、元のキーマテリアルのコピーを保持するための要件は同じです。

キーマテリアルをインポートするときに、インポートしたキーマテリアルの有効期限が切れているかどうか、および、いつ切れるのかを判断します。ただし、キーマテリアルを削除して再インポートすれば、有効期限を有効または無効にしたり、新しい有効期限を設定したりできます。の `ExpirationModel`パラメータImportKeyMaterialを使用して有効期限をオン (`KEY_MATERIAL_EXPIRES`) およびオフ (`KEY_MATERIAL_DOES_NOT_EXPIRE`) にし、 `ValidTo`パラメータを使用して有効期限を設定します。最大日数はインポートした日から 365 日目です。最小日数はありませんが、時刻は未来の時刻でなければなりません。

キーマテリアルのインポート (コンソール)

AWS Management Console を使用して、キーマテリアルをインポートできます。

1. 「ラップされたキーマテリアルのアップロード」のページが表示されている場合は、Step 8 に進んでください。
2. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
3. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
4. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
5. 公開キーとインポートトークンをダウンロードした KMS キーの、キー ID またはエイリアスを選択します。
6. [暗号構成] タブを選択し、その値を表示します。タブは、KMS キーの詳細ページの一般設定セクションにあります。

   キーマテリアルは、EXTERNAL (キーマテリアルのインポート) の Origin を含む KMS キーにのみインポートできます。インポートされたキーマテリアルで KMS キーを作成する方法の詳細については、キーの AWS KMS キーマテリアルのインポート を参照してください。
7. [キーマテリアル] タブを選択し、[キーマテリアルをインポート] を選択します。[キーマテリアル] タブは、Origin 値が EXTERNAL (キーマテリアルのインポート) の KMS キーに対してのみ表示されます。

キーマテリアルをダウンロードし、トークンをインポートし、キーマテリアルを暗号化した場合は、[次へ] を選択します。

8. [暗号化されたキーマテリアルとインポートトークン] セクションで、次の操作を行います。

   a. [ラップされたキーマテリアル] で、[ファイルを選択] を選択します。次に、ラップされた (暗号化された) キーマテリアルを含むファイルをアップロードします。

   b. [トークンのインポート] で、[ファイルを選択] を選択します。ダウンロードしたインポートトークンを含むファイルをアップロードします。

9. [有効期限オプション] セクションで、キーマテリアルの有効期限が切れているかどうかを判断します。有効期限の日時を設定するには、[キーマテリアルの有効期限] を選択し、カレンダーを使用して日付と時刻を選択します。現在の日付から 365 日後を上限として、日付を指定できます。

10. [キーマテリアルのアップロード] を選択します。

キーマテリアルをインポートする (AWS KMS API)

キーマテリアルをインポートするには、 ImportKeyMaterialオペレーションを使用します。次の例では AWS CLI を使用しますが、サポートされているすべてのプログラミング言語を使用できます。

この例を使用するには:

1. *1234abcd-12ab-34cd-56ef-1234567890ab* を、公開キーとインポートトークンをダウンロードしたときに指定した KMS キーのキー ID と置き換えます。KMS キーを識別するには、そのキー ID またはキー ARN を使用します。このオペレーションにエイリアス名やエイリアス ARN を使用することはできません。

2. *EncryptedKeyMaterial.bin* を、暗号化されたキーマテリアルを含むファイル名に置き換えます。

3. *ImportToken.bin* を、インポートトークンを含むファイル名に置き換えます。

4. インポートしたキーマテリアルを有効期限切れにする場合は、`expiration-model` パラメータの値をデフォルトの `KEY_MATERIAL_EXPIRES` に変更するか、`expiration-model` パラメータを省略します。次に `valid-to` パラメータの値を、キーマテリアルを有効期限切れにする日時に置き換えます。リクエストの時点から 365 日後までの日時を設定できます。

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
```

```
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2023-06-17T12:00:00-08:00
```

インポートしたキーマテリアルを有効期限切れにしたくない場合は、expiration-model パラメータを KEY_MATERIAL_DOES_NOT_EXPIRE に設定し、コマンドの valid-to パラメータを省略します。

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_DOES_NOT_EXPIRE
```

> **Tip**
>
> コマンドが成功しない場合は、KMSInvalidStateException または NotFoundException が表示されることがあります。リクエストは再試行できます。

カスタムキーストア

キーストア は、暗号化キーを保存するための安全な場所です。AWS KMS のデフォルトのキーストアは、保存しているキーを生成および管理するためのメソッドもサポートしています。デフォルトでは、AWS KMS で作成される AWS KMS keys の暗号化キーマテリアルは、FIPS 140-2 で検証された暗号化モジュールであるハードウェアセキュリティモジュール (HSM) によって、生成され保護されます。KMS キーのキーマテリアルは、暗号化されずに HSM の外に出ることはありません。

ただし、HSM の管理をさらに強化する必要がある場合は、カスタムキーストアを作成します。

カスタムキーストアとは AWS KMS の内部にある論理キーストアのことで、ユーザーが所有し管理している AWS KMS の、外部のキーマネージャーがバックアップしています。カスタムキーストアには、AWS KMS のキー管理のための便利な包括的インターフェイスに、キーマテリアルと暗号化オペレーションを所有し制御する機能が統合されています。カスタムキーストアで KMS キーを使用する場合、暗号化のオペレーションは、ユーザーのキーマネージャーが、ユーザーの暗号化キーを使用して実行します。それにより、暗号化キーの可用性と耐久性、および HSM のオペレーションに対するユーザーの責任が増えます。

AWS KMS は、2 種類のカスタムキーストアをサポートしています。

- AWS CloudHSM キーストアは、AWS CloudHSM クラスターにバックアップされた AWS KMS カスタムキーストアです。AWS CloudHSM キーストアで KMS キーを作成すると、AWS KMS が、256 ビットの永続的でエクスポート不可の Advanced Encryption Standard (AES) 対称キーを、関連する AWS CloudHSM クラスターに作成します。このキーマテリアルは、暗号化されずに AWS CloudHSM クラスターの外に出ることはありません。AWS CloudHSM キーストアで KMS キーを使用すると、暗号化オペレーションはクラスター内の HSM で実行されます。AWS CloudHSM クラスターは、FIPS 140-2 Level 3 で認定されたハードウェアセキュリティモジュール (HSM) によってバックアップされます。
- 外部キーストアは、AWS の外部でユーザーが所有し管理している外部キーマネージャーによってバックアップされた AWS KMS カスタムキーストアです。外部キーストアで KMS キーを使用すると、すべての暗号化および復号化のオペレーションは、外部のキーマネージャーによって、ユーザーの暗号化キーを使用して実行されます。外部キーストアは、さまざまなベンダーの、さまざまな外部キーマネージャーをサポートするように設計されています。

  AWS KMS は、外部のキーマネージャーまたは暗号化キーを直接閲覧したり、それらにアクセスしたり、それらを操作したりすることはありません。外部キーストアの KMS キーを使って暗号化または複合化を行うとき、オペレーションは、外部キーマネージャーによって、ユーザーの外部キーを使用して実行されます。ユーザーは、AWS を操作することなく暗号化のオペレーションを拒否または中止できることを含め、暗号化キーに対する完全な制御を保持します。ただし、その距離と処理の追加とにより、外部キーストアの KMS キーを使用するとレイテンシーとパフォーマンスが低下し、AWS KMS でキーマテリアルを持つ KMS キーを使用する場合に比べ、可用性に変化が出る場合があります。AWS KMS 外部キーストア機能と互換性のあるキーマネージャーに関する詳細は、「AWS Key Management Service のよくある質問」の「XKS Proxy 仕様をサポートしている外部ベンダーは?」を参照してください。

これら 2 種類のカスタムキーストアは、標準の AWS KMS キーストアとはまったく異なり、互いにもまったく異なります。それぞれのセキュリティモデル、責任の所在、パフォーマンス、料金、ユースケースもまた、大きく異なります。カスタムキーストアを選択するときは、事前に関連のドキュメントをよく読み、制御を増やせばそれだけ設定とメンテナンスの責任も増えることを確認します。ただし、オペレーションを行う際のルールや規制によりキーマテリアルを直接管理する必要がある場合は、カスタムキーストアを使用するのが良いでしょう。

サポートされていない機能

AWS KMS はカスタムキーストアで次の機能をサポートしていません。

- 非対称 KMS キー

- 非対称データキーペア
- HMAC KMS キー
- インポートされたキーマテリアルを持つ KMS キー
- 自動キーローテーション
- マルチリージョンキー

トピック

- AWS CloudHSM キーストア
- 外部キーストア

AWS CloudHSM キーストア

AWS CloudHSM キーストアは、AWS CloudHSM クラスターを基盤とするカスタムキーストアです。AWS KMS keyカスタムキーストア内にを作成すると、KMS キー用の抽出不可能なキーマテリアルを、 AWS KMS AWS CloudHSM 所有および管理するクラスターに生成して保存します。カスタムキーストアで KMS キーを使用する際、暗号化オペレーションはクラスター内の HSM で実行されます。この機能は、 AWS KMS の便利で広範囲にわたる統合と、 AWS CloudHSM クラスター内のクラスターの制御の強化を組み合わせたものです。 AWS アカウント

AWS KMS カスタムキーストアの作成、使用、管理をコンソールと API で完全にサポートします。任意の KMS キーを使用するのと同じ方法で、カスタムキーストア内の KMS キーを使用できます。例えば、KMS キーを使用して、データキーの生成やデータの暗号化を実行できます。カスタムキーストアの KMS キーは、 AWS カスタマーマネージドキーをサポートするサービスでも使用できます。

カスタムキーストアは必要ですか?

ほとんどのユーザーにとって、FIPS 140-2 AWS KMS で検証済みの暗号モジュールで保護されているデフォルトのキーストアは、ユーザーのセキュリティ要件を満たしています。メンテナンス責任や追加サービスへの依存の強化は必要ありません。

ただし、組織に次のいずれかの要件がある場合、カスタムキーストアの作成を検討してください。

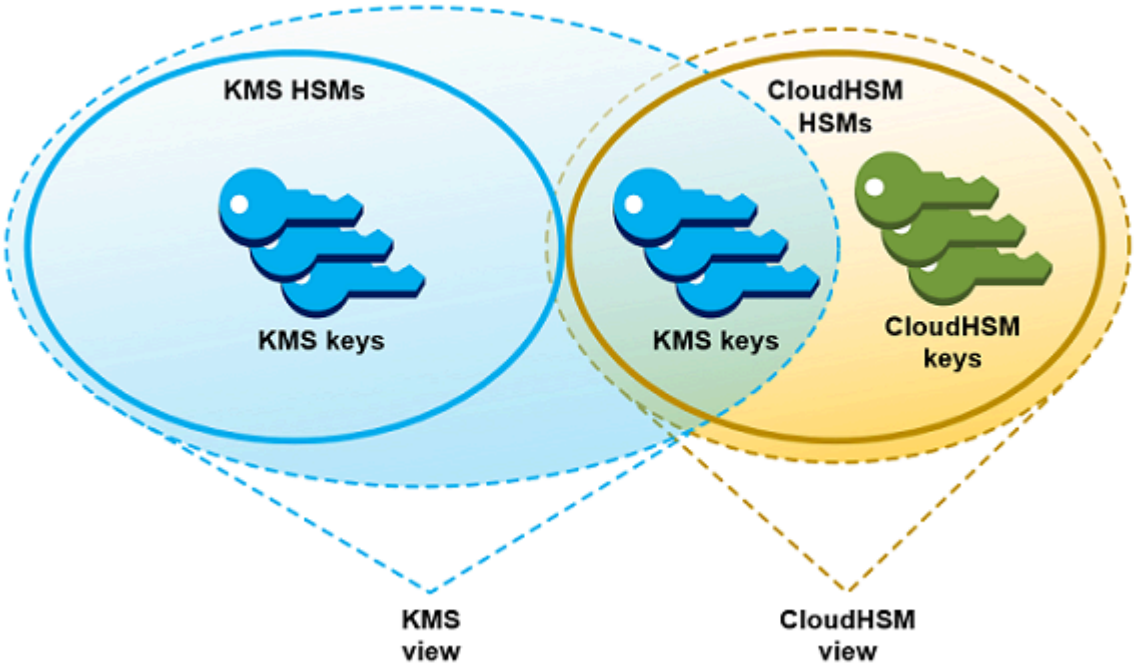
- 単一テナント HSM または直接制御できる HSM における保護が明示的に要求されているキーがある。
- キーマテリアルをからすぐに削除できる必要があります。 AWS KMS

- キーの使用状況は、 AWS KMS またはとは別にすべて監査できる必要があります AWS CloudTrail。

カスタムキーストアはどのように機能しますか?

AWS CloudHSM 各カスタムキーストアは内のクラスタに関連付けられています AWS アカウント。カスタムキーストアをそのクラスターに接続すると、 AWS KMS 接続をサポートするネットワークインフラストラクチャが作成されます。次に、クラスター内の専用暗号ユーザーの認証情報を使用して、 AWS CloudHSM クラスター内のキークライアントにログインします。

AWS KMS カスタムキーストアはで作成および管理し、HSM AWS CloudHSMクラスターはで作成および管理します。 AWS KMS keys AWS KMS カスタムキーストアで作成すると、で KMS キーを表示および管理します。 AWS KMSただし、クラスター内の他のキーと同様に AWS CloudHSM、でキーマテリアルを表示および管理することもできます。



AWS KMS カスタムキーストアで生成されたキーマテリアルを使用して、対称暗号化 KMS キーを作成できます。次に、キーストアの KMS キーに使用するのと同じ手法を使用して、カスタムキーストアの KMS キーを表示および管理します。 AWS KMS IAM ポリシーとキーポリシーを使用してアクセスを制御したり、タグやエイリアスを作成したり、KMS キーを有効および無効にしたり、キーの削除をスケジューリングルしたりできます。KMS キーは暗号化操作に使用でき、 AWS と統合するサービスでも使用できます。 AWS KMS

さらに、HSM の作成や削除、バックアップの管理など、 AWS CloudHSM クラスターを完全に制御できます。 AWS CloudHSM クライアントとサポートされているソフトウェアライブラリを使用して、KMS キーのキーマテリアルを表示、監査、管理できます。 AWS KMS カスタムキーストアが切断されている間はアクセスできず、ユーザーはカスタムキーストアの KMS キーを暗号化操作に使用できません。こうした制御の強化により、カスタムキーストアがそれを必要とする組織のための強力なソリューションとなります。

開始方法

AWS CloudHSM キーストアを作成および管理するには、およびの機能を使用します。 AWS KMS AWS CloudHSM

1. から始めましょう AWS CloudHSM。アクティブな AWS CloudHSM クラスターを作成するか、既存のクラスターを選択します。クラスターには、少なくとも 2 つの異なるアベイラビリティーゾーンのアクティブな HSM が必要です。次に、 AWS KMSのクラスターで専用 crypto user (CU) アカウントを作成します。
2. で AWS KMS、AWS CloudHSM 選択したクラスタに関連するカスタムキーストアを作成します。AWS KMS には、カスタムキーストアを作成、表示、編集、削除できる完全な管理インターフェイスが用意されています。
3. カスタムキーストアを使用する準備ができたら、 AWS CloudHSM 関連するクラスターに接続します。 AWS KMS 接続をサポートするのに必要なネットワークインフラストラクチャを作成します。次に、専用 crypto user アカウント認証情報を使用してクラスターへのログインが行われ、クラスターでキーマテリアルを生成して管理できるようになります。
4. これで、カスタムキーストアで対称暗号化 KMS キーを作成できるようになりました。カスタムキーストアを指定するだけで、KMS キーを作成できます。

どの時点で問題が発生しても、「カスタムキーストアのトラブルシューティング」トピックでヘルプを見つけることができます。回答が見つからない場合は、このガイドの各ページの下部にあるフィードバックリンクを使用するか、または AWS Key Management Service ディスカッションフォーラムに質問を投稿してください。

クォータ

AWS KMS 接続状態に関係なく、AWS アカウントAWS CloudHSM キーストアと外部キーストアの両方を含め、各リージョンとリージョンに最大 10 個のカスタムキーストアを許可します。さらに、キーストアでの KMS AWS KMS キーの使用にはリクエストクォータがあります。 AWS CloudHSM

料金表

AWS KMS カスタムキーストアとカスタムキーストア内のカスタマー管理キーのコストについては、料金をご覧ください。AWS Key Management Service AWS CloudHSM クラスターと HSM のコストについては、「AWS CloudHSM 料金表」を参照してください。

リージョン

AWS KMS アジア太平洋 (メルボルン)、中国 (北京)、中国 (寧夏)、ヨーロッパ (スペイン) を除き、AWS CloudHSM AWS リージョン サポートされているすべての地域のキーストアをサポートします。 AWS KMS

サポートされていない機能

AWS KMS カスタムキーストアでは以下の機能はサポートされていません。

- 非対称 KMS キー
- 非対称データキーペア
- HMAC KMS キー
- インポートされたキーマテリアルを持つ KMS キー
- 自動キーローテーション
- マルチリージョンキー

トピック

- AWS CloudHSM キーストアの概念
- AWS CloudHSM キーストアへのアクセスの制御
- CloudHSM カスタムキーストアの管理
- CloudHSM キーストアでの KMS キーの管理
- カスタムキーストアのトラブルシューティング

AWS CloudHSM キーストアの概念

このトピックでは、AWS CloudHSM カスタムキーストアで使用されるいくつかの概念について説明します。

AWS CloudHSM キーストア

AWS CloudHSM キーストアは、ユーザーが所有し管理する AWS CloudHSM クラスターに関連付けられたカスタムキーストアです。AWS CloudHSM クラスターは、FIPS 140-2 レベル 3 の認定を受けたハードウェアセキュリティモジュールによりバックアップされています。

AWS CloudHSM キーストアで KMS キーを作成すると、AWS KMS が、256 ビットの永続的でエクスポート不可の Advanced Encryption Standard (AES) 対称キーを、関連する AWS CloudHSM クラスターに作成します。このキーマテリアルは、HSM を非暗号化のままにしません。AWS CloudHSM カスタムキーストアで KMS キーを使用するときは、暗号化オペレーションはクラスター内の HSM で実行されます。

AWS CloudHSM キーストアは、AWS KMS の便利で包括的なキー管理インターフェイスと、AWS アカウント アカウントの AWS CloudHSM クラスターによって提供される追加のコントロールを組み合わせたものです。この統合された機能により、クラスター、HSM、バックアップの管理など、キーマテリアルを保存する HSM を完全に制御しながら、AWS KMS で KMS キーを作成、管理、使用することができます。AWS KMS コンソールと API を使用することで、AWS CloudHSM カスタムキーストアとその KMS キーを管理できます。関連するクラスターを管理するには、AWS CloudHSM コンソール、API、クライアントソフトウェア、および関連するソフトウェアライブラリを使用することもできます。

AWS CloudHSM キーストアの表示と管理、プロパティの編集、関連付けられた AWS CloudHSM クラスターの接続と切断が行えます。AWS CloudHSM キーストアを削除する必要がある場合は、削除をスケジュールし、猶予期間が終了するまで待機した後で、まず AWS CloudHSM キーストア内の KMS キーを削除します。AWS CloudHSM キーストアを削除すると AWS KMS からリソースが削除されますが、AWS CloudHSM クラスターには影響しません。

AWS CloudHSM クラスター

すべての AWS CloudHSM キーストアは、1 つの AWS CloudHSM クラスターに関連付けられています。AWS CloudHSM キーストアに AWS KMS key を作成すると、AWS KMS が、関連付けられたクラスターでキーマテリアルを作成します。AWS CloudHSM キーストアで KMS キーを使用すると、関連付けられたクラスターで、暗号化オペレーションが実行されます。

各 AWS CloudHSM クラスターは、1 つの AWS CloudHSM キーストアにのみ関連付けることができます。選択したクラスターを別の AWS CloudHSM キーストアに関連付けたり、別の AWS CloudHSM キーストアに関連付けられたクラスターとバックアップ履歴を共有したりすることはできません。クラスターは初期化され、アクティブで、AWS CloudHSM カスタムキーストアと同じ AWS アカウント およびリージョンに存在している必要があります。新しいクラスターを作成した

り、既存のクラスターを使用したりすることができます。AWS KMS はクラスターの排他的使用を必要としません。AWS CloudHSM キーストアに KMS キーを作成するときは、関連付けられたクラスターにアクティブな HSM が 2 つ以上含まれている必要があります。他のすべてのオペレーションでは、1 つの HSM しか必要ありません。

AWS CloudHSM キーストアを作成するときに AWS CloudHSM クラスターを指定しますが、これは変更できません。ただし、バックアップ履歴を共有するクラスターは、元のクラスターに置き換えることができます。これにより、必要に応じてクラスターを削除し、バックアップの 1 つから作成したクラスターに置き換えることができます。関連する AWS CloudHSM クラスターを完全に制御することにより、ユーザーとキーを管理し、HSM を作成および削除して、バックアップを使用および管理できます。

AWS CloudHSM キーストアを使用する準備ができたら、関連付けられた AWS CloudHSM クラスターにそれを接続します。いつでもカスタムキーストアを接続および切断できます。カスタムキーストアが接続されている場合は、その KMS キーを作成して使用できます。接続が切断されると、AWS CloudHSM キーストアとその KMS キーを表示および管理できます。ただし、暗号化オペレーションの AWS CloudHSM キーストアで、新しい KMS キーを作成、使用することはできません。

`kmsuser` Crypto User

関連する AWS CloudHSM クラスターのキーマテリアルを作成および管理するために、AWS KMS は専用の AWS CloudHSM crypto user (CU) を`kmsuser` という名前のクラスターに追加します。`kmsuser` CU は、クラスター内のすべての HSM に自動的に同期された標準の CU アカウントで、クラスターバッグに保存されます。

AWS CloudHSM キーストアを作成する前に、cloudhsm_mgmt_util で createUser コマンドを使用して、AWS CloudHSM クラスターの `kmsuser` CU アカウントを作成します。次に、AWS CloudHSM キーストアを作成するときに、`kmsuser` アカウントのパスワードを AWS KMS に入力します。カスタムキーストアを接続すると、AWS KMS が `kmsuser` CU としてクラスターにログインし、パスワードをローテーションします。AWS KMS は `kmsuser` パスワードを暗号化して安全に保存します。パスワードがローテーションされる際、新しいパスワードは同じ方法で暗号化されて保存されます。

AWS CloudHSM キーストアが接続されている限り、AWS KMS は `kmsuser` としてログインし続けます。この CU アカウントは他の目的では使用しないでください。ただし、`kmsuser` CU アカウントの最終的な制御は保持されます。`kmsuser` が所有するキーのキーハンドルは、いつでも検索することができます。必要に応じて、カスタムキーストアの切断、`kmsuser` パスワードの変更、`kmsuser` でクラスターへログイン、および `kmsuser` が所有するキーの表示と管理が行えます。

kmsuser CU アカウントの作成手順については、「kmsuser Crypto User を作成する」を参照してください。

AWS CloudHSM キーストアの KMS キー

AWS KMS または AWS KMS API を使用すると、AWS CloudHSM キーストアに AWS KMS keys を作成できます。KMS キーで使用するのと同じ方法を使用します。唯一の違いは、AWS CloudHSM キーストアを識別し、キーマテリアルのオリジンが AWS CloudHSM クラスターであることを指定する必要があることです。

AWS CloudHSM キーストアに KMS キーを作成する場合、AWS KMS が、AWS KMS に KMS キーを作成し、関連付けられたクラスターに、256 ビットの永続的でエクスポート不能な Advanced Encryption Standard (AES) 対称キーマテリアルを生成します。暗号化オペレーションで AWS KMS キーを使用すると、オペレーションは、クラスターベースの AES キーを使用して、AWS CloudHSM クラスターで実行されます。AWS CloudHSM は異なるタイプの対称キーと非対称キーをサポートしていますが、AWS CloudHSM キーストアは AES 対称暗号化キーのみをサポートします。

AWS KMS コンソールの AWS CloudHSM キーストアで KMS キーを表示し、コンソールオプションを使用して、カスタムキーストア ID を表示できます。DescribeKey オペレーションを使用して、AWS CloudHSMキーストア ID とAWS CloudHSMクラスター ID を検索することもできます。

AWS CloudHSM キーストアの KMS キーは、AWS KMS の KMS キーと同じように動作します。認可されたユーザーは、KMS キーの使用と管理のために同じアクセス許可が必要です。同じコンソールの手順と API オペレーションを使用して、AWS CloudHSM キーストアで KMS キーを表示および管理します。これには、KMS キーの有効化と無効化、タグとエイリアスの作成と使用、IAM ポリシーとキーポリシーの設定と変更が含まれます。AWS CloudHSM キーストアの KMS キーは暗号化オペレーションに使用でき、それを、カスタマーマネージドキーの使用をサポートする統合された AWS サービスに使用することができます。ただし、自動キーローテーションを有効化したり、AWS CloudHSM キーストアの KMS キーにキーマテリアルをインポートしたりすることはできません。

AWS CloudHSM キーストア内の KMS キーのスケジュール削除にも同じプロセスを使用します。待機期間が終了すると、AWS KMS は KMS から KMS キーを削除します。次に、関連付けられた AWS CloudHSM クラスターから、KMS キーのキーマテリアルを可能な限り削除します。ただし、クラスターとそのバックアップから、手動で孤立したキーマテリアルを削除する必要があります。

AWS CloudHSM キーストアへのアクセスの制御

AWS CloudHSM カスタムキーストアと AWS CloudHSM クラスターへのアクセスを制御するときは、IAM ポリシーを使用します。キーポリシー、IAM ポリシー、グラントを使用すれば、AWS CloudHSM キーストアの AWS KMS keys へのアクセスを制御できます。ユーザー、グループ、およ

びロールには、実行する可能性が高いタスクに必要なアクセス許可のみを与えることをお勧めします。

トピック

- AWS CloudHSM キーストアのマネージャーとユーザーの承認
- AWS CloudHSM および Amazon EC2 リソースの管理を AWS KMS に認可する

AWS CloudHSM キーストアのマネージャーとユーザーの承認

AWS CloudHSM キーストアを設計するときは、そのキーストアを使用および管理するプリンシパルに必要なアクセス許可のみが付与されていることを確認してください。次のリストは、AWS CloudHSM キーストアの管理者とユーザーに必要な最小限のアクセス許可を示しています。

- AWS CloudHSM キーストアを作成および管理するプリンシパルは、AWS CloudHSM キーストア API オペレーションを使用するために次のアクセス許可を必要とします。
  - `cloudhsm:DescribeClusters`
  - `kms:CreateCustomKeyStore`
  - `kms:ConnectCustomKeyStore`
  - `kms:DeleteCustomKeyStore`
  - `kms:DescribeCustomKeyStores`
  - `kms:DisconnectCustomKeyStore`
  - `kms:UpdateCustomKeyStore`
  - `iam:CreateServiceLinkedRole`
- AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスターを作成し管理するプリンシパルには、AWS CloudHSM クラスターを作成し初期化するアクセス許可が必要です。これには、仮想プライベートクラウド (VPC) の作成または使用、サブネットの作成、Amazon EC2 インスタンスの作成権限が含まれます。また、HSM の作成と削除、およびバックアップの管理が必要な場合もあります。必要なアクセス権限のリストについては、「AWS CloudHSM ユーザーガイド」の「AWS CloudHSM のアイデンティティとアクセス管理」を参照してください。
- AWS CloudHSM キーストアで AWS KMS keys を作成し管理するプリンシパルには、AWS KMS で KMS キーを作成し管理するプリンシパルと同じアクセス許可が必要です。AWS CloudHSM キーストアの KMS キーのデフォルトキーポリシーは、AWS KMS の KMS キーのデフォルトキーポリシーと同じです。タグとエイリアスを使用して KMS キーへのアクセス許可を管理する属性ベースのアクセスコントロール (ABAC) は、AWS CloudHSM キーストアの KMS キーでも有効です。

- AWS CloudHSM キーストアで暗号化オペレーションに KMS キーを使用するプリンシパルには、KMS キーで暗号化オペレーション (kms:Decrypt など) を実行するためのアクセス許可が必要です。　これらのアクセス許可は、キーポリシー、IAM ポリシーで付与できます。ただし、AWS CloudHSM キーストアで KMS キーを使用するための追加のアクセス許可は必要ありません。

AWS CloudHSM および Amazon EC2 リソースの管理を AWS KMS に認可する

AWS CloudHSM キーストアをサポートするため、AWS KMS には AWS CloudHSM クラスターに関する情報を取得するためのアクセス許可が必要になります。また、AWS CloudHSM キーストアを AWS CloudHSM クラスターに接続するネットワークインフラストラクチャを作成するための、アクセス許可も必要です。これらのアクセス許可を取得するために、 はサービスAWSServiceRoleForKeyManagementServiceCustomKeyStoresにリンクされたロールを にAWS KMS作成しますAWS アカウント。AWS CloudHSM キーストアを作成するユーザーには、サービスにリンクされたロールの作成を許可する `iam:CreateServiceLinkedRole` アクセス許可が必要です。

トピック

- AWS KMS サービスにリンクされたロールについて
- サービスにリンクされたロールの作成
- サービスにリンクされたロールの説明を編集する
- サービスにリンクされたロールを削除する

AWS KMS サービスにリンクされたロールについて

サービスリンクロールは、ユーザーの代わりに他の AWS サービスを呼び出す 1 つの AWS サービスアクセス許可を付与する IAM ロールです。これは、複数の統合された AWS サービスの機能を、複雑な IAM ポリシーを作成したり維持したりせずに簡単に使用できるように設計されました。詳細については、「AWS KMS のサービスにリンクされたロールの使用」を参照してください。

AWS CloudHSM キーストアの場合、 はAWSKeyManagementServiceCustomKeyStoresServiceRolePolicyポリシーを使用してAWSServiceRoleForKeyManagementServiceCustomKeyStoresサービスにリンクされたロールAWS KMSを作成します。このポリシーはロールに以下のアクセス許可を与えます。

- cloudhsm:Describe* – カスタムキーストアにアタッチされている AWS CloudHSM クラスター内の変更を検出します。

- ec2:CreateSecurityGroup — AWS CloudHSMキーストアを接続して、 AWS KMS とAWS CloudHSMクラスター間のネットワークトラフィックフローを有効にするセキュリティグループを作成するときに使用されます。
- ec2:AuthorizeSecurityGroupIngress — AWS CloudHSMキーストアを接続して、 からAWS CloudHSMクラスターを含む VPC AWS KMS へのネットワークアクセスを許可するときに使用されます。
- ec2:CreateNetworkInterface — AWS CloudHSMキーストアを接続して、 AWS KMSとAWS CloudHSMクラスター間の通信に使用されるネットワークインターフェイスを作成するときに使用されます。
- ec2:RevokeSecurityGroupEgress — AWS CloudHSMキーストアを接続して、 がAWS KMS作成したセキュリティグループからすべてのアウトバウンドルールを削除するときに使用されます。
- ec2:DeleteSecurityGroup — AWS CloudHSMキーストアを切断して、AWS CloudHSMキーストアの接続時に作成されたセキュリティグループを削除するときに使用されます。
- ec2:DescribeSecurityGroups — AWS CloudHSMクラスターを含む VPC でAWS KMS作成されたセキュリティグループの変更をモニタリングし、障害発生時に が明確なエラーメッセージを表示AWS KMSできるようにするために使用されます。
- ec2:DescribeVpcs — AWS CloudHSMクラスターを含む VPC の変更をモニタリングし、障害発生時に が明確なエラーメッセージを表示AWS KMSできるようにするために使用されます。
- ec2:DescribeNetworkAcls — AWS CloudHSMクラスターを含む VPC ACLs の変更をモニタリングし、障害発生時に が明確なエラーメッセージを表示AWS KMSできるようにするために使用されます。
- ec2:DescribeNetworkInterfaces — AWS CloudHSMクラスターを含む VPC でAWS KMS作成されたネットワークインターフェイスの変更をモニタリングし、障害発生時に が明確なエラーメッセージを表示AWS KMSできるようにするために使用されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudhsm:Describe*",
        "ec2:CreateNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
```

```
        "ec2:RevokeSecurityGroupEgress",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

AWSServiceRoleForKeyManagementServiceCustomKeyStores サービスにリンクされたロールは のみを信頼するためcks.kms.amazonaws.com、 のみがこのサービスにリンクされたロールを引き受けAWS KMSることができます。このロールは、AWS KMS が AWS CloudHSM クラスターを表示し、AWS CloudHSM キーストアをそれに関連付けられた AWS CloudHSM クラスターに接続するために必要なオペレーションに限定されています。AWS KMS に対して追加のアクセス許可は付与されません。たとえば、AWS KMS に、AWS CloudHSM クラスター、HSM、またはバックアップを作成、管理、または削除するためのアクセス許可はありません。

リージョン

AWS CloudHSM キーストア機能と同様に、AWSServiceRoleForKeyManagementServiceCustomKeyStoresロールは AWS KMSと AWS リージョンが利用可能なすべての でサポートAWS CloudHSMされています。各サービスがサポートしている AWS リージョン のリストについては、「Amazon Web Services 全般のリファレンス」の「AWS Key Management Service エンドポイントとクォータ」および「AWS CloudHSM エンドポイントとクォータ」を参照してください。

AWS のサービスでサービスリンクロールを使用する方法の詳細については、IAM ユーザーガイドの「サービスリンクロールの使用」を参照してください。

サービスにリンクされたロールの作成

AWS KMS は、 AWS CloudHSMキーストアを作成するAWS アカウントときに、ロールがまだ存在しない場合、AWSServiceRoleForKeyManagementServiceCustomKeyStoresサービスにリンクされたロールを に自動的に作成します。このサービスにリンクされたロールを直接作成または再作成することはできません。

サービスにリンクされたロールの説明を編集する

ロール名または AWSServiceRoleForKeyManagementServiceCustomKeyStores サービスにリンクされたロールのポリシーステートメントを編集することはできませんが、ロールの説明を編集できます。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編集」を参照してください。

サービスにリンクされたロールを削除する

AWS KMS すべてのAWS CloudHSMキーストアを削除AWS アカウントしても、 は からAWSServiceRoleForKeyManagementServiceCustomKeyStoresサービスにリンクされたロールを削除しません。現在、AWSServiceRoleForKeyManagementServiceCustomKeyStoresサービスにリンクされたロールを削除する手順はありませんが、 AWS KMSは、アクティブなAWS CloudHSM キーストアがない限り、このロールを引き受けたり、アクセス許可を使用したりしません。

CloudHSM カスタムキーストアの管理

AWS Management Console と AWS KMS API を使用して、カスタムキーストアを管理できます。たとえば、カスタムキーストアを表示、そのプロパティを編集、関連付けられた AWS CloudHSM クラスターからカスタムキーストアに接続および切断、およびカスタムキーストアを削除できます。

トピック

- AWS CloudHSM キーストアの作成
- AWS CloudHSM キーストアの表示
- AWS CloudHSM キーストア設定の編集
- AWS CloudHSM キーストアの接続と切断
- AWS CloudHSM キーストアの削除

AWS CloudHSM キーストアの作成

自分のアカウントには 1 つまたは複数の AWS CloudHSM キーストアを作成できます。各 AWS CloudHSM キーストアは、同じ AWS アカウント とリージョンにある 1 つの AWS CloudHSM クラスターに関連付けられます。AWS CloudHSM キーストアを作成するときは、事前に前提条件を構成する必要があります。次に、AWS CloudHSM キーストアを使用する前にそのキーストアを AWS CloudHSM クラスターに接続します。

> **Note**
>
> 接続が解除された既存の AWS CloudHSM キーストアと同じプロパティ値をすべて使って AWS CloudHSM キーストアを作成しようとすると、AWS KMS は、新しい AWS CloudHSM キーストアを作成せず、例外のスローやエラーの表示を行いません。代わりに AWS KMS は、この重複は再試行の結果である可能性が高いと認識し、既存の AWS CloudHSM キーストアの ID を返します。

> **Tip**
>
> AWS CloudHSM キーストアをただちに接続する必要はありません。使用する準備ができるまで切断された状態にしておくことができます。ただし、正しく設定されていることを確認するために、接続して、接続状態を表示してから、切断するとよいかも知れません。

トピック

前提条件を構成する

各 AWS CloudHSM キーストアは AWS CloudHSM クラスターにサポートされています。AWS CloudHSM キーストアを作成するには、別のキーストアに関連付けられていない、アクティブな AWS CloudHSM クラスターを指定する必要があります。AWS KMS がユーザーの代わりにキーを作成して管理できる、クラスターの HSM に専用 Crypto User (CU) を作成する必要もあります。

AWS CloudHSM キーストアを作成する前に、次の手順を実行します。

AWS CloudHSM クラスターを選択する

各 AWS CloudHSM キーストアは、AWS CloudHSM クラスターに 1 つずつ関連付けられます。AWS CloudHSM キーストアで AWS KMS keys を作成すると、AWS KMS は AWS KMS で ID や Amazon リソースネーム (ARN) などの KMS キーメタデータを作成します。その後で、関連付けられたクラスターの HSM でキーマテリアルを作成します。新しい AWS CloudHSM クラスターを作成するか、既存のものを使用できます。AWS KMS ではクラスターへの排他的アクセスが要求されません。

選択した AWS CloudHSM クラスターは、AWS CloudHSM キーストアに完全に関連付けられます。AWS CloudHSM キーストアを作成した後は、関連付けられたクラスターのクラスター ID を変更できますが、指定したクラスターは元のクラスターとバックアップ履歴を共有する必要があります。無関係のクラスターを使用するには、新しい AWS CloudHSM キーストアを作成する必要があります。

選択した AWS CloudHSM クラスターには次の特性が必要です。

- クラスターがアクティブである必要があります。

  クラスターを作成して初期化し、プラットフォームに AWS CloudHSM クライアントソフトウェアをインストールして、クラスターをアクティブ化する必要もあります。手順については、「AWS CloudHSM ユーザーガイド」の「AWS CloudHSM の開始方法」を参照してください。

- クラスターは、AWS CloudHSM キーストアと同じ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。あるリージョンの AWS CloudHSM キーストアを別のリージョンのクラスターに関連付けることはできません。マルチリージョンの主要なインフラストラクチャを作成するには、各リージョンに AWS CloudHSM キーストアとクラスターを作成する必要があります。

- クラスターを同じアカウントおよびリージョン内の別のカスタムキーストアに関連付けることはできません。アカウントおよびリージョン内のそれぞれの AWS CloudHSM キーストアは、異なる AWS CloudHSM クラスターに関連付けられている必要があります。カスタムキーストアに関連付け済みのクラスターまたは関連付け済みのクラスターとバックアップ履歴を共有するクラスターを指定することはできません。バックアップ履歴を共有するクラスターには同じクラスター証明書があります。クラスターのクラスター証明書を表示するには、 AWS CloudHSMコンソールまたは DescribeClustersオペレーションを使用します。

  AWS CloudHSM クラスターを別のリージョンにバックアップする場合、そのクラスターは別のクラスターと見なされ、そのリージョン内のカスタムキーストアにバックアップを関連付けることができます。ただし、2 つのカスタムキーストアの KMS キーは、同じバッキングキーを持っていても、相互運用できません。AWS KMS は、メタデータを暗号文にバインドし、暗号化した KMS キーによってのみ復号できるようにします。

- クラスターは、リージョンの 2 つ以上のアベイラビリティーゾーンでプライベートサブネットを使用して設定する必要があります。AWS CloudHSM はすべてのアベイラビリティーゾーンでサポートされていないため、リージョン内のすべてのアベイラビリティーゾーンでプライベートサブネットを作成することをお勧めします。既存のクラスターのサブネットを再構成することはできませんが、クラスター構成の異なるサブネットを持つバックアップからクラスターを作成することはできます。

> **Important**
>
> AWS CloudHSM キーストアを作成した後は、AWS CloudHSM クラスター用に設定されたプライベートサブネットを削除しないでください。AWS KMS がクラスター構成内ですべてのサブネットを見つけることができない場合、カスタムキーストアへの接続を試みると、SUBNET_NOT_FOUND 接続エラー状態で失敗します。詳細については、「接続障害の修復方法」を参照してください。

- クラスターのセキュリティグループ (cloudhsm-cluster-*<cluster-id>*-sg) には、ポート 2223 ~ 2225 で TCP トラフィックを許可するインバウンドルールとアウトバウンドルールを含める必要があります。インバウンドルールの Source とアウトバウンドルールの Destination は、セキュリティグループ ID と一致している必要があります。これらのルールは、クラスターの作成時にデフォルトで設定されます。変更または削除しないでください。
- クラスターは、異なるアベイラビリティーゾーンの少なくとも 2 つのアクティブな HSM を含む必要があります。HSMs、 AWS CloudHSMコンソールまたは DescribeClustersオペレーションを使用します。必要に応じて、HSM を追加できます。

信頼アンカー証明書を見つける

カスタムキーストアを作成する場合、AWS CloudHSM クラスターのトラストアンカー証明書を AWS KMS にアップロードする必要があります。AWS KMS には、AWS CloudHSM キーストアを関連付けられた AWS CloudHSM クラスターに接続するために、トラストアンカー証明書が必要です。

すべてのアクティブな AWS CloudHSM クラスターには信頼アンカー証明書があります。クラスターを初期化する際、この証明書を生成し、customerCA.crt ファイルに保存して、クラスターに接続するホストにコピーしてください。

AWS KMS 用の **kmsuser** Crypto User を作成する

AWS CloudHSM キーストアを管理するために、AWS KMS は、選択したクラスターで kmsuser Crypto User (CU) アカウントにログインします。AWS CloudHSM キーストアを作成する前に kmsuser CU を作成する必要があります。次に、AWS CloudHSM キーストアを作成するときに、kmsuser のパスワードを AWS KMS に指定します。AWS CloudHSM　カスタムキーストアを関連付けられた AWS CloudHSM クラスターに接続すると、AWS KMS は kmsuser として ログインし、kmsuser パスワードをローテーションします。

> **Important**
>
> kmsuser CU を作成するとき、2FA オプションを指定しないでください。指定すると、AWS KMS はログインできず、AWS CloudHSM キーストアはこの AWS CloudHSM クラスターに接続できません。2FA を指定すると、元に戻すことはできません。代わりに、CU を削除して再作成する必要があります。

kmsuser CU を作成するには、次の手順に従います。

1. cloudhsm_mgmt_util を、「AWS CloudHSMユーザーガイド」の「Getting started with CloudHSM Management Utility (CMU)」(CloudHSM 管理ユーティリティ (CMU) の開始方法) のトピックにある説明に従って起動します。
2. cloudhsm_mgmt_util で createUser コマンドを使用して、kmsuser という名前の CU を作成します。パスワードは 7〜32 の英数字で構成する必要があります。大文字と小文字が区別され、特殊文字を含めることはできません。

   たとえば、次のコマンド例では、パスワードが kmsPswd の kmsuser CU を作成します。

   ```
   aws-cloudhsm> createUser CU kmsuser kmsPswd
   ```

AWS CloudHSM キーストアを作成する (コンソール)

AWS Management Console で AWS CloudHSM キーストアを作成するときに、ワークフローの一部として前提条件を追加および作成することができます。ただし、プロセスは事前に構成しておくとより迅速になります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。
4. [キーストアの作成] を選択します。
5. カスタムキーストアのわかりやすい名前を入力します。名前は、アカウント内のすべてのカスタムキーストアの間で、一意でなければなりません。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

6. AWS CloudHSM キーストアの AWS CloudHSM クラスターを選択します。または、新しい AWS CloudHSM クラスターを作成するには、AWS CloudHSM クラスターの作成リンクを選択します。

   メニューに、AWS CloudHSM キーストアにまだ関連付けられていないアカウントおよびリージョンの AWS CloudHSM キーストアが表示されます。クラスターは、カスタムキーストアとの関連付けの要件を満たす必要があります。

7. [Choose file] (ファイルを選択) を選択し、選択した AWS CloudHSM クラスターのトラストアンカー証明書をアップロードします。これは、クラスターを初期化する際に作成した `customerCA.crt` ファイルです。
8. 選択したクラスターで作成した `kmsuser` Crypto User (CU) のパスワードを入力します。
9. [Create] (作成) を選択します。

手順が完了すると、アカウントとリージョンの AWS CloudHSM キーストアのリストに新しい AWS CloudHSM キーストアが表示されます。正常に完了しなかった場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「カスタムキーストアのトラブルシューティング」を参照してください。

接続が解除された既存の AWS CloudHSM キーストアと同じプロパティ値をすべて使って AWS CloudHSM キーストアを作成しようとすると、AWS KMS は、新しい AWS CloudHSM キーストアを作成せず、例外のスローやエラーの表示を行いません。代わりに AWS KMS は、この重複は再試行の結果である可能性が高いと認識し、既存の AWS CloudHSM キーストアの ID を返します。

次の手順: 新しい AWS CloudHSM キーストアが自動的に接続されない。AWS CloudHSM カスタムキーストアで AWS KMS keys を作成する前に、関連付けられた AWS CloudHSM クラスターにカスタムキーストアを接続する必要があります。

AWS CloudHSM キーストアを作成する (API)

CreateCustomKeyStore オペレーションを使用して、アカウントとリージョンの AWS CloudHSMクラスターに関連付けられた新しいAWS CloudHSMキーストアを作成できます。これらの例では AWS

Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用できます。

`CreateCustomKeyStore` オペレーションでは、次のパラメータ値が必要です。

- CustomKeyStoreName – アカウント内で一意のカスタムキーストアのわかりやすい名前。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

- CloudHsmClusterId – AWS CloudHSMキーストアの要件を満たす AWS CloudHSMクラスターのクラスター ID。
- KeyStorePassword – 指定されたクラスター内の `kmsuser` CU アカウントのパスワード。
- TrustAnchorCertificate – クラスターを初期化したときに作成した `customerCA.crt` ファイルの内容。

次の例では、架空のクラスター ID を使用します。コマンドを実行する前に、有効なクラスター ID と置き換えます。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate <certificate-goes-here>
```

AWS CLI を使用している場合、その内容ではなく、信頼アンカー証明書ファイルを指定できます。次の例では、`customerCA.crt` ファイルはルートディレクトリにあります。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate file://customerCA.crt
```

オペレーションが正常に終了したら、次のレスポンス例に示すように、`CreateCustomKeyStore` はカスタムキーストア ID を返します。

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

オペレーションが失敗した場合は、例外で示されているエラーを修正して、もう一度試してください。その他のヘルプについては、「カスタムキーストアのトラブルシューティング」を参照してください。

接続が解除された既存の AWS CloudHSM キーストアと同じプロパティ値をすべて使って AWS CloudHSM キーストアを作成しようとすると、AWS KMS は、新しい AWS CloudHSM キーストアを作成せず、例外のスローやエラーの表示を行いません。代わりに AWS KMS は、この重複は再試行の結果である可能性が高いと認識し、既存の AWS CloudHSM キーストアの ID を返します。

次の手順: AWS CloudHSM キーストアを使用するには、これを AWS CloudHSM クラスターに接続します。

AWS CloudHSM キーストアの表示

AWS KMS コンソールまたは DescribeCustomKeyStoresオペレーションを使用して、各アカウントとリージョンのAWS CloudHSMキーストアを表示できます。

以下も参照してください。

- 外部キーストアを表示する
- AWS CloudHSM キーストアでの KMS キーの表示
- AWS KMS による AWS CloudTrail API コールのログ記録

トピック

- AWS CloudHSM キーストアを表示する (コンソール)
- AWS CloudHSM キーストアを表示する (API)

AWS CloudHSM キーストアを表示する (コンソール)

AWS Management Console で AWS CloudHSM キーストアを表示すると、以下を確認できます。

- カスタムキーストアの名前と ID
- 関連付け済み AWS CloudHSM クラスターの ID
- クラスター内の HSM の数

- 現在の接続ステータス

Disconnected の接続ステータス ([Status]) 値は、カスタムキーストアが新しく、まだ接続されたことがないこと、または AWS CloudHSM クラスターから意図的に切断されたことを示します。ただし、接続されているカスタムキーストアで KMS キーの使用を試みると失敗する場合は、カスタムキーストアまたはその AWS CloudHSM クラスターに問題がある可能性があります。ヘルプについては、「失敗した KMS キーを修正するには」を参照してください。

指定されたアカウントとリージョンで AWS CloudHSM キーストアを表示するには、以下の手順に従います。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。

表示をカスタマイズするには、[Create key store (キーストアを作成)] ボタンの下に表示される歯車アイコンをクリックします。

AWS CloudHSM キーストアを表示する (API)

AWS CloudHSM キーストアを表示するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、`CustomKeyStoreId` または `CustomKeyStoreName` パラメータのどちらかを使用して (両方は使用できません) 出力を特定のカスタムキーストアに制限できます。AWS CloudHSM キーストアでは、出力は、カスタムキーストアの ID と名前、カスタムキーストアのタイプ、関連付けられた AWS CloudHSM クラスターの ID、接続ステータスから構成されています。接続状態はエラーを示す場合、出力にエラーの理由を説明するエラーコードも含まれています。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

たとえば、次のコマンドは、アカウントとリージョンのすべてのカスタムキーストアを返します。`Limit` パラメータと `Marker` パラメータを使用して、出力のカスタムキーストアをページ分割できます。

```
$ aws kms describe-custom-key-stores
```

次のコマンド例では、CustomKeyStoreName パラメータを使用して ExampleCloudHSMKeyStore というフレンドリ名の唯一のカスタムキーストアを取得します。各コマンドで CustomKeyStoreName パラメータまたは CustomKeyStoreId パラメータのどちらかを使用できますが、両方を使用することはできません。

次の出力例は、AWS CloudHSM クラスターに接続されている AWS CloudHSM カスタムキーストアを表します。

> **Note**
>
> AWS CloudHSM キーストアと外部のキーストアを区別するために、DescribeCustomKeyStores レスポンスに CustomKeyStoreType フィールドが追加されました。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleCloudHSMKeyStore
{
   "CustomKeyStores": [
      {
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "ConnectionState": "CONNECTED",
         "CreationDate": "1.499288695918E9",
         "CustomKeyStoreId": "cks-1234567890abcdef0",
         "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
         "CustomKeyStoreType": "AWS_CLOUDHSM",
         "TrustAnchorCertificate": "<certificate appears here>"
      }
   ]
}
```

Disconnected の ConnectionState は、カスタムキーストアが接続されたことがないこと、または意図的に AWS CloudHSM クラスターから切断されたことを示します。ただし、KMS キーを、接続済みの AWS CloudHSM キーストアで使うことに失敗した場合は、AWS CloudHSM キーストアかその AWS CloudHSM クラスターに問題があることを示している可能性があります。ヘルプについては、「失敗した KMS キーを修正するには」を参照してください。

カスタムキーストアの ConnectionState が FAILED である場合、DescribeCustomKeyStores レスポンスには、エラーの理由を説明する ConnectionErrorCode 要素が含まれています。

たとえば、次の出力では、INVALID_CREDENTIALS 値は、kmsuser パスワードが無効であるために、カスタムキーストアの接続に失敗したことを示しています。このエラーやその他の接続エラーに関するヘルプについては、「」を参照してください カスタムキーストアのトラブルシューティング。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      {
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "ConnectionErrorCode": "INVALID_CREDENTIALS",
         "ConnectionState": "FAILED",
         "CustomKeyStoreId": "cks-1234567890abcdef0",
         "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
         "CustomKeyStoreType": "AWS_CLOUDHSM",
         "CreationDate": "1.499288695918E9",
         "TrustAnchorCertificate": "<certificate appears here>"
      }
   ]
}
```

AWS CloudHSM キーストア設定の編集

既存の AWS CloudHSM キーストアの設定は、変更できます。カスタムキーストアは、AWS CloudHSM クラスターから切断されている必要があります。

AWS CloudHSM キーストア設定を編集するには:

1. カスタムキーストアの AWS CloudHSM クラスターから、カスタムキーストアを切断します。カスタムキーストアが切断されている間は、カスタムキーストアで AWS KMS keys (KMS キー) を作成したり、暗号化オペレーション用に含められている KMS キーを使用したりすることはできません。
2. 1 つ以上の AWS CloudHSM キーストア設定を編集します。
3. カスタムキーストアを AWS CloudHSM クラスターに再接続します。

カスタムキーストアで次の設定を編集できます。

カスタムキーストアのわかりやすい名前。

新しい分かりやすい名前を入力します。新しい名前は、AWS アカウント 内のすべてのカスタムキーストアの間で一意でなければなりません。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

関連付け済みの AWS CloudHSM クラスターのクラスター ID。

この値を編集して、関連する AWS CloudHSM クラスターを元のクラスターと置き換えます。この機能を使用して、AWS CloudHSM クラスターが破損した場合、または削除された場合にカスタムキーストアを修復できます。

元のクラスターとバックアップ履歴を共有する AWS CloudHSM クラスターを指定して、異なるアベイラビリティーゾーンの 2 つのアクティブな HSM を含むカスタムキーストアとの関連付けの要件を満たします。バックアップ履歴を共有するクラスターには同じクラスター証明書があります。クラスターのクラスター証明書を表示するには、 DescribeClustersオペレーションを使用します。編集機能を使用してカスタムキーストアを無関係な AWS CloudHSM クラスターと関連付けることはできません。

`kmsuser` Crypto User (CU) の現在のパスワード。

AWS CloudHSM クラスター内の `kmsuser` CU の現在のパスワードを AWS KMS に伝えます。このアクションでは、AWS CloudHSM クラスター内の `kmsuser` CU のパスワードは変更されません。

AWS CloudHSM クラスター内の `kmsuser` CU のパスワードを変更する場合は、この機能を使用して、AWS KMS に 新しい `kmsuser` のパスワードを指定します。それ以外の場合、AWS KMS はクラスターにログインできず、カスタムキーストアをクラスターに接続しようとするすべての試行は失敗します。

トピック

- AWS CloudHSM キーストアを編集する (コンソール)
- AWS CloudHSM キーストアを編集する (API)

AWS CloudHSM キーストアを編集する (コンソール)

AWS CloudHSM キーストアを編集すると、設定可能な任意の値を変更できます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。
4. 編集する AWS CloudHSM キーストアの行を選択します。

   [接続の状態] 列の値が [切断済み] になっていない場合は、カスタムキーストアを切断してから編集する必要があります。([Key store actions] (キーストアアクション) メニューから [Disconnect] (切断) を選択します)。

   AWS CloudHSM キーストアが切断されている間は AWS CloudHSM キーストアとその KMS キーを管理できますが、AWS CloudHSM キーストアで KMS キーを作成または使用することはできません。
5. [Key store actions] (キーストアアクション) メニューから [Edit] (編集) を選択します。
6. 次のいずれかのアクションを実行します。

   - カスタムキーストアの新しいわかりやすい名前を入力します。
   - 関連する AWS CloudHSM クラスターのクラスター ID を入力します。
   - 関連付けられた AWS CloudHSM クラスターに `kmsuser` 暗号化ユーザーの現在のパスワードを入力します。
7. [保存] を選択します。

   プロシージャが正常に完了すると、編集した設定を説明するメッセージが表示されます。正常に行われなかった場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「カスタムキーストアのトラブルシューティング」を参照してください。
8. カスタムキーストアを再接続します。

   AWS CloudHSM キーストアを使用するときは、編集後にこれを再接続する必要があります。AWS CloudHSM キーストアは切断されたままにすることができます。ただし、切断されている間は、AWS CloudHSM キーストアで KMS キーを作成したり、暗号化オペレーションで AWS CloudHSM キーストア内の KMS キーを使用したりすることはできません。

AWS CloudHSM キーストアを編集する (API)

AWS CloudHSM キーストアのプロパティを変更するには、 UpdateCustomKeyStoreオペレーションを使用します。同じコマンドで、カスタムキーストアの複数のプロパティを変更できます。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。変更が有効であることを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

まずDisconnectCustomKeyStore、 を使用してカスタムキーストアをクラスターから切断します。AWS CloudHSM例のカスタムキーストア ID cks-1234567890abcdef0 を実際の ID に置き換えます。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

最初の例では、 を使用してAWS CloudHSMキーストアのフレンドリ名を UpdateCustomKeyStoreに変更しますDevelopmentKeys。このコマンドでは、CustomKeyStoreId パラメータを使用して AWS CloudHSM キーストアを識別し、CustomKeyStoreName でカスタムキーストアの新しい名前を指定します。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name DevelopmentKeys
```

次の例では、AWS CloudHSM に関連付けられたクラスターを、同じクラスターの別のバックアップに変更します。このコマンドでは、CustomKeyStoreId パラメータを使用して AWS CloudHSM キーストアを識別し、CloudHsmClusterId パラメータで新しいクラスター ID を指定します。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-
hsm-cluster-id cluster-1a23b4cdefg
```

次の例では、AWS KMS に現在の kmsuser のパスワードは ExamplePassword であると伝えます。このコマンドでは、CustomKeyStoreId パラメータを使用して AWS CloudHSM キーストアを識別し、KeyStorePassword パラメータで現在のパスワードを指定します。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-
store-password ExamplePassword
```

最後のコマンドは、AWS CloudHSM キーストアを AWS CloudHSM クラスターに再接続します。カスタムキーストアは切断された状態のままにできますが、新しい KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする前に接続する必要があります。カスタムキーストア ID 例を実際の ID と置き換えます。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

AWS CloudHSM キーストアの接続と切断

新しい AWS CloudHSM キーストアが接続されていません。AWS CloudHSM キーストアで AWS KMS keys を作成して使用するときは、事前に、関連付けられた AWS CloudHSM クラスターにこれを接続しておく必要があります。AWS CloudHSM キーストアはいつでも接続および切断することができ、その接続ステータスを表示できます。

AWS CloudHSM キーストアを自分で接続する必要はありません。AWS CloudHSM キーストアは、無期限に切断状態のままにし、使用の必要がある場合のみ接続することができます。ただし、定期的に接続をテストして、接続が正しく接続できることを確認するとよいでしょう。

> **Note**
>
> AWS CloudHSM キーストアは、ユーザーがキーストアを一度も接続したことがないか明示的に切断した場合のみ、接続ステータスが `DISCONNECTED` になります。AWS CloudHSM キーストアの接続ステータスが `CONNECTED` になっているのに使用できないときは、それに関連付けられた AWS CloudHSM クラスターがアクティブになっていること、および、アクティブな HSM が 1 つ以上含まれていることを確認します。接続障害については、the section called “カスタムキーストアのトラブルシューティング” を参照してください。

トピック

- AWS CloudHSM キーストアを接続する
- AWS CloudHSM キーストアを切断する
- AWS CloudHSM キーストアを接続する (コンソール)
- カスタムキーストアを接続する (API)
- AWS CloudHSM キーストアを切断する (コンソール)
- AWS CloudHSM キーストアを切断する (API)

AWS CloudHSM キーストアを接続する

AWS CloudHSM キーストアを接続すると、AWS KMS は関連付けられた AWS CloudHSM クラスターを検索し、そのクラスターに接続して、kmsuser Crypto User (CU) として AWS CloudHSM クライアントにログインし、その後 kmsuser パスワードをローテーションします。AWS CloudHSM キーストアが接続されている限り、AWS KMS は AWS CloudHSM クライアントにログインし続けます。

接続を確立するために、AWS KMS は kms-*<custom key store ID>* という名前のセキュリティグループを、クラスターの仮想プライベートクラウド (VPC) で作成します。セキュリティグループには、クラスターセキュリティグループからのインバウンドトラフィックを許可する単一のルールがあります。AWS KMS は、クラスターのプライベートサブネットの各アベイラビリティーゾーンに Elastic Network Interface (ENI) を作成します。AWS KMS は kms-*<cluster ID>* セキュリティグループとクラスターのセキュリティグループに ENI を追加します。各 ENI の説明は `KMS managed ENI for cluster` *<cluster-ID>* です。

接続プロセスは、完了するまでに長い時間 (最長 20 分) がかかる場合があります。

AWS CloudHSM キーストアを接続する前に、要件を満たしているかどうかを確認します。

- 関連付け済みの AWS CloudHSM クラスターに少なくとも 1 つのアクティブな HSM が含まれている必要があります。クラスター内の HSMs の数を確認するには、AWS CloudHSMコンソールでクラスターを表示するか、 DescribeClustersオペレーションを使用します。必要に応じて、HSM を追加できます。
- クラスターには kmsuser Crypto User (CU) アカウントが必要ですが、AWS CloudHSM キーストアを接続しているときは、この CU はクラスターにログインできません。ログアウトのヘルプについては、「ログアウトして再接続する方法」を参照してください。
- AWS CloudHSM キーストアの接続ステータスは、`DISCONNECTING` または `FAILED` にすることはできません。接続状態を表示するには、 AWS KMS コンソールまたは DescribeCustomKeyStoresレスポンスを使用します。接続ステータスが `FAILED` の場合、カスタムキーストアを切断し、問題を解決してから接続します。

接続障害については、接続障害の修復方法 を参照してください。

AWS CloudHSM キーストアが接続されているときは、そこで KMS キーを作成し、暗号化オペレーションで既存の KMS キーを使用することができます。

AWS CloudHSM キーストアを切断する

AWS CloudHSM キーストアを切断すると、AWS KMS は、AWS CloudHSM クライアントからログアウトし、関連付けられた AWS CloudHSM クラスターとの接続を切り、接続をサポートするために作成されたネットワークインフラストラクチャを削除します。

AWS CloudHSM キーストアが切断されている間は AWS CloudHSM キーストアとその KMS キーを管理できますが、AWS CloudHSM キーストアで KMS キーを作成または使用することはできません。`PendingDeletion` でない限り、キーストアの接続ステータスは `DISCONNECTED` で、カスタムキーストアの KMS キーのキーステータスは `Unavailable` です。AWS CloudHSM キーストアはいつでも再接続できます。

カスタムキーストアを切断すると、そのキーストアの KMS キーはただちに使用できなくなります (結果整合性の影響を受ける)。ただし、KMS キーで保護されたデータキーにより暗号化されたリソースは、KMS キーがデータキーの復号化などで再び使用されるまでは影響を受けません。この問題は AWS のサービス に影響します。その多くが、リソースを保護するためにデータキーを使用しています。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

> **Note**
>
> カスタムキーストアが切断されている間は、カスタムキーストアで KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする試みはすべて失敗します。このオペレーションにより、ユーザーが機密データを保存したりアクセスしたりすることを防ぐことができます。

カスタムキーストアの切断の影響をより正確に推定するには、カスタムキーストアで KMS キーを識別し、その過去の使用状況を特定します。

AWS CloudHSM キーストアを切断する理由としては、次のようなものが挙げられます。

- パスワードをローテーション**`kmsuser`**する。AWS KMS は AWS CloudHSM クラスターに接続するたびに `kmsuser` パスワードを変更します。パスワードローテーションを強制的に実行するには、切断して再接続します。
- AWS CloudHSM クラスターで KMS キーのキーマテリアルを監査するには。カスタムキーストアを切断すると、AWS KMS は AWS CloudHSM クライアントの `kmsuser` 暗号化ユーザーアカウントから口グアウトします。これにより、クラスターに `kmsuser` CU として口グインし、KMS キーのキーマテリアルを監査および管理することができます。

- AWS CloudHSM キーストアですべての KMS キーをただちに無効にするには キーAWS CloudHSMストアで KMS キーを無効または再度有効にするには、 AWS Management Console または DisableKeyオペレーションを使用します。これらのオペレーションは迅速に完了しますが、一度に処理される KMS キーは 1 つです。AWS CloudHSM キーストアを切断すると、AWS CloudHSM キーストアのすべての KMS キーのキーステータスがただちに `Unavailable` に変更され、暗号化オペレーションで使用できなくなります。
- 失敗した接続試行を修復するには。AWS CloudHSM キーストアを接続する試みに失敗した場合 (カスタムキーストアの接続ステータスが `FAILED` になる) は、再度接続を試みる前に AWS CloudHSM キーストアを切断する必要があります。

AWS CloudHSM キーストアを接続する (コンソール)

AWS Management Console で AWS CloudHSM キーストアを接続するには、まず、[Custom key stores] (カスタムキーストア) ページで AWS CloudHSM キーストアを選択します。この接続処理には、完了までに最大で 20 分かかります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。
4. 接続する AWS CloudHSM キーストアの行を選択します。

   AWS CloudHSM キーストアの接続状態が [失敗] になっている場合は、カスタムキーストアを切断してからキーストアを接続する必要があります。
5. [Key store actions] (キーストアアクション) メニューから [Connect] (接続) を選択します。

AWS KMS がカスタムキーストアを結合するプロセスを開始します。関連付け済みの AWS CloudHSM クラスターを見つけ、必要なネットワークインフラストラクチャを構築し、そのインフラストラクチャに接続し、AWS CloudHSM クラスターに `kmsuser` CU としてログインして、`kmsuser` パスワードをローテーションします。操作が完了すると、接続状態が [接続済み] に変わります。

オペレーションが失敗すると、失敗の理由を説明するエラーメッセージが表示されます。再度接続を試みる前に、AWS CloudHSM キーストアの接続ステータスを表示します。状態が [失敗] になってい

る場合は、カスタムキーストアを切断してからキーストアをもう一度接続する必要があります。ヘルプが必要な場合は、「カスタムキーストアのトラブルシューティング」を参照してください。

次の手順: the section called “AWS CloudHSM キーストアでの KMS キーの作成”

カスタムキーストアを接続する (API)

切断されたAWS CloudHSMキーストアを接続するには、 ConnectCustomKeyStoreオペレーションを使用します。関連付けられた AWS CloudHSM クラスターにはアクティブな HSM が 1 つ以上含まれている必要があります。また、接続ステータスを FAILED にすることはできません。

接続プロセスは、完了するまでに長い時間 (最長 20 分) かかります。すぐに失敗しない限り、オペレーションは HTTP 200 レスポンスとプロパティを含まない JSON オブジェクトを返します。ただし、この初期レスポンスは接続に成功したことを示していません。カスタムキーストアの接続状態を確認するには、 DescribeCustomKeyStoresレスポンスを参照してください。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

AWS CloudHSM キーストアを識別するには、カスタムキーストア ID を使用します。ID は、 コンソールのカスタムキーストアページで、またはパラメータなしで DescribeCustomKeyStoresオペレーションを使用して確認できます。この例を実行する前に、例の ID を有効な ID に置き換えます。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

AWS CloudHSM キーストアが接続されていることを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、CustomKeyStoreId または CustomKeyStoreName パラメータのどちらかを使用して (両方は使用できません) レスポンスを特定のカスタムキーストアに制限できます。CONNECTED の ConnectionState 値は、カスタムキーストアがその AWS CloudHSM クラスターに接続されていることを示します。

> **Note**
>
> AWS CloudHSM キーストアと外部のキーストアを区別するために、DescribeCustomKeyStores レスポンスに CustomKeyStoreType フィールドが追加されました。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

ConnectionState 値が [FAILED] の場合、ConnectionErrorCode 要素が失敗の原因を示します。この場合は、AWS KMS が、クラスター ID cluster-1a23b4cdefg のアカウントで AWS CloudHSM クラスターを見つけることができませんでした。クラスターを削除した場合、元のクラスターのバックアップから復元することができ、その後でカスタムキーストアのクラスター ID を編集できます。接続エラーコードの対処方法については「接続障害の修復方法」を参照してください。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
      "ConnectionErrorCode": "CLUSTER_NOT_FOUND"
   ],
}
```

次の手順: AWS CloudHSM キーストアでの KMS キーの作成

AWS CloudHSM キーストアを切断する (コンソール)

AWS Management Console で接続済みの AWS CloudHSM キーストアを切断するには、まず、[Custom Key Stores] (カスタムキーストア) ページで AWS CloudHSM キーストアを選択します。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。
4. 切断する外部キーストアの行を選択します。
5. [Key store actions] (キーストアアクション) メニューから [Disconnect] (切断) を選択します。

操作が完了すると、接続状態が [切断] から [切断済み] に変わります。オペレーションが失敗した場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「カスタムキーストアのトラブルシューティング」を参照してください。

AWS CloudHSM キーストアを切断する (API)

接続されたAWS CloudHSMキーストアを切断するには、 DisconnectCustomKeyStoreオペレーションを使用します。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

この例では、AWS CloudHSM キーストアを切断します。この例を実行する前に、例の ID を有効な ID に置き換えます。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

AWS CloudHSM キーストアが切断されていることを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、CustomKeyStoreId または CustomKeyStoreName パラメータ のどちらかを使用して (両方は使用できません) レスポンスを特定のカスタムキーストアに制限できます。DISCONNECTED の ConnectionState 値は、この例の AWS CloudHSM キーストアが AWS CloudHSM クラスターから切断されていないことを示します。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
```

```
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>"
   ],
}
```

AWS CloudHSM キーストアの削除

AWS CloudHSM キーストアを削除すると、AWS KMS は、AWS CloudHSM クラスターと関連付けられている情報を含め、AWS CloudHSM キーストアに関するすべてのメタデータを KMS から削除します。このオペレーションは、AWS CloudHSM クラスター、その HSM、またはそのユーザーには影響しません。同じ AWS CloudHSM クラスターに関連付けられた、新しい AWS CloudHSM キーストアを作成することはできますが、削除オペレーションを元に戻すことはできません。

削除できるのは、AWS CloudHSM クラスターから切断され、AWS KMS keys を含んでいない AWS CloudHSM キーストアのみです。カスタムキーストアを削除する前に、次の手順を実行します。

- いずれの暗号化オペレーションでも、キーストア内で KMS キーを一切使用する必要がないことを確認します。次に、キーストアからのすべての KMS キーの削除をスケジュールします。AWS CloudHSM キーストアで KMS キーを検索する方法については、「AWS CloudHSM キーストアで KMS キーを検索する」を参照してください。
- すべての KMS キーが削除されたことを確認します。AWS CloudHSM キーストアで KMS キーを表示するには、「AWS CloudHSM キーストアでの KMS キーの表示」を参照してください。
- AWS CloudHSM クラスターからAWS CloudHSM キーストアを切断します。

AWS CloudHSM キーストアを削除する代わりに、関連付けられた AWS CloudHSM クラスターから切断することを検討します。AWS CloudHSM キーストアが切断されている間は、AWS CloudHSM キーストアとその AWS KMS keys を管理できます。ただし、AWS CloudHSM キーストアで KMS キーを作成または使用することはできません。AWS CloudHSM キーストアはいつでも再接続できます。

トピック

- AWS CloudHSM キーストアを削除する (コンソール)
- AWS CloudHSM キーストアを削除する (API)

AWS CloudHSM キーストアを削除する (コンソール)

AWS Management Console で AWS CloudHSM キーストアを削除するには、まず、[Custom key stores] (カスタムキーストア) ページで AWS CloudHSM キーストアを選択します。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタムキーストア]、[AWS CloudHSM キーストア] の順に選択します。
4. 削除する AWS CloudHSM キーストアを表示している行を探します。AWS CloudHSM キーストアの [接続の状態] が [切断済み] になっていない場合は、AWS CloudHSM キーストアを切断してから削除する必要があります。
5. [Key store actions] (キーストアアクション) メニューから [Delete] (削除) を選択します。

オペレーションが完了すると成功メッセージが表示され、この AWS CloudHSM キーストアはカスタムキーストアリストに表示されなくなります。オペレーションが正常に行われなかった場合、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「カスタムキーストアのトラブルシューティング」を参照してください。

AWS CloudHSM キーストアを削除する (API)

AWS CloudHSM キーストアを削除するには、 DeleteCustomKeyStoreオペレーションを使用します。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。

まず、AWS CloudHSM キーストアに AWS KMS keys が含まれていないことを確認します。KMS キーが含まれているカスタムキーストアを削除することはできません。最初のコマンド例では、ListKeysと を使用してDescribeKey、*cks-1234567890abcdef0* カスタムAWS CloudHSMキーストア ID の例を含む キーストアAWS KMS keysで を検索します。この場合、コマンドは KMS キーを返しません。その場合は、 ScheduleKeyDeletionオペレーションを使用して、各 KMS キーの削除をスケジュールします。

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
```

```
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

次に、AWS CloudHSM キーストアを切断します。このコマンド例では、DisconnectCustomKeyStoreオペレーションを使用して、 AWS CloudHSMキーストアをAWS CloudHSMクラスターから切断します。このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

カスタムキーストアが切断されたら、 DeleteCustomKeyStoreオペレーションを使用して削除できます。

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

CloudHSM キーストアでの KMS キーの管理

AWS CloudHSM キーストアでは、AWS KMS keys の作成、表示、管理、使用と、削除のスケジュールが行えます。手順は、他の KMS キーを使用する際の手順に非常によく似ています。唯一の違いは、KMS キーを作成する際は AWS CloudHSM キーストアを指定する点です。次に、AWS KMS

は、AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスター内の KMS キーに対して、抽出不可能なキーマテリアルを作成します。AWS CloudHSM カスタムキーストアで KMS キーを使用するときは、暗号化オペレーションはクラスター内の HSM で実行されます。

サポートされている機能

このセクションで説明している手順のほかに、AWS CloudHSM キーストアでは KMS キーを使用して次の手順を実行できます。

- KMS キーへのアクセスを承認するときは、キーポリシー、IAM ポリシー、グラントを使用します。
- KMS キーを有効および無効にします。
- タグを割り当ててエイリアスを作成し、属性ベースのアクセス制御 (ABAC) を使用して KMS キーへのアクセスを承認します。
- 暗号化、復号、再暗号化、データキーの生成などの暗号化オペレーションには KMS キーを使用します。
- AWS KMS を統合し、カスタマーマネージドキーをサポートする AWS サービスで KMS キーを使用します。
- AWS CloudTrail ログおよび Amazon CloudWatch モニタリングツールでの KMS キーの使用を追跡します。

サポートされていない機能

- AWS CloudHSM キーストアは、対称暗号化 KMS キーのみをサポートします。AWS CloudHSM キーストアでは、HMAC KMS キー、非対称 KMS キー、非対称データキーペアは作成できません。
- AWS CloudHSM キーストア内の KMS キーにキーマテリアルをインポートすることはできません。AWS KMS は、KMS キーのキーマテリアルを AWS CloudHSM クラスター内に生成します。
- AWS CloudHSM キーストアで、KMS キーのキーマテリアルの自動ローテーションを有効または無効にすることはできません。

トピック

- AWS CloudHSM キーストアでの KMS キーの作成
- AWS CloudHSM キーストアでの KMS キーの表示
- AWS CloudHSM キーストアで KMS キーを使用する

- KMS キーとキーマテリアルを検索する
- KMS キーを AWS CloudHSM キーストアから削除することをスケジュールする

AWS CloudHSM キーストアでの KMS キーの作成

AWS CloudHSM キーストアを作成すると、キーストアで AWS KMS keys を作成できます。これらは、AWS KMS が生成するキーマテリアルを持つ対称暗号化 KMS キーである必要があります。カスタムキーストアで非対称 KMS キー、HMAC KMS キー、またはインポートされたキーマテリアルを持つ KMS キーを作成することはできません。カスタムキーストア内の対称暗号化 KMS キーを使用して、非対称データキーペアを生成することもできません。

AWS CloudHSM キーストアで KMS キーを作成するには、AWS CloudHSM キーストアが、関連付けられた AWS CloudHSM クラスターに接続され、このクラスターに、アベイラビリティーゾーンの異なる 2 つ以上のアクティブな HSM が含まれている必要があります。接続ステータスと HSM の数を確認するには、AWS CloudHSM キーストアのページを AWS Management Console に表示します。API オペレーションを使用する場合は、 DescribeCustomKeyStoresオペレーションを使用して、 AWS CloudHSM キーストアが接続されていることを確認します。クラスター内のアクティブな HSMs の数とそのアベイラビリティーゾーンを確認するには、 AWS CloudHSMDescribeClustersオペレーションを使用します。

AWS CloudHSM キーストアで KMS キーを作成すると、AWS KMS が AWS KMS で KMS キーを作成します。ただし、関連付けられた AWS CloudHSM クラスターで KMS キーのキーマテリアルが作成されます。具体的には、AWS KMS が作成した `kmsuser` CU としてクラスターにサインインします。次に、クラスター内に永続的で抽出不可能な 256 ビットの Advanced Encryption Standard (AES) 対称キーが作成され、AWS KMS がキーラベルの属性値を設定します。これはクラスター内で、KMS キーの Amazon リソースネーム (ARN) にのみ表示されます。

コマンドが成功すると、新しいKMS キーのキーステータスは `Enabled` になり、そのオリジンは `AWS_CLOUDHSM` になります。作成後に KMS キーのオリジンを変更することはできません。AWS KMS コンソールで AWS CloudHSMキーストアの KMS キーを表示するか、 DescribeKeyオペレーションを使用すると、キー ID、キーステータス、作成日などの一般的なプロパティを確認できます。カスタムキーストア ID と AWS CloudHSM クラスター ID (オプション) を確認することもできます。詳細については、「AWS CloudHSM キーストアでの KMS キーの表示」を参照してください。

AWS CloudHSM キーストアで KMS キーを作成しようとして失敗した場合は、エラーメッセージを使えばその原因を特定できます。メッセージには、AWS CloudHSM キーストアが接続されていない (`CustomKeyStoreInvalidStateException`)、または、関連付けられた AWS CloudHSM クラスターにオペレーションに必要な 2 つのアクティブな HSM がない

(CloudHsmClusterInvalidConfigurationException)、との原因が記されている可能性があります。ヘルプについては、を参照してください カスタムキーストアのトラブルシューティング。

AWS CloudHSM キーストアで KMS キーを作成するオペレーションの、AWS CloudTrail ログの例については、「CreateKey」を参照してください。

トピック

- AWS CloudHSM キーストアで KMS キーを作成する (コンソール)
- AWS CloudHSM キーストアで KMS キーを作成する (API)

AWS CloudHSM キーストアで KMS キーを作成する (コンソール)

AWS CloudHSM キーストアで対称暗号化 KMS キーを作成するときは、次の手順に従います。

> **Note**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. [Create key] (キーの作成) を選択します。
5. [対称] を選択します。
6. [Key usage] (キーの使用) では、[Encrypt and decrypt] (暗号化および復号化) オプションがすでに選択されています。この設定は変更しないでください。
7. [Advanced options (詳細オプション)] を選択します。
8. [キーマテリアルのオリジン] で、[AWS CloudHSM キーストア] を選択します。

   マルチリージョンキーは、AWS CloudHSM キーストアでは作成できません。
9. [次へ] をクリックします。
10. 新しい KMS キーの AWS CloudHSM キーストアを選択します。新しい AWS CloudHSM キーストアを作成するには、[Create custom key store] (カスタムキーストアの作成) を選択します。

選択する AWS CloudHSM キーストアは、ステータスが [接続済み] になっている必要があります。関連付けられた AWS CloudHSM クラスターがアクティブで、異なるアベイラビリティーゾーンに少なくとも 2 つのアクティブな HSM が含まれている必要があります。

AWS CloudHSM キーストアの接続に関するヘルプは、「AWS CloudHSM キーストアの接続と切断」を参照してください。HSM の追加については、AWS CloudHSM ユーザーガイドの HSM の追加を参照してください。

11. [次へ] をクリックします。
12. KMS キーのエイリアスおよびオプションの説明を入力します。
13. (オプション)。[Add Tags] (タグの追加) ページで、KMS キーを識別または分類するタグを追加します。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。
14. [次へ] をクリックします。
15. [Key administrators] (キー管理者) セクションで、KMS キーを管理できる IAM ユーザーとロールを選択します。詳細については、「KMS キーの管理をキー管理者に許可する」を参照してください。

    > **Note**
    >
    > IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。
    > IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。
16. (オプション) これらのキー管理者がこの KMS キーを削除できないようにするには、ページの下部にある [Allow key administrators to delete this key] (キー管理者がこのキーを削除できるようにする) チェックボックスをオフにします。
17. [次へ] をクリックします。

18. [This account] (このアカウント) セクションで、KMS キーを暗号化オペレーションで使用できる、この AWS アカウント の IAM ユーザーとロールを選択します。詳細については、「KMS キーの使用をキーユーザーに許可する」を参照してください。

> **Note**
>
> IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

19. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

> **Note**
>
> ユーザーが IAM ポリシーを作成して KMS キーにアクセスすることを、他の AWS アカウント 管理者が許可する必要もあります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

20. [次へ] を選択します。
21. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。
22. 終了したら、[Finish] (完了) を選択し、キーを作成します。

この手順が完了すると、選択した AWS CloudHSM キーストアに新しい KMS キーが表示されます。新しい KMS キーの名前やエイリアスを選択すると、その詳細ページの [Cryptographic configuration] (暗号化設定) タブに、KMS キー (AWS CloudHSM) のオリジン、カスタムキーストアの名前、ID、タイプ、AWS CloudHSM クラスターの ID が表示されます。手順が失敗すると、失敗を説明するエラーメッセージが表示されます。

> **Tip**
>
> カスタムキーストアで KMS キーをより簡単に識別できるようにするには、[Customer managed keys (カスタマーマネージドキー)] ページで、[Custom key store ID (カスタムキーストア ID)] 列を表示に追加します。右上隅にある歯車アイコンをクリックし、[Custom key store ID (カスタムキーストア ID)] を選択します。詳細については、「KMS キーテーブルをカスタマイズする」を参照してください。

AWS CloudHSM キーストアで KMS キーを作成する (API)

キーストアに新しい AWS KMS key (KMS AWS CloudHSM キー) を作成するには、 CreateKeyオペレーションを使用します。`CustomKeyStoreId` パラメータを使用してカスタムキーストアを識別し、`AWS_CLOUDHSM` の `Origin` 値を指定します。

また、キーポリシーを指定するために `Policy` パラメータが必要になる場合もあります。キーポリシー (PutKeyPolicy) を変更し、説明やタグなどのオプション要素をいつでも追加できます。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

次の例では、 DescribeCustomKeyStoresオペレーションの呼び出しから始めて、AWS CloudHSM キーストアが関連付けられた AWS CloudHSM クラスターに接続されていることを確認します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。特定の AWS CloudHSM キーストアのみを記述するときは、`CustomKeyStoreId` か `CustomKeyStoreName` のパラメータ (一方のみ) を使用します。

このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

> **Note**
>
> `Description` フィールドまたは `Tags` フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
```

```
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

次のコマンド例では、 DescribeClustersオペレーションを使用して、 ExampleKeyStore (cluster-1a23b4cdefg) に関連付けられているAWS CloudHSMクラスターに少なくとも 2 つのアクティブな HSMs があることを確認します。クラスターにある HSM が 2 つに満たない場合、CreateKey オペレーションは失敗します。

