ユーザーガイド

AWS リソースとタグエディタのタグ付け

Version 1.0

AWS リソースとタグエディタのタグ付け: ユーザーガイド

Table of Contents

タグエディタとは

タグエディタを使用すると、タグを効果的に管理できます。タグは、 AWS リソースを整理するためのメタデータとして機能するキーと値のペアです。ほとんどの AWS リソースでは、リソースの作成時にタグを追加するオプションがあります。リソースの例としては、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、Amazon Simple Storage Service (Amazon S3) バケット、 のシークレットなどがあります AWS Secrets Manager。

> **Important**
>
> 個人を特定できる情報 (PII) やその他の機密情報や機密情報をタグに保存しないでください。タグを使用して、課金および管理サービスを提供します。タグは、プライベートデータや機密データに使用することを意図していません。

タグは、リソースの管理、識別、整理、検索、フィルタリングに役立ちます。タグを作成することで、リソースを目的、所有者、環境その他の基準別に分類できます。

各タグは 2 つの部分で構成されます。

- タグキー (例： `CostCenter`、`Environment`、または `Project`)。タグキーでは、大文字と小文字が区別されます。
- タグ値 (例: `111122223333` または `Production`)。タグキーと同様に、タグ値は大文字と小文字が区別されます。

> **Note**
>
> タグキーは大文字と小文字が区別されますが、 IAMには、ケーシングでのみ異なるタグキーの適用を防ぐためのIAMリソースに対する追加の検証があります。ケーシングのみが異なるキーは使用しないでください。代わりに、サービスコントロールポリシー (SCPs) を使用できます。これにより、組織内のIAMユーザーとIAMロールが利用できるアクセス許可の最大数を一元的に制御できます。

リソースのタグ付け方法

AWS リソースにタグを追加するには、次の 3 つの方法があります。

- AWS のサービス API オペレーション – タグ付けAPIオペレーションは を直接サポートしています AWS のサービス。各 AWS のサービス が提供するタグ付け機能については、ドキュメントAWS インデックス のサービスのドキュメントを参照してください。
- タグエディタコンソール – タグエディタコンソールによるタグ付けをサポートするサービスもあります。
- Resource Groups Tagging API – ほとんどのサービスでは、 を使用したタグ付けもサポートされていますAWS Resource Groups Tagging API。

> **Note**
>
> AWS Service Catalog TagOptions Library を使用して、プロビジョニングされた製品のタグを簡単に管理することもできます。TagOption は、Service Catalog で管理されるキーと値のペアです。タグではありませんが AWS 、 に基づいて AWS タグを作成するためのテンプレートとして機能します TagOption。

AWSのコストが発生するすべてのサービスのリソースにタグ付けできます。以下のサービスでは、は、お客様のユースケースをより適切に満たすためのタグ付けをサポートする新しい代替案 AWS を推奨 AWS のサービス します。

Amazon Cloud Directory	Amazon CloudSearch	Amazon Cognito Sync
AWS Data Pipeline	Amazon Elastic Transcoder	Amazon Machine Learning
AWS OpsWorks Stacks	Amazon S3 Glacier Direct	Amazon SimpleDB
Amazon WorkSpaces Application Manager	AWS DeepLens	

詳細

このページでは、 AWS リソースのタグ付けに関する一般的な情報を提供します。特定の AWS サービスのリソースのタグ付けの詳細については、そのドキュメントを参照してください。タグ付けに関する適切な情報源を以下に示します。

- の詳細については AWS Resource Groups Tagging API、「Resource Groups Tagging APIリファレンスガイド」を参照してください。
- 各 AWS のサービス が提供するタグ付け機能の詳細については、ドキュメントAWS インデックスのサービスのドキュメントを参照してください。
- IAM ポリシーでタグを使用して AWS リソースを表示および操作できるユーザーを制御する方法については、IAM「 ユーザーガイド」の「タグを使用したIAMユーザーとロールへのアクセスとアクセスの制御」を参照してください。

ベストプラクティスと戦略

このセクションでは、 AWS リソースにタグを付け、タグエディタを使用する際のベストプラクティスと戦略について説明します。

タグ付けのベストプラクティス

AWS リソースのタグ付け戦略を作成するときは、ベストプラクティスに従います。

- 個人を特定できる情報 (PII) やその他の機密情報や機密情報をタグに追加しないでください。タグには、請求を含む多くの AWS のサービスからアクセスできます。タグは、プライベートデータや機密データに使用することを意図していません。
- タグには、標準化された、大文字と小文字の区別がある形式を使用し、すべてのリソースタイプに一貫して適用します。
- リソースアクセスコントロールの管理、コスト追跡、オートメーション、整理など、複数の目的に対応したタグガイドラインを考慮します。
- 自動化されたツールを使用して、リソースタグを管理できます。タグエディタと Resource Groups Tagging API を使用すると、タグをプログラムで制御できるため、タグとリソースの自動管理、検索、フィルタリングが容易になります。
- タグは、多めに使用します。
- ビジネス要件の変化に合わせてタグを変更するのは簡単ですが、将来の変更の影響を考慮してください。たとえば、アクセス制御タグを変更した場合、そのタグを参照してリソースへのアクセスを制御するポリシーも更新する必要があります。
- AWS Organizationsを使用してタグポリシーを作成およびデプロイすることで、組織が採用するタグ付け標準を自動的に適用することができます。タグポリシーでは、有効なキー名と各キーに有効な値を定義するタグ付けルールを指定することができます。モニタリングのみを選択して、既存のタグを評価し、クリーンアップすることもできます。選択した標準にタグが準拠したら、タグポリ

シーで適用を有効にして、非準拠のタグが作成されないようにすることができます。詳細については、AWS Organizations ユーザーガイドのタグポリシーを参照してください。

タグの命名に関するベストプラクティス

ここでは、タグに関する命名規則に関するベストプラクティスについて説明します。

AWS タグのキー名では大文字と小文字が区別されるため、一貫して使用してください。たとえば、タグキーの `CostCenter` と `costcenter` は異なります。一方のタグキーは財務分析とレポート用のコスト配分タグとして設定され、もう一方は同じ用途には設定されていないかもしれません。

多数のタグは、 によって事前定義 AWS されているか、さまざまな によって自動的に作成されます AWS のサービス。多くのAWS 生成されたタグは、すべて小文字のキー名を使用し、名前に含まれる単語はハイフンで区切られ、タグのソースサービスを識別するプレフィックスにコロンが続きます。例えば、以下を参照してください。

- `aws:ec2spot:fleet-request-id` は、インスタンスを起動した Amazon EC2スポットインスタンスリクエストを識別するタグです。
- `aws:cloudformation:stack-name` は、リソースを作成した AWS CloudFormation スタックを識別するタグです。
- `elasticbeanstalk:environment-name` は、リソースを作成したアプリケーションを識別するタグです。

次のルールを使用してタグに名前を付けることを検討してください。

- 単語にはすべて小文字を使用してください。
- 単語を区切るにはハイフンを使用してください。
- プレフィックスに続けてコロンを付けると、組織名または省略名を識別できます。

例えば、 という名前の架空の会社の場合AnyCompany、次のようなタグを定義できます。

- `anycompany:cost-center` のタグは、内部のコストセンターのコードを識別するのに使用。
- `anycompany:environment-type` のタグは、開発、テスト、本番のいずれの環境であるかを識別するのに使用。
- `anycompany:application-id` のタグは、リソースが作成されたアプリケーションを識別するのに使用。

プレフィックスを使用すると、タグが組織で定義されているとおりに明確に認識され、 AWS や使用しているサードパーティーのツールでは認識されません。すべて小文字を使用し、単語をハイフンで区切ることにより、タグ名に大文字を使用した場合の混乱を避けることができます。例えば、anycompany:project-id の方が、ANYCOMPANY:ProjectID、anycompany:projectID、Anycompany:ProjectId よりも覚えるのが簡単です。

タグの命名制限と要件

タグには、次の基本的な命名要件と使用要件が適用されます。

- 各リソースは、最大 50 個のユーザー作成タグを持つことができます。
- aws: で始まるシステム作成タグは AWS に使用するために予約されており、この制限にはカウントされません。aws: プレフィックスで始まるタグを編集または削除することはできません。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- タグキーは、UTF-8 で 1 ~ 128 文字の Unicode 文字である必要があります。
- タグ値は、UTF-8 で最小 0 文字、最大 256 文字の Unicode 文字である必要があります。
- 使用できる文字はサービスによって異なります AWS 。特定の AWS サービスのリソースにタグを付けるために使用できる文字については、そのドキュメントを参照してください。一般に、使用できる文字は、文字、数字、UTF-8 で表されるスペース、および _ . : / = + - @ です。
- タグのキーと値では、大文字と小文字が区別されます。ベストプラクティスとして、タグを大文字にするための戦略を決定し、その戦略をすべてのリソースタイプにわたって一貫して実装します。たとえば、Costcenter、costcenter、CostCenter のいずれを使用するかを決定し、すべてのタグに同じ規則を使用します。大文字と小文字の扱いについて、同様のタグに整合性のない規則を使用することは避けてください。

一般的なタグ付け戦略

以下のタグ付け戦略を使用すると、 AWS リソースの識別と管理に役立ちます。

内容

- アクセス制御のタグ
- タグ付けのガバナンス

リソース整理のタグ

タグは、 で AWS リソースを整理する良い方法です AWS Management Console。タグと共にリソースが表示されるように設定したり、タグで検索やフィルタリングを行ったりできます。 AWS Resource Groups サービスを使用すると、1 つ以上のタグまたはタグの一部に基づいて AWS リソースのグループを作成できます。また、 AWS CloudFormation スタックでの出現に基づいてグループを作成することもできます。リソースグループとタグエディタを使用すると、複数のサービス、リソース、リージョンで構成されるアプリケーションのデータを 1 か所にまとめて表示できます。

コスト配分のタグ

AWS Cost Explorer と詳細な請求レポートを使用すると、 AWS コストをタグ別に分類できます。通常、コストセンター/ビジネスユニット、顧客、プロジェクトなどのビジネスタグを使用して、 AWS コストを従来のコスト配分ディメンションに関連付けます。ただし、コスト配分レポートで使用できるタグに制限はありません。特定のアプリケーション、環境、コンプライアンスプログラムなど、技術やセキュリティに関するディメンションを使って、コストの関連付けを行うことができます。

一部のサービスでは、 が AWS生成するcreatedByタグをコスト配分の目的で使用して、そうしないと分類されない可能性のあるリソースを考慮するのに役立ちます。createdBy タグは、サポートされている AWS のサービスとリソースにのみ使用できます。その値には、特定の APIまたはコンソールイベントに関連付けられたデータが含まれます。詳細については、AWS Billing and Cost Management ユーザーガイドの「AWS生成コスト配分タグ」を参照してください。

オートメーションのタグ

リソースまたはサービスに固有のタグは、多くの場合、オートメーションアクティビティ中にリソースをフィルタリングする目的で使用します。オートメーションタグは、自動タスクのオプトインまたはオプトアウト、またはアーカイブ、更新、削除の対象となるリソースのバージョンの特定に使用します。たとえば、オートメーションにした start または stop スクリプトを実行して業務時間外に開発環境をオフにすれば、コストが削減できます。このシナリオでは、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスタグは、このアクションをオプトアウトするインスタンスを識別する簡単な方法です。古い Amazon EBSスナップショット、、 out-of-dateまたはローリング Amazon スナップショットを検索して削除するスクリプトの場合、スナップショットタグは検索条件のディメンションを追加することができます。

アクセス制御のタグ

IAM ポリシーはタグベースの条件をサポートしており、特定のタグまたはタグ値に基づいてIAMアクセス許可を制限できます。例えば、IAMユーザーまたはロールのアクセス許可には、タグに基づいて特定の環境 (開発、テスト、本番稼働など) へのEC2API呼び出しを制限する条件を含めることができます。同じ戦略を使用して、特定の Amazon Virtual Private Cloud (Amazon VPC) ネットワークへのAPI呼び出しを制限できます。タグベースのリソースレベルのIAMアクセス許可のサポートは、サービス固有です。アクセス制御にタグベースの条件を使用する場合は、タグを変更できるユーザーを定義することで、タグの変更を制限してください。タグを使用して AWS リソースAPIへのアクセスを制御する方法の詳細については、「 ユーザーガイド」のAWS 「 と連携する のサービスIAMIAM」を参照してください。

タグ付けのガバナンス

効果的なタグ付け戦略では、標準化されたタグを使用し、 AWS リソース全体に一貫してプログラムで適用します。リアクティブアプローチとプロアクティブアプローチの両方を使用して、 AWS 環境内のタグを管理できます。

- リアクティブガバナンスは、Resource Groups Tagging API、 AWS Config ルール、カスタムスクリプトなどのツールを使用して適切にタグ付けされていないリソースを見つけるためのものです。リソースを手動で検索するには、タグエディタと請求明細レポートを使用します。
- プロアクティブガバナンスは AWS CloudFormation、、Service Catalog、 のタグポリシー、またはIAMリソースレベルのアクセス許可などのツールを使用して AWS Organizations、リソースの作成時に標準化されたタグが一貫して適用されるようにします。

  例えば、 プロパティを使用して AWS CloudFormation `Resource Tags`リソースタイプにタグを適用できます。サービス· カタログ では、ポートフォリオと製品タグを追加すれば、製品の開始時に自動的にポートフォリオと製品タグの組み合わせが適用されます。より厳格なプロアクティブガバナンスには、自動タスクが含まれます。例えば、Resource Groups Tagging を使用して環境のタグAPIを検索 AWS したり、スクリプトを実行して不適切にタグ付けされたリソースを隔離または削除したりできます。

カテゴリのタグ付け

タグを最も効果的に使用している企業は、ビジネス関連のタググループを作成し、リソースを技術、ビジネス、セキュリティといったディメンションで整理しています。自動プロセスを使用してインフラストラクチャを管理する企業は、それに加えてオートメーション関連のタグも使用します。

技術タグ	オートメーションのタグ	ビジネスタグ	セキュリティタグ
• 名前 — 個々のリソースを識別する • アプリケーション ID — 特定のアプリケーションに関連するリソースを特定する • アプリケーションロール — 特定のリソース (ウェブサーバー、メッセージブローカー、データベースなど) の機能について説明する • クラスター — 共通の構成を共有し、アプリケーションに対して特定の機能を実行するリソースファーム • 環境 — 開発、テスト、本番稼働用リソースを区別する • バージョン — リソースまたはアプリケーションのバージョンを区別するのに役立つ	• 日付/時刻 — リソースの開始、停止、削除、またはローテーションを行う日付または時刻 • オプトイン/オプトアウト — インスタンスの開始、停止、サイズ変更などの自動アクティビティにそのリソースを含めるかどうか • セキュリティ — Amazon VPCフローログの暗号化や有効化などの要件を決定し、追加の調査が必要なルートテーブルやセキュリティグループを特定します。	• プロジェクト — リソースがサポートするプロジェクト • 所有者 — リソースの責任者 • コストセンター/ビジネスユニット — リソースに関連付けられたコストセンターまたはビジネスユニットで、通常はコストの配分と追跡に使用する • 顧客 — リソースグループを利用するクライアント	• 機密性 — リソースがサポートするデータ機密性レベルの識別子 • コンプライアンス — 特定のコンプライアンス要件に準拠する必要があるワークロードの識別子

タグエディタ を開始します。

> **Important**
>
> 個人を特定できる情報 (PII) やその他の機密情報や機密情報をタグに保存しないでください。タグを使用して、課金および管理サービスを提供します。タグは、プライベートデータや機密データに使用することを意図していません。

複数のリソースにタグを一度に追加する、あるいは複数のリソースのタグを一度に編集または削除するには、タグエディタを使用します。タグエディタを使用してタグ付けするリソースを検索し、検索結果からそのリソースのタグを管理します。

タグエディタを起動するには

1. にサインインします。AWS Management Console.
2. 次のいずれかのステップを実行します。
   - サービス を選択してください。管理とガバナンスで、リソースグループとタグエディタ を選択します。左側のナビゲーションペインで、タグエディタを選択します。
   - 直接リンクを使用します。 AWS タグエディタ コンソール 。

すべてのリソースが適用されるタグを持つことができるわけではありません。タグエディタ がサポートするリソースの詳細については、「」の「サポートされているリソースタイプ」の「タグエディタのタグ付けhttps://docs.aws.amazon.com/ARG/latest/userguide/supported-resources.html」列を参照してください。 AWS Resource Groups ユーザーガイド。タグ付けするリソースタイプがサポートされていない場合は、 AWS コンソールウィンドウの左下隅にあるフィードバックを選択して、 を把握します。

リソースのタグ付けに必要なアクセス許可やロールの詳細については、「アクセス許可の設定」を参照してください。

トピック

- タグエディタ を使用するための前提条件
- アクセス許可の設定

タグエディタ を使用するための前提条件

リソースにタグを付ける作業を開始する前に、 がアクティブであることを確認してください。 AWS アカウント 既存のリソースと、リソースにタグを付けてグループを作成するための適切な権限を持つ 。

トピック

- にサインアップする AWS アカウント
- 管理アクセスを持つユーザーを作成する
- リソースの作成

にサインアップする AWS アカウント

をお持ちでない場合 AWS アカウントで、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/サインアップ を開きます。
2. オンラインの手順に従います。

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

   にサインアップするとき AWS アカウント、 AWS アカウントのルートユーザー が作成されます。ルートユーザーはすべての にアクセスできます AWS のサービス アカウントの および リソース。セキュリティのベストプラクティスとして、ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用してルートユーザーアクセスが必要なタスクを実行してください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示し、アカウントを管理できます。

管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

   ルートユーザーを使用してサインインする方法については、「」の「ルートユーザーとしてサインインする」を参照してください。 AWS サインイン ユーザーガイド。
2. ルートユーザーの多要素認証 (MFA) を有効にします。

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

管理アクセスを持つユーザーを作成する

1. IAM Identity Center を有効にします。

   手順については、「 の有効化」を参照してください。 AWS IAM Identity Center ()AWS IAM Identity Center ユーザーガイド。
2. IAM Identity Center で、ユーザーに管理アクセス権を付与します。