```
$ aws cloudhsmv2 describe-clusters
{
    "Clusters": [
        {
            "SubnetMapping": {
               ...
            },
            "CreateTimestamp": 1507133412.351,
            "ClusterId": "cluster-1a23b4cdefg",
            "SecurityGroup": "sg-865af2fb",
            "HsmType": "hsm1.medium",
            "VpcId": "vpc-1a2b3c4d",
            "BackupPolicy": "DEFAULT",
            "Certificates": {
                "ClusterCertificate": "-----BEGIN CERTIFICATE-----\...\n-----END
 CERTIFICATE-----\n"
            },
            "Hsms": [
                {
                    "AvailabilityZone": "us-west-2a",
                    "EniIp": "10.0.1.11",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-a6b10bd1",
                    "HsmId": "hsm-abcdefghijk",
                    "State": "ACTIVE"
                },
                {
```

```
                    "AvailabilityZone": "us-west-2b",
                    "EniIp": "10.0.0.2",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-b6b10bd2",
                    "HsmId": "hsm-zyxwvutsrqp",
                    "State": "ACTIVE"
                },
            ],
            "State": "ACTIVE"
        }
    ]
}
```

このコマンド例では、 CreateKeyオペレーションを使用して キーストアに KMS AWS CloudHSM キーを作成します。AWS CloudHSM キーストアに KMS キーを作成するときは、AWS CloudHSM キーストアのカスタムキーストア ID を入力し、`AWS_CLOUDHSM` の `Origin` 値を指定する必要があります。

応答には、カスタムキーストアと AWS CloudHSM クラスターの ID が含まれています。

このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

```
$ aws kms create-key --origin AWS_CLOUDHSM --custom-key-store-id cks-1234567890abcdef0
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1.499288695918E9,
    "Description": "Example key",
    "Enabled": true,
    "MultiRegion": false,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_CLOUDHSM"
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CustomKeyStoreId": "cks-1234567890abcdef0"
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
```

```
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

AWS CloudHSM キーストアでの KMS キーの表示

AWS CloudHSM キーストアで AWS KMS keys を表示するときは、AWS KMS カスタマーマネージドキーを表示するときと同じ方法を使用します。基本については、「キーの表示」を参照してください。KMS キーのキーマテリアルとして機能する AWS CloudHSM クラスターでキーを識別するには、KMS キーとキーマテリアルを検索する を参照してください。カスタムキーストアでのすべての API オペレーションを記録する AWS CloudTrail ログの表示方法については、「AWS KMS による AWS CloudTrail API コールのログ記録」を参照してください。

AWS KMS コンソールでは、カスタマーキーストア内の KMS キーは、AWS アカウント およびリージョンにある他のすべてのカスタマーマネージドキーと一緒に [Customer managed keys] (カスタマーマネージドキー) ページに表示されます。

ただし、次の値は AWS CloudHSM キーストアの KMS キーに固有です。

- KMS キーを保存する AWS CloudHSM キーストアの名前と ID。
- キーマテリアルを含む、関連付けられた AWS CloudHSM クラスターのクラスター ID。
- AWS KMS コンソールの `AWS CloudHSM` の `Origin` 値、または API レスポンスの `AWS_CLOUDHSM`。
- キーストアの値は `Unavailable` である可能性があります。ステータスの解決については、を参照してください 使用できない KMS キーを修正するには。

AWS CloudHSM キーストアで KMS キーを表示するには (コンソール)

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 右上で歯車アイコンを選択し、[カスタムキーストア ID] および [オリジン] を選択して、[確認] を選択します。
5. 任意の AWS CloudHSM キーストアで KMS キーを識別するには、AWS CloudHSM の [Origin] (オリジン) 値で KMS キーを特定します。特定の AWS CloudHSM キーストアで KMS キーを識別するには、[Custom key store ID] (カスタムキーストア ID) 列の値を表示します。

6. AWS CloudHSM キーストアで、KMS キーのエイリアスまたはキー ID を選択します。

   このページでは、Amazon リソースネーム (ARN)、キーポリシー、タグを含む、KMS キーに関する詳細情報が表示されます。

7. [Cryptographic configuration] (暗号化の設定) タブを選択します。これらのタブは、[General configuration] (一般設定) セクションの下にあります。

   このセクションには、KMS キーの AWS CloudHSM キーストアと AWS CloudHSM クラスターに関する情報が含まれています。

カスタムキーストアで KMS キーを表示するには (API)

、ListKeys、 など、任意の KMS キーに使用する AWS CloudHSMキーストアの KMS キーを表示するにはDescribeKey、同じ AWS KMS API オペレーションを使用しますGetKeyPolicy。例えば、次の AWS CLI の `describe-key` オペレーションでは、AWS CloudHSM キーストアの KMS キーの特別なフィールドが表示されます。このようなコマンドを実行する前に、サンプル KMS キー ID を有効な値に置き換えます。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

{
   "KeyMetadata": {
      "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "AWSAccountId": "111122223333",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CreationDate": 1537582718.431,
      "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "Description": "Key in custom key store",
      "Enabled": true,
      "EncryptionAlgorithms": [
         "SYMMETRIC_DEFAULT"
      ],
      "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
      "KeyManager": "CUSTOMER",
      "KeySpec": "SYMMETRIC_DEFAULT",
      "KeyState": "Enabled",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "MultiRegion": false,
      "Origin": "AWS_CLOUDHSM"
```

```
    }
}
```

AWS CloudHSM キーストアで KMS キーを検索したり、KMS キーのキーマテリアルとして機能する AWS CloudHSM クラスターでキーを識別したりする方法については、「KMS キーとキーマテリアルを検索する」を参照してください。

AWS CloudHSM キーストアで KMS キーを使用する

AWS CloudHSM キーストアで対称暗号化 KMS キーを作成すれば、それを、以下の暗号化オペレーションで使用できます。

- 暗号化
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

非対称データキーペア GenerateDataKeyPairおよび を生成するオペレーション はGenerateDataKeyPairWithoutPlaintext、カスタムキーストアではサポートされていません。

リクエストで KMS キーを使用するときは、ID またはエイリアスで KMS キーを識別します。AWS CloudHSM キーストアまたは AWS CloudHSM クラスターを指定する必要はありません。レスポンスには、対称暗号化 KMS キーについて返されるものと同じフィールドが含まれます。

ただし、AWS CloudHSM キーストアで KMS キーを使用すると、暗号化オペレーションは、AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスターの内部のみで実行されます。オペレーションでは、選択した KMS キーに関連付けられているクラスターのキーマテリアルが使用されます。

これを可能にするには、次の条件が必要です。

- KMS キーのキーストアは `Enabled` である必要があります。キーステータスを確認するには、AWS KMSコンソールの Status フィールドまたはDescribeKeyレスポンスの `KeyState`フィールドを使用します。
- AWS CloudHSM キーストアは、その AWS CloudHSM クラスターに接続されている必要があります。AWS KMS コンソールまたは`ConnectionState`DescribeCustomKeyStoresレスポンスのステータスは である必要がありま`CONNECTED`す。

- カスタムキーストアに関連付けられている AWS CloudHSM クラスターには、少なくとも 1 つのアクティブな HSM が含まれている必要があります。クラスター内のアクティブな HSMs の数を確認するには、AWS KMSコンソール、AWS CloudHSMコンソール、または DescribeClustersオペレーションを使用します。
- AWS CloudHSM クラスターには KMS キーのキーマテリアルを含める必要があります。キーマテリアルがクラスターから削除された場合、または HSM がキーマテリアルを含まないバックアップから作成された場合、暗号化オペレーションは失敗します。

これらの条件が満たされていない場合、暗号化オペレーションは失敗し、AWS KMS は `KMSInvalidStateException` 例外を返します。通常は、AWS CloudHSM キーストアを再接続するだけで済みます。その他のヘルプについては、「失敗した KMS キーを修正するには」を参照してください。

AWS CloudHSM キーストアで KMS キーを使用するときは、各 AWS CloudHSM キーストア内の KMS キーが、暗号化オペレーションで、カスタムキーストアのリクエストクォータを共有することに注意する必要があります。クォータを超えた場合、AWS KMS は `ThrottlingException` を返します。AWS CloudHSM キーストアに関連付けられている AWS CloudHSM クラスターが、AWS CloudHSM キーストアに関連付けられていないものを含む膨大な数のコマンドを処理すると、`ThrottlingException` が発生する割合がさらに低くなる可能性があります。すべてのリクエストで `ThrottlingException` が表示される場合、リクエスト速度を下げ、再度コマンドを試してください。カスタムキーストアのリクエストクォータの詳細については、「カスタムキーストアのリクエストクォータ」を参照してください。

KMS キーとキーマテリアルを検索する

AWS CloudHSM キーストアを管理するときに、各 AWS CloudHSM キーストア内の KMS キーの識別が必要になる場合があります。例えば、次のタスクの一部を実行する必要がある場合があります。

- AWS CloudTrail ログで、AWS CloudHSM キーストアの KMS キーを追跡します。
- AWS CloudHSM キーストアの切断による KMS キーへの影響を予想します。
- AWS CloudHSM キーストアを削除する前に、KMS キーの削除をスケジュールします。

また、KMS キーのキーマテリアルとして機能する AWS CloudHSM クラスター内のキーを特定してください。AWS KMS は KMS キーとキーマテリアルを管理しますが、AWS CloudHSM クラスター、HSM とバックアップ、HSM のキーの制御と管理に関する責任は引き続きユーザーが保持します。キーマテリアルを監査したり、誤って削除されないようにしたり、KMS キーの削除後に HSM やクラスターバックアップから削除したりするために、キーを識別する必要がある場合があります。

AWS CloudHSM キーストア内にある KMS キーの、すべてのキーマテリアルは、kmsuser Crypto User (CU) が所有します。AWS KMS は、KMS キーの Amazon リソースネーム (ARN) に対して、AWS CloudHSM のみで閲覧できるキーラベルの属性を設定します。

KMS キーとキーマテリアルを検索するには、次のいずれかの方法を使用します。

- AWS CloudHSM キーストアで KMS キーを検索する — 1 つまたは全部の AWS CloudHSM キーストアの KMS キーを識別する方法。
- AWS CloudHSM キーストアのすべてのキーを検索する — AWS CloudHSM キーストアで KMS キーのキーマテリアルとして機能する、クラスター内すべてのキーを検索する方法。
- KMS キーの AWS CloudHSM キーを検索する — AWS CloudHSM キーストアで特定の KMS キーのキーマテリアルとして機能する、クラスターのキーを検索する方法。
- AWS CloudHSM キーの KMS キーを検索する — クラスターで特定のキーの KMS キーを検索する方法。

AWS CloudHSM キーストアで KMS キーを検索する

AWS CloudHSM キーストアを管理するときに、各 AWS CloudHSM キーストア内の KMS キーの識別が必要になる場合があります。この情報を使用すると、AWS CloudTrail ログで KMS キーオペレーションを追跡したり、カスタムキーストアの切断による KMS キーへの影響を予想したり、AWS CloudHSM キーストアを削除する前に KMS キーの削除をスケジュールしたりできます。

AWS CloudHSM キーストアで KMS キーを検索するには (コンソール)

特定の AWS CloudHSM キーストアで KMS キーを検索するには、[Customer Managed Keys] (カスタマーマネージドキー) ページで、[Custom Key Store Name] (カスタムキーストア名) フィールドまたは [Custom Key Store ID] (カスタムキーストア ID) フィールドの値を表示します。任意の AWS CloudHSM キーストアで KMS キーを識別するには、AWS CloudHSM の [Origin] (オリジン) 値で KMS キーを特定します。オプションの列をディスプレイに追加するには、ページの右上隅にある歯車アイコンを選択します。

AWS CloudHSM キーストアで KMS キーを検索するには (API)

キーストアで KMS AWS CloudHSMキーを検索するには、 ListKeys および DescribeKeyオペレーションを使用し、`CustomKeyStoreId`値でフィルタリングします。例を実行する前に、架空のカスタムキーストア ID の値を有効な値に置き換えます。

Bash

特定の AWS CloudHSM キーストアで KMS キーを検索するには、アカウントとリージョンですべての KMS キーを取得します。次に、カスタムキーストア ID でフィルタをかけます。

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

アカウントおよびリージョン内の任意の AWS CloudHSM キーストアで KMS キーを取得するには、`AWS_CloudHSM` 値を持つ `CustomKeyStoreType` を検索します。

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

特定のキーストアで KMS AWS CloudHSMキーを検索するには、Get-KmsKeyListcmdlets と Get-KmsKeycmdlets を使用して、アカウントとリージョンのすべての KMS キーを取得します。次に、カスタムキーストア ID でフィルタをかけます。

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

アカウントとリージョンの任意のキーストアで KMS AWS CloudHSMキーを取得するには、 の CustomKeyStoreType 値をフィルタリングしますAWS_CLOUDHSM。

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

AWS CloudHSM キーストアのすべてのキーを検索する

AWS CloudHSM クラスターで AWS CloudHSM キーストアのキーマテリアルとして機能するキーを、識別することができます。そのためには、cloudhsm_mgmt_util の findAllKeys コマンドを使用して、 がkmsuser所有または共有するすべてのキーのキーハンドルを見つけます。kmsuser としてログインせず、また AWS KMS の外部でキーを作成していなければ、kmsuser が所有するすべてのキーは KMS キーのキーマテリアルを表します。

クラスター内のすべての Crypto Officer は、AWS CloudHSM キーストアを切断することなくこのコマンドを実行できます。

1. cloudhsm_mgmt_util を、「Getting started with CloudHSM Management Utility (CMU)」(CloudHSM 管理ユーティリティ (CMU) の使用方法) のトピックに記載された手順に従って起動します。
2. Crypto Officer (CO) アカウントを使用してcloudhsm_mgmt_util にログインします。
3. listUsers コマンドを使用して、 kmsuser 暗号ユーザのユーザ ID を検索します。

   この例では、kmsuser にユーザー ID 3 があります。

```
aws-cloudhsm> listUsers
Users on server 0(10.0.0.1):
Number of users found:3

    User Id             User Type       User Name            MofnPubKey
 LoginFailureCnt         2FA
         1              PCO             admin                    NO
 0               NO
         2              AU              app_user                 NO
 0               NO
         3              CU              kmsuser                  NO
 0               NO
```

4. findAllKeys コマンドを使用して、 がkmsuser所有または共有するすべてのキーのキーハンドルを検索します。例にあるユーザー ID (3) を、クラスター内の kmsuser の、実際のユーザー ID に置き換えます。

   出力例では、クラスター内の両方の HSM で、kmsuser が、キーハンドルが 8、9、および 262162 のキーを所有していることを示しています。

```
aws-cloudhsm> findAllKeys 3 0
Keys on server 0(10.0.0.1):
Number of keys found 3
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 0(10.0.0.1)

Keys on server 1(10.0.0.2):
Number of keys found 6
number of keys matched from start index 0::6
8,9,262162
findAllKeys success on server 1(10.0.0.2)
```

AWS CloudHSM キーの KMS キーを検索する

kmsuser がクラスター内で所有するキーのキーハンドルがわかっている場合は、キーラベルを使用すれば、AWS CloudHSM キーストアにある、関連付けられた KMS キーを識別できます。

AWS KMS が KMS キーのキーマテリアルを AWS CloudHSM クラスターで作成すると、キーラベルに KMS キーの Amazon リソースネーム (ARN) が書き込まれます。ラベル値を変更しない限り、key_mgmt_util または cloudhsm_mgmt_util の getAttribute コマンドを使用して、キーを KMS キーに関連付けることができます。

この手順を実行するときは、kmsuser CU としてログインできるよう AWS CloudHSM キーストアを一時的に切断する必要があります。

> **Note**
>
> カスタムキーストアが切断されている間は、カスタムキーストアで KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする試みはすべて失敗します。このオペレーションにより、ユーザーが機密データを保存したりアクセスしたりすることを防ぐことができます。

1. AWS CloudHSM キーストアがまだ切断されていなければ切断し、切断してログインする方法 の説明に従って kmsuser として key_mgmt_util にログインします。
2. key_mgmt_util または cloudhsm_mgmt_util で getAttribute コマンドを使用して、特定のキーハンドルのためのラベル属性 (OBJ_ATTR_LABEL、属性 3) を取得します。

   例えば、このコマンドは、cloudhsm_mgmt_util で getAttribute を使用して、キーハンドルが 3 のキーのラベルの属性 (属性262162) を取得します。出力は、キー 262162 が ARN arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab を使用して、KMS キーのキーマテリアルとして機能することを示します。このコマンドを実行する前に、キーハンドル例を有効なキーハンドルに置き換えます。

   キー属性のリストについては、listAttributes コマンドを使用するか、AWS CloudHSMユーザーガイドのキー属性リファレンスを参照してください。

   ```
   aws-cloudhsm> getAttribute 262162 3

   Attribute Value on server 0(10.0.1.10):
   OBJ_ATTR_LABEL
   ```

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

3. key_mgmt_util または cloudhsm_mgmt_util からログアウトし、「ログアウトして再接続する方法」の説明に従って AWS CloudHSM キーストアを再接続します。

KMS キーの AWS CloudHSM キーを検索する

AWS CloudHSM キーストア内で KMS キーの KMS キー ID を使用すれば、キーマテリアルとして機能する AWS CloudHSM クラスターでキーを識別できます。その後、そのキーハンドルを使用して、AWS CloudHSM クライアントコマンドでキーを識別できます。

AWS KMS が KMS キーのキーマテリアルを AWS CloudHSM クラスターで作成すると、キーラベルに KMS キーの Amazon リソースネーム (ARN) が書き込まれます。ラベル値を変更しない限り、key_mgmt_util の findKey コマンドを使用して、KMS キーのキーマテリアルのキーハンドルを取得できます。この手順を実行するときは、kmsuser CU としてログインできるよう AWS CloudHSM キーストアを一時的に切断する必要があります。

> **Note**
>
> カスタムキーストアが切断されている間は、カスタムキーストアで KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする試みはすべて失敗します。このオペレーションにより、ユーザーが機密データを保存したりアクセスしたりすることを防ぐことができます。

1. AWS CloudHSM キーストアがまだ切断されていなければ切断し、切断してログインする方法 の説明に従って kmsuser として key_mgmt_util にログイン します。
2. key_mgmt_util の findKey コマンドを使用して、AWS CloudHSM キーストア内で KMS キーの ARN に一致するラベルを持つキーを検索します。-l (「ラベル」の L は小文字) パラメータ値のサンプル KMS キー ARN を、有効な KMS キー ARN と置き換えます。

   例えば、このコマンドでは、サンプル KMS キー ARN、arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab と一致するラベルを持つキーを検索します。出力例は、キーハンドル 262162 のキーが、そのラベルに指定された KMS キー ARN を持つことを示しています。他の key_mgmt_util コマンドで、このキーハンドルを使用できるようになりました。

```
Command: findKey -l arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

```
Total number of keys present 1

 number of keys matched from start index 0::1
262162

        Cluster Error Status
        Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
        Node id 0 and err state 0x00000000 : HSM Return: SUCCESS

        Cfm3FindKey returned: 0x00 : HSM Return: SUCCESS
```

3. key_mgmt_util からログアウトし、の説明に従ってカスタムキーストアを再接続 ログアウトして再接続する方法します。

KMS キーを AWS CloudHSM キーストアから削除することをスケジュールする

暗号化オペレーションに AWS KMS key を使用する必要がないことが確実な場合は、KMS キーの削除をスケジュールできます。AWS KMS からの KMS キーの削除をスケジュールするのと同じ手順を使用します。さらに、AWS CloudHSM キーストアの接続を維持します。そうすれば AWS KMS は、待機期間の終了後に、関連付けられた AWS CloudHSM クラスターから対応するキーマテリアルを削除できます。

KMS キーのスケジューリング、キャンセル、削除は、AWS CloudTrail ログでモニタリングできます。

> **Warning**
>
> KMS キーの削除は、破壊的で潜在的に危険なオペレーションであり、これを実行すると KMS キーで暗号化されたすべてのデータを回復できなくなります。KMS キーの削除をスケジュールする前に、KMS キーの過去の使用状況を調べ、削除保留中に誰かが KMS キーを使用しようとしたときに警告する Amazon CloudWatch アラームを作成します。可能な限り、削除ではなくKMS キーを無効化します。

AWS CloudHSM キーストアからの KMS キーの削除をスケジュールすると、[key state] (キーステータス) が [Pending deletion] (削除保留中) に変わります。KMS キーは、カスタムキーストアの切断によって KMS キーが使用できなくなった場合でも、待期期間中を通して削除保留中ステータスを維持します。これにより、待機期間中はいつでも KMS キーの削除をキャンセルできます。

待機期間が終了すると、AWS KMS は AWS KMS から KMS キーを削除します。次に、AWS KMS では、関連付けられた AWS CloudHSM クラスターからキーマテリアルを可能な限り削除します。キーストアが AWS KMS から切断されるなど、AWS KMS でキーマテリアルを削除できない場合は、クラスターから手動で孤立したキーマテリアルを削除できます。

AWS KMS は、クラスターのバックアップからキーマテリアルを削除しません。AWS KMS から KMS キーを削除し、AWS CloudHSM クラスターからそのキーマテリアルを削除した場合でも、バックアップから作成されたクラスターには、削除したキーマテリアルが含まれている可能性があります。キーマテリアルを完全に削除するには、KMS キーの作成日を表示します。次に、キーのマテリアルを含む可能性のある すべてのクラスタバックアップを削除します 。

AWS CloudHSM キーストアから KMS キーを削除することをスケジュールすると、その KMS キーはただちに使用できなくなります (結果整合性の影響を受ける)。ただし、KMS キーで保護されたデータキーで暗号化されているリソースは、KMS キーが (データキーの復号などで) 再度使用されるまで、その影響を受けません。この問題は AWS のサービス に影響します。その多くが、リソースを保護するためにデータキーを使用しています。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

カスタムキーストアのトラブルシューティング

AWS CloudHSM キーストアは、常に使用可能で、回復力を持つように設計されています。ただし、AWS CloudHSM キーストアをオペレーション可能な状態に保つために、いくつかのエラー条件を修正しなければならない場合があります。

トピック

- 使用できない KMS キーを修正するには
- 失敗した KMS キーを修正するには
- 接続障害の修復方法
- 暗号化オペレーションの失敗に対応するには
- 無効な kmsuser 認証情報の修正方法
- 孤立したキーマテリアルを削除する方法
- KMS キーの削除されたキーマテリアルを復旧するには
- kmsuser としてログインする方法

使用できない KMS キーを修正するには

AWS CloudHSM キーストアの AWS KMS keys のキーステータスは、通常 Enabled です。すべての KMS キーと同様、AWS CloudHSM キーストアで KMS キーを無効にするか、削除するようにスケジュールすると、キーステータスが変更されます。ただし、他の KMS キーとは異なり、カスタムキーストアの KMS キーには、Unavailable のキーステータスもあります。

Unavailable のキーステータスには、KMS キーが、意図的に切断されたカスタムキーストアの内部に存在し、再接続の試み (実行している場合) に失敗したことを示しています。KMS キーは使用できませんが、KMS キーを表示および管理することはできます。ただし、暗号化オペレーションで使用することはできません。

KMS キーのキーステータスを確認するには、[Customer managed keys] (カスタマーマネージドキー) ページで、KMS キーの [Status] (ステータス) フィールドを表示します。または、DescribeKeyオペレーションを使用して、レスポンスで KeyState要素を表示します。詳細については、「キーの表示」を参照してください。

切断されたカスタムキーストアの KMS キーは、Unavailable または PendingDeletion のキーステータスとなります。カスタムキーストアが切断されている場合でも、カスタムキーストアからの削除がスケジュールされている KMS キーのキーステータスは Pending Deletion になります。これにより、カスタムキーストアに再接続することなく、スケジュールされたキーの削除をキャンセルできます。

使用できない KMS キーを修正するには、カスタムキーストアを再接続します。カスタムキーストアを再接続すると、カスタムキーストア内の KMS キーのキーステータスは、Enabled や Disabled などの以前のステータスに自動的に復元されます。削除保留中の KMS キーは PendingDeletion ステータスのままです。ただし、問題が解決しない間は、使用できない KMS キーを有効および無効にしても、キーステータスは変更されません。有効または無効のアクションは、キーが使用可能になったときにのみ適用されます。

接続の失敗に関するヘルプについては、「接続障害の修復方法」を参照してください。

失敗した KMS キーを修正するには

AWS CloudHSM キーストアで KMS キーを作成し使用する際の問題は、AWS CloudHSM カスタムキーストア、それに関連付けられた AWS CloudHSM クラスター、KMS キー、そのキーマテリアルのいずれかに起因します。

AWS CloudHSM キーストアが AWS CloudHSM クラスターから切断されると、カスタムキーストア内の KMS キーのキーステータスは Unavailable になります。切断された AWS CloudHSM キース

トアに KMS キーを作成するリクエストは、すべて CustomKeyStoreInvalidStateException の例外を返します。データキーを暗号化、復号、再暗号化、または生成するすべてのリクエストは、KMSInvalidStateException 例外を返します。この問題を修正するには、AWS CloudHSM キーストアを再接続します。

ただし、暗号化オペレーションのために AWS CloudHSM キーストアで KMS キーを使用する試みは、たとえキーステータスが Enabled で、AWS CloudHSM キーストアの接続ステータスが Connected であっても、失敗する場合があります。これは、以下のいずれかの条件によって発生する可能性があります。

- KMS キーのキーマテリアルが、関連付けられた AWS CloudHSM クラスターから削除された可能性があります。調査するには、KMS キーのキーマテリアルのキーハンドルを探し、必要に応じてキーマテリアルの復旧を試みます。
- すべての HSM は、AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスターから削除されました。暗号化オペレーションで AWS CloudHSM キーストアの KMS キーを使用するときは、その AWS CloudHSM クラスターにアクティブな HSM が 1 つ以上含まれている必要があります。AWS CloudHSM クラスター内の HSMs の数と状態を確認するには、 AWS CloudHSMコンソールまたは オペレーションを使用します。 DescribeClustersHSM をクラスターに追加するには、 AWS CloudHSMコンソールまたは CreateHsmオペレーションを使用します。
- AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスターが削除されました。この問題を解決するには、元のクラスターのバックアップ、または元のクラスターの作成に使用されたバックアップなど、元のクラスターに関連するバックアップからクラスターを作成します。次に、カスタムキーストアの設定で、クラスター ID を編集します。手順については、「KMS キーの削除されたキーマテリアルを復旧するには」を参照してください。
- このカスタムキーストアに関連付けられた AWS CloudHSM クラスターには、使用可能な PKCS #11 セッションがありません。通常これは、トラフィックの処理に追加のセッションが必要な大規模なバーストトラフィックが起きている間に発生します。PKCS #11 セッションに関するエラーメッセージで KMSInternalException に応答するときは、リクエストをいったん取り消してから、改めて試行します。

接続障害の修復方法

AWS CloudHSM クラスターに AWS CloudHSM キーストアを接続しようとしてオペレーションに失敗すると、AWS CloudHSM キーストアの接続ステータスは FAILED に変わります。AWS CloudHSM キーストアの接続状態を確認するには、 AWS KMSコンソールまたは DescribeCustomKeyStoresオペレーションを使用します。

また、簡単に検出できるクラスター設定エラーが原因で接続の試行がすぐに失敗することがあります。この場合、接続ステータスは DISCONNECTED のままです。これらのエラーは、試行が失敗した理由を説明するエラーメッセージまたは 例外 を返します。例外の内容とクラスターの要件を確認し、問題を修正し、必要に応じてAWS CloudHSM キーストアを更新したら、再度接続を実行します。

接続状態が の場合FAILED、 DescribeCustomKeyStoresオペレーションを実行し、レスポンスで ConnectionErrorCode要素を確認します。

> **Note**
>
> AWS CloudHSM キーストアの接続ステータスが FAILED の場合は、再度接続を試みる前に AWS CloudHSM カスタムキーストアを切断します。接続ステータスが FAILED の AWS CloudHSM キーストアは、接続できません。

- CLUSTER_NOT_FOUND は、AWS KMS が指定のクラスター ID を持つ AWS CloudHSM クラスターを見つけられないことを示します。これは、誤ったクラスター ID が API オペレーションに提供されたか、クラスターが削除されて置き換えられなかったために発生する可能性があります。このエラーを修正するには、AWS CloudHSMコンソールや DescribeClustersオペレーションを使用するなどして、クラスター ID を確認します。クラスタが削除された場合は、元の バックアップからクラスタを作成します 。次に、AWS CloudHSM キーストアを切断し、AWS CloudHSM キーストアのクラスター ID の 設定を編集して、AWS CloudHSM キーストアをこのクラスターに再接続します。
- INSUFFICIENT_CLOUDHSM_HSMS は、関連付けられた AWS CloudHSM クラスターに HSM が含まれていないことを示します。クラスターに接続するには、少なくとも 1 つの HSM を持っている必要があります。クラスター内の HSMsの数を確認するには、 DescribeClustersオペレーションを使用します。このエラーを解決するには、クラスターに 少なくとも 1 つの HSM を追加します 。複数の HSM を追加する場合は、別のアベイラビリティーゾーンでそれらを作成することをお勧めします。
- INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET は、クラスターに関連付けられている少なくとも 1 つのプライベートサブネットに、利用可能な IP アドレスがなかったため、AWS KMS が AWS CloudHSM キーストアをその AWS CloudHSM クラスターに接続できなかったことを示しています。AWS CloudHSM キーストアの接続には、関連付けられた各プライベートサブネットに空き IP アドレスが 1 つ (できれば 2 つ) 必要です。

既存のサブネットに IP アドレスを追加できません (CIDR ブロック)。可能であれば、未使用の EC2 インスタンスや Elastic Network Interface など、サブネット内の IP アドレスを使用している他のリソースを移動または削除します。または、より多くの空きアドレス空間のある新しいまたは既存のプライベートサブネットを持つ AWS CloudHSM クラスターの最近のバックアップからクラスターを作成することができます。次に、新しいクラスターを AWS CloudHSM キーストアに関連付けるため、カスタムキーストアを切断し、AWS CloudHSM キーストアのクラスター ID を新しいクラスターの ID に変更して、もう一度接続を試みます。

> **Tip**
>
> kmsuser パスワードをリセットしないようにするには、AWS CloudHSM クラスターの最新のバックアップを使用します。

- `INTERNAL_ERROR` は、内部エラーのために AWS KMS がリクエストを完了できなかったことを示します。リクエストを再試行します。`ConnectCustomKeyStore` リクエストの場合、AWS CloudHSM キーストアの接続を切断してから、再度接続を実行します。
- `INVALID_CREDENTIALS`は、適切な `kmsuser` アカウントのパスワードがないため、AWS KMS が関連付けられた AWS CloudHSM クラスターにログインできないことを示します。このエラーのヘルプについては、「無効な kmsuser 認証情報の修正方法」を参照してください。
- `NETWORK_ERRORS` は通常、一時的なネットワークの問題を示します。AWS CloudHSMキーストアを切断し、数分待ってから、再度接続します。
- `SUBNET_NOT_FOUND` は、AWS CloudHSM クラスター設定のサブネットが少なくとも 1 つ削除されたことを示します。AWS KMS が、クラスター構成内のすべてのサブネットを特定できない場合、AWS CloudHSM キーストアを AWS CloudHSM クラスターに接続する試みは失敗します。

  このエラーを修正するには、同じ AWS CloudHSM クラスターの最近のバックアップからクラスターを作成します。(このプロセスでは、VPC とプライベートサブネットを持つ新しいクラスター設定が作成されます)。新しいクラスターが カスタムキーストアの要件を満たしていることを確認し、新しいクラスター ID を書き留めます。次に、新しいクラスターを AWS CloudHSM キーストアに関連付けるため、カスタムキーストアを切断し、AWS CloudHSM キーストアのクラスター ID を新しいクラスターの ID に変更して、もう一度接続を試みます。

  > **Tip**
  >
  > kmsuser パスワードをリセットしないようにするには、AWS CloudHSM クラスターの最新のバックアップを使用します。

- USER_LOCKED_OUT は、失敗したパスワードの試行回数が多すぎるため、kmsuser Crypto User (CU) アカウントが関連付けられた AWS CloudHSM クラスターからロックアウトされたことを示します。このエラーのヘルプについては、「無効な kmsuser 認証情報の修正方法」を参照してください。

  このエラーを修正するには、AWS CloudHSM カスタムキーストアを切断し、cloudhsm_mgmt_util の changePswd コマンドを使用して kmsuser アカウントのパスワードを変更します。次に、カスタムキーストアの kmsuser のパスワード設定を編集し、接続し直してみてください。ヘルプについては、「無効な kmsuser 認証情報の修正方法 トピック」で説明されている手順を使用してください。

- USER_LOGGED_IN は、kmsuser CU アカウントが関連付けられた AWS CloudHSM クラスターにログインしていることを示します。これにより、AWS KMS が kmsuser アカウントパスワードをローテーションしてクラスターにログインするのを防ぐことができます。このエラーを修正するには、クラスターから kmsuser CU をログアウトします。クラスターにログインするために kmsuser パスワードを変更した場合は、AWS CloudHSM キーストアのキーストアパスワード値も更新する必要があります。ヘルプについては、「ログアウトして再接続する方法」を参照してください。

- USER_NOT_FOUND は、AWS KMS が関連付けられた AWS CloudHSM クラスターで kmsuser CU アカウントを見つけられないことを示します。このエラーを修正するには、クラスターで kmsuser CU アカウントを作成し、AWS CloudHSM　キーストアのキーストアパスワード値を更新します。ヘルプについては、「無効な kmsuser 認証情報の修正方法」を参照してください。

暗号化オペレーションの失敗に対応するには

カスタムキーストアで KMS キーを使用する暗号化オペレーション
は、KMSInvalidStateException で失敗することがあります。次のエラーメッセージに
は KMSInvalidStateException が関連することがあります。

KMS は CloudHSM クラスターと通信できません。これは一時的なネットワークの問題である可能性があります。このエラーが繰り返し表示される場合は、AWS CloudHSM クラスターの VPC のネットワーク ACL とセキュリティグループルールが正しいことを確認してください。

- これは HTTPS 400 エラーですが、一時的なネットワークの問題が原因である可能性があります。対応するには、まずリクエストを再試行します。それでも失敗する場合は、ネットワークコンポーネントの設定を調べます。このエラーはほとんどの場合、発信トラフィックをブロックしている

ファイアウォールルールや VPC セキュリティグループルールなど、ネットワークコンポーネントの誤った設定が原因です。

kmsuser がロックアウトされているため、KMS は AWS CloudHSM クラスターと通信できません。このエラーが繰り返し表示される場合は、AWS CloudHSM キーストアの接続を解除して、kmsuser アカウントのパスワードをリセットしてください。カスタムキーストアの kmsuser のパスワードを更新して、リクエストを再試行してください。

- このエラーメッセージは、失敗したパスワードの試行回数が多すぎるため、kmsuser Crypto User (CU) アカウントが関連付けられた AWS CloudHSM クラスターからロックアウトされたことを示します。このエラーのヘルプについては、「切断してログインする方法」を参照してください。

無効な **kmsuser** 認証情報の修正方法

AWS CloudHSM キーストアを接続すると、AWS KMSが、関連付けられた AWS CloudHSM クラスターに kmsuser Crypto User (CU) としてログインします。このログインは、AWS CloudHSM キーストアが切断されるまで維持されます。DescribeCustomKeyStores レスポンスは、次の例に示すように、INVALID_CREDENTIALS の FAILED と ConnectionErrorCode の値の ConnectionState を示します。

AWS CloudHSM キーストアを切断して kmsuser パスワードを変更すると、AWS KMS は、AWS CloudHSM クラスターに kmsuser CU アカウントの認証情報でログインすることができなくなります。その結果、AWS CloudHSM キーストアへの接続の試みはすべて失敗します。DescribeCustomKeyStores レスポンスは、次の例に示すように、INVALID_CREDENTIALS の FAILED と ConnectionErrorCode の値の ConnectionState を示します。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
```

```
}
```

また、誤ったパスワードでクラスターにログインしようとして 5 回失敗すると、AWS CloudHSM はユーザーアカウントをロックします。このクラスターにログインするには、アカウントのパスワードを変更する必要があります。

AWS KMS が kmsuser CU としてクラスターにログインしようとしてロックアウトのレスポンスを受け取った場合、AWS CloudHSM キーストアへの接続リクエストは失敗します。DescribeCustomKeyStores レスポンスには、次の例に示すようにUSER_LOCKED_OUT、ConnectionStateの FAILEDと ConnectionErrorCodeの値が含まれます。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "USER_LOCKED_OUT"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
}
```

これらの条件を修復するには、次の手順を実行します。

1. AWS CloudHSM キーストアを切断します。
2. DescribeCustomKeyStores オペレーションを実行し、レスポンスの ConnectionErrorCode要素の値を表示します。

   - ConnectionErrorCode の値が INVALID_CREDENTIALS の場合は、kmsuser アカウントの現在のパスワードを特定します。必要に応じて、cloudhsm_mgmt_util で changePswd コマンドを使用して、パスワードを既知の値に設定します。
   - ConnectionErrorCode 値が USER_LOCKED_OUT の場合、cloudhsm_mgmt_util で changePswd コマンドを使用して kmsuser パスワードを変更する必要があります。
3. kmsuser パスワード設定を編集して、クラスター内の現在の kmsuser パスワードと一致させます。このアクションは、AWS KMS にクラスターにログインするために使用するパスワードを指示します。クラスターの kmsuser パスワードは変更されません。
4. カスタムキーストアを接続します。

孤立したキーマテリアルを削除する方法

AWS CloudHSM キーストアから KMS キーを削除することをスケジュールした場合、対応するキーマテリアルを、関連付けられた AWS CloudHSM クラスターから手動で削除しなければならなくなることがあります。

AWS CloudHSM キーストアで KMS キーを作成すると、AWS KMS は AWS KMS で KMS キーメタデータを作成し、関連付けられた AWS CloudHSM クラスターでキーマテリアルを生成します。AWS CloudHSM キーストアで KMS キーの削除をスケジュールすると、待機期間の後、AWS KMS が KMS キーメタデータを削除します。その後、AWS KMS は、AWS CloudHSM クラスターから対応するキーマテリアルを可能な限り削除します。AWS CloudHSM キーストアから切断されていたり `kmsuser` パスワードが変更されていたりするなどして AWS KMS がクラスターにアクセスできない場合、この試みは失敗する可能性があります。AWS KMS は、クラスターのバックアップからキーマテリアルを削除することはありません。

AWS KMS は、クラスターからキーマテリアルを削除しようと試みた結果を AWS CloudTrail ログの `DeleteKey` イベントエントリに報告します。この結果は、次のエントリ例に示されているように、`additionalEventData` 要素の `backingKeysDeletionStatus` 要素に表示されます。エントリには、KMS キー ARN、AWS CloudHSM クラスター ID、およびキーマテリアルのキーハンドル (`backing-key-id`) が含まれます。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"keyHandle\":\"01\",\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"keyHandle\":\"16\",\"backingKeyId\":
\"backing-key-id\",\"deletionStatus\":\"FAILURE\"}]"
```

```
    },
    "eventID": "c21f1f47-f52b-4ffe-bff0-6d994403cf40",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```

関連する AWS CloudHSM クラスターからキーマテリアルを削除するには、次のような手順を使用します。この例では、AWS CLI および AWS CloudHSM コマンドラインツールを使用していますが、CLI の代わりに AWS Management Console を使用することもできます。

1. AWS CloudHSM キーストアがまだ切断されていなければ切断し、切断してログインする方法 の説明に従って key_mgmt_util にログイン します。
2. クラスター内の HSM から キーを削除するには、key_mgmt_util の DeleteKey コマンドを使用します。

   例えば、このコマンドは、クラスターの HSM 262162 からキーを削除します。キーハンドルは CloudTrail ログエントリに表示されます。

   ```
   Command: deleteKey -k 262162

           Cfm3DeleteKey returned: 0x00 : HSM Return: SUCCESS

           Cluster Error Status
           Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
           Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
           Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
   ```

3. key_mgmt_util からログアウトし、ログアウトして再接続する方法 の説明に従って AWS CloudHSM キーストアを再接続します。

KMS キーの削除されたキーマテリアルを復旧するには

AWS KMS key のキーマテリアルが削除された場合、KMS キーは使用できず、KMS キーで暗号化されたすべての暗号文は復号できません。こうしたことは、AWS CloudHSM キーストアにある KMS キーのキーマテリアルが、関連付けられた AWS CloudHSM クラスターから削除された場合に発生します。ただし、キーマテリアルを復元することは可能な場合もあります。

AWS CloudHSM キーストアで AWS KMS key (KMS キー) を作成すると、AWS KMS が、関連付けられた AWS CloudHSM クラスターにログインし、KMS キーのキーマテリアルを作成します。また、パスワードを、それのみが知っている値に変更し、AWS CloudHSM キーストアが接続されている間だけログイン状態を維持します。キー所有者、つまりキーを作成した CU のみがキーを削除できるため、誤って HSM からキーが削除されることはありません。

ただし、KMS キーのキーマテリアルがクラスター内の HSM から削除されると、KMS キーのキーステータスは最終的に `UNAVAILABLE` に変わります。暗号化オペレーションに KMS キーを使用しようとすると、`KMSInvalidStateException` の例外によりオペレーションは失敗します。最も重大なのは、KMS キーで暗号化されたデータが復号できないことです。

特定の状況では、キーマテリアルを含む バックアップからクラスターを作成することで、 削除されたキーマテリアルを回復できます。この方法は、キーが存在していて削除される前に少なくとも 1 つのバックアップが作成された場合にのみ機能します。

キーマテリアルを復旧するには、次の手順を実行します。

1. キーマテリアルが格納されているクラスターバックアップを見つけます。バックアップには、クラスターとその暗号化されたデータをサポートするために必要なすべてのユーザーとキーも含まれている必要があります。

   DescribeBackups オペレーションを使用して、クラスターのバックアップを一覧表示します。バックアップのタイムスタンプを使用すると、バックアップの選択に役立ちます。AWS CloudHSM キーストアに関連付けられているクラスターへの出力を制限するには、次の例のように `Filters` パラメータを使用します。

```
$ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
{
    "Backups": [
        {
            "ClusterId": "cluster-1a23b4cdefg",
            "BackupId": "backup-9g87f6edcba",
            "CreateTimestamp": 1536667238.328,
            "BackupState": "READY"
```

```
        },
            ...
    ]
}
```

2. 選択したバックアップからクラスタを作成します。バックアップに削除されたキーおよびクラスターに必要な他のユーザーとキーが含まれていることを確認します。
3. AWS CloudHSM キーストアを切断すればそのプロパティを編集できます。
4. AWS CloudHSM キーストアのクラスター ID を編集します。バックアップから作成したクラスターのクラスター ID を入力します。クラスターは元のクラスターとバックアップ履歴を共有するため、新しいクラスター ID が有効である必要があります。
5. AWS CloudHSM キーストアを再接続します。

kmsuser として ログインする方法

AWS CloudHSM キーストアの AWS CloudHSM クラスターでキーマテリアルを作成し管理するとき、AWS KMS は kmsuser Crypto User (CU) アカウントを使用します。クラスターに kmsuser CU アカウントを作成して、AWS CloudHSM カスタムキーストアを作成するときにそのパスワードを AWS KMS に入力します。

一般的に、AWS KMS は kmsuser アカウントを管理します。ただし、一部のタスクでは、AWS CloudHSM キーストアを切断し、kmsuser CU としてクラスターにログインして、cloudhsm_mgmt_util と key_mgmt_util コマンドラインツールを使用する必要があります。

> **Note**
>
> カスタムキーストアが切断されている間は、カスタムキーストアで KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする試みはすべて失敗します。このオペレーションにより、ユーザーが機密データを保存したりアクセスしたりすることを防ぐことができます。

このトピックでは、AWS CloudHSM キーストアを切断して kmsuser としてログインし、AWS CloudHSM コマンドラインツールを実行し、AWS CloudHSM キーストアからログアウトして再度接続する方法について説明します。

トピック

- 切断してログインする方法

- ログアウトして再接続する方法

切断してログインする方法

関連するクラスターに kmsuser CU としてログインする必要があるたびに、次の手順を実行します。

1. AWS CloudHSM キーストアがまだ切断されていなければ、切断します。AWS KMS コンソールまたは AWS KMS API を使用できます。

   AWS CloudHSM キーが接続されている間、AWS KMS は kmsuser としてログインします。これにより、kmsuser としてログインしたり、kmsuser パスワードを変更することができなくなります。

   例えば、このコマンドは DisconnectCustomKeyStoreを使用してサンプルキーストアを切断します。例にある AWS CloudHSM キーストア ID を、有効な ID に置き換えます。

   ```
   $ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

2. cloudhsm_mgmt_util を起動します。AWS CloudHSM ユーザーガイドの Prepare to run cloudhsm_mgmt_util セクションで説明されている手順に従います。
3. Crypto Officer (CO) として AWS CloudHSM クラスターの cloudhsm_mgmt_util にログインします。

   例えば、このコマンドは admin という名前の CO としてログインします。CO ユーザー名とパスワードの例を有効な値に置き換えます。

   ```
   aws-cloudhsm>loginHSM CO admin <password>
   loginHSM success on server 0(10.0.2.9)
   loginHSM success on server 1(10.0.3.11)
   loginHSM success on server 2(10.0.1.12)
   ```

4. changePswd コマンドを使用して、kmsuser アカウントのパスワードを自分が知っているパスワードに変更します (AWS KMS は、AWS CloudHSM キーストアに接続するときにパスワードをローテーションします)。パスワードは 7〜32 の英数字で構成する必要があります。大文字と小文字が区別され、特殊文字を含めることはできません。

   例えば、このコマンドは、kmsuser パスワードを tempPassword に変更します。

   ```
   aws-cloudhsm>changePswd CU kmsuser tempPassword
   ```

```
************************CAUTION*******************************
This is a CRITICAL operation, should be done on all nodes in the
cluster. Cav server does NOT synchronize these changes with the
nodes on which this operation is not executed or failed, please
ensure this operation is executed on all nodes in the cluster.
**************************************************************

Do you want to continue(y/n)?y
Changing password for kmsuser(CU) on 3 nodes
```

5. 設定したパスワードを使用して、key_mgmt_util または cloudhsm_mgmt_util に kmsuser としてログインします。詳細な手順については、「 cloudhsm_mgmt_util の使用開始」および「key_mgmt_utilの使用開始」を参照してください。使用するツールは、タスクによって異なります。

   例えば、このコマンドは key_mgmt_util にログインします。

```
Command: loginHSM -u CU -s kmsuser -p tempPassword
Cfm3LoginHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
Node id 2 and err state 0x00000000 : HSM Return: SUCCESS
```

ログアウトして再接続する方法

1. タスクを実行し、コマンドラインツールからログアウトします。ログアウトしないと、AWS CloudHSM キーストアへの再接続は失敗します。

```
Command:  logoutHSM
Cfm3LogoutHSM returned: 0x00 : HSM Return: SUCCESS

Cluster Error Status
Node id 0 and err state 0x00000000 : HSM Return: SUCCESS
Node id 1 and err state 0x00000000 : HSM Return: SUCCESS
```

2. カスタムキーストアの kmsuser パスワード設定を編集します。

これにより、クラスターの kmsuser の現在のパスワードが AWS KMS に通知されます。このステップを省略すると、AWS KMS は kmsuser としてクラスターにログインできなくなり、カスタムキーストアの再接続がすべて失敗します。AWS KMS コンソールまたは UpdateCustomKeyStoreオペレーションの KeyStorePasswordパラメータを使用できます。

例えば、このコマンドは現在のパスワードが AWS KMS あることを tempPassword に通知します。例のパスワードを実際のパスワードに置き換えます。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
key-store-password tempPassword
```

3. AWS KMS キーストアを AWS CloudHSM クラスターに再接続します。例にある AWS CloudHSM キーストア ID を、有効な ID に置き換えます。接続処理中に、AWS KMS は kmsuser パスワードを、それだけが知っている値に変更します。

   ConnectCustomKeyStore オペレーションは迅速に返されますが、接続プロセスには長時間かかる場合があります。最初のレスポンスは、接続プロセスの成功を示していません。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

4. DescribeCustomKeyStores オペレーションを使用して、 AWS CloudHSM キーストアが接続されていることを確認します。例にある AWS CloudHSM キーストア ID を、有効な ID に置き換えます。

   この例では、接続ステータスのフィールドには AWS CloudHSM キーストアが接続済みであると示されています。

```
$ aws kms describe-custom-key-stores --custom-key-store-
id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

外部キーストア

外部キーストアを使用すると、AWS リソースを AWS 以外の暗号化キーを使って保護することができます。この高度な機能は、ユーザーが管理する外部のキー管理システムに保存された暗号化キーで保護しなければならない、規制対象のワークロード向けに設計されています。外部キーストアは「AWS のデジタル統制に関するお客様との約束」をサポートしているため、AWS の外部でお客様が所有し管理しているキーマテリアルを使って暗号化する機能など、AWS 内のデータをお客様ご自身で制御できます。

外部キーストアとは、AWS の外部で所有し管理している外部キーマネージャーによってバックアップされるカスタムキーストアです。外部キーマネージャーは、物理的または仮想的なハードウェアセキュリティモジュール (HSM) である場合もあれば、暗号化キーの生成や使用を行うハードウェアベースまたはソフトウェアベースのシステムである場合もあります。外部キーストアの KMS キーを使用した暗号化と復号のオペレーションは、外部キーマネージャーがユーザーの暗号化キーマテリアルを使って実行します。この機能は Hold Your Own Key (HYOK) と呼ばれています。

AWS KMS は、ユーザーの外部キーマネージャーを直接操作することはなく、ユーザーのキーの作成、閲覧、管理、削除は行えません。代わりに、AWS KMS は、ユーザーが提供する外部キーストアプロキシ (XKS プロキシ) ソフトウェアを操作します。この外部キーストアプロキシは、AWS KMS と外部キーマネージャーとのあらゆるやり取りを仲介します。AWS KMS のリクエストを外部キーマネージャーに送信し、外部キーマネージャーのレスポンスを AWS KMS に送信します。また、外部キーストアプロキシは、AWS KMS の一般的なリクエストを、外部キーマネージャが理解できる、ベンダー固有の形式に変換して、ユーザーが外部キーストアをさまざまなベンダーのキーマネージャーで使用できるようにします。

外部キーストアの KMS キーは、AWS Encryption SDK など、クライアント側の暗号化に使用できます。外部キーストアはサーバー側の暗号化にとって重要なリソースであり、AWS 外部の暗号化キーを使用して、複数の AWS のサービス で AWS リソースを保護します。対称暗号化のカスタマーマネージドキーをサポートしている AWS のサービス は、外部キーストアの KMS キーもサポートします。サービスサポートの詳細については、「AWS サービス統合」を参照してください。

外部キーストアを使用すれば、暗号化キーを AWS の外部で保存して使用することが求められる規制対象のワークロードで、AWS KMS を使えるようになります。ただし、こうした使用は標準の責任共有モデルから大きく逸脱するものであり、オペレーション上の負担が増えることになります。可用性とレイテンシーに関するリスクが増えれば、そのリスクは、大半のお客様にとって外部キーストアのセキュリティにおける利点を上回るものとなります。

外部キーストアを使用すると、ユーザーは、信頼の基点を制御することができます。外部キーストアの KMS キーで暗号化されたデータは、ユーザーが管理する外部キーマネージャーでしか復号するこ

とはできません。外部キーストアを切断したり外部キーマネージャーと外部キーストアプロキシを切断したりするなどして外部キーマネージャーへのアクセスを一時的に無効にした場合、それが復元されるまで、AWS は暗号化キーにまったくアクセスできなくなります。その間、ユーザーの KMS キーで暗号化された暗号文は復号できません。外部キーマネージャーへのアクセスを永久に無効にすると、外部キーストアの KMS キーで暗号化された暗号文は、すべて回復不能になります。唯一の例外が、KMS キーで保護されたデータキーを一時的にキャッシュする AWS サービスです。これらのデータキーは、リソースを非アクティブにするまで、またはキャッシュの有効期限が切れるまで、機能し続けます。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

外部キーストアを使用すれば、暗号化キーをユーザーが管理し AWS からアクセスできないようにする必要のある規制対象のワークロードに関するいくつかのユースケースの問題を解消できます。しかし、こうした使用はクラウドベースのインフラストラクチャの運用方法を大きく変えるもので、責任共有モデルに著しい変化をもたらします。ほとんどのワークロードでオペレーションの負担が増えることになり、可用性とパフォーマンスのリスクが増え、そのリスクは、外部キーストアを利用するセキュリティ上の利点を上回ります。

詳細はこちら:

- AWS ニュースブログ「AWS KMS 外部キーストアの発表」。

自分は外部キーストアを使う必要がありますか?

FIPS 140-2 セキュリティレベル 3 検証済みハードウェアセキュリティモジュールで保護されているデフォルトの AWS KMS キーストアは、ほとんどのユーザーのセキュリティ、管理、規制に関する要件を満たします。　外部キーストアを使用するユーザーには、多額のコスト、メンテナンスやトラブルシューティングなどの負担、レイテンシー、可用性、信頼性のリスクが生じます。

外部キーストアの使用を検討する際は、代わりの手段についても理解しておきましょう。例えば、ユーザーが所有し管理する、AWS CloudHSM クラスターでバックアップされた AWS CloudHSM キーストアや、ユーザーが自身の HSM で生成し、必要に応じて KMS キーから削除できる、インポートされたキーマテリアルを使う KMS キーなどです。特に、有効期限が非常に短いキーマテリアルをインポートすると、パフォーマンスや可用性のリスクなしに、同様のレベルの制御が可能になる場合があります。

以下の要件がある場合は、外部キーストアが組織にとって適切なソリューションとなる可能性があります。

- 暗号化キーを、オンプレミスのキーマネージャー、またはユーザーが管理する AWS 以外のキーマネージャーで使用する必要がある場合。
- 暗号化キーが、クラウド以外の場所で、ユーザーが単独で管理して保持されていることを証明する必要がある場合。
- 暗号化と復号で、独立した認証を受けた暗号化キーを使用する必要がある場合。
- キーマテリアルは、補助的な、独立した監査パスの対象とする必要があります。

外部キーストアを選択する場合は、その使用を、AWS 以外の暗号化キーによる保護を必要とするワークロードに限定します。

責任共有モデル

標準の KMS キーは、AWS KMS が所有し管理する HSM で生成され使用される、キーマテリアルを使用します。ユーザーは、KMS キーのアクセス制御ポリシーを作成し、KMS キーを使用してリソースを保護するように AWS のサービス を設定します。AWS KMS は、KMS キーに含まれるキーマテリアルのセキュリティ、可用性、レイテンシー、耐久性に対して責任を負います。

外部キーストアの KMS キーは、外部キーマネージャーにおけるキーマテリアルとオペレーションに依存します。そのため、責任のバランスはユーザー側にシフトします。外部キーマネージャーにある暗号化キーのセキュリティ、信頼性、耐久性、パフォーマンスに対する責任はユーザーが負います。リクエストへの迅速な対応、外部キーストアプロキシとのやり取り、セキュリティ基準の維持に対する責任は、AWS KMS が負います。外部キーストアの暗号文の安全性を、標準の AWS KMS 暗号文と同等以上に維持するため、AWS KMS は、まずすべてのプレーンテキストを、ユーザーの KMS キーに固有の AWS KMS キーマテリアルで暗号化し、これを外部キーマネージャーに送信して外部キーを使って暗号化します。この方法は二重暗号化と呼ばれています。それにより、AWS KMS も外部キーマテリアルの所有者も、二重に暗号化された暗号文を単独で復号することはできなくなります。

ユーザーは、規制およびパフォーマンス基準を満たす外部キーマネージャーの管理、AWS KMS 外部キーストアプロキシ API 仕様に準拠する外部キーストアプロキシの提供と維持、およびキーマテリアルの可用性と耐久性の確保に責任を負います。また、は外部キーストアの作成、設定、維持もユーザーが行います。ユーザーが管理しているコンポーネントが原因でエラーが生じた場合、ユーザーは、AWS サービスが過度の中断なくユーザーのリソースにアクセスできるよう、エラーを特定して解決するための準備を整えておかなくてはなりません。AWS KMS には、問題の原因を特定して、可能な解決策を見きわめるのに役立つトラブルシューティングのガイドが用意されています。

が外部キーストアAWS KMSに記録する Amazon CloudWatch メトリクスとディメンションを確認します。 では、パフォーマンスや運用上の問題の初期兆候を事前に検出できるように、外部キーストアをモニタリングする CloudWatch アラームを作成することをAWS KMS強くお勧めします。

違いは何か

外部キーストアは、対称暗号化 KMS キーのみをサポートしています。AWS KMS では、ユーザーは、アクセス制御ポリシーの設定やキー使用のモニタリングなど、他のカスタマーマネージドキーを管理する方法とほぼ同じやり方で、外部キーストアの KMS キーを使用し管理します。ユーザーが KMS キーに使用している、外部キーストアの KMS キーを使った暗号化オペレーションをリクエストするときは、同じ API を同じパラメータで使用します。料金も、標準の KMS キーと同じです。詳細については、「外部キーストアで KMS キーを管理する」、「外部キーストアで KMS キーを使用する」、「AWS Key Management Service の料金」をそれぞれ参照してください。

ただし、外部キーストアには以下のような原則の変更があります。

- キーオペレーションの可用性、耐久性、レイテンシーの責任はユーザーが負う。
- 外部キーマネージャーシステムの開発、購入、オペレーション、ライセンス供与のコストは、ユーザーが負担する。
- ユーザーは、AWS KMS から外部キーストアプロキシへのすべてのリクエストに、独立した認証を実装できる。
- ユーザーは、外部キーストアプロキシのすべてのオペレーション、AWS KMS リクエストに関連する外部キーマネージャーのすべてのオペレーションをモニタリング、監査、記録できる。

開始方法

外部キーストアを作成して管理するには、外部キーストアプロキシ接続オプションを選択し、前提条件を構成し、外部キーストアを作成、設定する必要があります。使用を開始するには「外部キーストアの計画」を参照してください。

クォータ

AWS KMS では、接続状態にかかわらず、AWS CloudHSM キーストアと外部キーストアの両方を含め、各 AWS アカウント とリージョンで最大 10 個のカスタムキーストアを使用できます。また、外部キーストアで KMS キーを使用するときは、AWS KMS リクエストクォータが適用されます。

外部キーストアプロキシに VPC プロキシ接続を選択すると、VPC、サブネット、Network Load Balancer などの必要なコンポーネントにもクオータが適用される場合があります。これらのクォータの詳細については、Service Quotas コンソールを参照してください。

リージョン

ネットワークのレイテンシーを最小限に抑えるには、外部キーマネージャーに最も近い AWS リージョン に外部キーストアコンポーネントを作成します。可能な場合は、ネットワークラウンドトリップ時間 (RTT) が 35 ミリ秒以下のリージョンを選択します。

外部キーストアは、中国 (北京) と中国 (寧夏) を除き、AWS KMS がサポートされるすべての AWS リージョンでサポートされています。

サポートされていない機能

AWS KMS はカスタムキーストアで次の機能をサポートしていません。

- 非対称 KMS キー
- 非対称データキーペア
- HMAC KMS キー
- インポートされたキーマテリアルを持つ KMS キー
- 自動キーローテーション
- マルチリージョンキー

トピック

外部キーストアのコンセプト

このトピックでは、外部のカスタムキーストアで使用するいくつかの概念について説明します。

トピック

- 外部キーストア
- 外部キーマネージャー
- 外部キー
- 外部キーストアプロキシ
- 外部キーストアプロキシ接続
- 外部キーストアのプロキシ認証の認証情報
- プロキシ API
- 二重暗号化

外部キーストア

外部キーストアは、AWS の外部でユーザーが所有し管理している外部キーマネージャーによってバックアップされた AWS KMS カスタムキーストアです。外部キーストアの各 KMS キーは、外部キーマネージャーの外部キーに関連付けられています。外部キーストアの KMS キーを使って暗号化または複合化を行う場合、オペレーションは、外部キーマネージャーでユーザーの外部キーを使用して実行されます。この方法は Hold your Own Keys (HYOK) と呼ばれます。この機能は、暗号化キーを自社の外部キーマネージャーで管理する必要のある組織向けに設計されています。

外部キーストアを使用することで、ユーザーの AWS リソースを保護している暗号化キーとオペレーションを、ユーザーが管理する外部キーマネージャーで保持できます。AWS KMS は、外部キーマネージャーにデータの暗号化と復号のリクエストを送信しますが、外部キーを作成、削除、管理することはできません。AWS KMS から外部キーマネージャーへのリクエストはすべて、ユーザーが提供、所有、管理する外部キーストアプロキシのソフトウェアコンポーネントが仲介します。

AWS KMS カスタマーマネージドキーをサポートしている AWS サービスは、外部キーストアの KMS キーを使用してデータを保護することができます。それによりデータは、最終的にユーザーのキーによって、外部キーマネージャーで暗号化オペレーションを使用して保護されます。

外部キーストアの KMS キーは、信頼モデル、責任共有の取り決め、期待されるパフォーマンスが標準の KMS キーとは根本的に異なります。外部キーストアを使用した場合、ユーザーは、キーマテリアルのセキュリティと整合性、および暗号化のオペレーションに責任を負います。外部キーストアにある KMS キーの可用性とレイテンシーは、ハードウェア、ソフトウェア、ネットワークコンポーネント、そして AWS KMS と外部キーマネージャーとの距離の影響を受けます。また、外部キーマネージャーのコストと、外部キーマネージャーが AWS KMS とやり取りする際に必要となるネットワークおよびロードバランサーインフラストラクチャのコストが、追加で発生する可能性もあります。

外部キーストアは、より広範なデータ保護戦略の一環として使用することができます。ユーザーが保護する AWS リソースについては、外部キーストアの KMS キーを必要とするリソースや標準の KMS キーで保護できるリソースを、ユーザーがそれぞれ判断できます。これにより、KMS キーを、特定のデータ分類、アプリケーション、プロジェクト向けに柔軟に選択できます。

外部キーマネージャー

外部キーマネージャーは、256 ビット AES 対称キーを生成し、対称暗号化と復号を実行できる AWS の外部コンポーネントです。外部キーストアの外部キーマネージャーには、物理ハードウェアセキュリティモジュール (HSM)、仮想 HSM、HSM コンポーネントの有無にかかわらずソフトウェアキーマネージャー、のいずれかを使用できます。これらは、ローカルまたはリモートのデータセンター、もしくは任意のクラウドなど、AWS の外部のどこにでも配置できます。外部キーストアは、単一の外部キーマネージャー、または暗号化キーを共有している、複数の関連するキーマネージャーインスタンス (HSM クラスターなど) によってバックアップできます。外部キーストアは、さまざまなベンダーの、さまざまな外部マネージャーをサポートするように設計されています。外部キーマネージャーの要件の詳細については、「外部キーストアの計画」を参照してください。

外部キー

外部キーストアの各 KMS キーは、外部キーとも呼ばれる、外部キーマネージャーにある暗号化キーに関連付けられています。外部キーストアの KMS キーを使って暗号化または複合化を行うとき、暗号化オペレーションは、外部キーマネージャーによって、ユーザーの外部キーを使用して実行されます。

> **Warning**
>
> 外部キーは、KMS キーのオペレーションに不可欠なものです。外部キーを紛失したり削除したりすると、関連付けられた KMS キーで暗号化された暗号文は回復不能になります。

外部キーストアの場合、外部キーは、有効化され、暗号化と復号を実行できる、256 ビット AES キーでなければなりません。外部キー要件の詳細については、「外部キーストアの KMS キーの要件」を参照してください。

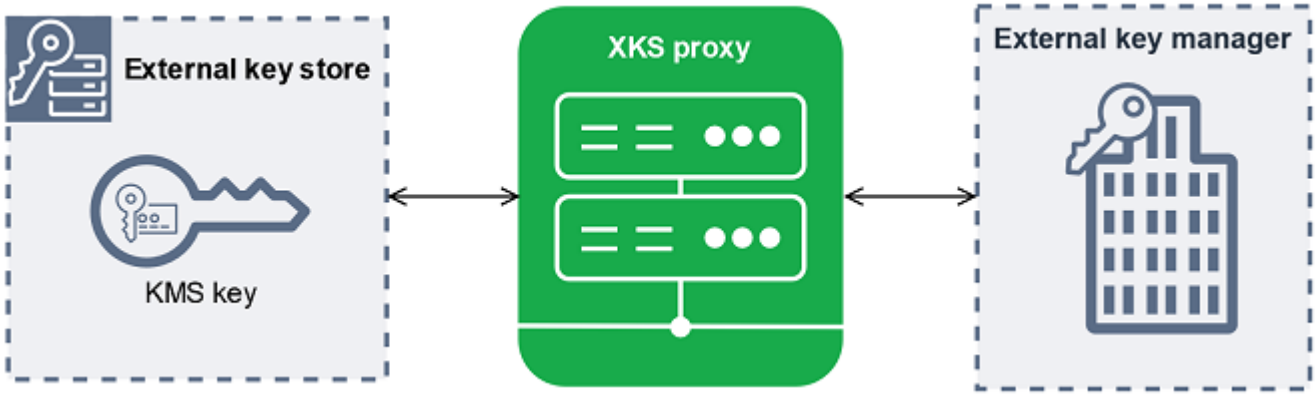
AWS KMS は、外部キーを作成、削除、管理することはできません。ユーザーの暗号化キーマテリアルは、外部キーマネージャーの外に出ることはありません。外部キーストアで KMS キーを作成するときは、ユーザーが外部キーの ID (`XksKeyId`) を指定します。KMS キーに関連付けられた外部キー ID は変更できません。ただし、外部キーマネージャは、外部キー ID に関連付けられたキーマテリアルをローテーションできます。

外部キーのほかに、外部キーストアの KMS キーには AWS KMS キーマテリアルもあります。KMS キーで保護されたデータは、最初に AWS KMS キーマテリアルを使用して AWS KMS によって暗号化され、次に、外部キーを使用して外部キーマネージャーによって暗号化されます。この二重暗号化プロセスにより、KMS キーで保護された暗号文は、AWS KMS のみで保護された暗号文と同等以上の安全性を維持できます。

暗号化キーの多くは、異なる種類の識別子を持ちます。外部キーストアに KMS キーを作成するときは、外部キーストアプロキシが外部キーを参照するときに使用する、外部キーの ID を指定します。誤った識別子を使用すると、外部キーで KMS キーを作成するときに失敗します。

外部キーストアプロキシ

外部キーストアプロキシ (「XKS プロキシ」) は顧客が所有および管理するソフトウェアアプリケーションで、AWS KMS と外部キーマネージャーとのすべてのやり取りを仲介します。また、一般的な AWS KMS リクエストを、ベンダー固有の外部キーマネージャーが理解可能な形式に変換します。外部キーストアには、外部キーストアプロキシが必要です。各外部キーストアは、1 つの外部キーストアプロキシに関連付けられています。



AWS KMS は、外部キーを作成、削除、管理することはできません。ユーザーの暗号化キーマテリアルは、ユーザーの外部キーマネージャーの外に出ることはありません。AWS KMS と外部キーマネージャーのやり取りはすべて、外部キーストアプロキシによって仲介されます。AWS KMS は、外部キーストアプロキシにリクエストを送信し、外部キーストアプロキシからレスポンスを受けとります。AWS KMS から外部キーマネージャーにリクエストを送信し、外部キーマネージャーのレスポンスを AWS KMS に送り返す責任は、外部キーストアプロキシが負います。

外部キーストアの外部キーストアプロキシを所有および管理して、そのメンテナンスとオペレーションを行う責任は、ユーザーが負います。ユーザーは、AWS KMS が公開しているオープンソースの外部キーストアプロキシ API 仕様に基づいて外部キーストアプロキシを開発するか、ベンダーからプロキシアプリケーションを購入することができます。外部キーストアプロキシは、外部キーマネー

ジャーに含まれている場合があります。プロキシ開発をサポートするために、 は、外部キーストアプロキシのサンプル (aws-kms-xks-proxy) と、外部キーストアプロキシが仕様に準拠していることを検証するテストクライアント (xks-kms-xksproxy-testクライアント) AWS KMSも提供します。

AWS KMS への認証のため、プロキシはサーバー側の TLS 証明書を使用します。ユーザーのプロキシを認証するため、AWS KMS は、SigV4 のプロキシ認証の認証情報を使って、外部キーストアプロキシへのすべてのリクエストに署名します。任意で、プロキシで相互TLS (mTLS) を有効にすれば、さらに確実に AWS KMS のリクエストのみを受け入れるようにすることが可能です。

外部キーストアプロキシは、以下の暗号スイートのうち少なくとも 1 つを含む HTTP/1.1 以降と TLS 1.2 以降をサポートしている必要があります。

- TLS_AES_256_GCM_SHA384 (TLS 1.3)
- TLS_CHACHA20_POLY1305_SHA256 (TLS 1.3)

> **Note**
>
> AWS GovCloud (US) Regionでは TLS_CHACHA20_POLY1305_SHA256 はサポートされていません。

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (TLS 1.2)

外部キーストアの KMS キーを作成し使用するときは、最初に外部キーストアを外部キーストアプロキシに接続しておく必要があります。外部キーストアは、必要に応じてプロキシから切断することも可能です。切断すると、外部キーストアのすべての KMS キーが使用できなくなり、いずれの暗号化オペレーションでもそれらを使用できなくなります。

外部キーストアプロキシ接続

外部キーストアプロキシ接続 (「XKS プロキシ接続」) とは、AWS KMS が外部キーストアプロキシとの通信に使用する方法のことをいいます。

外部キーストアを作成するときにプロキシ接続のオプションを指定すると、そのプロキシ接続は外部キーストアのプロパティになります。プロキシ接続オプションは、カスタムキーストアプロパティを更新することで変更できますが、外部キーストアプロキシが引き続き同じ外部キーにアクセスできることを、確認しておく必要があります。

AWS KMS は、次の接続オプションをサポートしています。

- パブリックエンドポイント接続 — AWS KMS は、外部キーストアプロキシへのリクエストを、インターネットを介して、ユーザーが管理しているパブリックエンドポイントに送信します。このオプションは簡単に作成し管理することができますが、すべてのインストールのセキュリティ要件を満たしているとは限りません。
- VPC エンドポイントサービス接続 - AWS KMS は、ユーザーが作成し管理している Amazon Virtual Private Cloud (Amazon VPC) エンドポイントにリクエストを送信します。外部キーストアプロキシを Amazon VPC 内でホストするか、AWS の外部でホストして、Amazon VPC を通信にのみ使用することができます。

外部キーストアプロキシ接続のオプションの詳細については、「プロキシ接続オプションの選択」を参照してください。

外部キーストアのプロキシ認証の認証情報

外部キーストアのプロキシを認証するために、AWS KMS は、Signature V4 (SigV4) 認証の認証情報を使って外部キーストアプロキシへのすべてのリクエストに署名します。プロキシでこの認証の認証情報を確立した後、外部ストアを作成する際に、この認証情報を AWS KMS に提供します。

> **Note**
>
> AWS KMS が XKS プロキシへのリクエストに署名する際に使用する SigV4 認証情報は、AWS アカウント の AWS Identity and Access Management プリンシパルに関連付けられた SigV4 認証情報とは無関係です。IAM SigV4 認証情報を外部キーストアプロキシに再利用しないでください。

各プロキシ認証の認証情報は、2 つの要素から成ります。外部キーストアを作成するとき、または外部キーストア用にプロキシ認証の認証情報を更新するときは、これら両方の要素を指定する必要があります。

- アクセスキー ID: シークレットアクセスキーを識別します。この ID はプレーンテキストで入力できます。
- シークレットアクセスキー: 認証情報のシークレットの部分です。AWS KMS は、認証情報のシークレットアクセスキーを暗号化し、その後保存します。

認証情報の設定は、間違った値を入力したとき、プロキシの認証情報を変更するとき、プロキシが認証情報をローテーションするときなどにいつでも編集できます。外部キーストアプロキシの AWS

KMS 認証に関する技術的な詳細については、「AWS KMS 外部キーストアプロキシ API 仕様」の「認証」を参照してください。

外部キーストアの KMS キーを使用している AWS のサービス を中断せずに、認証情報をローテーションできるようにするために、外部キーストアプロキシで、AWS KMS に有効なプロキシ認証の認証情報を 2 つ以上サポートしておくことが推奨されています。これにより、新しい認証情報を AWS KMS に提供している間も、以前の認証情報が引き続き機能します。

プロキシ認証情報の経過時間を追跡しやすくするために、 は Amazon CloudWatch メトリクス AWS KMSを定義しますXksProxyCredentialAge。このメトリクスを使用して、認証情報の有効期間が設定したしきい値に達したときに通知する CloudWatch アラームを作成できます。

外部キーストアプロキシが AWS KMS のみに応答することをさらに確実にするため、一部の外部キープロキシは Mutual Transport Layer Security (mTLS) をサポートしています。詳細については、「mTLS 認証 (オプション)」を参照してください。

プロキシ API

AWS KMS 外部キーストアをサポートするには、「AWS KMS 外部キーストアプロキシ API 仕様」に記載のとおり、外部キーストアプロキシに必要なプロキシ API が実装されている必要があります。これらのプロキシ API リクエストは、AWS KMS がプロキシに送信する唯一のリクエストです。ユーザーはこれらのリクエストを直接送信することはありませんが、これらについて知っておくと、外部キーストアやそのプロキシで発生する問題を修正する際に役に立つ場合があります。例えば、 は、これらの API コールのレイテンシーと成功率に関する情報を外部キーストアの Amazon CloudWatch メトリクスにAWS KMS含めます。詳細については、「外部キーストアのモニタリング」を参照してください。

以下の表には各プロキシ API が一覧表示され、それぞれが説明されてます。また、プロキシ API への呼び出しをトリガーする AWS KMS オペレーションや、プロキシ API 関連の AWS KMS オペレーションの例外も記載されています。

プロキシ API	説明	関連の AWS KMS オペレーション
Decrypt	AWS KMS は、復号する暗号文と、使用する外部キーの ID を送信します。必要な暗号化アルゴリズムは AES_GCM です。	復号、 ReEncrypt

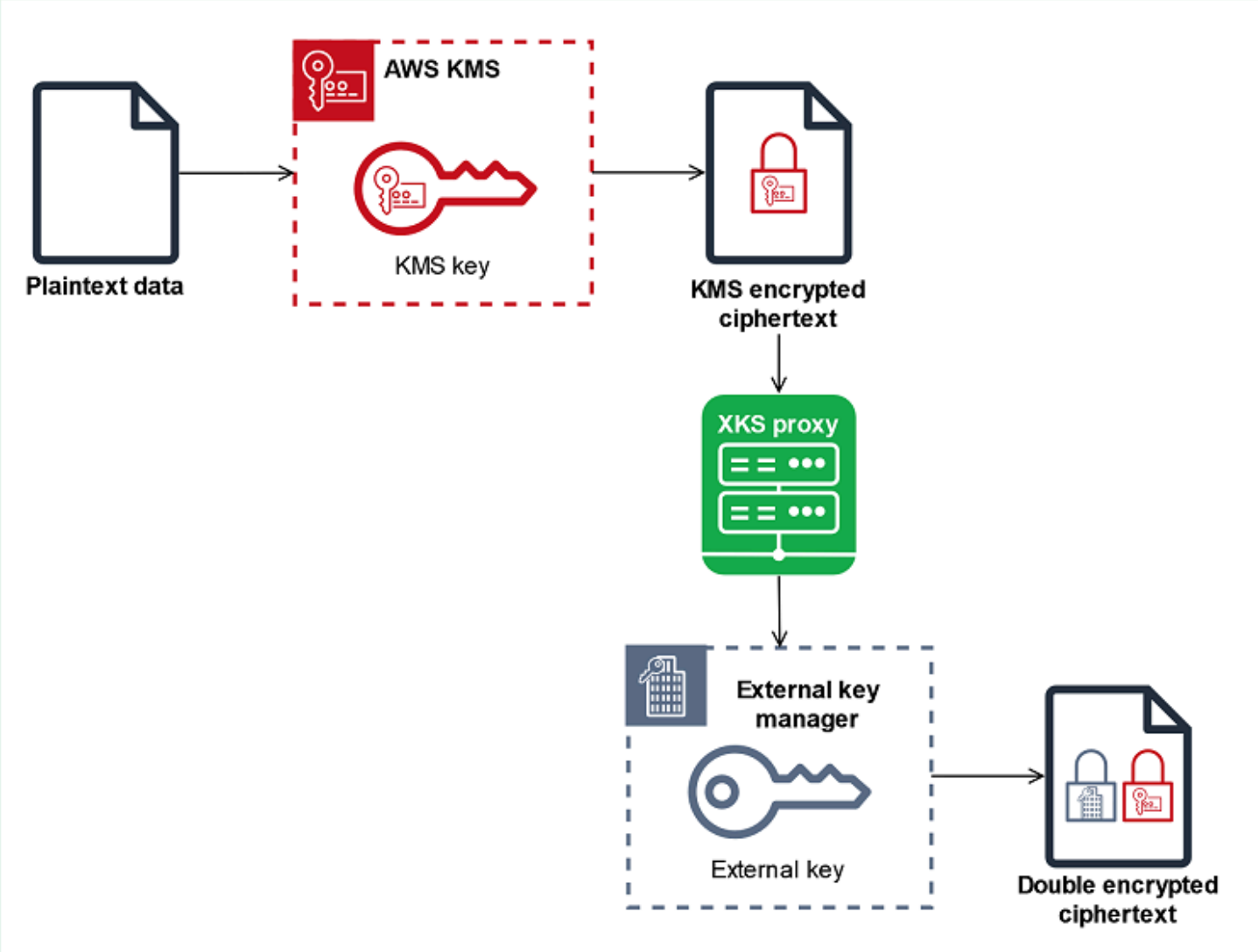
プロキシ API	説明	関連の AWS KMS オペレーション
暗号化	AWS KMS は、暗号化するデータと、使用する外部キーの ID を送信します。必要な暗号化アルゴリズムは AES_GCM です。	、、GenerateDataKey、 を暗号化GenerateDataKeyWithoutPlaintextする ReEncrypt
GetHealthStatus	AWS KMS は、プロキシと外部キーマネージャーのステータスに関する情報をリクエストします。 各外部キーマネージャーのステータスは、以下のいずれかになります。 • Active: 正常。トラフィックを処理できる • Degraded: 異常。ただし、トラフィックを処理できる • Unavailable : 異常。トラフィックを処理できない	CreateCustomKeyStore (パブリックエンドポイント接続 の場合)、ConnectCustomKeyStore (VPC エンドポイントサービス接続 の場合) すべての外部キーマネージャーインスタンスが Unavailable である場合、キーストアを作成または接続しようとすると XksProxyUriUnreachableException で失敗します。
GetKeyMetadata	AWS KMS は、外部キーストアの KMS キーに関連付けられた外部キーに関する情報をリクエストします。 応答には、キーのスペック (AES_256)、キーの使用 ([ENCRYPT, DECRYPT])、外部キーが ENABLED または DISABLED かどうか、が含まれます。	CreateKey キーのスペックが AES_256 でない場合、キーの使用が [ENCRYPT, DECRYPT] でない場合、ステータスが DISABLED である場合、CreateKey オペレーションは XksKeyInvalidConfigurationException により失敗します。

二重暗号化

外部キーストアの KMS キーで暗号化されるデータは、2 回暗号化されます。まず、AWS KMS が KMS キーに固有の AWS KMS キーマテリアルを使ってデータを暗号化します。次に、AWS KMS で

暗号化された暗号文が、外部キーマネージャーによって外部キーを使用して暗号化されます。このプロセスは二重暗号化と呼ばれます。

二重暗号化を使用すれば、外部キーストアの KMS キーで暗号化されたデータを、標準の KMS キーで暗号化された暗号文と同等以上の安全性で維持できます。また、AWS KMS から外部キーストアプロキシに送信されているプレーンテキストを保護することもできます。二重暗号化を使うことで、暗号文を完全に制御できます。外部キーへの AWS アクセスを外部プロキシ経由で永久に無効にすると、AWS に残っている暗号文は暗号によって実質的に細断されます。



二重暗号化を有効にするには、外部キーストア内の各 KMS キーに 2 つの暗号化バッキングキーを作成します。

- KMS キーに固有の AWS KMS キーマテリアル。このキーマテリアルは AWS KMS FIPS 140-2 セキュリティレベル 3 認定ハードウェアセキュリティモジュール (HSM) で生成され、この HSM でのみ使用されます。
- 外部キーマネージャーの外部キー。

二重暗号化には次の効果があります。

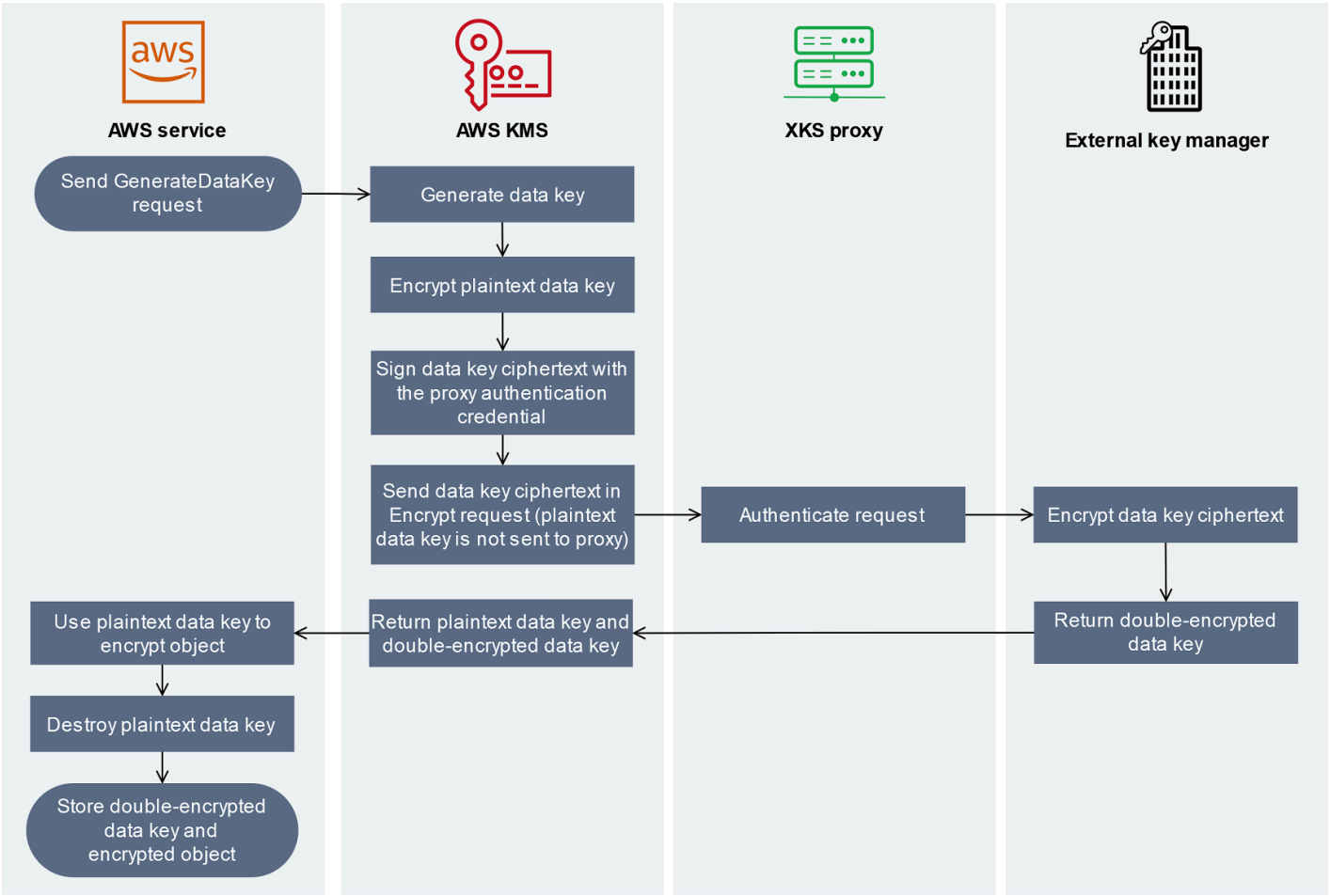
- AWS KMS は、外部キーストアプロキシ経由で外部キーにアクセスしない限り、外部キーストアの KMS キーで暗号化された暗号文を復号できない。
- AWS 以外のキーストアにある KMS キーで暗号化された暗号文は、その外部キーマテリアルがある場合でも、復号できない。
- 外部キーストアから削除された KMS キーは、その外部キーマテリアルがある場合でも、再作成できない。各 KMS キーには、対称暗号文に含まれる固有のメタデータがある。新しい KMS キーでは、同じ外部キーマテリアルを使用する場合でも、元のキーで暗号化された暗号文を復号することはできない。

実際の二重暗号化の例については、「外部キーストアの仕組み」を参照してください。

外部キーストアの仕組み

外部キーストア、外部キーストアプロキシ、外部キーマネージャーは、連携して AWS リソースを保護します。次の手順は、KMS キーで保護された一意のデータキーに基づいて各オブジェクトを暗号化する、一般的な AWS のサービス の暗号化ワークフローを示したものです。この場合、オブジェクトを保護するために、外部キーストアの KMS キーが選択されています。この例では、AWS KMS が 二重暗号化をどのように使用し、送信中のデータキーを保護するのか、また、外部キーストアの KMS キーによって生成された暗号文が、AWS KMS のキーマテリアルを使用し標準の対称 KMS キーで暗号化された暗号文と同等以上の安全性を、どう確保しているのかを説明します。

AWS KMS と統合する実際の AWS のサービス が使用する暗号化の方法は、それぞれ異なります。詳細については、AWS のサービス ドキュメントの「セキュリティ」の項にある「データ保護」のトピックを参照してください。



1. AWS のサービス リソースに新しいオブジェクトを追加します。オブジェクトを暗号化するために、 は外部キーストアの KMS キーAWS KMSを使用して に GenerateDataKey リクエストAWS のサービスを送信します。
2. AWS KMS が、256 ビットの対称データキーを生成し、プレーンテキストデータキーのコピーを外部キーストアプロキシを介して外部キーマネージャーに送信する準備をします。AWS KMS が、外部キーストアの KMS キーに関連付けられた AWS KMS キーマテリアルを使ってプレーンテキストのデータキーを暗号化すると、二重暗号化のプロセスが開始します。
3. AWS KMS が、外部キーストアに関連付けられた外部キーストアプロキシに暗号化リクエストを送信します。このリクエストには、暗号化するデータキーの暗号文と、KMS キーに関連付けられた外部キーの ID が含まれています。AWS KMS は、外部キーストアプロキシのプロキシ認証の認証情報を使ってこのリクエストに署名します。

   データキーのプレーンテキストのコピーは、外部キーストアプロキシには送信されません。
4. 外部キーストアプロキシがリクエストを認証し、暗号化リクエストを外部のキーマネージャーに渡します。

外部キーストアプロキシの中には、特定の条件下で、選択されたプリンシパルのみが操作されるように、オプションの認証ポリシーを実装しているものもあります。

5. 外部キーマネージャーは、指定された外部キーを使用して、データキーの暗号文を暗号化します。外部キーマネージャは二重に暗号化されたデータキーを外部キーストアプロキシに返し、外部キーストアプロキシはそれを AWS KMS に返します。
6. AWS KMS は、データキーのプレーンテキストデータキーと、そのデータキーの二重暗号化コピーを AWS のサービス に返します。
7. AWS のサービス は、プレーンテキストデータキーを使ってリソースオブジェクトを暗号化し、プレーンテキストデータキーを破棄し、暗号化されたデータキーを暗号化されたオブジェクトと共に保存します。

   AWS のサービス の中には、プレーンテキストのデータキーをキャッシュして複数のオブジェクトに使用したり、リソースの使用中にこれを再利用したりするものもあります。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

暗号化されたオブジェクトを復号するには、AWS のサービス は、暗号化されたデータキーを Decrypt リクエストで AWS KMS に送り返す必要があります。暗号化されたデータキーを復号するには、AWS KMS は、暗号化されたデータキーを、外部キーの ID とともに外部キーストアプロキシに送り返す必要があります。何らかの理由で、外部キーストアプロキシへの復号リクエストに失敗した場合、AWS KMS は、暗号化されたデータキーを復号できず、AWS のサービス は、暗号化されたオブジェクトを復号できません。

外部キーストアへのアクセスの制御

標準の KMS キーで使用するすべての AWS KMS アクセス制御機能 (キーポリシー、IAM ポリシー、グラント) は、外部キーストアの KMS キーでも同様に機能します。IAM ポリシーを使うことで、外部キーストアを作成し管理するための API オペレーションへのアクセスを、制御できます。外部キーストアの AWS KMS keys へのアクセスを制御するには、IAM ポリシーとキーポリシーを使用します。また、外部キーストアの KMS キーへのアクセスを制御するには、AWS 組織のサービス制御ポリシーと VPC エンドポイントポリシーを使用することもできます。

ユーザーとロールには、それらが実行する可能性の高いタスクに必要なアクセス許可のみ、付与することが推奨されます。

トピック

- 外部キーストアマネージャーの承認

- 外部キーストアにおける KMS キーのユーザー認証
- AWS KMS と外部キーストアプロキシとの通信の承認
- 外部キーストアプロキシ認証 (オプション)
- mTLS 認証 (オプション)

外部キーストアマネージャーの承認

外部キーストアを作成し管理するプリンシパルには、カスタムキーストアオペレーションへのアクセス許可が必要になります。次のリストは、 外部キーストアマネージャーに必要な最小限のアクセス許可です。カスタムキーストアは AWS リソースではないため、ユーザーは、他の AWS アカウントキーストアのプリンシパルに、外部キーストアへのアクセス許可を付与することはできません。

- `kms:CreateCustomKeyStore`
- `kms:DescribeCustomKeyStores`
- `kms:ConnectCustomKeyStore`
- `kms:DisconnectCustomKeyStore`
- `kms:UpdateCustomKeyStore`
- `kms:DeleteCustomKeyStore`

外部キーストアを作成するプリンシパルは、外部キーストアコンポーネントを作成し構成するためのアクセス許可が必要になります。プリンシパルは、外部キーストアを自分のアカウントのみで作成できます。VPC エンドポイントサービスに接続できる外部キーストアを作成するには、プリンシパルは、次のコンポーネントを作成するためのアクセス許可を持っている必要があります。

- An Amazon VPC
- パブリックサブネットおよびプライベートサブネット
- Network Load Balancer とターゲットグループ
- Amazon VPC エンドポイントサービス

詳細については、「Amazon VPC の Identity and Access Management」、「VPC エンドポイントおよび VPC エンドポイントサービスの Identity and Access Management」、「Elastic Load Balancing API のアクセス許可」を参照してください。

外部キーストアにおける KMS キーのユーザー認証

外部キーストアで AWS KMS keys を作成し管理するプリンシパルは、AWS KMS で KMS キーを作成し管理するプリンシパルと同じアクセス許可が必要になります。外部キーストアの KMS キーの、デフォルトのキーポリシーは、AWS KMS の KMS キーの、デフォルトのキーポリシーと同一です。タグとエイリアスを使用して KMS キーへのアクセスを制御する属性ベースのアクセス制御 (ABAC) は、カスタムキーストアの KMS キーでも同様に有効です。

カスタムキーストアで暗号化オペレーションに KMS キーを使用するプリンシパルには、KMS キーで暗号化オペレーション (KMS: Decryptなど) を実行するアクセス許可が必要です。　これらのアクセス権限は、IAM またはキーポリシーで指定できます。ただし、カスタムキーストアで KMS キーを使用するための追加のアクセス許可は必要ありません。

外部キーストアの KMS キーにのみ適用されるアクセス許可を設定するには、値が EXTERNAL_KEY_STORE の `kms:KeyOrigin` ポリシー条件を使用します。この条件を使用して、kms:CreateKey アクセス許可、または KMS キーリソースに固有のアクセス許可を制限できます。例えば、次の IAM ポリシーを使えば、アタッチされた ID は、KMS が外部キーストアにある限り、アカウントのすべての KMS キーで、指定されたオペレーションを呼び出すことができます。外部キーストアの KMS キーと AWS アカウント の KMS キーへのアクセス許可は制限できますが、アカウントの特定の外部キーストアへのアクセス許可は制限できため、注意が必要です。