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

管理アクセス権を持つユーザーとしてサインインする

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

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

追加のユーザーにアクセス権を割り当てる

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

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

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

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

リソースの作成

には リソースが必要です AWS アカウント をタグ付けします。サポートされているリソースタイプの詳細については、「」の「サポートされているリソースタイプ」の「タグエディタのタグ付けhttps://docs.aws.amazon.com/ARG/latest/userguide/supported-resources.html」列を参照してください。 AWS Resource Groups ユーザーガイド。

アクセス許可の設定

タグエディタ を最大限に活用するには、リソースをタグ付けする、またはリソースのタグキーとタグ値を表示するための追加アクセス許可が必要になる場合があります。これらのアクセス許可は次のように分類されます。

- 個々のサービスに対するアクセス許可。これらのサービスからのリソースをタグ付けし、リソースグループに含めることができます。
- タグエディタ コンソールを使用するために必要なアクセス許可。

管理者の場合は、 を使用してポリシーを作成することで、ユーザーに許可を付与できます。 AWS Identity and Access Management （ IAM) サービス。まずIAMロール、ユーザー、またはグループを作成し、必要なアクセス許可でポリシーを適用します。IAM ポリシーの作成とアタッチの詳細については、「ポリシーの使用」を参照してください。

個々のサービスに対するアクセス許可

> **Important**
>
> このセクションでは、他の のリソースにタグを付ける場合に必要となるアクセス許可について説明します。 AWS サービスコンソールと APIs。

リソースにタグを追加するには、リソースが属するサービスに必要なアクセス許可が必要です。例えば、Amazon EC2インスタンスにタグを付けるには、Amazon などAPI、そのサービスの のタグ付けオペレーションに対するアクセス許可が必要です。 EC2CreateTags オペレーション。

タグエディタ コンソールを使用するために必要なアクセス許可

タグエディタ コンソールを使用してリソースを一覧表示およびタグ付けするには、 でユーザーのポリシーステートメントに次のアクセス許可を追加する必要がありますIAM。次のいずれかを追加できます。 AWS によって維持され、最新の状態に保たれる マネージドポリシー AWS、または独自のカスタムポリシーを作成して管理できます。

使用 AWS タグエディタ のアクセス許可の マネージドポリシー

タグエディタ は以下をサポートします。 AWS ユーザーにアクセス許可の事前定義されたセットを提供するために使用できる マネージドポリシー。これらのマネージドポリシーは、作成した他のポリシーと同様に、任意のロール、ユーザー、グループにアタッチできます。

ResourceGroupsandTagEditorReadOnlyAccess

このポリシーは、アタッチされたIAMロールまたはユーザーに、両方の読み取り専用オペレーションを呼び出すアクセス許可を付与します。 AWS Resource Groups およびタグエディタ。リソースのタグを読み取るには、別のポリシーを使用して、そのリソースに対するアクセス許可も必要です。詳細については、以下の重要な注意事項を参照してください。

ResourceGroupsandTagEditorFullAccess

このポリシーは、アタッチされたIAMロールまたはユーザーに、タグエディタで Resource Groups オペレーションと読み取り/書き込みタグオペレーションを呼び出すアクセス許可を付与します。リソースタグに対する読み取りまたは書き込みを行うには、別のポリシーを使用して、そのリソースに対するアクセス許可も必要です。詳細については、以下の重要な注意事項を参照してください。

> **Important**
>
> 上記の 2 つのポリシーは、タグエディタ のオペレーションを呼び出し、タグエディタ コンソールを使用するアクセス許可を付与します。しかしながら、オペレーションを呼び出すアクセス許可だけでなく、アクセスしようとしているタグがある特定のリソースに対する適切なアクセス許可も必要です。タグへのアクセス許可を付与するには、次のいずれかのポリシーをアタッチする必要があります。

- - AWS マネージドポリシー ReadOnlyAccess は、すべてのサービスのリソースの読み取り専用オペレーションにアクセス許可を付与します。 AWS は、このポリシーを新しい で自動的に最新の状態に保つ AWS のサービス 利用可能になると、 が実行されます。
- 多くの サービスはサービス固有の読み取り専用を提供します AWS そのサービスによって提供されるリソースのみにアクセスを制限するために使用できる マネージドポリシー。例えば、Amazon EC2は AmazonEC2ReadOnlyAccess.
- ユーザーがアクセスできるようにするいくつかのサービスとリソースに対して、限定される読み取り専用オペレーションにのみアクセス許可を付与する独自のポリシーを作成することができます。このポリシーでは、許可リスト戦略または拒否リスト戦略のいずれかを使用します。

  許可リスト戦略では、ポリシーで明示的に許可するまで、アクセスはデフォルトで拒否されるという事実を利用します。そのため、次の例のようなポリシーを使用できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [ "tag:*" ],
            "Resource": "<ARNs of resources to allow tagging>"
        }
    ]
}
```

  または、明示的にブロックするリソース以外のすべてのリソースへのアクセスを許可する拒否リスト戦略を使用することもできます。これには、アクセスを許可する関連ユーザーに適用される別のポリシーが必要です。次のポリシー例では、Amazon リソースネーム () でリストされている特定のリソースへのアクセスを拒否しますARN。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
             "Action": [ "tag:*" ],
            "Resource": "<ARNs of resources to disallow tagging>"
        }
    ]
```

```
}
```

タグエディタ のアクセス許可を手動で追加する

- `tag:*` (このアクセス許可は、すべての タグエディタ でのアクションを許可します。代わりに、ユーザーが使用できるアクションを制限する場合は、アスタリスクを特定のアクション、またはカンマで区切ったアクションのリストに置き換えることができます)
- `tag:GetResources`
- `tag:TagResources`
- `tag:UntagResources`
- `tag:getTagKeys`
- `tag:getTagValues`
- `resource-explorer:*`
- `resource-groups:SearchResources`
- `resource-groups:ListResourceTypes`

> **Note**
>
> アクセス`resource-groups:SearchResources`許可により、タグエディタはタグキーまたは値を使用して検索をフィルタリングするときにリソースを一覧表示できます。
> アクセス`resource-explorer:ListResources`許可により、検索タグを定義せずにリソースを検索するときに、タグエディタ がリソースを一覧表示できるようになります。

タグエディタ を使用するためのアクセス許可を付与する

を使用するためのポリシーを追加するには AWS Resource Groups およびタグエディタをロールにタグ付けするには、次の手順を実行します。

1. IAM コンソールを開き、ロールページを開きます。
2. タグエディタ のアクセス許可を付与するロールを見つけます。ロール名を選択して、ロールの「概要」ページを開きます。
3. 権限タブで、権限を追加するを選択します。

4. 既存のポリシーを直接添付するを選択します。
5. [Create policy] を選択します。
6. JSON タブに、次のポリシーステートメントを貼り付けます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources",
        "tag:TagResources",
        "tag:UntagResources",
        "tag:getTagKeys",
        "tag:getTagValues",
        "resource-explorer:*",
        "resource-groups:SearchResources",
        "resource-groups:ListResourceTypes"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> このポリシーステートメントの例は、 タグエディタ のアクションに対してのみを実行するアクセス許可を付与します。

7. 次へ: タグ次へ: 確認の順に選択します。
8. 新しいポリシーの名前と説明を入力します。例えば、**AWSTaggingAccess** と指定します。
9. [Create policy] を選択します。

ポリシーが に保存されたのでIAM、ロール、グループ、ユーザーなどの他のプリンシパルにアタッチできます。プリンシパルにポリシーを追加する方法の詳細については、「 ユーザーガイド」のIAM「ID アクセス許可の追加と削除IAM」を参照してください。

タグに基づく認可とアクセス制御

AWS のサービス は以下をサポートします。

- アクションに戻づくポリシー – 例えば、ユーザーに、 `GetTagKeys` もしくは `GetTagValues` のオペレーションの実行を許可し、それ以外のオペレーションを許可しないポリシーを作成できます。
- ポリシーのリソースレベルのアクセス許可 — 多くの のサービスではARNs、 を使用してポリシーで個々のリソースを指定することをサポートしています。
- タグに基づいた認可 – 多くのサービスでは、ポリシーの条件にリソースタグを使用できます。たとえば、 ユーザーに、同じタグを持つグループへのフルアクセスを許可するポリシーを作成できます。詳細については、「 とはABAC」を参照してください。 AWSの？ AWS Identity and Access Management ユーザーガイド 。
- 一時的な認証情報 – ユーザーは、タグエディタ のオペレーションを許可するポリシーが関連付けられたロールを引き受けることができます。

タグエディタ はサービスにリンクされたロールを使用しません。

タグエディタ と の統合方法の詳細については、「」を参照してください。 AWS Identity and Access Management （ IAM ）、「」の以下のトピックを参照してください。 AWS Identity and Access Management ユーザーガイド：

- AWS と連携する のサービス IAM
- タグエディタ のアクション、リソース、および条件キー
- へのアクセスの制御 AWS ポリシーを使用する リソース

タグ付けするリソースの検索

タグエディタ を使用して、タグ付けに使用できる 1 AWS リージョン つ以上の 内のリソースを検索するクエリを構築します。最大 20 の個々のリソースタイプを選択でき、また すべてのリソースタイプ に対するクエリを構築できます。クエリには、既にタグがあるリソースを含めることができ、タグがないリソースを含めることもできます。詳細については、「AWS Resource Groups ユーザーガイド」の「サポートされているリソースタイプ」の 「タグエディタ のタグ付け」列を参照してください。

タグ付けするリソースを検索した後、タグエディタを使用してタグを追加、タグを表示、編集、または削除できます。

タグ付けするリソースを検索するには

1. タグエディタ コンソールを開きます
2. （オプション）タグ付け AWS リージョン するリソースを検索する を選択します。デフォルトでは、現在のリージョンが使われています。この手順では、us-east-1 および us-west-2 を選択します。
3. リリースタイプ ドロップダウンリストから少なくとも 1 つのリソースタイプを選択します。一度に最大 20 の個々のリインスタンスインスタンスソースタイプのタグを追加または編集でき、または すべてのリソースタイプ を選択できます。この手順では、AWS::EC2::Instance および AWS::S3::Bucket を選択します。
4. 「オプショナルI」タグフィールドで、タグキーまたはタグのキーと値のペアを指定して、現在の AWS リージョン 内のリソースを指定された値でタグ付けされたもののみに制限します。タグキーを入力すると、現在のリージョンで一致するタグキーがリストに表示されます。リストからタグキーを選択できます。既存のキーと一致する十分な文字を入力すると、タグエディタがタグキーを自動補完します。タグ付けが完了したら、追加 を選択するか、Enter キーを押します。この例では、ステージ のタグキーを含むリソースをフィルタリングします。タグ値はオプションですが、クエリの結果を絞り込むことができます。さらにタグを追加するには、追加 を選択します。クエリは AND 演算子をタグに割り当てます。そのため、クエリによって、指定されたリソースタイプおよび指定されたすべてのタグと一致するリソースのみが返ります。

> **Note**
>
> タグエディタ コンソールは現在、ワイルドカードをサポートしていません。

タグキーに複数の値があるリソースを検索するには、クエリに同じキーの別のタグを追加できますが、別の値を指定します。この結果には、同じタグキーでタグ付けされたすべてのリソースと、選択した値のいずれかがあるすべてのリソースが含まれています 検索では、大文字と小文字が区別されます。

Tags (タグ) ボックスを空のままにして、選択された AWS リージョンで指定されたタイプのすべてのリソースを見つけます。このクエリは、任意のタグがあるリソースを返し、これにはタグがないリソースも含まれます。クエリからタグを削除するには、タグのラベルで X を選択します。

タグがあるが値が空のリソースを検索するには、（空の値）を選択します。

> **Note**
>
> 指定されたタグでリソースを検索する前に、現在の AWS リージョンの指定されたタイプの少なくとも 1 つのリソースに適用されている必要があります。

5. クエリの準備ができたら、リソースの検索 を選択します。結果は リソース検索の結果 領域に表として表示されます。

   大量のリソースをフィルタリングするには、リソースのフィルター) に、リソース名の一部などのフィルターテキストを入力します。

   > **Note**
   >
   > 部分文字列を使用して、結果をフィルタリングします。

6. （オプション）タグエディタ がリソースの検索結果に表示する列を設定するには、リソースの検索結果 で 設定 歯車 アイコンを選択します。

   設定 ページで、検索結果に表示する行数を選択します。表内のすべてのテキストを表示したい場合は、「行の折り返し」チェックボックスを選択します。

   タグエディタで結果に表示する列をオンにします。検索結果に含まれるそれぞれのタグの列、または検索結果のうち選択したサブセットを表示できます。これは、タグ付けするリソースを検出した後、いつでも実行できます。列を有効にするには、タグの隣にあるスイッチアイコンを選択して、オフ からオン に変更します。

表示可能な列と表示される行の数の設定が終了したら、確認 を選択します。

選択したリソースの既存のタグを表示および編集する

タグエディタでは、タグ付けするリソースを検索 クエリの結果にある、選択したリソースの既存のタグを表示します。

前のセクションで説明したように タグ列のいずれかを有効にした場合、各リソースのタグの現在の値が検索結果に表示されます。

> **Note**
>
> このトピックでは、個々のリソースのタグを編集する方法について説明します。同時に複数の選択されたリソースのタグを一括編集することもできます。詳細については、「タグエディタ によるタグの管理」を参照してください。

検索結果テーブルでタグをインラインで編集するには

1. リソースの編集するタグの値を選択します。

   > **Note**
   >
   > - 現在、選択したリソースに選択したキーのタグがない場合、値は タグ付けなし と表示されます。
   > - 選択したリソースに選択したキーのタグがあるが、値がない場合、値は「–」と表示されます。

2. 新しい値を入力するか、他のリソースに既に存在するこのタグが付いた値のいずれかを選択できます。また、タグの削除 を選択して、この 1 つのリソースからタグを削除することもできます。

個々のリソースのすべてのタグを表示するには

1. タグ付けするリソースを検索 クエリの結果で、既存のタグを表示するリソースの Tags (タグ) 列で数字を選択します。タグ 列でダッシュの付いたリソースには既存のタグがありません。

2. リソースタグ で既存のタグを表示します。「タグの管理」ページでタグを変更または削除するときに、「選択したリソースのタグを管理」を選択してこのウィンドウを開くこともできます。

> **Note**
>
> 最近リソースに加えたタグが表示されない場合は、ブラウザウィンドウを更新してください。

.csv ファイルへの結果のエクスポート

タグ付けするリソースを検索 クエリの結果をカンマ区切り値 (.csv) ファイルにエクスポートすることができます。.csv ファイルには、リソース名、サービス、リージョン、リソース IDs、タグの合計数、コレクション内の一意のタグキーごとの列が含まれます。.csv ファイルは、組織内のリソースのタグ付け戦略の決定、またはリソース間でのタグ付けに重複または不整合が存在する場所の特定に役立ちます。

1. クエリにタグ付けするリソースの検索の結果で、リソースを にエクスポートを選択します CSV。
2. ブラウザでプロンプトが表示されたら、CSV ファイルを 開くか、あるいは便利な場所に保存するかを選択します。

タグエディタ によるタグの管理

タグ付けするリソースを見つけたら、検索結果の一部またはすべてについて、タグを追加、削除、または編集できます。タグエディタ は、リソースにアタッチされているタグを表示します。また、これらのタグがタグエディタ、リソースのサービスコンソール、または を使用して追加されたかどうかも表示されますAPI。

> **Important**
>
> 個人を特定できる情報 (PII) やその他の機密情報や機密情報をタグに保存しないでください。タグを使用して、課金および管理サービスを提供します。タグは、プライベートデータや機密データに使用することを意図していません。

> **タグを管理するその他の方法**
>
> このトピックでは、 のタグエディタを使用してリソースにタグ付けする方法について説明します。 AWS Management Console。 ただし、 のタグを管理することもできます。 AWS 次のツールを使用して リソースを作成します。
>
> - シェルプロンプトでコマンドを入力またはスクリプトするには、`resourcegroupstaggingapi` AWS Command Line Interface (AWS CLI).
> - を作成して実行できます PowerShell を使用した スクリプト AWS Resource Groups API でのタグ付け AWS Tools for PowerShell Core.
> - 利用可能な のいずれかを使用してプログラムを作成して実行できます。 AWS SDKs Python のタグ付けや Java のタグ付けなど、 のタグ付けを行うリソースグループAPIsを使用するAPIs。 APIs

既存のタグを追加、削除、または編集すると、 タグ付けするリソースを見つける クエリの結果のうち選択したリソースのタグのみが変更されます。タグを管理するリソースを最大 500 個まで選択できます。

選択したリソースにタグを追加する

タグエディタを使用して、タグ付けするリソースを見つけるクエリの結果に含まれる選択したリソースにタグを追加してタグを追加できます。

> **Note**
>
> このトピックでは、複数リソースのタグを一括編集する方法について説明します。個々のリソースのタグ値を編集することもできます。詳細については、「選択したリソースの既存のタグを表示および編集する」を参照してください。

1. タグエディタコンソール を開き、タグ付けしたい複数のリソースを返すクエリを送信します。
2. タグ付けするリソースを見つける クエリの結果表で、タグを追加するリソースの横にあるチェックボックスを選択します。リソースの名前、ID 、タグキー、またはタグ値の一部をフィルリングするには、表上部にある リソースをフィルタリングする() にテキスト文字列を入力します。タグ列で、結果内のリソースに既にタグが適用されていることに注意してください。
3. 1 つ以上のリソースのチェックボックスを選択して、選択したリソースのタグの管理 () を選択します。
4. [タグの管理] ページで、選択したリソースのタグを表示します。元のクエリからより多くのリソースが返されましたが、ステップ 1 で選択したリソースにのみタグが追加されています。タグを追加 () を選択します。
5. タグキーとオプションのタグ値を入力します。この手順では、タグキー **Team** とタグ値 **Development** を追加します。

   > **Note**
   >
   > リソースには、最大 50 個のユーザー適用タグを含めることができます。ユーザーが適用したタグが 50 個に近づいている場合、リソースに新しいタグを追加できない場合があります。 AWS が生成したタグは、50 タグの制限には適用されません。タグキーも選択したリソース内で一意である必要があります。選択したリソースに既に存在するタグキーと一致するキーで新しいタグを追加することはできません。