```
{
  "Sid": "AllowKeysInExternalKeyStores",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL_KEY_STORE"
    }
  }
}
```

AWS KMS と外部キーストアプロキシとの通信の承認

AWS KMS は、外部キーマネージャーとの通信を、提供された外部キーストアプロキシを介してのみ行います。AWS KMS は、指定された外部キーストアプロキシ認証の認証情報を持つ Signature Version 4 (SigV4) プロセスを使用してリクエストに署名することで、プロキシを承認します。外部キーストアプロキシにパブリックエンドポイント接続を使用している場合、AWS KMS に追加のアクセス許可は必要ありません。

ただし、VPC エンドポイントサービス接続を使用している場合は、AWS KMS に、インターフェイスエンドポイントを作成するためのアクセス許可を付与する必要があります。このアクセス許可は、外部キーストアプロキシが VPC にあるか、または外部キーストアプロキシが他の場所にあるかに関係なく必要となりますが、AWS KMS との通信には VPC エンドポイントサービスを使用します。

AWS KMS がインターフェイスエンドポイントを作成できるようにするには、Amazon VPC コンソールまたは ModifyVpcEndpointServicePermissionsオペレーションを使用します。次のプリンシパルにアクセス許可を付与します:`cks.kms.<region>.amazonaws.com`。

例えば、次の AWS CLI コマンドは、AWS KMS が米国西部 (オレゴン) (us-west-2) リージョンにある指定された VPC エンドポイントサービスに接続することを許可します。このコマンドを使用するときは、先に、Amazon VPC サービス ID と AWS リージョン を、設定上の有効な値に置き換えます。

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

このアクセス許可を削除するには、Amazon VPC コンソールまたは を `RemoveAllowedPrincipals`パラメータModifyVpcEndpointServicePermissionsとともに使用します。

外部キーストアプロキシ認証 (オプション)

外部キーストアプロキシの中には、外部キーを使用するための認証要件を実装しているものがあります。外部キーストアプロキシは、特定のユーザーが特定の条件下でのみ特定のオペレーションをリクエストすることを許可する、認証スキームの設計と実装が許可されていますが、必須ではありません。例えばプロキシは、ユーザー A に、特定の外部キーを使用した暗号化は許可するが、それを使用した復号は許可しないように設計されていることがあります。

プロキシ認証は、AWS KMS がすべての外部キーストアプロキシに必要とする、SigV4 ベースのプロキシ認証からは独立しています。また、外部キーストアやその KMS キーに影響するオペレーションへのアクセスを認証する、キーポリシー、IAM ポリシー、グラントとも無関係です。

外部キーストアプロキシによる認証を有効にするため、AWS KMS には、各プロキシ API リクエストに、呼び出し元、KMS キー、AWS KMS オペレーション、AWS のサービス (存在する場合) などのメタデータが含まれています。外部キープロキシ API のバージョン 1 (v1) のリクエストメタデータは、次のとおりです。

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

例えば、特定のプリンシパル (awsPrincipalArn) からのリクエストを、そのリクエストがプリンシパルの代わりに特定の AWS のサービス (kmsViaService) によって実行された場合のみ許可するよう、プロキシを設定することができます。

プロキシ認証に失敗すると、関連する AWS KMS オペレーションは失敗し、エラーの内容を説明するメッセージが表示されます。詳細については、「プロキシの承認に関する問題 」を参照してください。

mTLS 認証 (オプション)

外部キーストアプロキシが AWS KMS のリクエストを認証できるようにするために、AWS KMS は、外部キーストアへの Signature V4 (SigV4) プロキシ認証の認証情報を使って、外部キーストアプロキシへのすべてのリクエストに署名します。

外部キーストアプロキシが AWS KMS リクエストにのみ応答することをさらに確実にするために、一部の外部キープロキシでは mutual Transport Layer Security (mTLS) がサポートされています。mTLS ではトランザクションの両者が、互いを認証するために証明書を使用します。mTLS によって、標準 TLS が提供しているサーバー側認証に、クライアント側認証 (外部キーストアプロキシサーバーが AWS KMS クライアントを認証する) がさらに追加されます。万が一、プロキシ認証の認証情報が漏洩した場合、mTLS は、第三者が外部キーストアプロキシに API リクエストを実行することを防ぎます。

mTLS を実装するには、以下のプロパティを持つクライアント側の TLS 証明書のみを受け入れるように、外部キーストアプロキシを設定します。

- TLS 証明書のサブジェクトの共通名は cks.kms.*<Region>*.amazonaws.com のようにします。例えば、cks.kms.*eu-west-3*.amazonaws.com などです。
- この証明書は、Amazon Trust Services に関連づけられた認証局に、紐付けられている必要があります。

外部キーストアの計画

外部キーストアを作成する前に、AWS KMS と外部キーストアコンポーネントとの通信方法を決定する接続オプションを選択します。選択した接続オプションによって、残りの計画プロセスが決まります。

詳細はこちら:

- 前提条件の組み合わせを含む、外部キーストアを作成するためのプロセスを確認します。外部キーストアを作成する際に、必要なコンポーネントがすべて揃っていることを確認するのに役立ちます。
- 外部キーストア管理者およびユーザーが必要とする許可を含む、外部キーストアへのアクセスを制御する方法について説明します。
- が外部キーストアAWS KMSに記録する Amazon CloudWatch のメトリクスとディメンションについて説明します。パフォーマンスや運用上の問題の兆候を早期に検出するために、外部キーストアをモニタリングするアラームを作成することを強くお勧めします。

プロキシ接続オプションの選択

外部キーストアを作成する場合は、AWS KMS と外部キーストアプロキシとの通信方法を決定する必要があります。この選択によって、必要なコンポーネントとその設定方法が決まります。AWS KMS は、次の接続オプションをサポートしています。パフォーマンスとセキュリティの目標に合ったオプションを選択します。

開始する前に、外部キーストアが必要であることを確認してください。ほとんどのお客様は、AWS KMS キーマテリアルによってバックアップされた KMS キーを使用できます。

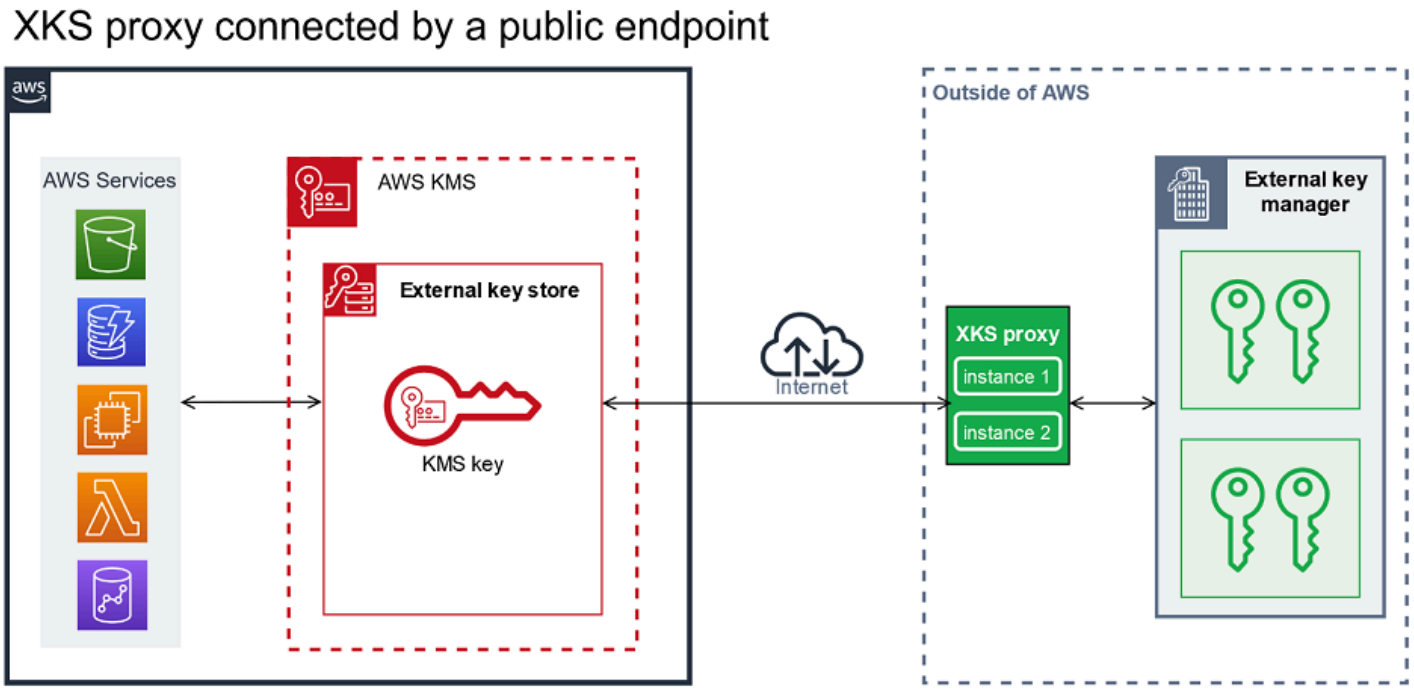
> **Note**
>
> 外部キーストアプロキシが外部キーマネージャーに組み込まれている場合は、接続が事前に決められている可能性があります。ガイダンスについては、外部キーマネージャーまたは外部キーストアプロキシのドキュメントを参照してください。

外部キーストアプロキシの接続オプションは、稼働中の外部キーストアでも変更できます。ただし、中断を最小限に抑え、エラーを回避し、データを暗号化する暗号化キーに継続的にアクセスできるように、プロセスを慎重に計画して実行する必要があります。

パブリックエンドポイント接続

AWS KMS は、パブリックエンドポイントを使用して、インターネット経由で外部キーストアプロキシ (XKS プロキシ) に接続します。

この接続オプションはセットアップと保守が簡単で、一部のキー管理モデルとも問題なく連携します。ただし、一部の組織のセキュリティ要件を満たしていない場合があります。



要件

パブリックエンドポイント接続を選択する場合、以下が必要です。

- 外部キーストアプロキシは、パブリックにルーティング可能なエンドポイントからアクセスできる必要があります。
- プロキシ URI パス値が異なる場合は、複数の外部キーストアに同じパブリックエンドポイントを使用できます。
- キーストアが異なる AWS アカウント にある場合でも、同じ AWS リージョン にパブリックエンドポイント接続がある外部キーストアと VPC エンドポイントサービス接続を備えた外部キーストアに、同じエンドポイントを使用することはできません。
- 外部キーストアでサポートされている公開認証機関が発行した TLS 証明書を取得する必要があります。リストについては、「Trusted Certificate Authorities」(信頼された証明機関) を参照してください。

  TLS 証明書のサブジェクト共通名 (CN) は、外部キーストアプロキシのプロキシ URI エンドポイントのドメイン名と一致する必要があります。例えば、パブリックエンドポイントが `https://myproxy.xks.example.com` の場合、TLS、TLS 証明書の CN は、`myproxy.xks.example.com` または `*.xks.example.com` である必要があります。
- AWS KMS と外部キーストアプロキシ間にあるファイアウォールが、プロキシのポート 443 との間のトラフィックを許可していることを確認します。AWS KMS はポート 443 で通信します。この値は設定できません。

外部キーストアのすべての要件については、前提条件を構成するを参照してください。

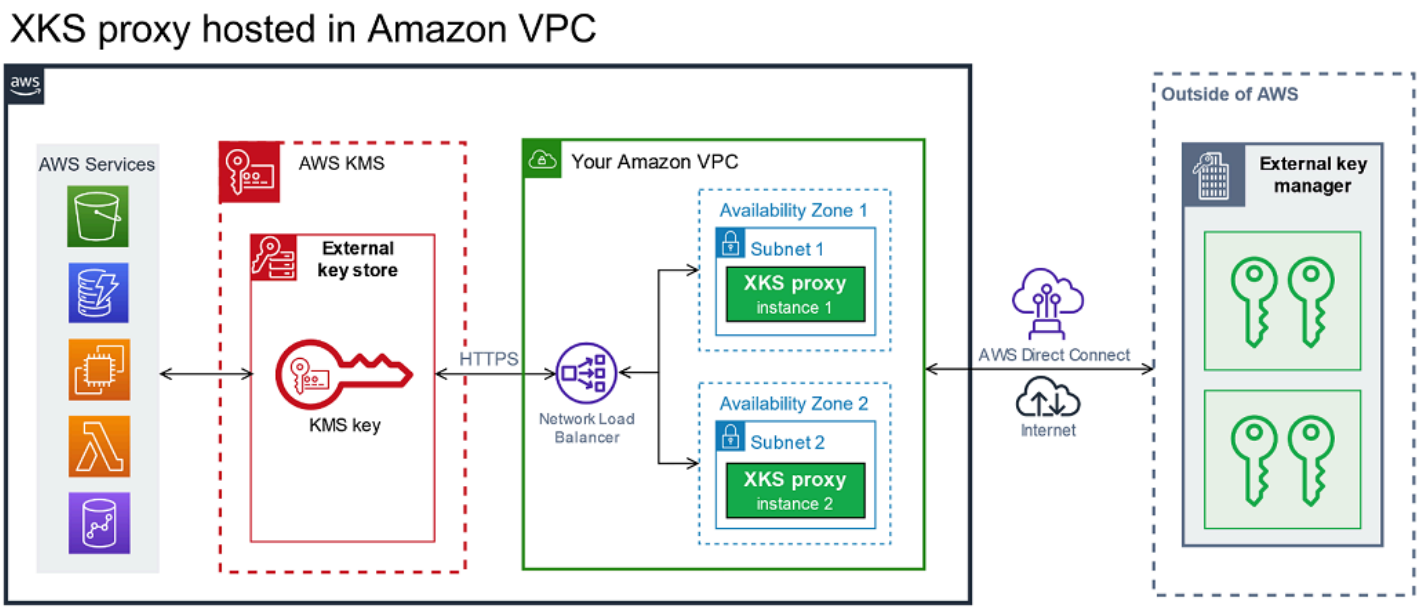
VPC エンドポイントサービス接続

AWS KMS は、作成および設定した Amazon VPC エンドポイントサービスへのインターフェイスエンドポイントを作成することにより、外部キーストアプロキシ (XKS プロキシ) に接続します。ユーザーは、VPC エンドポイントサービスを作成し、VPC を外部キーマネージャーに接続する責任があります。

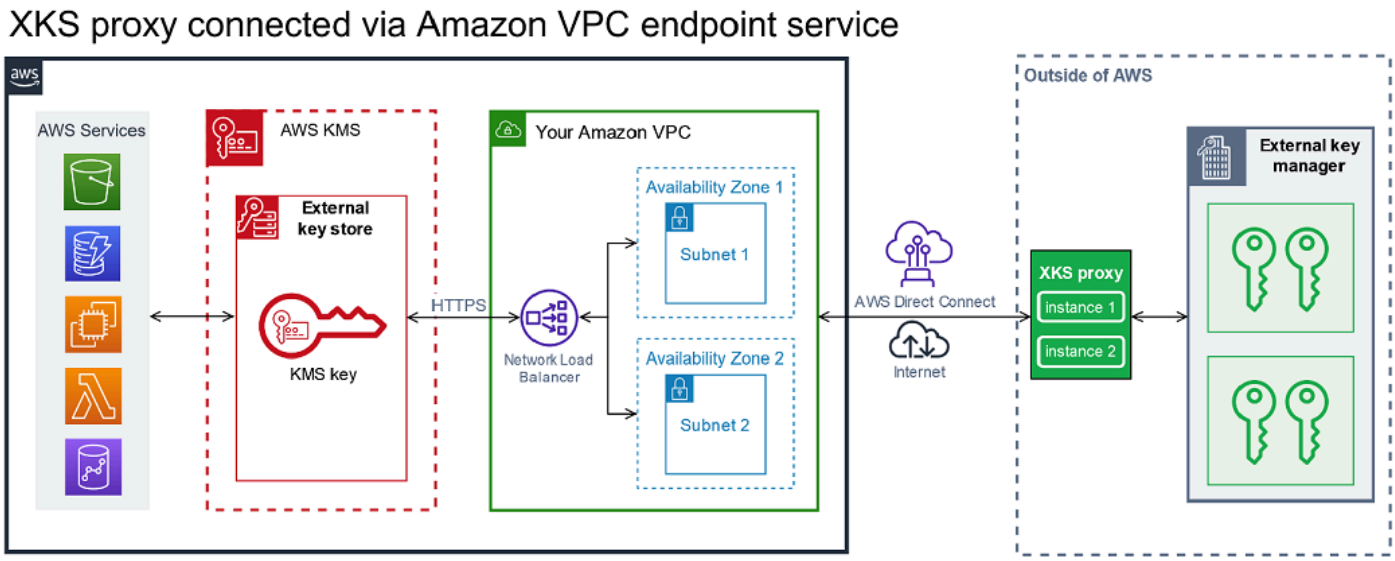
エンドポイントサービスの通信には、AWS Direct Connect を含むすべてのサポートされている Network-to-Amazon VPC オプションを使用できます。

この接続オプションは、セットアップと保守が複雑です。ただし、AWS PrivateLink を使用しているため、AWS KMS は、パブリックインターネットを使用せずに Amazon VPC と外部キーストアプロキシにプライベートに接続できます。

外部キーストアプロキシは Amazon VPC にあります。



または、外部キーストアプロキシを AWS の外部に配置し、Amazon VPC エンドポイントサービスを AWS KMS との安全な通信のみに使用することもできます。



VPC エンドポイントサービス接続の設定

このセクションのガイダンスを使用して、VPC エンドポイントサービス接続を使用する外部キーストアに必要な、AWS リソースと関連コンポーネントを作成および設定します。この接続オプションに一覧表示されているリソースは、すべての外部キーストアに必要なリソースを補足するものです。必要なリソースを作成、設定後に、外部キーストアを作成できます。

外部キーストアプロキシは、Amazon VPC 内に配置することも、AWS の外部に配置して、VPC エンドポイントサービスを使用して通信することもできます。

開始する前に、外部キーストアが必要であることを確認してください。ほとんどのお客様は、AWS KMS キーマテリアルによってバックアップされた KMS キーを使用できます。

> **Note**
>
> VPC エンドポイントサービスの接続に必要な要素の一部は、外部キーマネージャーに含まれている場合があります。また、ソフトウェアに追加の設定要件が存在する場合があります。このセクションの AWS リソースを作成、設定する前に、プロキシとキーマネージャーのドキュメントを参照してください。

トピック

- VPC エンドポイントサービス接続の要件
- Amazon VPC とサブネットを作成する
- ターゲットグループを作成する
- Network Load Balancer を作成する
- VPC エンドポイントサービスを作成する
- プライベート DNS 名ドメインの検証
- AWS KMS が VPC エンドポイントサービスに接続することを承認する

VPC エンドポイントサービス接続の要件

外部キーストアの VPC エンドポイントサービス接続を選択する場合、次のリソースが必要です。

ネットワークのレイテンシーを最小限に抑えるには、外部キーマネージャーに最も近い、サポートされている AWS リージョン で AWS コンポーネントを作成します。可能な場合は、ネットワークラウンドトリップ時間 (RTT) が 35 ミリ秒以下のリージョンを選択します。

- 外部キーマネージャーに接続されている Amazon VPC。2 つの異なるアベイラビリティーゾーンで 2 つ以上のプライベートサブネットが必要です。

  外部キーストアでの使用の要件を満たしている場合は、外部キーストアに既存の Amazon VPC を使用できます。複数の外部キーストアが Amazon VPC を共有できますが、各外部キーストアに独自の VPC エンドポイントサービスとプライベート DNS 名が必要です。
- Network Load Balancer とターゲットグループを備え、AWS PrivateLink を搭載した Amazon VPC エンドポイントサービスです。

エンドポイントサービスは承認を要求できません。また、許可されたプリンシパルとして AWS KMS を追加する必要があります。これにより、AWS KMS はインターフェイスエンドポイントを作成して、外部キーストアプロキシと通信できるようになります。

- AWS リージョン 内で一意の、VPC エンドポイントサービスのプライベート DNS 名です。

  プライベート DNS 名は、上位のパブリックドメインのサブドメインである必要があります。例えば、プライベート DNS 名が `myproxy-private.xks.example.com` の場合、`xks.example.com` または `example.com` などのパブリックドメインのサブドメインである必要があります。

  プライベート DNS 名の DNS ドメインの所有権を検証する必要があります。
- 外部キーストアプロキシのサポートされている公開認証機関が発行した TLS 証明書。

  TLS 証明書のサブジェクト共通名 (CN) は、プライベート DNS 名と一致している必要があります。例えば、プライベート DNS 名が `myproxy-private.xks.example.com` の場合、TLS 証明書の CN は、`myproxy-private.xks.example.com` または `*.xks.example.com` である必要があります。

外部キーストアのすべての要件については、前提条件を構成するを参照してください。

Amazon VPC とサブネットを作成する

VPC エンドポイントサービス接続には、2 つ以上のプライベートサブネットをもつ外部キーマネージャーに接続された Amazon VPC が必要です。Amazon VPC を作成することも、外部キーストアの要件を満たす既存の Amazon VPC を使用することもできます。新しい Amazon VPC 作成のヘルプについては、「Amazon Virtual Private Cloud ユーザーガイド」の「VPC を作成する」を参照してください。

Amazon VPC の要件

VPC エンドポイントサービス接続を使用して外部キーストアを操作するには、Amazon VPC に次のプロパティが必要です。

- 外部キーストアと同じ AWS アカウント、およびサポートされているリージョンに存在する必要があります。
- それぞれ異なるアベイラビリティーゾーンに、2 つ以上のプライベートサブネットが必要です。
- Amazon VPC のプライベート IP アドレス範囲が、外部キーマネージャーをホストするデータセンターのプライベート IP アドレス範囲と重複しないようにする必要があります。

- すべてのコンポーネントが IPv4 を使用する必要があります。

Amazon VPC を外部キーストアプロキシに接続するには、多数の方法があります。必要なパフォーマンスとセキュリティのニーズを満たすオプションを選択します。リストについては、「VPC を他のネットワークに接続する」および「Network-to-Amazon VPC の接続オプション」を参照してください。詳細については、「AWS Direct Connect」、および「AWS Site-to-Site VPN ユーザーガイド」を参照してください。

外部キーストア用 Amazon VPC を作成する

次の手順に従って、外部キーストア用の Amazon VPC を作成します。Amazon VPC は、VPC エンドポイントサービス接続オプションを選択した場合にのみ必要です。外部キーストアの要件を満たす既存の Amazon VPC を使用できます。

次の必須値を使用して、「VPC、サブネット、他の VPC リソースを作成する」トピックの手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
IPv4 CIDR ブロック	VPC の IP アドレスを入力します。Amazon VPC のプライベート IP アドレス範囲が、外部キーマネージャーをホストするデータセンターのプライベート IP アドレス範囲と重複しないようにする必要があります。
アベイラビリティーゾーン数 (AZ)	2 以上
パブリックサブネット数	何も必要ありません (0)
プライベートサブネット数	AZ ごとに 1 つ
NAT ゲートウェイ	何も必要ありません。
VPC エンドポイント	何も必要ありません。

フィールド	値
[Enable DNS hostnames] (DNS ホスト名を有効化)	はい
DNS 解決を有効にする	はい

必ず VPC 通信をテストしてください。例えば、外部キーストアプロキシが Amazon VPC にない場合は、Amazon VPC に Amazon EC2 インスタンスを作成し、Amazon VPC が外部キーストアプロキシと通信できることを検証します。

VPC を外部キーマネージャーに接続する

Amazon VPC がサポートするネットワーク接続オプションのいずれかを使用して、外部キーマネージャーをホストするデータセンターに VPC を接続します。VPC 内の Amazon EC2 インスタンス (または、VPC 内にある場合は外部キーストアプロキシ) がデータセンターおよび外部キーマネージャーと通信できることを確認します。

ターゲットグループを作成する

必要な VPC エンドポイントサービスを作成する前に、必要なコンポーネント、Network Load Balancer (NLB)、ターゲットグループを作成します。Network Load Balancer (NLB) は、リクエストを複数の正常なターゲットに分散し、いずれのターゲットもリクエストを処理できるようにします。このステップでは、外部キーストアプロキシ用に 2 つ以上のホストを含むターゲットグループを作成し、そのターゲットグループに IP アドレスを登録します。

次の必須値を使用して、「ターゲットグループの設定」トピックの手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
対象タイプ	IP アドレス
プロトコル	TCP
ポート	443

フィールド	値
IP アドレスタイプ	IPv4
VPC	外部キーストアの VPC エンドポイントサービスを作成する VPC を選択します。
ヘルスチェックプロトコルとパス	ヘルスチェックプロトコルとパスは、外部キーストアプロキシの設定に応じて異なります。外部キーマネージャーまたは外部キーストアプロキシのドキュメントを参照してください。 ターゲットグループのヘルスチェック設定に関する一般情報は、Elastic Load Balancing ユーザーガイドの Network Load Balancer の項目にある「ターゲットグループのヘルスチェック」を参照してください。
ネットワーク	その他のプライベート IP アドレス
IPv4 アドレス	外部キーストアプロキシのプライベートアドレス
ポート	443

Network Load Balancer を作成する

Network Load Balancer は、外部キーストアプロキシへの AWS KMS からのリクエストなどのネットワークトラフィックを、設定されたターゲットに分散します。

「ロードバランサーとリスナーの設定」トピックの手順に従ってリスナーを設定および追加し、次の必須値を使用してロードバランサーを作成します。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
スキーム	内部
IP アドレスタイプ	IPv4
ネットワークマッピング	外部キーストアの VPC エンドポイントサービスを作成する VPC を選択します。

フィールド	値
マッピング	VPC サブネットに設定した両方のアベイラビリティーゾーン (2 つ以上) を選択します。サブネット名とプライベート IP アドレスを検証します。
プロトコル	TCP
ポート	443
デフォルトアクション: 転送	Network Load Balancer のターゲットグループを選択します。

VPC エンドポイントサービスを作成する

通常、サービスへのエンドポイントを作成します。ただし、VPC エンドポイントサービスを作成すると自分がプロバイダーになり、AWS KMS はサービスへのエンドポイントを作成します。外部キーストアの場合は、前のステップで作成した Network Load Balancer で VPC エンドポイントサービスを作成します。VPC エンドポイントサービスは、外部キーストアと同じ AWS アカウント、およびサポートされているリージョンに存在する必要があります。

複数の外部キーストアが Amazon VPC を共有できますが、各外部キーストアに独自の VPC エンドポイントサービスとプライベート DNS 名が必要です。

「エンドポイントサービスの作成」トピックの手順に従って、次の必須値を含む VPC エンドポイントサービスを作成します。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
ロードバランサーのタイプ	ネットワーク
使用可能なロードバランサー	前のステップで作成した Network Load Balancer を選択します。 新しいロードバランサーがリストに表示されない場合は、その状態がアクティブであることを確認します。ロードバランサーの状態がプロビジョニングからアクティブに変わるまでに、数分かかる場合があります。
承認が必要です	False。チェックボックスをオフにします。

フィールド	値
	承認を必要としません。AWS KMS は、手動による承認なしに VPC エンドポイントサービスに接続できません。承認が必要な場合、外部キーストアを作成しようとすると、XksProxyInvalidConfigurationException の例外が発生して失敗します。
プライベート DNS 名を有効にする	プライベート DNS 名をサービスに関連付ける
プライベート DNS 名	AWS リージョン で一意のプライベート DNS 名を入力します。 プライベート DNS 名は、上位のパブリックドメインのサブドメインである必要があります。例えば、プライベート DNS 名が myproxy-private.xks.example.com の場合、xks.example.com または example.com などのパブリックドメインのサブドメインである必要があります。 このプライベート DNS 名は、外部キーストアプロキシに設定されている TLS 証明書のサブジェクト共通名 (CN) と一致する必要があります。例えば、プライベート DNS 名が myproxy-private.xks.example.com の場合、TLS 証明書の CN は、myproxy-private.xks.example.com または *.xks.example.com である必要があります。 証明書とプライベート DNS 名が一致しない場合、外部キーストアを外部キーストアプロキシに接続しようとすると、XKS_PROXY_INVALID_TLS_CONFIGURATION の接続エラーコードが発生して失敗します。詳細については、「一般的な設定エラー」を参照してください。
サポートされている IP アドレスのタイプ	IPv4

プライベート DNS 名ドメインの検証

VPC エンドポイントサービスを作成すると、そのドメイン検証ステータスは pendingVerification になります。VPC エンドポイントサービスを使用して外部キーストアを作成する前に、このステータスが verified になっている必要があります。プライベート DNS 名

に関連付けられたドメインを所有していることを検証するには、パブリック DNS サーバーに TXT レコードを作成する必要があります。

例えば、VPC エンドポイントサービスのプライベート DNS 名が `myproxy-private.xks.example.com` の場合、`xks.example.com` または `example.com` などのパブリックドメインで TXT レコードを作成する必要があります。AWS PrivateLink は、最初に `xks.example.com` で TXT レコードを検索し、次に `example.com` で検索を続行します。

> **Tip**
>
> TXT レコードを追加した後、[Domain verification status] (ドメイン検証ステータス) の値が `pendingVerification` から `verify` に変わるまでに数分かかる場合があります。

最初に、次のいずれかの方法でドメインの検証ステータスを確認します。有効な値は、`verified`、`pendingVerification`、`failed` です。

- Amazon VPC コンソールで、[Endpoint services] (エンドポイントサービス) を選択し、エンドポイントサービスを選択します。詳細ペインで、[Domain verification status] (ドメイン検証ステータス) を参照してください。
- DescribeVpcEndpointServiceConfigurations 操作を使用します。`State` 値は `ServiceConfigurations.PrivateDnsNameConfiguration.State` フィールドにあります。

検証ステータスが `verified` でない場合は、ドメイン所有権の検証トピックの手順に従ってドメインの DNS サーバーに TXT レコードを追加し、TXT レコードが公開されていることを検証します。次に、検証ステータスを再度チェックします。

プライベート DNS ドメイン名の A レコードを作成する必要はありません。AWS KMS が VPC エンドポイントサービスへのインターフェイスエンドポイントを作成すると、AWS PrivateLink は AWS KMS VPC のプライベートドメイン名に必要な A レコードを含むホストゾーンを自動的に作成します。これは、VPC エンドポイントサービス接続を備える外部キーストアの場合、外部キーストアを外部キーストアプロキシに接続したときに発生します。

AWS KMS が VPC エンドポイントサービスに接続することを承認する

VPC エンドポイントサービスの [Allow principals] (許可プリンシパル) リストに AWS KMS を追加する必要があります。これにより、AWS KMS は VPC エンドポイントサービスへのインターフェイス

エンドポイントを作成できます。AWS KMS が許可されていないプリンシパルの場合、外部キーストアを作成しようとすると、`XksProxyVpcEndpointServiceNotFoundException` の例外が発生して失敗します。

「AWS PrivateLink ガイド」の「許可を管理する」トピックの手順に従ってください。次の必須値を使用します。

フィールド	値
ARN	`cks.kms.<region>.amazonaws.com` 例えば、次のようになります: `cks.kms.us-east-1.amazonaws.com`

Next: 外部キーストアの作成

外部キーストアの管理

AWS KMS コンソールまたは AWS KMS API を使用して、外部キーストアを管理できます。外部キーストアの、作成、プロパティの表示と編集、パフォーマンスのモニタリング、外部キーストアプロキシへの接続と切断、削除を行うことができます。

トピック

- 外部キーストアの作成
- 外部キーストアのプロパティの編集
- 外部キーストアを表示する
- 外部キーストアのモニタリング
- カスタムキーストアの接続と切断
- 外部キーストアの削除

外部キーストアの作成

AWS アカウント とリージョンで、それぞれ 1 つ以上の外部キーストアを作成できます。各外部キーストアは、AWS の外部にある外部キーマネージャー、および AWS KMS と外部キーマネージャーとの通信を仲介する外部キーストアプロキシ (XKS プロキシ) に関連付けられている必要があります。詳細については、「外部キーストアの計画」を参照してください。開始する前に、外部キーストアが

必要であることを確認してください。ほとんどのお客様は、AWS KMS キーマテリアルによってバックアップされた KMS キーを使用できます。

> **Tip**
>
> 外部キーマネージャーの中には、外部 キーを作成するための簡単な方法が用意されているものもあります。詳細については外部キーマネージャーのドキュメントを参照してください。

外部キーストアを作成するときは、事前に前提条件を構成する必要があります。作成中に、外部キーストアのプロパティを指定します。最も重要なことは、AWS KMS の外部キーストアが、外部キーストアプロキシへの接続に、パブリックエンドポイントまたは VPC エンドポイントサービスを使用するのかどうかを示すことです。また、プロキシの URI エンドポイントや、AWS KMS が API リクエストをプロキシに送信するプロキシエンドポイント内のパスなど、接続の詳細も指定します。

- パブリックエンドポイント接続を使用する場合は、AWS KMS が HTTPS 接続を使用してインターネット経由でプロキシと通信できることを確認します。これには、外部キーストアプロキシでの TLS の設定や、AWS KMS とプロキシとの間のあらゆるファイアウォールで、プロキシのポート 443 とのトラフィックを許可するようにすることが含まれます。パブリックエンドポイント接続を使用して外部キーストアを作成する際、AWS KMS は、外部キーストアプロキシにステータスリクエストを送信することで接続をテストします。このテストでは、エンドポイントが到達可能であることと、外部キーストアプロキシが外部キーストアプロキシ認証の認証情報で署名されたリクエストを受け入れることを、確認します。このテストリクエストに失敗すると、外部キーストアを作成するオペレーションは失敗します。
- VPC エンドポイントサービス接続を使用する場合は、Network Load Balancer、プライベート DNS 名、VPC エンドポイントサービスが正しく設定され動作していることを確認します。外部キーストアプロキシが VPC にない場合は、VPC エンドポイントサービスが外部キーストアプロキシと通信できることを確認する必要があります (AWS KMS は、外部キーストアプロキシに外部キーストアを接続するときに VPC エンドポイントサービスの接続をテストします)。

追加の考慮事項

- AWS KMS は、特に外部キーストアの Amazon CloudWatch メトリクスとディメンションを記録します。これらのメトリックの一部に基づいたモニタリングのグラフは、各外部キーストアの AWS KMS コンソールに表示されます。これらのメトリクスを使用して外部キーストアをモニタリングするアラームを作成することが強く推奨されます。これらのアラームは、パフォーマンスやオペ

レーションに関する問題の初期兆候を、発生前にユーザーに警告します。手順については、「外部キーストアのモニタリング」を参照してください。

- 外部キーストアはリソースクォータの影響を受けます。外部キーストアで KMS キーを使用すると、リクエストクォータの影響を受けます。外部キーストアの実装を設計するときは、事前にこれらのクォータを確認します。

> **Note**
>
> 動作を妨げる可能性のある依存関係の循環がないか、設定を確認します。
> たとえば、AWS リソースを使用して外部キーストアプロキシを作成する場合、プロキシを操作するときに、そのプロキシ経由でアクセスされる外部キーストアに KMS キーがあることが必須になっていないことを確認します。

すべての新しい外部キーストアは、切断された状態で作成されます。外部キーストアに KMS キーを作成するときは、外部キーストアを外部キーストアプロキシに接続しておく必要があります。外部キーストアのプロパティを変更するには、外部キーストアの設定を編集します。

トピック

- 前提条件を構成する
- プロキシ設定ファイル
- 外部キーストアを作成する (コンソール)
- 外部キーストアを作成する (API)

前提条件を構成する

外部キーストアを作成する前に、必要なコンポーネント、例えば、外部キーストアのサポートに使用する外部キーマネージャーや、AWS KMS リクエストを外部キーマネージャーが理解できるフォーマットに変換する外部キーストアプロキシなどを、組み立てておく必要があります。

以下のコンポーネントは、すべての外部キーストアに必要です。これらのコンポーネントの他に、選択した外部キーストアプロキシ接続オプションをサポートするコンポーネントを、指定する必要があります。

> **Tip**
>
> 外部キーマネージャーにこれらのコンポーネントの一部が含まれているか、あるいはそれらが自動的に構成されている場合があります。詳細については外部キーマネージャーのドキュメントを参照してください。
> AWS KMS コンソールで外部キーストアを作成する場合は、プロキシ URI パスとプロキシ認証の認証情報を指定する、JSON ベースのプロキシ設定ファイルをアップロードすることができます。一部の外部キーストアプロキシでは、このファイルは自動的に生成されます。詳細については、外部キーストアプロキシか外部キーマネージャーのドキュメントを参照してください。

外部キーマネージャー

各外部キーストアには 1 つ以上の外部キーマネージャーインスタンスが必要です。これには、物理的または仮想的なハードウェアセキュリティモジュール (HSM)、またはキー管理ソフトウェアを使用できます。

キーマネージャーは 1 つだけ使用できますが、冗長性を確保するために、暗号化キーを共有する、関連付けられたキーマネージャーインスタンスを 2 つ以上用意しておくことが推奨されます。外部キーストアでは、外部キーマネージャーを独占的に使用する必要はありません。ただし、外部キーマネージャーには、リソースを保護するために外部キーストアの KMS キーを使用している AWS サービスから届く、予想される頻度での暗号化と復号化のリクエストを処理できる能力が必要です。外部キーマネージャーは、1 秒あたり最大 1,800 件のリクエストを処理し、各リクエストで 250 ミリ秒のタイムアウト内に応答するように設定する必要があります。外部キーマネージャーを AWS リージョン に近い場所に配置し、ネットワークラウンドトリップタイム (RTT) を 35 ミリ秒以下にすることが推奨されます。

外部キーストアプロキシで許可されている場合は、外部キーストアプロキシに関連付ける外部キーマネージャーを変更できますが、新しい外部キーマネージャーは、同じキーマテリアルを持つバックアップまたはスナップショットでなければなりません。KMS キーに関連付ける外部キーが、外部キーストアプロキシで使用できなくなった場合、AWS KMS は、KMS キーで暗号化された暗号文を復号することはできません。

外部キーマネージャーは、外部キーストアプロキシからアクセスできなければなりません。プロキシからのGetHealthStatusレスポンスで、すべての外部キーマネージャーインスタンスが であると報告された場合Unavailable、外部キーストアの作成はすべて で失敗しますXksProxyUriUnreachableException。

外部キーストアプロキシ

ユーザーは、AWS KMS 外部キーストアプロキシ API 仕様の設計要件に従う外部キーストアプロキシ (XKS プロキシ) を指定する必要があります。ユーザーは、外部キーストアプロキシを開発または購入することができ、外部キーマネージャーが提供するまたはそこに組み込まれる外部キーストアプロキシを使用することもできます。AWS KMS では、外部キーストアプロキシを、1 秒あたり最大 1,800 件のリクエストを処理し、各リクエストの 250 ミリ秒のタイムアウト内に応答するよう設定しておくことが推奨されています。外部キーマネージャーを AWS リージョン に近い場所に配置し、ネットワークラウンドトリップタイム (RTT) を 35 ミリ秒以下にすることが推奨されます。

外部キーストアプロキシは複数の外部キーストアに使用できますが、各外部キーストアには、固有の URI エンドポイントと、リクエストに対応する外部キーストアプロキシ内のパスが必要です。

VPC エンドポイントサービス接続を使用している場合、Amazon VPC に外部キーストアプロキシを配置できますが必須ではありません。プロキシは、プライベートデータセンターなど、AWS の外部に配置でき、VPC エンドポイントサービスは、プロキシとの通信にのみ使用できます。

プロキシ認証の認証情報

外部キーストアを作成するときは、外部キーストアプロキシ認証の認証情報 (`XksProxyAuthenticationCredential`) を指定する必要があります。

AWS KMS 用の認証情報 (`XksProxyAuthenticationCredential`) を外部キーストアプロキシに作成します。AWS KMS は、Signature Version 4 (SigV4) プロセスを使用して、外部キーストアプロキシ認証の認証情報でリクエストに署名することで、プロキシを認証します。外部キーストアを作成するときに認証情報を指定します。これはいつでも変更できます。プロキシが認証情報をローテーションする場合は、外部キーストアの認証情報値を更新する必要があります。

プロキシ認証の認証情報は、2 つの要素から成ります。外部キーストアでは両方の要素を指定する必要があります。

- アクセスキー ID: シークレットアクセスキーを識別します。この ID はプレーンテキストで入力できます。
- シークレットアクセスキー: 認証情報のシークレットの部分です。AWS KMS は、認証情報のシークレットアクセスキーを暗号化し、その後保存します。

AWS KMS が外部キーストアプロキシへのリクエストに署名する際に使用する SigV4 認証情報は、AWS アカウントの AWS Identity and Access Management プリンシパルに関連付けられた SigV4 認証情報とは無関係です。IAM SigV4 認証情報を外部キーストアプロキシに再利用しないでください。

プロキシ接続

外部キーストアを作成するときは、外部キーストアプロキシ認証の接続オプション (XksProxyConnectivity) を指定する必要があります。

AWS KMS は、パブリックエンドポイントまたは Amazon Virtual Private Cloud (Amazon VPC) エンドポイントサービスを使用することで、外部キーストアプロキシと通信できます。パブリックエンドポイントは設定と保守が容易ですが、あらゆるインストールのセキュリティ要件を満たしているとは限りません。Amazon VPC エンドポイントサービスの接続オプションを選択する場合は、必要なコンポーネント (2 つの異なるアベイラビリティーゾーンに 2 つ以上のサブネットがある Amazon VPC、ネットワークロードバランサーとターゲットグループを含む VPC エンドポイントサービス、VPC エンドポイントサービスのプライベート DNS 名など) を作成し、管理する必要があります。

外部キーストアのプロキシ接続オプションは変更できます。ただし、外部キーストアの、KMS キーに関連付けられたキーマテリアルが、引き続き利用可能であることを確認する必要があります。利用できない場合、AWS KMS は KMS キーで暗号化された暗号文を復号できません。

外部キーストアに最適なプロキシ接続オプションを決める方法については、プロキシ接続オプションの選択 を参照してください。VPC エンドポイントサービス接続の作成と設定に関するヘルプは、VPC エンドポイントサービス接続の設定 を参照してください。

プロキシ URI エンドポイント

外部キーストアを作成するには、AWS KMS が外部キーストアプロキシにリクエストを送信するときに使用するエンドポイント (XksProxyUriEndpoint) を、指定する必要があります。

このプロトコルは HTTPS でなければなりません。AWS KMS はポート 443 で通信します。プロキシ URI エンドポイント値のポートを指定しないようにします。

- パブリックエンドポイント接続 — 外部キーストアプロキシ用に、公開されているエンドポイントを指定します。このエンドポイントには、外部キーストアを作成する前にアクセスできる必要があります。
- VPC エンドポイントサービス接続 — 後に VPC エンドポイントサービスのプライベート DNS 名が続く `https://` を指定します。

外部キーストアプロキシに設定された TLS サーバ証明書は、外部キーストアプロキシ URI エンドポイントのドメイン名と一致し、外部キーストアでサポートされた認証局が発行している必要があります。リストについては、「Trusted Certificate Authorities」(信頼された証明機関) を参照してください。証明機関は、TLS 証明書を発行する前に、ドメイン所有権の証明を要求します。

TLS 証明書のサブジェクト共通名 (CN) は、プライベート DNS 名と一致している必要があります。例えば、プライベート DNS 名が myproxy-private.xks.example.com の場合、TLS 証明書の CN は、myproxy-private.xks.example.com または *.xks.example.com である必要があります。

プロキシ URI エンドポイントは変更が可能ですが、必ず、外部キーストアプロキシが、外部キーストアの KMS キーに関連付けられたキーマテリアルにアクセスできることを確認します。アクセスできない場合、AWS KMS は KMS キーで暗号化された暗号文を復号できません。

一意性の要件

- プロキシ URI エンドポイント (XksProxyUriEndpoint) とプロキシ URI パス (XksProxyUriPath) を組み合わせた値は、AWS アカウント およびリージョン内で、一意でなければなりません。
- パブリックエンドポイントに接続可能な外部キーストアは、プロキシ URI パス値が異なっている限り、同じプロキシ URI エンドポイントを共有できます。
- パブリックエンドポイントに接続可能な外部キーストアは、同じ AWS リージョン の中では、キーストアが異なる AWS アカウント にあっても、VPC エンドポイントサービスに接続可能な外部キーストアと同じ URI エンドポイント値を使用することはできません。
- VPC エンドポイントに接続可能な各外部キーストアには、それぞれ独自のプライベート DNS 名が必要です。プロキシ URI エンドポイント (プライベート DNS 名) は、AWS アカウント およびリージョン内に一意でなければなりません。

プロキシ URI パス

外部キーストアを作成するには、外部キーストアプロキシで、必要なプロキシ API へのベースパスを指定する必要があります。値は / から始まり、/kms/xks/v1 で終わる必要があります。v1 は、外部キーストアプロキシの AWS KMS API のバージョンを表しています。このパスでは、必須の要素の間に、/example-prefix/kms/xks/v1 のようなプレフィクスをオプションで含めることができます。この値は、外部キーストアプロキシのドキュメントでご確認いただけます。

AWS KMS は、プロキシ URI エンドポイントとプロキシ URI パスの連結により指定されたアドレスに、プロキシリクエストを送信します。例えば、プロキシ URI エンドポイントが https://myproxy.xks.example.com で、プロキシ URI パスが /kms/xks/v1 である場合、AWS KMS はそのプロキシ API リクエストを https://myproxy.xks.example.com/kms/xks/v1 に送信します。

プロキシ URI パスは変更が可能ですが、必ず、外部キーストアプロキシが、外部キーストアの KMS キーに関連付けられたキーマテリアルにアクセスできることを確認します。アクセスできない場合、AWS KMS は KMS キーで暗号化された暗号文を復号できません。

一意性の要件

- プロキシ URI エンドポイント (`XksProxyUriEndpoint`) とプロキシ URI パス (`XksProxyUriPath`) を組み合わせた値は、AWS アカウント およびリージョン内で、一意でなければなりません。

VPC エンドポイントサービス

外部キーストアプロキシとの通信に使用する Amazon VPC エンドポイントサービスの名前を指定します。このコンポーネントが必要なのは、VPC エンドポイントサービス接続を使用する外部キーストアのみです。外部キーストア向け VPC エンドポイントサービスの、セットアップと設定方法のヘルプについては、VPC エンドポイントサービス接続の設定 を参照してください。

VPC エンドポイントサービスは、以下のプロパティを持つ必要があります。

- VPC エンドポイントサービスは、外部キーストアと同じ AWS アカウント およびリージョンにあること。
- Network Load Balancer (NLB) は、それぞれが異なるアベイラビリティーゾーンに存在する 2 つ以上のサブネットに接続されていること。
- VPC エンドポイントサービスの許可プリンシパルリストには、リージョン: `cks.kms.<region>.amazonaws.com` (`cks.kms.us-east-1.amazonaws.com` など) の AWS KMS サービスプリンシパルが含まれていること。
- 接続リクエストの承認を要求しないこと。
- 上位レベルのパブリックドメインにプライベート DNS 名があること。例えば、パブリック `xks.example.com` ドメインに myproxy-private.xks.example.com というプライベート DNS 名を設定できる。

  VPC エンドポイントサービスに接続可能な、外部キーストアのプライベート DNS 名は、その AWS リージョン で一意であること。
- プライベート DNS 名ドメインのドメイン検証ステータスは、`verified` であること。
- 外部キーストアプロキシで設定された TLS サーバ証明書で、エンドポイントに到達可能なプライベート DNS ホスト名が指定されていること。

一意性の要件

- VPC エンドポイントに接続可能な外部キーストアは Amazon VPC を共有することができますが、各外部キーストアには独自の VPC エンドポイントサービスとプライベート DNS 名が必要です。

プロキシ設定ファイル

プロキシ設定ファイルは、外部キーストアのプロキシ URI パスとプロキシ認証の認証情報プロパティの値を含む、オプションの JSON ベースファイルです。AWS KMS コンソールで外部キーストアを作成または編集する場合は、プロキシ設定ファイルをアップロードすることで、外部キーストアの設定値を指定できます。このファイルを使用すれば入力や貼り付けのエラーを回避でき、外部キーストアの値を外部キーストアプロキシの値と確実に一致させることができます。

プロキシ設定ファイルは、外部キーストアプロキシによって生成されます。外部キーストアプロキシでプロキシ設定ファイルが提供されているかどうかは、外部キーストアプロキシのドキュメントでご確認いただけます。

以下は、正しい形式の Proxy 設定ファイルの例です。使用している値は架空のものです。

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
  }
}
```

プロキシ設定ファイルをアップロードできるのは、AWS KMS コンソールで外部キーストアを作成または編集するときだけです。CreateCustomKeyStore または UpdateCustomKeyStoreオペレーションで使用することはできませんが、プロキシ設定ファイルの値を使用してパラメータ値が正しいことを確認することができます。

外部キーストアを作成する (コンソール)

外部キーストアを作成するときは、事前に 外部キーストアの計画 を確認し、プロキシの接続タイプを選択して、必要なコンポーネントがすべて作成され設定されていることを確認します。必要な値を見つける際に不明なことがあれば、外部キーストアプロキシまたはキー管理ソフトウェアのドキュメントを参照してください。

> **Note**
>
> AWS Management Console で外部キーストアを作成するときは、プロキシ URI パスとプロキシ認証の認証情報の値を含む、JSON ベースのプロキシ設定ファイルをアップロードできます。一部のプロキシでは、このファイルは自動的に生成されます。この値は必須ではありません。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. [Create external key store] (外部キーストアの作成) を選択します。
5. カスタムキーストア用のフレンドリ名を入力します。名前は、アカウント内のすべての外部キーストアの中で、一意でなければなりません。

   > **Important**
   >
   > このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

6. プロキシの接続タイプを選択します。

   選択した接続によって、外部キーストアプロキシに必要なコンポーネントが決まります。選択方法の詳細については、プロキシ接続オプションの選択 を参照してください。
7. この外部キーストアの、VPC エンドポイントサービスの名前を選択または入力します。このステップは、外部キーストアプロキシの接続タイプが [VPC endpoint service] (VPC エンドポイントサービス) である場合のみ表示されます。

   VPC エンドポイントサービスとその VPC は、外部キーストアの要件を満たしている必要があります。詳細については、「the section called “前提条件を構成する”」を参照してください。
8. プロキシ URI エンドポイントを入力します。このプロトコルは HTTPS でなければなりません。AWS KMS はポート 443 で通信します。プロキシ URI エンドポイント値のポートを指定しないようにします。

AWS KMS が、前のステップで指定した VPC エンドポイントサービスを認識すると、このフィールドは自動的に入力されます。

パブリックエンドポイント接続の場合は、公開されているエンドポイント URI を入力します。VPC エンドポイントサービス接続では、後ろに VPC エンドポイントサービスのプライベート DNS 名が続く `https://` を入力します。

9. プロキシ URI パスプレフィクスとプロキシ認証の認証情報の値を入力するには、プロキシの設定ファイルをアップロードするか、値を手動で入力します。

   - プロキシ URI パスとプロキシ認証の認証情報の値を含む、オプションのプロキシ設定ファイルがある場合は、[Upload configuration file] (設定ファイルをアップロード) を選択します。ステップに従ってファイルをアップロードします。

     ファイルがアップロードされると、コンソールの編集可能なフィールドに、ファイルの値が表示されます。ここで値を変更できますが、外部キーストアを作成した後でもこれらの値を編集できます。

     シークレットアクセスキーの値を表示するには、[Show secret access key] (シークレットアクセスキーを表示) を選択します。

   - プロキシ設定ファイルがなければ、プロキシ URI パスとプロキシ認証の認証情報の値を手動で入力できます。

     a. プロキシ設定ファイルがなければ、プロキシ URI を手動で入力できます。コンソールに、必要な [/kms/xks/v1] 値が入力されます。

        プロキシ URI パスに、*/example-prefix*/kms/xks/v1 の `example-prefix` など、オプションのプレフィクスが含まれている場合は、そのプレフィクスを [Proxy URI path prefix] (プロキシ URI パスプレフィクス) フィールドに入力します。含まれていなければ、フィールドは空のままにします。

     b. プロキシ設定ファイルがなければ、プロキシ認証の認証情報を手動で入力します。アクセスキー ID とシークレットアクセスキーの両方が必要です。

        - [Proxy credential: Access key ID] (プロキシ認証情報: アクセスキー ID) に、プロキシ認証の認証情報のアクセスキー ID を入力します。アクセスキー ID は、シークレットアクセスキーを識別します。
        - [Proxy credential: Secret access key] (プロキシ認証情報: シークレットアクセスキー) に、プロキシ認証の認証情報の、シークレットアクセスキーを入力します。

シークレットアクセスキーの値を表示するには、[Show secret access key] (シークレットアクセスキーを表示) を選択します。

この手順では、外部キーストアプロキシで作成した認証情報を、設定または変更することはできません。これらの値を、外部キーストアに関連付けるだけです。プロキシ認証の認証情報を設定、変更、ローテーションする方法については、外部キーストアプロキシまたはキー管理ソフトウェアのドキュメントを参照してください。

プロキシ認証の認証情報が変更された場合は、外部キーストアの認証情報の設定を編集します。

10. [Create external key store] (外部キーストアの作成) を選択します。

手順が完了すると、アカウントとリージョンの外部キーストアのリストに、新しい外部キーストアが表示されます。正常に完了しなかった場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「CreateKey 外部キーの エラー」を参照してください。

次の手順: 新しい外部キーストアが自動で接続されない。外部キーストアに AWS KMS keys キーを作成するときは、事前に、外部キーストアプロキシに外部キーストアを接続しておく必要があります。

外部キーストアを作成する (API)

CreateCustomKeyStore オペレーションを使用して、新しい外部キーストアを作成できます。必要なパラメータの値を見つける際に不明なことがあれば、外部キーストアプロキシまたはキー管理ソフトウェアのドキュメントを参照してください。

> **Tip**
>
> `CreateCustomKeyStore` オペレーションを使用しているときは、プロキシ設定ファイルはアップロードできません。ただし、プロキシ設定ファイルの値を使って、パラメータ値が正しいことを確認することは可能です。

外部キーストアを作成するには、`CreateCustomKeyStore` オペレーションでは以下のパラメータ値が必要になります。

- `CustomKeyStoreName` — アカウント内で一意である外部キーストアのフレンドリ名。

> **Important**
>
> このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

- `CustomKeyStoreType` - `EXTERNAL_KEY_STORE` を指定します。
- `XksProxyConnectivity` - `PUBLIC_ENDPOINT` または `VPC_ENDPOINT_SERVICE` を指定します。
- `XksProxyAuthenticationCredential` — アクセスキー ID とシークレットアクセスキーの両方を指定します。
- `XksProxyUriEndpoint` - AWS KMS が外部キーストアプロキシとの通信に使用するエンドポイントです。
- `XksProxyUriPath` — プロキシ内でのプロキシ API へのパスです。
- `XksProxyVpcEndpointServiceName` — `XksProxyConnectivity` 値が `VPC_ENDPOINT_SERVICE` の場合のみ必要です。

> **Note**
>
> AWS CLI バージョン 1.0 を使用している場合は、次のコマンドを実行してから HTTP または HTTPS 値を持つパラメータ (`XksProxyUriEndpoint` パラメータなど) を指定します。
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> そうしないと、AWS CLI バージョン 1.0 が、パラメータ値をこの URI アドレスにある内容に置き換えることになり、以下のエラーが発生します。
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

以下の例の値は架空の値です。コマンドを実行する前に、外部キーストアに有効な値に置き換えてください。

パブリックエンドポイント接続を使用して外部キーストアを作成します。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStorePublic \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity PUBLIC_ENDPOINT \
        --xks-proxy-uri-endpoint https://myproxy.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

VPC エンドポイントサービス接続を使用して外部キーストアを作成します。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStoreVPC \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
        --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-svc-
example \
        --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

オペレーションが正常に終了したら、次のレスポンス例に示すように、CreateCustomKeyStore はカスタムキーストア ID を返します。

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

オペレーションが失敗した場合は、例外で示されているエラーを修正して、もう一度試してください。その他のヘルプについては、「外部キーストアのトラブルシューティング」を参照してください。

次の手順: 外部キーストアを使用するには、これを外部キーストアプロキシに接続します。

外部キーストアのプロパティの編集

既存の外部キーストアの選択したプロパティは、編集が可能です。

一部のプロパティは、外部キーストアが接続または切断されている間、編集することができます。それ以外のプロパティでは、先に、外部キーストアプロキシから外部キーストアを切断する必要があ

ります。外部キーストアの接続ステータスは DISCONNECTED でなければなりません。外部キーストアが切断されている間はキーストアとその KMS キーを管理することはできますが、外部キーストアで KMS キーを作成または使用することはできません。外部キーストアの接続状態を確認するには、DescribeCustomKeyStoresオペレーションを使用するか、外部キーストアの詳細ページの「一般的な設定」セクションを参照してください。

外部キーストアのプロパティを更新する前に、 は新しい値を使用して外部キーストアプロキシに GetHealthStatus リクエストAWS KMSを送信します。リクエストが成功すると、更新されたプロパティ値を使って外部キーストアプロキシに接続し、これを認証できることが示されます。リクエストが失敗すると、編集オペレーションは失敗し、エラーを特定する例外が生じます。

編集オペレーションが完了すると、外部キーストア用の更新されたプロパティ値が、AWS KMS コンソールと DescribeCustomKeyStores レスポンスに表示されます。ただし、変更が完全に有効になるまでに最大で 5 分かかります。

AWS KMS コンソールで外部キーストアを編集する場合は、プロキシ URI パスとプロキシ認証の認証情報を指定する、JSON ベースのプロキシ設定ファイルをアップロードすることができます。一部の外部キーストアプロキシでは、このファイルは自動的に生成されます。詳細については、外部キーストアプロキシか外部キーマネージャーのドキュメントを参照してください。

> **Warning**
>
> 更新されたプロパティ値により、外部キーストアは、以前の値と同じ外部キーマネージャーのプロキシに接続するか、同じ暗号化キーを持つ外部キーマネージャーのバックアップまたはスナップショットのプロキシに接続します。外部キーストアが KMS キーに関連付けられた外部キーへのアクセスを完全に失うと、それらの外部キーで暗号化された暗号文は回復不能になります。特に、外部キーストアのプロキシ接続を変更した場合、AWS KMS は外部キーにアクセスできなくなる可能性があります。

> **Tip**
>
> 外部キーマネージャーの中には、外部キーストアプロパティを編集するための簡単な方法が用意されているものもあります。詳細については外部キーマネージャーのドキュメントを参照してください。

外部キーストアの次のプロパティは、変更が可能です。

編集可能な外部キーストアプロパティ	どの接続ステータスでも可	切断済み状態が必要
カスタムキーストア名 カスタムキーストアの必須フレンドリ名 **Important** このフィールドには、機密情報や重要情報を含めないでください。このフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。	✓	
プロキシ認証情報 (XksProxyAuthenticationCredential) (1 つの要素のみを変更する場合でも、アクセスキー ID とシークレットアクセスキーの両方を指定する必要がある)。	✓	
プロキシ URI パス (XksProxyUriPath)	✓	
プロキシ接続 (XksProxyConnectivity) (プロキシ URI エンドポイントも更新する必要がある。VPC エンドポイントサービス接続に変更する場合、プロキシ VPC エンドポイントサービス名を指定する)。		✓
プロキシ URI エンドポイント (XksProxyUriEndpoint) プロキシエンドポイント URI を変更する場合、関連付けられた TLS 証明書の変更も必要になる場合がある。		✓

編集可能な外部キーストアプロパティ	どの接続ステータスでも可	切断済み状態が必要
プロキシ VPC エンドポイントサービス名 (XksProxyVpcEndpointServiceName) (このフィールドは VPC エンドポイントサービスの接続に必要)。		✓

トピック

- 外部キーストアを編集する (コンソール)
- 外部キーストアを編集する (API)

外部キーストアを編集する (コンソール)

キーストアを編集すると、編集可能な任意の値を変更できます。一部の変更では、外部キーストアを外部キーストアプロキシから切断することが必要になります。

プロキシ URI パスまたはプロキシ認証の認証情報を編集する場合は、新しい値を入力するか、新しい値を含む、外部キーストアのプロキシ設定ファイルをアップロードします。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. 編集する外部キーストアの行を選択します。
5. 必要に応じて、外部キーストアを外部キーストアプロキシから切断します。[Key store actions] (キーストアアクション) メニューから [Disconnect] (切断) を選択します。
6. [Key store actions] (キーストアアクション) メニューから [Edit] (編集) を選択します。
7. 1 つまたは複数の編集可能な外部キーストアプロパティを変更します。また、プロキシ URI パスとプロキシ認証の認証情報の値を含む、外部キーストアプロキシ設定ファイルをアップロードすることもできます。プロキシ設定ファイルは、ファイル内の指定された値が変更されていなくても、使用できます。
8. [Update external key store] (外部キーストアの更新) を選択します。

9. 警告を確認し、続行する場合は警告を確定し、[Update external key store] (外部キーストアの更新) を選択します。

   手順が正常に完了すると、編集したプロパティについて説明するメッセージが表示されます。正常に行われなかった場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。

10. 必要に応じて、外部キーストアを再接続します。[Key store actions] (キーストアアクション) メニューから [Connect] (接続) を選択します。

    外部キーストアは切断された状態にしておくことができます。ただし、切断されている間は外部キーストアで KMS キーを作成したり、暗号化オペレーションで外部キーストア内の KMS キーを使用したりすることはできません。

外部キーストアを編集する (API)

外部キーストアのプロパティを変更するには、 UpdateCustomKeyStoreオペレーションを使用します。同じオペレーションで外部キーストアの複数のプロパティを変更できます。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。

外部キーストアを識別するには `CustomKeyStoreId` パラメータを使用します。プロパティを変更するには他のパラメータを使用します。`UpdateCustomKeyStore` オペレーションでは、プロキシ設定ファイルを使用できません。プロキシ設定ファイルがサポートされているのは、AWS KMS コンソールのみです。ただし、プロキシ設定ファイルは、外部キーストアプロキシの正しいパラメータ値を決める際に役立ちます。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

始める前に、必要に応じて、外部キーストアプロキシから外部キーストアを切断します。更新後、必要に応じて、外部キーストアプロキシに外部キーストアを再接続できます。外部キーストアは切断された状態にしておくことができますが、キーストアに新しい KMS キーを作成したり、暗号化オペレーションのためにキーストアで既存の KMS キーを使用したりするときは、先に接続しておく必要があります。

> **Note**
>
> AWS CLI バージョン 1.0 を使用している場合は、次のコマンドを実行してから HTTP または HTTPS 値を持つパラメータ (`XksProxyUriEndpoint` パラメータなど) を指定します。

```
aws configure set cli_follow_urlparam false
```

そうしないと、AWS CLI バージョン 1.0 が、パラメータ値をこの URI アドレスにある内容に置き換えることになり、以下のエラーが発生します。

```
Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
https:// : received non 200 status code of 404
```