6. タグの追加が終了したら、変更を確認して適用 を選択します。
7. 変更を受け入れる場合は、選択したすべてに変更を適用する を選択します。
8. 選択するリソースの数によっては、新しいタグを適用するのに数分かかる場合があります。同じブラウザタブでページを離れたり、別のページを開いたりしないでください。変更が成功した場合は、緑色の成功バナーがページ上部に表示されます。続行する前に、成功または失敗のバナーがページに表示されるのを待ちます。

一部またはすべてのリソースに対するタグの変更が成功しなかった場合は、「タグ変更のトラブルシューティング」を参照してください。失敗したタグの変更「アクセス権の不足など」を解決した後は、タグの変更に失敗したリソースでタグの変更を再試行できます。詳細については、「the section called “失敗したタグの変更を再試行する”」を参照してください。

選択したリソースのタグの編集

タグエディタを使用して、タグ付けするリソースを見つけるクエリ の結果に含まれる選択したリソースの既存のタグ値を変更できます。タグを編集すると、同じタグキーを持つ選択したすべてのリソースのタグの値が変更されます。タグキーの名前を変更することはできませんが、タグを削除して新しい名前のタグを作成して元のタグキーと置き換えることはできます。これにより、選択したリソースのそのキーを持つすべてのタグが削除されます。

> **Important**
>
> 個人を特定できる情報 (PII) やその他の機密情報や機密情報をタグに保存しないでください。タグを使用して、課金および管理サービスを提供します。タグは、プライベートデータや機密データに使用することを意図していません。

1. タグ付けするリソースを見つけるクエリの結果で、既存のタグを変更するリソースの横にあるチェックボックスをオンにします。リソースをフィルタリングするにテキスト文字列を入力して、リソースの名前または ID の一部をフィルタリングします。タグ列で、結果内のリソースに既にタグが適用されていることに注意してください。
2. 選択したリソースのタグの管理 を選択します。
3. タグの管理 ページの 選択したリソースのタグの編集 で、選択したリソースのタグを表示します。元のクエリはより多くのリソースを返したかもしれませんが、ステップ 1 で選択したリソースのタグのみを変更しています。
4. タグ値を変更、追加、または削除します。既存のタグにはタグキーが必要ですが、タグ値はオプションです。

   この手順では、**Team** タグの値を **QA** に変更します。

   選択したリソースが同じキーに対して異なる値を持つ場合、選択したリソースのタグ値は異なります が タグ値 フィールドに表示されます。この場合、ボックス内にカーソルを置くと、選択したリソース内のこのタグキーに使用できるすべての値のドロップダウンリストが開きます。

選択内のリソースに必要なタグ値がある場合は、入力時にそのタグ値が強調表示されます。たとえば、選択内のリソースにすでにタグ値 **QA** が付いている場合は、**Q** と入力するとその値が強調表示されます。ドロップダウンリストの値は、タグ値をリソース間で一貫性を保つのに役立ちます。タグ値は、選択したすべてのリソースで変更されます。この例では、**Team** タグキーを持つ選択したすべてのリソースのタグ値が **QA** に変更されます。**Team** タグを持たない選択されたリソースの場合、値 **QA** を持つ **Team** タグが追加されます。

5. タグの変更が完了したら、変更を確認して適用 を選択します。
6. 変更を受け入れる場合は、選択したすべてに変更を適用する を選択します。
7. 選択したリソースの数によっては、タグの編集には数分かかることがあります。同じブラウザタブでページを離れたり、別のページを開いたりしないでください。変更が成功した場合は、緑色の成功バナーがページ上部に表示されます。続行する前に、成功または失敗のバナーがページに表示されるのを待ちます。

   一部またはすべてのリソースに対するタグの変更が成功しなかった場合は、「タグ変更のトラブルシューティング」を参照してください。失敗したタグの変更 (アクセス権の不足など) の根本的な原因を解決した後は、タグの変更に失敗したリソースでタグの変更を再試行できます。詳細については、「the section called “失敗したタグの変更を再試行する”」を参照してください。

選択したリソースからタグを削除する

タグエディタを使用して、タグ付するリソースを見つける クエリの結果に含まれる選択したリソースからタグを削除できます。タグを削除すると、そのタグを持つ選択されたすべてのリソースからタグが削除されます。タグキーは編集できないため、タグキーを編集する必要がある場合は、タグを削除して新しいタグに置き換えることができます。これにより、選択したリソースのそのキーを持つすべてのタグが削除されます。

1. タグ付けするリソースを見つける クエリの結果で、タグを削除するリソースの横にあるチェックボックスをオンにします。リソースをフィルタリングする にテキスト文字列を入力して、リソースの名前または ID の一部をフィルタリングします。
2. 選択したリソースのタグの管理 を選択します。
3. タグの管理 ページの、選択したリソースのタグの管理で、選択したリソースのタグを表示します。元のクエリはより多くのリソースを返したかもしれませんが、ステップ 1 で選択したリソースのタグのみを変更しています。
4. 削除するタグの横にある タグの削除 を選択します。この手順では、**Team** タグを削除します。

> **Note**
>
> タグの削除 を選択すると、そのタグを持つ選択したすべてのリソースからタグが削除されます。

5. 変更を確認して適用 を選択します。
6. 確認ページで、選択したすべてに変更を適用 を選択します。
7. 選択したリソースの数によっては、タグの削除に数分かかることがあります。同じブラウザタブでページを離れたり、別のページを開いたりしないでください。変更が成功した場合は、緑色の成功バナーがページ上部に表示されます。続行する前に、成功または失敗のバナーがページに表示されるのを待ちます。

   一部またはすべてのリソースに対するタグの変更が成功しなかった場合は、「タグ変更のトラブルシューティング」を参照してください。失敗したタグの変更 (アクセス権の不足など) の根本的な原因を解決した後は、タグの変更に失敗したリソースでタグの変更を再試行できます。詳細については、「the section called “失敗したタグの変更を再試行する”」を参照してください。

アクセスIAM許可ポリシーでのタグの使用

AWS Identity and Access Management （ IAM ） は AWS のサービス 、 AWS リソースにアクセスできるユーザーを決定するアクセス許可ポリシーを作成および管理するために使用する です。 AWS サービスにアクセスしたり、 AWS リソースの読み取りまたは書き込みを試みるたびに、 IAMポリシーによってアクセスが制御されます。

これらのポリシーにより、リソースへのきめ細かなアクセスを提供できます。このアクセスを微調整するために使用できる機能の 1 つが、ポリシーの `Condition` 要素です。この要素を使用すると、リクエストと一致する必要がある条件を指定して、リクエストが続行できるかどうかを判断できます。`Condition` エレメントで確認できる項目には、次のものがあります。

- そのリクエストを行っているユーザーまたはロールにアタッチされているタグ。
- リクエストの目的であるリソースに添付されたタグ。

タグおよび属性ベースのアクセスコントロール

タグは、 AWS アクセスコントロール戦略の重要な部分です。属性ベースのアクセスコントロール (ABAC) 戦略で属性としてタグを使用する方法については、IAM「 ユーザーガイド」の「タグを使用した AWS リソースへのアクセスの制御」および「タグ を使用したIAMユーザーとロールへのアクセスとアクセスの制御」を参照してください。

チュートリアルでは、タグを使用してさまざまなプロジェクトやグループへのアクセスを許可する方法を示す包括的なIAMチュートリアルがあります。「 ユーザーガイド」の「タグに基づいて AWS リソースにアクセスするためのアクセス許可を定義する」。 AWS Identity and Access Management

シングルサインインに SAMLベースの ID プロバイダー (IdP) を使用する場合は、ユーザーにアクセスを提供する引き受けたロールにタグをアタッチできます。詳細については、「 ユーザーガイドIAM」の「チュートリアル: のSAMLセッションタグABACを使用するAWS Identity and Access Management 」を参照してください。

タグに関連する条件キー

次の表は、タグに基づいてアクセスを制御するアクセスIAM許可ポリシーで使用できる条件キーを示しています。これらの条件キーで以下のことが実行できます。

- オペレーションを呼び出したプリンシパルのタグを比較します。

- パラメータとしてオペレーションに与えられたタグを比較します。
- オペレーションでアクセスされるリソースにアタッチされたタグを比較します。

条件キーとその使用方法の詳細については、条件キー名列でリンクされたページを参照してください。

条件キー名	説明
aws:PrincipalTag	リクエストを行うプリンシパル (IAM ロールまたはユーザー) にアタッチされたタグと、ポリシーで指定したタグを比較します。
aws:RequestTag	リクエストにパラメータとして渡されたタグキーと値のペアと、ポリシーで指定したタグキーと値のペアを比較します。
aws:ResourceTag	ポリシーで指定したタグキーと値のペアと、リソースにアタッチされているキーと値のペアを比較します。
aws:TagKeys	リクエスト内のタグキーとポリシーで指定したキーのみを比較します。

タグを使用するIAMポリシーの例

Example 例 1: ユーザーがリソースを作成するときに特定のタグをアタッチするように強制する

次のIAMアクセス許可ポリシーの例は、IAMポリシーのタグを作成または変更するユーザーに、キーを持つタグを含めるように強制する方法を示していますOwner。またポリシーでは、タグの値を、現在呼び出し元プリンシパルにアタッチされている Owner タグと同じ値に設定する必要があります。この戦略が機能するためには、すべてのプリンシパルに Owner タグをアタッチし、ユーザーがそのタグを変更できないようにする必要があります。Owner タグを含めずにポリシーを作成または変更しようとすると、ポリシーが一致せず、その操作は許可されません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TagCustomerManagedPolicies",
            "Effect": "Allow",
            "Action": [
                "iam:CreatePolicy",
```

```
                "iam:TagPolicy"
            ],
            "Resource": "arn:aws:iam::123456789012:policy/*",
            "Condition": {
                "StringEquals": {"aws:RequestTag/Owner": "${aws:PrincipalTag/Owner}"}
            }
        }
    ]
}
```

Example 例 2 : タグを使用して、リソースへのアクセスをその「所有者」に制限する

次のIAMアクセス許可ポリシーの例では、呼び出し元のプリンシパルがEC2インスタンスと同じprojectタグ値でタグ付けされている場合にのみ、実行中の Amazon インスタンスを停止できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "ec2:StopInstances"
      ],
      "Resource": [
        "arn:aws:iam::123456789012:instance/*"
      ],
      "Condition": {
        "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag/project}"}
      }
    }
  ]
}
```

この例は、属性ベースのアクセスコントロール (ABAC) の例です。IAM ポリシーを使用してタグベースのアクセスコントロール戦略を実装する方法の詳細とその他の例については、AWS Identity and Access Management 「 ユーザーガイド」の以下のトピックを参照してください。

- タグを使用した AWS リソースへのアクセスの制御
- タグを使用したIAMユーザーとロールへのアクセスの制御

- IAM チュートリアル: タグに基づいて AWS リソースにアクセスするためのアクセス許可を定義する — 複数のタグを使用してさまざまなプロジェクトやグループへのアクセスを許可する方法を示します。

AWS Organizations タグポリシー

タグポリシーは、 で作成するポリシーの一種です。 AWS Organizations。 タグポリシーを使用すると、組織のアカウント内のリソース全体でタグを標準化できます。タグポリシーを使用するには、の「タグポリシーの開始方法」で説明されているワークフローに従うことをお勧めします。 AWS Organizations ユーザーガイド 。そのページで説明されているように、推奨されるワークフローには、非準拠のタグの検出および修正が含まれます。これらのタスクを実行するには、タグエディタコンソールを使用します。

前提条件とアクセス許可

タグエディタ でタグポリシーのコンプライアンスを評価する前に、要件を満たし、必要なアクセス許可を設定する必要があります。

トピック

- タグポリシーのコンプライアンスを評価するための前提条件
- アカウントのコンプライアンスを評価するためのアクセス許可
- 組織全体のコンプライアンスを評価するためのアクセス許可
- レポートを保存するための Amazon S3 バケットポリシー

タグポリシーのコンプライアンスを評価するための前提条件

タグポリシーのコンプライアンスを評価するには、以下のようにする必要があります。

- まず、 でこの機能を有効にする必要があります。 AWS Organizations、タグポリシーを作成してアタッチします。詳細については、 の以下のページを参照してください。 AWS Organizations ユーザーガイド：
  - タグポリシーを管理するための前提条件とアクセス許可
  - タグポリシーの有効化
  - タグポリシーの開始方法
- アカウントのリソースで非準拠のタグを検出する場合は、そのアカウントのサインイン資格情報と、アカウントのコンプライアンスを評価するためのアクセス許可 に記載されているアクセス許可が必要です。
- 組織全体のコンプライアンスを評価する場合は、組織の管理アカウントのサインイン認証情報と、組織全体のコンプライアンスを評価するためのアクセス許可 に記載されているアクセス許可

が必要です。コンプライアンスレポートは、 からのみリクエストできます。 AWS リージョン 米国東部 (バージニア北部)

アカウントのコンプライアンスを評価するためのアクセス許可

アカウントのリソースで非準拠のタグを検出するには、以下のアクセス許可が必要です。

- `organizations:DescribeEffectivePolicy` — アカウントの有効なタグポリシーの内容を取得します。
- `tag:GetResources` — アタッチされたタグポリシーに準拠していないリソースのリストを取得します。
- `tag:TagResources` - タグを追加または更新します。タグを作成するには、サービス固有のアクセス許可も必要です。例えば、Amazon Elastic Compute Cloud (Amazon EC2) のリソースにタグを付けるには、 のアクセス許可が必要です`ec2:CreateTags`。
- `tag:UnTagResources` — タグを削除します。タグを削除するには、サービス固有のアクセス許可も必要です。例えば、Amazon のリソースのタグを解除するにはEC2、 のアクセス許可が必要です`ec2:DeleteTags`。

次の例 AWS Identity and Access Management （ IAM) ポリシーは、アカウントのタグコンプライアンスを評価するためのアクセス許可を提供します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EvaluateAccountCompliance",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeEffectivePolicy",
                "tag:GetResources",
                "tag:TagResources",
                "tag:UnTagResources"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM ポリシーとアクセス許可の詳細については、IAM「 ユーザーガイド」を参照してください。

組織全体のコンプライアンスを評価するためのアクセス許可

タグポリシーへの組織全体のコンプライアンスを評価するには、以下のアクセス許可が必要です。

- organizations:DescribeEffectivePolicy — 組織、組織単位 (OU)、またはアカウントにアタッチされているタグポリシーの内容を取得します。
- tag:GetComplianceSummary — 組織内のすべてのアカウント内の非準拠リソースの概要を取得します。
- tag:StartReportCreation — 最新のコンプライアンス評価の結果をファイルにエクスポートします。組織全体のコンプライアンスは 48 時間ごとに評価されます。
- tag:DescribeReportCreation — レポート作成のステータスを確認します。
- s3:ListAllMyBuckets — 組織全体のコンプライアンスレポートへのアクセスを支援するため。
- s3:GetBucketAcl – コンプライアンスレポートを受け取る Amazon S3 バケットのアクセスコントロールリスト (ACL) を検査します。 Amazon S3
- s3:GetObject – サービス所有の Amazon S3 バケットからコンプライアンスレポートを取得します。
- s3:PutObject – 指定された Amazon S3 バケットにコンプライアンスレポートを配置します。

次のIAMポリシー例では、組織全体のコンプライアンスを評価するためのアクセス許可を提供します。各 を置き換える *placeholder* 独自の情報：

- *bucket_name* – Amazon S3 バケット名
- *organization_id* – 組織の ID

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EvaluateAccountCompliance",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeEffectivePolicy",
                "tag:StartReportCreation",
                "tag:DescribeReportCreation",
                "tag:GetComplianceSummary",
```

```
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        },
        {
            "Sid": "GetBucketAclForReportDelivery",
            "Effect": "Allow",
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::bucket_name",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
                }
            }
        },
        {
            "Sid": "GetObjectForReportDelivery",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::*/tag-policy-compliance-reports/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
                }
            }
        },
        {
            "Sid": "PutObjectForReportDelivery",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/AwsTagPolicies/organization_id/*",
            "Condition": {
                "StringEquals": {
                    "aws:CalledViaLast": "tagpolicies.tag.amazonaws.com"
                },
                "StringLike": {
                    "s3:x-amz-copy-source": "*/tag-policy-compliance-reports/*"
                }
            }
        }
    ]
}
```

IAM ポリシーとアクセス許可の詳細については、IAM「 ユーザーガイド」を参照してください。

レポートを保存するための Amazon S3 バケットポリシー

組織全体のコンプライアンスレポートを作成するには、 の呼び出しに使用する ID に、米国東部 (バージニア北部) リージョンの Amazon Simple Storage Service (Amazon S3) バケットにアクセスしてレポートを保存StartReportCreationAPIする必要があります。タグポリシーは、呼び出し元 ID の認証情報を使用して、指定されたバケットにコンプライアンスレポートを配信します。

の呼び出しに使用されているバケットと ID が同じアカウント にStartReportCreationAPI属している場合、このユースケースでは追加の Amazon S3 バケットポリシーは必要ありません。

の呼び出しに使用される ID に関連付けられたアカウントStartReportCreationAPIが Amazon S3 バケットを所有するアカウントと異なる場合、次のバケットポリシーをバケットにアタッチする必要があります。各 を置き換える *placeholder* 独自の情報 :

- *bucket_name* – Amazon S3 バケット名
- *organization_id* – 組織の ID
- *ID_ARN* – ARNの呼び出しに使用される IAM ID の 。 StartReportCreation API

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CrossAccountTagPolicyACL",
            "Effect": "Allow",
            "Principal": {
                "AWS": "identity_ARN"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Sid": "CrossAccountTagPolicyBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "AWS": "identity_ARN"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/AwsTagPolicies/organization_id/*"
```

```
        }
    ]
}
```

アカウントのコンプライアンスの評価

有効なタグポリシーを使用して、組織内のアカウントのコンプライアンスを評価できます。

> **Important**
>
> タグ付けされていないリソースは、結果で非準拠と表示されません。
> アカウント内のタグなしリソースを検索するには、 を使用するクエリ AWS Resource Explorer で を使用します**tag:none**。詳細については、「AWS Resource Explorer ユーザーガイド」の「タグ付けされていないリソースの検索」を参照してください。