外部キーストアの名前を変更する

最初の例では、 UpdateCustomKeyStoreオペレーションを使用して、外部キーストアのフレンドリ名を に変更しますXksKeyStore。このコマンドでは、CustomKeyStoreId パラメータを使用してカスタムキーストアを識別し、CustomKeyStoreName でカスタムキーストアの新しい名前を指定します。例にある値は、すべて外部キーストアの実際の値に置き換えます。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name XksKeyStore
```

プロキシ認証の認証情報を変更する

次の例では、AWS KMS が外部キーストアプロキシの認証に使用する、プロキシ認証の認証情報を更新します。認証情報を更新するとき、それがプロキシ上でローテーションされる場合は、このようなコマンドを使用できます。

まず、外部キーストアプロキシで認証情報を更新します。次に、この機能を使用して AWS KMS に変更を報告します。(プロキシは、ユーザーが AWS KMS で認証情報を更新できるよう一時的に新旧両方の認証情報をサポートします)。

変更する値が 1 つのみの場合でも、アクセスキー ID とシークレットアクセスキーの両方を認証情報で指定する必要があります。

最初の 2 つのコマンドは、認証情報値を保持する変数を設定します。UpdateCustomKeyStore オペレーションは CustomKeyStoreId パラメータを使って外部キーストアを識別します。XksProxyAuthenticationCredential パラメータをその AccessKeyId と RawSecretAccessKey フィールドで使用し、新しい認証情報を指定します。例にある値は、すべて外部キーストアの実際の値に置き換えます。

```
$ accessKeyID=access key id
$ secretAccessKey=secret access key

$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
       --xks-proxy-authentication-credential \
           AccessKeyId=$accessKeyId,RawSecretAccessKey=$secretAccessKey
```

プロキシ URI パスを変更する

次の例では、プロキシ URI パス (XksProxyUriPath) を更新します。プロキシ URI エンドポイントとプロキシ URI パスを組み合わせた値は、AWS アカウント およびリージョン内で一意でなければなりません。例にある値は、すべて外部キーストアの実際の値に置き換えます。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-uri-path /kms/xks/v1
```

VPC エンドポイントサービス接続を変更する

次の例では、 UpdateCustomKeyStoreオペレーションを使用して、外部キーストアプロキシの接続タイプを に変更しますVPC_ENDPOINT_SERVICE。この変更を行うには、VPC エンドポイントサービスの接続に必要な値 (VPC エンドポイントサービス名 (XksProxyVpcEndpointServiceName)、VPC エンドポイントサービスのプライベート DNS 名を含むプロキシ URI エンドポイント (XksProxyUriEndpoint) 値など) を指定する必要があります。例にある値は、すべて外部キーストアの実際の値に置き換えます。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-connectivity "VPC_ENDPOINT_SERVICE" \
           --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
           --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-
svc-example
```

パブリックエンドポイント接続の変更

次の例では、外部キーストアプロキシの接続タイプを PUBLIC_ENDPOINT に変更します。この変更を行うときは、プロキシ URI エンドポイント (XksProxyUriEndpoint) の値を更新する必要があります。例にある値は、すべて外部キーストアの実際の値に置き換えます。

> **Note**
>
> VPC エンドポイント接続は、パブリックエンドポイント接続に比べてセキュリティが優れています。パブリックエンドポイント接続に変更するときは、外部キーストアプロキシのオンプレミスでの配置や、通信のみに VPC を使用するといった他の選択肢を先に検討します。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
            --xks-proxy-connectivity "PUBLIC_ENDPOINT" \
            --xks-proxy-uri-endpoint https://myproxy.xks.example.com
```

外部キーストアを表示する

AWS KMS コンソールまたは DescribeCustomKeyStoresオペレーションを使用して、各アカウントとリージョンの外部キーストアを表示できます。

外部キーストアを表示すると、以下を確認できます。

- フレンドリ名、ID、キーストアタイプ、作成日など、キーストアに関する基本情報。
- 接続タイプ、プロキシ URI エンドポイントとパス、現在のプロキシ認証情報のアクセスキー ID など、外部キーストアプロキシの設定情報。
- 外部キーストアプロキシが VPC エンドポイントサービス接続を使用する場合、コンソールには VPC エンドポイントサービス名が表示されます。
- 現在の接続状態。

> **Note**
>
> [Disconnected] (切断) の接続状態は、外部キーストアが一度も接続されたことがないこと、または意図的に外部キーストアプロキシから切断されたことを示します。ただし、接続されている外部キーストアで KMS キーの使用が失敗する場合は、外部キーストアまたはそのプロキシに問題がある可能性があります。ヘルプについては、「外部キーストア接続エラー」を参照してください。

- 外部キーストアの問題の検出と解決に役立つように設計された Amazon CloudWatch メトリクスのグラフを含むモニタリングセクション。グラフの解釈、計画とトラブルシューティングでの使用、グラフのメトリクスに基づく CloudWatch アラームの作成については、「」を参照してください外部キーストアのモニタリング。

以下も参照してください。

- 外部キーストアで KMS キーを表示する
- AWS KMS による AWS CloudTrail API コールのログ記録

トピック

- 外部キーストアのプロパティ
- 外部キーストアを表示する (コンソール)
- 外部キーストアを表示する (API)

外部キーストアのプロパティ

外部キーストアの次のプロパティは、AWS KMSコンソールとDescribeCustomKeyStoresレスポンスに表示されます。

カスタムキーストアのプロパティ

各カスタムキーストアの詳細ページの General configuration (一般設定) セクションに次の値が表示されます。これらのプロパティは、AWS CloudHSM キーストアや外部キーストアを含むすべてのカスタムキーストアに適用されます。

カスタムキーストア ID

AWS KMS がカスタムキーストアに割り当てる一意の ID です。

カスタムキーストア名

カスタムキーストア作成時にカスタムキーストアに割り当てるフレンドリ名です。この値はいつでも変更できます。

カスタムキーストアのタイプ

カスタムキーストアのタイプです。有効な値は AWS CloudHSM (`AWS_CLOUDHSM`) または外部キーストア (`EXTERNAL_KEY_STORE`) です。カスタムキーストア作成後、タイプを変更することはできません。

作成日

カスタムキーストアが作成された日付です。この日付は、AWS リージョン の現地時間で表示されます。

接続状態

カスタムキーストアがバッキングキーストアに接続されているかどうかを示します。カスタムキーストアがバッキングキーストアに一度も接続されていないか、意図的に切断されていない限り、接続状態は DISCONNECTED です。詳細については、「the section called “接続状態”」を参照してください。

外部キーストア設定プロパティ

次の値は、各外部キーストアの詳細ページの外部キーストアプロキシ設定セクションと、DescribeCustomKeyStoresレスポンスの XksProxyConfiguration要素に表示されます。一意性要件や、各フィールドの正しい値を決定する際のヘルプなど、各フィールドの詳細な説明については、「外部キーストアの作成」トピックの「the section called “前提条件を構成する”」を参照してください。

プロキシ接続

外部キーストアがパブリックエンドポイント接続を使用しているか、VPC エンドポイントサービス接続を使用しているかを示します。

プロキシ URI エンドポイント

AWS KMS が外部キーストアプロキシへの接続に使用するエンドポイントです。

プロキシ URI パス

AWS KMS がプロキシ API リクエストを送信するプロキシ URI エンドポイントからのパスです。

プロキシ認証情報: アクセスキー ID

外部キーストアプロキシに設定するプロキシ認証情報の一部です。アクセスキー ID は、認証情報のシークレットアクセスキーを識別します。

AWS KMS は、SigV4 署名プロセスとプロキシ認証情報を使用して、外部キーストアプロキシへのリクエストに署名します。署名に含まれる認証情報により、外部キーストアプロキシはユーザーに代わって AWS KMS からのリクエストを認証できます。

VPC エンドポイントサービス名

外部キーストアをサポートする Amazon VPC エンドポイントサービスの名前です。この値は、外部キーストアが VPC エンドポイントサービス接続を使用している場合にのみ表示されます。外部キーストアプロキシを VPC 内に配置するか、VPC エンドポイントサービスを使用して、外部キーストアプロキシと安全に通信できます。

外部キーストアを表示する (コンソール)

任意のアカウントとリージョンで外部キーストアを表示するには、以下の手順に従います。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. 外部キーストアの詳細を表示するには、キーストア名を選択します。

外部キーストアを表示する (API)

外部キーストアを表示するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、`CustomKeyStoreId` または `CustomKeyStoreName` パラメータのどちらかを使用して (両方は使用できません) 出力を特定のカスタムキーストアに制限できます。

カスタムキーストアでは、出力は、カスタムキーストア ID、名前、タイプ、およびキーストアの接続状態で構成されています。接続状態が `FAILED` の場合、出力にはエラーの理由を説明する `ConnectionErrorCode` も含まれています。外部キーストアの `ConnectionErrorCode` を解釈する方法については、「外部キーストアの接続エラーコード」を参照してください。

外部キーストアでは、出力に `XksProxyConfiguration` 要素も含まれています。この要素には、接続タイプ、プロキシ URI エンドポイント、プロキシ URI パス、およびプロキシ認証情報のアクセスキー ID が含まれています。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

たとえば、次のコマンドは、アカウントとリージョンのすべてのカスタムキーストアを返します。`Limit` パラメータと `Marker` パラメータを使用して、出力のカスタムキーストアをページ分割できます。

```
$ aws kms describe-custom-key-stores
```

次のコマンドは、`CustomKeyStoreName` パラメータを使用して、`ExampleXksPublic` というフレンドリ名のサンプル外部キーストアのみを取得します。このサンプルキーストアは、パブリックエンドポイント接続を使用しています。また、外部キーストアプロキシに接続されています。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksPublic
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleXksPublic",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-14T20:17:36.419000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://xks.example.com:6443",
        "UriPath": "/example/prefix/kms/xks/v1"
      }
    }
  ]
}
```

次のコマンドは、VPC エンドポイントサービス接続を備えたサンプル外部キーストアを取得します。この例では、外部キーストアがその外部キーストアプロキシに接続されています。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

Disconnected の ConnectionState は、外部キーストアが一度も接続されたことがないこと、または意図的に外部キーストアプロキシから切断されたことを示します。ただし、接続されている外部キーストアで KMS キーの使用が失敗する場合は、外部キーストアプロキシまたは他の外部コンポーネントに問題がある可能性があります。

外部キーストアの ConnectionState が FAILED の場合、DescribeCustomKeyStores レスポンスには、エラーの理由を説明する ConnectionErrorCode 要素が含まれています。

例えば、次の出力では、XKS_PROXY_TIMED_OUT 値は AWS KMS が外部キーストアプロキシに接続できることを示していますが、外部キーストアプロキシが割り当てられた時間内に AWS KMS に応答しなかったために、接続が失敗しました。この接続エラーコードが繰り返し表示される場合は、外部キーストアプロキシベンダーに通知してください。このエラーやその他の接続エラーに関するヘルプについては、「」を参照してください 外部キーストアのトラブルシューティング。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

外部キーストアのモニタリング

AWS KMS は、外部キーストアとのやり取りごとにメトリクスを収集し、 CloudWatch アカウントに公開します。これらのメトリクスは、各外部キーストアの詳細ページのモニタリングセクションで、グラフを生成するために使用されます。次のトピックでは、グラフを使用して、外部キーストアに影響する運用上および設定上の問題を特定し、トラブルシューティングする方法の詳細を説明

します。 CloudWatch メトリクスを使用して、外部キーストアが想定どおりに動作しない場合に通知するアラームを設定することをお勧めします。詳細については、「Amazon によるモニタリング CloudWatch」を参照してください。

トピック

- グラフを表示する
- グラフの解釈
- アラームの設定

グラフを表示する

さまざまな詳細レベルでグラフを表示できます。デフォルトでは、各グラフは 3 時間の時間範囲と 5 分の集計期間を使用します。コンソール内でグラフビューを調整できますが、外部キーストアの詳細ページを閉じるかブラウザを更新すると、変更はデフォルト設定に戻ります。Amazon CloudWatch の用語については、「Amazon の CloudWatch 概念」を参照してください。

データポイントの詳細を表示する

各グラフのデータは、AWS KMS メトリクスによって収集されます。特定のデータポイントに関する詳細を表示するには、折れ線グラフ上のデータポイントにマウスカーソルを合わせます。これにより、グラフの生成元であるメトリクスに関する詳細情報のポップアップが表示されます。各リスト項目には、そのデータポイントで記録されたディメンション値が表示されます。そのデータポイントのディメンション値に使用できるメトリクスデータがない場合、ポップアップには NULL 値 (—) が表示されます。一部のグラフでは、1 つのデータポイントに対して複数のディメンションと値が記録されます。信頼性グラフなどの他のグラフでは、メトリクスによって収集されたデータを使用して固有の値が計算されます。各リスト項目は、各折れ線グラフの色に関連付けられています。

時間範囲を変更する

グラフの時間範囲を変更するには、モニタリングセクションの右上隅にある事前定義済みの時間範囲の 1 つを選択します。事前定義済みの時間範囲は、1 時間から 1 週間です (1 時間、3 時間、12 時間、1 日、3 日、1 週間)。これにより、すべてのグラフの時間範囲が調整されます。特定のグラフを別の時間範囲で表示する場合、またはカスタムの時間範囲を設定する場合は、グラフを拡大するか、Amazon CloudWatch コンソールで表示します。

グラフを拡大する

ミニマップズーム機能を使用すると、ズームインビューとズームアウトビュー間を変更することなく、折れ線グラフと積み上げ面グラフのセクションに焦点を合わせることができます。例えば、ミニ

マップズーム機能を使用して折れ線グラフのピークに焦点を合わせると、同じタイムラインのモニタリングセクション内の他のグラフに対してスパイクを比較できます。

1. 焦点を合わせるグラフの領域を選択してドラッグし、マウスボタンを放します。
2. ズームをリセットするには、[Reset zoom] (ズームのリセット) アイコンを選択します。これは、内側にマイナス (-) 記号が付いた虫眼鏡のような見た目です。

グラフを拡大する

グラフを拡大するには、個々のグラフの右上隅にあるメニューアイコンを選択してから、[Enlarge] (拡大) を選択します。グラフにカーソルを合わせるとメニューアイコンの横に表示される、拡大アイコンを選択することもできます。

グラフを拡大すると、別の期間、カスタム時間範囲、更新間隔を指定して、グラフの表示をさらに変更できます。これらの変更は、拡大表示を閉じるとデフォルト設定に戻ります。

期間を変更する

1. [Period options] (期間オプション) メニューを選択します。デフォルトでは、このメニューには [5 minutes] (5 分) の値が表示されます。
2. 期間を選択します。事前定義された期間は 1 秒から 30 日です。

   たとえば、1 分間の表示を選択できます。これは、トラブルシューティング時に役立ちます。または、詳細度がより低い 1 時間表示を選択します。これは、時間の経過に伴う傾向を確認できるように、より広い期間 (3 日間など) を表示するときに便利です。詳細については、「Amazon CloudWatch ユーザーガイド」の「期間」を参照してください。

時間範囲またはタイムゾーンを変更する

1. 1 時間 ~ 1 週間の事前定義済み時間範囲 (1 時間、3時間、12 時間、1 日、3 日、1 週間) から 1 つを選択します。また、[Custom] (カスタム) を選択して独自の時間範囲を設定することもできます。
2. [Custom] (カスタム) を選択します。
   a. 時間範囲ボックスの左上隅にある [Absolute] (絶対値) タブを選択します。カレンダーのピッカーまたはテキストフィールドボックスを使用して、時間範囲を指定します。
   b. タイムゾーン: ボックスの右上隅にあるドロップダウンを選択します。タイムゾーンは [UTC] または [Local time zone] (ローカルタイムゾーン) に変更できます。
3. 時間範囲を指定したら、[Apply] (適用) を選択します。

グラフのデータ更新頻度を変更する

1. 右上隅にある [Refresh options] (更新オプション) メニューを選択します。
2. 更新間隔 (オフ、10 秒、1 分、2分、5 分、15 分) を選択します。

Amazon CloudWatch コンソールでグラフを表示する

モニタリングセクションのグラフは、 が Amazon AWS KMS に発行する事前定義されたメトリクスから派生しています CloudWatch。 CloudWatch コンソール内で開き、 CloudWatch ダッシュボードに保存できます。外部キーストアが複数ある場合は、それぞれのグラフを で開き CloudWatch 、1 つのダッシュボードに保存して、その状態と使用状況を比較できます。

CloudWatch ダッシュボードに追加する

右上隅にあるダッシュボードに追加 を選択して、すべてのグラフを Amazon CloudWatch ダッシュボードに追加します。既存のダッシュボードを選択するか、新しいロールを作成できます。このダッシュボードを使用してグラフとアラームのカスタマイズされたビューを作成する方法については、「Amazon CloudWatch ユーザーガイド」の「Amazon ダッシュボードの使用」を参照してください。 CloudWatch

CloudWatch メトリクスで表示する

個々のグラフの右上隅にあるメニューアイコンを選択し、メトリクスで表示を選択して、Amazon CloudWatch コンソールでこのグラフを表示します。 CloudWatch コンソールから、この単一のグラフをダッシュボードに追加し、時間範囲、期間、更新間隔を変更できます。詳細については、「Amazon CloudWatch ユーザーガイド」の「メトリクスのグラフ化」を参照してください。

グラフの解釈

AWS KMS には、AWS KMS コンソール内の外部キーストアの状態をモニタリングするためのグラフがいくつか用意されています。これらのグラフは自動的に設定され、AWS KMS メトリクスから生成されます。

グラフデータは、外部キーストアと外部キーへの呼び出しの一部として収集されます。呼び出しを行わなかった時間範囲のデータがグラフに表示される場合があります。このデータは、外部キーストアプロキシと外部キーマネージャーのステータスを確認するために、AWS KMS がユーザーに代わって定期的な `GetHealthStatus` の呼び出しから取得したものです。グラフに [No data available] (利用可能なデータなし) というメッセージが表示された場合は、その時間帯に呼び出しが記録されなかったか、外部キーストアが `DISCONNECTED` 状態であったことを示します。ビューをより広い時間範囲に調整することで、外部キーストアが切断された時間を特定できる場合があります。

トピック

Total requests

特定の時間範囲に特定の外部キーストアで受信された AWS KMS リクエストの合計です。このグラフを使用して、スロットリングのリスクがあるかどうかを判断します。

AWS KMS は、外部キーマネージャーが 1 秒あたり最大 1,800 件の暗号化オペレーションリクエストを処理できるようにすることを推奨しています。5 分間で呼び出しが 54 万件近くになると、スロットリングのリスクが生じます。

AWS KMS が ExternalKeyStoreThrottle メトリクスでスロットルする外部キーストアの、KMS キーに対する暗号化オペレーションのリクエスト数をモニタリングできます。

「リクエスト率が極めて高いため」、リクエストが拒否されたことを説明するメッセージが表示され、`KMSInvalidStateException` エラーが頻発する場合は、外部キーマネージャーまたは外部キーストアプロキシが現在のリクエストレートに対応できていない可能性があります。可能な場合は、リクエスト率を下げます。また、カスタムキーストアのリクエストクォータ値の引き下げをリクエストすることも検討してください。このクォータ値を減少させるとスロットリングが増える可能性がありますが、これは、超過リクエストが外部キーストアプロキシまたは外部キーマネージャーに送信される前に、AWS KMS が直ちに拒否することを意味します。クォータの削減をリクエストするには、AWS Support センターにアクセスしてケースを作成してください。

リクエストの合計のグラフは、外部キーストアプロキシから AWS KMS が受信した成功と失敗の両方のレスポンスに関するデータを収集する、XksProxyErrors メトリクスから生成されます。特定のデータポイントを表示すると、ポップアップには `CustomKeyStoreId` ディメンションの値と、そのデータポイントで記録された AWS KMS リクエストの合計が表示されます。`CustomKeyStoreId` は常に同じになります。

信頼性

外部キーストアプロキシが成功のレスポンスまたは再試行できないエラーのいずれかを返した AWS KMS リクエストの割合です。このグラフを使用して、外部キーストアプロキシのオペレーション状態を評価します。

グラフに 100% 未満の値が表示されている場合は、プロキシが応答しなかったか、再試行可能なエラーで応答したことを示します。これは、ネットワークの問題、外部キーストアプロキシまたは外部キーマネージャーの速度低下、または実装上のバグを示している可能性があります。

リクエストに不正な認証情報が含まれていてプロキシが AuthenticationFailedException で応答した場合でも、プロキシは外部キーストアプロキシ API リクエストで誤った値を識別するため、グラフには 100% の信頼性が表示され、失敗することが予想されます。信頼性グラフのパーセンテージが 100% の場合、外部キーストアプロキシは想定どおりに応答しています。グラフに 100% 未満の値が表示されている場合、プロキシは再試行可能なエラーで応答したか、タイムアウトしています。例えば、リクエスト率が極めて高いためにプロキシが「ThrottlingException」を返した場合、プロキシはリクエスト失敗の原因となった特定の問題を識別できないため、信頼性の割合が低下します。これは、再試行可能なエラーは一時的な問題である可能性が高く、リクエストを再試行することで解決できるためです。

次のエラーレスポンスは、信頼性の割合を低下させます。例外の上位 5 位 グラフと XksProxyErrors メトリクスを使用して、プロキシが再試行可能な各エラーを返す頻度をさらにモニタリングすることができます。

- InternalException
- DependencyTimeoutException
- ThrottlingException
- XksProxyUnreachableException

信頼性グラフは、AWS KMS が外部キーストアプロキシから受信する成功と失敗の両方のレスポンスに関するデータを収集する、XksProxyErrors メトリクスから生成されます。信頼性の割合は、レスポンスが Retryable の ErrorType 値をもつ場合にのみ低下します。特定のデータポイントを表示すると、ポップアップには CustomKeyStoreId ディメンションの値とともに、そのデータポイントで記録された AWS KMS リクエストの信頼性の割合が表示されます。CustomKeyStoreId は常に同じになります。

XksProxyErrors メトリクスを使用して、1 分間に再試行可能なエラーが 5 回以上記録されたときに警告することで、ネットワークの問題の可能性を通知する 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、「再試行可能なエラーに対する Amazon CloudWatch アラームの作成」を参照してください。

レイテンシー

外部キーストアプロキシが AWS KMS リクエストに応答するまでにかかるミリ秒数です。このグラフを使用して、外部キーストアプロキシと外部キーマネージャーのパフォーマンスを評価します。

AWS KMS では、外部キーストアプロキシが各リクエストに 250 ミリ秒以内に応答することを想定しています。ネットワークがタイムアウトした場合、AWS KMS はリクエストを 1 回再試行します。プロキシが 2 回失敗した場合、記録されるレイテンシーは、両方のリクエスト試行のタイムアウト制限を合わせたもので、グラフには約 500 ミリ秒が表示されます。それ以外の、プロキシが 250 ミリ秒のタイムアウト制限内に応答しないすべての場合、記録されるレイテンシーは 250 ミリ秒です。プロキシが暗号化と復号オペレーションで頻繁にタイムアウトする場合は、外部プロキシ管理者に相談してください。レイテンシー問題のトラブルシューティングについては、「レイテンシーとタイムアウトエラー」を参照してください。

応答が遅い場合は、外部キーマネージャーが現在のリクエストトラフィックを処理できていない可能性もあります。AWS KMS では、外部キーマネージャーが 1 秒あたり最大 1,800 件の暗号化オペレーションリクエストを処理できることを推奨しています。外部キーマネージャーが 1 秒あたり 1,800 件のリクエストを処理できない場合は、カスタムキーストアの KMS キーリクエストクォータの引き下げをリクエストすることを検討してください。外部キーストアの KMS キーを使用した暗号化オペレーションのリクエストは、外部キーストアプロキシまたは外部キーマネージャーによって処理され、後で拒否されるのではなく、スロットリング例外でフェイルファストします。

レイテンシーグラフは XksProxyLatency メトリクスから生成されます。特定のデータポイントを表示すると、ポップアップには `KmsOperation` および `XksOperation` のディメンション値とともに、そのデータポイントで記録されたオペレーションの平均レイテンシーが表示されます。リスト項目は、最大レイテンシーから最低レイテンシーの順に並べられます。

XksProxyLatency メトリクスを使用して、レイテンシーがタイムアウト制限に近づいたときに通知する 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、「応答タイムアウトの Amazon CloudWatch アラームの作成」を参照してください。

例外の上位 5 位

任意の時間範囲で暗号化オペレーションと管理オペレーションが失敗した場合の、例外の上位 5 位です。このグラフを使用して、最も頻発するエラーを追跡することで、エンジニアリング作業に優先順位を付けることができます。

この数には、AWS KMS が外部キーストアプロキシから受信した例外と、AWS KMS が外部キーストアプロキシとの通信を確立できない場合に内部で返される XksProxyUnreachableException が含まれています。

再試行可能なエラーの発生率が高い場合はネットワークエラーの可能性、再試行できないエラーの発生率が高い場合は、外部キーストアの設定に関する問題である可能性があります。例えば、AuthenticationFailedExceptions のスパイクは、AWS KMS で設定されている認証情報と外部キーストアプロキシ間に不一致があることを示します。外部キーストア設定を確認するには、「外部キーストアを表示する」を参照してください。外部キーストア設定を編集するには、「外部キーストアのプロパティの編集」を参照してください。

外部キーストアプロキシから AWS KMS が受け取る例外は、オペレーションが失敗したときに AWS KMS から返される例外とは異なります。AWS KMS 暗号化オペレーションでは、外部キーストアの外部設定または接続状態に関連するすべての障害に対して KMSInvalidStateException が返されます。問題を特定するには、添付のエラーメッセージテキストを使用します。

次の表は、上位 5 位の例外グラフに表示される可能性のある例外と、AWS KMS が返す対応する例外を示しています。

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
再試行不可	**AccessDeniedException** トラブルシューティングヘルプについては、プロキシの承認に関する問題 を参照してください。	CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**AuthenticationFailedException** トラブルシューティングヘルプについては、認証情報エラー を参照してください。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyIncorrectAuthenticationCredentialException**

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
		CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行可能	**DependencyTimeoutException** トラブルシューティングヘルプについては、レイテンシーとタイムアウトエラー を参照してください。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyUriUnreachableException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
再試行可能	**InternalException** 外部キーマネージャーと通信できないため、外部キーストアプロキシがリクエストを拒否しました。外部キーストアプロキシ設定が正しく、外部キーマネージャーが使用可能であることを検証します。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyInvalidResponseException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**InvalidCiphertextException** トラブルシューティングヘルプについては、復号エラー を参照してください。	暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**InvalidKeyUsageException** トラブルシューティングヘルプについては、外部キーの暗号化オペレーションエラー を参照してください。	CreateKey オペレーションに対応する **XksKeyInvalidConfigurationException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
再試行不可	**InvalidStateException** トラブルシューティングヘルプについては、外部キーの暗号化オペレーションエラー を参照してください。	CreateKey オペレーションに対応する **XksKeyInvalidConfigurationException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**InvalidUriPathException** トラブルシューティングヘルプについては、一般的な設定エラー を参照してください。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyInvalidConfigurationException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**KeyNotFoundException** トラブルシューティングヘルプについては、外部キーエラー を参照してください。	CreateKey オペレーションに対応する **XksKeyNotFoundException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
再試行可能	**ThrottlingException** リクエスト率が極めて高いため、外部キーストアプロキシがリクエストを拒否しました。この外部キーストアの KMS キーを使用して、呼び出しの頻度を減らします。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyUriUnreachableException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行不可	**UnsupportedOperationException** トラブルシューティングヘルプについては、外部キーの暗号化オペレーションエラー を参照してください。	CreateKey オペレーションに対応する **XksKeyInvalidResponseException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。

エラータイプ	グラフに表示される例外	AWS KMS が返す例外
再試行不可	**ValidationException** トラブルシューティングヘルプについては、プロキシの問題 を参照してください。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyInvalidResponseException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。
再試行可能	**XksProxyUnreachableException** このエラーが繰り返し表示される場合は、外部キーストアプロキシがアクティブでネットワークに接続されていること、および外部キーストアの URI パスとエンドポイント URI または VPC サービス名が正しいことを確認します。	CreateCustomKeyStore および UpdateCustomKeyStore オペレーションに対応する **XksProxyUriUnreachableException** CreateKey オペレーションに対応する **CustomKeyStoreInvalidStateException** 。 暗号化オペレーションに対応する **KMSInvalidStateException** 。

上位 5 位の例外グラフは、XksProxyErrors メトリクスから生成されます。任意のデータポイントを表示すると、ポップアップに ExceptionName ディメンションの値と、そのデータポイントで例外

が記録された回数が表示されます。5 つのリスト項目は、最高頻度から最低頻度の例外の順に並べられます。

XksProxyErrors メトリクスを使用して、1 分間に再試行不可能なエラーが 5 つ以上記録されたときに警告することで、潜在的な設定問題を通知する 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、「再試行できないエラーに対する Amazon CloudWatch アラームの作成」を参照してください。

証明書の有効期限日数

外部キーストアプロキシエンドポイント (`XksProxyUriEndpoint`) の TLS 証明書の有効期限が切れるまでの日数です。このグラフを使用して、TLS 証明書の有効期限をモニタリングします。

証明書の有効期限が切れると、AWS KMS は外部キーストアプロキシと通信できなくなります。証明書が更新されるまで、外部キーストアの KMS キーで保護されているすべてのデータにアクセスできなくなります。

証明書の有効期限までの日数のグラフは、XksProxyCertificateDaysToExpire メトリクスから生成されます。このメトリクスを使用して、今後の有効期限を通知する CloudWatch アラームを作成することを強くお勧めします。証明書の有効期限が切れると、暗号化されたリソースにアクセスできなくなる可能性があります。アラームを設定することで、組織は有効期限が切れる前に証明書を更新する時間をもつことができます。詳細については、「証明書の有効期限に関する Amazon CloudWatch アラームの作成」を参照してください。

アラームの設定

モニタリングセクションのグラフには、一定期間内の、外部キーストアと外部キーストアの KMS キーの状態の概要が表示されます。ただし、外部キーストアメトリクスに基づいて Amazon CloudWatch アラームを作成して、メトリクス値が指定したしきい値を超えたときに通知を受け取ることができます。アラームは、Amazon Simple Notification Service (Amazon SNS) トピックまたは Amazon EC2 Auto Scaling ポリシーにメッセージを送信できます。 CloudWatch アラームの詳細については、「Amazon ユーザーガイド」の「Amazon CloudWatch アラームの使用」を参照してください。 CloudWatch

Amazon CloudWatch アラームを作成する前に、Amazon SNS トピックが必要です。詳細については、「Amazon ユーザーガイド」の「Amazon SNS トピックの作成 CloudWatch 」を参照してください。

トピック

- 証明書の有効期限に関する Amazon CloudWatch アラームの作成
- 応答タイムアウトの Amazon CloudWatch アラームの作成
- 再試行可能なエラーに対する Amazon CloudWatch アラームの作成
- 再試行できないエラーに対する Amazon CloudWatch アラームの作成

証明書の有効期限に関する Amazon CloudWatch アラームの作成

このアラームは、 が AWS KMS に発行する XksProxyCertificateDaysToExpireメトリクス CloudWatch を使用して、外部キーストアプロキシエンドポイントに関連付けられた TLS 証明書の予想される有効期限を記録します。アカウント内のすべての外部キーストアに対して 1 つのアラームを作成したり、今後作成する可能性のある外部キーストアに対してアラームを作成したりすることはできません。

証明書の有効期限が切れる 10 日前に警告するようアラームを設定することをお勧めしますが、ニーズに最適なしきい値を設定する必要があります。

アラームの作成

「以下の必須値を使用して静的しきい値に基づいて CloudWatch アラームを作成する」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスの選択	[KMS]、[XKS Proxy Certificate Metrics] (XKS プロキシ証明書メトリクス) の順に選択します。 モニタリングする `XksProxyCertificateName` の横にあるチェックボックスをオンにします。 次に [Select metric] (メトリクスの選択) を選択します。
統計)	最小値
[Period] (期間)	5 分
しきい値タイプ	静的
Whenever ..。	XksProxyCertificateDaysToExpire が `Lower`より小さい場合`10`。

応答タイムアウトの Amazon CloudWatch アラームの作成

このアラームは、 が AWS KMS に CloudWatch発行する XksProxyLatencyメトリクスを使用して、外部キーストアプロキシが AWS KMSリクエストに応答するのにかかるミリ秒数を記録します。アカウント内のすべての外部キーストアに対して 1 つのアラームを作成したり、今後作成する可能性のある外部キーストアに対してアラームを作成したりすることはできません。

AWS KMS では、外部キーストアプロキシが各リクエストに 250 ミリ秒以内に応答することを想定しています。外部キーストアプロキシが応答に 200 ミリ秒以上かかった場合に警告するアラームを設定することをお勧めしますが、ニーズに最適なしきい値を設定する必要があります。

アラームの作成

「以下の必須値を使用して静的しきい値に基づいて CloudWatch アラームを作成する」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスの選択	[KMS]、[XKS Proxy Latency Metrics] (XKS プロキシレイテンシーメトリクス) の順に選択します。 モニタリングする `KmsOperation` の横にあるチェックボックスをオンにします。 次に [Select metric] (メトリクスの選択) を選択します。
統計)	[Average] (平均)
[Period] (期間)	5 分
しきい値タイプ	静的
Whenever ..。	XksProxyLatency が `Greater`より小さい場合`200`。

再試行可能なエラーに対する Amazon CloudWatch アラームの作成

このアラームは、 が AWS KMS に発行する XksProxyErrorsメトリクス CloudWatch を使用して、外部キーストアプロキシへのAWS KMSリクエストに関連する例外の数を記録します。アカウント内のすべての外部キーストアに対して 1 つのアラームを作成したり、今後作成する可能性のある外部キーストアに対してアラームを作成したりすることはできません。

再試行可能なエラーは信頼性の割合を低下させます。また、ネットワークエラーを示している可能性があります。1 分間に再試行可能なエラーが 5 回以上記録された場合に警告するアラームを設定することをお勧めしますが、ニーズに最適なしきい値を設定する必要があります。

「以下の必須値を使用して静的しきい値に基づいて CloudWatch アラームを作成する」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスの選択	[Queries] (クエリ) タブを開きます。 [Namespace] (名前空間) として AWS/KMS を選択します。 [Metric name] (メトリクス名) に SUM(XksProxyErrors) を入力します。 [Filter by] (フィルタリング基準) に ErrorType = Retryable を入力します。 [実行] を選択します。次に [Select metric] (メトリクスの選択) を選択します。
ラベル	#########
[Period] (期間)	1 分
しきい値タイプ	静的
Whenever ..。	[q1] が 5 よりも Greater の場合はいつでも。

再試行できないエラーに対する Amazon CloudWatch アラームの作成

このアラームは、 が AWS KMS に発行する XksProxyErrorsメトリクス CloudWatch を使用して、外部キーストアプロキシへのAWS KMSリクエストに関連する例外の数を記録します。アカウント内のすべての外部キーストアに対して 1 つのアラームを作成したり、今後作成する可能性のある外部キーストアに対してアラームを作成したりすることはできません。

再試行不可能なエラーは、外部キーストアの設定に問題があることを示している可能性があります。1 分間に再試行不可能なエラーが 5 回以上記録された場合に警告するアラームを設定することをお勧めしますが、ニーズに最適なしきい値を設定する必要があります。

「以下の必須値を使用して静的しきい値に基づいて CloudWatch アラームを作成する」の手順に従います。他のフィールドについては、デフォルト値を受け入れ、必要に応じて名前を指定します。

フィールド	値
メトリクスの選択	[Queries] (クエリ) タブを開きます。 [Namespace] (名前空間) として AWS/KMS を選択します。 [Metric name] (メトリクス名) に SUM(XksProxyErrors) を入力します。 [Filter by] (フィルタリング基準) に ErrorType = Non-retryable を入力します。 [実行] を選択します。次に [Select metric] (メトリクスの選択) を選択します。
ラベル	##########
[Period] (期間)	1 分
しきい値タイプ	静的
Whenever ..。	[q1] が 5 よりも Greater の場合はいつでも。

カスタムキーストアの接続と切断

新しい外部キーストアが接続されていません。外部キーストアの AWS KMS keys を作成し使用するときは、外部キーストアを外部キーストアプロキシに接続する必要があります。外部キーストアはいつでも接続と切断ができ、その接続ステータスを表示できます。

外部キーストアが切断されている間は、AWS KMS は外部キーストアプロキシと通信できません。それにより、外部キーストアと既存の KMS キーの表示と管理が行えます。ただし、外部キーストアに KMS キーを作成したり、その KMS キーを暗号化オペレーションに使用したりすることはできません。プロパティを編集するときなど外部キーストアの切断が必要になる場合があります。状況に応じて適切に計画を立てます。キーストアを切断すると、その KMS キーを使用する AWS サービスのオペレーションが中断することがあります。

外部キーストアを自分で接続する必要はありません。外部キーストアは、無期限に切断状態のままにし、使用の必要がある場合のみ接続することができます。ただし、定期的に接続をテストして、接続が正しく接続できることを確認するとよいでしょう。

カスタムキーストアを切断すると、そのキーストアの KMS キーはただちに使用できなくなります (結果整合性の影響を受ける)。ただし、KMS キーで保護されたデータキーにより暗号化されたリソースは、KMS キーがデータキーの復号化などで再び使用されるまでは影響を受けません。この問題は AWS のサービス に影響します。その多くが、リソースを保護するためにデータキーを使用しています。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

> **Note**
>
> 外部キーストアは、一度も接続されたことがないか明示的に切断した場合のみ DISCONNECTED ステータスになります。CONNECTED ステータスは、外部キーストアやそのサポートコンポーネントが効率的に機能していることを示すものではありません。外部キーストアコンポーネントのパフォーマンスについては、各外部キーストアの詳細ページにある [Monitoring] (モニタリング) セクションのグラフを参照してください。詳細については、「外部キーストアのモニタリング」を参照してください。
>
> 外部キーマネージャには、AWS KMS 外部キーストアと外部キーストアプロキシ間、または外部キーストアプロキシと外部キーマネージャー間の通信を、停止および再開するその他の方法が用意されている場合があります。詳細については外部キーマネージャーのドキュメントを参照してください。

トピック

- 外部キーストアの接続
- 外部キーストアの切断
- 接続状態
- 外部キーストアを接続する (コンソール)
- 外部キーストアを接続する (API)
- 外部キーストアを切断する (コンソール)
- 外部キーストアを切断する (API)

外部キーストアの接続

外部カスタムキーストアを外部キーストアプロキシに接続しているときは、外部キーストアに KMS キーを作成し、その既存の KMS キーをを暗号化オペレーションに使用できます。

外部キーストアを外部キーストアプロキシに接続するプロセスは、外部キーストアの接続性に応じて異なります。

- 外部キーストアをパブリックエンドポイント接続に接続すると、 は外部キーストアプロキシにGetHealthStatus リクエストAWS KMSを送信して、プロキシ URI エンドポイント 、プロキシ URI パス 、プロキシ認証情報 を検証します。プロキシからのレスポンスにより、プロキシ URI エンドポイントとプロキシ URI パスが正確かつアクセス可能であること、および、外部キーストアのプロキシ認証の認証情報で署名されたリクエストをプロキシが認証したことが確定します。
- VPC エンドポイントサービス接続を使用して外部キーストアを外部キーストアプロキシに接続すると、AWS KMS が次の処理を実行します。
  - プロキシ URI エンドポイントで指定されたプライベート DNS 名のドメインが検証済みであることを確認する。
  - AWS KMS VPC から VPC エンドポイントサービスへのインターフェイスエンドポイントを作成する。
  - プロキシ URI エンドポイントで指定された、プライベート DNS 名のプライベートホストゾーンを作成する。
  - GetHealthStatus リクエストを外部キーストアプロキシに送信します。プロキシからのレスポンスにより、プロキシ URI エンドポイントとプロキシ URI パスが正確かつアクセス可能であること、および、外部キーストアのプロキシ認証の認証情報で署名されたリクエストをプロキシが認証したことが確定します。

接続オペレーションによりカスタムキーストアを接続するプロセスが開始しますが、外部キーストアが外部プロキシに接続されるまでに約 5 分かかります。接続オペレーションからのレスポンスは、外部キーストアが接続されたことを示すものではありません。接続が成功したことを確認するには、AWS KMSコンソールまたは DescribeCustomKeyStoresオペレーションを使用して、 キーストアの外部接続状態を表示します。

接続ステータスが FAILED である場合、接続エラーコードは AWS KMS コンソールに表示されて、DescribeCustomKeyStore レスポンスに追加されます。接続エラーコードの解釈については、「外部キーストアの接続エラーコード」を参照してください。

外部キーストアの切断

VPC エンドポイントサービスに接続している外部キーストアを、外部キーストアプロキシから切断すると、AWS KMS は、VPC エンドポイントサービスとのインターフェイスエンドポイントを削除し、接続をサポートするために作成されたネットワークインフラストラクチャを削除します。パブリックエンドポイントに接続している外部キーストアには、同等のプロセスは必要ありません。こ

のアクションは、VPC エンドポイントサービスやそのサポートコンポーネントには影響せず、外部キーストアプロキシや外部コンポーネントにも影響しません。

外部キーストアが切断されている間は、AWS KMS は外部キーストアプロキシにリクエストを送信しません。外部キーストアの接続状ステータスは DISCONNECTED です。切断された外部キーストアの KMS キーは、(削除が保留されている場合を除き) UNAVAILABLE のキーステータスになります。つまり、暗号化オペレーションには使用できません。しかし、外部キーストアと既存の KMS キーの表示および管理は引き続き行えます。

切断状態は一時的なものとして、また元に戻せるように設計されています。外部キーストアはいつでも再接続できます。通常、再度設定する必要はありません。ただし、関連付けられた外部キーストアプロキシのプロパティが、プロキシ認証の認証情報のローテーションなどにより変更されている場合は、再度接続する前に、外部キーストアの設定を編集する必要があります。

> **Note**
>
> カスタムキーストアが切断されている間は、カスタムキーストアで KMS キーを作成したり、暗号化オペレーションで既存の KMS キーを使用したりする試みはすべて失敗します。このオペレーションにより、ユーザーが機密データを保存したりアクセスしたりすることを防ぐことができます。

外部キーストアの切断による影響を、より正確に予測するには、外部キーストアで KMS キーを識別して過去の使用状況を特定します。

外部キーストアを切断する理由としては、次のようなものが挙げられます。

- プロパティを編集するには。外部キーストアが接続されている間は、カスタムキーストア名、プロキシ URI パス、プロキシ認証の認証情報の編集が行えます。ただし、プロキシの接続タイプ、プロキシ URI エンドポイント、VPC エンドポイントのサービス名を編集するには、先に外部キーストアを切断しておく必要があります。詳細については、「外部キーストアのプロパティの編集」を参照してください。
- AWS KMS と外部キーストアプロキシとの通信をすべて停止するには。エンドポイントまたは VPC エンドポイントサービスを無効にすれば、AWS KMS とプロキシとの通信を停止することもできます。さらに、外部キーストアプロキシまたはキー管理ソフトウェアには、AWS KMS とプロキシとの通信を防いだり、プロキシと外部キーマネージャーとのアクセスを防いだりする追加の機能が用意されている場合があります。

- 外部キーストアですべての KMS キーを無効にするには。AWS KMS コンソールまたは DisableKeyオペレーションを使用して、外部キーストアで KMS キーを無効または再度有効にすることができます。これらのオペレーションはすぐに完了します (結果整合性の影響を受ける) が、一度に 1 つの KMS キーしか処理されません。外部キーストアを切断すると外部キーストアのすべての KMS キーのキーステータスが `Unavailable` に変更され、暗号化オペレーションに使用できなくなります。
- 失敗した接続試行を修復するには。外部キーストアを接続する試みに失敗した場合 (カスタムキーストアの接続ステータスが `FAILED` になる) は、再度接続を試みる前に外部キーストアを切断する必要があります。

接続状態

接続と切断により、カスタムキーストアの接続ステータスが変更されます。接続ステータスの値は、AWS CloudHSM キーストアと外部キーストアで同じです。

カスタムキーストアの接続状態を表示するには、 DescribeCustomKeyStoresオペレーションまたは AWS KMS コンソールを使用します。[Connection state] (接続ステータス) は、各カスタムキーストアテーブルに表示され、各カスタムキーストアの詳細ページ内の [General configuration] (一般設定) のセクション、およびカスタムキーストアの KMS キーの [Cryptographic configuration] (暗号化設定) タブに表示されます。詳細については、「AWS CloudHSM キーストアの表示」および「外部キーストアを表示する」を参照してください。

カスタムキーストアの接続ステータスは、次のいずれかになります。

- `CONNECTED`: カスタムキーストアはバッキングキーストアに接続された状態です。ユーザーは、カスタムキーストアで KMS キーを作成し使用することができます。

  AWS CloudHSM キーストアのバッキングキーストアは、関連付けられた AWS CloudHSM クラスターです。外部キーストアのバッキングキーストアは、外部キーストアプロキシと、それがサポートしている外部キーマネージャーです。

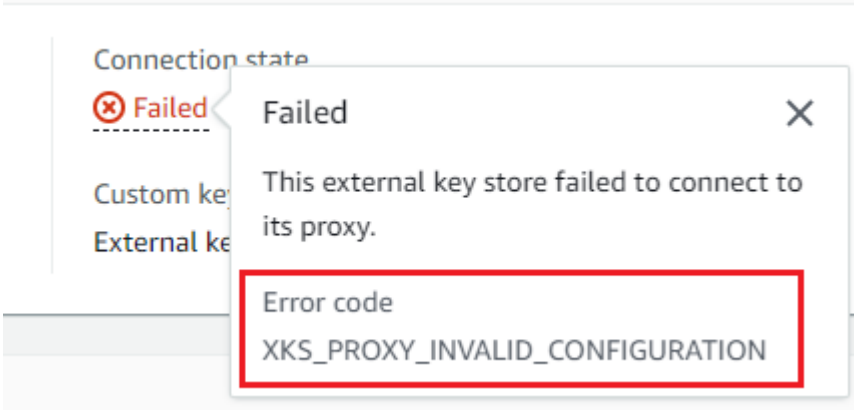
  CONNECTED ステータスは、接続に成功し、カスタムキーストアが意図的に切断されていない状態を示します。接続が正常に機能していることを示すものではありません。AWS CloudHSM キーストアに関連付けられたAWS CloudHSMクラスターのステータスについては、 AWS CloudHSM ユーザーガイドの「 の CloudWatch メトリクスAWS CloudHSMの取得」を参照してください。外部キーストアのステータスとオペレーションについては、各外部キーストアの詳細ページにある [Monitoring] (モニタリング) セクションのグラフを参照してください。詳細については、「外部キーストアのモニタリング」を参照してください。

- CONNECTING: カスタムキーストアの接続処理が進行している状態です。これは過渡的な状態です。
- DISCONNECTED: カスタムキーストアがバッキングに接続されていないか、AWS KMSコンソールまたは DisconnectCustomKeyStoreオペレーションを使用して意図的に切断されました。
- DISCONNECTING: カスタムキーストアを接続する処理が進行中です。これは過渡的な状態です。
- FAILED: カスタムキーストアを接続しようとして失敗した状態です。DescribeCustomKeyStores レスポンスConnectionErrorCodeの は問題を示します。

カスタムキーストアを接続するには、接続ステータスが DISCONNECTED 担っていなければなりません。接続ステータスが FAILED である場合、ConnectionErrorCode を使用して問題を特定し、解決します。カスタムキーストアを切断し、再度接続を試みます。接続障害については、外部キーストア接続エラー を参照してください。接続エラーコードの対処方法については「外部キーストアの接続エラーコード」を参照してください。

接続エラーコードを表示するには:

- DescribeCustomKeyStores レスポンスで、 ConnectionErrorCode要素の値を表示します。この要素は、ConnectionState が FAILED の場合にのみ、DescribeCustomKeyStores レスポンスに表示されます。
- AWS KMS コンソールに接続エラーコードを表示するには、外部キーストアの詳細ページで、[Failed] (失敗) 値にカーソルを合わせます。



外部キーストアを接続する (コンソール)

外部キーストアを外部キーストアプロキシに接続するには、AWS KMS コンソールを使用します。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. 接続する外部キーストアの行を選択します。

   外部キーストアの接続ステータスが FAILED である場合、外部キーストアを切断してから接続します。
5. [Key store actions] (キーストアアクション) メニューから [Connect] (接続) を選択します。

通常、接続プロセスが完了するまでに約 5 分かかります。オペレーションが完了すると、接続ステータスは CONNECTED に変わります。

接続状態が Failed になった場合は、接続ステータスにカーソルを合わせると、エラーの原因を示す接続エラーコードが表示されます。接続エラーコードの対処方法については「外部キーストアの接続エラーコード」を参照してください。接続ステータスが Failed である外部キーストアに接続するには、先にカスタムキーストアを切断します。

外部キーストアを接続する (API)

切断された外部キーストアを接続するには、 ConnectCustomKeyStoreオペレーションを使用します。

接続する前、外部キーストアの接続ステータスは `DISCONNECTED` になっているはずです。接続ステータスが `FAILED` である場合は、外部キーストアを切断してから接続します。

この接続処理は、約 5 分で完了します。すぐに失敗しない限り、`ConnectCustomKeyStore` は、HTTP 200 レスポンスと、プロパティを含まない JSON オブジェクトを返します。ただし、この初期レスポンスは接続に成功したことを示していません。外部キーストアが接続されているかどうかを判断するには、DescribeCustomKeyStoresレスポンスの接続状態を参照してください。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

外部キーストアを識別するには、カスタムキーストア ID を使用します。ID は、 コンソールのカスタムキーストアページで、または DescribeCustomKeyStoresオペレーションを使用して確認できます。この例を実行する前に、例の ID を有効な ID に置き換えます。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

ConnectCustomKeyStore オペレーションは、そのレスポンスで ConnectionState を返しません。外部キーストアが接続されていることを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、CustomKeyStoreId または CustomKeyStoreName パラメータのどちらかを使用して (両方は使用できません) レスポンスを特定のカスタムキーストアに制限できます。CONNECTED の ConnectionState 値は、外部キーストアが外部キーストアプロキシに接続されていることを示します。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

DescribeCustomKeyStores の ConnectionState 値が FAILED である場合、ConnectionErrorCode 要素に失敗した原因が示されます。

次の例では、ConnectionErrorCode の XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND 値は、AWS KMS が、外部キーストアプロキシとの通信に使用する VPC エンドポイントサービスを特定できないことを示します。XksProxyVpcEndpointServiceName が正しいこと、AWS KMS サービスプリンシパルが Amazon VPC エンドポイントサービスで許可されたプリンシパルであること、VPC エンドポイントサービスで接続リクエストを受け入れる必要がないことを確認します。接続エラーコードの対処方法については「外部キーストアの接続エラーコード」を参照してください。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

外部キーストアを切断する (コンソール)

外部キーストアを外部キーストアプロキシに接続するには、AWS KMS コンソールを使用します。この処理は約 5 分で完了します。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. 切断する外部キーストアの行を選択します。
5. [Key store actions] (キーストアアクション) メニューから [Disconnect] (切断) を選択します。

オペレーションが完了すると、接続状態が [DISCONNECTING] から [DISCONNECTED] に変わります。オペレーションが失敗した場合は、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「外部キーストア接続エラー」を参照してください。

外部キーストアを切断する (API)

接続されている外部キーストアを切断するには、 DisconnectCustomKeyStoreオペレーションを使用します。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。この処理は約 5 分で完了します。外部キーストアの接続状態を確認するには、 DescribeCustomKeyStoresオペレーションを使用します。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

こちらの例では、VPC エンドポイントサービスに接続した外部キーストアを切断します。このコマンドを実行する前に、例にあるカスタムキーストア ID を有効な ID に置き換えます。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

外部キーストアが切断されていることを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。デフォルトでは、このオペレーションは、アカウントとリージョンのすべてのカスタムキーストアを返します。ただし、CustomKeyStoreId または CustomKeyStoreName パラメータ のどちらかを使用して (両方は使用できません) レスポンスを特定のカスタムキーストアに制限できます。DISCONNECTED の ConnectionState 値は、例にある外部キーストアが、外部キーストアプロキシに接続されていないことを示します。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

外部キーストアの削除

外部キーストアを削除すると、AWS KMS は、外部キーストアプロキシに関する情報を含め、外部キーストアに関するすべてのメタデータを AWS KMS から削除します。このオペレーションは、外部キーストアプロキシ、外部キーマネージャー、外部キー、または、外部キーストアをサポートするために作成された AWS リソース (Amazon VPC や VPC エンドポイントサービスなど) には影響しません。

外部キーストアを削除するときは、先に、キーストアからすべての KMS キーを削除し、外部キーストアプロキシからキーストアを切断しておきます。さもないと、キーストアを削除することができません。

外部キーストアを削除すると元には戻せませんが、新しい外部キーストアを作成し、同じ外部キーストアプロキシと外部キーマネージャーに関連付けることができます。ただし、同じ外部キーマテリアルにアクセスできる場合であっても、外部キーストアに対称暗号化 KMS キーを再作成することはできません。AWS KMS には、各 KMS キーに固有の対称暗号文のメタデータが含まれています。このセキュリティ機能により、データを暗号化した KMS キーのみが、そのデータを復号することができます。

外部キーストアを削除するのではなく、切断することを検討します。外部キーストアが切断されている間は、外部キーストアとその AWS KMS keys を管理することはできますが、外部キーストアで KMS キーを作成または使用することはできません。外部キーストアはいつでも再接続でき、KMS キーを使用して、データの暗号化と復号化を再開できます。切断された外部キーストアプロキシや使用できない KMS キーには、コストは発生しません。

トピック

- 外部キーストアを削除する (コンソール)
- 外部キーストアを削除する (API)

外部キーストアを削除する (コンソール)

AWS KMS コンソールを使用することで、外部キーストアを削除できます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。

4. 削除する外部キーストアが表示されている行を探します。外部キーストアの [Connection state] (接続ステータス) が DISCONNECTED になっていなければ、先にカスタムキーストアを切断し、その後に削除します。
5. [Key store actions] (キーストアアクション) メニューから [Delete] (削除) を選択します。

オペレーションが完了すると成功メッセージが表示され、この外部 キーストアはカスタムキーストアリストに表示されなくなります。オペレーションが正常に行われなかった場合、問題を説明し、修正方法を示すエラーメッセージが表示されます。さらにヘルプが必要な場合は、「外部キーストアのトラブルシューティング」を参照してください。

外部キーストアを削除する (API)

外部キーストアを削除するには、 DeleteCustomKeyStoreオペレーションを使用します。オペレーションが成功すると、AWS KMS は HTTP 200 レスポンスおよびプロパティなしの JSON オブジェクトを返します。

まず、外部キーストアを切断します。このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

外部キーストアが切断されたら、 DeleteCustomKeyStoreオペレーションを使用して削除できます。

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

外部キーストアが削除されたことを確認するには、 DescribeCustomKeyStoresオペレーションを使用します。

```
$ aws kms describe-custom-key-stores

{
    "CustomKeyStores": []
}
```

存在しないカスタムキーストア名または ID を指定すると、AWS KMS が `CustomKeyStoreNotFoundException` 例外を返します。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
```

```
An error occurred (CustomKeyStoreNotFoundException) when calling the
 DescribeCustomKeyStore operation:
```

外部キーストアで KMS キーを管理する

外部キーストアで KMS キーを作成、表示、管理、使用し、削除をスケジュールするには、他の KMS キーに使用する手順と極めてよく似た手順を使用します。ただし、外部キーストアに KMS キーを作成する場合は、外部キーストアと外部キーを指定します。外部キーストアで KMS キーを使用すると、指定された外部キーにより、外部キーマネージャーが暗号化および復号オペレーションを実行します。

AWS KMS は、外部キーマネージャーで暗号化キーを作成、表示、更新、削除することはできません。また、AWS KMS が外部キーマネージャーや外部キーに直接アクセスすることはありません。暗号化オペレーションのリクエストはすべて、外部キーストアプロキシによって仲介されます。外部キーストアで KMS キーを使用するには、KMS キーをホストする外部キーストアを、外部キーストアプロキシに接続する必要があります。

サポートされている機能

このセクションで説明する手順に加えて、外部キーストアでは KMS キーを使用して次のことを実行できます。

- キーポリシー、IAM ポリシー、グラントを使用して、KMS キーへのアクセスを管理します。
- KMS キーを有効および無効にします。これらのアクションは、外部キーマネージャーの外部キーには影響しません。
- タグを割り当ててエイリアスを作成し、属性ベースのアクセス制御 (ABAC) を使用して KMS キーへのアクセスを承認します。
- AWS KMS を統合し、カスタマーマネージドキーをサポートする AWS のサービスで KMS キーを使用します。

サポートされていない機能

- 外部キーストアは、対称暗号化 KMS キーのみをサポートしています。外部キーストアで HMAC KMS キーや非対称 KMS キーを作成することはできません。
- GenerateDataKeyPair および GenerateDataKeyPairWithoutPlaintextは、外部キーストアの KMS キーではサポートされていません。
- AWS CloudFormation テンプレートを使用して外部キーストアを作成したり、外部キーストアに KMS キーを作成したりすることはできません。

- マルチリージョンキーは、外部キーストアではサポートされていません。
- キーマテリアルがインポートされた KMS キーは、外部キーストアではサポートされていません。
- 自動キーローテーションは、カスタムキーストアの KMS キーではサポートされていません。

トピック

- 外部キーストアで KMS キーを作成する
- 外部キーストアで KMS キーを表示する
- 外部キーストアで KMS キーを使用する
- 外部キーストアの KMS キーの削除をスケジュールする

外部キーストアで KMS キーを作成する

外部キーストアを作成、接続した後で、AWS KMS keys をキーストアで作成できます。これらは、[External key store] (外部キーストア) (EXTERNAL_KEY_STORE) のオリジン値をもつ対称暗号化 KMS キーである必要があります。カスタムキーストアで非対称 KMS キー、HMAC KMS キー、またはインポートされたキーマテリアルを持つ KMS キーを作成することはできません。カスタムキーストア内の対称暗号化 KMS キーを使用して、非対称データキーペアを生成することもできません。

外部キーストアの KMS キーは AWS の外部にあるコンポーネントに依存するため、標準 KMS キーよりも遅延、耐久性、可用性に劣る可能性があります。外部キーストアで KMS キーを作成または使用する前に、外部キーストアプロパティをもつキーが必要であることを確認してください。

> **Note**
>
> 一部の外部キーマネージャーは、外部キーストアに KMS キーを作成する簡単な方法を提供しています。詳細については外部キーマネージャーのドキュメントを参照してください。

外部キーストアで KMS キーを作成するには、以下を指定します。

- 外部キーストアの ID。
- 外部キーストア (EXTERNAL_KEY_STORE) のキーマテリアルオリジン。
- 外部キーストアに関連付けられた外部キーマネージャー内の既存の外部キー ID。この外部キーは、KMS キーのキーマテリアルとして機能します。KMS キーの作成後は、外部キー ID を変更できません。

AWS KMS は、暗号化および復号化オペレーションのリクエストで外部キーストアプロキシに外部キー ID を提供します。AWS KMS は、外部キーマネージャーやその暗号化キーに直接アクセスすることはできません。

外部キーに加えて、外部キーストアの KMS キーにも AWS KMS キーマテリアルがあります。KMS キーで暗号化されたすべてのデータは、最初にキーの AWS KMS キーマテリアルを使用して AWS KMS で暗号化され、次に外部キーを使用して外部キーマネージャーによって暗号化されます。この二重暗号化プロセスにより、外部キーストアの KMS キーで保護された暗号文は、少なくとも AWS KMS のみで保護されている暗号文と同等の強度を持つことが保証されます。詳細については、「外部キーストアの仕組み」を参照してください。

CreateKey オペレーションが成功すると、新しい KMS キーのキーステータスは Enabled になります。外部キーストアで KMS キーを表示すると、キー ID、キースペック、キーの使用方法、キーの状態、作成日などの一般的なプロパティが表示されます。また、外部キーストアの ID と接続状態、および外部キーの ID も表示されます。

外部キーストアで KMS キーの作成を試みて失敗した場合は、エラーメッセージを使用して原因を特定します。外部キーストアが接続されていない可能性（CustomKeyStoreInvalidStateException）、外部キーストアプロキシが指定された外部キー ID（XksKeyNotFoundException）の外部キーを検出できない可能性、または外部キーが同じ外部キーストア XksKeyAlreadyInUseException 内の KMS キーにすでに関連付けられている可能性があります。

.

外部キーストアで KMS キーを作成するオペレーションの AWS CloudTrail ログの例については、「CreateKey」を参照してください。

トピック

- 外部キーストアの KMS キーの要件
- 外部キーストアで KMS キーを作成する (コンソール)
- 外部キーストアで KMS キーを作成する (AWS KMS API)

外部キーストアの KMS キーの要件

外部キーストアに KMS キーを作成するには、外部キーストア、KMS キー、および KMS キーの外部暗号化キーマテリアルとなる外部キーの次のプロパティが必要です。

外部キーストアの要件

- 外部キーストアプロキシに接続する必要があります。

  外部キーストアの接続状態を確認するには、「外部キーストアを表示する」を参照してください。外部キーストアを接続するには、「カスタムキーストアの接続と切断」を参照してください。

KMS キーの要件

KMS キーの作成後にこれらのプロパティを変更することはできません。

- キースペック: SYMMETRIC_DEFAULT
- キーの使用方法: ENCRYPT_DECRYPT
- キーマテリアルのオリジン: EXTERNAL_KEY_STORE
- マルチリージョン: FALSE

外部キーの要件

- 256 ビット AES 暗号化キー (256 ランダムビット)。外部キーの KeySpec は AES_256 である必要があります。
- 有効かつ使用可能です。外部キーの Status は ENABLED である必要があります。
- 暗号化および複合化用に設定されています。外部キーの KeyUsage には ENCRYPT と DECRYPT を含める必要があります。
- この KMS キーでのみ使用されます。外部キーストアの各 KMS key は、異なる外部キーに関連付けられている必要があります。

  AWS KMS は、外部キーを外部キーストア専用に使用することをお勧めします。この制限により、キーに関する問題の特定と解決が容易になります。
- 外部キーストアの外部キーストアプロキシからアクセスできます。

  外部キーストアプロキシが指定された外部キー ID を使用してキーを検出できない場合、CreateKey オペレーションは失敗します。
- AWS のサービス を使用することで生成される予想トラフィックを処理できます。AWS KMS は、外部キーを毎秒最大 1,800 のリクエストを処理できるように準備することをお勧めします。

外部キーストアで KMS キーを作成する (コンソール)

外部キーストアで KMS キーを作成するには 2 つの方法があります。

- 方法 1 (推奨): 外部キーストアを選択し、その外部キーストアに KMS キーを作成します。
- 方法 2: KMS キーを作成し、それが外部キーストアにあることを示します。

キーを作成する前に外部キーストアを選択する方法 1 を使用する場合、AWS KMS は必要な KMS キープロパティをすべて選択し、外部キーストアの ID を入力します。この方法により、KMS キーの作成時に発生する可能性のあるエラーを回避できます。

> **Note**
>
> エイリアス、説明、またはタグには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

方法 1 (推奨): 外部キーストアで起動する

この方法を使用するには、外部キーストアを選択し、KMS キーを作成します。AWS KMS コンソールは必要なプロパティをすべて選択し、外部キーストアの ID を入力します。この方法により、KMS キーの作成時に発生する可能性のある多くのエラーを回避できます。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[Custom key stores] (カスタムキーストア)、[External key stores] (外部キーストア) の順に選択します。
4. 外部キーストアの名前を選択します。
5. 右上隅にある [Create a KMS key in this key store] (このキーストアのKMS キーを作成する) を選択します。

   外部キーストアが接続されていない場合は、接続するように求められます。接続に失敗した場合は、問題を解決し、外部キーストアに接続してから新しい KMS キーを作成する必要があります。

外部キーストアが接続されている場合は、キーを作成するための [Customer managed keys] (カスタマーマネージドキー) ページにリダイレクトされます。必要な[Key configuration] (キー設定) 値は既に選択されています。また、外部キーストアのカスタムキーストア ID も入力されていますが、変更可能です。

6. 外部キーマネージャーに外部キーのキー ID を入力します。この外部キーは、KMS キーで使用するための要件を満たしている必要があります。キーの作成後にこの値を変更することはできません。

   外部キーに複数の ID がある場合は、外部キーストアプロキシが外部キーの識別に使用するキー ID を入力します。

7. 指定された外部キーストアに KMS キーを作成する予定であることを確認します。
8. [次へ] をクリックします。

   この手順の残りの部分は、標準の KMS キー作成と同じです。

9. エイリアス (必須) と KMS キーの説明 (オプション) を入力します。
10. （オプション）。[Add Tags] (タグの追加) ページで、KMS キーを識別または分類するタグを追加します。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

11. [次へ] をクリックします。
12. [Key administrators] (キー管理者) セクションで、KMS キーを管理できる IAM ユーザーとロールを選択します。詳細については、「KMS キーの管理をキー管理者に許可する」を参照してください。

> **Note**
>
> IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。
> IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

13. (オプション) これらのキー管理者がこの KMS キーを削除できないようにするには、[Allow key administrators to delete this key] (キー管理者がこのキーを削除できるようにする) のチェックボックスをオフにします。

    KMS キーの削除は破壊的で元に戻せないオペレーションであり、暗号文を回復不能にする可能性があります。外部キーマテリアルが存在しても、外部キーストアに対称 KMS キーを再作成することはできません。ただし、KMS キーを削除しても関連付けられた外部キーには影響しません。外部キーストアから KMS キーを削除する方法については、「外部キーストアの KMS キーの削除をスケジュールする」を参照してください。

14. [次へ] をクリックします。

15. [This account] (このアカウント) セクションで、KMS キーを暗号化オペレーションで使用できる、この AWS アカウント の IAM ユーザーとロールを選択します。詳細については、「KMS キーの使用をキーユーザーに許可する」を参照してください。

    > **Note**
    >
    > IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。
    > IAM ベストプラクティスでは、長期の認証情報を持つ IAM ユーザーの使用は推奨されていません。可能な限り、一時的な認証情報を提供する IAM ロールを使用してください。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。

16. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

    > **Note**
    >
    > ユーザーが IAM ポリシーを作成して KMS キーにアクセスすることを、他の AWS アカウント 管理者が許可する必要もあります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

17. [次へ] を選択します。

18. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。

19. 終了したら、[Finish] (完了) を選択し、キーを作成します。

方法 2: カスタマーマネージドキーから開始する

この手順は、AWS KMS キーマテリアルを使用して対称暗号化キーを作成する手順と同じです。ただし、この手順では、外部キーストアのカスタムキーストア ID と外部キーのキー ID を指定します。また、外部キーストアの KMS キーに必要なプロパティ値 (キースペックやキーの使用法など) を指定する必要があります。

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. [Create key] (キーの作成) を選択します。
5. [対称] を選択します。
6. [Key usage] (キーの使用) では、[Encrypt and decrypt] (暗号化および復号化) オプションがすでに選択されています。この設定は変更しないでください。
7. [Advanced options (詳細オプション)] を選択します。
8. [Key material origin] (キーマテリアルのオリジン) で、[External key store] (外部キーストア) を選択します。
9. 指定された外部キーストアに KMS キーを作成する予定であることを確認します。
10. [次へ] をクリックします。
11. 新しい KMS キーの外部キーストアを表す行を選択します。

    接続されていない外部キーストアは選択できません。切断されたキーストアを接続するには、キーストア名を選択し、[Key store actions] (キーストアアクション) で [Connect] (接続) を選択します。詳細については、「外部キーストアを接続する (コンソール)」を参照してください。
12. 外部キーマネージャーに外部キーのキー ID を入力します。この外部キーは、KMS キーで使用するための要件を満たしている必要があります。キーの作成後にこの値を変更することはできません。

    外部キーに複数の ID がある場合は、外部キーストアプロキシが外部キーの識別に使用するキー ID を入力します。
13. [次へ] をクリックします。

    この手順の残りの部分は、標準の KMS キー作成と同じです。
14. KMS キーのエイリアスおよびオプションの説明を入力します。

15. (オプション)。[Add Tags] (タグの追加) ページで、KMS キーを識別または分類するタグを追加します。

    AWS リソースにタグを追加すると、使用量とコストがタグごとに集計されたコスト配分レポートが AWS によって生成されます。タグは、KMS キーへのアクセスの制御にも使用できます。KMS キーのタグ付けについては、キーのタグ付け および AWS KMS の ABAC を参照してください。

16. [次へ] をクリックします。

17. [Key administrators] (キー管理者) セクションで、KMS キーを管理できる IAM ユーザーとロールを選択します。詳細については、「KMS キーの管理をキー管理者に許可する」を参照してください。

    > **Note**
    >
    > IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。

18. (オプション) これらのキー管理者がこの KMS キーを削除できないようにするには、[Allow key administrators to delete this key] (キー管理者がこのキーを削除できるようにする) のチェックボックスをオフにします。

    KMS キーの削除は破壊的で元に戻せないオペレーションであり、暗号文を回復不能にする可能性があります。外部キーマテリアルが存在しても、外部キーストアに対称 KMS キーを再作成することはできません。ただし、KMS キーを削除しても関連付けられた外部キーには影響しません。外部キーストアから KMS キーを削除する方法については、「外部キーストアの KMS キーの削除をスケジュールする」を参照してください。

19. [次へ] をクリックします。

20. [This account] (このアカウント) セクションで、KMS キーを暗号化オペレーションで使用できる、この AWS アカウント の IAM ユーザーとロールを選択します。詳細については、「KMS キーの使用をキーユーザーに許可する」を参照してください。

    > **Note**
    >
    > IAM ポリシーでは、KMS キーを使用するアクセス許可を他の IAM ユーザーおよびロールに付与できます。

21. (オプション) 他の AWS アカウント が暗号化オペレーションにこの KMS キーを使用できるようにします。これを行うには、ページの下部にある [Other AWS アカウント] セクションで、[Add another AWS アカウント] を選択し、外部アカウントの AWS アカウント ID を入力します。複数の外部アカウントを追加するには、この手順を繰り返します。

    > **Note**
    >
    > ユーザーが IAM ポリシーを作成して KMS キーにアクセスすることを、他の AWS アカウント 管理者が許可する必要もあります。詳細については、「他のアカウントのユーザーに KMS キーの使用を許可する」を参照してください。

22. [次へ] を選択します。
23. 選択したキー設定を確認します。戻って、すべての設定を変更することもできます。
24. 終了したら、[Finish] (完了) を選択し、キーを作成します。

手順が成功すると、選択した外部キーストアに新しい KMS キーが表示されます。新しい KMS キーの名前やエイリアスを選択すると、その詳細ページの [Cryptographic configuration] (暗号化設定) タブに、KMS キー ([External key store] (外部キーストア)) のオリジン、カスタムキーストアの名前、ID、タイプ、外部キーの ID、キー使用方法、ステータスが表示されます。手順が失敗すると、失敗を説明するエラーメッセージが表示されます。の場合は、「外部キーストアのトラブルシューティング」を参照してください。

> **Tip**
>
> カスタムキーストアで KMS キーをより簡単に識別できるようにするには、[Customer managed keys] (カスタマーマネージドキー) ページで、[Origin] (オリジン) と [Custom key store ID] (カスタムキーストア ID) 列を表示に追加します。テーブルフィールドを変更するには、ページの右上隅にある歯車アイコンを選択します。詳細については、「KMS キーテーブルをカスタマイズする」を参照してください。

外部キーストアで KMS キーを作成する (AWS KMS API)

外部キーストアで新しい KMS キーを作成するには、 CreateKeyオペレーションを使用します。以下のパラメータは必須です。

- `Origin` の値は `EXTERNAL_KEY_STORE` にする必要があります。

- CustomKeyStoreId パラメータは外部キーストアを識別します。指定された外部キーストアの ConnectionState は、CONNECTED である必要があります。CustomKeyStoreId と ConnectionState を検出するには、DescribeCustomKeyStores オペレーションを使用します。
- XksKeyId パラメータは外部キーを識別します。この外部キーは、KMS キーとの関連付けの要件を満たしている必要があります。

Policy または タグパラメータを使用するなど、CreateKey オペレーションの任意のオプションパラメータを使用することもできます。

> **Note**
>
> Description フィールドまたは Tags フィールドには、機密情報や重要情報を含めないでください。これらのフィールドは、 CloudTrail ログやその他の出力にプレーンテキストで表示される場合があります。

このセクションの例では AWS Command Line Interface (AWS CLI) を使用しますが、サポートされている任意のプログラミング言語を使用することができます。

このコマンド例では、 CreateKeyオペレーションを使用して、外部キーストアに KMS キーを作成します。応答には、KMS キーのプロパティ、外部キーストアの ID、外部キーの ID、使用方法、ステータスが含まれます。これらのフィールドの詳細については、「外部キーストアで KMS キーを表示する」を参照してください。