有効なタグポリシーは、アカウントに適用されるタグ付けルールを指定するものです。有効なタグポリシーは、アカウントが継承する任意のタグポリシーと、アカウントに直接アタッチされたタグポリシーの集約したものです。タグポリシーを組織ルートにアタッチすると、組織内のすべてのアカウントに適用されます。タグポリシーを組織単位 (OU) にアタッチすると、OU OUsに属するすべてのアカウントと に適用されます。

> **Note**
>
> タグポリシーをまだ作成していない場合は、AWS Organizations ユーザーガイドのタグポリシーの開始方法を参照してください。

非準拠のタグを検出するには、次のアクセス許可が必要です。

- `organizations:DescribeEffectivePolicy`
- `tag:GetResources`
- `tag:TagResources`
- `tag:UntagResources`

アカウントの有効なタグポリシーへのコンプライアンスを評価するには (コンソール)

1. コンプライアンスを確認するアカウントにサインインしているときにタグポリシー を選択します。
2. 有効なタグポリシーセクションには、ポリシーが最後に更新された日時と、定義されたタグキーが表示されます。タグキーを展開すると、その値、大文字と小文字の区分、および値が特定のリソースタイプに適用されるかどうかに関する情報を表示できます。

   > **Note**
   >
   > 管理アカウントにサインインしている場合は、アカウントを選択して有効なポリシーを表示し、コンプライアンス情報を表示する必要があります。

3. 「非準拠のタグを持つリソース」セクション AWS リージョン で、非準拠のタグを検索する を指定します。必要に応じて、リソースタイプで検索することもできます。次に リソースを検索する を選択します。

   リアルタイムの結果は 検索結果セクションに表示されます。ページまたは表示する列ごとに返される結果の数を変更するには、設定アイコンを選択します。
4. 検索結果で、非準拠のタグを持つリソースを選択します。
5. リソースのタグが一覧表示されたダイアログボックスで、ハイパーリンクを選択し、リソースが作成された AWS のサービス を開きます。そのコンソールから、非準拠のタグを修正します。

   > **Tip**
   >
   > 非準拠のタグが不明な場合は、タグエディタ コンソールのアカウントの 有効なタグポリシーセクションに移動します。タグキーを展開すると、そのタグ付けルールを表示できます。

6. 必要なアカウントリソースが各リージョンで準拠するまで、タグを検出して修正するプロセスを繰り返します。

非準拠のタグを検索するには (AWS CLI、 AWS API)

以下のコマンドおよび操作を使用して、非準拠のタグを検出します。

- AWS Command Line Interface (AWS CLI):
  - aws resourcegroupstaggingapi get-resources

  - aws resourcegroupstaggingapi tag-resources
  - aws resourcegroupstaggingapi untag-resources

  でタグポリシーを使用する手順については AWS CLI、 ユーザーガイドの「 でのタグポリシー AWS CLIの使用AWS Organizations 」を参照してください。
- AWS Resource Groups Tagging API:
  - GetResources
  - TagResources
  - UntagResources

次のステップ

コンプライアンスの問題を検出して修正するプロセスを繰り返すことをお勧めします。必要なアカウントのリソースが、各リージョンの有効なタグポリシーに準拠するまで続行します。

非準拠のタグの検出と修正は、次のような複数の理由で反復的なプロセスと言えます。

- 組織のタグポリシーの使用は、時間の経過とともに進化する可能性があります。
- リソースの作成時に、組織の変更を反映させるには時間がかかります。
- コンプライアンスは、新しいリソースが作成されたとき、または新しいタグがリソースに割り当てられるときにいつでも変更できます。
- アカウントの有効なタグポリシーは、タグポリシーがアタッチされるか、アカウントからデタッチされるたびに更新されます。また、有効なタグポリシーは、アカウントが継承するポリシーにタグを付けるために変更が発生するたびに更新されます。

組織の管理アカウントとしてサインインしている場合は、レポートを生成することもできます。このレポートには、組織のアカウントにあるすべてのタグ付きリソースに関する情報が表示されます。詳細については、「組織全体のコンプライアンスを評価する」を参照してください。

組織全体のコンプライアンスを評価する

有効なタグポリシーを使用して、組織のコンプライアンスを評価できます。組織全体のアカウントにあるすべてのタグ付きリソースと、各リソースが有効なタグポリシーに準拠しているかどうかを一覧表示するレポートを生成できます。

> **Important**
>
> タグ付けされていないリソースは、結果で非準拠と表示されません。
> アカウント内のタグなしリソースを検索するには、 を使用します。 AWS Resource Explorer を使用するクエリを持つ **tag:none**。詳細については、「」の「タグなしリソースの検索」を参照してください。 AWS Resource Explorer ユーザーガイド。

の組織の管理アカウントからレポートを生成できます。 us-east-1 AWS リージョン のみ。レポートを生成するアカウントは、米国東部 (バージニア北部)リージョンの Amazon S3 バケットへのアクセス権が必要です。「Amazon S3 バケット Policy for Storing Report」に示されているように、バケットにはバケットポリシーがアタッチされている必要があります。

組織全体のコンプライアンスレポートを生成するには、次のアクセス許可が必要です。

- organizations:DescribeEffectivePolicy
- tag:GetComplianceSummary
- tag:StartReportCreation
- tag:DescribeReportCreation
- s3:ListAllMyBuckets
- s3:GetBucketAcl
- s3:GetObject
- s3:PutObject

これらのアクセス許可を表示するIAMポリシーの例については、「組織全体のコンプライアンスを評価するためのアクセス許可」を参照してください。

組織全体のコンプライアンスレポートを生成するには (コンソール)

1. タグポリシー コンソールを開きます。
2. この組織のルートタブを選択し、ページの下部近くにある レポートを生成を選択します。
3. レポートの生成画面で、レポートの保存場所を指定します。
4. エクスポートの開始を選択します。

レポートが完了したら、組織ルートタブの 非準拠レポートセクションからダウンロードすることができます。

> **メモ**
>
> 組織全体のコンプライアンスは 48 時間ごとに評価されます。この結果は以下のようになります。
>
> - タグポリシーまたはリソースに加えた変更が組織全体のコンプライアンスレポートに表示されるまで、最大で 48 時間かかる可能性があります。例えば、リソースタイプに対して新しい標準化されたタグを定義するタグポリシーがあるとします。レポートでは、このタイプでこのタグを持たないリソースが最大 48 時間にわたって準拠していると表示される可能性があります。
> - レポートはいつでも生成できますが、レポートの結果は次の評価が完了するまで更新されません。
> - NoncompliantKeys 列には、有効なタグポリシーに準拠していないリソースのタグキーが一覧表示されます。
> - KeysWithNonCompliantValues 列には、リソースにある有効なポリシーで定義されているキーが、大文字と小文字の扱いが間違っているか、非準拠の値で一覧表示されます。
> - を閉じた場合 AWS アカウント 組織のメンバーであった は、タグコンプライアンスレポートに最大 90 日間表示し続けることができます。

組織全体のコンプライアンスレポートを生成するには (AWS CLI, AWS API)

次のコマンドと操作を使用して、組織全体のコンプライアンスレポートを生成し、そのステータスを確認し、レポートを表示します。

- AWS Command Line Interface AWS CLI):
  - aws resourcegroupstaggingapi start-report-creation
  - aws resourcegroupstaggingapi describe-report-creation
  - aws resourcegroupstaggingapi get-compliance-summary

  でタグポリシーを使用するための完全な手順については、 AWS CLI、「」の「タグポリシーの使用」を参照してください。 AWS CLI ()AWS Organizations ユーザーガイド。
- AWS API:
  - StartReportCreation

- DescribeReportCreation
- GetComplianceSummary

サーバーレスワークフローと Amazon でタグの変更をモニタリングする EventBridge

Amazon EventBridge は、 AWS リソースのタグ変更をサポートしています。この EventBridge タイプを使用すると、タグの変更を照合し、イベントを 1 つ以上のターゲットにルーティングするルールを構築 EventBridgeできます。例えば、ターゲットは自動ワークフローを呼び出す AWS Lambda 関数である場合があります。このトピックでは、Lambda を使用して、 AWS リソースのタグ変更を安全に処理するための費用対効果の高いサーバーレスソリューションを構築するためのチュートリアルを提供します。

タグの変更により EventBridge イベントが生成されます

EventBridge は、 リソースの変更 AWS を記述するシステムイベントのほぼリアルタイムのストリームを提供します。多くの AWS リソースはタグをサポートしています。タグは、 AWS リソースを簡単に整理および分類するためのカスタムのユーザー定義属性です。タグの一般的な使用例としては、コスト配分の分類、アクセス制御セキュリティ、自動化などがあります。