このコマンドを実行する前に、例のカスタムキーストア ID を有効な ID に置き換えます。

```
$ aws kms create-key --origin EXTERNAL_KEY_STORE --custom-key-store-
id cks-1234567890abcdef0 --xks-key-id bb8562717f809024
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
```

```
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

外部キーストアで KMS キーを表示する

外部キーストアで KMS キーを表示するには、 AWS KMSコンソールまたは DescribeKeyオペレーションを使用します。AWS KMS カスタマーマネージドキーを表示するのと同じ方法を使用できます。基本については、「キーの表示」を参照してください。

AWS KMS コンソールでは、外部キーストア内の KMS キーが、AWS アカウント とリージョンの他のすべてのカスタマーマネージドキーとともに、[Customer managed keys] (カスタマーマネージドキー) ページに表示されます。外部キーストアの KMS キーを識別するには、固有のオリジン値、[External key store] (外部キーストア)、カスタムキーストア ID でフィルタリングします。

詳細については、「外部キーストアを表示する」、「外部キーストアのモニタリング」、および「AWS KMS による AWS CloudTrail API コールのログ記録」を参照してください。

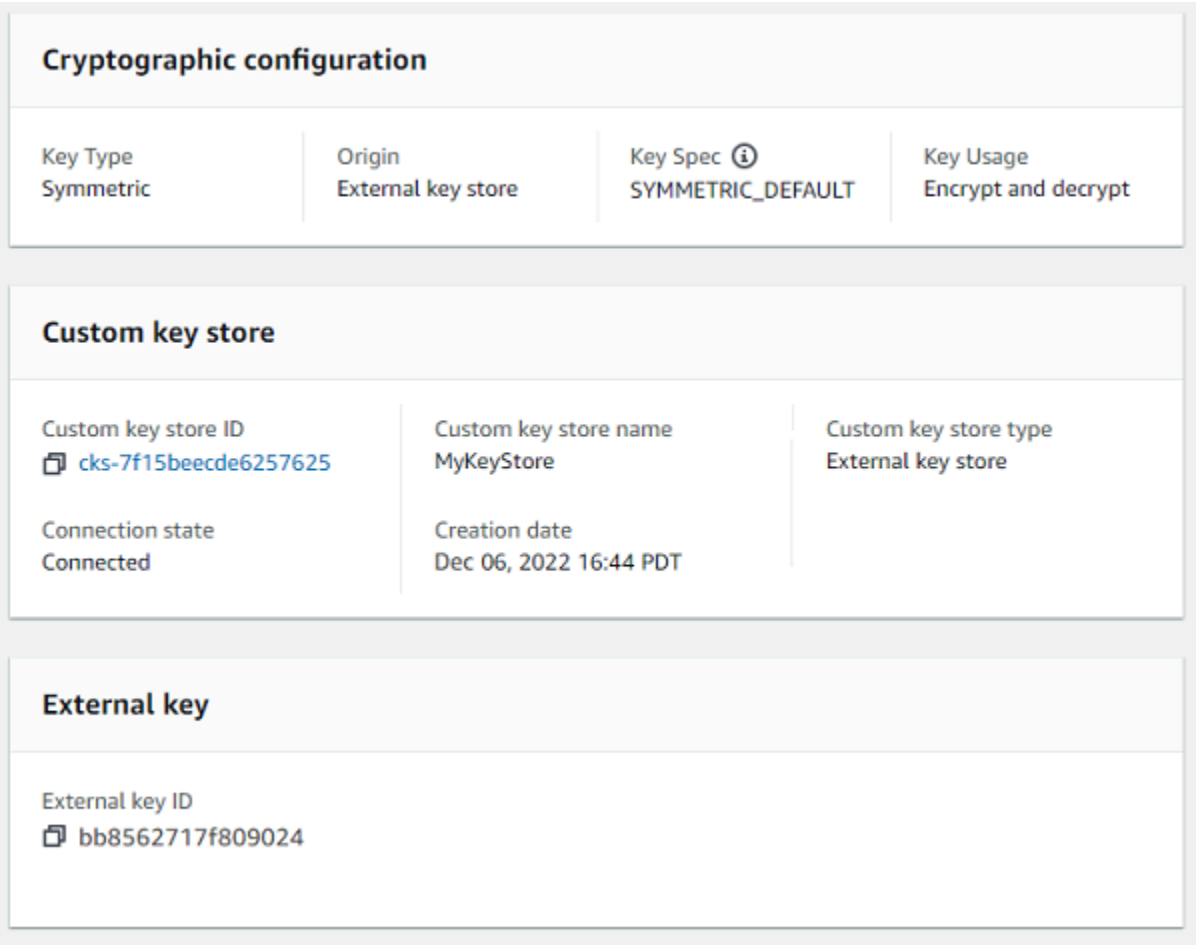
トピック

- 外部キーストアの KMS キーのプロパティ
- 外部キーストアで KMS キーを表示する (コンソール)
- 外部キーストア (AWS KMS API) で KMS キーを表示する

外部キーストアの KMS キーのプロパティ

すべての KMS キーと同様に、外部キーストアの KMS キーには、キー ARN、キースペック、キー用途値があり、加えて、外部キーストアの KMS キー固有のプロパティとプロパティ値もあります。例えば、外部キーストアにあるすべての KMS キーの [Origin] (オリジン) 値は、[External key store] (外部キーストア) です。

外部キーストアの KMS キーの場合、AWS KMS コンソールの [Cryptographic configuration] (暗号化設定) タブには、[Custom key store] (カスタムキーストア) と [External key] (外部キー) の 2 つのセクションが含まれています。



カスタムキーストアのプロパティ

次の値は、暗号化設定タブのカスタムキーストアセクションとDescribeKeyレスポンスに表示されます。これらのプロパティは、AWS CloudHSM キーストアや外部キーストアを含むすべてのカスタムキーストアに適用されます。

カスタムキーストア ID

AWS KMS がカスタムキーストアに割り当てる一意の ID です。

カスタムキーストア名

カスタムキーストア作成時にカスタムキーストアに割り当てるフレンドリ名です。この値はいつでも変更できます。

カスタムキーストアのタイプ

カスタムキーストアのタイプです。有効な値は AWS CloudHSM (AWS_CLOUDHSM) または外部キーストア (EXTERNAL_KEY_STORE) です。カスタムキーストア作成後、タイプを変更することはできません。

作成日

カスタムキーストアが作成された日付です。この日付は、AWS リージョン の現地時間で表示されます。

接続状態

カスタムキーストアがバッキングキーストアに接続されているかどうかを示します。カスタムキーストアがバッキングキーストアに一度も接続されていないか、意図的に切断されていない限り、接続状態は DISCONNECTED です。詳細については、「the section called “接続状態”」を参照してください。

外部キープロパティ

外部キープロパティは、暗号化設定タブの外部キーセクションとDescribeKeyレスポンスのXksKeyConfiguration要素に表示されます。

[External key] (外部キーセクション) は、外部キーストアの KMS キー専用の AWS KMS コンソールに表示されます。このセクションは、KMS キーに関連付けられている外部キーの情報を表示します。外部キーは、外部キーストア内の KMS キーのキーマテリアルとして機能する、AWS の外部にある暗号化キーです。KMS キーを使用して暗号化または復号する際、オペレーションは、指定された外部キーを使用し、外部キーマネージャーによって実行されます。

[External key] (外部キー) セクションには、次の値が表示されます。

外部キー ID

外部キーマネージャーの外部キーの識別子です。これは、外部キーストアプロキシが外部キーを識別するために使用する値です。外部キー ID は KMS キーの作成時に指定します。この ID を変更することはできません。KMS キーの作成に使用した外部キー ID の値が変更または無効になった場合は、KMS キーを削除するようにスケジュールして、正しい外部キー ID 値を使用し、新しい KMS キーを作成する必要があります。

外部キーストアで KMS キーを表示する (コンソール)

カスタムキーストアで KMS キーを表示するには (コンソール)

1. AWS KMS コンソール (https://console.aws.amazon.com/kms) を開きます。
2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 外部キーストアの KMS キーを識別するには、キーテーブルに [Origin] (オリジン) フィールドと [Custom key store ID] (カスタムキーストア ID) フィールドを追加します。外部キーストアの KMS キーの [Origin] (オリジン) 値は、[External key store] (外部キーストア) です。

   右上隅で歯車アイコンを選択し、[Origin] (オリジン)、[Custom key store ID] (カスタムキーストア ID)、[Confirm] (確認) の順に選択します。
5. 外部カスタムキーストアで、KMS キーのエイリアスまたはキー ID を選択します。
6. 外部キーストアの KMS キー固有のプロパティを表示するには、[Cryptographic configuration] (暗号化設定) タブを選択します。外部キーストアの KMS キーの特殊な値は、[Custom key store] (カスタムキーストア) セクションと [External key] (外部キー) セクションに表示されます。

外部キーストア (AWS KMS API) で KMS キーを表示する

外部キーストア (API) で KMS キーを表示するには

、ListKeys、 など、KMS キーに使用する外部キーストアの KMS キーを表示するには、同じ AWS KMS API オペレーションを使用しますDescribeKeyGetKeyPolicy。例えば、次の AWS CLI の `describe-key` オペレーションでは、外部キーストアの KMS キーの特別なフィールドが表示されます。このようなコマンドを実行する前に、サンプル KMS キー ID を有効な値に置き換えます。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
```

```
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

外部キーストアで KMS キーを使用する

外部キーストアで対称暗号化 KMS キーを作成後に、それを以下の暗号化オペレーションで使用できます。

- 暗号化
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

非対称データキーペア GenerateDataKeyPairおよび を生成する対称暗号化オペレーションはGenerateDataKeyPairWithoutPlaintext、カスタムキーストアではサポートされていません。

暗号化コンテキストは、外部キーストアの KMS キーを使用するすべての暗号化オペレーションでサポートされています。通常どおり、暗号化コンテキストを使用することが、AWS KMS が推奨するセキュリティのベストプラクティスです。

リクエストで KMS キーを使用すると、キー ID、キー ARN、エイリアス、エイリアス ARN により KMS キーを識別します。外部キーストアを指定する必要はありません。レスポンスには、対称暗号化 KMS キーについて返されるものと同じフィールドが含まれます。ただし、外部キーストアで KMS キーを使用すると、暗号化および複合化オペレーションは KMS キーに関連付けられた外部キーを使用して、外部キーマネージャーによって実行されます。

外部キーストアの KMS キーで暗号化された暗号文が標準の KMS キーで暗号化された暗号文と同等以上に安全であることを確認するために、AWS KMS は二重暗号化を使用します。データは最初に、AWS KMS キーマテリアルを使用して AWS KMS で暗号化されます。次に、KMS キーの外部キーを使用して、外部キーマネージャーによって暗号化されます。二重に暗号化された暗号文を復号するには、まず KMS キーの外部キーを使用して、外部キーマネージャーによって暗号文を復号します。次に、KMS キーの AWS KMS キーマテリアルを使用して、AWS KMS で復号します。

これを可能にするには、次の条件が必要です。

- KMS キーのキーストアは `Enabled` である必要があります。キーステータスを確認するには、AWS KMSコンソールのカスタマーマネージドキーのステータスフィールドまたはDescribeKeyレスポンスの `KeyState`フィールドを参照してください。
- KMS キーをホストする外部キーストアは、その外部キーストアプロキシに接続する必要があります。つまり、外部キーストアの接続状態は `CONNECTED` である必要があります。

  接続状態は、AWS KMSコンソールの外部キーストアページまたはDescribeCustomKeyStoresレスポンスで確認できます。外部キーストアの接続状態は、AWS KMS コンソールの KMS キーの詳細ページにも表示されます。詳細ページで [Cryptographic configuration] (暗号化設定) タブを選択し、[Custom key store] (カスタムキーストア) セクションの [Connection state] (接続状態) フィールドを確認します。

  接続状態が `DISCONNECTED` の場合、最初に接続する必要があります。接続状態が `FAILED` の場合、問題を解決してから外部キーストアを切断し、接続する必要があります。手順については、「カスタムキーストアの接続と切断」を参照してください。
- 外部キーストアプロキシが外部キーを検出できる必要があります。
- 外部キーを有効にして、暗号化と復号を実行する必要があります。

  外部キーのステータスは、KMS キーの有効化や無効化など、KMS キーのキーステータスの変化とは無関係で、影響を受けません。同様に、外部キーを無効化または削除しても KMS キーのキーステータスは変わりませんが、関連する KMS キーを使用する暗号化オペレーションは失敗します。

これらの条件が満たされていない場合、暗号化オペレーションは失敗し、AWS KMS は `KMSInvalidStateException` 例外を返します。外部キーストアを再接続するか、外部キーマネージャーツールを使用して、外部キーを再設定または修復する必要がある場合があります。その他のヘルプについては、「the section called “外部キーストアのトラブルシューティング”」を参照してください。

外部キーストアで KMS キーを使用する場合、各外部キーストアの KMS キーは、暗号化オペレーションでカスタムキーストアのリクエストクォータを共有することに注意してください。クォータを超えた場合、AWS KMS は ThrottlingException を返します。カスタムキーストアのリクエストクォータの詳細については、「カスタムキーストアのリクエストクォータ」を参照してください。

外部キーストアの KMS キーの削除をスケジュールする

暗号化オペレーションに AWS KMS key を使用する必要がないことが確実な場合は、KMS キーの削除をスケジュールできます。AWS KMS からの KMS キーの削除をスケジュールするのと同じ手順を使用します。外部キーストアから KMS キーを削除しても、キーマテリアルとして使用されていた外部キーには影響しません。

KMS キーのスケジュールされた削除は、必須の待期期間中はキャンセル可能です。ただし、削除された KMS キーは回復できません。同じ外部キーを使用しても、外部キーストアに対称暗号化 KMS キーを再作成することはできません。外部キーストアの各対称 KMS キーには一意の AWS KMS キーマテリアルとメタデータがあるため、対称暗号文を暗号化した AWS KMS キーのみがそれを復号できます。

> **Warning**
>
> KMS キーの削除は、破壊的で潜在的に危険なオペレーションであり、これを実行すると KMS キーで暗号化されたすべてのデータを回復できなくなります。KMS キーの削除をスケジュールする前に、KMS キーの過去の使用状況を調べ、削除保留中に誰かが KMS キーを使用しようとしたときに警告する Amazon CloudWatch アラームを作成します。可能な限り、削除ではなくKMS キーを無効化します。

外部キーストアの KMS キーの削除をスケジュールすると、キーステータスが[Pending deletion] (削除保留中) に変わります。KMS キーは、外部キーストアの切断によって KMS キーが使用できなくなった場合でも、待期期間中を通して[Pending deletion] (削除保留中) ステータスを維持します。これにより、待機期間中はいつでも KMS キーの削除をキャンセルできます。待機期間が終了すると、AWS KMS は AWS KMS から KMS キーを削除します。

外部キーストアの KMS キーの削除をスケジュールすると、その KMS キーは直ちに使用できなくなります (結果整合性の影響を受けます)。ただし、KMS キーで保護されたデータキーで暗号化されているリソースは、KMS キーが (データキーの復号などで) 再度使用されるまで、その影響を受けません。この問題は AWS のサービス に影響します。その多くが、リソースを保護するためにデータキーを使用しています。詳細については、「使用できない KMS キーがデータキーに及ぼす影響」を参照してください。

KMS キーのスケジューリング、キャンセル、削除は、AWS CloudTrail ログでモニタリングできます。

外部キーストアのトラブルシューティング

外部キーストアに関するほとんどの問題の解決方法は、例外が発生するたびに AWS KMS が表示するエラーメッセージ、または外部キーストアを外部キーストアプロキシに接続しようとして失敗したときに AWS KMS が返す接続エラーコードによって示されます。ただし、一部の問題はもう少し複雑です。

外部キーストアの問題を診断する際は、最初に原因を特定します。これにより、対処法の範囲が狭まり、トラブルシューティングがより効率的になります。

- AWS KMS — 外部キーストア設定の値が正しくないなど、AWS KMS 内部に問題がある可能性があります。
- 外部 — 外部キーストアプロキシ、外部キーマネージャー、外部キー、VPC エンドポイントサービスの設定やオペレーションに関する問題など、AWS KMS の外部に問題がある可能性があります。
- ネットワーク — プロキシエンドポイント、ポート、プライベート DNS、ドメインの問題など、接続またはネットワークに問題がある可能性があります。

> **Note**
>
> 外部キーストアで管理オペレーションが失敗すると、複数の異なる例外が生成されます。ただし、AWS KMS 暗号化オペレーションでは、外部キーストアの外部設定または接続状態に関連するすべての障害に対して `KMSInvalidStateException` が返されます 問題を特定するには、添付のエラーメッセージテキストを使用します。
> 接続プロセスが完了する前に、ConnectCustomKeyStoreオペレーションはすぐに成功します。接続プロセスが成功したかどうかを判断するには、外部キーストアの接続状態を表示します。接続プロセスが失敗した場合、AWS KMS は原因を説明し、対処方法を提案する接続エラーコードを返します。

トピック

- 外部キーストアのトラブルシューティングツール
- 設定エラー
- 外部キーストア接続エラー
- レイテンシーとタイムアウトエラー

- 認証情報エラー
- キーステータスエラー
- 復号エラー
- 外部キーエラー
- プロキシの問題
- プロキシの承認に関する問題

外部キーストアのトラブルシューティングツール

AWS KMS では、外部キーストアとそのキーに関する問題を特定し、解決するのに役立ついくつかのツールをご用意しています。これらのツールは、外部キーストアプロキシおよび外部キーマネージャーに付属するツールと組み合わせて使用します。

> **Note**
>
> 外部キーストアプロキシと外部キーマネージャーを使用すると、外部キーストアとその KMS キーを簡単に作成し、維持できます。詳細については、外部ツールのドキュメントを参照してください。

AWS KMS の例外とエラーメッセージ

AWS KMS では、発生した問題に関する詳細なエラーメッセージが表示されます。AWS KMS の例外に関する追加情報は、「AWS Key Management Service API リファレンス」および「AWS SDK」を参照してください。AWS KMS コンソールを使用している場合でも、これらの参考資料が役立つ場合があります。例えば、`CreateCustomKeyStores` オペレーションの場合は、エラーリストを参照してください。

外部キーストアの KMS キーを使用して別の AWS サービスのリソースを保護する場合など、別の AWS サービスで問題が表面化した際に、AWS サービスから問題の特定に役立つ追加情報が提供される場合があります AWS サービスがメッセージを提供しない場合は、KMS キーの使用を記録するCloudTrail ログにエラーメッセージを表示できます。

CloudTrail ログ

AWS KMS コンソールでのアクションを含むすべての AWS KMS API オペレーションは、AWS CloudTrail ログに記録されます。AWS KMS は、成功および失敗したオペレーションのログエントリを記録します。オペレーションが失敗した場合、ログエントリには AWS KMS の例外名

(errorCode) とエラーメッセージ (errorMessage) が含まれます。この情報を使用してエラーを特定し、解決できます。例については「外部キーストアの KMS キーを使用した復号の失敗」を参照してください。

ログエントリにはリクエスト ID も含まれます。リクエストが外部キーストアプロキシに到達した場合は、ログエントリのリクエスト ID を使用して、プロキシログで対応するリクエストを検索できます (プロキシが提供している場合)。

CloudWatch メトリクス

AWS KMS は、レイテンシー、スロットリング、プロキシエラー、外部キーマネージャーのステータス、TLS 証明書の有効期限が切れるまでの日数、プロキシ認証情報の報告された経過時間など、外部キーストアのオペレーションとパフォーマンスに関する詳細な Amazon CloudWatch メトリクスを記録します。これらのメトリクスを使用して、外部キーストアのオペレーション用のデータモデルと、差し迫った問題が発生する前に警告する CloudWatch アラームを開発できます。

> **Important**
>
> AWS KMS では、外部キーストアのメトリクスをモニタリングする CloudWatch アラームを作成することをお勧めします。これらのアラームは、問題が発生する前に問題の初期兆候を警告します。

モニタリンググラフ

AWS KMS は、 AWS KMSコンソールの各外部キーストアの詳細ページに外部キーストア CloudWatch メトリクスのグラフを表示します。グラフのデータを使用して、エラーの原因の特定、差し迫った問題の検出、ベースラインの確立、 CloudWatch アラームのしきい値の調整を行うことができます。モニタリンググラフの解釈とデータ使用方法の詳細については、「外部キーストアのモニタリング」を参照してください。

外部キーストアと KMS キーの表示

AWS KMS は、外部キーストアと KMS キーに関する詳細情報を AWS KMSコンソールの外部キーストアに表示し、 DescribeCustomKeyStores および DescribeKeyオペレーションへのレスポンスに表示します。これらの表示には、外部キーストアの接続状態や KMS キーに関連付けられている外部キーの ID など、トラブルシューティングに使用できる情報を含む、外部キーストアと KMS キーの特別なフィールドが含まれています。詳細については、「外部キーストアを表示する」および「外部キーストアで KMS キーを表示する」を参照してください。

XKS プロキシテストクライアント

AWS KMS では、外部キーストアプロキシが AWS KMS 外部キーストアプロキシ API 仕様に準拠していることを検証する、オープンソースのテストクライアントをご用意しています。このテストクライアントを使用して、外部キーストアプロキシの問題を特定し、解決できます。

設定エラー

外部キーストアを作成するときは、プロキシ認証情報、プロキシ URI エンドポイント、プロキシ URI パス、VPC エンドポイントサービス名など、外部キーストアの設定を構成するプロパティ値を指定します。AWS KMS がプロパティ値のエラーを検出すると、オペレーションは失敗し、エラー値を示すエラーが返されます。

設定に関する問題の多くは、誤った値を修正することで解決できます。無効なプロキシ URI パスやプロキシ認証情報は、外部キーストアを切断せずに修正できます。一意性の要件を含む、これらの値の定義については「前提条件を構成する」を参照してください。これらの値を更新する手順については、「外部キーストアのプロパティの編集」を参照してください。

プロキシ URI パスとプロキシ認証情報値のエラーを避けるために、外部キーストアを作成または更新するときは、プロキシ設定ファイルを AWS KMS コンソールにアップロードします。これは、外部キーストアプロキシまたは外部キーマネージャーから提供される、プロキシ URI パスとプロキシ認証情報値を含む JSON ベースのファイルです。AWS KMS API オペレーションではプロキシ設定ファイルを使用できませんが、ファイル内の値を使用して、プロキシ内の値と一致する API リクエストのパラメータ値を指定できます。

一般的な設定エラー

例外: `CustomKeyStoreInvalidStateException` (`CreateKey`)、`KMSInvalidStateException` (暗号化オペレーション)、`XksProxyInvalidConfigurationException` (管理オペレーション、`CreateKey` を除く)

接続エラーコード:
`XKS_PROXY_INVALID_CONFIGURATION`、`XKS_PROXY_INVALID_TLS_CONFIGURATION`

パブリックエンドポイント接続を備えた外部キーストアの場合、AWS KMS は、外部キーストアの作成および更新時にプロパティ値をテストします。VPC エンドポイントサービス接続を備えた外部キーストアの場合、AWS KMS は、外部キーストアの接続および更新時にプロパティ値をテストします。

> **Note**
>
> 外部キーストアをその外部キーストアのプロキシに接続する試みが失敗しても、非同期の `ConnectCustomKeyStore` オペレーションは成功する場合があります。この場合、例外はありませんが、外部キーストアの接続状態は「失敗」で、エラーメッセージを説明する接続エラーコードが表示されます。詳細については、「外部キーストア接続エラー」を参照してください。

AWS KMS がプロパティ値でエラーを検出すると、オペレーションは失敗し、次のいずれかのエラーメッセージとともに `XksProxyInvalidConfigurationException` が返されます。

URI パスが無効なため、外部キーストアプロキシがリクエストを拒否しました。外部キーストアの URI パスを検証し、必要に応じて更新します。

- プロキシ URI パスは、プロキシ API への AWS KMS リクエストのベースパスです。このパスが間違っていると、プロキシへのリクエストはすべて失敗します。外部キーストアの現在のプロキシ URI パスを表示するには、AWS KMS コンソールまたは `DescribeCustomKeyStores` オペレーションを使用します。正しいプロキシ URI パスを検出するには、外部キーストアプロキシのドキュメントを参照してください。プロキシ URI パス値の修正については、「外部キーストアのプロパティの編集」を参照してください。
- 外部キーストアプロキシのプロキシ URI パスは、外部キーストアプロキシまたは外部キーマネージャーの更新によって変更される可能性があります。これらの変更については、外部キーストアプロキシまたは外部キーマネージャーのドキュメントを参照してください。

`XKS_PROXY_INVALID_TLS_CONFIGURATION`
AWS KMS が、外部キーストアプロキシへの TLS 接続を確立できません。証明書を含む、TLS 設定を検証します。

- すべての外部キーストアプロキシには TLS 証明書が必要です。TLS 証明書は、外部キーストアでサポートされている公開認証機関 (CA) によって発行される必要があります。サポートされている CA のリストについては、「AWS KMS 外部キーストアプロキシ API 仕様」の、「信頼できる認証機関」を参照してください。

- パブリックエンドポイント接続の場合、TLS 証明書のサブジェクト共通名 (CN) が、外部キーストアプロキシのプロキシ URI エンドポイントのドメイン名と一致する必要があります。例えば、パブリックエンドポイントが https://myproxy.xks.example.com の場合、TLS、TLS 証明書の CN は、`myproxy.xks.example.com` または `*.xks.example.com` である必要があります。
- VPC エンドポイントサービス接続の場合、TLS 証明書のサブジェクト共通名 (CN) が、VPC エンドポイントサービスのプライベート DNS 名と一致する必要があります。例えば、プライベート DNS 名が myproxy-private.xks.example.com の場合、TLS 証明書の CN は、`myproxy-private.xks.example.com` または `*.xks.example.com` である必要があります。
- TLS 証明書を有効期限切れにすることはできません。TLS 証明書の有効期限を取得するには、OpenSSL などの SSL ツールを使用します。外部キーストアに関連付けられた TLS 証明書の有効期限をモニタリングするには、 XksProxyCertificateDaysToExpire CloudWatch メトリクスを使用します。TLS 証明書の有効期限までの日数は、AWS KMS コンソールの モニタリングセクションにも表示されます。
- パブリックエンドポイント接続を使用している場合は、SSL テストツールを使用して SSL 設定をテストします。TLS 接続エラーは、証明書チェーンが間違っていることが原因である可能性があります。

VPC エンドポイントサービス接続設定のエラー

例外: `XksProxyVpcEndpointServiceNotFoundException`、`XksProxyVpcEndpointServiceInvalidCo`

一般的な接続の問題に加えて、VPC エンドポイントサービス接続を使用して外部キーストアを作成、接続、更新する際に、次の問題が発生する可能性があります。AWS KMS は、外部キーストアの作成、接続、更新中に、VPC エンドポイントサービス接続を使用して、外部キーストアのプロパティ値をテストします。設定エラーが原因で管理オペレーションが失敗すると、次の例外が生成されます。

XksProxyVpcEndpointServiceNotFound例外

原因は、次のいずれかである可能性があります。

- VPC エンドポイントのサービス名が間違っています。外部キーストアの VPC エンドポイントサービス名が正しく、外部キーストアのプロキシ URI エンドポイント値と一致していることを検証します。VPC エンドポイントサービス名を検索するには、Amazon VPC コンソールまたは DescribeVpcEndpointServices オペレーションを使用します。既存の外部キーストアの VPC エン

ドポイントサービス名とプロキシ URI エンドポイントを検索するには、 AWS KMSコンソールまたは DescribeCustomKeyStoresオペレーションを使用します。詳細については、「外部キーストアを表示する」を参照してください。

- VPC エンドポイントサービスは、外部キーストアとは異なる AWS リージョン に存在する場合があります。VPC エンドポイントサービスと外部キーストアが同じリージョンにあることを検証します。(などのリージョン名の外部名はus-east-1、com.amazonaws.vpce.us-east-1 などの VPC エンドポイントサービス名の一部ですvpce-svc-example。) 外部キーストアの VPC エンドポイントサービス要件のリストについては、「VPC エンドポイントサービス」を参照してください。VPC エンドポイントサービスまたは外部キーストアを別のリージョンに移動することはできません。ただし、VPC エンドポイントサービスと同じリージョンに新しい外部キーストアを作成できます。詳細については、「VPC エンドポイントサービス接続の設定」および「外部キーストアの作成」を参照してください。
- AWS KMS は、VPC エンドポイントサービスのプリンシパルとして許可されていません。VPC エンドポイントサービスの [Allow principals] (許可プリンシパル) リストには、`cks.kms.eu-west-3.amazonaws.com` のような `cks.kms.<region>.amazonaws.com` 値を含める必要があります。この値を追加する手順については、『AWS PrivateLink ガイド』の「許可の管理」を参照してください。

XksProxyVpcEndpointServiceInvalidConfiguration例外

このエラーは、VPC エンドポイントサービスが次の要件のいずれかを満たしていない場合に発生します。

- VPC には、それぞれが異なるアベイラビリティーゾーンに存在する 2 つ以上のプライベートサブネットが必要です。VPC へのサブネットの追加の詳細については、『Amazon VPC ユーザーガイド』の「VPC でのサブネットの作成」を参照してください。
- VPC エンドポイントのサービスタイプには、ゲートウェイロードバランサーではなく、Network Load Balancer を使用する必要があります。
- VPC エンドポイントサービスに承認は不要です (承認が必要は誤りです)。各接続リクエストを手動で承認する必要がある場合、AWS KMS では、VPC エンドポイントサービスを使用して外部キーストアプロキシに接続することはできません。詳細については、『AWS PrivateLink ガイド』の「接続リクエストの承認または拒否」を参照してください。
- VPC エンドポイントサービスには、パブリックドメインのサブドメインであるプライベート DNS 名が必要です。例えば、プライベート DNS 名が `https://myproxy-`

`private.xks.example.com` の場合、`xks.example.com` または `example.com` ドメインにはパブリック DNS サーバーが必要です。VPC エンドポイントサービスのプライベート DNS 名を表示または変更するには、『AWS PrivateLink ガイド』の「VPC エンドポイントサービスの DNS 名の管理」を参照してください。

- プライベート DNS 名ドメインのドメイン検証ステータスは、`verified` である必要があります プライベート DNS 名ドメインの検証ステータスを表示、更新するには、「プライベート DNS 名ドメインの検証」を参照して下さい。必要なテキストレコードを追加した後、更新された検証ステータが表示されるまでに数分かかる場合があります。

> **Note**
>
> プライベート DNS ドメインは、パブリックドメインのサブドメインである場合にのみ検証できます。サブドメインでない場合は、必要な TXT レコードを追加後も、プライベート DNS ドメインの検証ステータスは変わりません。

- VPC エンドポイントサービスのプライベート DNS 名は、外部キーストアのプロキシ URI エンドポイント値と一致する必要があります。VPC エンドポイントサービス接続を備えた外部キーストアの場合、プロキシ URI エンドポイントは `https://` で、その後に VPC エンドポイントサービスのプライベート DNS 名が続く必要があります。プロキシ URI エンドポイント値を表示するには、「外部キーストアを表示する」を参照してください。プロキシ URI エンドポイント値を変更するには、「外部キーストアのプロパティの編集」を参照してください。

外部キーストア接続エラー

外部キーストアを外部キーストアプロキシに接続するプロセスは、完了までに約 5 分かかります。すぐに失敗しない限り、`ConnectCustomKeyStore` オペレーションは HTTP 200 レスポンスと、プロパティを含まない JSON オブジェクトを返します。ただし、この初期レスポンスは接続に成功したことを示していません。外部キーストアが接続されているかどうかを判断するには、その接続状態を参照してください。接続が失敗すると、外部キーストアの接続状態は `FAILED` に変わり、AWS KMS は失敗の原因を説明する接続エラーコードを返します。

> **Note**
>
> カスタムキーストアの接続状態が `FAILED` の場合は、再接続を試みる前にカスタムキーストアを切断する必要があります。`FAILED` 接続ステータスでカスタムキーストアに接続することはできません。

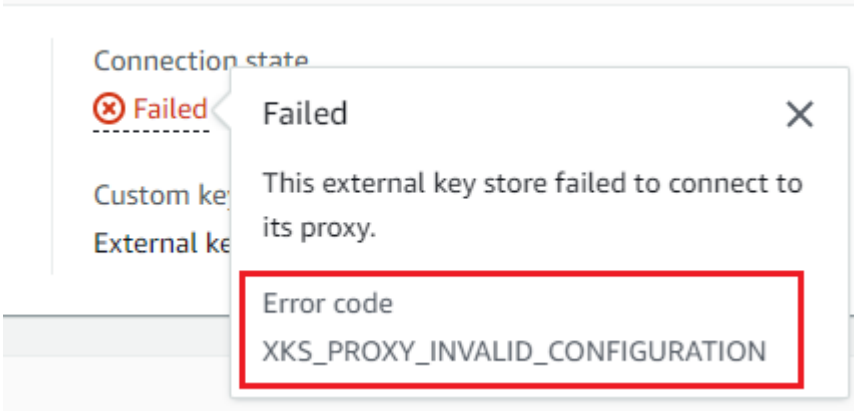
外部キーストアの接続状態を表示するには:

- DescribeCustomKeyStores レスポンスで、 ConnectionState要素の値を表示します。
- AWS KMS コンソールの外部キーストアテーブルに [Connection state] (接続状態) が表示されます。また、各外部キーストアの詳細ページの [General configuration] (一般設定) セクションに、[Connection state] (接続状態) が表示されます。

接続状態が FAILED の場合、接続エラーコードがエラーを説明します。

接続エラーコードを表示するには:

- DescribeCustomKeyStores レスポンスで、 ConnectionErrorCode要素の値を表示します。この要素は、ConnectionState が FAILED の場合にのみ、DescribeCustomKeyStores レスポンスに表示されます。
- AWS KMS コンソールに接続エラーコードを表示するには、外部キーストアの詳細ページで、[Failed] (失敗) 値にカーソルを合わせます。



外部キーストアの接続エラーコード

次の接続エラーコードは、外部キーストアに適用されます

INTERNAL_ERROR

AWS KMS は内部エラーのためにリクエストを完了できませんでした。リクエストを再試行します。ConnectCustomKeyStore リクエストの場合、カスタムキーストアの接続を切断してから接続を再実行します。

`INVALID_CREDENTIALS`

指定された外部キーストアプロキシでは、`XksProxyAuthenticationCredential` 値の一方または両方が無効です。

`NETWORK_ERRORS`

ネットワークエラーにより、AWS KMS はカスタムキーストアをバッキングキーストアに接続できません。

`XKS_PROXY_ACCESS_DENIED`

AWS KMS リクエストは、外部キーストアプロキシへのアクセスを拒否されます。外部キーストアプロキシに承認ルールがある場合は、そのルールがユーザーに代わって、AWS KMS とプロキシの通信を許可していることを検証します。

`XKS_PROXY_INVALID_CONFIGURATION`

設定エラーにより、外部キーストアはプロキシに接続できません。`XksProxyUriPath` の値を検証します。

`XKS_PROXY_INVALID_RESPONSE`

AWS KMS は、外部キーストアプロキシからの応答を解釈できません。この接続エラーコードが繰り返し表示される場合は、外部キーストアプロキシベンダーに通知してください。

`XKS_PROXY_INVALID_TLS_CONFIGURATION`

TLS 設定が無効なため、AWS KMS は外部キーストアプロキシに接続できません。外部キーストアプロキシが TLS 1.2 または 1.3 をサポートしていることを検証します。また、TLS 証明書の有効期限が切れていないこと、`XksProxyUriEndpoint` 値のホスト名と一致していること、信頼できる認証機関リストに含まれる信頼できる認証機関によって署名されていることを検証します。

`XKS_PROXY_NOT_REACHABLE`

AWS KMS は、外部キーストアプロキシと通信できません。`XksProxyUriEndpoint` と `XksProxyUriPath` が正しいことを検証します。外部キーストアプロキシのツールを使用して、プロキシがアクティブで、ネットワーク上で使用可能であることを検証します。また、外部キーマネージャーインスタンスが正しく動作していることを検証します。外部キーマネージャーインスタンスがすべて使用できないとプロキシが報告した場合、接続試行はこの接続エラーコードで失敗します。

`XKS_PROXY_TIMED_OUT`

AWS KMS は外部キーストアプロキシに接続できますが、プロキシは割り当てられた時間内に AWS KMS に応答しません。この接続エラーコードが繰り返し表示される場合は、外部キーストアプロキシベンダーに通知してください。

`XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION`

Amazon VPC エンドポイントサービス設定が、AWS KMS 外部キーストアの要件に準拠していません。

- VPC エンドポイントサービスは、呼び出し元の AWS アカウント のインターフェイスエンドポイント用のエンドポイントサービスである必要があります。
- Network Load Balancer (NLB) は、それぞれが異なるアベイラビリティーゾーンに存在する 2 つ以上のサブネットに接続されている必要があります。
- `Allow principals` リストには、`cks.kms.us-east-1.amazonaws.com` のような、リージョンの AWS KMS サービスプリンシパルである `cks.kms.<region>.amazonaws.com` を含める必要があります。
- 接続リクエストの承認を要求しないでください。
- プライベート DNS 名を付ける必要があります。`VPC_ENDPOINT_SERVICE` 接続の外部キーストアのプライベート DNS 名は、その AWS リージョン 内で一意である必要があります。
- プライベート DNS 名のドメインの検証ステータスは、`verified` である必要があります。
- TLS 証明書は、エンドポイントにアクセス可能なプライベート DNS ホスト名を指定します。

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

AWS KMS は、外部キーストアプロキシとの通信に使用する VPC エンドポイントサービスを検出できません。`XksProxyVpcEndpointServiceName` が正しく、AWS KMS サービスプリンシパルに、Amazon VPC エンドポイントサービスに対するサービスコンシューマー許可があることを検証します。

レイテンシーとタイムアウトエラー

例外: `CustomKeyStoreInvalidStateException` (`CreateKey`)、`KMSInvalidStateException` (暗号化オペレーション)、`XksProxyUriUnreachableException` (管理オペレーション)

接続エラーコード: `XKS_PROXY_NOT_REACHABLE`、`XKS_PROXY_TIMED_OUT`

AWS KMS が250 ミリ秒のタイムアウト間隔内にプロキシに接続できない場合、例外がスローされ、CreateCustomKeyStore および UpdateCustomKeyStore は XksProxyUriUnreachableException を返します。暗号化オペレーションにより、問題を説明するエラーメッセージとともに標準 KMSInvalidStateException が返されます。ConnectCustomKeyStore が失敗した場合、AWS KMS は問題を説明する接続エラーコードを返します。

タイムアウトエラーは一時的な問題のため、リクエストを再試行することで解決できる場合があります。問題が解決しない場合は、外部キーストアプロキシがアクティブでネットワークに接続されていること、そのプロキシ URI エンドポイント、プロキシ URI パス、 VPC エンドポイントのサービス名 (存在する場合) が外部キーストアで正しいことを検証します。また、外部キーマネージャーが外部キーストアの AWS リージョン の近くにあることを確認します。これらの値のいずれかを更新する必要がある場合は、「外部キーストアのプロパティの編集」を参照して下さい。

レイテンシーパターンを追跡するには、 AWS KMS コンソールのモニタリングセクションで、XksProxyLatency CloudWatch メトリクスと平均レイテンシーグラフ (そのメトリクスに基づく) を使用します。外部キーストアプロキシは、レイテンシーとタイムアウトを追跡するログやメトリクスも生成する場合があります。

`XksProxyUriUnreachableException`
AWS KMS は外部キーストアプロキシと通信できません。これは一時的なネットワークの問題である可能性があります。このエラーが繰り返し表示される場合は、外部キーストアプロキシがアクティブでネットワークに接続されていること、エンドポイント URI が外部キーストアで正しいことを検証します。

- 外部キーストアプロキシが、250 ミリ秒のタイムアウト間隔内に AWS KMS プロキシ API リクエストに応答しません。これは、一時的なネットワークの問題か、プロキシのオペレーションまたはパフォーマンスの問題である可能性があります。再試行しても問題が解決しない場合は、外部キーストアプロキシ管理者に通知してください。

多くの場合、レイテンシーやタイムアウトエラーは接続障害として表示されます。ConnectCustomKeyStore オペレーションが失敗すると、外部キーストアの接続状態が に変わり、エラーを説明する接続エラーコードFAILEDAWS KMSが返されます。接続エラーコードのリストとエラー解決方法については、「外部キーストアの接続エラーコード」を参照してください。[All custom key stores] (すべてのカスタムキーストア) と [External key stores] (外部キーストア) の接続

コードリストは、外部キーストアに適用されます。次の接続エラーは、レイテンシーとタイムアウトに関連しています。

`XKS_PROXY_NOT_REACHABLE`

-または-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
AWS KMS は外部キーストアプロキシと通信できません。外部キーストアプロキシがアクティブでネットワークに接続されていること、外部キーストアの URI パスとエンドポイント URI または VPC サービス名が正しいことを検証します。

このエラーは、次の原因によって発生する可能性があります。

- 外部キーストアプロキシがアクティブでないか、ネットワークに接続されていない。
- 外部キーストア設定のプロキシ URI エンドポイント、プロキシ URI パス、または VPC エンドポイントのサービス名 (該当する場合) の値にエラーがある。外部キーストアの設定を表示するには、 DescribeCustomKeyStoresオペレーションを使用するか、 AWS KMSコンソールで外部キーストアの詳細ページを表示します。
- AWS KMS と外部キーストアプロキシ間のネットワークパスにポートエラーなどのネットワーク設定エラーがある可能性があります。AWS KMS はポート 443 で外部キーストアプロキシと通信します。この値は設定できません。
- 外部キーストアプロキシが、すべての外部キーマネージャーインスタンスが であることを (GetHealthStatusレスポンスで) 報告した場合`UNAVAILABLE`、 `ConnectionErrorCode`のでConnectCustomKeyStoreオペレーションは失敗します`XKS_PROXY_NOT_REACHABLE`。ヘルプについては、外部キーマネージャーのドキュメントを参照してください。
- このエラーは、外部キーマネージャーと外部キーストアの AWS リージョン 間の物理的な距離が長いために発生する可能性があります。AWS リージョン と外部キーマネージャー間の ping レイテンシー (ネットワークラウンドトリップ時間 (RTT)) は、35 ミリ秒未満である必要があります。場合により、外部キーマネージャー付近の AWS リージョン に外部キーストアを作成するか、AWS リージョン 付近のデータセンターに外部キーマネージャーを移動する必要があります。

`XKS_PROXY_TIMED_OUT`

-または-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
外部キーストアプロキシが時間内に応答しなかったため、AWS KMS はリクエストを拒否しました。リクエストを再試行します。このエラーが繰り返し表示される場合は、外部キーストアプロキシ管理者に報告してください。

このエラーは、次の原因によって発生する可能性があります。

- このエラーは、外部キーマネージャーと外部キーストアプロキシ間の物理的な距離が長いために発生する可能性があります。可能な場合は、外部キーストアプロキシを外部キーマネージャーの近くに移動します。
- タイムアウトエラーは、プロキシが AWS KMS からのリクエストのボリュームや頻度を処理するよう設計されていない場合に発生する可能性があります。 CloudWatch メトリクスが永続的な問題を示している場合は、外部キーストアプロキシ管理者に通知してください。
- タイムアウトエラーは、外部キーマネージャーと外部キーストアの Amazon VPC 間の接続が正しく機能していない場合に発生する可能性があります。AWS Direct Connect を使用している場合は、VPC と外部キーマネージャーが効果的に通信できることを検証します。問題を解決するには、『AWS Direct Connect ユーザーガイド』の「AWS Direct Connect のトラブルシューティング」を参照してください。

`XKS_PROXY_TIMED_OUT`

-または-

`CustomKeyStoreInvalidStateException` , `KMSInvalidStateException` , `XksProxyUriUnreachableException`
外部キーストアプロキシが割り当てられた時間内にリクエストに応答しません。リクエストを再試行します。このエラーが繰り返し表示される場合は、外部キーストアプロキシ管理者に報告してください。

- このエラーは、外部キーマネージャーと外部キーストアプロキシ間の物理的な距離が長いために発生する可能性があります。可能な場合は、外部キーストアプロキシを外部キーマネージャーの近くに移動します。

認証情報エラー

例外: CustomKeyStoreInvalidStateException (CreateKey)、KMSInvalidStateException (暗号化オペレーション)、XksProxyIncorrectAuthenticationCredentialException (CreateKey 以外の管理オペレーション)

外部キーストアプロキシのAWS KMS の認証情報を確立して維持します。次に、外部キーストアの作成時に、AWS KMS で認証情報の値を指定します。認証情報を変更する場合は、外部キーストアプロキシで変更します。次に、外部キーストアの認証情報を更新します。プロキシが認証情報をローテーションする場合は、外部キーストアの認証情報を更新する必要があります。

外部キーストアプロキシが、外部キーストアのプロキシ認証情報で署名されたリクエストを認証しない場合、その影響はリクエストによって異なります。

- CreateCustomKeyStore および UpdateCustomKeyStore は XksProxyIncorrectAuthenticationCredentialException で失敗します。
- ConnectCustomKeyStore は成功するが接続は失敗します。接続状態は FAILED で、接続エラーコードは INVALID_CREDENTIALS です。詳細については、「外部キーストア接続エラー」を参照してください。
- 暗号化オペレーションでは、外部キーストアのすべての外部設定エラーと接続状態エラーの KMSInvalidStateException が返されます。添付のエラーメッセージは問題を説明しています。

> 外部キーストアプロキシは AWS KMS を認証できなかったため、リクエストを拒否しました。外部キーストアの認証情報を検証し、必要に応じて更新します。

このエラーは、次の原因によって発生する可能性があります。

- 外部キーストアのアクセスキー ID またはシークレットアクセスキーが、外部キーストアプロキシに設定された値と一致しない。

  このエラーを修正するには、外部キーストアのプロキシ認証情報を更新します。この変更は、外部キーストアを切断せずに行うことができます。
- AWS KMS と外部キーストアプロキシ間のリバースプロキシが、SigV4 署名を無効にするような方法で HTTP ヘッダーを操作している可能性があります。このエラーを修正するには、プロキシ管理者に通知してください。

キーステータスエラー

例外: KMSInvalidStateException

KMSInvalidStateException は、カスタムキーストアの KMS キーに、2 つの異なる目的で使用されます。

- CancelKeyDeletion などの管理オペレーションが失敗してこの例外が返された場合は、KMS キーのキーステータスにオペレーションとの互換性がないことを示します。
- カスタムキーストアの KMS キーに対する暗号化オペレーションが KMSInvalidStateException で失敗した場合、KMS キーのキーステータスに問題がある可能性があります。ただし、AWS KMS 暗号化オペレーションでは、外部キーストアのすべての外部設定エラーと接続状態エラーの KMSInvalidStateException が返されます。問題を特定するには、例外に付随するエラーメッセージを使用します。

AWS KMS API オペレーションに必要なキーステータスを確認するには、「AWS KMS キーのキーステータス」を参照してください。KMS キーのキーステータスを確認するには、[Customer managed keys] (カスタマーマネージドキー) ページで、KMS キーの [Status] (ステータス) フィールドを表示します。または、 DescribeKeyオペレーションを使用して、レスポンスで KeyState要素を表示します。詳細については、「キーの表示」を参照してください。

> **Note**
>
> 外部キーストアの KMS キーのキーステータスは、関連付けられている外部キーのステータスを示すものではありません。外部キーステータスに関する情報は、外部キーマネージャーと外部キーストアプロキシツールを参照してください。
> CustomKeyStoreInvalidStateException は、KMS キーのキーステータスではなく、外部キーストアの接続状態を参照します。

KMS キーのキーステータスが Unavailable または PendingDeletion の場合、カスタムストアの KMS キーの暗号化オペレーションは失敗する可能性があります。(無効化されたキーは DisabledException を返します)。

- KMS キーは、AWS KMSコンソールで、または DisableKeyオペレーションを使用して KMS Disabledキーを意図的に無効にした場合にのみ、キーステータスになります。KMS キーが無効になっている間、キーを表示および管理することはできますが、暗号化オペレーションで使用する

ことはできません。この問題を解決するには、キーを有効にします。詳細については、「キーの有効化と無効化」を参照してください。

- KMS キーは、外部キーストアが外部キーストアプロキシから切断されたときに Unavailable キーステータスになります。使用できない KMS キーを修正するには、外部キーストアを再接続します。外部キーストアを再接続すると、外部キーストア内の KMS キーのキーステータスは、Enabled や Disabled などの以前のステータスに自動的に復元されます。

  KMS キーは、削除予定の待機期間中、PendingDeletion キーステータスになります。削除保留中の KMS キーのキーステータスエラーは、そのキーが暗号化に使用されているか、復号に必要であるため、キーを削除するべきではないことを示します。KMS キーを再度有効にするには、スケジュールされた削除をキャンセルしてから、キーを有効にします。詳細については、「キーの削除のスケジュールとキャンセル」を参照してください。

復号エラー

例外: KMSInvalidStateException

外部キーストアの KMS キーを使用した複合オペレーションが失敗すると、AWS KMS は、外部キーストアでのすべての外部設定エラーと接続状態エラーに関して、暗号化オペレーションが使用する標準 KMSInvalidStateException を返します。問題を示すエラーメッセージ。

外部キーマネージャーは、二重暗号化を使用して暗号化された暗号文を復号するために、最初に外部キーを使用して暗号文の外側のレイヤーを復号します。次に、AWS KMS は KMS キー内の AWS KMS キーマテリアルを使用して、暗号文の内層を復号します。無効、または破損した暗号文は、外部キーマネージャーまたは AWS KMS によって拒否される可能性があります。

復号に失敗すると、次のエラーメッセージが KMSInvalidStateException に付随します。このエラーメッセージは、リクエスト内の暗号文またはオプションの暗号化コンテキストに問題があることを示します。

> 指定された暗号文または追加の認証データが破損している、欠損している、または無効であるため、外部キーストアプロキシはリクエストを拒否しました。

- 外部キーストアプロキシまたは外部キーマネージャーが、暗号文またはその暗号化コンテキストが無効であると報告した場合、通常は、AWS KMS に送信される Decrypt リクエストの暗号文または暗号化コンテキストに問題があることを示します。Decrypt オペレーションの場合、AWS

KMS は、Decrypt リクエストで受信したものと同じ暗号文と暗号化コンテキストをプロキシに送信します。

このエラーは、ビットの反転など、転送中のネットワークの問題が原因である可能性があります。Decrypt リクエストを再試行します。問題が解決しない場合は、暗号文が変更されていないこと、または破損していないことを検証します。また、AWS KMS への Decrypt リクエストの暗号化コンテキストが、データを暗号化したリクエストの暗号化コンテキストと一致することを検証します。

外部キーストアプロキシが復号のために送信した暗号文、または暗号化コンテキストが、破損しているか、欠損しているか、無効になっています。

- AWS KMS がプロキシから受信した暗号文を拒否した場合は、外部キーマネージャーまたはプロキシが、無効な、または破損した暗号文を AWS KMS に返したことを示します。

  このエラーは、ビットの反転など、転送中のネットワークの問題が原因である可能性があります。Decrypt リクエストを再試行します。問題が解決しない場合は、外部キーマネージャーが正しく動作していること、外部キーストアプロキシが外部キーマネージャーから受信した暗号文を AWS KMS に返す前に変更していないことを検証します。

外部キーエラー

外部キーは、KMS キーの外部キーマテリアルとして機能する外部キーマネージャーの暗号化キーです。AWS KMS は外部キーには直接アクセスできません。外部キーマネージャーに、(外部キーストアプロキシ経由で) 外部キーを使用してデータを暗号化するか、暗号文を復号するように要求する必要があります。

外部キーストアで KMS キーを作成する際に、外部キーマネージャーで外部キーの ID を指定します。KMS キー作成後に外部キー ID を変更することはできません。KMS キーに関する問題を防ぐために、CreateKey オペレーションは外部キーストアプロキシに、外部キーの ID と設定の検証を要求します。外部キーが KMS キーでの使用要件を満たしていない場合、CreateKey オペレーションは失敗し、問題を特定する例外とエラーメッセージが表示されます。

ただし、KMS キー作成後に問題が発生する可能性があります。外部キーの問題によって暗号化オペレーションが失敗すると、オペレーションは失敗し、問題を示すエラーメッセージとともに KMSInvalidStateException が返されます。

CreateKey 外部キーの エラー

例外:
XksKeyAlreadyInUseException、XksKeyNotFoundException、XksKeyInvalidConfigurationE

CreateKey オペレーションは、外部キー ID (コンソール) または XksKeyId (API) パラメータで指定した外部キーの ID とプロパティの検証を試みます。このプラクティスは、KMS キーで外部キーを使用する前にエラーを早期に検出することを目的としています。

外部キー使用中

外部キーストアの各 KMS キーは、異なる外部キーを使用する必要があります。CreateKey が KMS キーの外部キー ID (XksKeyId) が外部キーストアで一意ではないことを認識すると、 で失敗しますXksKeyAlreadyInUseException。

同じ外部キーに複数の ID を使用する場合、CreateKey は重複しているものを認識しません。ただし、同じ外部キーの KMS キーは、AWS KMS キーマテリアルとメタデータが異なるため、相互運用できません。

外部キーを検出できない

外部キーストアプロキシが KMS キーの外部キー ID (XksKeyId) を使用して外部キーを検出できないことを報告すると、CreateKeyオペレーションは失敗し、次のエラーメッセージXksKeyNotFoundExceptionとともに が返されます。

> 外部キーを検出できなかったため、外部キーストアプロキシはリクエストを拒否しました。

このエラーは、次の原因によって発生する可能性があります。

- KMS キーの外部キー (XksKeyId) ID が無効である可能性があります。外部キープロキシが外部キーを識別するために使用する ID を検出するには、外部キーストアプロキシまたは外部キーマネージャーのドキュメントを参照してください。
- 外部キーが外部キーマネージャーから削除された可能性があります。調査するには、外部キーマネージャーツールを使用します。外部キーが完全に削除された場合は、KMS キーで別の外部キーを使用します。外部キーのリストまたは要件については、「外部キーストアの KMS キーの要件」を参照してください。

外部キー要件が満たされていない

外部キーストアプロキシが、外部キーが KMS キーでの使用要件を満たしていないことを報告すると、CreateKey オペレーションは失敗し、次のいずれかのエラーメッセージを含む XksKeyInvalidConfigurationException が返されます。

外部キーのキースペックは AES_256 である必要があります。指定された外部キーのキースペックは *<key-spec>* です。

- 外部キーは、AES_256 のキースペックをもつ 256 ビット対称暗号化キーである必要があります。指定された外部キーが別のタイプである場合は、この要件を満たす外部キー ID を指定します。

外部キーのステータスは ENABLED である必要があります。指定された外部キーのステータスは *<status>* です。

- 外部キーは、外部キーマネージャーで有効にする必要があります。指定した外部キーが有効になっていない場合は、外部キーマネージャーツールを使用して有効にするか、有効な外部キーを指定します。

外部キーのキー用途には、[ENCRYPT] と [DECRYPT] を含める必要があります。指定された外部キーのキー用途は <*key-usage* > です。

- 外部キーは、外部キーマネージャーで暗号化と復号を行うように設定する必要があります。指定された外部キーにこれらのオペレーションが含まれていない場合は、外部キーマネージャーツールを使用してオペレーションを変更するか、別の外部キーを指定します。

外部キーの暗号化オペレーションエラー

例外: KMSInvalidStateException

外部キーストアプロキシが KMS キーに関連付けられた外部キーを検出できない場合、または外部キーが KMS キーでの使用要件を満たしていない場合、暗号化オペレーションは失敗します。

暗号化オペレーション中に検出された外部キーの問題は、KMS キー作成前に検出された外部キーの問題よりも解決が困難です。KMS キー作成後に外部キー ID を変更することはできません。KMS

キーでまだデータが暗号化されていない場合は、KMS キーを削除し、別の外部キー ID を使用して新しい KMS キーを作成できます。ただし、KMS キーで生成された暗号文は、キーメタデータや AWS KMS キーマテリアルが異なるため、外部キーが同じであっても他の KMS キーでは復号できません。代わりに、外部キーマネージャーツールを使用して、可能な限り外部キーの問題を解決してください。

外部キーストアプロキシが外部キーに関する問題を報告すると、暗号化オペレーションは、問題を特定するエラーメッセージとともに `KMSInvalidStateException` を返します。

外部キーを検出できない

外部キーストアプロキシが KMS キーの外部キー ID (XksKeyId) を使用して外部キーを検出できないことを報告すると、暗号化オペレーションは次のエラーメッセージ`KMSInvalidStateException`を含む を返します。

> 外部キーを検出できなかったため、外部キーストアプロキシはリクエストを拒否しました。

このエラーは、次の原因によって発生する可能性があります。

- KMS キーの外部キー (`XksKeyId`) ID が無効になりました。

  KMS キーに関連付けられている外部キー ID を検出するには、KMS キーの詳細を参照してください。外部キープロキシが外部キーを識別するために使用する ID を検出するには、外部キーストアプロキシまたは外部キーマネージャーのドキュメントを参照してください。

  AWS KMS は、外部キーストアに KMS キーを作成する際に、外部キー ID を検証します。ただし、特に外部キー ID 値がエイリアスまたは可変名の場合は、ID が無効になる可能性があります。既存の KMS キーに関連付けられている外部キー ID を変更することはできません。KMS キーで暗号化された暗号文を復号するには、外部キーを既存の外部キー ID に再度、関連付ける必要があります。

  KMS キーを使用してデータを暗号化していない場合は、有効な外部キー ID を使用して新しい KMS キーを作成できます。ただし、KMS キーを使用して暗号文を生成した場合は、同じ外部キーを使用しても、他の KMS キーを使用して暗号文を復号することはできません。
- 外部キーが外部キーマネージャーから削除された可能性があります。調査するには、外部キーマネージャーツールを使用します。可能な場合は、外部キーマネージャーのコピーまたはバックアップから、キーマテリアルの復元を試みてください。外部キーが完全に削除された場合、関連する KMS キーで暗号化された暗号文は回復できません。

外部キー設定エラー

外部キーストアプロキシが、外部キーが KMS キーでの使用要件を満たしていないことを報告すると、暗号化オペレーションは、次のいずれかのエラーメッセージを含む KMSInvalidStateException を返します。

外部キーがリクエストされたオペレーションをサポートしていないため、外部キーストアプロキシは要求を拒否しました。

- 外部キーは、暗号化と復号の両方をサポートしている必要があります。キーの用途に暗号化と復号が含まれていない場合は、外部キーマネージャーツールを使用して、キーの用途を変更します。

外部キーが外部キーマネージャーで有効になっていないため、外部キーストアプロキシはリクエストを拒否しました。

- 外部キーは、外部キーマネージャーで使用するために、有効にして使用可能にする必要があります。外部キーのステータスが [Enabled] でない場合は、外部キーマネージャーツールを使用して有効にします。

プロキシの問題

例外:

CustomKeyStoreInvalidStateException (CreateKey)、KMSInvalidStateException (暗号化オペレーション)、UnsupportedOperationException、XksProxyUriUnreachableException、XksProxyInval (CreateKey 以外の管理オペレーション)

外部キーストアプロキシは、AWS KMS と外部キーマネージャー間のすべての通信を仲介します。外部キーストアプロキシは、一般的な AWS KMS リクエストを外部キーマネージャーが理解できる形式に変換します。外部キーストアプロキシが AWS KMS 外部キーストアプロキシ API 仕様に準拠していない場合、正しく動作していない場合、AWS KMS と通信できない場合、外部キーストアで KMS キーを作成または使用することはできません。

外部キーストアプロキシは外部キーストアアーキテクチャで重要な役割を果たすため、多くのエラーに外部キーストアプロキシが記載されますが、これらの問題は、外部キーマネージャーまたは外部キーに起因する可能性があります。

このセクションの問題は、外部キーストアプロキシの設計または運用に関する問題と関連しています。これらの問題を解決するには、プロキシソフトウェアを変更する必要がある場合があります。プロキシ管理者に相談してください。AWS KMS は、プロキシの問題の診断に役立つように、外部キーストアプロキシが AWS KMS 外部キーストアプロキシ API 仕様に準拠していることを検証する、オープンソースのテストクライアントである XKS プロキシテキストクライアントを提供しています。

`CustomKeyStoreInvalidStateException` 、`KMSInvalidStateException` 、または `XksProxyUriUnreachableException`
外部キーストアプロキシが異常な状態です。このメッセージが繰り返し表示される場合は、外部キーストアプロキシ管理者に通知してください。

- このエラーは、外部キーストアプロキシの運用上の問題またはソフトウェアエラーを示している可能性があります。各エラーを生成した AWS KMS API オペレーションの CloudTrail ログエントリを見つけることができます。このエラーは、オペレーションを再試行することで解決する場合があります。ただし、それでも解決しない場合は、外部キーストアプロキシ管理者に通知してください。
- 外部キーストアプロキシが、すべての外部キーマネージャーインスタンスが であると (GetHealthStatusレスポンスで) 報告した場合UNAVAILABLE、外部キーストアの作成または更新の試行はこの例外で失敗します。このエラーが解決されない場合は、外部キーマネージャーのドキュメントを参照してください。

`CustomKeyStoreInvalidStateException` 、`KMSInvalidStateException` 、または `XksProxyInvalidResponseException`
AWS KMS は、外部キーストアプロキシからの応答を解釈できません。このエラーが繰り返し表示される場合は、外部キーストアプロキシ管理者に相談してください。

- AWS KMS が解析または解釈できない未定義の応答をプロキシが返すと、AWS KMS オペレーションでこの例外が生成されます。このエラーは、一時的な外部の問題や散発的なネットワークエラーが原因で発生することがあります。ただし、問題が解決しない場合は、外部キーストアプロキシ

が、AWS KMS 外部キーストアプロキシ API 仕様に準拠していない可能性があります。外部キーストア管理者またはベンダーに通知してください。

`CustomKeyStoreInvalidStateException` 、`KMSInvalidStateException` 、または `UnsupportedOperationException`

リクエストされた暗号化オペレーションをサポートしていないため、外部キーストアプロキシはリクエストを拒否しました。

- 外部キーストアプロキシは、AWS KMS 外部キーストアプロキシ API 仕様で定義されているすべてのプロキシ API をサポートしている必要があります。このエラーは、プロキシがリクエストに関連するオペレーションをサポートしていないことを示します。外部キーストア管理者またはベンダーに通知してください。

プロキシの承認に関する問題

例外: `CustomKeyStoreInvalidStateException`、`KMSInvalidStateException`

外部キーストアプロキシの中には、外部キーを使用するための認証要件を実装しているものがあります。外部キーストアプロキシーは、特定のユーザーが任意の条件下で任意のオペレーションをリクエストすることを許可する承認スキームを設計および実装することを許可されていますが、必須ではありません。例えば、プロキシでは、ユーザーに任意の外部キーによる暗号化を許可しても、その外部キーによる復号は許可しない場合があります。詳細については、「外部キーストアプロキシ認証 (オプション)」を参照してください。

プロキシの承認は、AWS KMS がプロキシへのリクエストに含めるメタデータに基づいています。`awsSourceVpc` および `awsSourceVpce` フィールドは、リクエストが VPC エンドポイントからのものである場合と、呼び出し側が KMS キーと同じアカウントにある場合にのみ、メタデータに含まれます。

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
```

```
    "kmsViaService": string // optional
}
```

承認の失敗によってプロキシがリクエストを拒否すると、関連する AWS KMS オペレーションは失敗します。その結果、CreateKeyは CustomKeyStoreInvalidStateException を返し、AWS KMS 暗号化オペレーションは KMSInvalidStateException を返します。どちらも以下のエラーメッセージを使用します。

> 外部キーストアプロキシはオペレーションへのアクセスを拒否しました。ユーザーと外部キーの両方がこのオペレーションを承認していることを検証し、リクエストを再試行します。

- エラーを解決するには、外部キーマネージャーまたは外部キーストアプロキシツールを使用して、承認が失敗した理由を特定します。次に、未承認リクエストの原因となった手順を更新するか、外部キーストアプロキシツールを使用して、承認ポリシーを更新します。AWS KMS ではこのエラーを解決できません。

キータイプリファレンス

AWS KMS は、さまざまなタイプの KMS キーに対して異なる機能をサポートします。例えば、対称データキーと非対称データキーペアの生成には、対称暗号化 KMS キーのみを使用できます。また、キーマテリアルのインポートと自動キーローテーションは対称暗号化 KMS キーのみでサポートされ、カスタムキーストアでは対称暗号化 KMS キーのみを作成できます。

このリファレンスには 2 つの表が含まれています。

- キータイプの表には、対称 KMS キー、非対称 KMS キー、HMAC KMS キーで有効な AWS KMS オペレーションが記載されています。
- 特殊な機能の表には、マルチリージョン KMS キー、インポートしたキーマテリアルを含む KMS キー、カスタムキーストアの KMS キーで有効な AWS KMS オペレーションが記載されています。

キータイプの表

この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
CancelKeyDeletion	✓	✓	✓	✓
CreateAlias	✓	✓	✓	✓
CreateGrant	✓	✓	✓	✓
CreateKey	✓	✓	✓	✓
Decrypt	✓	⊗	✓	⊗
DeleteAlias	✓	✓	✓	✓
DeleteImportedKeyMaterial インポートしたキーマテリアルを含む KMS キーのみで有効です (`Origin` は `EXTERNAL`)。	✓	✓	✓	✓
DescribeKey	✓	✓	✓	✓
DisableKey	✓	✓	✓	✓
DisableKeyRotation	✓ AWS KMS キーマテリアルを	⊗	⊗	⊗

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
	含む KMS キーのみで有効です (Origin は AWS_KMS)。			
EnableKey	✓	✓	✓	✓
EnableKeyRotation	✓ AWS KMS キーマテリアルを含む KMS キーのみで有効です (Origin は AWS_KMS)。	⊗	⊗	⊗
暗号化	✓	⊗	✓	⊗
GenerateDataKey	✓	⊗	⊗	⊗

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
GenerateDataKeyPair 対称暗号化 KMS キーによって保護される非対称データキーペアを生成します。	✓ カスタムキーストアの KMS キーでは有効ではありません。	⊗	⊗	⊗
GenerateDataKeyPairWithoutPlaintext 対称暗号化 KMS キーによって保護される非対称データキーペアを生成します。	✓ カスタムキーストアの KMS キーでは有効ではありません。	⊗	⊗	⊗
GenerateDataKeyWithoutPlaintext	✓	⊗	⊗	⊗
GenerateMac	⊗	✓	⊗	⊗
GetKeyPolicy	✓	✓	✓	✓

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
GetKeyRotationStatus	✓	✓ (KeyRotationEnabled は常に false になります。)	✓ (KeyRotationEnabled は常に false になります。)	✓ (KeyRotationEnabled は常に false になります。)
GetParametersForImport インポートしたキーマテリアルを含む KMS キーのみで有効です (Origin は EXTERNAL)。	✓	✓	✓	✓
GetPublicKey	⊗	⊗	✓	✓
ImportKeyMaterial インポートしたキーマテリアルを含む KMS キーのみで有効です (Origin は EXTERNAL)。	✓	✓	✓	✓
ListAliases	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
ListResourceTags	✓	✓	✓	✓
ListRetirableGrants	✓	✓	✓	✓
PutKeyPolicy	✓	✓	✓	✓
ReEncrypt	✓	⊗	✓	⊗
ReplicateKey – マルチリージョンキーでのみ有効	✓	✓	✓	✓
RetireGrant	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓	✓
Sign	⊗	⊗	⊗	✓
TagResource	✓	✓	✓	✓
UntagResource	✓	✓	✓	✓

AWS KMS API オペレーション	対称暗号化 KMS キー	HMAC KMS キー	非対称 KMS キー (ENCRYPT_DECRYPT)	非対称 KMS キー (SIGN_VERIFY)
UpdateAlias 現在の KMS キーと新しい KMS キーは同じタイプでなければならず (両方とも対称か両方とも非対称、もしくは両方ともHMAC)、キーの用途が同じでなければなりません。	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	✓	✓
UpdateReplicaRegion – マルチリージョンキーでのみ有効	✓	✓	✓	✓
検証	⊗	⊗	⊗	✓
VerifyMac	⊗	✓	⊗	⊗

特殊な機能の表

この表には、特殊な用途を持つキーの各タイプでサポートされている AWS KMS APIオペレーションが記されています。

この表を見るときは、次の相互関係に注意します。

- マルチリージョンキー:
  - マルチリージョンキーは、対称暗号化 KMS キー、非対称 KMS キー、HMAC KMS キー、インポートされたキーマテリアルを含む KMS キーのいずれかです。

  - カスタムキーストアでマルチリージョンキーを作成することはできません。
- インポートされたキーマテリアル
  - 対称暗号化 KMS キー、非対称 KMS キー、HMAC KMS キーのキーマテリアルをインポートできます。
  - インポートしたキーマテリアルのマルチリージョンキーを作成できます。
  - カスタムキーストアでは、インポートしたキーマテリアルを持つキーは作成できません。
  - 自動キーローテーション (EnableKeyRotation、DisableKeyRotation) は、インポートされたキーマテリアルを持つ KMS キーではサポートされません。
- カスタムキーストア
  - カスタムキーストアは、対称暗号化 KMS キーのみをサポートします。
  - 非対称キーペア (GenerateDataKeyPair、GenerateDataKeyPairWithoutPlaintext) の対称オペレーションは、カスタムキーストアの KMS キーではサポートされていません。
  - 自動キー回転機能 (EnableKeyRotation、DisableKeyRotation) は、カスタムキーストアの KMS キーではサポートされていません。
  - カスタムキーストアでマルチリージョンキーを作成することはできません。

この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey マルチリージョンプライマリキー、キーマテリアルがインポートされた KMS キー、カスタムキーストアの KMS キーのいずれかを作成するとき	✓	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
は CreateKey を使用できます。マルチリージョンのレプリカキーを作成するには、ReplicateKey リソースを使用します。			
Decrypt	✓ KeyUsage が ENCRYPT_DECRYPT の場合のみ有効です。	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ インポートされたキーマテリアルを含むキーのみで有効です (Origin は EXTERNAL)。	✓	⊗
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
DisableKeyRotation	✓ AWS KMS キーマテリアルを含む対称暗号化キーのみで有効です (`Origin` は `AWS_KMS`)。	⊗	⊗
EnableKey	✓ 対称暗号化 KMS キーのみで有効です。	✓	✓
EnableKeyRotation	✓ AWS KMS キーマテリアルを含む対称暗号化キーのみで有効です (`Origin` は `AWS_KMS`)。	⊗	⊗
暗号化	✓ `KeyUsage` が `ENCRYPT_DECRYPT` の場合のみ有効です。	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
GenerateDataKey	✓ 対称暗号化 KMS キーのみで有効です。	✓	✓
GenerateDataKeyPair	✓ 対称暗号化 KMS キーのみで有効です。	✓	⊗
GenerateDataKeyPairWithoutPlaintext	✓ 対称暗号化 KMS キーのみで有効です。	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ 対称暗号化 KMS キーのみで有効です。	✓	✓
GenerateMac HMAC KMS キーのみで有効です。	✓	✓	⊗
GetKeyPolicy	✓	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
GetKeyRotationStatus	✓	✓ (KeyRotationEnabled は常に false になります。)	⊗
GetParametersForImport	✓ インポートされたキーマテリアルを含むキーのみで有効です (Origin は EXTERNAL)。	✓	⊗
GetPublicKey 非対称 KMS キーのみで有効です。	✓	✓	⊗
ImportKeyMaterial	✓ インポートされたキーマテリアルを含むキーのみで有効です (Origin は EXTERNAL)。	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
ListKeyPolicies	✓	✓	✓
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ KeyUsage が ENCRYPT_DECRYPT の場合のみ有効です。	✓	✓
ReplicateKey	✓ マルチリージョンプライマリキーのみで有効です。	✓ マルチリージョンプライマリキーのみで有効です。	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
ScheduleKeyDeletion	✓	✓	✓

AWS KMS API オペレーション	マルチリージョンキー	インポートされたキーマテリアル	カスタムキーストアの KMS キー
Sign KeyUsage が SIGN_VERIFY の場合のみ有効です。	✓	✓	⊗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓
UpdateAlias - 現在の KMS キーと新しい KMS キーは同じタイプでなければならず (両方とも対称か両方とも非対称、もしくは両方とも HMAC)、キーの用途が同じでなければなりません。	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ マルチリージョンキーでのみ有効です。	⊗
検証 KeyUsage が SIGN_VERIFY の場合にのみ有効です。	✓	✓	⊗
VerifyMac HMAC KMS キーのみで有効です。	✓	✓	⊗

AWS Key Management Service 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 のお客様は、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 とお客様の間の共有責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS サービスを実行するインフラストラクチャを保護する責任を負います。また、AWS は、安全に使用できるサービスを提供します。AWS コンプライアンスプログラムの一環として、サードパーティー監査者が定期的にセキュリティの有効性をテストおよび検証します。AWS Key Management Service (AWS KMS) に適用するコンプライアンスプログラムの詳細については、コンプライアンスプログラムによる AWS 対象範囲内のサービス」および「」を参照してください。
- クラウド内のセキュリティ – お客様の責任は使用する AWS のサービスによって決定されます。AWS KMS では、お客様は、AWS KMS keys の設定および使用に加えて、データの機密性、会社の要件、適用される法律や規制などのその他の要素についても責任を負います。

このドキュメントは、AWS Key Management Service を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。ここでは、セキュリティとコンプライアンスの目標を満たすように AWS KMS を設定する方法を説明します。

トピック

AWS Key Management Service でのデータ保護

AWS Key Management Service は暗号化キーを保存して保護し、高可用性を実現すると同時に、強力で柔軟なアクセス制御を提供します。

トピック

キーマテリアルの保護

デフォルトでは、AWS KMS は KMS キーの暗号化キーマテリアルを生成し、保護します。さらに、AWS KMS は AWS KMS の外部で作成および保護されるキーマテリアルのオプションもあります。KMS キーとキーマテリアルの技術的な詳細については、「AWS Key Management Service 暗号化の詳細」を参照してください。

AWS KMS で生成されるキーマテリアルの保護

KMS キーを作成すると、デフォルトで AWS KMS がその KMS キーの暗号化マテリアルを生成し、保護します。

KMS キーのキーマテリアルを保護するために、AWS KMS は、FIPS 140-2 セキュリティレベル 3 検証済みハードウェアセキュリティモジュール (HSM) の分散フリートを使用します。各 AWS KMS HSM は、AWS KMS のセキュリティおよびスケーラビリティ要件を満たす専用の暗号化機能を提供するように設計された専用のスタンドアロンのハードウェアアプライアンスです。(中国リージョンで、AWS KMS が使用する HSM は、OSCCA によって証明され、関連するすべての中国の規制に準拠していますが、FIPS 140-2 暗号化モジュール検証プログラムでは検証されていません)。

KMS キーのキーマテリアルは、HSM で生成されるときにデフォルトで暗号化されます。キーマテリアルは HSM の揮発性メモリ内でのみ、暗号化オペレーションで使用するのにかかる数ミリ秒の間だけ復号化されます。キーマテリアルがアクティブに使用されていない場合は常に、HSM 内で暗号化され、耐久性の高い (99.999999999%)、低レイテンシーの永続ストレージに転送され、そこで HSM とは別の場所に保管されます。プレーンテキストのキーマテリアルは、HSM セキュリティ境界を離れることはありません。また、ディスクに書き込まれることも、ストレージメディアに保持されることもありません。(唯一の例外は、非対称キーペアのパブリックキーです。これはシークレットではありません)。

AWS は、任意の AWS のサービス の任意のタイプのプレーンテキストの暗号化キーマテリアルとの手動による介入のない基本のセキュリティ原則としてアサートされます。AWS のサービス オペレーターを含め、誰かがプレーンテキストのキーマテリアルを表示、アクセス、またはエクスポートするメカニズムはありません。この原則は、壊滅的な障害やディザスタリカバリ中にも適用されます。AWS KMS のプレーンテキストカスタマーキーマテリアルは、お客様またはその代理人によるサービスへの許可されたリクエストに応じてのみ、AWS KMS FIPS 検証された HSM 内の暗号化オペレーションに使用されます。

カスタマーマネージドキーの場合、キーを作成する AWS アカウント は、そのキーの唯一で譲渡できない所有者です。所有しているアカウントは、キーへのアクセスを制御する権限付与ポリシーを完全かつ排他的に制御できます。AWS マネージドキー の場合、AWS アカウント は AWS のサービスに対するリクエストを承認する IAM ポリシーを完全に制御できます。

AWS KMS の外部で生成されるキーマテリアルの保護

AWS KMS は AWS KMS で生成されるキーマテリアルの代替を提供します。

オプションの AWS KMS 機能であるカスタムキーストアを使用すると、AWS KMS の外部で生成され、使用されるキーマテリアルによってバックアップされる KMS キーを作成できます。AWS CloudHSM キーストアの KMS キーは、ユーザーが制御する AWS CloudHSM ハードウェアセキュリティモジュールのキーによってバックアップされます。これらの HSM は、FIPS 140-2 セキュリティレベル 3で認定されています。外部キーストアの KMS キーは、AWS の外部で制御され、管理される外部キーマネージャーのキーによってバックアップされます (プライベートデータセンターの物理 HSM など)。

もう 1 つのオプション機能により、KMS キーのキーマテリアルをインポートできます。AWS KMS への転送中にインポートされたキーマテリアルを保護するには、AWS KMS HSM で生成される RSA キーペアのパブリックキーを使用してキーマテリアルを暗号化します。インポートされたキーマテリアルは AWS KMS HSM で復号され、HSM の対称キーで再暗号化されます。すべての AWS KMS キーマテリアルと同様に、プレーンテキストでインポートされるキーマテリアルは HSM を未暗号化状態のままにしません。ただし、キーマテリアルを提供したお客様は、AWS KMS の外部におけるキーマテリアルの安全な使用、耐久性、メンテナンスに対して責任を持ちます。

データ暗号化

AWS KMS のデータは、AWS KMS keys と、それらが表す暗号化キーマテリアルで構成されます。このキーマテリアルは、AWS KMS ハードウェアセキュリティモジュール (HSM) 内でのみ、かつ使用中の場合にのみ、プレーンテキストで存在します。それ以外の場合、キー素材は暗号化され、耐久性のある永続ストレージに保存されます。

KMS キー用に AWS KMS が生成するキーマテリアルは、AWS KMS HSM の境界に暗号化されずに残ることはありません。これは、どの AWS KMS API オペレーションでもエクスポートまたは送信されることはありません。マルチリージョンキーの場合は例外で、AWS KMS がクロスリージョンレプリケーションメカニズムを使用して、マルチリージョンキーのキーマテリアルを、1 つの AWS リージョン にある HSM から、別の AWS リージョン にある HSM にコピーします。詳細については、AWS Key Management Service 暗号化の詳細の「マルチリージョンキーのレプリケーションプロセス」を参照してください。

トピック

- 保管中の暗号化
- 転送中の暗号化

保管中の暗号化

AWS KMS は、FIPS 140-2 セキュリティレベル 3 準拠のハードウェアセキュリティモジュール (HSM) で AWS KMS keys のキーマテリアルを生成します。唯一の例外は中国リージョンで、AWS KMS が KMS キーを生成するために使用する HSM は、関連するすべての中国の規制に準拠していますが、FIPS 140-2 暗号化モジュール検証プログラムでは検証されていません。使用されていない場合、キーマテリアルは HSM キーによって暗号化され、耐久性のある永続的なストレージに書き込まれます。KMS キーのキーマテリアルおよびキーマテリアルを保護する暗号化キーは、HSM をプレーンテキスト形式のままにしません。

KMS キーのキーマテリアルの暗号化と管理は、AWS KMS によって完全に処理されます。

詳細については、AWS Key Management Service の暗号化の詳細の Working with AWS KMS keys を参照してください。

転送中の暗号化

KMS キー用に AWS KMS が生成するキーマテリアルは、AWS KMS API オペレーションでエクスポートまたは送信されることはありません。AWS KMS は、API オペレーションの KMS キーを表すためにキー識別子を使用します。同様に、AWS KMS カスタムキーストアの KMS キーのキーマテリアルはエクスポート不可能であり、AWS KMS または AWS CloudHSM API オペレーションでは送信されません。

ただし、一部の AWS KMS API オペレーションはデータキーを返します。お客様は API オペレーションを使用して、選択した KMS キーのキーマテリアルをインポートすることもできます。

すべての AWS KMS API 呼び出しは、Transport Layer Security (TLS) を使用して署名および送信する必要があります。 AWS KMS には TLS 1.2 が、すべてのリージョンで TLS 1.3 が推奨されます。AWS KMS は、中国リージョンを除くすべてのリージョンの AWS KMS サービスエンドポイントのハイブリッドポスト量子 TLS もサポートしています。AWS KMS は、AWS GovCloud (US) の FIPS エンドポイントのハイブリッドポスト量子 TLS をサポートしていません。AWS KMS の呼び出しには、PFS (Perfect Forward Secrecy) をサポートする最新の暗号スイートも必要です。つまり、プライベートキーなどのシークレットが漏洩した場合でも、セッションキーは漏洩しません。

コマンドラインインターフェイスまたは API により AWS にアクセスするときに FIPS 140-2 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。標準の AWS KMS エンドポイントまたは AWS KMS FIPS エンドポイントを使用するには、クライアントが TLS 1.2 以降をサポートする必要があります。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] をご参照ください。AWS KMS FIPS エンドポイントのリストについては、「AWS 全般のリファレンス」の「AWS Key Management Service エンドポイントとクォータ」を参照してください。

AWS KMS サービスホストと HSM 間の通信は、楕円曲線暗号 (ECC) と高度暗号化規格 (AES) を使用して認証された暗号化方式で保護されます。詳細については、AWS Key Management Service の暗号化の詳細の「内部通信セキュリティ」を参照してください。

インターネットトラフィックのプライバシー

AWS KMS は AWS Management Console および API オペレーションのセットをサポートします。これにより、AWS KMS keys を作成および管理し、暗号化オペレーションで使用することができます。

AWS KMS はプライベートネットワークから AWS への、2 つのネットワーク接続オプションをサポートします。

- インターネット経由の IPsec VPN 接続
- AWS Direct Connect は、お客様の内部ネットワークを AWS Direct Connect ロケーションに、標準のイーサネット光ファイバケーブルを介して接続するサービスです。

AWS KMS API コールはすべて、署名し、Transport Layer Security (TLS) を使用して送信する必要があります。コールには、 完全な転送秘密をサポートする最新の暗号スイートも必要です。KMS キーのキーマテリアルを保存するハードウェアセキュリティモジュール (HSM) へのトラフィックは、AWS の内部ネットワーク経由で既知の AWS KMS API ホストからのみ許可されます。

トラフィックをパブリックインターネット経由で送信せずに、Virtual Private Cloud (VPC) から AWS KMS に直接接続するには、AWS PrivateLink によって提供される VPC エンドポイントを使用します。詳細については、「VPC エンドポイントを介した AWS KMS への接続」を参照してください。

AWS KMS は、Transport Layer Security (TLS) ネットワーク暗号化プロトコル用のハイブリッドポスト量子キー交換オプションもサポートするようになりました。このオプションは、AWS KMS API エンドポイントへの接続時に TLS で使用できます。

AWS Key Management Service のためのアイデンティティおよびアクセス管理

AWS Identity and Access Management (IAM) は、AWS リソースへのアクセスを安全に制御するのに役立ちます。管理者は、認証 (サインイン) するユーザー、および AWS KMS リソースの使用許可 (アクセス許可) を付与するユーザーを制御します。詳細については、「AWS KMS で IAM ポリシーを使用する」を参照してください。

キーポリシーは、AWS KMS でKMS キーへのアクセスを制御するための主要メカニズムです。すべての KMS キーにはキーポリシーが必要です。IAM ポリシーと権限をキーポリシーとともに使用して、KMS キーへのアクセスを制御することもできます。詳細については、「AWS KMS の認証とアクセスコントロール」を参照してください。

Amazon Virtual Private Cloud (Amazon VPC) を使用している場合は、AWS PrivateLink を使用した AWS KMS へのインターフェイス VPC エンドポイントを作成することができます。また、VPC エンドポイントポリシーを使用して、AWS KMS エンドポイントにアクセスできるプリンシパル、実行できる API コール、アクセスできる KMS キーを決定することもできます。詳細については、「VPC エンドポイントへのアクセスの制御」を参照してください。

AWS Key Management Service でのログ記録とモニタリング

モニタリングは、AWS KMS での AWS KMS keys の可用性、ステータス、使用状況を理解する上で重要なパートです。モニタリングは、AWS ソリューションのセキュリティ、信頼性、可用性、パフォーマンスを維持するのに役立ちます。AWS には、KMS キーをモニタリングするための各種ツールが用意されています。

AWS CloudTrail ログ

AWS KMS API オペレーションへのすべての呼び出しは、AWS CloudTrail ログにイベントとしてキャプチャされます。これらのログには、AWS KMS コンソールからのすべての API 呼び出し、および AWS KMS やその他の AWS のサービスによる呼び出しが記録されます。異なる で KMS

キーを使用する呼び出しなど、クロスアカウント API 呼び出しはAWS アカウント、両方のアカウントの CloudTrail ログに記録されます。

トラブルシューティングまたは監査を行う際、ログを使用して KMS キーのライフサイクルを再構築できます。また、暗号化オペレーションにおける KMS キーの管理および使用を表示することもできます。詳細については、「the section called “AWS CloudTrail でのログ記録”」を参照してください。

Amazon CloudWatch Logs

AWS CloudTrail およびその他のソースのログファイルをモニタリング、保存、アクセスします。詳細については、「Amazon ユーザーガイド CloudWatch 」を参照してください。

の場合AWS KMS、 は、KMS キーとその保護対象のリソースに関する問題を防ぐのに役立つ有用な情報 CloudWatch を保存します。詳細については、「the section called “によるモニタリング CloudWatch”」を参照してください。

Amazon EventBridge

AWS KMS は、KMS キーがローテーションまたは削除されたとき、または KMS キーにインポートされたキーマテリアルの有効期限が切れたときに EventBridge イベントを生成します。???AWS KMS イベント (API 操作) を検索し、1 つ以上のターゲット関数またはストリームにルーティングして、状態情報を取得します。詳細については、the section called “Amazon によるモニタリング EventBridge”「」および「Amazon ユーザーガイド EventBridge 」を参照してください。

Amazon CloudWatch メトリクス

から未加工データを収集してパフォーマンス CloudWatch メトリクスAWS KMSに処理するメトリクスを使用して、KMS キーをモニタリングできます。データは 2 週間間隔で記録されるため、現在および過去の情報の傾向を表示できます。これにより、KMS キーがどのように使用され、それらの使用が時間の経過とともにどのように変化するかを理解できます。 CloudWatch メトリクスを使用して KMS キーをモニタリングする方法については、「」を参照してくださいAWS KMS のメトリクスとディメンション。

Amazon CloudWatch アラーム

指定した期間における単一のメトリックスの変化を監視します。次に、一定期間におけるしきい値に対するメトリックの値に基づいてアクションを実行します。例えば、暗号化オペレーションで削除が予定されている KMS キーを誰かが使用しようとしたときにトリガーされる CloudWatch アラームを作成できます。これは、KMS キーがまだ使用中であり、削除すべきではないことを示します。詳細については、「the section called “アラームを作成する”」を参照してください。

AWS Security Hub

AWS Security Hub を使用すると、セキュリティ業界標準とベスト プラクティスへの準拠について AWS KMS の使用状況をモニターリングできます。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。詳細については、「AWS Security Hub ユーザーガイド」の「AWS Key Management Service コントロール」を参照してください。

AWS Key Management 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 AWS Key Management Service コンプライアンスプログラムの一環として AWS のセキュリティとコンプライアンスを評価します。これらのプログラムには、SOC、PCI、FedRAMP、HIPAA などがあります。

トピック

- コンプライアンスとセキュリティに関するドキュメント
- 詳細はこちら

コンプライアンスとセキュリティに関するドキュメント

次のコンプライアンスおよびセキュリティドキュメントは AWS KMS を対象としています。表示するには、AWS Artifact を使用します。

- クラウドコンピューティングコンプライアンスコントロールカタログ (C5)
- ISO 27001:2013 適用宣言書 (SoA)
- ISO 27001:2013 認証
- ISO 27017:2015 適用宣言書 (SoA)
- ISO 27017:2015 認証
- ISO 27018:2015 適用宣言書 (SoA)
- ISO 27018:2014 認証
- ISO 9001:2015 認証
- PCI DSS Attestation of Compliance (AOC) と Responsibility Summary
- Service Organization Controls (SOC) 1 レポート
- Service Organization Controls (SOC) 2 レポート

- Service Organization Controls (SOC) 2 機密性に関するレポート
- フェドランプ高

AWS Artifact の使用方法については、AWS Artifact でのレポートのダウンロードを参照してください。

詳細はこちら

AWS KMS を使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性や貴社のコンプライアンス目的、適用可能な法律および規制によって決定されます。AWS KMS の使用が、公開されている規格に準拠していることを前提としている場合、AWS では、次の支援リソースを提供しています。

- コンプライアンスプログラム対象範囲内の AWS サービス – このページには、特定のコンプライアンスプログラムの対象となる AWS のサービスが一覧表示されています。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。
- 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、機密性とコンプライアンスに焦点を当てたベースライン環境を AWS にデプロイするためのステップが示されています。
- AWS コンプライアンスのリソース - ワークブックとお客様の業界や所在地に適用される場合があるガイドのコレクション。
- AWS Config - この AWS のサービスでは、自社プラクティス、業界ガイドライン、および規制に対するリソースの設定の準拠状態を評価します。
- AWS Security Hub この AWS サービスは、AWS 内のセキュリティ状態に関する包括的な表示を提供します。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。

AWS Key Management Service での耐障害性

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、およびスケーラビリティが優れています。

AWS では、AWS KMS 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。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。

リージョンの隔離

AWS Key Management Service (AWS KMS) は、すべての AWS リージョン で利用できる自立型のリージョナルサービスです。AWS KMS のリージョンに関して分離された設計により、 AWS リージョン の可用性の問題が他のリージョンの AWS KMS オペレーションに影響を与えないことが保証されます。AWS KMS は、計画されたダウンタイムをゼロにするように設計されており、すべてのソフトウェア更新とスケーリングオペレーションがシームレスかつ気付かれることなく実行されます。

AWS KMS サービスレベルアグリーメント (SLA) には、すべての KMS API に対して 99.999% のサービスコミットメントが含まれています。このコミットメントを実現するために、AWS KMS は API リクエストの実行に必要なすべてのデータと認証情報が、リクエストを受信するすべてのリージョンのホストで利用可能であることを確認します。

AWS KMS インフラストラクチャは、各リージョンの 3 つ以上のアベイラビリティーゾーン (AZ) にレプリケートされます。複数のホスト障害による AWS KMS のパフォーマンスへの影響を受けないように、リージョンの どの AZ からの顧客トラフィックでも処理するように、AWS KMS は設計されています。

KMS キーのプロパティまたは許可に加えた変更は、リージョン内のすべてのホストにレプリケートされ、後続のリクエストがリージョン内の任意のホストで正しく処理されるようにします。KMS キーを使用した暗号化オペレーションのリクエストは、AWS KMS ハードウェアセキュリティモジュール (HSM) のフリートに転送され、そのいずれもが KMS キーを使用してオペレーションを実行できます。

マルチテナント設計

AWS KMS のマルチテナント設計により、99.999% の可用性 SLA を満たし、高いリクエストレートを維持しながら、キーとデータの機密性を保護できます。

暗号化オペレーションに指定した KMS キーが常に使用されるキーであることを保証するために、複数の整合性強制メカニズムがデプロイされます。

KMS キーのプレーンテキストキーマテリアルは、広範囲に保護されています。キーマテリアルは作成後すぐに HSM で暗号化され、暗号化されたキーマテリアルはセキュアで低レイテンシーのストレージに即座に移動されます。暗号化されたキーは、HSM 内で取得され、使用に間に合うように復号されます。プレーンテキストキーは、暗号化オペレーションを完了するのに必要な時間だけ HSM メモリに残ります。その後、HSM で再暗号化され、暗号化されたキーがストレージに返されます。プレーンテキストのキーマテリアルが HSM を離れることはありません。永続ストレージに書き込まれることもありません。

AWS KMS がキーを保護するために使用するメカニズムの詳細は、AWS Key Management Service 暗号化の詳細を参照してください。

AWS KMS でのレジリエンスのベストプラクティス

AWS KMS リソースのレジリエンスを最適化するには、以下の戦略を検討してください。

- バックアップおよびディザスタリカバリ戦略をサポートするには、1 つの AWS リージョン で作成された KMS キーであり、指定したリージョンにのみレプリケートされる、マルチリージョンキーを検討します。マルチリージョンキーを使用すると、暗号化されたリソースをプレーンテキストを公開することなく AWS リージョン (同じパーティション内) で移動し 、必要に応じて任意の送信先リージョンでリソースを復号化することができます。関連するマルチリージョンキーは、同じキーマテリアルとキー ID を共有するため、相互運用可能ですが、高解像度のアクセスコントロールのための独立したキーポリシーがあります。詳細については、AWS KMS のマルチリージョンキーを参照してください。
- AWS KMS のようなマルチテナントのサービスでキーを保護するには、キーポリシーおよび IAM ポリシーを含め、必ずアクセスコントロールを使用してください。さらに、AWS PrivateLinkによってサポートされる VPC インターフェイスエンドポイントを使用して、リクエストを AWS KMS に送信できます。To support your backup and disaster recovery strategy, その場合、Amazon VPC 間のすべての通信および AWS KMS は、VPC に制限された専用 AWS KMS エンドポイントを使用して AWS ネットワーク内で完全に実行されます。VPC エンドポイントポリシーを使用して追加の認証レイヤーを作成することで、これらのリクエストをさらに保護できます。詳細については、「VPC エンドポイント経由で AWS KMS に接続する」を参照してください。

AWS Key Management Service 内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 AWS Key Management Service (AWS KMS) は、Amazon Web Services: セキュリティプロセスの概要で説明されている AWS グローバルネットワークのセキュリティ手順によって保護されます。

ネットワークを介して AWS KMS にアクセスするには、「AWS Key Management Service API リファレンス」で説明されている AWS KMS API オペレーションを呼び出すことができます。AWS KMS には TLS 1.2 が、すべてのリージョンで TLS 1.3 が推奨されます。AWS KMS は、中国リージョンを除くすべてのリージョンの AWS KMS サービスエンドポイントのハイブリッドポスト量子 TLS もサポートします。AWS KMS は、AWS GovCloud (US) の FIPS エンドポイントのハイブリッドポスト量子 TLS をサポートしていません。標準の AWS KMS エンドポイントまたは AWS KMSFIPS エンドポイントを使用するには、クライアントが TLS 1.2 以降をサポートする必要があります。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどの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。

これらの API オペレーションは、任意のネットワークロケーションから呼び出すことができますが、AWS KMS では、ソース IP アドレス、VPC、VPC エンドポイントに基づいて KMS キーへのアクセスを制御するグローバルポリシー条件をサポートします。これらの条件キーは、キーポリシーと IAM ポリシーで使用できます。ただし、これらの条件により、AWS がユーザーの代わりに KMS キーを使用できなくなる可能性があります。詳細については、「AWS グローバル条件キー」を参照してください。

例えば、次のキーポリシーステートメントでは、ソース IP アドレスがポリシーで指定された IP アドレスの 1 つでない限り、`KMSTestRole` ロールを引き受けることができるユーザーが、指定された暗号化オペレーションでこの AWS KMS key を使用できます。

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

```
    "Effect": "Allow",
    "Principal": {"AWS":
    "arn:aws:iam::111122223333:role/KMSTestRole"},
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24",
          "203.0.113.0/24"
        ]
      }
    }
  }
}
```

物理ホストの分離

AWS KMS が使用する物理的インフラストラクチャのセキュリティは、Amazon Web Services: セキュリティプロセスの概要の物理的および環境的セキュリティのセクションで説明されている制御に支配されます。詳細については、前のセクションにリストされたコンプライアンスレポートとサードパーティーの監査結果を参照してください。

AWS KMS は、物理的な攻撃に抵抗するための特定のコントロールを使用して設計された、専用の強化ハードウェアセキュリティモジュール (HSM) によってサポートされています。HSM は、ハイパーバイザーなどの仮想化レイヤー を持たない 物理デバイスで、物理デバイスを複数の論理テナント間で共有します。AWS KMS keys のキーマテリアルは、KMS キーの使用中にのみ、HSM の揮発性メモリにのみ保存されます。このメモリは、HSM が意図したシャットダウンやリセットなど、動作状態から移行すると消去されます。AWS KMS HSM のオペレーションの詳細については、AWS Key Management Service の暗号化の詳細を参照してください。

AWS Key Management Service のセキュリティに関するベストプラクティス

AWS Key Management Service (AWS KMS) は、暗号化キーの保護を強化するために実装できる多くのセキュリティ機能をサポートしています。これにはキーポリシーおよび IAM ポリシー、対称型暗号キーの暗号化オペレーションのための暗号化コンテキストオプション、キーポリシーや IAM ポリシーを洗練させるための広範な条件キーのセット、および許可を制限する制約の付与などが含まれます。

これらのセキュリティ機能の詳細は、AWS Key Management Service のベストプラクティス (PDF) に記載されています。本技術書の一般的なガイドラインは、完全なセキュリティソリューションを提供するものではありません。すべてのベストプラクティスがあらゆる状況に適しているわけではないため、これらは規範的なものではありません。

以下の資料も参照してください。

- IAM ポリシーのベストプラクティス
- AWS KMS グラントのベストプラクティス
- 「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」

クォータ

すべてのユーザーに対して AWS KMS の応答性とパフォーマンスを向上させるために、AWS KMS ではリソースクォータとリクエストクォータの 2 種類のクォータが適用されます。各クォータは、各 AWS アカウント のリージョンごとに個別に計算されます。

すべての AWS KMS クォータは、キーポリシードキュメントのサイズのリソースクォータとAWS CloudHSM キーストアのリクエストクォータを除いて調整可能です。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。クォータの引き下げリクエスト、Service Quotas に一覧表示されていないクォータの変更、AWS KMS の Service Quotas を使用できない AWS リージョン のクォータの変更を行うには、AWS Support センターにアクセスしてケースを作成します。

トピック

- リソースクォータ
- クォータのリクエスト
- AWS KMS リクエストのスロットリング

リソースクォータ

AWS KMS は、リソースクォータを設け、すべてのお客様に迅速かつ回復力のあるサービスを提供できるようにします。一部のリソースクォータはユーザーが作成したリソースにのみ適用され、AWS のサービスによって作成されたリソースには適用されません。AWS 所有のキー など、使用するリソースで AWS アカウント に含まれていないものは、これらのクォータにはカウントされません。

リソースの上限を超えた場合、そのタイプの追加リソースの作成をリクエストすると、`LimitExceededException` エラーメッセージが生成されます。

すべての AWS KMS リソースクォータは、キーポリシードキュメントのサイズクォータを除いて、調整可能です。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。クォータの引き下げリクエスト、Service Quotas に一覧表示されていないクォータの変更、AWS KMS の Service Quotas を使用できない AWS リージョン のクォータの変更を行うには、AWS Support センターにアクセスしてケースを作成します。

次の表に、各 AWS アカウント アカウントとリージョンの AWS KMS リソースクォータを一覧表示し、その説明を示します。

クォータ名	デフォルト値	適用先	調整可能
AWS KMS keys	100,000	カスタマーマネージドキー	はい
KMS キーごとのエイリアス	50	カスタマー作成のエイリアス	はい
KMS キーごとのグラント	50,000	カスタマーマネージドキー	はい
キーポリシードキュメントのサイズ	32 KB (32,768 バイト)	カスタマーマネージドキー AWS マネージドキー	いいえ
カスタムキーストアのリソースクォータ	10	AWS アカウント とリージョン	はい

リソースクォータに加えて、AWS KMS はリクエストクォータを使用して、サービスの応答性を確保します。詳細については、「the section called “クォータのリクエスト”」を参照してください。

AWS KMS keys: 100,000

AWS アカウント の各リージョンで、最大 100,000 個のカスタマーマネージドキーを持つことができます。このクォータは、キー仕様またはキーステータスに関係なく、すべての AWS リージョン ですべてのカスタマー管理キーに適用されます。各 KMS キーは 1 つのリソースと見なされます。AWS マネージドキー および AWS 所有のキー はこのクォータにはカウントされません。

KMS キーごとのエイリアス: 50

最大 50 個のエイリアスを各カスタマーマネージドキーに関連付けることができます。AWS により AWS マネージドキー に関連付けられているエイリアスは、このクォータにはカウントされません。エイリアスを作成または更新するときに、このクォータが発生することがあります。

> **Note**
>
> kms:ResourceAliases 条件は、KMS キーがこのクォータに準拠している場合にのみ有効です。KMS キーがこのクォータを超えると、KMS キーを `kms:ResourceAliases` 条件で使

用するよう認可されたプリンシパルは、KMS キーへのアクセスを拒否されます。詳細については、「エイリアスクォータによりアクセスが拒否された」を参照してください。

KMS キークォータごとのエイリアスは、AWS アカウント で各リージョンのエイリアスの合計数を制限するリージョンクォータごとのエイリアスに取って代わります。AWS KMS では、リージョンクォータごとのエイリアスが排除されました。

KMS キーあたりのグラント: 50,000

各カスタマーマネージドキーは、AWS KMS と統合されている AWS サービスによって作成されるグラントを含めて、最大 50,000 個のグラントを持つことができます。このクォータは、AWS マネージドキー または AWS 所有のキー には適用されません。

このクォータの効果の 1 つは、同じ KMS キーを同時に使用する 50,000 を超える許可されたオペレーションを実行できないことです。このクォータ到達後は、アクティブなグラントが廃止または取り消しになった場合にのみ、KMS キーで新しいグラントを作成できます。

例えば、Amazon Elastic Block Store (Amazon EBS) ボリュームを Amazon Elastic Compute Cloud (Amazon EC2) インスタンスにアタッチすると、ボリュームは復号化され、読めるようになります。データを復号する許可を得るために、Amazon EBS は各ボリュームに対してグラントを作成します。したがって、すべての Amazon EBS ボリュームが同じ KMS キーを使用する場合、一度に 50,000 を超えるボリュームをアタッチすることはできません。

キーポリシードキュメントのサイズ: 32 KB

各キーポリシードキュメントの最大長は 32 KB (32,768 バイト) です。より大きなポリシードキュメントを使用して KMS キーのキーポリシーを作成または更新すると、オペレーションは失敗します。

これは調整可能なクォータではありません。Service Quotas を使用したり、AWS Support でケースを作成したりして、この値を増やすことはできません。キーポリシーが制限に近づいている場合は、ポリシーステートメントの代わりにグラントの使用を検討してください。グラントは、一時的または特定のアクセス許可に特に適しています。

キーポリシードキュメントは、 のデフォルトビューまたはポリシービュー、または オペレーションを使用して、キーポリシーを作成または変更するたびに使用します。 AWS Management Console PutKeyPolicyこのクォータは、JSON ステートメントを直接編集しない AWS KMS コンソールでデフォルトビューを使用する場合にも、キーポリシードキュメントに適用されます。

カスタムキーストアのリソースクォータ: 10

各 AWS アカウント とリージョンに最大 10 のカスタムキーストアを作成できます。さらに を作成しようとすると、CreateCustomKeyStoreオペレーションは失敗します。

このクォータは、接続状態に関わりなく、すべての AWS CloudHSM キーストア と 外部キーストアを含む、各アカウントとリージョンのカスタムキーストアの合計数に適用されます。

クォータのリクエスト

AWS KMS は、1 秒あたりに要求される API オペレーションの数にクォータを設定します。リクエストクォータは、API オペレーション、AWS リージョン、および KMS キータイプなどのその他の要因によって異なります。API リクエストのクォータを超えると、AWS KMS はリクエストを調整します。

AWS KMS リクエストクォータは、AWS CloudHSM キーストアのリクエストクォータを除き、すべて調整可能です。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。クォータの引き下げリクエスト、Service Quotas に一覧表示されていないクォータの変更、AWS KMS の Service Quotas を使用できない AWS リージョン のクォータの変更を行うには、AWS Support センターにアクセスしてケースを作成します。

GenerateDataKey オペレーションのリクエストクォータを超えている場合は、 のデータキーキャッシュ機能の使用を検討してくださいAWS Encryption SDK。データキーを再利用することで、AWS KMS へのリクエストの頻度が低下することがあります。

AWS KMS では、リクエストクォータに加えて、リソースクォータを使用してすべてのユーザーの容量を確保します。詳細については、「リソースクォータ」を参照してください。

リクエストレートのトレンドを表示するには、Service Quotas コンソールを使用してください。リクエストレートがクォータ値の特定のパーセンテージに達したときに警告する Amazon CloudWatch アラームを作成することもできます。詳細については、 AWS セキュリティブログの「Service Quotas と Amazon を使用して AWS KMS API リクエストレートを管理する CloudWatch」を参照してください。

トピック

- AWS KMS API オペレーションごとにクォータをリクエストする
- リクエストクォータの適用
- 暗号化オペレーションの共有クォータ

- ユーザーに代わって API が実行するリクエスト
- クロスアカウントリクエスト
- カスタムキーストアのリクエストクォータ

AWS KMS API オペレーションごとにクォータをリクエストする

この表では、Service Quotas のクォータコードと、各 AWS KMS リクエストクォータのデフォルト値を示しています。AWS KMS リクエストクォータは、AWS CloudHSM キーストアのリクエストクォータを除き、すべて調整可能です。

> **Note**
>
> この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

クォータ名	デフォルト値 (1 秒あたりのリクエスト)

`Cryptographic operations (symmetric) request rate`
 	
適用先:
 	
• `Decrypt`
• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
• `GenerateMac`
• `GenerateRandom`
• `ReEncrypt`
• `VerifyMac`
	これらの共有クォータは、AWS リージョン およびリクエストで使用される KMS キーのタイプによって異なります。各クォータは個別に計算されます。
	
	• 5,500 (共有)
	• 以下のリージョンでは 10,000 (共有):
	  • 米国東部 (オハイオ)、us-east-2
	  • アジアパシフィック (シンガポール)、ap-southeast-1
	  • アジアパシフィック (シドニー)、ap-southeast-2
	  • アジアパシフィック (東京)、ap-northeast-1
	  • ヨーロッパ (フランクフルト)、eu-central-1
	  • ヨーロッパ (ロンドン)、eu–west-2
	• 以下のリージョンでは 50,000 (共有):

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
	• 米国東部 (バージニア北部)、us-east-1 • 米国西部 (オレゴン)、us-west-2 • ヨーロッパ (アイルランド)、eu-west-1
`Cryptographic operations (RSA) request rate` 適用先: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	RSA KMS キーの場合は 500 (共有)
`Cryptographic operations (ECC) request rate` 適用先: • `Sign` • `Verify`	楕円曲線 (ECC) KMS キーの場合は 300 (共有)
`Cryptographic operations (SM) request rate` 適用先: • `Decrypt` • `Encrypt` • `ReEncrypt` • `Sign` • `Verify`	SM2 (中国リージョンのみ) KMS キーの場合は 300 (共有)

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
Custom key store request quotas 適用先: • Decrypt • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlaintext • GenerateRandom • ReEncrypt	カスタムキーストアのリクエストクォータは、カスタムキーストアごとに個別に計算されます。 • AWS CloudHSM キーストアごとに 1,800 (共有) • 外部キーストアごとに 1,800 (共有)
CancelKeyDeletion request rate	5
ConnectCustomKeyStore request rate	5
CreateAlias request rate	5
CreateCustomKeyStore request rate	5
CreateGrant request rate	50
CreateKey request rate	5
DeleteAlias request rate	15
DeleteCustomKeyStore request rate	5
DeleteImportedKeyMaterial request rate	5
DescribeCustomKeyStores request rate	5
DescribeKey request rate	2000

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
DisableKey request rate	5
DisableKeyRotation request rate	5
DisconnectCustomKeyStore request rate	5
EnableKey request rate	5
EnableKeyRotation request rate	15
GenerateDataKeyPair (ECC_NIST_P256) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P384) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P521) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
GenerateDataKeyPair (ECC_SECG_P256K1) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (RSA_2048) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	1
GenerateDataKeyPair (RSA_3072) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	0.5 (2 秒に 1 回)
GenerateDataKeyPair (RSA_4096) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	0.1 (10 秒に 1 回)

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
GenerateDataKeyPair (SM2 – China Regions only) request rate 適用先: • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	25
GetKeyPolicy request rate	1,000
GetKeyRotationStatus request rate	1,000
GetParametersForImport request rate	0.25 (4 秒に 1 回)
GetPublicKey request rate	2000
ImportKeyMaterial request rate	5
ListAliases request rate	500
ListGrants request rate	100
ListKeyPolicies request rate	100
ListKeys request rate	500
ListResourceTags request rate	2000
ListRetirableGrants request rate	100
PutKeyPolicy request rate	15

クォータ名	デフォルト値 (1 秒あたりのリクエスト)
ReplicateKey request rate ReplicateKey オペレーションは、プライマリキーのリージョンでは 1 つ ReplicateKey リクエスト、レプリカのリージョンでは 2 つの CreateKey リクエストとしてカウントされます。以下のうち 1 つの CreateKey リクエストは、キー作成前に潜在的な問題を検出するためのドライランです。	5
RetireGrant request rate	30
RevokeGrant request rate	30
ScheduleKeyDeletion request rate	15
TagResource request rate	10
UntagResource request rate	5
UpdateAlias request rate	5
UpdateCustomKeyStore request rate	5
UpdateKeyDescription request rate	5
UpdatePrimaryRegion request rate UpdatePrimaryRegion オペレーションは、2 つの影響を受けるリージョンごとに、1 つのリクエストとしてカウントされる、2 つの UpdatePrimaryRegion リクエストです。	5

リクエストクォータの適用

リクエストクォータを確認するときは、次の点に注意してください。

- リクエストクォータは、カスタマーマネージドキーと AWS マネージドキー の両方に適用されます。AWS 所有のキー の使用は、アカウント内のリソースを保護するために使用される場合でも、AWS アカウント のリクエストクォータにはカウントされません。
- リクエストクォータは、FIPS エンドポイントおよび非 FIPS エンドポイントに送信されるリクエストに適用されます。AWS KMS サービスエンドポイントのリストについては、「AWS 全般のリファレンス」の「AWS Key Management Service エンドポイントとクォータ」を参照してください。
- スロットリングは、リージョン内のすべてのタイプの KMS キーに対するすべてのリクエストに基づいています。この合計には、ユーザーの代わりとなる AWS のサービスからのリクエストを含む、AWS アカウント のすべてのプリンシパルからのリクエストが含まれます。
- 各要求クォータは個別に計算されます。例えば、 CreateKeyオペレーションのリクエストは、CreateAliasオペレーションのリクエストクォータには影響しません。`CreateAlias` リクエストが調整されていても、`CreateKey` リクエストは正常に完了できます。
- 暗号化オペレーションはクォータを共有しますが、共有クォータは他のオペレーションのクォータとは無関係に計算されます。例えば、Encrypt オペレーションと Decrypt オペレーションの呼び出しはリクエストクォータを共有しますが、そのクォータは などの管理オペレーションのクォータとは無関係ですEnableKey。例えば、欧州 (ロンドン) リージョンでは、対称 KMS キーに対して 10,000 オペレーションの暗号化オペレーションに加えて、スロットルなしで 1 秒あたり 5 回の `EnableKey` オペレーションを実行できます。

暗号化オペレーションの共有クォータ

AWS KMS 暗号化オペレーションは、リクエストクォータを共有します。KMS キーでサポートされている暗号化オペレーションの任意の組み合わせをリクエストできます。これにより、暗号化オペレーションの合計数がそのタイプの KMS キーのリクエストクォータを超過しません。例外は GenerateDataKeyPairと でGenerateDataKeyPairWithoutPlaintext、個別のクォータを共有します。

異なるタイプの KMS キーのクォータは、個別に計算されます。各クォータは、1 秒間隔の特定のキータイプを持つ AWS アカウント およびリージョンで、これらのオペレーションに対するすべてのリクエストに適用されます。

- 暗号化オペレーション (対称) リクエストの頻度は、アカウントとリージョンで対称 KMS キーを使用する暗号化オペレーションの共有リクエストクォータです。このクォータは、対称暗号化キーと、同じく対称である HMAC キーでの暗号化オペレーションに適用されます。

  例えば、1 秒あたり 10,000 リクエストの共有クォータを持つ AWS リージョン で対称 KMS キーを使用しているとします。1 秒あたり 7,000 GenerateDataKeyリクエスト、1 秒あたり 2,000

リクエストの復号化を行うと、 AWS KMSはリクエストをスロットリングしません。ただし、毎秒 9,500 件の `GenerateDataKey` リクエストと 1,000 件の Encrypt リクエストを行うと、共有クォータを超えているため、AWS KMS はリクエストを制限します。

カスタムキーストア内の対称暗号化 KMS キーに対する暗号化オペレーションは、アカウントの暗号化オペレーション (対称) リクエストレートおよびカスタムキーストアのカスタムキーストアのリクエストクォータの両方にカウントされます。

- 暗号化オペレーション (RSA) リクエストの頻度は、RSA 非対称 KMS キーを使用する暗号化オペレーションの共有リクエストクォータです。

  例えば、1 秒あたり 500 オペレーションのリクエストクォータでは、暗号化および復号が可能な RSA KMS キーを使用して、200 件の Encrypt リクエストと 100 件の Decrypt リクエストを実行できます。加えて、署名と検証が可能な RSA KMS キーを使用して、50 件の Sign リクエストと 150 件の Verify リクエストを実行できます。
- 暗号化オペレーション (ECC) リクエストの頻度は、楕円曲線 (ECC) 非対称 KMS キーを使用する暗号化オペレーションの共有リクエストクォータです。

  例えば、リクエストクォータが 1 秒あたり 300 オペレーションであれば、署名と検証が可能な RSA KMS キーを使用して、100 件の署名リクエストと 200 件の検証リクエストを実行できます。
- 暗号化オペレーション (SM - 中国リージョンのみ) リクエストの頻度は、SM 非対称 KMS キーを使用する暗号化オペレーションの共有リクエストクォータです。

  例えば、1 秒あたり 300 オペレーションのリクエストクォータでは、暗号化および復号が可能な SM2 KMS キーを使用して、100 件の Encrypt リクエストと 100 件の Decrypt リクエストを実行できます。加えて、署名と検証が可能な SM2 KMS キーを使用して、50 件の Sign リクエストと 50 件の Verify リクエストを実行できます。
- カスタムキーストアのリクエストクォータは、カスタムキーストアの KMS キーに対する暗号化オペレーションの共有リクエストクォータです。このクォータは、各カスタムキーストアで個別に計算されます。

  カスタムキーストア内の対称暗号化 KMS キーに対する暗号化オペレーションは、アカウントの暗号化オペレーション (対称) リクエストレートおよびカスタムキーストアのカスタムキーストアのリクエストクォータの両方にカウントされます。

異なるキータイプのクォータも個別に計算されます。例えば、アジアパシフィック (シンガポール) リージョンでは、対称および非対称の KMS キーの両方を使用する場合、対称 KMS キー (HMAC キーを含む) を使用した 1 秒あたり最大 10,000 件の呼び出しに加えて、RSA 非対称 KMS キーを使

用した 1 秒あたり最大 500 件の追加呼び出しと、ECC ベースの KMS キーを使用した 1 秒あたり最大 300 件の追加リクエストを実行できます。

ユーザーに代わって API が実行するリクエスト

API が直接リクエストを行うこと、あるいは統合された AWS サービスを使用して、ユーザーに代わって API が AWS KMS にリクエストを行うようにできます。クォータはどちらの種類のリクエストにも適用されます。

たとえば、KMS キー (SSE-KMS) によるサーバー側の暗号化を使用して Amazon S3 にデータを保存できます。SSE-KMS で暗号化された S3 オブジェクトをアップロードまたはダウンロードするたびに、Amazon S3 はユーザーの代わりに、AWS KMS に `GenerateDataKey` (アップロード用) または `Decrypt` (ダウンロード用) リクエストを作成します。これらのリクエストはクォータに対してカウントされるため、AWS KMS は、SSE-KMS を使用して暗号化された S3 オブジェクトの 1 秒あたりのアップロード数またはダウンロード数の合計が 5,500 件 (または使用している AWS リージョンに応じて 10,000 件もしくは 50,000 件) を超える場合にリクエストをスロットルします。

クロスアカウントリクエスト

1 つの AWS アカウント のアプリケーションが、異なるアカウントの所有する KMS キーを使用することは、クロスアカウントリクエストと呼ばれます。クロスアカウントリクエストでは、AWS KMS は、KMS キーを所有するアカウントではなく、リクエストを行うアカウントを調整します。例えば、アカウント A のアプリケーションがアカウント B の KMS キーを使用する場合、KMS キーの使用はアカウント A のクォータにのみ適用されます。

カスタムキーストアのリクエストクォータ

AWS KMS は、カスタムキーストアの KMS キーに対する暗号化オペレーションのリクエストクォータを維持します。これらのリクエストクォータは、カスタムキーストアごとに個別に計算されます。

カスタムキーストアのリクエストクォータ	各カスタムキーストアのデフォルト値 (リクエスト数/秒)	調整可能
AWS CloudHSM キーストアのリクエストクォータ	1800	いいえ
外部キーストアのリクエストクォータ	1800	はい

> **Note**
>
> AWS KMS カスタムキーストアのリクエストクォータは、Service Quotas コンソールに表示されません。Service Quotas API オペレーションを使用して、これらのクォータを表示または管理することはできません。外部キーストアのリクエストクォータへの変更をリクエストするには、AWS Support センターにアクセスしてケースを作成します。
>
> AWS CloudHSM キーストアに関連付けられた AWS CloudHSM クラスターが、カスタムキーストアに関連しないものを含む多数のコマンドを処理している場合、 が lower-than-expected 一定のレートAWS KMS`ThrottlingException`で発生する可能性があります。この問題が発生した場合は、AWS KMS へのリクエストレートを下げる、関連のないコマンドの処理を減らす、AWS CloudHSM キーストアで専用の AWS CloudHSM クラスターを使用するなどの対策を取ります。
>
> AWS KMS は、 `ExternalKeyStoreThrottle` CloudWatch メトリクス内の外部キーストアリクエストのスロットリングを報告します。このメトリクスを使用して、スロットリングパターンを表示したり、アラームを作成したり、外部キーストアのリクエストクォータを調整したりできます。

カスタムキーストアの KMS キーに対する暗号化オペレーションのリクエストは、2 クォータにカウントされます。

- 暗号化オペレーション (対称) のリクエストレートクォータ (アカウントあたり)

  カスタムキーストアの KMS キーに対する暗号化オペレーションのリクエストは、各 AWS アカウント およびリージョンの `Cryptographic operations (symmetric) request rate` クォータにカウントされます。例えば、米国東部 (バージニア北部) (us-east-1) の場合、各 AWS アカウント は、カスタムキーストアで KMS キーを使用するリクエストを含め、対称暗号化 KMS キーに対して 1 秒あたり最大 5 万リクエストを処理できます。

- カスタムキーストアのリクエストクォータ (カスタムキーストアあたり)

  カスタムキーストアの KMS キーに対する暗号化オペレーションのリクエストも、1 秒あたり 1,800 オペレーションの `Custom key store request quota` にカウントされます。これらのクォータは、カスタムキーストアごとに個別に計算されます。カスタムキーストアで KMS キーを使用する、複数の AWS アカウント からのリクエストが含まれる場合があります。

例えば、米国東部 (バージニア北部) (us-east-1) リージョンのカスタムキーストア (いずれかのタイプ) の KMS キーに対する暗号化オペレーションは、アカウントとリージョンの `Cryptographic`

operations (symmetric) request rate アカウントレベルのクォータ (1秒あたり5万リクエスト)、およびカスタムキーストアの Custom key store request quota (1秒あたり1,800 リクエスト) にカウントされます。ただし、カスタムキーストアの KMS キーに対する などの管理オペレーションのリクエストはPutKeyPolicy、アカウントレベルのクォータ (1 秒あたり 15 リクエスト) にのみ適用されます。

AWS KMS リクエストのスロットリング

AWS KMS はすべてのお客様からの API リクエストに迅速で信頼性の高いレスポンスを確実に返すために、特定の境界を超える API リクエストを制御します。

スロットリングは、AWS KMS が拒否しなければ有効となってしまうリクエストを拒否して、次のような ThrottlingException エラーを返すときに発生します。

```
You have exceeded the rate at which you may call KMS. Reduce the frequency of your
 calls.
(Service: AWSKMS; Status Code: 400; Error Code: ThrottlingException; Request ID: <ID>
```

AWS KMS は、次の条件に対するリクエストを制御します。

- アカウントおよびリージョンの、1 秒あたりのリクエストレートが AWS KMS リクエストクォータを超えている。

  例えば、アカウントのユーザーが 1 秒に 1,000 DescribeKey リクエストを送信する場合、AWS KMS はその後の DescribeKey リクエストすべてを制御します。

  スロットリングに対応するには、バックオフと再試行戦略を使用します。この戦略は、一部の AWS SDK で、HTTP 400 エラー用に自動的に実装されています。
- 同じ KMS キーステータスを変更するための、リクエストのバーストレートまたは持続的な高いレート。多くの場合、この条件は「ホットキー」と呼ばれます。

  例えば、アカウント内のアプリケーションが同じ KMS キーに対して、持続的かつ一斉に EnableKey および DisableKey リクエストを送信した場合、AWS KMS はリクエストを制御します。このスロットリングは、リクエストが EnableKeyおよび DisableKeyオペレーションの request-per-second リクエスト制限を超えない場合でも発生します。

  スロットリングに対応するには、アプリケーションロジックを調整して必要なリクエストのみを作成するか、複数の関数のリクエストを統合します。

- キーAWS CloudHSMストアに関連付けられたAWS CloudHSMクラスターが、キーストアに関係のないコマンドを含む多数のコマンドを処理している場合、AWS CloudHSMキーストア内の KMS AWS CloudHSMキーに対するオペレーションのリクエストは、 lower-than-expected スロットリングされる可能性があります。

  (AWS KMS では、AWS CloudHSM クラスターで使用可能な PKCS #11 セッションがない場合、AWS CloudHSM キーストア内の KMS キーに対するオペレーションリクエストを調整しなくなりました。その代わりに、`KMSInternalException` がスローされます。リクエストを再試行することを推奨します)。

リクエストレートのトレンドを表示するには、Service Quotas コンソールを使用してください。リクエストレートがクォータ値の特定のパーセンテージに達したときに警告する Amazon CloudWatch アラームを作成することもできます。詳細については、 AWS セキュリティブログの「Service Quotas と Amazon を使用して AWS KMS API リクエストレートを管理する CloudWatch」を参照してください。

すべての AWS KMS クォータは、キーポリシードキュメントのサイズのリソースクォータとAWS CloudHSM キーストアのリクエストクォータを除いて調整可能です。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。クォータの引き下げリクエスト、Service Quotas に一覧表示されていないクォータの変更、AWS KMS の Service Quotas を使用できない AWS リージョン のクォータの変更を行うには、AWS Support センターにアクセスしてケースを作成します。

> **Note**
>
> AWS KMS カスタムキーストアのリクエストクォータは、Service Quotas コンソールに表示されません。Service Quotas API オペレーションを使用して、これらのクォータを表示または管理することはできません。外部キーストアのリクエストクォータへの変更をリクエストするには、AWS Support センターにアクセスしてケースを作成します。

AWS のサービスで AWS KMS を使用する方法

AWS サービスの多くは、AWS KMS を使用してデータの暗号化をサポートします。AWS サービスが AWS KMS と統合されている場合は、アカウントで AWS KMS keys を使用して、サービスがユーザーの代わりに受信、保存、管理するデータを保護できます。AWS KMS と統合されている AWS サービスの完全なリストについては、AWS サービス統合を参照してください。

次のトピックでは、サービスがサポートするKMS キー、データキーの管理方法、必要なアクセス許可、アカウント内で各サービスの KMS キーの使用を追跡する方法などを含む、特定のサービスが AWS KMS を使用する方法の詳細を説明します。

> **Important**
>
> AWS KMS と統合された AWS のサービスは、データの暗号化に対称暗号化 KMS キーのみを使用します。これらのサービスは、非対称 KMS キーを使用する暗号化をサポートしません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

トピック

- AWS CloudTrail で AWS KMS を使用する方法
- Amazon DynamoDB が AWS KMS を使用する方法
- Amazon Elastic Block Store (Amazon EBS) が AWS KMS を使用する方法
- Amazon Elastic Transcoder が AWS KMS を使用する方法
- Amazon EMR が AWS KMS を使用する方法
- AWS Nitro Enclaves が AWS KMS を使用する方法
- Amazon Redshift が AWS KMS を使用する方法
- Amazon Relational Database Service (Amazon RDS) が AWS KMS を使用する方法
- AWS Secrets Manager で AWS KMS を使用する方法
- Amazon Simple Email Service (Amazon SES) が AWS KMS を使用する方法
- Amazon Simple Storage Service (Amazon S3) が AWS KMS を使用する方法
- AWS Systems Manager Parameter Store が AWS KMS を使用する方法
- Amazon が WorkMail を使用する方法 AWS KMS
- が WorkSpaces を使用する方法 AWS KMS

AWS CloudTrail で AWS KMS を使用する方法

AWS CloudTrail を使用して、AWS API コールおよび AWS アカウント のその他のアクティビティを記録したり、選択した Amazon Simple Storage Service (Amazon S3) バケットのログファイルに記録した情報を保存したりできます。デフォルトでは、 が S3 バケット CloudTrail に配置するログファイルは、Amazon S3 が管理する暗号化キーによるサーバー側の暗号化 (SSE-S3) を使用して暗号化されます。ただし、その代わりに KMS キーによるサーバー側の暗号化 (SSE-KMS) の使用を選択することもできます。で CloudTrail ログファイルを暗号化する方法についてはAWS KMS、 AWS CloudTrailユーザーガイドのAWS KMS keys「 (SSE-KMS) による CloudTrail ログファイルの暗号化」を参照してください。

> **Important**
>
> AWS CloudTrail および Amazon S3 は、対称 AWS KMS keys のみをサポートします。非対称 KMS キーを使用して CloudTrail ログを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、「非対称 KMS キーの識別」を参照してください。

SSE-KMS キーで暗号化されたログファイル CloudTrail を読み書きする場合、キー使用料金は発生しません。ただし、SSE-KMS キーで暗号化された CloudTrail ログファイルにアクセスする場合は、キー使用料が発生します。AWS KMS の料金については、「AWS Key Management Service の料金」を参照してください。 CloudTrail 料金の詳細については、「 AWS CloudTrailユーザーガイド」の「 のAWS CloudTrail料金」および「コストの管理」を参照してください。

トピック

- KMS キーを使用するタイミングについて

KMS キーを使用するタイミングについて

を使用した CloudTrail ログファイルの暗号化は、 AWS KMS key (SSE-KMS) によるサーバー側の暗号化と呼ばれる Amazon S3 機能上にAWS KMS構築されます。SSE-KMS に関する詳細は、本ガイドの「Amazon Simple Storage Service (Amazon S3) が AWS KMS を使用する方法」、または「Amazon Simple Storage Service ユーザーガイド」の「KMS キーによるサーバー側の暗号化 (SSE-KMS) を使用したデータの保護」を参照してください。

SSE-KMS AWS CloudTrailを使用してログファイルを暗号化するように を設定する CloudTrail と、Amazon S3 はそれらのサービスで特定のアクションを実行AWS KMS keysするときに を使用し

ます。以下のセクションでは、これらのサービスが KMS キーをいつ、どのように使用するかについて説明し、この説明を検証するために使用できる追加情報を示します。

CloudTrail および Amazon S3 が KMS キーを使用する原因となるアクション

- でログファイルを暗号化 CloudTrail するように を設定する AWS KMS key
- CloudTrail は S3 バケットにログファイルを配置します。
- S3 バケットから暗号化されたログファイルを取得する

でログファイルを暗号化 CloudTrail するように を設定する AWS KMS key

KMS キー を使用するように設定を更新する CloudTrailと、 CloudTrail は に `GenerateDataKey` リクエストを送信AWS KMSして、KMS キーが存在し、それを encryption. CloudTrail does に使用するアクセス許可 CloudTrail を持っていることを確認します。結果のデータキーは使用されません。

`GenerateDataKey` リクエストには、暗号化コンテキストの次の情報が含まれています。

- CloudTrail 証跡の Amazon リソースネーム (ARN)
- S3 バケットの ARN と CloudTrail ログファイルが配信されるパス

`GenerateDataKey` リクエストの結果、次の例のようなエントリが CloudTrail ログに記録されます。このようなログエントリが表示された場合は、 CloudTrail (1) が AWS KMS () `GenerateDataKey`オペレーション (2) を特定の証跡 () として呼び出したことを特定できます4。は特定の KMS キー (3) でデータキーAWS KMSを作成しました5。

> **Note**
>
> 次のログエントリの例に示されているコールアウトの一部を表示するには、右にスクロールする必要があります。

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::086441151436:user/
AWSCloudTrail",(1)
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "userName": "AWSCloudTrail",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T21:15:33Z"
    }},
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:15:33Z",
  "eventSource":
 "kms.amazonaws.com",(2)
  "eventName":
 "GenerateDataKey",(3)
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAliasForCloudTrailKMS
 key",
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",(4)
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/AWSLogs/111122223333/"
    },
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "581f1f11-88b9-11e5-9c9c-595a1fb59ac0",
  "eventID": "3cdb2457-c035-4890-93b6-181832b9e766",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",(5)
    "accountId": "111122223333"
```

```
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

CloudTrail は S3 バケットにログファイルを配置します。

CloudTrail がログファイルを S3 バケットに入れるたびに、Amazon S3 は AWS KMS に代わってにGenerateDataKeyリクエストを送信します CloudTrail。このリクエストに応じて、AWS KMS は一意のデータキーを生成し、データキーの 2 つのコピーを Amazon S3 に送信します。1 つはプレーンテキストで、もう 1 つは指定された KMS キーで暗号化されます。Amazon S3 は、プレーンテキストデータキーを使用して CloudTrail ログファイルを暗号化し、使用後できるだけ早くプレーンテキストデータキーをメモリから削除します。Amazon S3 は、暗号化されたデータキーをメタデータとして暗号化された CloudTrail ログファイルに保存します。

GenerateDataKey リクエストには、暗号化コンテキストの次の情報が含まれています。

- CloudTrail 証跡の Amazon リソースネーム (ARN)
- S3 オブジェクトの ARN (CloudTrail ログファイル)

各GenerateDataKeyリクエストの結果、次の例のようなエントリが CloudTrail ログに記録されます。このようなログエントリが表示された場合は、 CloudTrail (1) が特定の証跡 () に対して AWS KMS (2) GenerateDataKeyオペレーション (3) を呼び出して、特定のログファイル () を保護するように決定できます5 。は、指定された KMS キー (4) の下にデータキーAWS KMSを作成しました。これは、同じログエントリで 6 2 回表示されます。

> **Note**
>
> 次のログエントリの例に示されているコールアウトの一部を表示するには、右にスクロールする必要があります。

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:i-34755b85",
    "arn": "arn:aws:sts::086441151436:assumed-role/AWSCloudTrail/
i-34755b85",(1)
    "accountId": "086441151436",
    "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-11-11T20:45:25Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::086441151436:role/AWSCloudTrail",
        "accountId": "086441151436",
        "userName": "AWSCloudTrail"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:15:58Z",
  "eventSource":
 "kms.amazonaws.com",(2)
  "eventName":
 "GenerateDataKey",(3)
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
```

```
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",④
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"⑤
    },
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",⑥
    "keySpec": "AES_256"
  },
  "responseElements": null,
  "requestID": "66f3f74a-88b9-11e5-b7fb-63d925c72ffe",
  "eventID": "7738554f-92ab-4e27-83e3-03354b1aa898",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",⑥
    "accountId": "111122223333"
  }],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333"
}
```

S3 バケットから暗号化されたログファイルを取得する

S3 バケットから暗号化された CloudTrail ログファイルを取得するたびに、Amazon S3 はAWS KMS ユーザーに代わって にDecryptリクエストを送信し、ログファイルの暗号化されたデータキーを復号します。このリクエストに応じて、AWS KMS は KMS キーを使用してデータキーを復号し、プレーンテキストのデータキーを Amazon S3 に送信します。Amazon S3 は、プレーンテキストのデータキーを使用して CloudTrail ログファイルを復号し、使用後できるだけ早くプレーンテキストのデータキーをメモリから削除します。

Decrypt リクエストには、暗号化コンテキストの次の情報が含まれています。

- CloudTrail 証跡の Amazon リソースネーム (ARN)
- S3 オブジェクトの ARN (CloudTrail ログファイル)

各Decryptリクエストの結果、次の例のようなエントリが CloudTrail ログに記録されます。このようなログエントリが表示された場合、AWS アカウント

(1))
のユーザーが AWS KMS
(2))
Decrypt オペレーション
(3))
を特定の追跡
(4))
および特定のログファイル
(5))
のために呼び出したことを特定できます。AWS KMS は特定の KMS キー
(6))
でデータキーを復号します。

> **Note**
>
> 次のログエントリの例に示されているコールアウトの一部を表示するには、右にスクロールする必要があります。

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::111122223333:role/cloudtrail-
admin",(1)
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "cloudtrail-admin",
    "sessionContext": {"attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2015-11-11T20:48:04Z"
    }},
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2015-11-11T21:20:52Z",
  "eventSource":
 "kms.amazonaws.com",(2)
```

```
  "eventName":
 "Decrypt",③
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:cloudtrail:arn": "arn:aws:cloudtrail:us-west-2:111122223333:trail/
Default",④
      "aws:s3:arn": "arn:aws:s3:::example-bucket-for-CT-logs/
AWSLogs/111122223333/CloudTrail/us-west-2/2015/11/11/111122223333_CloudTrail_us-
west-2_20151111T2115Z_7JREEBimdK8d2nC9.json.gz"⑤
    }
  },
  "responseElements": null,
  "requestID": "16a0590a-88ba-11e5-b406-436f15c3ac01",
  "eventID": "9525bee7-5145-42b0-bed5-ab7196a16daa",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",⑥
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

Amazon DynamoDB が AWS KMS を使用する方法

Amazon DynamoDB は、完全マネージド型のスケーラブルな NoSQL データベースサービスです。DynamoDB は AWS Key Management Service (AWS KMS) と統合され、サーバー側の暗号化機能である保管時の暗号化をサポートします。

保存時の暗号化, を使用すると、DynamoDB は、テーブルがディスクに永続化されるたびに、プライマリキー、ローカルセカンダリインデックスおよびグローバル セカンダリインデックス, など、DynamoDB テーブル内のすべての顧客データを透過的に暗号化します。(テーブルにソートキーが存在する場合、範囲の境界線を示すソートキーの一部が、プレーンテキスト形式でテーブルメタデータに保存されます。) テーブルにアクセスすると、DynamoDB はテーブルのデータを透過的に復号化します。暗号化されたテーブルの使用あるいは管理のためにアプリケーションを変更する必要はありません。

保管時の暗号化では、これらのオブジェクトが耐久性のあるメディアに保存されるたびに、DynamoDB Streams、グローバルテーブル、および バックアップ も保護されます。このトピックではテーブルを取り上げていますが、その内容はこれらのオブジェクトにも当てはまります。

すべての DynamoDB テーブルが暗号化されます。新規のテーブルでも既存のテーブルでも、暗号化を有効または無効にするオプションはありません。デフォルトでは、すべてのテーブルは DynamoDB サービスアカウントの AWS 所有のキー で暗号化されます。ただし、テーブルの一部またはすべてを暗号化するオプションを、カスタマーマネージドキーまたはアカウントの DynamoDB の AWS マネージドキー から選択できます。

Amazon DynamoDB による KMS キーのサポートの詳細については、『Amazon DynamoDB 開発者ガイド』の「保管時の DynamoDB の暗号化」を参照してください。

Amazon Elastic Block Store (Amazon EBS) が AWS KMS を使用する方法

このトピックでは、Amazon Elastic Block Store (Amazon EBS) が AWS KMS を使用してボリュームとスナップショットを暗号化する方法の詳細を説明します。Amazon EBS ボリュームの暗号化に関する基本的な手順については、「Amazon EBS 暗号化」を参照してください。

トピック

- Amazon EBS 暗号化
- KMS キーとデータキーを使用する
- Amazon EBS 暗号化コンテキスト
- Amazon EBS 障害の検出
- AWS CloudFormation を使用して、暗号化された Amazon EBS ボリュームを作成する

Amazon EBS 暗号化

暗号化された Amazon EBS ボリューム をサポートされている Amazon Elastic Compute Cloud (Amazon EC2) インスタンスタイプ にアタッチすると、ボリュームに保存されたデータ、ディスク I/O、ボリュームから作成されたスナップショットはすべて暗号化されます。暗号化は、Amazon EC2 インスタンスをホストするサーバーで行われます。

この機能は、すべての Amazon EBS ボリュームタイプでサポートされています。暗号化されたボリュームには、他のボリュームにアクセスする場合と同じ方法でアクセスできます。暗号化と復号化

は透過的に処理され、ユーザー、EC2 インスタンス、アプリケーションからの追加アクションは不要です。暗号化されたボリュームのスナップショットは自動的に暗号化され、暗号化されたスナップショットから作成されたボリュームも、自動的に暗号化されます。

EBS ボリュームの暗号化ステータスは、ボリュームの作成時に決定されます。既存のボリュームの暗号化ステータスを変更することはできません。ただし、暗号化されたボリュームと暗号化されていないボリューム間で データを移行 し、スナップショットのコピー中に新しい暗号化ステータスを適用できます。

Amazon EBS では、デフォルトでオプションの暗号化がサポートされています。AWS アカウントおよびリージョン内のすべての新しい EBS ボリュームとスナップショットコピーに対して自動的に暗号化を有効にすることができます。この構成設定は、既存のボリュームやスナップショットには影響しません。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドのデフォルトでの暗号化または Windows インスタンス用 Amazon EC2 ユーザーガイドを参照してください。

KMS キーとデータキーを使用する

暗号化 Amazon EBS ボリュームを作成するときは、AWS KMS key を指定します。デフォルトでは、Amazon EBS はアカウント (aws/ebs) の Amazon EBS 用 AWS マネージドキー を使用します。ただし、ユーザーは作成および管理するカスタマーマネージドキーを指定することができます。

カスタマーマネージドキーを使用するには、ユーザーに代わって KMS キーを使用する許可を Amazon EBS に付与する必要があります。必要なアクセス許可のリストについては、Linux インスタンス用 Amazon EC2 ユーザーガイドの IAM ユーザーのアクセス許可または Windows インスタンス用 Amazon EC2 ユーザーガイドを参照してください。

> **Important**
>
> Amazon EBS は、対称 KMS キーのみをサポートします。非対称 KMS キーを使用して Amazon EBS ボリュームを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

Amazon EBS では、ボリュームごとに指定した KMS キーで暗号化された一意のデータキーを生成するように AWS KMS に要求します。Amazon EBS は、暗号化されたデータキーをボリュームとともに保存します。次に、ボリュームを Amazon EC2 インスタンスにアタッチすると、Amazon EBS は AWS KMS を呼び出してデータキーを復号します。Amazon EBS は、ハイパーバイザーメモリ内のプレーンテキストデータキーを使用して、ボリュームへのすべてのディスク I/O を暗号化します。

詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの EBS 暗号化の仕組みまたは Windows インスタンス用 Amazon EC2 ユーザーガイドを参照してください。

Amazon EBS 暗号化コンテキスト

に対する GenerateDataKeyWithoutPlaintextおよび Decrypt リクエストではAWS KMS、Amazon EBS は、リクエスト内のボリュームまたはスナップショットを識別する名前と値のペアを持つ暗号化コンテキストを使用します。暗号化コンテキストの名前は変わりません。

暗号化コンテキスト は、一連のキー値のペアおよび任意非シークレットデータを含みます。データを暗号化するリクエストに暗号化コンテキストを組み込むと、AWS KMS は暗号化コンテキストを暗号化されたデータに暗号化してバインドします。データを復号するには、同じ暗号化コンテキストに渡す必要があります。

すべてのボリュームと Amazon EBS CreateSnapshotオペレーションで作成された暗号化されたスナップショットの場合、Amazon EBS はボリューム ID を暗号化コンテキスト値として使用します。CloudTrail ログエントリの `requestParameters` フィールドで、暗号化コンテキストは以下のようになります。

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

Amazon EC2 CopySnapshotオペレーションで作成された暗号化されたスナップショットの場合、Amazon EBS はスナップショット ID を暗号化コンテキスト値として使用します。 CloudTrail ログエントリの `requestParameters` フィールドで、暗号化コンテキストは以下のようになります。

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

Amazon EBS 障害の検出

暗号化された EBS ボリュームを作成するか、ボリュームを EC2 インスタンスにアタッチするには、Amazon EBS および Amazon EC2 インフラストラクチャで、EBS ボリュームの暗号化に指定した KMS キーを使用できる必要があります。KMS キーを使用できない場合 (キーステータスが `Enabled` ではない場合など)、ボリュームの作成またはボリュームのアタッチメントは失敗します。

この場合、Amazon EBS はイベントを Amazon EventBridge（ 以前の CloudWatch イベント) に送信して、障害を通知します。では EventBridge、これらのイベントに応じて自動アクションをトリ

ガーするルールを設定できます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EBS の Amazon CloudWatch イベント」、特に以下のセクションを参照してください。 Amazon EC2

- ボリュームのアタッチ時または再アタッチ時の無効な暗号化キー
- ボリューム作成時の無効な暗号化キー

これらの障害を修正するには、EBS ボリューム暗号化のために指定した KMS キーが有効になっていることを確認します。これを行うには、最初に KMS キーを表示して、現在のキーステータス (AWS Management Console の Status 列) を特定します。次に、以下のリンクのいずれかで情報を確認します。

- KMS キーのキーステータスが無効になっている場合は、有効にします。
- KMS キーのキーステータスがインポート保留中になっている場合は、キーマテリアルをインポートします。
- KMS キーのキーステータスが削除保留中になっている場合は、キーの削除をキャンセルします。

AWS CloudFormation を使用して、暗号化された Amazon EBS ボリュームを作成する

AWS CloudFormation を使用して、暗号化された Amazon EBS ボリュームを作成できます。詳細については、「AWS CloudFormation ユーザーガイド」の 「AWS::EC2::Volume」を参照してください。

Amazon Elastic Transcoder が AWS KMS を使用する方法

Amazon Elastic Transcoder を使用して、Amazon S3 バケットに格納されているメディアファイルを、コンシューマー再生デバイスで必要な形式に変換できます。入力ファイルと出力ファイルの両方を暗号化し復号することができます。以下のセクションでは、両方の処理で AWS KMS を使用する方法について説明します。

トピック

- 入力ファイルの暗号化
- 入力ファイルの復号
- 出力ファイルの暗号化

- HLS のコンテンツ保護
- Elastic Transcoder の暗号化コンテキスト

入力ファイルの暗号化

Elastic Transcoder を使用する前に、Amazon S3 バケットを作成し、そのバケットにメディアファイルをアップロードする必要があります。アップロード前に AES クライアント側の暗号化を使用するか、アップロード後に Amazon S3 サーバー側の暗号化を使用してファイルを暗号化できます。

AES を使用してクライアント側の暗号化を選択した場合、Amazon S3 にアップロードする前にファイルを暗号化する必要があり、また、暗号化キーへの Elastic Transcoder アクセスを提供する必要があります。これを行うには、対称 AWS KMS AWS KMS key を使用して、メディアファイルの暗号化に使用した AES 暗号化キーを保護します。

サーバー側の暗号化を選択した場合、Amazon S3 がユーザーに代わってすべてのファイルを暗号化および復号化することを許可します。Amazon S3 を設定し、3 種類の暗号化キーのうちの 1 つを使用して、ファイルを暗号化する一意のデータキーを保護できます。

- Amazon S3 キー。Amazon S3 が所有および管理する暗号化キー。AWS アカウント の一部ではありません。
- Amazon S3 の AWS マネージドキー、アカウントの一部である KMS キーは、AWS によって作成および管理されます。
- AWS KMS を使用して作成する任意の対称カスタマーマネージドキー

> **Important**
>
> クライアント側とサーバー側の両方の暗号化において、Elastic Transcoder は 対称 KMS キーのみをサポートします。非対称 KMS キー を使用して Elastic Transcoder ファイルを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

Amazon S3 コンソールまたは適切な Amazon S3 API を使用して、暗号化を有効にし、キーを指定できます。Amazon S3 による暗号化の実行方法の詳細については、「Amazon Simple Storage Service ユーザーガイド」の「KMS キー (SSE-KMS) でサーバー側の暗号化を使用してデータを保護する」を参照してください。

アカウントまたはカスタマーマネージドキーで Amazon S3 の AWS マネージドキー を使用して入力ファイルを保護する際、Amazon S3 と AWS KMS は次のようにやり取りします。

1. Amazon S3 は、プレーンテキストのデータキーおよび指定された KMS キーで暗号化されたデータキーのコピーをリクエストします。
2. AWS KMS はデータキーを作成し、指定された KMS キーで暗号化して、プレーンテキストデータキーと暗号化されたデータキーの両方を Amazon S3 に送信します。
3. Amazon S3 は、プレーンテキストデータキーを使用してメディアファイルを暗号化し、指定した Amazon S3 バケットにファイルを保存します。
4. Amazon S3 は、暗号化されたメディアファイルとともに暗号化されたデータキーを保存します。

入力ファイルの復号

入力ファイルを暗号化するために Amazon S3 サーバー側の暗号化を選択した場合、Elastic Transcoder はファイルを復号化しません。代わりに、Elastic Transcoder は、ジョブとパイプラインを作成するときに指定した設定に応じて、Amazon S3 を使用して復号を実行します。

設定は以下の組み合わせが可能です。

暗号化モード	AWS KMS キー	意味
S3	デフォルト値	Amazon S3 は、メディアファイルの暗号化と復号に使用されるキーを作成し、管理します。その処理をユーザーから見ることはできません。
S3-AWS-KMS	デフォルト値	Amazon S3 は、アカウント内のデフォルトの Amazon S3 用 AWS マネージドキー で暗号化されたデータキーを使用して、メディアファイルを暗号化します。
S3-AWS-KMS	カスタム (ARN 付き)	Amazon S3 は、指定された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

暗号化モード	AWS KMS キー	意味
		キーを使用して、メディアファイルを暗号化します。

S3-AWS-KMS が指定されている場合、Amazon S3 と AWS KMS は次のように連携して復号を実行します。

1. Amazon S3 は暗号化されたデータキーを AWS KMS に送信します。
2. AWS KMS は適切な KMS キーを使用してデータキーを復号し、プレーンテキストデータキーを Amazon S3 に返信します。
3. Amazon S3 は、プレーンテキストのデータキーを使用して暗号文を復号します。

AES キーを使用してクライアント側の暗号化を選択した場合、Elastic Transcoder は Amazon S3 バケットから暗号化されたファイルを取得し、復号します。Elastic Transcoder は、パイプラインの作成時に指定した KMS キーを使用して AES キーを復号し、AES キーを使用してメディアファイルを復号します。

出力ファイルの暗号化

Elastic Transcoder は、ジョブとパイプラインを作成するときに暗号化設定を指定する方法に応じて、出力ファイルを暗号化します。以下のオプションが利用できます。

暗号化モード	AWS KMS キー	意味
S3	デフォルト値	Amazon S3 は、出力ファイルの暗号化に使用するキーを作成および管理します。
S3-AWS-KMS	デフォルト値	Amazon S3 は、AWS KMS によって作成され、アカウントの Amazon S3 用 AWS マネージドキー によって暗号化されたデータキーを使用します。
S3-AWS-KMS	カスタム (ARN 付き)	Amazon S3 は、ARN で指定されたカスタマーマネージ

暗号化モード	AWS KMS キー	意味
		ドキーを使用して暗号化されたデータキーを使用して、メディアファイルを暗号化します。
AES-	デフォルト値	Elastic Transcoder は、アカウント内の Amazon S3 用 AWS マネージドキー を使用して指定された AES キーを復号し、そのキーを使用して出力ファイルを暗号化します。
AES-	カスタム (ARN 付き)	Elastic Transcoder は、ARN で指定されたカスタマーマネージドキーを使用して指定された AES キーを復号し、そのキーを使用して出力ファイルを暗号化します。

アカウント内の Amazon S3 用 AWS マネージドキー またはカスタマーマネージドキーを使用して出力ファイルを暗号化するように指定すると、Amazon S3 と AWS KMS は次のような方法でやり取りします。

1. Amazon S3 は、プレーンテキストのデータキーおよび指定された KMS キーで暗号化されたデータキーのコピーをリクエストします。
2. AWS KMS はデータキーを作成し、KMS キーで暗号化して、プレーンテキストデータキーと暗号化されたデータキーの両方を Amazon S3 に送信します。
3. Amazon S3 は、データキーを使用してメディアを暗号化し、指定した Amazon S3 バケットに格納します。
4. Amazon S3 は、暗号化されたデータキーを、暗号化されたメディアファイルとともに保存します。

指定した AES キーを出力ファイルの暗号化に使用するように指定した場合、AES キーは AWS KMS の KMS キーを使用して暗号化される必要があります。Elastic Transcoder、AWS KMS、ユーザーは次の方法で対話します。

1. AWS KMS API で Encrypt オペレーションを呼び出して AES キーを暗号化します。AWS KMS は、指定された KMS キーを使用してキーを暗号化します。パイプラインの作成時に使用する KMS キーを指定します。
2. Elastic Transcoder ジョブを作成するときに、暗号化された AES キーを含むファイルを指定します。
3. Elastic Transcoder は、AWS KMS API で Decrypt オペレーションを呼び出し、暗号化されたキーを暗号文として渡します。
4. Elastic Transcoder は、復号化された AES キーを使用して出力メディアファイルを暗号化し、復号化された AES キーをメモリから削除します。ジョブで定義した元のキーを暗号化したキーのみがディスクに保存されます。
5. ユーザーは、暗号化された出力ファイルをダウンロードして、定義した元の AES キーを使用してローカルにファイルを復号できます。

> **Important**
>
> AWS がプライベート暗号化キーを保存することはありません。したがって、キーを安全に管理することが重要です。キーを紛失すると、データを復号できなくなります。

HLS のコンテンツ保護

HTTP Live Streaming (HLS) は適応型のストリーミングプロトコルです。Elastic Transcoder は、入力ファイルを メディアセグメントと呼ばれる小さな個別のファイルに分割することで、HLS をサポートします。対応する個別のメディアセグメントには、異なるビットレートでエンコードされた同じ素材が含まれているため、プレーヤーは利用可能な帯域幅に最適なストリームを選択することができます。Elastic Transcoder は、ストリーミング可能なさまざまなセグメントのメタデータを含むプレイリストも作成します。

HLS のコンテンツ保護を有効にすると、各メディアセグメントは 128 ビットの AES 暗号化キーを使用して暗号化されます。再生中にコンテンツが表示されると、プレイヤーはキーをダウンロードし、メディアセグメントを復号します。

KMS キーとデータキーの 2 種類のキーが使用されます。データキーの暗号化と復号に使用する KMS キーを作成する必要があります。Elastic Transcoder は、データキーを使用してメディアセグメントを暗号化および復号します。データキーは AES-128 である必要があります。同じコンテンツのすべてのバリエーションとセグメントは、同じデータキーを使用して暗号化されます。データキーを指定するか、Elastic Transcoder にデータキーを作成させることができます。

KMS キーは、次のポイントでデータキーの暗号化に使用することができます。

- 独自のデータキーを指定する場合は、Elastic Transcoder に渡す前に暗号化する必要があります。
- Elastic Transcoder がデータキーを生成するように要求した場合、Elastic Transcoder はデータキーを暗号化します。

KMS キーは、次のポイントでデータキーの復号に使用することができます。

- Elastic Transcoder は、データキーを使用して出力ファイルを暗号化する必要がある場合、または入力ファイルを復号する必要がある場合に、指定されたデータキーを復号化します。
- Elastic Transcoder によって生成されたデータキーを復号化し、それを使用して出力ファイルを復号します。

詳細については、「Amazon Elastic Transcoder デベロッパーガイド」の「HLS のコンテンツ保護」を参照してください。

Elastic Transcoder の暗号化コンテキスト

暗号化コンテキスト は、一連のキー値のペアおよび任意非シークレットデータを含みます。データを暗号化するリクエストに暗号化コンテキストを組み込むと、AWS KMS は暗号化コンテキストを暗号化されたデータに暗号化してバインドします。データを復号するには、同じ暗号化コンテキストに渡す必要があります。

Elastic Transcoder は、すべての AWS KMS API リクエストで同じ暗号化コンテキストを使用して、データキーの生成、暗号化、復号を行います。

```
"service" : "elastictranscoder.amazonaws.com"
```

暗号化コンテキストは、特定の KMS AWS KMS キーがどのように使用されたかを理解するのに役立つように CloudTrail ログに書き込まれます。 CloudTrail ログファイルの `requestParameters`フィールドでは、暗号化コンテキストは次のようになります。

```
"encryptionContext": {
  "service" : "elastictranscoder.amazonaws.com"
}
```

Elastic Transcoder ジョブを設定して、サポートされる暗号化オプションに使用する方法の詳細については、Amazon Elastic Transcoder デベロッパーガイドのデータ暗号化のオプションを参照してください。

Amazon EMR が AWS KMS を使用する方法

Amazon EMR クラスターを使用する場合、永続的ストレージの場所に保存する前に、保管中のデータを暗号化するようにクラスターを設定できます。保存データは、EMR ファイルシステム (EMRFS) かクラスターノードのストレージボリューム、またはその両方で暗号化できます。保管中のデータを暗号化するには、AWS KMS key を使用します。以下のトピックでは、Amazon EMR クラスターが KMS キーを使用して保管中のデータを暗号化する方法について説明します。

> **Important**
>
> Amazon EMR は、対称 KMS キーのみをサポートします。非対称 KMS キーを使用して、Amazon EMR クラスター内の保管中のデータを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

Amazon EMR クラスターは、 転送中のデータも暗号化します。つまり、クラスターはネットワーク経由でデータを送信する前にデータを暗号化します。KMS キーを使用して送信中のデータを暗号化することはできません。詳細については、Amazon EMR 管理ガイドの転送時のデータ暗号化を参照してください。

Amazon EMR で使用できるすべての暗号化オプションの詳細については、Amazon EMR 管理ガイドの暗号化オプションを参照してください。

トピック

- EMR ファイルシステム (EMRFS) のデータを暗号化する
- クラスターノードのストレージボリュームのデータを暗号化する
- 暗号化コンテキスト

EMR ファイルシステム (EMRFS) のデータを暗号化する

Amazon EMR クラスターは、次の 2 つの分散ファイルシステムを使用します。

- Hadoop Distributed File System (HDFS) HDFS 暗号化は、AWS KMS で KMS キーを使用しません。
- EMR ファイルシステム (EMRFS) EMRFS は HDFS の実装で、Amazon EMR クラスターが Amazon Simple Storage Service (Amazon S3) にデータを格納できるようにします。EMRFS は、4 種類の暗号化オプションをサポートしており、そのうち 2 種類は AWS KMS で KMS キーを使用します。EMRFS 暗号化オプションの 4 種類すべての詳細については、Amazon EMR 管理ガイドの暗号化オプションを参照してください。

KMS キーを使用する 2 種類の EMRFS 暗号化オプションは、Amazon S3 が提供する次の暗号化機能を使用します。

- AWS Key Management Service によるサーバー側の暗号化 (SSE-KMS) を使用したデータの保護。Amazon EMR クラスターは、Simple Storage Service (Amazon S3) にデータを送信します。Amazon S3 は、KMS キーを使用してデータを暗号化してから、そのデータを S3 バケットに保存します。この仕組みについては、「SSE-KMS を使用して EMRFS のデータを暗号化するプロセス」を参照してください。
- クライアント側の暗号化を使用したデータの保護 (CSE-KMS)。Amazon EMR のデータは、AWS KMS key で暗号化されてから、ストレージのために Amazon S3 に送信されます。この仕組みについては、「CSE-KMS を使用して EMRFS のデータを暗号化するプロセス」を参照してください。

KMS を使用して EMRFS 上のデータを暗号化するように Amazon EMR クラスターを設定するときは、Amazon S3 または Amazon EMR クラスターで使用する KMS キーを選択します。SSE-KMS を使用して、エイリアス aws/s3 を持つ Amazon S3 の AWS マネージドキー、または作成する対称カスタマーマネージドキーを選択できます。クライアント側の暗号化では、作成する対称カスタマーマネージドキーを選択する必要があります。カスタマーマネージドキーを選択する際に、Amazon EMR クラスターに KMS キーの使用許可があることを確認する必要があります。詳細については、Amazon EMR 管理ガイドの Using AWS KMS keys for encryption を参照してください。

サーバー側の暗号化とクライアント側の暗号化のどちらの場合でも、選択する KMS キーがエンベロープ暗号化ワークフローのルートキーになります。データは、AWS KMS で KMS キーによって暗号化される一意のデータキーで暗号化されます。暗号化されたデータとその暗号化されたデータキーのコピーは、1 つの暗号化オブジェクトとして S3 バケットに一緒に保存されます。この仕組みについては、次のトピックを参照してください。

トピック

SSE-KMS を使用して EMRFS のデータを暗号化するプロセス

SSE-KMS を使用するように Amazon EMR クラスターを設定すると、暗号化プロセスは次のように動作します。

1. クラスターは、S3 バケットに格納するために Amazon S3 にデータを送信します。
2. Amazon S3 はAWS KMS、SSE-KMS を使用するようにクラスターを設定したときに選択した KMS キーのキー ID を指定して、 に GenerateDataKey リクエストを送信します。リクエストには暗号化コンテキストが含まれます。詳細については、「暗号化コンテキスト」を参照してください。
3. AWS KMS は、一意のデータ暗号化キー (データキー) を生成し、このデータキーの 2 つのコピーを Amazon S3 に送信します。コピーのうち一方は暗号化されない形式 (プレーンテキスト) で、もう一方は KMS キーで暗号化されます。
4. Amazon S3 は、プレーンテキストデータキーを使用してステップ 1 で受信したデータを暗号化し、使用後できるだけ早くプレーンテキストデータキーをメモリから削除します。
5. Amazon S3 は、暗号化されたデータとデータキーの暗号化されたコピーを、1 つの暗号化されたオブジェクトとして S3 バケットに格納します。

この復号プロセスは、次のように行われます。

1. クラスターは、暗号化されたデータオブジェクトを S3 バケットへ要求します。
2. Amazon S3 は、S3 オブジェクトから暗号化されたデータキーを抽出し、暗号化されたデータキーを Decrypt リクエストで AWS KMS に送信します。リクエストには暗号化コンテキストが含まれます。
3. AWS KMS は、暗号化に使用したのと同じ KMS キーを使用して、暗号化されたデータキーを復号し、復号した (プレーンテキスト) データキーを Amazon S3 に送信します。
4. Amazon S3 は、プレーンテキストデータキーを使用して暗号化されたデータを復号化し、使用後できるだけ早くプレーンテキストデータキーをメモリから削除します。
5. Amazon S3 は、復号化されたデータをクラスターに送信します。

CSE-KMS を使用して EMRFS のデータを暗号化するプロセス

CSE-KMS を使用するように Amazon EMR クラスターを設定すると、暗号化プロセスは次のように動作します。

1. Amazon S3 にデータを保存する準備ができたら、クラスターは に GenerateDataKeyリクエストを送信しAWS KMS、CSE-KMS を使用するようにクラスターを設定したときに選択した KMS キーのキー ID を指定します。リクエストには暗号化コンテキストが含まれます。詳細については、「暗号化コンテキスト」を参照してください。
2. AWS KMS は、一意のデータ暗号化キー (データ キー) を生成し、このデータキーの 2 つのコピーをクラスターに送信します。コピーのうち一方は暗号化されない形式 (プレーンテキスト) で、もう一方は KMS キーで暗号化されます。
3. クラスターは、プレーンテキストデータキーを使用してデータを暗号化し、使用後できるだけ早くそのプレーンテキストデータキーをメモリから削除します。
4. クラスターは、暗号化データと暗号化されたデータキーのコピーを 1 つの暗号化オブジェクトにまとめます。
5. クラスターは、暗号化されたオブジェクトを Amazon S3 に送信してストレージします。

この復号プロセスは、次のように行われます。

1. クラスターは、暗号化されたデータオブジェクトを S3 バケットへ要求します。
2. Amazon S3 は、暗号化されたオブジェクトをクラスターに送信します。
3. クラスターは、暗号化されたオブジェクトから暗号化されたデータキーを抽出し、暗号化されたデータキーを Decrypt リクエストで AWS KMS に送信します。リクエストには暗号化コンテキストが含まれます。
4. AWS KMS は、暗号化に使用したのと同じ KMS キーを使用して、暗号化されたデータキーを復号し、復号した (プレーンテキスト) データキーをクラスターに送信します。
5. クラスターは、そのプレーンテキストデータキーを使用して、暗号化されたデータを復号し、使用後できるだけ早くプレーンテキストデータキーをメモリから削除します。

クラスターノードのストレージボリュームのデータを暗号化する

Amazon EMR クラスターは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの集合です。クラスター内のインスタンスはそれぞれ、クラスターノードまたはノードと呼ばれます。各ノードには、インスタンスストアボリュームと Amazon Elastic Block Store (Amazon EBS) ボリューム

の 2 種類のストレージボリュームがあります。これらのノードでストレージボリュームをいずれも暗号化するには、クラスターを構成して、Linux Unified Key Setup (LUKS) を使用できます (各ノードの起動ボリュームは不可)。これは、ローカルディスクの暗号化と呼ばれます。

クラスター向けにローカルディスクの暗号化を有効にすると、AWS KMS の KMS キーを使用して LUKS キーを暗号化できます。作成したカスタマーマネージドキーを選択する必要があります。AWS マネージドキー を使用することはできません。カスタマーマネージドキーを選択する場合は、Amazon EMR クラスターに KMS キーを使用するアクセス許可があることを確認する必要があります。詳細については、Amazon EMR 管理ガイドの Using AWS KMS keys for encryption を参照してください。

KMS キーを使用してローカルディスクの暗号化を有効にする際、暗号化プロセスは次のようになります。

1. 各クラスターノードが起動すると、クラスターのローカルディスク暗号化を有効にしたときに選択した KMS キーのキー ID を指定してAWS KMS、 GenerateDataKey リクエストが に送信されます。
2. AWS KMS は、一意のデータ暗号化キー (データ キー) を生成し、このデータキーの 2 つのコピーをノードに送信します。コピーのうち一方は暗号化されない形式 (プレーンテキスト) で、もう一方は KMS キーで暗号化されます。
3. ノードでは、LUKS キーを保護するパスワードとして、base64 エンコードバージョンのプレーンテキストデータキーを使用します。ノードは、暗号化されたデータキーのコピーを起動ボリュームに保存します。
4. ノードが再起動すると、再起動したノードは暗号化されたデータキーを、Decrypt リクエストで AWS KMS に送信します。
5. AWS KMS は、暗号化に使用したのと同じ KMS キーを使用して、暗号化されたデータキーを復号し、復号した (プレーンテキスト) データキーをノードに送信します。
6. ノードでは、LUKS キーのロックを解除するパスワードとして、base64 エンコードバージョンのプレーンテキストデータキーを使用します。

暗号化コンテキスト

AWS KMS と統合されている各 AWS サービスでは、サービスが AWS KMS を使用してデータキーを生成したり、データを暗号化または復号したりするときに、暗号化 コンテキストを指定できます。暗号化コンテキストは、データの整合性を調べるために AWS KMS で使用される追加の認証情報です。サービスにおいて、暗号化オペレーションのために暗号化コンテキストを指定する際、復号

オペレーションと同じ暗号コンテキストを指定する必要があります。指定しない場合は復号できません。暗号化コンテキストは AWS CloudTrail ログファイルにも書き込まれるため、特定の KMS キーが使用された原因を理解するのに役立ちます。

以下のセクションでは、KMS キーを使用する Amazon EMR 暗号化の各シナリオで使用される暗号化コンテキストについて説明します。

SSE-KMS による EMRFS 暗号化の暗号化コンテキスト

SSE-KMS を使用すると、Amazon EMR クラスターは Amazon S3 にデータを送信し、次に Amazon S3 が KMS キーを使用してデータを暗号化してから S3 バケットに保存します。この場合、Amazon S3 は に送信する各 GenerateDataKeyおよび Decrypt リクエストで、S3 オブジェクトの Amazon リソースネーム (ARN) を暗号化コンテキストとして使用しますAWS KMS。次の例は、Amazon S3 が使用する暗号化コンテキストの JSON 表現を示しています。

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

CSE-KMS による EMRFS 暗号化の暗号化コンテキスト

CSE-KMS では、Amazon EMR クラスターは KMS キーを使用してデータを暗号化してから Amazon S3 に送信して保存します。この場合、クラスターは KMS キーの Amazon リソースネーム (ARN) を暗号化コンテキストとして使用し、各 GenerateDataKeyおよび に送信する Decrypt リクエストを使用しますAWS KMS。次の例では、クラスターが使用する暗号化コンテキストの JSON 表現を示します。

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

LUKS によるローカルディスク暗号化の暗号化コンテキスト

Amazon EMR クラスターが LUKS でローカルディスク暗号化を使用する場合、クラスターノードは、 に送信する GenerateDataKeyおよび Decrypt リクエストで暗号化コンテキストを指定しませんAWS KMS。

AWS Nitro Enclaves が AWS KMS を使用する方法

AWS KMS は、AWS Nitro Enclaves の暗号化アテステーションをサポートします。AWS Nitro Enclaves をサポートするアプリケーションは、エンクレーブの署名付きアテステーションドキュメントを使用して次の AWS KMS 暗号化オペレーションを呼び出します。これらの AWS KMS API

は、アテステーションドキュメントが Nitro Enclave からのものであることを確認します。次にこれらの API は、レスポンスでプレーンテキストデータを返す代わりに、アテステーションドキュメントのパブリックキーを使用してプレーンテキストを暗号化し、エンクレーブ内の対応するプライベートキーによってのみ復号できる暗号文を返します。

- Decrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

次の表は、Nitro Enclave リクエストへのレスポンスが各 API オペレーションの標準的なレスポンスとどのように異なるかを示しています。

AWS KMS オペレーション	標準的なレスポンス	AWS Nitro Enclaves のレスポンス
`Decrypt`	プレーンテキストデータを返します	アテステーションドキュメントのパブリックキーによって暗号化されたプレーンテキストデータを返します
`GenerateDataKey`	データキーのプレーンテキストのコピーを返します (KMS キーによって暗号化されたデータキーのコピーも返します)	アテステーションドキュメントのパブリックキーによって暗号化されたデータキーのコピーを返します (KMS キーによって暗号化されたデータキーのコピーも返します)
`GenerateDataKeyPair`	プライベートキーのプレーンテキストのコピーを返します (パブリックキーおよび KMS キーによって暗号化されたプライベートキーのコピーも返します)	アテステーションドキュメントのパブリックキーによって暗号化されたプライベートキーのコピーを返します (パブリックキーおよび KMS キーによって暗号化されたプ

AWS KMS オペレーション	標準的なレスポンス	AWS Nitro Enclaves のレスポンス
		ライベートキーのコピーも返します)
`GenerateRandom`	乱数バイト文字列を返します	アテステーションドキュメントのパブリックキーによって暗号化された乱数バイト文字列を返します

AWS KMS はポリシー条件キーをサポートします。このキーを使用し、アテステーションドキュメントの内容に基づく AWS KMS キーを使用したエンクレーブオペレーションを許可または拒否できます。また、AWS CloudTrail ログで Nitro Enclave の AWS KMS に対するリクエストを監視することもできます。

トピック

- Nitro Enclave のAWS KMS API を呼び出す方法
- AWS Nitro Enclaves の AWS KMS 条件キー
- Nitro Enclaves に対するリクエストの監視

Nitro Enclave のAWS KMS API を呼び出す方法

Nitro Enclaveの AWS KMS API を呼び出すには、リクエスト内の `Recipient` パラメーターを使用して、エンクレーブの署名付きアテステーションドキュメントと、エンクレーブのパブリックキーで使用する暗号化アルゴリズムを指定します。リクエストに `Recipient` パラメーターと署名付きアテステーションドキュメントが含まれている場合、レスポンスには `CiphertextForRecipient` フィールドとパブリックキーによって暗号化された暗号文が含まれます。プレーンテキストフィールドは null または空です。

`Recipient` パラメーターは、AWS Nitro Enclave からの署名付きアテステーションドキュメントを指定する必要があります。AWS KMS は、エンクレーブのアテステーションドキュメントのデジタル署名を使用して、リクエストのパブリックキーが有効なエンクレーブからのものであることを証明します。アテステーションドキュメントにデジタル署名をして独自の証明書を提供することはできません。

`Recipient` パラメーターを指定するには、AWS Nitro Enclaves SDK または任意の AWS SDK を使用します。AWS Nitro Enclaves SDK は、Nitro Enclave 内でのみサポートされており、`Recipient` パラメーターとその値をすべての AWS KMS リクエストに自動的に追加します。AWS SDK で Nitro Enclaves をリクエストするには、`Recipient` パラメーターとその値を指定する必要があります。AWS SDK での Nitro Enclave 暗号化アテステーションのサポートは、2023 年 3 月に導入されました。

AWS KMS はポリシー条件キーをサポートします。このキーを使用し、アテステーションドキュメントの内容に基づく AWS KMS キーを使用したエンクレーブオペレーションを許可または拒否できます。また、AWS CloudTrail ログで Nitro Enclave の AWS KMS に対するリクエストを監視することもできます。

`Recipient` パラメータと AWS `CiphertextForRecipient`レスポンスフィールドの詳細については、 AWS Key Management Service API リファレンス、Nitro Enclaves SDK、または任意の AWS SDK の Decrypt 、、、GenerateDataKeyGenerateDataKeyPairおよび GenerateRandomトピックを参照してください。 AWS暗号化のデータおよびデータキーの設定については、Using cryptographic attestation with AWS KMS を参照してください。

AWS Nitro Enclaves の AWS KMS 条件キー

AWS KMS リソースへのアクセスを制御するキーポリシーおよびIAM ポリシーで条件キーを指定できます。条件キーを含むポリシーステートメントは、その条件が満たされたときにのみ有効です。

AWS KMS には、リクエスト内の署名付きアテステーションドキュメントの内容に基づいてGenerateDataKeyPair、Decrypt 、GenerateDataKey、、および GenerateRandomオペレーションのアクセス許可を制限する条件キーが用意されています。これらの条件キーは、AWS KMS オペレーションのリクエストに `Recipient` パラメータと AWS Nitro Enclave からの有効なアテステーションドキュメントが含まれている場合にのみ機能します。`Recipient` パラメーターを指定するには、AWS Nitro Enclaves SDK または任意の AWS SDK を使用します。

エンクレーブ固有の AWS KMS 条件キーは、IAM コンソールまたは IAMサービス認証リファレンスに表示されない場合でも、キーポリシーステートメントおよび IAM ポリシーステートメントで有効です。

kms:RecipientAttestation：ImageSha384

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ

`kms:RecipientAttestation:ImageSha384`	文字列	単一値	`Decrypt`
			`GenerateDataKey`
			`GenerateDataKeyPair`
			`GenerateRandom`
				キーポリシーと IAM ポリシー

`kms:RecipientAttestation:ImageSha384` 条件キーは、リクエストの署名付きアテステーションドキュメントからのイメージダイジェストが条件キーの値と一致する場合、KMS キーを使用して `Decrypt`、`GenerateDataKey`、`GenerateDataKeyPair`、および `GenerateRandom` へのアクセスを制御します。`ImageSha384` 値は、アテステーションドキュメントの PCR0 に対応します。この条件キーは、リクエストの `Recipient` パラメータが AWS Nitro Enclave の署名付きアテステーションドキュメントを指定している場合にのみ有効です。

この値は、Nitro Enclaves の へのリクエストAWS KMSのCloudTrailイベントにも含まれます。

> **Note**
>
> この条件キーは、IAM コンソールまたは IAM サービス認証リファレンスに表示されない場合でも、キーポリシーステートメントおよび IAM ポリシーステートメントで有効です。

例えば、次のキーポリシーステートメントでは、`data-processing`ロールが Decrypt、、、GenerateDataKeyGenerateDataKeyPairおよび GenerateRandomオペレーションに KMS キーを使用することを許可します。`kms:RecipientAttestation:ImageSha384` 条件キーでは、リクエスト内のアテステーションドキュメントのイメージダイジェスト値 (PCR0) が条件内のイメージダイジェスト値と一致する場合にのみ、オペレーションを許可します。この条件キーは、リクエストの `Recipient` パラメータが AWS Nitro Enclave の署名付きアテステーションドキュメントを指定している場合にのみ有効です。

リクエストに AWS Nitro Enclave の有効なアテステーションドキュメントが含まれていない場合は、この条件が満たされないため、アクセス権限は拒否されます。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms:RecipientAttestation:PCR<PCR_ID>

AWS KMS 条件キー	条件の種類	値の型	API オペレーション	ポリシータイプ
kms:RecipientAttestation:PCR<PCR_ID>	文字列	単一値	Decrypt GenerateDataKey GenerateDataKeyPair GenerateRandom	キーポリシーと IAM ポリシー

`kms:RecipientAttestation:PCR<PCR_ID>` 条件キーは、リクエスト内の署名付きアテステーションドキュメントからのプラットフォーム設定登録 (PCR) が条件キーの PCR と一致する場合にのみ、KMS を使用して `Decrypt`、`GenerateDataKey`、`GenerateDataKeyPair`、および `GenerateRandom` へのアクセスを制御します。この条件キーは、リクエストの `Recipient` パラメータが AWS Nitro Enclave からの署名付きアテステーションドキュメントを指定している場合にのみ有効です。

この値は、Nitro Enclaves AWS KMSの へのリクエストを表すCloudTrailイベントにも含まれます。

> **Note**
>
> この条件キーは、IAM コンソールまたは IAM サービス認証リファレンスに表示されない場合でも、キーポリシーステートメントおよび IAM ポリシーステートメントで有効です。

PCR 値を指定するには、次の形式を使用します。PCR ID を条件キー名に連結します。PCR 値は、最大 96 バイトの小文字の 16 進文字列である必要があります。

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

たとえば、次の条件キーは PCR1 の特定の値を指定します。これは、エンクレーブとブートストラッププロセスに使用されるカーネルのハッシュに対応します。

```
kms:RecipientAttestation:PCR1:
 "0x1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcdef
```

次のキーポリシーステートメントの例では、`data-processing` ロールに Decrypt オペレーションでの KMS キーの使用を許可します。

このステートメントの `kms:RecipientAttestation:PCR` 条件キーでは、リクエスト内の署名付きアテステーションドキュメントの PCR1 値が条件の `kms:RecipientAttestation:PCR1` 値と一致した場合にのみ、オペレーションを許可します。`StringEqualsIgnoreCase` ポリシー演算子を使用して、PCR 値の大文字と小文字を区別しない比較を要求します。

リクエストにアテステーションドキュメントが含まれない場合は、この条件が満たされないため、アクセス許可は拒否されます。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
```

```
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
 "0x1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f15
    }
  }
}
```

Nitro Enclaves に対するリクエストの監視

AWS CloudTrail ログを使用して、 AWS Nitro Enclave の Decrypt 、GenerateDataKey、GenerateDataKeyPair、および GenerateRandomオペレーションをモニタリングできます。これらのログエントリの `additionalEventData` フィールドには、リクエスト内のアテステーションドキュメントからのモジュール ID (`attestationDocumentModuleId`)、イメージダイジェスト (`attestationDocumentEnclaveImageDigest`)、およびプラットフォーム設定登録 (PCR) を含む `recipient` フィールドがあります。これらのフィールドは、リクエストの `Recipient` パラメータが AWS Nitro Enclave からの署名付きアテステーションドキュメントを指定している場合にのみ含まれます。

モジュール ID は Nitro Enclave のエンクレーブ ID です。イメージダイジェストは、エンクレーブイメージの SHA384 ハッシュです。キーポリシーと IAM ポリシーの条件でイメージダイジェストおよび PCR 値を使用できます。PCR の詳細については、『AWS Nitro Enclaves ユーザーガイド』の「エンクレーブの測定値の入手方法」を参照してください。

このセクションでは、 に対するサポートされている Nitro Enclave リクエストの CloudTrail ログエントリの例を示しますAWS KMS。

Decrypt (エンクレーブ用)

次の例は、AWS Nitro Enclave の Decrypt オペレーションの AWS CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
    "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey（エンクレーブの場合）

次の例は、 AWS Nitro Enclave の GenerateDataKeyオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair（エンクレーブの場合）

次の例は、 AWS Nitro Enclave の GenerateDataKeyPairオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
```

```
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom（エンクレーブの場合）

次の例は、 AWS Nitro Enclave の GenerateRandomオペレーションの AWS CloudTrailログエントリを示しています。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
```

```
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Amazon Redshift が AWS KMS を使用する方法

このトピックでは、Amazon Redshift が AWS KMS を使用してデータを暗号化する方法について説明します。

トピック

- Amazon Redshift 暗号化
- 暗号化コンテキスト

Amazon Redshift 暗号化

Amazon Redshift データウェアハウスは、ノードと呼ばれるコンピューティングリソースの集合で、クラスターと呼ばれるグループに編成されています。各クラスターは Amazon Redshift エンジンを実行し、1 つ以上のデータベースを含みます。

Amazon Redshift は、暗号化に 4 階層のキーベースのアーキテクチャを使用します。アーキテクチャは、データ暗号化キー、データベースキー、クラスターキー、ルートキーで構成されます。ルートキーとして AWS KMS key を使用できます。

データ暗号化キーは、クラスター内のデータブロックを暗号化します。各データブロックに、ランダムに生成された AES-256 キーが割り当てられます。これらのキーは、クラスターのデータベースキーを使用して暗号化されます。

データベースキーは、クラスターのデータ暗号化キーを暗号化します。データベースキーは、ランダムに生成された AES-256 キーです。これは Amazon Redshift クラスターとは別のネットワークのディスクに保存され、安全なチャネルを介してクラスターに渡されます。

クラスターキーは、Amazon Redshift クラスターのデータベースキーを暗号化します。クラスターキーを管理するには、AWS KMS、AWS CloudHSM、または外部のハードウェアセキュリティモジュール (HSM) を使用できます。詳細については、 Amazon Redshift データベース暗号化 のドキュメントを参照してください。

暗号化をリクエストするには、Amazon Redshift コンソールで適切なチェックボックスをオンにします。暗号化ボックスの下に表示されるリストからカスタマーマネージドキーを 1 つ選択して、指定できます。カスタマーマネージドキーを指定しない場合、Amazon Redshift はアカウントで Amazon Redshift の AWS マネージドキー を使用します。

> **Important**
>
> Amazon Redshift は、対称暗号化 KMS キーのみをサポートします。Amazon Redshift 暗号化ワークフローでは、非対称KMS キーは使用できません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

暗号化コンテキスト

AWS KMS と統合された各サービスでは、データキー、暗号化および復号化をリクエストするときに暗号化コンテキストが指定されます。暗号化コンテキストは、AWS KMS が データの整合性をチェックするために使用する追加の認証 データ (AAD) です。つまり、暗号化オペレーションで暗号化コンテキストを指定すると、復号オペレーションでもそのコンテキストが指定され、指定しなかった場合、復号は成功しません。Amazon Redshift は、暗号化コンテキストにクラスター ID と作成時間を使用します。 CloudTrail ログファイルの `requestParameters`フィールドでは、暗号化コンテキストは次のようになります。

```
"encryptionContext": {
    "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",
    "aws:redshift:createtime": "20150206T1832Z"
},
```

CloudTrail ログ内のクラスター名を検索して、 AWS KMS key (KMS キー) を使用して実行されたオペレーションを理解できます。このオペレーションには、クラスターの暗号化、クラスターの復号、およびデータキーの生成が含まれます。

Amazon Relational Database Service (Amazon RDS) が AWS KMS を使用する方法

Amazon Relational Database Service (Amazon RDS) を使用して、クラウドでリレーショナルデータベースをセットアップ、運用、スケーリングできます。Amazon RDS リソースは AWS マネージドキー またはカスタマーマネージドキーで暗号化できます。Amazon RDS は、 Amazon Elastic Block Store (Amazon EBS) 暗号化 に基づいて構築され、データベースボリュームの完全なディスク暗号化を提供します。

Amazon RDS が KMS キーを使用してリソースを保護する方法の詳細については、「Amazon RDS ユーザーガイド」の「Amazon RDS リソースの暗号化」および「AWS KMS key 管理」を参照してください。

AWS Secrets Manager で AWS KMS を使用する方法

AWS Secrets Manager は、シークレットを暗号化して保存し、それらを透過的に復号してプレーンテキストで返す、AWS のサービスです。これは定期的に変更され、ハードコードしたり、アプリケーションにプレーンテキストで保存したりするべきではないアプリケーションシークレット (ログイン認証情報など) を保存するために専用に設計されています。ハードコードされた資格情報またはテーブル参照の代わりに、アプリケーションは Secrets Manager を呼び出します。

Secrets Manager は、一般的に使用されるデータベースに関連付けられているシークレットを定期的にローテーションする機能もサポートしています。常に、保存する前に新しく更新されたシークレットを暗号化します。

Secrets Manager は AWS Key Management Service (AWS KMS) と統合されており、AWS KMS key で保護された一意のデータキーを使用して、すべてのシークレット値の全バージョンを暗号化します。この統合により、AWS KMS を暗号化されないままにしない暗号化キーにより、シークレットが保護されます。また、KMS キーにカスタムアクセス許可を設定し、シークレットを保護するデータキーを生成、暗号化、復号するオペレーションを監査することができます。

Secrets Manager が KMS キーを使用してシークレットを保護する方法の詳細については、AWS Secrets Manager ユーザーガイドのシークレットの暗号化と復号を参照してください。

Amazon Simple Email Service (Amazon SES) が AWS KMS を使用する方法

Amazon Simple Email Service (Amazon SES) を使用して E メールを受信し、（オプションで）受信した E メールメッセージを暗号化してから、選択した Amazon Simple Storage Service (Amazon S3) バケットに保存することができます。Amazon SES が E メールメッセージを暗号化するように設定する際、Amazon SES がメッセージを暗号化する AWS KMS AWS KMS key を選択する必要があります。Amazon SES 用の AWS マネージドキー (エイリアスは aws/ses)、または AWS KMS で作成した対称カスタマーマネージドキーを選択できます。

> **Important**
>
> Amazon SES は、対称 KMS キーのみをサポートします。非対称 KMS キーを使用して Amazon SES E メールメッセージを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

Amazon SES を使用する E メールの受信方法の詳細については、Amazon Simple Email Service デベロッパーガイドの Amazon SES を使用して E メールを受信するを参照してください。

トピック

- AWS KMS を使用する Amazon SES 暗号化の概要
- Amazon SES 暗号化コンテキスト
- AWS KMS key を使用するためのアクセス許可を Amazon SES に付与する
- E メールメッセージの取得と復号

AWS KMS を使用する Amazon SES 暗号化の概要

S3 バケットに保存する前に E メールを受信し、E メールメッセージを暗号化するように Amazon SES を設定すると、プロセスは次のように動作します。

1. Amazon SES の受信ルールを作成し、S3 アクション、ストレージ用の S3 バケット、暗号化用の AWS KMS key を指定します。
2. Amazon SES は、受信ルールに一致する E メールメッセージを受信します。
3. Amazon SES は、該当する受信ルールで指定した KMS キーで暗号化された、一意のデータキーをリクエストします。

4. AWS KMS は、新しいデータキーを作成し、指定された KMS キーで暗号化してから、データキーの暗号化されたコピーとプレーンテキストのコピーを Amazon SES に送信します。
5. Amazon SES は、プレーンテキストデータキーを使用して E メールメッセージを暗号化し、使用後できるだけ早くプレーンテキストデータキーをメモリから削除します。
6. Amazon SES は、暗号化された E メールメッセージと暗号化されたデータキーを指定した S3 バケットに配置します。暗号化されたデータキーは、暗号化された E メールメッセージとともにメタデータとして保存されます。

Step 6 で Step 3 を実現するために、Amazon SES は AWS (提供された Amazon S3 暗号化クライアント) を使用します。同じクライアントを使用して、Amazon S3 から暗号化された E メールメッセージを取得し、復号化します。詳細については、「E メールメッセージの取得と復号」を参照してください。

Amazon SES 暗号化コンテキスト

Amazon SES が受信した E メールメッセージを暗号化するデータキーをリクエストすると (Step 3 で AWS KMS を使用する Amazon SES 暗号化の概要) 、リクエストに 暗号化コンテキスト が含まれます。暗号化コンテキストは、データの整合性を保証するために AWS KMS で使用される追加の認証データ (AAD) を提供します。また、暗号化コンテキストは AWS CloudTrail ログファイルにも書き込まれるため、特定の AWS KMS key (KMS キー) が使用された原因を理解するのに役立ちます。Amazon SES では、次の暗号化コンテキストが使用されます。

- E メールメッセージを受信するように Amazon SES を設定した AWS アカウント の ID
- E メールメッセージで S3 アクションを呼び出した Amazon SES 受信ルールのルール名
- E メールメッセージの Amazon SES メッセージ ID

次の例は、Amazon SES が使用する暗号化コンテキストの JSON 表現を示しています。

```
{
  "aws:ses:source-account": "111122223333",
  "aws:ses:rule-name": "example-receipt-rule-name",
  "aws:ses:message-id": "d6iitobk75ur44p8kdnnp7g2n800"
}
```

AWS KMS key を使用するためのアクセス許可を Amazon SES に付与する

E メールメッセージを暗号化するには、Amazon SES (aws/ses) にアカウントの AWS マネージドキーを使用する、または作成したカスタマーマネージドキーを使用できます。Amazon SES は、ユーザーの代わりに AWS マネージドキー を使用する許可をすでに持っています。ただし、S3 アクションを Amazon SES 受信ルールに追加する際にカスタマーマネージドキーを指定する場合は、KMS キーを使用して E メールメッセージを暗号化する許可を、Amazon SES に付与する必要があります。

Amazon SES にカスタマーマネージドキーの使用許可を付与するには、次のステートメントをその KMS キーのキーポリシーに追加します。

```
{
  "Sid": "Allow SES to encrypt messages using this KMS key",
  "Effect": "Allow",
  "Principal": {"Service": "ses.amazonaws.com"},
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:ses:rule-name": false,
      "kms:EncryptionContext:aws:ses:message-id": false
    },
    "StringEquals": {"kms:EncryptionContext:aws:ses:source-account": "ACCOUNT-ID-
WITHOUT-HYPHENS"}
  }
}
```

E メールメッセージを受信するように Amazon SES を設定した AWS アカウント の 12 桁の ID を、*ACCOUNT-ID-WITHOUT-HYPHENS* の代わりに置き換えます。このポリシーステートメントは以下の条件下でのみ、Amazon SES がこの KMS キーを使用してデータを暗号化することを許可します。

- Amazon SES では、AWS KMS API リクエストの EncryptionContext で aws:ses:rule-name と aws:ses:message-id を指定する必要があります。

- Amazon SES では、AWS KMS API リクエストの `EncryptionContext` で `aws:ses:source-account` を指定し、`aws:ses:source-account` の値がキーポリシーで指定された AWS アカウント ID と一致する必要があります。

Amazon SES が E メールメッセージを暗号化するときに使用する暗号化コンテキストの詳細については、「Amazon SES 暗号化コンテキスト」を参照してください。AWS KMS が暗号化テキストをどのように使用するかについての一般情報は、 暗号化コンテキストを参照してください。

E メールメッセージの取得と復号

Amazon SES には、暗号化された E メールメッセージを復号化する権限がないため、復号化できません。Amazon S3 から E メールメッセージを取得して復号化するには、コードを記述する必要があります。これを簡単にするには、Amazon S3 暗号化クライアントを使用します。以下の AWS SDK には、Amazon S3 暗号化クライアントが含まれます。

- AWS SDK for Java – AWS SDK for Java API リファレンスの AmazonS3EncryptionClient および AmazonS3EncryptionClientV2 を参照してください。
- AWS SDK for Ruby – AWS SDK for Ruby API リファレンスの Aws::S3::Encryption::Client を参照してください。
- AWS SDK for .NET – AWS SDK for .NET API リファレンスの AmazonS3EncryptionClient を参照してください。
- AWS SDK for Go – AWS SDK for Go API リファレンスの s3crypto を参照してください。

Amazon S3 暗号化クライアントは、暗号化された E メールメッセージを取得するための Amazon S3 へのリクエストの作成、メッセージの暗号化されたデータキーを復号するための AWS KMS へのリクエストの作成、および E メールメッセージの復号を簡素化します。例えば、暗号化されたデータキーを正常に復号するには、AWS KMS (AWS KMS を使用する Amazon SES 暗号化の概要の Step 3) からデータキーをリクエストするときに、Amazon SES が渡したのと同じ暗号化コンテキストを渡す必要があります。Amazon S3 暗号化クライアントは、この作業とその他の作業の多くを処理します。

AWS SDK for Java で Amazon S3 暗号化クライアントを使用して、クライアント側の復号を実行するサンプルコードについては、以下を参照してください。

- Amazon Simple Storage Service ユーザーガイドの「AWS KMS に保存されている KMS キーの使用」。
- AWS デベロッパーブログの Amazon S3 Encryption with AWS Key Management Service。

Amazon Simple Storage Service (Amazon S3) が AWS KMS を使用する方法

Amazon Simple Storage Service (Amazon S3) は、データをオブジェクトとしてバケットに保存するオブジェクトストレージサービスです。バケットとその中のオブジェクトはプライベートであり、アクセス許可を明示的に付与した場合にのみアクセスできます。

Amazon S3 は、Amazon S3 オブジェクトのサーバー側の暗号化を提供するために AWS Key Management Service (AWS KMS) と統合します。Amazon S3 は、Amazon S3 オブジェクトを暗号化するために AWS KMS キーを使用します。オブジェクトを保護するこれらの暗号化キーは、AWS KMS を暗号化されていない状態のままにすることはありません。この統合によって、AWS KMS キーにアクセス許可を設定し、シークレットを保護するデータキーを生成、暗号化、および復号するオペレーションを監査することも可能になります。

への Amazon S3 呼び出しの量を減らすにはAWS KMS、Amazon S3 内で期間限定で再利用される KMS キーで保護された Amazon S3 バケットキーを使用します。 key-encryption-keys Amazon S3 バケットキーは、AWS KMS リクエストのコストを最大 99% 削減できます。Amazon S3 バケットのすべてのオブジェクトのバケットキー、または Amazon S3 バケットの特定のオブジェクトのバケットキーを設定できます。

Amazon S3 が AWS KMS を使用する方法の詳細については、「Amazon S3 ユーザーガイド」の「KMS キーによるサーバー側の暗号化 (SSE-KMS) を使用したデータの保護」を参照してください。

AWS Systems Manager Parameter Store が AWS KMS を使用する方法

AWS Systems Manager Parameter Store を使用すると、Secure String パラメータを作成できます。これは、プレーンテキストのパラメータ名と暗号化されたパラメータ値を持つパラメータです。Parameter Store は、AWS KMS を使用してSecure String パラメータのパラメータ値を暗号化および復号します。

Parameter Store を使用すると、値を持つパラメータとしてデータを作成、格納、管理できます。Parameter Store でパラメータを作成し、設計するポリシーとアクセス許可の対象となる複数のアプリケーションおよびサービスで使用できます。パラメータ値を変更する必要がある場合は、多数のソースに対してエラーが発生しやすい変更を管理するのではなく、1 つのインスタンスを変更しま

す。Parameter Store は、パラメータ名の階層構造をサポートしているため、特定の用途に合わせてパラメータを修飾できます。

機密データを管理するために、セキュアな文字列パラメータを作成できます。Parameter Store は、AWS KMS keys を使用して、ユーザーが作成または変更した際に、Secure String パラメータのパラメータ値を暗号化します。また、アクセス時に KMS キーを使用してパラメータ値を復号します。Parameter Store がアカウント用に作成する AWS マネージドキー を使用するか、独自のカスタマーマネージドキーを指定できます。

> **Important**
>
> Parameter Store は、対称 KMS キーのみをサポートします。非対称 KMS キーを使用してパラメータを暗号化することはできません。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

Parameter Store は、スタンダードとアドバンストの 2 つのセキュリティで保護された文字列パラメータをサポートします。スタンダードパラメータは 4,096 バイトを上限とし、ユーザーが指定した KMS キーで直接、暗号化および復号されます。アドバンスト Secure String パラメータを暗号化および復号するために、Parameter Store は AWS Encryption SDK でエンベロープ暗号化を使用します。スタンダード Secure String パラメータをアドバンストパラメータに変換できますが、アドバンストパラメータをスタンダードパラメータに変換することはできません。スタンダードとアドバンスト Secure String パラメータの違いの詳細については、AWS Systems Manager ユーザーガイドの Systems Manager のアドバンストパラメータについてを参照してください。

トピック

- スタンダード Secure String パラメータの保護
- アドバンスト Secure String パラメータの保護
- パラメータ値を暗号化および復号するためのアクセス許可の設定
- Parameter Store の暗号化コンテキスト
- Parameter Store で KMS キーの問題をトラブルシューティングする

スタンダード Secure String パラメータの保護

Parameter Store は、暗号化オペレーションを実行しません。代わりに、AWS KMS に依存して、Secure String パラメータ値を暗号化および復号します。スタンダード Secure String パラメー

タ値を作成または変更すると、Parameter Store は AWS KMS Encrypt オペレーションを呼び出します。このオペレーションは、KMS キーを使用してデータキーを生成するのではなく、対称暗号化 KMS キーを直接使用してパラメータ値を暗号化します。

Parameter Store がパラメータ値を暗号化するために使用する KMS キーを選択できます。KMS キーを指定しない場合、Parameter Store は、Systems Manager がアカウントで自動的に作成する AWS マネージドキー を使用します。この KMS キー には aws/ssm エイリアスがあります。

アカウントのデフォルト KMS aws/ssm キーを表示するには、 AWS KMS API の DescribeKeyオペレーションを使用します。次の例では、aws/ssm エイリアス名を持つ AWS Command Line Interface (AWS CLI) の describe-key コマンドを使用します。

```
aws kms describe-key --key-id alias/aws/ssm
```

標準の Secure String パラメータを作成するには、Systems Manager API の PutParameterオペレーションを使用します。Tier パラメータを省略するか、デフォルトの値である Standard を指定します。値が SecureString の Type パラメータを含めます。KMS キーを指定するには、KeyId パラメータを使用します。デフォルトは、アカウントの AWS マネージドキー である aws/ssm です。

Parameter Store は KMS キーとプレーンテキストパラメータ値を使用して、AWS KMS Encrypt オペレーションを呼び出します。AWS KMS は暗号化されたパラメータ値を返します。Parameter Store はこの値をパラメータ名とともに保存します。

次の例では、AWS CLI で Systems Manager put-parameter コマンドとその --type パラメータを使用して、Secure String パラメータを作成します。コマンドではオプションの --tier パラメータと --key-id パラメータが省略されるため、Parameter Store はスタンダード Secure String パラメータを作成し、AWS マネージドキー で暗号化します。

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString
```

次の類似した例では、--key-id パラメータを使用してカスタマーマネージドキーを指定します。この例では、KMS キー ID を使用して KMS キーを識別しますが、任意の有効な KMS キー識別子を使用することができます。コマンドでは Tier パラメータ (--tier) を省略するため、Parameter Store では、アドバンストパラメータではなく、スタンダード Secure String パラメータが作成されます。

```
aws ssm put-parameter --name param1 --value "secret" --type SecureString --key-id
 1234abcd-12ab-34cd-56ef-1234567890ab
```

Parameter Store から Secure String パラメータを取得すると、その値は暗号化されます。パラメータを取得するには、Systems Manager API の GetParameter オペレーションを使用します。

次の例では、AWS CLI の Systems Manager get-parameter コマンドを使用して、値を復号せずに Parameter Store から `MyParameter` パラメータを取得します。

```
$  aws ssm get-parameter --name MyParameter

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

返す前にパラメータ値を復号するには、`GetParameter` の `WithDecryption` パラメータを `true` に設定します。`WithDecryption` を使用すると、Parameter Store はユーザーに代わって AWS KMS Decrypt オペレーションを呼び出し、パラメータ値を復号します。その結果、`GetParameter` リクエストは、次の例に示すように、プレーンテキストパラメータ値を持つパラメータを返します。

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

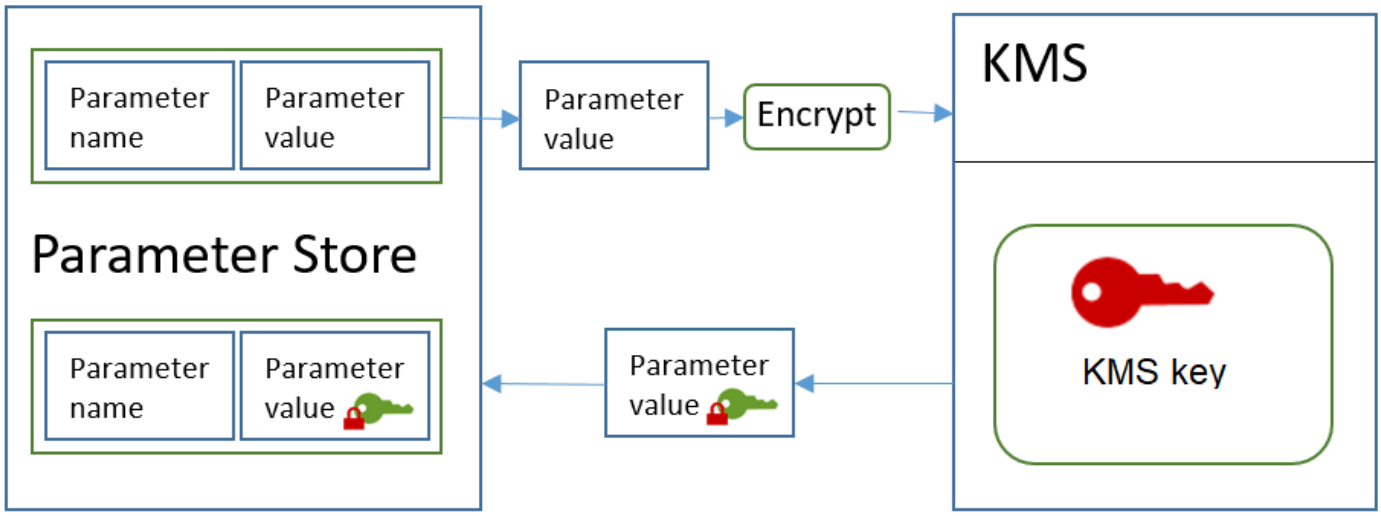
次のワークフローは、Parameter Store が KMS キーを使用してスタンダード Secure String パラメータを暗号化および復号する方法を示しています。

スタンダードパラメータの暗号化

1. `PutParameter` を使用して Secure String パラメータを作成すると、Parameter Store は AWS KMS に `Encrypt` リクエストを送信します。このリクエストには、プレーンテキストのパラメータ値、選択した KMS キー、Parameter Store の暗号化コンテキストが含まれます。AWS KMS へ

の送信時に、Secure String パラメータのプレーンテキスト値は Transport Layer Security (TLS) によって保護されます。

2. AWS KMS は、指定された KMS キーと暗号化コンテキストを使用して、パラメータ値を暗号化します。暗号文を Parameter Store に返します。Parameter Store には、パラメータ名とその暗号化された値が格納されます。



標準パラメータの復号

1. `GetParameter` リクエストに `WithDecryption` パラメータを含めると、Parameter Store は暗号化された Secure String パラメータ値と Parameter Store 暗号化コンテキストを使用して、AWS KMS に `Decrypt` リクエストを送信します。
2. AWS KMS は、同じ KMS キーと指定された暗号化コンテキストを使用して、暗号化された値を復号します。これは、プレーンテキスト (復号化された) パラメータ値を Parameter Store に返します。送信中、プレーンテキストのデータは TLS によって保護されます。
3. Parameter Store は、 `GetParameter` レスポンスでプレーンテキストのパラメータ値を返します。

アドバンスト Secure String パラメータの保護

`PutParameter` を使用してアドバンスト Secure String パラメータを作成する場合、Parameter Store は AWS Encryption SDK および対称暗号化 AWS KMS key によるエンベロープ暗号化を使用してパラメータ値を保護します。アドバンストパラメータ値は、それぞれ一意のデータキーで暗号化され、そのデータキーは KMS キーで暗号化されます。アカウントの AWS マネージドキー (`aws/ssm`) または任意のカスタマーマネージドキーを使用できます。

AWS Encryption SDK は、オープンソースのクライアント側ライブラリで、業界標準とベストプラクティスに沿ったデータの暗号化および復号に役立ちます。これは、複数のプラットフォームと、コマンドラインインターフェイスを含む複数のプログラミング言語でサポートされています。でソースコードを表示し、その開発に貢献できます GitHub。

Secure String パラメータ値ごとに、Parameter Store は を呼び出しAWS Encryption SDK、 AWS KMSが生成する一意のデータキー () を使用してパラメータ値を暗号化しますGenerateDataKey。AWS Encryption SDK は、暗号化されたパラメータ値と一意のデータキーの暗号化されたコピーを含む、暗号化されたメッセージを Parameter Store に返します。Parameter Store は、暗号化されたメッセージ全体を Secure String パラメータ値に格納します。次に、アドバンスト Secure String パラメータ値を取得すると、Parameter Store は AWS Encryption SDK を使用してパラメータ値を復号します。これには、AWS KMS を呼び出して、暗号化されたデータを復号する必要があります。

アドバンスト Secure String パラメータを作成するには、Systems Manager API の PutParameterオペレーションを使用します。`Tier` パラメータの値を `Advanced` に設定します。値が `SecureString` の `Type` パラメータを含めます。KMS キーを指定するには、`KeyId` パラメータを使用します。デフォルトは、アカウントの AWS マネージドキー である `aws/ssm` です。

```
aws ssm put-parameter --name MyParameter --value "secret_value" --type SecureString --
tier Advanced
```

次の類似した例では、`--key-id` パラメータを使用してカスタマーマネージドキーを指定します。例では、KMS キーの Amazon リソースネーム (ARN) を使用していますが、任意の有効な KMS キー識別子を使用することもできます。

```
aws ssm put-parameter --name MyParameter --value "secret_value"
 --type SecureString --tier Advanced --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

Parameter Store から Secure String パラメータを取得する場合、その値は AWS Encryption SDK から返される暗号化されたメッセージです。パラメータを取得するには、Systems Manager API の GetParameter オペレーションを使用します。

次の例では、Systems Manager `GetParameter` オペレーションを使用して、値を復号せずに Parameter Store から `MyParameter` パラメータを取得します。

```
$  aws ssm get-parameter --name MyParameter
```

```
{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value":
 "AQECAHgnOkMROh5LaLXkA4j0+vYi6tmM17Lg/9E464VRo68cvwAAAG8wbQYJKoZIhvcNAQcGoGAwXgIBADBZBgkqhkiG9
    }
}
```

返す前にパラメータ値を復号するには、GetParameter の WithDecryption パラメータを true に設定します。WithDecryption を使用すると、Parameter Store はユーザーに代わって AWS KMS Decrypt オペレーションを呼び出し、パラメータ値を復号します。その結果、GetParameter リクエストは、次の例に示すように、プレーンテキストパラメータ値を持つパラメータを返します。

```
$  aws ssm get-parameter --name MyParameter --with-decryption

{
    "Parameter": {
        "Type": "SecureString",
        "Name": "MyParameter",
        "Value": "secret_value"
    }
}
```

アドバンスト Secure String パラメータをスタンダードパラメータに変換することはできませんが、スタンダード Secure String パラメータをアドバンストパラメータに変換することはできます。スタンダード Secure String パラメータをアドバンスト Secure String に変換するには、Overwrite パラメータで PutParameter オペレーションを使用します。Type は SecureString、 Tier 値は Advanced である必要があります。カスタマーマネージドキーを識別する KeyId パラメータはオプションです。省略すると、Parameter Store はアカウントの AWS マネージドキー を使用します。スタンダードパラメータの暗号化に異なる KMS キーを使用した場合でも、プリンシパルが使用を許可されている任意の KMS キーを指定することができます。

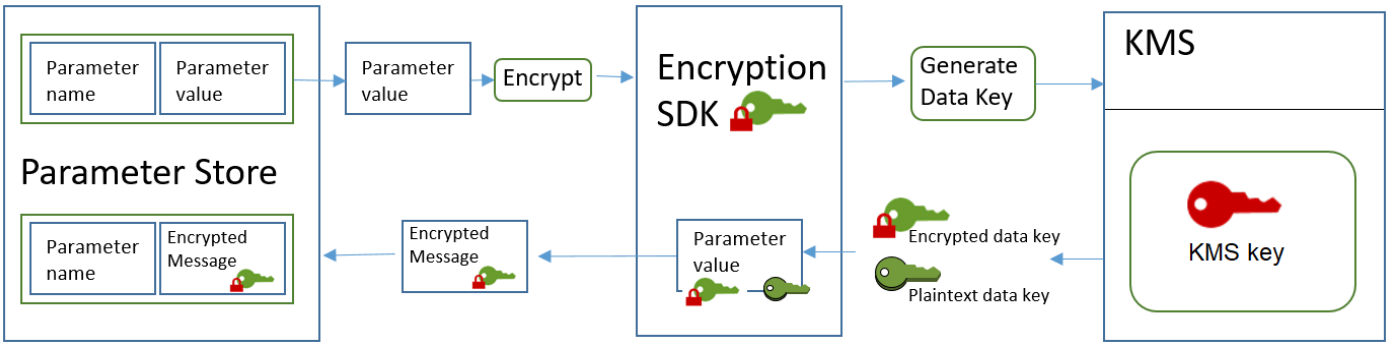
Overwrite パラメータを使用する際、Parameter Store は AWS Encryption SDK を使用してパラメータ値を暗号化します。次に、新しく暗号化されたメッセージを Parameter Store に格納します。

```
$ aws ssm put-parameter --name myStdParameter --value "secret_value"  --type
 SecureString --tier Advanced --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --overwrite
```

次のワークフローは、Parameter Store が KMS キーを使用して、アドバンスト Secure String パラメータを暗号化および復号する方法を示しています。

アドバンストパラメータの暗号化

1. `PutParameter` を使用してアドバンスト Secure String パラメータを作成すると、Parameter Store は AWS Encryption SDK と AWS KMS を使用してパラメータ値を暗号化します。Parameter Store は、パラメータ値、指定した KMS キー、Parameter Store の暗号化コンテキストを使用して AWS Encryption SDK を呼び出します。
2. は、指定した KMS キーの識別子と Parameter Store 暗号化コンテキストAWS KMSを使用して、にGenerateDataKeyリクエストAWS Encryption SDKを送信します。 は、一意のデータキーの 2 つのコピーAWS KMSを返します。1 つはプレーンテキストで、もう 1 つは KMS キーで暗号化されます。(暗号化コンテキストは、データキーを暗号化するときに使用します)。
3. AWS Encryption SDK はプレーンテキストのデータキーを使ってパラメータ値を暗号化し、暗号化されたパラメータ値、暗号化されたデータキー、および Parameter Store 暗号化コンテキストを含むその他のデータを含む暗号化されたメッセージを返します。
4. Parameter Store は、暗号化されたメッセージをパラメータ値として格納します。



アドバンストパラメータの復号

1. `GetParameter` リクエストに `WithDecryption` パラメータを含めると、アドバンスト Secure String パラメータを取得できます。これを行うと、Parameter Store は暗号化されたメッセージをパラメータ値から AWS Encryption SDK の復号メソッドに渡します。
2. AWS Encryption SDK は AWS KMS Decrypt オペレーションを呼び出します。暗号化されたメッセージから、暗号化されたデータキーと Parameter Store 暗号化コンテキストを渡します。
3. AWS KMS は、KMS キーと Parameter Store の暗号化コンテキストを使用して、暗号化されたデータキーを復号します。続いて、プレーンテキストの (復号化された) データキーを AWS Encryption SDK に返します。

4. AWS Encryption SDK は、プレーンテキストのデータキーを使ってパラメータ値を復号し、これは、プレーンテキストのパラメータ値を Parameter Store に返します。
5. Parameter Store は暗号化コンテキストを検証し、 `GetParameter` 応答でプレーンテキストのパラメータ値を返します。

パラメータ値を暗号化および復号するためのアクセス許可の設定

スタンダードの Secure String パラメータ値を暗号化するには、ユーザーに `kms:Encrypt` 権限が必要です。アドバンスト Secure String パラメータ値を暗号化するには、ユーザーに `kms:GenerateDataKey` 権限が必要です。どちらのタイプの Secure String パラメータ値を復号するにも、ユーザーに `kms:Decrypt` 権限が必要です。

IAM ポリシーを使用して、ユーザーが Systems Managerの `PutParameter` と `GetParameter` オペレーションを呼び出すアクセス許可を許可または拒否できます。

カスタマーマネージドキーを使用して Secure String パラメータ値を暗号化している場合は、IAM ポリシーとキーポリシーを使用して、暗号化と復号のアクセス許可を管理できます。ただし、デフォルトの `aws/ssm` KMS キーに対してアクセス制御ポリシーを確立することはできません。カスタマーマネージドキーへのアクセスを制御する方法の詳細については、AWS KMS の認証とアクセスコントロール を参照してください。

次の例は、スタンダードの Secure String パラメータ用に設計された IAM ポリシーを示しています。これにより、ユーザーは `FinancialParameters` パス内のすべてのパラメータに対して Systems Manager `PutParameter` オペレーションを呼び出すことができます。このポリシーにより、ユーザーはサンプルカスタマーマネージドキーで AWS KMS `Encrypt` オペレーションを呼び出すこともできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/
FinancialParameters/*"
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

次の例は、アドバンスト Secure String パラメータ用に設計された IAM ポリシーを示しています。これにより、ユーザーは `ReservedParameters` パス内のすべてのパラメータに対して Systems Manager `PutParameter` オペレーションを呼び出すことができます。このポリシーにより、ユーザーはサンプルカスタマーマネージドキーで AWS KMS `GenerateDataKey` オペレーションを呼び出すこともできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/
ReservedParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

最後の例では、スタンダードまたはアドバンスト Secure String パラメータに使用できる IAM ポリシーも示しています。これにより、ユーザーは `ITParameters` パス内のすべてのパラメータに対して Systems Manager の `GetParameter` オペレーション (および関連する操作) を呼び出すこと

ができます。このポリシーにより、ユーザーはサンプルカスタマーマネージドキーで AWS KMS Decrypt オペレーションを呼び出すこともできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/ITParameters/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Parameter Store の暗号化コンテキスト

暗号化コンテキスト は、一連のキー値のペアおよび任意非シークレットデータを含みます。データを暗号化するリクエストに暗号化コンテキストを組み込むと、AWS KMS は暗号化コンテキストを暗号化されたデータに暗号化してバインドします。データを復号するには、同じ暗号化コンテキストに渡す必要があります。

また、暗号化コンテキストを使用して、監査レコードおよびログ内の暗号化オペレーションを識別することもできます。暗号化コンテキストは、AWS CloudTrail ログなどのログにプレーンテキストで表示されます。

AWS Encryption SDK でも暗号化コンテキストを使用しますが、処理方法は異なります。Parameter Store は、暗号化メソッドに暗号化コンテキストを提供します。AWS Encryption SDK は、暗号化コンテキストを暗号化されたデータに暗号化してバインドします。また、暗号化されたメッセージを返す際、ヘッダーに暗号化コンテキストをプレーンテキストとして含めます。ただし、AWS KMS とは異なり、AWS Encryption SDK の復号メソッドは、入力として暗号化コンテキストを取りません。代わりに、データを復号するときに、AWS Encryption SDK は暗号化されたメッセージから暗号化コン

テキストを取得します。Parameter Store は、プレーンテキストのパラメータ値を返す前に、暗号化コンテキストに期待される値が含まれていることを確認します。

Parameter Store は、暗号化オペレーションで次の暗号化コンテキストを使用します。

- キー: PARAMETER_ARN
- 値: 暗号化されるパラメータの Amazon リソースネーム (ARN)。

暗号化コンテキストの形式は以下のとおりです。

```
"PARAMETER_ARN":"arn:aws:ssm:<REGION_NAME>:<ACCOUNT_ID>:parameter/<parameter-name>"
```

例えば、Parameter Store は 呼び出しにこの暗号化コンテキストを含めて、サンプルの AWS アカウント およびリージョンで MyParameter パラメータを暗号化および復号します。

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
```

パラメータが Parameter Store 階層パスにある場合、パスと名前は暗号化コンテキストに含まれます。例えば、サンプルの AWS アカウント およびリージョンの /ReadableParameters パスで MyParameter パラメータを暗号化および復号するときに、この暗号化コンテキストが使用されます。

```
"PARAMETER_ARN":"arn:aws:ssm:us-west-2:111122223333:parameter/ReadableParameters/
MyParameter"
```

暗号化された Secure String パラメータ値を復号するには、正しい暗号化コンテキストと Systems Manager AWS KMS オペレーションが返す暗号化されたパラメータ値を使用して、GetParameter Decrypt オペレーションを呼び出します。ただし、WithDecryption パラメータで GetParameter オペレーションを使用して、Parameter Store パラメータ値を復号することをお勧めします。

暗号化コンテキストを IAM ポリシーに含めることもできます。例えば、ユーザーが特定の 1 つのパラメータ値またはパラメータ値のセットのみを復号できるようにすることができます。

次の IAM ポリシーステートメントの例では、ユーザーが MyParameter パラメータの get 値と、指定した KMS キーを使用してその値を復号できるようにします。ただし、アクセス許可は、暗号化コンテキストが指定された文字列と一致する場合にのみ適用されます。これらのアクセス許可は他のパラメータや KMS キーには適用されません。また、暗号化コンテキストが文字列と一致しない場合、GetParameter への呼び出しは失敗します。

このようなポリシーステートメントを使用するときは、事前にサンプルの ARN を有効な値で置き換えてください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "arn:aws:ssm:us-west-2:111122223333:parameter/MyParameter"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Condition": {
                "StringEquals": {
                    "kms:EncryptionContext:PARAMETER_ARN":"arn:aws:ssm:us-
west-2:111122223333:parameter/MyParameter"
                }
            }
        }
    ]
}
```

Parameter Store で KMS キーの問題をトラブルシューティングする

Secure String パラメータに対して任意のオペレーションを実行するには、Parameter Store が目的のオペレーションに対して、指定した AWS KMS KMS キーを使用できる必要があります。KMS キーに関連する Parameter Store の障害のほとんどは、次の問題に起因します。

- アプリケーションが使用している認証情報には、KMS キーで指定されたアクションを実行するアクセス許可はありません。

  このエラーを解決するには、異なる認証情報を使用してアプリケーションを実行するか、オペレーションを妨げている IAM またはキーポリシーを修正します。AWS KMS IAM とキーポリシーのヘルプについては、「AWS KMS の認証とアクセスコントロール」を参照してください。

- KMS キーが見つからない。

  これは通常、KMS キーに誤った識別子を使用した場合に発生します。KMS キーの正しい識別子を見つけて、再度コマンドを試行します。
- KMS キーが有効になっていない。これが発生すると、Parameter Store は からの詳細な工ラーメッセージを含むInvalidKeyId例外を返しますAWS KMS。KMS キーのキーステータスが `Disabled` の場合は、有効にします。`Pending Import` の場合は、インポート手順を実施してください。キーステータスが `Pending Deletion` の場合は、キーの削除をキャンセルするか、別の KMS キーを使用します。

  AWS KMS コンソールで KMS キーのキーステータスを確認するには、カスタマーマネージドキーまたは AWS マネージドキー ページのステータス列を参照してください。AWS KMS API を使用して KMS キーのステータスを確認するには、 DescribeKeyオペレーションを使用します。

Amazon が WorkMail を使用する方法 AWS KMS

このトピックでは、Amazon が WorkMail AWS KMSを使用して E メールメッセージを暗号化する方法について説明します。

トピック

- Amazon WorkMail の概要
- Amazon WorkMail 暗号化
- KMS キーの使用を許可する
- Amazon WorkMail 暗号化コンテキスト
- との Amazon WorkMail インタラクションのモニタリング AWS KMS

Amazon WorkMail の概要

Amazon WorkMail は、既存のデスクトップおよびモバイル E メールクライアントをサポートする、安全でマネージド型のビジネス E メールおよびカレンダーサービスです。Amazon WorkMail 組織を作成し、所有している 1 つ以上の E メールドメインを割り当てることができます。その後、組織内の E メールユーザーとディストリビューショングループのメールボックスを作成できます。

Amazon は、メッセージがディスクに書き込まれる前に、すべての Amazon WorkMail 組織のメールボックス内のすべてのメッセージを WorkMail 透過的に暗号化し、ユーザーがメッセージにアクセスするときにメッセージを透過的に復号します。暗号化を無効にするオプションはありません。

メッセージを保護する暗号化キーを保護するために、Amazon WorkMail は AWS Key Management Service () と統合されていますAWS KMS。

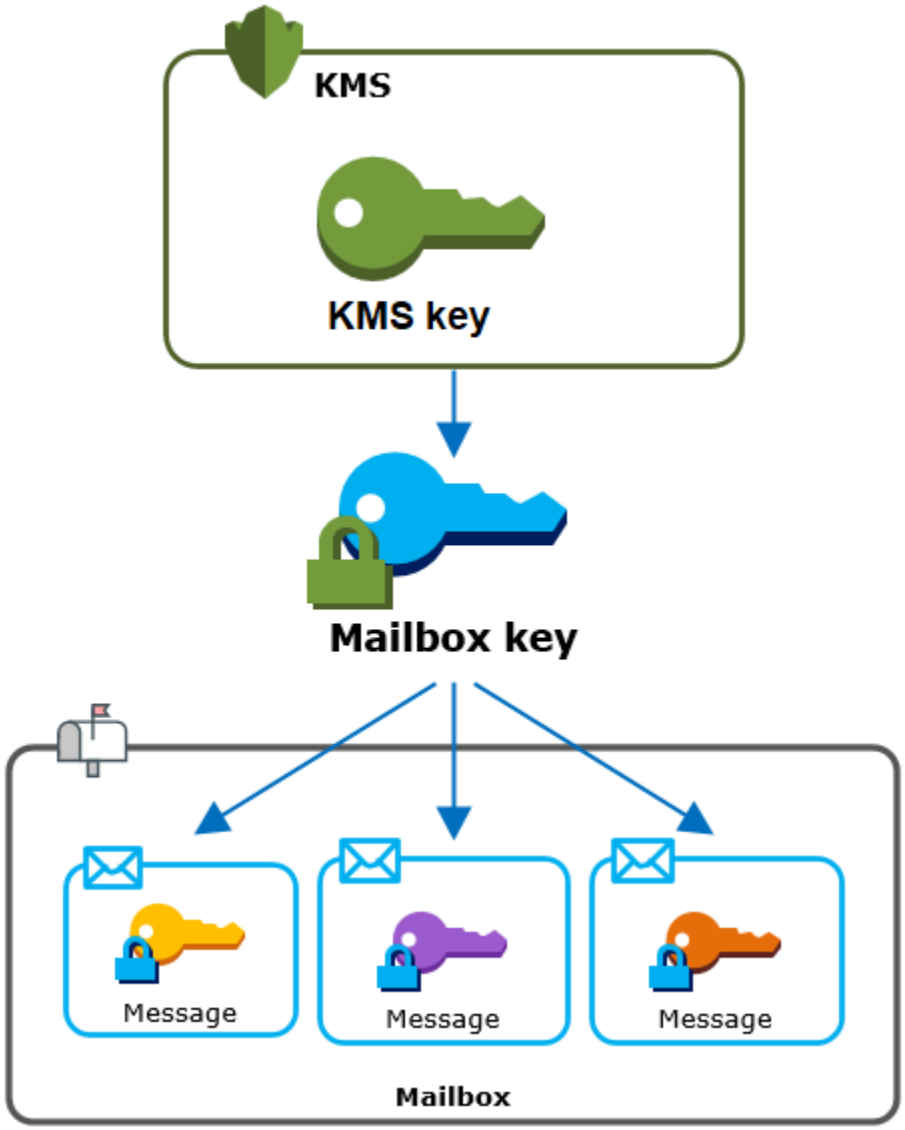
Amazon には、ユーザーが署名付きまたは暗号化された E メールを送信できるようにするオプション WorkMail もあります。この暗号化機能は AWS KMS を使用していません。

Amazon WorkMail 暗号化

Amazon では WorkMail、各組織には、組織内のユーザーごとに 1 つずつ、複数のメールボックスを含めることができます。E メール、カレンダーの項目などのすべてのメッセージはユーザーのメールボックスに保存されます。

Amazon WorkMail 組織内のメールボックスの内容を保護するために、Amazon はすべてのメールボックスメッセージをディスクに書き込む前に WorkMail 暗号化します。お客様から提供された情報がプレーンテキストで保存されることはありません。

各メッセージは、一意のデータ暗号化キーで暗号化されます。メッセージキーは、そのメールボックスでのみ使用される一意の暗号化キーであるメールボックスキーで保護されています。メールボックスキーは、組織の AWS KMS key で暗号化されるため、AWS KMS を暗号化されないままにしません。次の図表は、AWS KMS における、暗号化されたメッセージ、暗号化されたメッセージキー、暗号化されたメールボックスキー、組織の KMS キーの関係を示しています。



組織の KMS キー

Amazon WorkMail 組織を作成するときに、組織の AWS KMS keyを選択できます。この KMS キーはその組織内のすべてのメールボックスキーを保護します。

高速セットアップ手順を使用して組織を作成する場合、Amazon WorkMail は の Amazon WorkMail (`aws/workmail`) AWS マネージドキー用 を使用しますAWS アカウント。標準セットアップ を使用する場合は、Amazon AWS マネージドキー 用の 、 WorkMail または所有および管理するカスタマーマネージドキーを選択できます。各組織で同じ、または異なる KMS キーを選択できますが、選択した KMS キーを変更することはできません。

> **Important**
>
> Amazon は、対称暗号化 KMS キーのみ WorkMail をサポートしています。非対称 KMS キーを使用して Amazon のデータを暗号化することはできません WorkMail。KMS キーが対称か非対称かを判断する方法については、非対称 KMS キーの識別 を参照してください。

組織の KMS キーを検索するには、AWS KMS への呼び出しを記録する AWS CloudTrail ログエントリを使用します。

各メールボックスの一意の暗号化キー

新しいメールボックスを作成すると、Amazon WorkMail はメールボックスの一意の 256 ビット Advanced Encryption Standard (AES) 対称暗号化キーを生成します。これはメールボックスキー と呼ばれますAWS KMS。Amazon WorkMail はメールボックスキーを使用して、メールボックス内の各メッセージの暗号化キーを保護します。

メールボックスキーを保護するために、Amazon は を WorkMail 呼び出しAWS KMSて組織の KMS キーでメールボックスキーを暗号化します。その後、メールボックスのメタデータに暗号化されたメールボックスキーを保存します。

> **Note**
>
> Amazon WorkMail は、対称メールボックス暗号化キーを使用してメッセージキーを保護します。以前は、Amazon は各メールボックスを非対称キーペアで WorkMail 保護していました。パブリックキーを使用して各メッセージキーを暗号化し、プライベートキーで復号していました。プライベートメールボックスキーは組織の KMS キーで保護されていました。既存のメールボックスは今も非対称メールボックスキーペアを使用している場合があります。この変更により、メールボックスやそのメッセージのセキュリティに影響が生じることはありません。

各メッセージの一意の暗号化キー

メールボックスにメッセージを追加すると、Amazon は の外部でメッセージの一意の 256 ビット AES 対称暗号化キー WorkMail を生成しますAWS KMS。このメッセージキーを使用してメッセージを暗号化します。Amazon はメールボックスキーでメッセージキーを WorkMail 暗号化し、暗号化されたメッセージキーをメッセージとともに保存します。次に、組織の KMS キーでメールボックスキーを暗号化します。

新しいメールボックスの作成

Amazon は、新しいメールボックス WorkMail を作成するときに、次のプロセスを使用して、暗号化されたメッセージを保持するメールボックスを準備します。

- Amazon は、 の外部にあるメールボックス用に一意の 256 ビット AES 対称暗号化キー WorkMail を生成しますAWS KMS。
- Amazon は Encrypt AWS KMS オペレーションを WorkMail 呼び出します。また、メールボックスキーおよび組織の AWS KMS key の識別子を渡します。AWS KMS は KMS キーで暗号化されたメールボックスキーの暗号化テキストを返します。
- Amazon は、暗号化されたメールボックスキーをメールボックスメタデータとともに WorkMail 保存します。

メールボックスメッセージの暗号化

メッセージを暗号化するために、Amazon では以下のプロセス WorkMail を使用します。

1. Amazon はメッセージの一意の 256 ビット AES 対称キー WorkMail を生成します。プレーンテキストのメッセージキーと Advanced Encryption Standard (AES) アルゴリズムを使用して、AWS KMS の外部でメッセージを暗号化します。
2. メールボックスキーでメッセージキーを保護するために、Amazon はメールボックスキーを復号 WorkMail する必要があります。メールボックスキーは常に暗号化された形式で保存されます。

   Amazon は AWS KMS Decrypt オペレーションを WorkMail 呼び出し、暗号化されたメールボックスキーを渡します。 AWS KMSは組織の KMS キーを使用してメールボックスキーを復号し、プレーンテキストのメールボックスキーを Amazon に返します WorkMail。
3. Amazon WorkMail は、プレーンテキストのメールボックスキーと Advanced Encryption Standard (AES) アルゴリズムを使用して、 の外部でメッセージキーを暗号化しますAWS KMS。
4. Amazon WorkMail は、暗号化されたメッセージキーを暗号化されたメッセージのメタデータに保存し、復号化できるようにします。

メールボックスメッセージの復号

メッセージを復号化するために、Amazon は次のプロセス WorkMail を使用します。

1. Amazon は AWS KMS Decrypt オペレーションを WorkMail 呼び出し、暗号化されたメールボックスキーを渡します。 AWS KMSは組織の KMS キーを使用してメールボックスキーを復号し、プレーンテキストのメールボックスキーを Amazon に返します WorkMail。
2. Amazon WorkMail は、プレーンテキストのメールボックスキーと Advanced Encryption Standard (AES) アルゴリズムを使用して、 の外部で暗号化されたメッセージキーを復号しますAWS KMS。
3. Amazon WorkMail は、プレーンテキストのメッセージキーを使用して、暗号化されたメッセージを復号します。

メールボックスキーのキャッシュ

パフォーマンスを向上させ、 への呼び出しを最小限に抑えるためにAWS KMS、Amazon は各クライアントのプレーンテキストのメールボックスキーを最大 1 分間ローカルに WorkMail キャッシュします。キャッシュ期間の終了時に、メールボックスキーは削除されます。キャッシュ期間中にそのクライアントのメールボックスキーが必要な場合、Amazon は を呼び出す代わりにキャッシュからキーを取得 WorkMail できますAWS KMS。メールボックスキーはキャッシュで保護されており、プレーンテキストでディスクに書き込まれることはありません。

KMS キーの使用を許可する

Amazon は、暗号化オペレーションAWS KMS keyで WorkMail を使用する場合、メールボックス管理者に代わって動作します。

ユーザーに代わってシークレットに AWS KMS key を使用するには、管理者に次のアクセス許可が必要です。IAM ポリシーまたはキーポリシーで、これらの必要なアクセス許可を指定できます。

- `kms:Encrypt`
- `kms:Decrypt`
- `kms:CreateGrant`

Amazon から発信されるリクエストにのみ KMS キーを使用できるようにするには WorkMail、kms:ViaService 条件キーを `workmail.<region>.amazonaws.com`値とともに使用できます。

また、暗号化オペレーションに KMS キーを使用する条件として、暗号化コンテキストでキーまたは値を使用することもできます。例えば、IAM またはキーポリシードキュメントで文字列条件演算子を使用したり、許可で許可制約を使用したりできます。

キーポリシー (AWS マネージドキー 用)

Amazon AWS マネージドキーの のキーポリシーは、Amazon がユーザーに代わってリクエスト WorkMail を行った場合にのみ、指定されたオペレーションに KMS キーを使用するアクセス許可をユーザーに WorkMail 付与します。このキーポリシーでは、ユーザーが KMS キーを直接使用することは許可されません。

このキーポリシーは、すべての AWS マネージドキー のポリシーと同様に、サービスによって確立されます。キーポリシーは変更できませんが、いつでも表示できます。詳細については、「キーポリシーの表示」を参照してください。

このキーポリシーのポリシーステートメントには次の効果があります

- アカウントとリージョンのユーザーが KMS キーを暗号化オペレーションに使用して許可を作成できるようにします。ただし、リクエストが Amazon から送信された場合に限り WorkMail ます。kms:ViaService 条件キーで、この制限を適用します。
- KMS キープロパティを表示し、権限を取り消すことをユーザーに許可する IAM ポリシーを AWS アカウント が作成できるようにします。

以下は、Amazon の例のキーポリシーAWS マネージドキーです WorkMail。

```
{
  "Version" : "2012-10-17",
  "Id" : "auto-workmail-1",
  "Statement" : [ {
    "Sid" : "Allow access through WorkMail for all principals in the account that are
 authorized to use WorkMail",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "*"
    },
    "Action" : [ "kms:Decrypt", "kms:CreateGrant", "kms:ReEncrypt*", "kms:DescribeKey",
 "kms:Encrypt" ],
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "kms:ViaService" : "workmail.us-east-1.amazonaws.com",
        "kms:CallerAccount" : "111122223333"
      }
    }
  }, {
```

```
    "Sid" : "Allow direct access to key metadata to the account",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : [ "kms:Describe*", "kms:List*", "kms:Get*", "kms:RevokeGrant" ],
    "Resource" : "*"
  } ]
}
```

許可を使用した Amazon の認証 WorkMail

キーポリシーに加えて、Amazon WorkMail は権限を使用して、各組織の KMS キーにアクセス許可を追加します。アカウントの KMS キーに対する許可を表示するには、 ListGrantsオペレーションを使用します。

Amazon WorkMail はグラントを使用して、組織の KMS キーに次のアクセス許可を追加します。

- Amazon がメールボックスキーを暗号化 WorkMail できるようにする アクセス`kms:Encrypt`許可を追加します。
- Amazon が KMS キー WorkMail を使用してメールボックスキーを復号できるようにする アクセス`kms:Decrypt`許可を追加します。Amazon では、メールボックスメッセージを読み取るリクエストは、メッセージを読み取っているユーザーのセキュリティコンテキストを使用するため、許可でこのアクセス許可 WorkMail が必要です。リクエストは AWS アカウント の認証情報を使用しません。Amazon は、組織の KMS キーを選択すると、この権限 WorkMail を作成します。

許可を作成するために、Amazon は組織を作成したユーザーCreateGrantに代わって を WorkMail 呼び出します。権限付与を作成するアクセス許可はキーポリシーから付与されます。このポリシーでは、Amazon `CreateGrant`が承認されたユーザーに代わってリクエスト WorkMail を行うときに、アカウントユーザーが組織の KMS キーで を呼び出すことができます。

キーポリシーはアカウントルートが AWS マネージドキー で権限を取り消すことも許可します。ただし、許可を取り消すと、Amazon はメールボックス内の暗号化されたデータを復号化 WorkMail できません。

Amazon WorkMail 暗号化コンテキスト

暗号化コンテキストは、任意のシークレットデータを含まない、一連のキーと値のペアです。データを暗号化するリクエストに暗号化コンテキストを組み込むと、AWS KMS は暗号化コンテキストを暗

号化されたデータに暗号化してバインドします。データを復号するには、同じ暗号化コンテキストに渡す必要があります。

Amazon は、すべての暗号化オペレーションで同じAWS KMS暗号化コンテキスト形式 WorkMail を使用します。暗号化コンテキストを使用して、AWS CloudTrail などの監査レコードやログで、暗号化オペレーションを確認できます。また、ポリシーと許可で認可の条件として確認することもできます。

への暗号化および復号リクエストではAWS KMS、Amazon は暗号化コンテキスト WorkMail を使用します。ここで、キーは `aws:workmail:arn`で、値は組織の Amazon リソースネーム (ARN) です。

```
"aws:workmail:arn":"arn:aws:workmail:region:account ID:organization/organization ID"
```

例えば、次の暗号化コンテキストには、米国東部 (オハイオ) (`us-east-2`) リージョンの組織 ARN の例が含まれています。

```
"aws:workmail:arn":"arn:aws:workmail:us-east-2:111122223333:organization/
m-68755160c4cb4e29a2b2f8fb58f359d7"
```

との Amazon WorkMail インタラクションのモニタリング AWS KMS

AWS CloudTrail および Amazon CloudWatch Logs を使用して、Amazon がAWS KMSユーザーに代わって WorkMail に送信するリクエストを追跡できます。

暗号化

新しいメールボックスを作成すると、Amazon はメールボックスキー WorkMail を生成し、 を呼び出しAWS KMSてメールボックスキーを暗号化します。Amazon は、プレーンテキストのメールボックスキーと Amazon WorkMail 組織の KMS キーの識別子AWS KMSを使用して、Encrypt リクエストを WorkMail に送信します。

`Encrypt` 演算を記録するイベントは、次のようなサンプルイベントになります。ユーザーは Amazon WorkMail サービスです。パラメータには、KMS キー ID (`keyId`) と Amazon WorkMail 組織の暗号化コンテキストが含まれます。Amazon はメールボックスキー WorkMail も渡しますが、ログには記録されません CloudTrail 。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
```

```
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-19T10:01:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        },
        "keyId": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
    },
    "responseElements": null,
    "requestID": "76e96b96-7e24-4faf-a2d6-08ded2eaf63c",
    "eventID": "d5a59c18-128a-4082-aa5b-729f7734626a",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "d08e60f1-097e-4a00-b7e9-10bc3872d50c"
}
```


Decrypt

メールボックスメッセージを追加、表示、または削除すると、Amazon は AWS KMSにメールボックスキーの復号 WorkMail を要求します。Amazon は、暗号化されたメールボックスキーと Amazon WorkMail 組織の KMS キーの識別子AWS KMSを使用して、Decrypt リクエストを WorkMail に送信します。

Decrypt 演算を記録するイベントは、次のようなサンプルイベントになります。ユーザーは Amazon WorkMail サービスです。パラメータには、暗号化されたメールボックスキー (暗号化テキ

ストの blob として) が含まれ、ログには記録されません WorkMail 。 は、暗号化テキストから KMS キーの ID AWS KMSを取得します。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "workmail.eu-west-1.amazonaws.com"
    },
    "eventTime": "2019-02-20T11:51:10Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-1",
    "sourceIPAddress": "workmail.eu-west-1.amazonaws.com",
    "userAgent": "workmail.eu-west-1.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:workmail:arn": "arn:aws:workmail:eu-west-1:111122223333:organization/
m-c6981ff7642446fa8772ba99c690e455"
        }
    },
    "responseElements": null,
    "requestID": "4a32dda1-34d9-4100-9718-674b8e0782c9",
    "eventID": "ea9fd966-98e9-4b7b-b377-6e5a397a71de",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "accountId": "111122223333",
            "type": "AWS::KMS::Key"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "sharedEventID": "241e1e5b-ff64-427a-a5b3-7949164d0214"
}
```


が WorkSpaces を使用する方法 AWS KMS

を使用してWorkSpaces、エンドユーザーごとにクラウドベースのデスクトップ (WorkSpace) をプロビジョニングできます。新しい を起動するときに WorkSpace、そのボリュームを暗号化すること

を選択しAWS KMS key、暗号化に使用する を決定できます。 (aws/workspaces) AWS マネージドキーの WorkSpaces または対称カスタマーマネージドキー を選択できます。

> **Important**
>
> WorkSpaces は、対称暗号化 KMS キーのみをサポートします。非対称 KMS キーを使用してのボリュームを暗号化することはできません WorkSpaces。KMS キーが対称か非対称かを判断する方法については、「非対称 KMS キーの識別」を参照してください。

暗号化されたボリューム WorkSpaces を使用した の作成の詳細については、「Amazon WorkSpaces 管理ガイド」の「 の暗号化 WorkSpace」を参照してください。

トピック

- を使用した WorkSpaces 暗号化の概要 AWS KMS
- WorkSpaces 暗号化コンテキスト
- ユーザーに代わって KMS キーを使用するアクセス WorkSpaces 許可を付与する

を使用した WorkSpaces 暗号化の概要 AWS KMS

暗号化されたボリューム WorkSpaces で を作成すると、 は Amazon Elastic Block Store (Amazon EBS) WorkSpaces を使用してそれらのボリュームを作成および管理します。どちらのサービスも、AWS KMS key を使用して暗号化されたボリュームを操作します。EBS ボリュームの暗号化の詳細については、以下のドキュメントを参照してください。

- このガイドの「Amazon Elastic Block Store (Amazon EBS) が AWS KMS を使用する方法」
- Windows インスタンス用 Amazon EC2 ユーザーガイドの Amazon EBS 暗号化

暗号化されたボリューム WorkSpaces で を起動すると、 end-to-end プロセスは次のように動作します。

1. 暗号化に使用する KMS キーと、 WorkSpaceのユーザーとディレクトリを指定します。このアクションは、 が KMS キーをこの場合にのみ使用 WorkSpaces できるようにする許可を作成します。 WorkSpaceつまり、指定されたユーザーとディレクトリ WorkSpace に関連付けられたに対してのみ使用します。
2. WorkSpaces は、 の暗号化された EBS ボリュームを作成し、使用する KMS キーとボリュームのユーザーおよびディレクトリ (で指定した情報と同じ情報) WorkSpace を指定しますStep

1。このアクションにより、Amazon EBS がこの WorkSpace とボリュームにのみ KMS キーを使用できるようにする許可が作成されます。つまり、指定されたユーザーとディレクトリ WorkSpace に関連付けられた のみ、および指定されたボリュームにのみ使用できます。

3. Amazon EBS は、KMS キーで暗号化されたボリュームデータキーをリクエストし、暗号化コンテキストとして WorkSpace ユーザーの Sidとディレクトリ ID、ボリューム ID を指定します。
4. AWS KMS は、新しいデータキーを作成し、KMS キーによって暗号化して、暗号化されたデータキーを Amazon EBS に送信します。
5. WorkSpaces は Amazon EBS を使用して、暗号化されたボリュームを にアタッチします WorkSpace。Amazon EBS は、暗号化されたデータキーを DecryptリクエストAWS KMSとともに に送信し、 WorkSpace ユーザーの Sid、ディレクトリ ID、および暗号化コンテキスト として使用されるボリューム ID を指定します。
6. AWS KMS は、KMS キーを使用してデータキーを復号し、プレーンテキストのデータキーを Amazon EBS に送信します。
7. Amazon EBS は、プレーンテキストデータキーを使用して、暗号化されたボリュームを出入りするすべてのデータを暗号化します。Amazon EBS は、ボリュームが にアタッチされている限り、プレーンテキストのデータキーをメモリに保持します WorkSpace。
8. Amazon EBS は、暗号化されたデータキー (で受け取ったデータキーStep 4) をボリュームメタデータとともに保存し、 を再起動または再構築した場合に今後使用します WorkSpace。
9. を使用して AWS Management Consoleを削除する WorkSpace （ または WorkSpaces API で TerminateWorkspacesアクションを使用する) WorkSpaces と、Amazon EBS はその の KMS キーの使用を許可した許可を廃止にします WorkSpace。

WorkSpaces 暗号化コンテキスト

WorkSpaces は、 を暗号化オペレーション (Encrypt、、 など) AWS KMS keyに直接使用しません。つまりGenerateDataKey、 WorkSpaces は暗号化コンテキスト AWS KMS を含む Decryptにリクエストを送信しません。ただし、Amazon EBS が WorkSpaces （ Step 3 の を使用した WorkSpaces 暗号化の概要 AWS KMS) の暗号化されたボリュームに対して暗号化されたデータキーをリクエストする場合、およびそのデータキーのプレーンテキストコピー (Step 5) をリクエストする場合、リクエストに暗号化コンテキストが含まれます。暗号化コンテキストは、データの整合性を保証するために AWS KMS で使用される追加の認証データ (AAD) を提供します。また、暗号化コンテキストは AWS CloudTrail ログファイルにも書き込まれるため、特定の AWS KMS key が使用された原因を理解するのに役立ちます。Amazon EBS では、暗号化コンテキストとして次のものが使用されます。

- に関連付けられているsidAWS Directory Serviceユーザーの WorkSpace
- に関連付けられているディレクトリのAWS Directory Serviceディレクトリ ID WorkSpace
- 暗号化されたボリュームの ボリューム ID

次の例は、Amazon EBS が使用する暗号化コンテキストの JSON 表現を示しています。

```
{
  "aws:workspaces:sid-directoryid":
 "[S-1-5-21-277731876-1789304096-451871588-1107]@[d-1234abcd01]",
  "aws:ebs:id": "vol-1234abcd"
}
```

ユーザーに代わって KMS キーを使用するアクセス WorkSpaces 許可を付与する

(aws/workspaces) またはカスタマーマネージドキーAWS マネージドキーで WorkSpacesワークスペースデータを保護できます。カスタマーマネージドキーを使用する場合は、アカウントの管理者に代わって WorkSpaces KMS キーを使用する WorkSpaces アクセス許可を付与する必要があります。AWS マネージドキー の WorkSpaces には、デフォルトで必要なアクセス許可があります。

で使用するカスタマーマネージドキーを準備するには WorkSpaces、次の手順を使用します。

1. KMS キーのキーポリシーのキーユーザーのリストに WorkSpaces 管理者を追加する
2. IAM ポリシーを使用して WorkSpaces 管理者に追加のアクセス許可を付与する

WorkSpaces 管理者には、 を使用するアクセス許可も必要です WorkSpaces。これらのアクセス許可の詳細については、「Amazon WorkSpaces 管理ガイド」の「リソースへのアクセスのWorkSpaces制御」を参照してください。

パート 1: KMS キーのキーユーザーに WorkSpaces 管理者を追加する

WorkSpaces 管理者に必要なアクセス許可を付与するには、 AWS Management Consoleまたは AWS KMS API を使用できます。

KMS キーのキーユーザーとして WorkSpaces 管理者を追加するには (コンソール)

1. AWS Management Console にサインインし、AWS Key Management Service (AWS KMS) コンソール (https://console.aws.amazon.com/kms) を開きます。

2. AWS リージョン を変更するには、ページの右上隅にあるリージョンセレクターを使用します。
3. ナビゲーションペインで、[カスタマーマネージドキー] を選択します。
4. 任意のカスタマーマネージドキーのキー ID またはエイリアスを選択する
5. [キーポリシー] タブを選択します。[Key users] (キーユーザー) で [Add] (追加) を選択します。
6. IAM ユーザーとロールのリストで、 WorkSpaces 管理者に対応するユーザーとロールを選択し、アタッチを選択します。

KMS キーのキーユーザーとして WorkSpaces 管理者を追加するには (AWS KMS API)

1. GetKeyPolicy オペレーションを使用して既存のキーポリシーを取得し、ポリシードキュメントをファイルに保存します。
2. 任意のテキストエディタでポリシードキュメントを開きます。 WorkSpaces 管理者に対応する IAM ユーザーとロールを、キーユーザー にアクセス許可を付与するポリシーステートメントに追加します。その後、ファイルを保存します。
3. PutKeyPolicy オペレーションを使用して、KMS キーにキーポリシーを適用します。

パート 2: WorkSpaces 管理者に追加のアクセス許可を付与する

カスタマーマネージドキーを使用して WorkSpaces データを保護する場合、デフォルトのキーポリシー のキーユーザーセクションのアクセス許可に加えて、 WorkSpaces 管理者は KMS キーに権限を作成するアクセス許可が必要です。また、 を使用して暗号化されたボリューム WorkSpaces で AWS Management Consoleを作成する場合、 WorkSpaces 管理者はエイリアスを一覧表示し、キーを一覧表示するアクセス許可が必要です。IAM ユーザーポリシーの作成と編集については、IAM ユーザーガイドのマネージドポリシーとインラインポリシーを参照してください。

これらのアクセス許可を WorkSpaces 管理者に付与するには、IAM ポリシーを使用します。次の例のような ポリシーステートメントを各 WorkSpaces管理者の IAM ポリシーに追加します。サンプルKMS キー ARN (*arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab*) を有効なものに置き換えます。 WorkSpaces 管理者が (コンソールではなく) WorkSpaces API のみを使用する場合は、 `"kms:ListAliases"`および の`"kms:ListKeys"`アクセス許可を持つ 2 番目のポリシーステートメントを省略できます。

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

```
    {
      "Effect": "Allow",
      "Action": "kms:CreateGrant",
      "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListAliases",
        "kms:ListKeys"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS KMS API のプログラミング

AWS KMS APIを使用して KMS キーやカスタムキーストアなどの特別な機能の作成および管理ができ、KMS キーは暗号化オペレーションで使用できます。詳細については、「AWS Key Management Service API リファレンス」を参照してください。

以下のトピックのサンプルコードは、AWS SDK を使用して AWS KMS API を呼び出す方法を示しています。

AWS KMS コンソールを使用してこれらのタスクのいくつかを実行する方法については、「キーの管理」を参照してください。

トピック

- クライアントの作成
- キーの使用
- エイリアスの使用
- データキーの暗号化と復号
- キーポリシーの使用
- 許可の使用
- AWS KMS API 呼び出しをテストする
- AWS KMS の結果整合性

クライアントの作成

AWS SDK for Java、AWS SDK for .NET、、AWS SDK for Python (Boto3)、AWS SDK for RubyAWS SDK for PHP、または AWS SDK for を Node.js JavaScript で使用して、 AWS Key Management Service (AWS KMS) API を使用するコードを記述するには、まず AWS KMSクライアントを作成します。

作成するクライアントオブジェクトは、次のトピックのサンプルコードで使用されます。

Java

Java で AWS KMS クライアントを作成するには、クライアントビルダーを使用します。

```
AWSKMS kmsClient = AWSKMSClientBuilder.standard().build();
```

Java クライアントビルダーの使用に関する詳細は、以下のリソースを参照してください。

- AWS デベロッパーブログの Fluent Client Builders
- AWS SDK for Java デベロッパーガイドのサービスクライアントの作成
- 「AWSKMSClientBuilder API リファレンス」の「AWS SDK for Java」

C#

```
AmazonKeyManagementServiceClient kmsClient = new AmazonKeyManagementServiceClient();
```

Python

```
kms_client = boto3.client('kms')
```

Ruby

```
require 'aws-sdk-kms'  # in v2: require 'aws-sdk'

kmsClient = Aws::KMS::Client.new
```

PHP

PHP で AWS KMS クライアントを作成するには、AWS KMS クライアントオブジェクトを使用して、バージョン 2014-11-01 を指定します。詳細については、AWS SDK for PHP API リファレンスの KMSClient クラスを参照してください。

```
// Create a KMSClient
$KmsClient = new Aws\Kms\KmsClient([
    'profile' => 'default',
    'version' => '2014-11-01',
    'region'  => 'us-east-1'
]);
```

Node.js

```
const kmsClient = new AWS.KMS();
```

キーの使用

このトピックの例では、AWS KMS API を使用して AWS KMS AWS KMS keys を作成、表示、有効化、無効化し、データキーを生成します。

トピック

- KMS キーを作成する
- データキーの生成
- AWS KMS key の表示
- KMS キーのキー ID とキー ARN を取得する
- AWS KMS keys の有効化
- AWS KMS key の無効化

KMS キーを作成する

AWS KMS key (KMS キー) を作成するには、 CreateKeyオペレーションを使用します。このセクションの例は、対称暗号化 KMS キーを作成します。これらの例で使用されている `Description` パラメータはオプションです。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

AWS KMS コンソールで KMS キーを作成する方法については、キーの作成 を参照してください。

Java

詳細については、AWS SDK for Java API リファレンスの createKey メソッド を参照してください。

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest().withDescription(desc);
CreateKeyResult result = kmsClient.createKey(req);
```

C#

詳細については、AWS SDK for .NET の「CreateKey メソッド」を参照してください。

```
// Create a KMS key
//
String desc = "Key for protecting critical data";

CreateKeyRequest req = new CreateKeyRequest()
{
    Description = desc
};
CreateKeyResponse response = kmsClient.CreateKey(req);
```

Python

詳細については、AWS SDK for Python (Boto3) の「create_key メソッド」を参照してください。

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kms_client.create_key(
    Description=desc
)
```

Ruby

詳細については、AWS SDK for Ruby の「create_key インスタンスメソッド」を参照してください。

```
# Create a KMS key

desc = 'Key for protecting critical data'

response = kmsClient.create_key({
  description: desc
})
```

PHP

詳細については、AWS SDK for PHP の「CreateKey メソッド」を参照してください。

```
// Create a KMS key
```

```
//
$desc = "Key for protecting critical data";

$result = $KmsClient->createKey([
    'Description' => $desc
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の createKey プロパティを参照してください。

```
// Create a KMS key
//
const Description = 'Key for protecting critical data';

kmsClient.createKey({ Description }, (err, data) => {
  ...
});
```

PowerShell

で KMS キーを作成するには PowerShell、New-KmsKeycmdlet を使用します。

```
# Create a KMS key

$desc = 'Key for protecting critical data'
New-KmsKey -Description $desc
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

データキーの生成

対称データキー を生成するには、 GenerateDataKeyオペレーションを使用します。このオペレーションは、プレーンテキストのデータキーと、ユーザー指定の対称暗号化 KMS キーで暗号化されたデータキーのコピーを返します。各コマンドには、KeySpec または NumberOfBytes のいずれか (両方ではなく) を指定する必要があります。

データの暗号化にデータキーを使用する方法については、「AWS Encryption SDK」を参照してください。HMAC オペレーションでデータキーを使用することもできます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、 AWS SDK for Java API リファレンスの generateDataKey メソッドを参照してください。

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest();
dataKeyRequest.setKeyId(keyId);
dataKeyRequest.setKeySpec("AES_256");

GenerateDataKeyResult dataKeyResult = kmsClient.generateDataKey(dataKeyRequest);

ByteBuffer plaintextKey = dataKeyResult.getPlaintext();

ByteBuffer encryptedKey = dataKeyResult.getCiphertextBlob();
```

C#

詳細については、AWS SDK for .NET の「GenerateDataKey メソッド」を参照してください。

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
GenerateDataKeyRequest dataKeyRequest = new GenerateDataKeyRequest()
{
    KeyId = keyId,
    KeySpec = DataKeySpec.AES_256
};
```

```
GenerateDataKeyResponse dataKeyResponse = kmsClient.GenerateDataKey(dataKeyRequest);

MemoryStream plaintextKey = dataKeyResponse.Plaintext;

MemoryStream encryptedKey = dataKeyResponse.CiphertextBlob;
```

Python

詳細については、AWS SDK for Python (Boto3) の「generate_data_key メソッド」を参照してください。

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.generate_data_key(
    KeyId=key_id,
    KeySpec='AES_256'
)

plaintext_key = response['Plaintext']

encrypted_key = response['CiphertextBlob']
```

Ruby

詳細については、AWS SDK for Ruby の「generate_data_key インスタンスメソッド」を参照してください。

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.generate_data_key({
  key_id: key_id,
  key_spec: 'AES_256'
})
```

```
plaintext_key = response.plaintext

encrypted_key = response.ciphertext_blob
```

PHP

詳細については、AWS SDK for PHP の「GenerateDataKey メソッド」を参照してください。

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$keySpec = 'AES_256';

$result = $KmsClient->generateDataKey([
    'KeyId' => $keyId,
    'KeySpec' => $keySpec,
]);

$plaintextKey = $result['Plaintext'];

$encryptedKey = $result['CiphertextBlob'];
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の generateDataKey プロパティを参照してください。

```
// Generate a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const KeySpec = 'AES_256';
kmsClient.generateDataKey({ KeyId, KeySpec }, (err, data) => {
  if (err) console.log(err, err.stack);
  else {
    const { CiphertextBlob, Plaintext } = data;
      ...
  }
});
```

PowerShell

対称データキーを生成するには、New-KMSDataKey コマンドレットを使用します。

出力では、プレーンテキストキー (Plaintextプロパティ内) と暗号化キー (CiphertextBlobプロパティ内) はMemoryStreamオブジェクトです。それらを文字列に変換するには、 MemoryStream クラスの メソッド、または Convert モジュールの ConvertFrom-MemoryStream や ConvertFrom-Base64 関数などのMemoryStreamオブジェクトを文字列に変換するコマンドレットまたは関数を使用します。

```
# Generate a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$keySpec = 'AES_256'

$response = New-KmsDataKey -KeyId $keyId -KeySpec $keySpec
$plaintextKey = $response.Plaintext
$encryptedKey = $response.CiphertextBlob
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

AWS KMS key の表示

KMS キー ARN やキーステータス などAWS KMS key、 に関する詳細情報を取得するには、DescribeKeyオペレーションを使用します。 ???

DescribeKey はエイリアスを取得しません。エイリアスを取得するには、 ListAliasesオペレーションを使用します。例については、「エイリアスの使用」を参照してください。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

AWS KMS コンソールで KMS キーを表示する方法については、キーの表示 を参照してください。

Java

詳細については、AWS SDK for Java API リファレンスの describeKey メソッドを参照してください。

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest req = new DescribeKeyRequest().withKeyId(keyId);
DescribeKeyResult result = kmsClient.describeKey(req);
```

C#

詳細については、AWS SDK for .NET の「DescribeKey メソッド」を参照してください。

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DescribeKeyRequest describeKeyRequest = new DescribeKeyRequest()
{
    KeyId = keyId
};

DescribeKeyResponse describeKeyResponse = kmsClient.DescribeKey(describeKeyRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「describe_key メソッド」を参照してください。

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.describe_key(
```

```
    KeyId=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「describe_key インスタンスメソッド」を参照してください。

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.describe_key({
  key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の「DescribeKey メソッド」を参照してください。

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->describeKey([
    'KeyId' => $keyId,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の describeKey プロパティを参照してください。

```
// Describe a KMS key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.describeKey({ KeyId }, (err, data) => {
```

```
  ...
});
```

PowerShell

KMS キーに関する詳細情報を取得するには、Get-KmsKeycmdlet を使用します。

```
# Describe a KMS key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Get-KmsKey -KeyId $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

KMS キーのキー ID とキー ARN を取得する

のキー IDs とキー ARNs を取得するにはAWS KMS keys、 ListKeysオペレーションを使用します。これらの例では、オプションの `Limit` パラメータを使用して、各呼び出しで返される KMS キーの最大数を設定します。AWS KMS オペレーションで KMS キーを識別する方法については、キー識別子 (KeyId) を参照してください。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

AWS KMS コンソールでキー ID とキー ARN を検索する方法については、「キー ID とキー ARN を検索する」を参照してください。

Java

詳細については、AWS SDK for Java API リファレンスの listKeys メソッドを参照してください。

```
// List KMS keys in this account
//
Integer limit = 10;

ListKeysRequest req = new ListKeysRequest().withLimit(limit);
```

```
ListKeysResult result = kmsClient.listKeys(req);
```

C#

詳細については、AWS SDK for .NET の「ListKeys メソッド」を参照してください。

```
// List KMS keys in this account
//
int limit = 10;

ListKeysRequest listKeysRequest = new ListKeysRequest()
{
    Limit = limit
};
ListKeysResponse listKeysResponse = kmsClient.ListKeys(listKeysRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_keys メソッド」を参照してください。

```
# List KMS keys in this account

response = kms_client.list_keys(
    Limit=10
)
```

Ruby

詳細については、AWS SDK for Ruby の「list_keys インスタンスメソッド」を参照してください。

```
# List KMS keys in this account

response = kmsClient.list_keys({
  limit: 10
})
```

PHP

詳細については、AWS SDK for PHP の「ListKeys メソッド」を参照してください。

```
// List KMS keys in this account
```

```
//
$limit = 10;

$result = $KmsClient->listKeys([
    'Limit' => $limit,
]);
```

Node.js

詳細については、 SDK for in Node.js の listKeys プロパティを参照してください。 AWS JavaScript

```
// List KMS keys in this account
//
const Limit = 10;
kmsClient.listKeys({ Limit }, (err, data) => {
  ...
});
```

PowerShell

アカウントとリージョン内のすべての KMS キーのキー ID とキー ARN を取得するには、Get-KmsKeyListcmdlet を使用します。

出力オブジェクトの数を制限するために、この例では、リストコマンドレットで非推奨の `Limit` パラメータの代わりに Select-Object コマンドレットを使用します。AWS Tools for PowerShell での出力のページ分割については、「AWS Tools for PowerShell での出力ページ分割」を参照してください。

```
# List KMS keys in this account

$limit = 10
Get-KmsKeyList | Select-Object -First $limit
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

AWS KMS keys の有効化

無効化された を有効にするにはAWS KMS key、 EnableKeyオペレーションを使用します。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

AWS KMS コンソールで KMS キーを有効または無効にする方法については、キーの有効化と無効化を参照してください。

Java

Java の実装の詳細については、AWS SDK for Java API リファレンスの enableKey メソッドを参照してください。

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest req = new EnableKeyRequest().withKeyId(keyId);
kmsClient.enableKey(req);
```

C#

詳細については、AWS SDK for .NET の「EnableKey メソッド」を参照してください。

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

EnableKeyRequest enableKeyRequest = new EnableKeyRequest()
{
    KeyId = keyId
};
kmsClient.EnableKey(enableKeyRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「enable_key メソッド」を参照してください。

```
# Enable a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.enable_key(
    KeyId=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「enable_key インスタンスメソッド」を参照してください。

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.enable_key({
  key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の「EnableKey メソッド」を参照してください。

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->enableKey([
    'KeyId' => $keyId,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の enableKey プロパティを参照してください。

```
// Enable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.enableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

KMS キーを有効にするには、Enable-KmsKeycmdlet を使用します。

```
# Enable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Enable-KmsKey -KeyId $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

AWS KMS key の無効化

KMS キーを無効にするには、 DisableKeyオペレーションを使用します。KMS キーを無効にすることで、暗号化オペレーションで使用されるのを防ぎます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

AWS KMS コンソールで KMS キーを有効または無効にする方法については、キーの有効化と無効化を参照してください。

Java

詳細については、AWS SDK for Java API リファレンスの disableKey メソッドを参照してください。

```
// Disable a KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DisableKeyRequest req = new DisableKeyRequest().withKeyId(keyId);
kmsClient.disableKey(req);
```

C#

詳細については、AWS SDK for .NET の「DisableKey メソッド」を参照してください。

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

DisableKeyRequest disableKeyRequest = new DisableKeyRequest()
{
    KeyId = keyId
};
kmsClient.DisableKey(disableKeyRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「disable_key メソッド」を参照してください。

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.disable_key(
    KeyId=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「disable_key インスタンスメソッド」を参照してください。

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.disable_key({
  key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の「DisableKey メソッド」を参照してください。

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->disableKey([
    'KeyId' => $keyId,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の disableKey プロパティを参照してください。

```
// Disable a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.disableKey({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

KMS キーを無効にするには、Disable-KmsKey コマンドレットを使用します。

```
# Disable a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
Disable-KmsKey -KeyId $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

エイリアスの使用

このトピックの例では、AWS KMS API を使用してエイリアスを作成、表示、更新、および削除します。エイリアスの詳細については、を参照してください the section called “エイリアスの使用”。

トピック

- エイリアスの作成
- エイリアスのリスト化
- エイリアスの更新
- エイリアスの削除

エイリアスの作成

AWS Management Console で AWS KMS key を作成するときは、そのエイリアスを作成する必要があります。ただし、KMS キーを作成するCreateKeyオペレーションではエイリアスは作成されません。

エイリアスを作成するには、 CreateAliasオペレーションを使用します。エイリアスはアカウントと リージョンで一意であることが必要です。aws/ で始まるエイリアスを作成することはできません。aws/ プレフィックスは、AWS マネージドキー の Amazon Web Services によって予約されます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、AWS SDK for Java API リファレンスの createAlias メソッド を参照してください。

```
// Create an alias for a KMS key
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest req = new
 CreateAliasRequest().withAliasName(aliasName).withTargetKeyId(targetKeyId);
kmsClient.createAlias(req);
```

C#

詳細については、AWS SDK for .NET の「CreateAlias メソッド」を参照してください。

```
// Create an alias for a KMS key
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

CreateAliasRequest createAliasRequest = new CreateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};
kmsClient.CreateAlias(createAliasRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「create_alias メソッド」を参照してください。

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
```

```
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.create_alias(
    AliasName=alias_name,
    TargetKeyId=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「create_alias インスタンスメソッド」を参照してください。

```
# Create an alias for a KMS key

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
target_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.create_alias({
  alias_name: alias_name,
  target_key_id: target_key_id
})
```

PHP

詳細については、AWS SDK for PHP の「CreateAlias メソッド」を参照してください。

```
// Create an alias for a KMS key
//
$aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';


$result = $KmsClient->createAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
]);
```

Node.js

詳細については、 SDK for in Node.js の createAlias プロパティを参照してください。 AWS JavaScript

```
// Create an alias for a KMS key
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.createAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

エイリアスを作成するには、 New-KMSAlias コマンドレットを使用します。エイリアス名では、大文字と小文字が区別されます。

```
# Create an alias for a KMS key

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$targetKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

New-KMSAlias -TargetKeyId $targetKeyId -AliasName $aliasName
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

エイリアスのリスト化

アカウントとリージョンのエイリアスを一覧表示するには、 ListAliasesオペレーションを使用します。

デフォルトでは、ListAliases コマンドはアカウントとリージョンのすべてのエイリアスを返します。これには、ユーザーが作成してカスタマーマネージドキーに関連付けたエイリアスと、AWS が

作成してアカウントの AWS マネージドキー に関連付けたエイリアスが含まれます。レスポンスには、TargetKeyId フィールドがないエイリアスが含まれている場合もあります。これらは AWS が作成した定義済みのエイリアスですが、まだ KMS キーとは関連付けられていません。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

Java の実装の詳細については、AWS SDK for Java API リファレンスの listAliases method を参照してください。

```
// List the aliases in this AWS #####
//
Integer limit = 10;

ListAliasesRequest req = new ListAliasesRequest().withLimit(limit);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

詳細については、AWS SDK for .NET の「ListAliases メソッド」を参照してください。

```
// List the aliases in this AWS #####
//
int limit = 10;

ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    Limit = limit
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_aliases メソッド」を参照してください。

```
# List the aliases in this AWS #####

response = kms_client.list_aliases(
    Limit=10
```

```
)
```

Ruby

詳細については、list_aliases の AWS SDK for Ruby インスタンスメソッドを参照してください。

```
# List the aliases in this AWS #####

response = kmsClient.list_aliases({
  limit: 10
})
```

PHP

詳細については、AWS SDK for PHP の List Aliases method を参照してください。

```
// List the aliases in this AWS #####
//
$limit = 10;

$result = $KmsClient->listAliases([
    'Limit' => $limit,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の listAliases プロパティを参照してください。

```
// List the aliases in this AWS #####
//
const Limit = 10;
kmsClient.listAliases({ Limit }, (err, data) => {
  ...
});
```

PowerShell

アカウントとリージョンのエイリアスを一覧表示するには、Get-KMSAliasList コマンドレットを使用します。

出力オブジェクトの数を制限するために、この例では、リストコマンドレットで非推奨の `Limit` パラメータの代わりに Select-Object コマンドレットを使用します。AWS Tools for PowerShell での出力のページ分割については、「AWS Tools for PowerShell での出力ページ分割」を参照してください。

```
# List the aliases in this AWS #####
$limit = 10

$result = Get-KMSAliasList | Select-Object -First $limit
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

特定の KMS キーに関連付けられているエイリアスのみをリストするには、`KeyId` パラメータを使用します。その値として、リージョン内の任意の KMS キーのキー ID またはキー ARN を指定できます。エイリアス名またはエイリアス ARN を指定することはできません。

Java

Java の実装の詳細については、AWS SDK for Java API リファレンスの listAliases method を参照してください。

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListAliasesRequest req = new ListAliasesRequest().withKeyId(keyId);
ListAliasesResult result = kmsClient.listAliases(req);
```

C#

詳細については、AWS SDK for .NET の「ListAliases メソッド」を参照してください。

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
```

```
ListAliasesRequest listAliasesRequest = new ListAliasesRequest()
{
    KeyId = keyId
};
ListAliasesResponse listAliasesResponse = kmsClient.ListAliases(listAliasesRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_aliases メソッド」を参照してください。

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_aliases(
    KeyId=key_id
)
```

Ruby

詳細については、list_aliases の AWS SDK for Ruby インスタンスメソッドを参照してください。

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_aliases({
  key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の List Aliases method を参照してください。

```
// List the aliases for one KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listAliases([
    'KeyId' => $keyId,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の listAliases プロパティを参照してください。

```
// List the aliases for one KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
kmsClient.listAliases({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

KMS キーのエイリアスを一覧表示するには、Get-KMSAliasList コマンドレットの KeyIdパラメータを使用します。

```
# List the aliases for one KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

$response = Get-KmsAliasList -KeyId $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

エイリアスの更新

既存のエイリアスを別の KMS キーに関連付けるには、 UpdateAliasオペレーションを使用します。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

Java の実装の詳細については、AWS SDK for Java APIリファレンスの updateAlias メソッドを参照してください。

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest req = new UpdateAliasRequest()
     .withAliasName(aliasName)
     .withTargetKeyId(targetKeyId);

kmsClient.updateAlias(req);
```

C#

詳細については、AWS SDK for .NET の「UpdateAlias メソッド」を参照してください。

```
// Updating an alias
//
String aliasName = "alias/projectKey1";
// Replace the following example key ARN with a valid key ID or key ARN
String targetKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

UpdateAliasRequest updateAliasRequest = new UpdateAliasRequest()
{
    AliasName = aliasName,
    TargetKeyId = targetKeyId
};
```

```
kmsClient.UpdateAlias(updateAliasRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「update_alias メソッド」を参照してください。

```
# Updating an alias

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kms_client.update_alias(
    AliasName=alias_name,
    TargetKeyID=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「update_alias インスタンスメソッド」を参照してください。

```
# Updating an alias

alias_name = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

response = kmsClient.update_alias({
  alias_name: alias_name,
  target_key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の「UpdateAlias メソッド」を参照してください。

```
// Updating an alias
```

```
//
$aliasName = "alias/projectKey1";

// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';

$result = $KmsClient->updateAlias([
    'AliasName' => $aliasName,
    'TargetKeyId' =>  $keyId,
]);
```

Node.js

詳細については、 SDK for in Node.js の updateAlias プロパティを参照してください。 AWS JavaScript

```
// Updating an alias
//
const AliasName = 'alias/projectKey1';

// Replace the following example key ARN with a valid key ID or key ARN
const TargetKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';
kmsClient.updateAlias({ AliasName, TargetKeyId }, (err, data) => {
  ...
});
```

PowerShell

エイリアスに関連付けられている KMS キーを変更するには、Update-KMSAlias コマンドレットを使用します。エイリアス名では、大文字と小文字が区別されます。

Update-KMSAlias コマンドレットは出力を返しません。コマンドが機能したことを確認するには、Get-KMSAliasList コマンドレットを使用します。

```
# Updating an alias

$aliasName = 'alias/projectKey1'
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'
```

```
Update-KMSAlias -AliasName $aliasName -TargetKeyID $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

エイリアスの削除

エイリアスを削除するには、 DeleteAliasオペレーションを使用します。エイリアスを削除しても、関連付けられている KMS キーには影響しません。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、AWS SDK for Java API リファレンスの deleteAlias メソッドを参照してください。

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest req  = new DeleteAliasRequest().withAliasName(aliasName);
kmsClient.deleteAlias(req);
```

C#

詳細については、AWS SDK for .NET の「DeleteAlias メソッド」を参照してください。

```
// Delete an alias for a KMS key
//
String aliasName = "alias/projectKey1";

DeleteAliasRequest deleteAliasRequest = new DeleteAliasRequest()
{
    AliasName = aliasName
};
kmsClient.DeleteAlias(deleteAliasRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「delete_alias メソッド」を参照してください。

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kms_client.delete_alias(
    AliasName=alias_name
)
```

Ruby

詳細については、AWS SDK for Ruby の「delete_alias インスタンスメソッド」を参照してください。

```
# Delete an alias for a KMS key

alias_name = 'alias/projectKey1'

response = kmsClient.delete_alias({
  alias_name: alias_name
})
```

PHP

詳細については、AWS SDK for PHP の「DeleteAlias メソッド」を参照してください。

```
// Delete an alias for a KMS key
//
$aliasName = "alias/projectKey1";

$result = $KmsClient->deleteAlias([
    'AliasName' => $aliasName,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の deleteAlias プロパティ) を参照してください。

```
// Delete an alias for a KMS key
//
const AliasName = 'alias/projectKey1';
kmsClient.deleteAlias({ AliasName }, (err, data) => {
  ...
});
```

PowerShell

エイリアスを削除するには、 Remove-KMSAlias コマンドレットを使用します。エイリアス名では、大文字と小文字が区別されます。

このコマンドレットはエイリアスを完全に削除するため、 はコマンドを確認するよう PowerShell 促します。ConfirmImpact は High であるため、ConfirmPreference を使用してこのプロンプトを抑制することはできません。確認プロンプトを表示しないようにする必要がある場合は、Confirm 共通パラメータに -Confirm:$false の値を追加します (例: $false)。

Remove-KMSAlias コマンドレットは出力を返しません。コマンドが有効であることを確認するには、Get-KMSAliasList コマンドレットを使用します。

```
# Delete an alias for a KMS key

$aliasName = 'alias/projectKey1'
Remove-KMSAlias -AliasName $aliasName
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

データキーの暗号化と復号

このトピックの例では、 AWS KMS API の Encrypt 、DecryptReEncrypt 、および オペレーションを使用します。

これらのオペレーションは、データキーを暗号化および復号するように設計されています。暗号化オペレーションでは AWS KMS keys を使用しますが、4 KB (4,096 バイト) を超えるデータを受け付けることはできません。パスワードや RSA キーなどの少量データを暗号化するためにこれを使用できますが、アプリケーションデータを暗号化するために設計されていません。

アプリケーションデータを暗号化するには、AWS サービスのサーバー側の暗号化機能、またはクライアント側の暗号化ライブラリを使用します (AWS Encryption SDK や Amazon S3 暗号化クライアントなど)。

トピック

- データキーの暗号化
- データキーの復号
- 異なる AWS KMS key によるデータキーの再暗号化

データキーの暗号化

Encrypt オペレーションは、データキーを暗号化するように設計されていますが、頻繁に使用されていません。GenerateDataKey および GenerateDataKeyWithoutPlaintextオペレーションは、暗号化されたデータキーを返します。暗号化データを別のリージョンに移動し、新しいリージョンで KMS キーを使用してデータキーを暗号化するときに、このメソッドを使用できます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、「AWS SDK for Java API リファレンス」の「encrypt メソッド」を参照してください。

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
ByteBuffer plaintext = ByteBuffer.wrap(new byte[]{1,2,3,4,5,6,7,8,9,0});

EncryptRequest req = new EncryptRequest().withKeyId(keyId).withPlaintext(plaintext);
ByteBuffer ciphertext = kmsClient.encrypt(req).getCiphertextBlob();
```

C#

詳細については、AWS SDK for .NET の「Encrypt メソッド」を参照してください。

```
// Encrypt a data key
//
```

```
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
MemoryStream plaintext = new MemoryStream();
plaintext.Write(new byte[] { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 }, 0, 10);

EncryptRequest encryptRequest = new EncryptRequest()
{
    KeyId = keyId,
    Plaintext = plaintext
};
MemoryStream ciphertext = kmsClient.Encrypt(encryptRequest).CiphertextBlob;
```

Python

詳細については、AWS SDK for Python (Boto3) の「encrypt メソッド」を参照してください。

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00'

response = kms_client.encrypt(
    KeyId=key_id,
    Plaintext=plaintext
)

ciphertext = response['CiphertextBlob']
```

Ruby

詳細については、AWS SDK for Ruby の「encrypt インスタンスメソッド」を参照してください。

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
plaintext = "\x01\x02\x03\x04\x05\x06\x07\x08\x09\x00"

response = kmsClient.encrypt({
```

```
  key_id: key_id,
  plaintext: plaintext
})

ciphertext = response.ciphertext_blob
```

PHP

詳細については、AWS SDK for PHP の「Encrypt メソッド」を参照してください。

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$message = pack('c*',1,2,3,4,5,6,7,8,9,0);

$result = $KmsClient->encrypt([
    'KeyId' => $keyId,
    'Plaintext' => $message,
]);

$ciphertext = $result['CiphertextBlob'];
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の encrypt プロパティを参照してください。

```
// Encrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Plaintext = Buffer.from([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]);
kmsClient.encrypt({ KeyId, Plaintext }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { CiphertextBlob } = data;
      ...
  }
});
```

PowerShell

KMS キーのデータキーを暗号化するには、Invoke-KMSEncrypt コマンドレットを使用します。暗号文を MemoryStream (System.IO) オブジェクトとして返します。MemoryStreamこの MemoryStream オブジェクトは Invoke-KMSDecrypt コマンドレットへの入力として使用できます。

AWS KMS もまた、データキーを MemoryStream オブジェクトとして返します。この例では、プレーンテキストのデータキーをシミュレートするために、バイト配列を作成して MemoryStream オブジェクトに書き込みます。

Invoke-KMSEncrypt の Plaintext パラメータは、バイト配列 (byte[]) を受けとります。MemoryStream オブジェクトは必要ありません。 AWSPowerShell バージョン 4.0 以降、バイト配列とMemoryStreamオブジェクトを受け取るすべての AWSPowerShell モジュールのパラメータは、バイト配列、MemoryStreamオブジェクト、文字列、文字列配列、および FileInfo (System.IO) オブジェクトを受け入れます。FileInfoこれらのタイプのいずれも Invoke-KMSEncrypt に渡すことができます。

```
# Encrypt a data key

# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Simulate a data key
 # Create a byte array
[byte[]] $bytes = 1, 2, 3, 4, 5, 6, 7, 8, 9, 0

 # Create a MemoryStream
$plaintext = [System.IO.MemoryStream]::new()

 # Add the byte array to the MemoryStream
$plaintext.Write($bytes, 0, $bytes.length)

# Encrypt the simulated data key
$response = Invoke-KMSEncrypt -KeyId $keyId -Plaintext $plaintext

# Get the ciphertext from the response
$ciphertext = $response.CiphertextBlob
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

データキーの復号

データキーを復号するには、Decrypt オペレーションを使用します。

ciphertextBlob 指定する は、GenerateDataKey、、GenerateDataKeyWithoutPlaintextまたは Encrypt レスポンスの CiphertextBlobフィールドの値、または GenerateDataKeyPairまたは GenerateDataKeyPairWithoutPlaintextレスポンスの PrivateKeyCiphertextBlobフィールドである必要があります。また、Decrypt オペレーションを使用して、非対称 KMS キーのパブリックキーによって AWS KMS 外部で暗号化されたデータを復号することもできます。

対称暗号化 KMS キーで復号するときに KeyId パラメータは不要です。AWS KMS は、暗号文 blob 内のメタデータからのデータを暗号化するために使用された KMS キーを取得できます。ただし、ベストプラクティスは常に、使用している KMS キーを指定することです。この方法により、意図した KMS キーを使用することができ、信頼できない KMS キーを使用して暗号文が誤って復号されるのを防ぐことができます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、「AWS SDK for Java API リファレンス」の「decrypt メソッド」を参照してください。

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ByteBuffer ciphertextBlob = Place your ciphertext here;

DecryptRequest req = new
 DecryptRequest().withCiphertextBlob(ciphertextBlob).withKeyId(keyId);
ByteBuffer plainText = kmsClient.decrypt(req).getPlaintext();
```

C#

詳細については、AWS SDK for .NET の「Decrypt メソッド」を参照してください。

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

MemoryStream ciphertextBlob = new MemoryStream();
// Write ciphertext to memory stream

DecryptRequest decryptRequest = new DecryptRequest()
{
    CiphertextBlob = ciphertextBlob,
    KeyId = keyId
};
MemoryStream plainText = kmsClient.Decrypt(decryptRequest).Plaintext;
```

Python

詳細については、AWS SDK for Python (Boto3) の「decrypt メソッド」を参照してください。

```
# Decrypt a data key

# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
ciphertext = 'Place your ciphertext here'

response = kms_client.decrypt(
    CiphertextBlob=ciphertext,
    KeyId=key_id
)

plaintext = response['Plaintext']
```

Ruby

詳細については、AWS SDK for Ruby の「decrypt インスタンスメソッド」を参照してください。

```
# Decrypt a data key
```

```
# Replace the following example key ARN with any valid key identfier
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

response = kmsClient.decrypt({
  ciphertext_blob: ciphertext_packed,
  key_id: key_id
})

plaintext = response.plaintext
```

PHP

詳細については、AWS SDK for PHP の「Decrypt メソッド」を参照してください。

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$ciphertext = 'Place your cipher text blob here';

$result = $KmsClient->decrypt([
     'CiphertextBlob' => $ciphertext,
     'KeyId' => $keyId,
]);

$plaintext = $result['Plaintext'];
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の decrypt プロパティを参照してください。

```
// Decrypt a data key
//
// Replace the following example key ARN with any valid key identfier
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
```

```
const CiphertextBlob = 'Place your cipher text blob here';
kmsClient.decrypt({ CiphertextBlob, KeyId }, (err, data) => {
  if (err) console.log(err, err.stack); // an error occurred
  else {
    const { Plaintext } = data;
    ...
  }
});
```

PowerShell

データキーを復号するには、Invoke-KMSencrypt コマンドレットを使用します。

このコマンドレットは、プレーンテキストを MemoryStream (System.IO) オブジェクトとして返します。MemoryStreamプレーンテキストをバイト配列に変換するには、コマンドレットまたはバイト配列を MemoryStream オブジェクトに変換する関数 (変換モジュールの関数など) を使用します。

この例では、AWS KMS 暗号化コマンドレットが返す暗号テキストを使用するため、CiphertextBlob パラメータの値には MemoryStream オブジェクトを使用します。ただし、Invoke-KMSDecrypt の CiphertextBlob パラメータはバイト配列 (byte[]) を受けとります。MemoryStream オブジェクトは必要ありません。 AWSPowerShell バージョン 4.0 以降、バイト配列とMemoryStreamオブジェクトを受け取るすべての AWSPowerShell モジュールのパラメータは、バイト配列、MemoryStreamオブジェクト、文字列、文字列配列、および FileInfo (System.IO) オブジェクトを受け入れます。FileInfoこれらのタイプのいずれも Invoke-KMSDecrypt に渡すことができます。

```
# Decrypt a data key
# Replace the following example key ARN with any valid key identfier
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

[System.IO.MemoryStream]$ciphertext = Read-Host 'Place your cipher text blob here'

$response = Invoke-KMSDecrypt -CiphertextBlob $ciphertext -KeyId $keyId
$plaintext = $response.Plaintext
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

異なる AWS KMS key によるデータキーの再暗号化

暗号化されたデータキーを復号し、別の でデータキーをすぐに再暗号化するにはAWS KMS key、ReEncryptオペレーションを使用します。このオペレーションは AWS KMS 内のサーバー側で完全に実行されるため、プレーンテキストが AWS KMS の外部に公開されることはありません。

ciphertextBlob 指定する は、GenerateDataKey、、GenerateDataKeyWithoutPlaintextまたは Encrypt レスポンスの CiphertextBlobフィールドの値、または GenerateDataKeyPairまたは GenerateDataKeyPairWithoutPlaintextレスポンスの PrivateKeyCiphertextBlobフィールドである必要があります。また、ReEncrypt オペレーションを使用し、非対称 KMS キーのパブリックキーによって、AWS KMS 外部で暗号化されたデータを再暗号化することもできます。

対称暗号化 KMS キーで再暗号化するときに SourceKeyId パラメータは不要です。AWS KMS は、暗号文 blob 内のメタデータからのデータを暗号化するために使用された KMS キーを取得できます。ただし、ベストプラクティスは常に、使用している KMS キーを指定することです。この方法により、意図した KMS キーを使用することができ、信頼できない KMS キーを使用して暗号文が誤って復号されるのを防ぐことができます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、「AWS SDK for Java API リファレンス」の「reEncrypt メソッド」を参照してください。

```
// Re-encrypt a data key

ByteBuffer sourceCiphertextBlob = Place your ciphertext here;

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest req = new ReEncryptRequest();
req.setCiphertextBlob(sourceCiphertextBlob);
req.setSourceKeyId(sourceKeyId);
req.setDestinationKeyId(destinationKeyId);
ByteBuffer destinationCipherTextBlob = kmsClient.reEncrypt(req).getCiphertextBlob();
```

C#

詳細については、AWS SDK for .NET の「ReEncrypt メソッド」を参照してください。

```
// Re-encrypt a data key

MemoryStream sourceCiphertextBlob = new MemoryStream();
// Write ciphertext to memory stream

// Replace the following example key ARNs with valid key identfiers
String sourceKeyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String destinationKeyId = "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321";

ReEncryptRequest reEncryptRequest = new ReEncryptRequest()
{
    CiphertextBlob = sourceCiphertextBlob,
    SourceKeyId = sourceKeyId,
    DestinationKeyId = destinationKeyId
};
MemoryStream destinationCipherTextBlob =
 kmsClient.ReEncrypt(reEncryptRequest).CiphertextBlob;
```

Python

詳細については、AWS SDK for Python (Boto3) の「re_encrypt メソッド」を参照してください。

```
# Re-encrypt a data key
ciphertext = 'Place your ciphertext here'

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kms_client.re_encrypt(
    CiphertextBlob=ciphertext,
    SourceKeyId=source_key_id,
    DestinationKeyId=destination_key_id
```

```
)

destination_ciphertext_blob = response['CiphertextBlob']
```

Ruby

詳細については、AWS SDK for Ruby の「re_encrypt インスタンスメソッド」を参照してください。

```
# Re-encrypt a data key

ciphertext = 'Place your ciphertext here'
ciphertext_packed = [ciphertext].pack("H*")

# Replace the following example key ARNs with valid key identfiers
source_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
destination_key_id = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

response = kmsClient.re_encrypt({
  ciphertext_blob: ciphertext_packed,
  source_key_id: source_key_id,
  destination_key_id: destination_key_id
})

destination_ciphertext_blob = response.ciphertext_blob.unpack('H*')
```

PHP

詳細については、AWS SDK for PHP の「ReEncrypt メソッド」を参照してください。

```
// Re-encrypt a data key

$ciphertextBlob = 'Place your ciphertext here';

// Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321';

$result = $KmsClient->reEncrypt([
```

```
    'CiphertextBlob' => $ciphertextBlob,
    'SourceKeyId' => $sourceKeyId,
    'DestinationKeyId' => $destinationKeyId,
]);
```

Node.js

詳細については、 AWS SDK for JavaScript in Node.js の reEncrypt プロパティを参照してください。

```
// Re-encrypt a data key
const CiphertextBlob = 'Place your cipher text blob here';
// Replace the following example key ARNs with valid key identfiers
const SourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const DestinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321';

kmsClient.reEncrypt({ CiphertextBlob, SourceKeyId, DestinationKeyId }, (err, data)
 => {
  ...
});
```

PowerShell

同じ KMS キーまたは別の KMS キーで暗号文を再暗号化するには、Invoke-KMSReEncrypt コマンドレットを使用します。

この例では、AWS KMS 暗号化コマンドレットが返す暗号テキストを使用するため、CiphertextBlob パラメータの値には MemoryStream オブジェクトを使用します。ただし、Invoke-KMSReEncrypt の CiphertextBlob パラメータはバイト配列 (byte[]) を受けとります。MemoryStream オブジェクトは必要ありません。 AWSPowerShell バージョン 4.0 以降、バイト配列とMemoryStreamオブジェクトを受け取るすべての AWSPowerShell モジュールのパラメータは、バイト配列、MemoryStreamオブジェクト、文字列、文字列配列、および FileInfo (System.IO) オブジェクトを受け入れます。FileInfoこれらのタイプのいずれも Invoke-KMSReEncrypt に渡すことができます。

```
# Re-encrypt a data key

[System.IO.MemoryStream]$ciphertextBlob = Read-Host 'Place your cipher text blob
 here'
```

```
# Replace the following example key ARNs with valid key identfiers
$sourceKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$destinationKeyId = 'arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321'

$response = Invoke-KMSReEncrypt -Ciphertext $ciphertextBlob -SourceKeyId
 $sourceKeyId -DestinationKeyId $destinationKeyId
$reEncryptedCiphertext = $response.CiphertextBlob
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

キーポリシーの使用

このトピックの例では、AWS KMS API を使用して AWS KMS keys のキーポリシーを表示、変更します。

キーポリシー、IAM ポリシー、権限を使用して、KMS キーへのアクセスを管理する方法の詳細は、AWS KMS の認証とアクセスコントロール を参照してください。JSON ポリシードキュメントの記述と書式設定については、『 IAM ユーザーガイド』の「IAM JSON ポリシーリファレンス 」を参照してください。

トピック

- キーポリシー名のリスト化
- キーポリシーの取得
- キーポリシーの設定

キーポリシー名のリスト化

のキーポリシーの名前を取得するにはAWS KMS key、 ListKeyPoliciesオペレーションを使用します。返される唯一のキーポリシー名は、default です。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

Java の実装の詳細については、 AWS SDK for Java API リファレンスの listKeyPolicies メソッドを参照してください。

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListKeyPoliciesRequest req = new ListKeyPoliciesRequest().withKeyId(keyId);
ListKeyPoliciesResult result = kmsClient.listKeyPolicies(req);
```

C#

詳細については、AWS SDK for .NET の「ListKeyPolicies メソッド」を参照してください。

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

ListKeyPoliciesRequest listKeyPoliciesRequest = new ListKeyPoliciesRequest()
{
    KeyId = keyId
};
ListKeyPoliciesResponse listKeyPoliciesResponse =
 kmsClient.ListKeyPolicies(listKeyPoliciesRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_key_policies メソッド」を参照してください。

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
```

```
response = kms_client.list_key_policies(
    KeyId=key_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「list_key_policies インスタンスメソッド」を参照してください。

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_key_policies({
  key_id: key_id
})
```

PHP

詳細については、AWS SDK for PHP の「ListKeyPolicies メソッド」を参照してください。

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

$result = $KmsClient->listKeyPolicies([
    'KeyId' => $keyId
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の listKeyPolicies プロパティを参照してください。

```
// List key policies
//
// Replace the following example key ARN with a valid key ID or key ARN
```

```
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

kmsClient.listKeyPolicies({ KeyId }, (err, data) => {
  ...
});
```

PowerShell

デフォルトキーポリシーの名前を一覧表示するには、Get-KMSKeyPolicyList コマンドレットを使用します。

```
# List key policies

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$response = Get-KMSKeyPolicyList -KeyId $keyId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

キーポリシーの取得

のキーポリシーを取得するにはAWS KMS key、 GetKeyPolicyオペレーションを使用します。

GetKeyPolicy にはポリシー名が必要です。唯一の有効なポリシー名は、default です。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、 AWS SDK for Java API リファレンスの getKeyPolicy メソッドを参照してください。

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
```

```
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest req = new
 GetKeyPolicyRequest().withKeyId(keyId).withPolicyName(policyName);
GetKeyPolicyResult result = kmsClient.getKeyPolicy(req);
```

C#

詳細については、AWS SDK for .NET の「GetKeyPolicy メソッド」を参照してください。

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";

GetKeyPolicyRequest getKeyPolicyRequest = new GetKeyPolicyRequest()
{
    KeyId = keyId,
    PolicyName = policyName
};
GetKeyPolicyResponse getKeyPolicyResponse =
 kmsClient.GetKeyPolicy(getKeyPolicyRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「get_key_policy メソッド」を参照してください。

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'

response = kms_client.get_key_policy(
    KeyId=key_id,
    PolicyName=policy_name
)
```

Ruby

詳細については、AWS SDK for Ruby の「get_key_policy インスタンスメソッド」を参照してください。

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'

response = kmsClient.get_key_policy({
  key_id: key_id,
  policy_name: policy_name
})
```

PHP

詳細については、AWS SDK for PHP の「GetKeyPolicy メソッド」を参照してください。

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->getKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の getKeyPolicy プロパティを参照してください。

```
// Get the policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
```

```
const PolicyName = 'default';
kmsClient.getKeyPolicy({ KeyId, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

KMS キーのキーポリシーを取得するには、Get-KMSKeyPolicy コマンドレットを使用します。このコマンドレットは、Write-KMSKeyPolicy () コマンドで使用できる文字列 (System.StringPutKeyPolicy) としてキーポリシーを返します。JSON 文字列のポリシーをPSCustomObjectオブジェクトに変換するには、ConvertFrom-JSON コマンドレットを使用します。

```
# Get the policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'

$response = Get-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

キーポリシーの設定

KMS キーのキーポリシーを作成または置き換えるには、 PutKeyPolicyオペレーションを使用します。

PutKeyPolicy では、ポリシー名が必要です。唯一の有効なポリシー名は、default です。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、 AWS SDK for Java API リファレンスの putKeyPolicy メソッドを参照してください。

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleRole\"," +
                "    \"Effect\": \"Allow\"," +
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
                "  }]" +
                "}";

PutKeyPolicyRequest req = new
 PutKeyPolicyRequest().withKeyId(keyId).withPolicy(policy).withPolicyName(policyName);
kmsClient.putKeyPolicy(req);
```

C#

詳細については、AWS SDK for .NET の「PutKeyPolicy メソッド」を参照してください。

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String policyName = "default";
String policy = "{" +
                "  \"Version\": \"2012-10-17\"," +
                "  \"Statement\": [{" +
                "    \"Sid\": \"Allow access for ExampleUser\"," +
```

```
                "    \"Effect\": \"Allow\"," +
                // Replace the following example user ARN with a valid one
                "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/
ExampleKeyUserRole\"}," +
                "    \"Action\": [" +
                "      \"kms:Encrypt\"," +
                "      \"kms:GenerateDataKey*\"," +
                "      \"kms:Decrypt\"," +
                "      \"kms:DescribeKey\"," +
                "      \"kms:ReEncrypt*\"" +
                "    ]," +
                "    \"Resource\": \"*\"" +
                "  }]" +
                "}";

PutKeyPolicyRequest putKeyPolicyRequest = new PutKeyPolicyRequest()
{
    KeyId = keyId,
    Policy = policy,
    PolicyName = policyName
};
kmsClient.PutKeyPolicy(putKeyPolicyRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「put_key_policy メソッド」を参照してください。

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = """
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "Allow access for ExampleUser",
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
        "Action": [
            "kms:Encrypt",
```

```
            "kms:GenerateDataKey*",
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:ReEncrypt*"
        ],
        "Resource": "*"
    }]
}"""

response = kms_client.put_key_policy(
    KeyId=key_id,
    Policy=policy,
    PolicyName=policy_name
)
```

Ruby

詳細については、AWS SDK for Ruby の「put_key_policy インスタンスメソッド」を参照してください。

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
policy_name = 'default'
policy = "{" +
  "  \"Version\": \"2012-10-17\"," +
  "  \"Statement\": [{" +
  "    \"Sid\": \"Allow access for ExampleUser\"," +
  "    \"Effect\": \"Allow\"," +
  # Replace the following example user ARN with a valid one
  "    \"Principal\": {\"AWS\": \"arn:aws:iam::111122223333:role/ExampleKeyUserRole
\"}," +
  "    \"Action\": [" +
  "      \"kms:Encrypt\"," +
  "      \"kms:GenerateDataKey*\"," +
  "      \"kms:Decrypt\"," +
  "      \"kms:DescribeKey\"," +
  "      \"kms:ReEncrypt*\"" +
  "    ]," +
  "    \"Resource\": \"*\"" +
  "  }]" +
```

```
  "}"

response = kmsClient.put_key_policy({
  key_id: key_id,
  policy: policy,
  policy_name: policy_name
})
```

PHP

詳細については、AWS SDK for PHP の「PutKeyPolicy メソッド」を参照してください。

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$policyName = "default";

$result = $KmsClient->putKeyPolicy([
    'KeyId' => $keyId,
    'PolicyName' => $policyName,
    'Policy' => '{
        "Version": "2012-10-17",
        "Id": "custom-policy-2016-12-07",
        "Statement": [
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:user/root" },
            "Action": [ "kms:*" ],
            "Resource": "*" },
            { "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal":
               { "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole" },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*" }
```

```
        ]
    } '
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の putKeyPolicy プロパティを参照してください。

```
// Set a key policy for a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const PolicyName = 'default';
const Policy = `{
    "Version": "2012-10-17",
    "Id": "custom-policy-2016-12-07",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }
```

```
    ]
}`; // The key policy document

kmsClient.putKeyPolicy({ KeyId, Policy, PolicyName }, (err, data) => {
  ...
});
```

PowerShell

KMS キーのキーポリシーを設定するには、Write-KMSKeyPolicy コマンドレットを使用します。このコマンドレットは出力を返しません。コマンドが有効であることを確認するには、Get-KMSKeyPolicy コマンドレットを使用します。

Policy パラメータは文字列を受け取ります。リテラル文字列にするには、文字列を一重引用符で囲みます。リテラル文字列で継続文字や エスケープ文字を使用する必要はありません。

```
# Set a key policy for a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$policyName = 'default'
$policy = '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
            },
            "Action": [
                "kms:Encrypt*",
                "kms:GenerateDataKey*",
```

```
                "kms:Decrypt*",
                "kms:DescribeKey*",
                "kms:ReEncrypt*"
            ],
            "Resource": "*"
        }]
    }'

Write-KMSKeyPolicy -KeyId $keyId -PolicyName $policyName -Policy $policy
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

許可の使用

このトピックの例では、AWS KMS API を使用して、AWS KMS keys の権限の作成、表示、使用停止、取り消しをします。AWS KMS での許可の使用の詳細については、「AWS KMS でのグラント」を参照してください。

トピック

- グラントの作成
- 許可の表示
- 許可の廃止
- 許可の取り消し

グラントの作成

の許可を作成するにはAWS KMS key、 CreateGrantオペレーションを使用します。応答には、許可 ID と許可トークンのみが含まれます。許可に関する詳細情報を取得するには、「」に示すように、ListGrantsオペレーションを使用します許可の表示。

これらの例では、ExampleKeyUserロールを引き受けるユーザーが KeyIdパラメータで識別される KMS キーで GenerateDataKeyオペレーションを呼び出すことができる権限を作成します。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、AWS SDK for Java API リファレンスの createGrant メソッド を参照してください。

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey.toString();

CreateGrantRequest request = new CreateGrantRequest()
    .withKeyId(keyId)
    .withGranteePrincipal(granteePrincipal)
    .withOperations(operation);

CreateGrantResult result = kmsClient.createGrant(request);
```

C#

詳細については、AWS SDK for .NET の「CreateGrant メソッド」を参照してください。

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
String operation = GrantOperation.GenerateDataKey;

CreateGrantRequest createGrantRequest = new CreateGrantRequest()
{
    KeyId = keyId,
    GranteePrincipal = granteePrincipal,
    Operations = new List<string>() { operation }
};

CreateGrantResponse createGrantResult = kmsClient.CreateGrant(createGrantRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「create_grant メソッド」を参照してください。

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']

response = kms_client.create_grant(
    KeyId=key_id,
    GranteePrincipal=grantee_principal,
    Operations=operation
)
```

Ruby

詳細については、AWS SDK for Ruby の「create_grant インスタンスメソッド」を参照してください。

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee_principal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
operation = ['GenerateDataKey']

response = kmsClient.create_grant({
  key_id: key_id,
  grantee_principal: grantee_principal,
  operations: operation
})
```

PHP

詳細については、AWS SDK for PHP の「CreateGrant メソッド」を参照してください。

```
// Create a grant
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$granteePrincipal = "arn:aws:iam::111122223333:role/ExampleKeyUser";
$operation = ['GenerateDataKey']

$result = $KmsClient->createGrant([
    'GranteePrincipal' => $granteePrincipal,
    'KeyId' => $keyId,
    'Operations' => $operation
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の createGrant プロパティを参照してください。

```
// Create a grant
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const GranteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser';
const Operations: ["GenerateDataKey"];
kmsClient.createGrant({ KeyId, GranteePrincipal, Operations }, (err, data) => {
  ...
});
```

PowerShell

許可を作成するには、 New-KMSGrant コマンドレットを使用します。

```
# Create a grant

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$granteePrincipal = 'arn:aws:iam::111122223333:role/ExampleKeyUser'
$operation = 'GenerateDataKey'

$response = New-KMSGrant -GranteePrincipal $granteePrincipal -KeyId $keyId -
Operation $operation
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

許可の表示

KMS キーの許可に関する詳細情報を取得するには、 ListGrantsオペレーションを使用します。

> **Note**
>
> 通常、ListGrants レスポンスの GranteePrincipal フィールドには、グラントの被付与者プリンシパルが含まれます。ただし、権限の被付与者プリンシパルが AWS のサービスの場合、GranteePrincipal フィールドにはサービスプリンシパルが含まれます。これは、複数の異なる被付与者プリンシパルを表す場合があります。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

これらの例では、オプションの Limits パラメータを使用して、オペレーションが返す許可の数を決定します。

Java

Java の実装の詳細については、AWS SDK for Java API リファレンスの listGrants メソッドを参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
Integer limit = 10;

ListGrantsRequest req = new ListGrantsRequest().withKeyId(keyId).withLimit(limit);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

詳細については、AWS SDK for .NET の「ListGrants メソッド」を参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
int limit = 10;

ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    Limit = limit
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_grants メソッド」を参照してください。

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kms_client.list_grants(
    KeyId=key_id,
    Limit=10
)
```

Ruby

詳細については、AWS SDK for Ruby の「list_grants インスタンスメソッド」を参照してください。

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

response = kmsClient.list_grants({
```

```
  key_id: key_id,
  limit: 10
})
```

PHP

詳細については、AWS SDK for PHP の「ListGrants メソッド」を参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$limit = 10;

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'Limit' => $limit,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の listGrants プロパティを参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Limit = 10;
kmsClient.listGrants({ KeyId, Limit }, (err, data) => {
  ...
});
```

PowerShell

KMS キーのすべてのAWS KMS許可の詳細を表示するには、Get-KMSGrantList コマンドレットを使用します。

出力オブジェクトの数を制限するために、この例では、リストコマンドレットで非推奨の `Limit` パラメータの代わりに Select-Object コマンドレットを使用します。AWS Tools for PowerShell

での出力のページ分割については、「AWS Tools for PowerShell での出力ページ分割」を参照してください。

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$limit = 10

$response = Get-KMSGrantList -KeyId $keyId | Select-Object -First $limit
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

すべての ListGrants オペレーションで KMS キーを指定する必要があります。ただし、権限 ID または被付与者プリンシパルを指定することで、権限リストをさらにフィルタリングできます。次の例では、test-engineer ロールが被付与者プリンシパルのKMS キーの権限のみを取得します。

Java

Java の実装の詳細については、AWS SDK for Java API リファレンスの listGrants メソッドを参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String grantee = "arn:aws:iam::111122223333:role/test-engineer";

ListGrantsRequest req = new
 ListGrantsRequest().withKeyId(keyId).withGranteePrincipal(grantee);
ListGrantsResult result = kmsClient.listGrants(req);
```

C#

詳細については、AWS SDK for .NET の「ListGrants メソッド」を参照してください。

```
// Listing grants on a KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";
String grantee = "arn:aws:iam::111122223333:role/test-engineer";

ListGrantsRequest listGrantsRequest = new ListGrantsRequest()
{
    KeyId = keyId,
    GranteePrincipal = grantee
};
ListGrantsResponse listGrantsResponse = kmsClient.ListGrants(listGrantsRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「list_grants メソッド」を参照してください。

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kms_client.list_grants(
    KeyId=key_id,
    GranteePrincipal=grantee
)
```

Ruby

詳細については、AWS SDK for Ruby の「list_grants インスタンスメソッド」を参照してください。

```
# Listing grants on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
grantee = 'arn:aws:iam::111122223333:role/test-engineer'

response = kmsClient.list_grants({
```

```
  key_id: keyId,
  grantee_principal: grantee
})
```

PHP

詳細については、AWS SDK for PHP の「ListGrants メソッド」を参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
$grantee = 'arn:aws:iam::111122223333:role/test-engineer';

$result = $KmsClient->listGrants([
    'KeyId' => $keyId,
    'GranteePrincipal' => $grantee,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の listGrants プロパティを参照してください。

```
// Listing grants on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';
const Grantee = 'arn:aws:iam::111122223333:role/test-engineer';

kmsClient.listGrants({ KeyId, Grantee }, (err, data) => {
  ...
});
```

PowerShell

KMS キーのすべてのAWS KMS許可の詳細を表示するには、Get-KMSGrantList コマンドレットを使用します。

```
# Listing grants on a KMS key
```

```
# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'
$grantee = 'arn:aws:iam::111122223333:role/test-engineer'
$response = Get-KMSGrantList -KeyId $keyId -GranteePrincipal $grantee
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

許可の廃止

KMS キーのグラントを廃止にするには、 RetireGrantオペレーションを使用します。許可の使用が完了した後でクリーンアップを実行する場合に、許可を無効にする必要があります。

権限を使用停止にするには、権限トークン、または権限 ID と KMS キー ID の両方を指定します。このオペレーションでは、KMS キー ID が KMS キーの Amazon リソースネーム (ARN) である必要があります。許可トークンは CreateGrantオペレーションによって返されます。許可 ID は、CreateGrant および ListGrantsオペレーションによって返されます。

RetireGrant はレスポンスを返しません。有効であることを確認するには、 ListGrantsオペレーションを使用します。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、AWS SDK for Java API リファレンスの retireGrant メソッドを参照してください。

```
// Retire a grant
//
String grantToken = Place your grant token here;

RetireGrantRequest req = new RetireGrantRequest().withGrantToken(grantToken);
kmsClient.retireGrant(req);
```

C#

詳細については、AWS SDK for .NET の「RetireGrant メソッド」を参照してください。

```
// Retire a grant
//
String grantToken = "Place your grant token here";

RetireGrantRequest retireGrantRequest = new RetireGrantRequest()
{
    GrantToken = grantToken
};
kmsClient.RetireGrant(retireGrantRequest);
```

Python

詳細については、AWS SDK for Python (Boto3) の「retire_grant メソッド」を参照してください。

```
# Retire a grant

grant_token = Place your grant token here

response = kms_client.retire_grant(
    GrantToken=grant_token
)
```

Ruby

詳細については、AWS SDK for Ruby の「retire_grant インスタンスメソッド」を参照してください。

```
# Retire a grant

grant_token = Place your grant token here

response = kmsClient.retire_grant({
  grant_token: grant_token
})
```

PHP

詳細については、AWS SDK for PHP の「RetireGrant メソッド」を参照してください。

```
// Retire a grant
//
```

```
$grantToken = 'Place your grant token here';

$result = $KmsClient->retireGrant([
    'GrantToken' => $grantToken,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の retireGrant プロパティを参照してください。

```
// Retire a grant
//
const GrantToken = 'Place your grant token here';
kmsClient.retireGrant({ GrantToken }, (err, data) => {
  ...
});
```

PowerShell

許可を破棄するには、 Disable-KMSGrant コマンドレットを使用します。許可トークンを取得するには、 New-KMSgrant コマンドレットを使用します。GrantToken パラメータは文字列を受け取るため、Read-Host コマンドレットが返す出力を変換する必要はありません。

```
# Retire a grant

$grantToken = Read-Host -Message Place your grant token here
Disable-KMSGrant -GrantToken $grantToken
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

許可の取り消し

KMS キーへの許可を取り消すには、 RevokeGrantオペレーションを使用します。許可を取り消して、許可に依存しているオペレーションを明示的に拒否することができます。

クライアントオブジェクトを必要とする言語では、これらの例では「クライアントの作成」で作成した AWS KMS クライアントオブジェクトを使用します。

Java

詳細については、AWS SDK for Java API リファレンスの revokeGrant メソッドを参照してください。

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest req = new
 RevokeGrantRequest().withKeyId(keyId).withGrantId(grantId);
kmsClient.revokeGrant(req);
```

C#

詳細については、AWS SDK for .NET の「RevokeGrant メソッド」を参照してください。

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
String keyId = "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab";

// Replace the following example grant ID with a valid one
String grantId = "grant1";

RevokeGrantRequest revokeGrantRequest = new RevokeGrantRequest()
{
    KeyId = keyId,
    GrantId = grantId
};
kmsClient.RevokeGrant(revokeGrantRequest);
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、『AWS Tools for Windows PowerShell ユーザーガイド』を参照してください。

Python

詳細については、AWS SDK for Python (Boto3) の revoke_grant メソッドを参照してください。

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kms_client.revoke_grant(
    KeyId=key_id,
    GrantId=grant_id
)
```

Ruby

詳細については、AWS SDK for Ruby の「revoke_grant インスタンスメソッド」を参照してください。

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
key_id = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
grant_id = 'grant1'

response = kmsClient.revoke_grant({
  key_id: key_id,
  grant_id: grant_id
})
```

PHP

詳細については、AWS SDK for PHP の「RevokeGrant メソッド」を参照してください。

```
// Revoke a grant on a KMS key
```

```
//
// Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
$grantId = "grant1";

$result = $KmsClient->revokeGrant([
    'KeyId' => $keyId,
    'GrantId' => $grantId,
]);
```

Node.js

詳細については、 AWS SDK for in JavaScript Node.js の revokeGrant プロパティを参照してください。

```
// Revoke a grant on a KMS key
//
// Replace the following example key ARN with a valid key ID or key ARN
const KeyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab';

// Replace the following example grant ID with a valid one
const GrantId = 'grant1';
kmsClient.revokeGrant({ GrantId, KeyId }, (err, data) => {
  ...
});
```

PowerShell

許可を取り消すには、 Revoke-KMSgrant コマンドレットを使用します。

```
# Revoke a grant on a KMS key

# Replace the following example key ARN with a valid key ID or key ARN
$keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

# Replace the following example grant ID with a valid one
$grantId = 'grant1'
```

```
Revoke-KMSGrant -KeyId $keyId -GrantId $grantId
```

AWS KMS PowerShell コマンドレットを使用するには、AWS.Tools.KeyManagementService モジュールをインストールします。詳細については、「AWS Tools for Windows PowerShell ユーザーガイド」を参照してください。

AWS KMS API 呼び出しをテストする

AWS KMS を使用する場合は、AWS が API リクエストの認証に使用する認証情報が必要です。認証情報には、 KMS キーとエイリアスにアクセスするためのアクセス許可を含める必要があります。アクセス許可は、キーポリシー、IAM ポリシー、グラント、およびクロスアカウントアクセス制御によって決定されます。KMS キーへのアクセス制御に加えて、CloudHSM やカスタムキーストアへのアクセスを制御できます。

DryRun API パラメータを指定して、AWS KMS キーを使用するために必要なアクセス許可があることを確認できます。DryRun を使用して、AWS KMS API 呼び出しのリクエストパラメータが正しく指定されていることを確認することもできます。

トピック

- DryRun パラメータとは
- API DryRun を使用した の指定

DryRun パラメータとは

DryRun は、AWS KMS API 呼び出しが成功することを確認するために指定する API パラメータ (オプション) です。実際に AWS KMS を呼び出す前に、DryRun を使用して API 呼び出しをテストします。次のことを確認できます。

- AWS KMS キーを使用するために必要なアクセス許可があること。
- 呼び出しのパラメータが正しく指定されていること。

AWS KMS は 特定の API アクションでの DryRun パラメータの使用をサポートします。

- CreateGrant
- Decrypt
- 暗号化

- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign
- 検証
- VerifyMac

DryRun パラメータを使用すると料金が発生し、標準の API リクエストとして課金されます。AWS KMS の料金の詳細については、「AWS Key Management Service の料金」を参照してください。

DryRun パラメータを使用するすべての API リクエストは API のリクエストクォータに適用され、API リクエストクォータを超えた場合にスロットリング例外が発生する可能性があります。例えば、Decrypt を呼び出す際に DryRun を使用する場合でも DryRun を使用しない場合でも、同じ暗号化オペレーションクォータに対してカウントされます。詳細については、「AWS KMS リクエストのスロットリング」を参照してください。

AWS KMS API オペレーションへのすべての呼び出しは、AWS CloudTrail ログにイベントとしてキャプチャおよび記録されます。DryRun パラメータを指定するオペレーションの出力が CloudTrail ログに表示されます。詳細については、「AWS KMS による AWS CloudTrail API コールのログ記録」を参照してください。

API DryRun を使用した の指定

DryRun を使用するには、–dry-run パラメータをサポートする AWS CLI コマンドと AWS KMS API 呼び出しでパラメータを指定します。実行すると、呼び出しが成功するかどうかが AWS KMS によって検証されます。 DryRun を使用する AWS KMS 呼び出しは常に失敗し、呼び出しが失敗した理由に関する情報を含むメッセージが返されます。メッセージには次の例外が含まれます。

- DryRunOperationException - DryRun が指定されていなければリクエストは成功します。
- ValidationException - 間違った API パラメータが指定されたためリクエストが失敗しました。

- `AccessDeniedException` - KMS リソースで指定された API アクションを実行するアクセス許可がありません。

例えば、次のコマンドは CreateGrantオペレーションを使用し、`keyUserRole`ロールを引き受ける権限を持つユーザーが、指定された対称 KMS キー で Decrypt オペレーションを呼び出すことを許可する権限を作成します。`DryRun` パラメータが指定されます。

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

AWS KMS の結果整合性

システムの分散特性により、AWS KMS API は結果整合性モデルに従います。その結果、AWS KMS リソースへの変更は、それ以降に実行するコマンドにすぐに表示されない場合があります。

AWS KMS API 呼び出しを実行すると、変更が AWS KMS 全体に適用されるまでに若干の遅延が生じることがあります。通常、変更がシステム全体に反映されるまでに数秒もかかりませんが、場合によっては数分かかることがあります。この間、`InvalidStateException` または `NotFoundException` などの予期しない エラーが発生することがあります。例えば、`CreateKey` を呼び出した直後に `GetParametersForImport` を呼び出すと AWS KMS が `NotFoundException` を返すことがあります。

短い待機期間の後に自動的にオペレーションを再試行するよう AWS KMS クライアントで再試行戦略を設定することをお勧めします。詳細については、AWS SDK とツールのリファレンスガイドの「再試行動作」を参照してください。

API 呼び出し関連のグラントでは、グラントトークンを使用して潜在的な遅延を回避し、グラント内のアクセス許可をすぐに使用できます。詳細については、「結果整合性 (グラント用)」を参照してください。

リファレンス

次のリファレンスは、KMS キーの使用と管理に関する有用な情報を提供します。

- キータイプリファレンス。各 AWS KMS API オペレーションをサポートする KMS キーのタイプを一覧表示します。

  質問例: KMS キーを署名する RSA を有効または無効にすることはできますか?

- キーステータス表。KMS キーのキーステータスが、AWS KMS API オペレーションでの使用にどのように影響するかを示します。

  質問例: 削除が保留中である KMS キーのエイリアスを変更することはできますか?

- AWS KMS API アクセス許可のリファレンス。各 AWS KMS API オペレーションに必要なアクセス許可について説明します。

  検索するには: 別のAWSアカウントのキーGetKeyPolicyで を実行できますか？ IAM ポリシーで `kms:Decrypt` アクセス許可を付与できますか。

  - ViaService リファレンス。`kms:ViaService` 条件キーをサポートする AWS サービス一覧表示します。

    検索するには: `kms:ViaService`条件キーを使用して、Amazon からのアクセス許可のみを許可しますか ElastiCache？ Amazon Neptune の場合はどうですか。

- AWS KMS の料金。KMS キーの価格を一覧表示して説明します。

  質問例: 非対称キーの使用にかかる費用はどれくらいですか?

- AWS KMS リクエストクォータ。各アカウントとリージョンにおける AWS KMS API リクエストの 1 秒あたりのクォータを一覧表示します。

  質問例: Decrypt リクエストを毎秒いくつ実行できますか? カスタムキーストアの KMS キーで、いくつの Decrypt リクエストを実行できますか。

- AWS KMS リソースクォータ。AWS KMS リソースのクォータを一覧表示します。

  質問例: アカウントの各リージョンで何個の KMS キーを使用できますか? 各 KMS キーに何個のエイリアスを使用できますか。

- AWS KMS に統合される AWS サービス。作成、保存、管理するリソースの保護に KMS キーを使用する AWS サービスを一覧表示します。

質問例: Amazon Connect は、Connect リソースを保護するのに KMS キーを使用しますか?

ドキュメント履歴

このトピックでは、AWS Key Management Service デベロッパーガイドの重要な更新を説明しています。

トピック

最新の更新

以下の表は、このドキュメントの 2018 年 1 月以降の大きな変更点をまとめたものです。ここに表示されている主要な変更に加えて、その内容の説明と例を向上し、ユーザーから寄せられるフィードバックにも応える目的で、このドキュメントは頻繁に更新されます。重要な変更についての通知を受け取るには、RSS フィードをサブスクライブします。

この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

変更	説明	日付
マネージドポリシーの更新	AWS CloudHSM クラスターが含まれる VPC 内の変更を AWS KMS が監視することを可能にする新しい許可を `AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy` に追加して、障害が発生した場合に AWS KMS が明瞭なエラーメッセージを提供できるようにしました。	2023 年 11 月 10 日
機能更新	`DryRun` API パラメータのサポートを追加しました。	2023 年 7 月 5 日
機能更新	カスタムキーストアを除くすべてのタイプの AWS	2023 年 6 月 5 日

	KMS キーのキーマテリアルをインポートする機能の追加	
機能更新	Nitro Enclaves の AWS KMS API の更新	2023 年 3 月 10 日
機能更新	RSAES_PKCS1_V1_5 ラップアルゴリズムが廃止されました。AWS KMS は、米国国立標準技術研究所 (NIST) の暗号化キー管理ガイダンスに従って、2023 年 10 月 1 日までに RSAES_PKCS1_V1_5 のすべてのサポートを終了します。別のラップアルゴリズムをすぐに使い始めることをお勧めします。	2023 年 2 月 28 日
機能更新	外部キーストアのサポートが追加されました。この機能を使用すると、AWS 以外の暗号化キーを使って AWS リソースを保護できます。	2022 年 11 月 29 日
クォータの変更	各アカウントとリージョンにおける AWS KMS keys リソースクォータを 100,000 KMS キーに引き上げられました。	2022 年 7 月 8 日
機能更新	HMAC KMS キーをサポートする AWS リージョン の追加	2022 年 7 月 8 日
新しいトピック	AWS Key Management Service デベロッパーガイドのセキュリティの章にAWS KMS トピックの回復力を追加しました。	2022 年 6 月 14 日

新機能	HMAC コードを生成および検証する AWS KMS キーと API 操作のサポートを追加しました。	2022 年 4 月 19 日
ドキュメントの変更	カスタマーマスターキー (CMK) という条件が AWS KMS key および KMS キーに置き換えられています。	2021 年 8 月 30 日
新機能	マルチリージョンキーのサポートの追加、これは、同じキー ID とキーマテリアルを持つ異なるリージョンで、相互運用可能な KMS キーのセットです。マルチリージョンキーを使用して、あるリージョンのデータを暗号化し、別のリージョンのデータを復号できます。	2021 年 6 月 8 日
新機能	属性ベースのアクセスコントロール (ABAC) のサポートを追加しました。タグとエイリアスを使用して、AWS KMS keys へのアクセスを制御できます。	2020 年 12 月 17 日
新機能	VPC エンドポイントポリシーのサポートが追加されました。	2020 年 7 月 9 日
新しいコンテンツ	AWS KMS のセキュリティプロパティについて説明します。	2020 年 6 月 18 日

新機能	非対称 AWS KMS keys および非対称データキーのサポートが追加されました。	2019 年 11 月 25 日
更新された機能	AWS マネージドキー のキーポリシーを AWS KMS のコンソールで表示できます。この機能は、以前はカスタマーマネージドキーに限定されていました。	2019 年 11 月 15 日
新機能	AWS KMS への呼び出しに、TLS でハイブリッドポスト量子キー交換アルゴリズムを使用する方法について説明します。	2019 年 11 月 4 日
クォータの変更	KMS キーを管理する一部の API のリソースクォータを増やしました。	2019 年 9 月 18 日
クォータの変更	KMS キーごとの KMS キー、エイリアス、グラントのリソースクォータを変更しました。	2019 年 3 月 27 日
クォータの変更	カスタムキーストアで AWS KMS keys を使用する暗号化オペレーションに対して、1 秒あたりの共有リクエストクォータを変更しました。	2019 年 3 月 7 日

新機能	AWS KMS カスタムキーストアを作成して管理する方法について説明します。各キーストアは、ユーザーが所有し、制御する AWS CloudHSM クラスターによってサポートされています。	2018 年 11 月 26 日
新しいコンソール	新しい AWS KMS コンソールを使用する方法について説明します。このコンソールは IAM コンソールから独立しています。元のコンソールとその使用方法は、新しいコンソールに精通できるように、短期間ですが引き続き参照できます。	2018 年 11 月 7 日
クォータの変更	AWS KMS keys を使用するための共有リクエストクォータを変更しました。	2018 年 8 月 21 日
新しいコンテンツ	AWS Secrets Manager が AWS KMS を使用してシークレット内のシークレット値を暗号化する方法について説明します。	2018 年 7 月 13 日
新しいコンテンツ	DynamoDB が AWS KMSAWS KMS keys を使用してサーバー側の暗号化オプションをサポートする方法について説明します。	2018 年 5 月 23 日

新機能	VPC のプライベートエンドポイントを使用して、インターネット経由ではなく、AWS KMS に直接接続する方法について説明します。	2018 年 1 月 22 日

以前の更新

以下の表は、2018 年以前の AWS Key Management Service デベロッパーガイドの重要な変更点をまとめたものです。

この表のすべてのデータを表示するには、水平または垂直にスクロールする必要があります。

変更	説明	日付
新しいコンテンツ	キーのタグ付け に関するドキュメントを追加しました。	2017 年 2 月 15 日
新しいコンテンツ	AWS KMS keys のモニタリング および Amazon によるモニタリング CloudWatch に関するドキュメントを追加しました。	2016 年 8 月 31 日
新しいコンテンツ	インポートされたキーマテリアル に関するドキュメントを追加しました。	2016 年 8 月 11 日
新しいコンテンツ	IAM ポリシー、アクセス許可に関するリファレンス、および 条件キー のドキュメントを追加しました。	2016 年 7 月 5 日
更新	ドキュメントの「認証とアクセスコントロール」章の一部を更新しました。	2016 年 7 月 5 日

変更	説明	日付
更新	新しいデフォルトクォータを反映するように クォータ ページを更新しました。	2016 年 5 月 31 日
更新	新しいデフォルトのクォータを反映するように クォータ ページを更新し、わかりやすさと正確さを向上させるためにグラントトークンドキュメントを更新しました。	2016 年 4 月 11 日
新しいコンテンツ	複数の IAM プリンシパルが KMS キーにアクセスできるようにする および IP アドレス条件の使用 に関するドキュメントを追加しました。	2016 年 2 月 17 日
更新	明確さと正確性を向上させるため、「AWS KMS のキーポリシー」および「キーポリシーの変更」ページを更新しました。	2016 年 2 月 17 日
更新	内容をより明確にするために「 キーの管理」トピックのページが更新されました。	2016 年 1 月 5 日
新しいコンテンツ	AWS CloudTrail で AWS KMS を使用する方法 に関するドキュメントを追加しました。	2015 年 11 月 18 日
新しいコンテンツ	「キーポリシーの変更」の手順を追加しました。	2015 年 11 月 18 日

変更	説明	日付
更新	「Amazon Relational Database Service (Amazon RDS) が AWS KMS を使用する方法」に関するドキュメントが更新されました。	2015 年 11 月 18 日
新しいコンテンツ	が WorkSpaces を使用する方法 AWS KMS に関するドキュメントを追加しました。	2015 年 11 月 6 日
更新	内容をより明確にするために [AWS KMS のキーポリシー] ページが更新されました。	2015 年 10 月 22 日
新しいコンテンツ	AWS KMS keys を削除する に関するドキュメント (アラームを作成する、KMS キーの過去の使用状況を確認する の関連ドキュメントなど) を追加しました。	2015 年 10 月 15 日
新しいコンテンツ	AWS KMS keys へのアクセスを特定する に関するドキュメントを追加しました。	2015 年 10 月 15 日
新しいコンテンツ	AWS KMS キーのキーステータス に関するドキュメントを追加しました。	2015 年 10 月 15 日
新しいコンテンツ	Amazon Simple Email Service (Amazon SES) が AWS KMS を使用する方法 に関するドキュメントを追加しました。	2015 年 10 月 1 日

変更	説明	日付
更新	新しいリクエストクォータの説明がある クォータ ページを更新しました。	2015 年 8 月 31 日
新しいコンテンツ	AWS KMS の使用料金に関する情報を追加しました。「AWS KMS 料金表」を参照してください。	2015 年 8 月 14 日
新しいコンテンツ	AWS KMS クォータ にリクエストクォータを追加しました。	2015 年 6 月 11 日
新しいコンテンツ	`UpdateAlias` オペレーションの使用方法を示す新しい Java コードサンプルを追加しました。エイリアスの更新 を参照してください。	2015 年 6 月 1 日
更新	AWS Key Management Service リージョンの一覧を『AWS 全般のリファレンス』に移動しました。	2015 年 5 月 29 日
新しいコンテンツ	Amazon EMR が AWS KMS を使用する方法 に関するドキュメントを追加しました。	2015 年 1 月 28 日
新しいコンテンツ	Amazon が WorkMail を使用する方法 AWS KMS に関するドキュメントを追加しました。	2015 年 1 月 28 日

変更	説明	日付
新しいコンテンツ	Amazon Relational Database Service (Amazon RDS) が AWS KMS を使用する方法 に関するドキュメントを追加しました。	2015 年 1 月 6 日
新しいコンテンツ	Amazon Elastic Transcoder が AWS KMS を使用する方法 に関するドキュメントを追加しました。	2014 年 11 月 24 日
新規ガイド	AWS Key Management Service デベロッパーガイドを導入しました。	2014 年 11 月 12 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。