を使用すると EventBridge、タグの変更をモニタリングし、リソースのタグの状態 AWS を追跡できます。以前は、同様の機能を実現するために、複数の呼び出しを継続的にポーリングAPIsおよびオーケストレーションしていた可能性があります。これで、個々のサービス APIs、タグエディタ、タグ付けAPIを含むタグを変更すると、リソースイベントでタグの変更が開始されます。次の例は、タグの変更によって求められる一般的な EventBridge イベントを示しています。新規、更新、削除されたタグキーと、それに関連する値が表示されます。

```
{
  "version": "0",
  "id": "bddcf1d6-0251-35a1-aab0-adc1fb47c11c",
  "detail-type": "Tag Change on Resource",
  "source": "aws.tag",
  "account": "123456789012",
  "time": "2018-09-18T20:41:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa"
  ],
  "detail": {
    "changed-tag-keys": [
      "a-new-key",
```

```
      "an-updated-key",
      "a-deleted-key"
    ],
    "tags": {
      "a-new-key": "tag-value-on-new-key-just-added",
      "an-updated-key": "tag-value-was-just-changed",
      "an-unchanged-key": "tag-value-still-the-same"
    },
    "service": "ec2",
    "resource-type": "instance",
    "version": 3,
  }
}
```

すべての EventBridge イベントには同じ最上位フィールドがあります。

- バージョン– デフォルトでは、この値はすべてのイベントで `0` (ゼロ) に設定されます。
- id – 一意の値はすべてのイベントに対して生成されます。これは、イベントがルールからターゲットに移動して処理されるとき、それらのイベントを追跡するために役立ちます。
- detail-type (詳細-タイプ)– `source` フィールドと組み合わせて、詳細フィールドに表示されるフィールドと値を識別します。
- source— イベントのソースであったサービスを識別します。タグ変更のソースは `aws.tag` です。
- time — イベントの発生時刻です。
- リージョン – イベントが発生した AWS リージョン を識別します。
- resources – このJSON配列には、イベントに関係するリソースを識別する Amazon リソースネーム (ARNs) が含まれています。これはタグが変更されたリソースです。
- detail – イベントタイプによってコンテンツが異なるJSONオブジェクト。リソースのタグ変更には、以下の詳細フィールドが含まれます。
  - changed-tag-keys – このイベントによって変更されたタグキー。
  - service — リソースが属するサービス。この例では、サービスは でec2、Amazon ですEC2。
  - Resource type – サービスのリソースタイプ。この例では、これは Amazon EC2インスタンスです。
  - version — タグセットのバージョン。バージョンは 1 から始まり、タグが変更されるとインクリメントします。このバージョンを使用して、タグ変更イベントの順序を確認できます。
  - tags — 変更後にリソースに添付されたタグ。

詳細については、「Amazon ユーザーガイド」の「Amazon EventBridge イベントパターン」を参照してください。 EventBridge

を使用すると EventBridge、さまざまなフィールドに基づいて特定のイベントパターンに一致するルールを作成できます。チュートリアルで、これを行う方法を解説します。また、指定したタグがEC2インスタンスにアタッチされていない場合に Amazon インスタンスを自動的に停止する方法も示します。 EventBridge フィールドを使用して、Lambda 関数を起動するインスタンスのタグイベントと一致するパターンを作成します。

Lambda とサーバーレス

AWS Lambda はサーバーレスパラダイムに従ってクラウドでコードを実行します。サーバーについては考えずに、必要なときだけコードを実行します。料金は、コンピューティングに使用した正確な時間に対してのみ発生します。サーバーレスと呼ばれていますが、サーバーがないという意味ではありません。このコンテキストでは、サーバーレスとは、コードの実行に使用されるサーバーをプロビジョニング、設定、管理する必要がなくなることを意味します。 AWS はこれらすべてを自動的に行うため、コードに集中できます。Lambda の詳細については、「AWS Lambda 製品概要」を参照してください。

チュートリアル: 必要なタグがない Amazon EC2インスタンスを自動的に停止する

のプールとして AWS リソースと AWS アカウント 管理している は成長し、タグを使用してリソースの分類を容易にすることができます。タグは一般的に、コスト配分やセキュリティなどの重要な用途に使用されます。を効果的に管理するには AWS リソースには、一貫してタグ付けする必要があります。多くの場合、リソースはプロビジョニングされると適切なタグがすべて付けられます。ただし、後のプロセスでタグが変更され、企業のタグポリシーから逸脱する可能性があります。タグの変更を監視することで、タグドリフトを特定してすぐに対応できます。これにより、リソースが適切に分類されているかどうかにかかっているプロセスが、望ましい結果を生み出すという確信が持てます。

次の例は、Amazon EC2インスタンスでタグの変更をモニタリングして、指定されたインスタンスに必要なタグが引き続きあることを確認する方法を示しています。インスタンスのタグが変更され、インスタンスに必要なタグがなくなった場合、Lambda 関数が呼び出されてインスタンスを自動的にシャットダウンします。なぜこれを行いたいのか これにより、すべてのリソースが企業のタグポリシーに従ってタグ付けされ、効果的なコスト配分が可能になり、属性ベースのアクセスコントロール (ABAC) に基づいてセキュリティを信頼できるようになります。

> **Important**
>
> このチュートリアルは、重要なインスタンスをうっかりシャットダウンすることがない非運用アカウントで実行することを強くお勧めします。
> このチュートリアルのサンプルコードは、このシナリオの影響をインスタンス のリストにあるインスタンスのみに意図的に制限しますIDs。テストのためにシャットダウンIDsするインスタンスでリストを更新する必要があります。これにより、 のリージョン内のすべてのインスタンスを誤ってシャットダウンすることがなくなります。 AWS アカウント.
> テスト後は、すべてのインスタンスが貴社のタグ付け戦略に従ってタグ付けされていることを確認します。次に、関数をリストIDs内のインスタンスのみに制限するコードを削除できます。

この例では、 を使用します。JavaScript および の 16.x バージョン Node.js。 この例では、 の例を使用しています。 AWS アカウント ID 123456789012 と AWS リージョン 米国東部 (バージニア北部) ()us-east-1。テストアカウント ID とリージョンを自身のものに置き換えます。

> **Note**
>
> コンソールのデフォルトに別のリージョンを使用している場合は、コンソールを変更するたびに、このチュートリアルで使用しているリージョンを必ず切り替えてください。このチュートリアルが失敗する一般的な原因は、インスタンスと関数が 2 つの異なるリージョンにあることです。

us-east-1 とは異なるリージョンを使用する場合は、以下のコード例のすべての参照コードを、選択したリージョンに変更してください。

トピック

ステップ 1. Lambda 関数を作成する

Lambda 関数を作成するには

1. を開きますAWS Lambda マネジメントコンソール 。
2. 関数の作成を選択し、一から作成を選択します。
3. 関数名 に「**AutoEC2Termination**」と入力します。
4. ランタイム で Node.js 16.x を選択します。
5. 他のすべてのフィールドはデフォルト値のままにして、関数の作成 選択します。
6. AutoEC2Termination詳細ページの「コード」タブで、index.js ファイルを開いてコードを表示します。
   - index.js のタブが開いている場合は、そのタブの編集ボックスを選択してコードを編集できます。
   - index.js を含むタブが開いていない場合は、ナビゲーションペインの自動EC2Terminatorフォルダの下にある index.js ファイルをセカンダリクリックします。次に、Open を選択します。
7. index.js タブのエディタボックスに次のコードを貼り付け、既存のコードを置き換えます。

   RegionToMonitor 値を、この関数を実行したいリージョンに置き換えます。

```
// Set the following line to specify which Region's instances you want to monitor
// Only instances in this Region are succesfully stopped on a match

const RegionToMonitor = "us-east-1"

// Specify the instance ARNs to check.
// This limits the function for safety to avoid the tutorial shutting down all
 instances in account
// The first ARN is a "dummy" that matches the test event you create in Step 3.
// Replace the second ARN with one that matches a real instance that you want to
 monitor and that you can
// safely stop

const InstanceList = [
    "i-0000000aaaaaaaaaa",
    "i-05db4466d02744f07"
];
```

```
// The tag key name and value that marks a "valid" instance. Instances in the
 previous list that
// do NOT have the following tag key and value are stopped by this function

const ValidKeyName = "valid-key";
const ValidKeyValue = "valid-value";

// Load and configure the AWS SDK
const AWS = require('aws-sdk');
// Set the AWS Region
AWS.config.update({region: RegionToMonitor});
// Create EC2 service object.
const ec2 = new AWS.EC2({apiVersion: '2016-11-15'});

exports.handler = (event, context, callback) => {

  // Retrieve the details of the reported event.
  var detail = event.detail;
  var tags = detail["tags"];
  var service = detail["service"];
  var resourceType = detail["resource-type"];
  var resource = event.resources[0];
  var resourceSplit = resource.split("/");
  var instanceId = resourceSplit[resourceSplit.length - 1];

  // If this event is not for an EC2 resource, then do nothing.
  if (!(service === "ec2")) {
    console.log("Event not for correct service -- no action (", service, ")" );
    return;
  }

  // If this event is not about an instance, then do nothing.
  if (!(resourceType === "instance")) {
    console.log("Event not for correct resource type -- no action (", resourceType,
 ")" );
    return;
  }

  // CAUTION - Removing the following 'if' statement causes the function to run
 against
  //           every EC2 instance in the specified Region in the calling AWS ####
#.
  //           If you do this and an instance is not tagged with the approved tag
 key
```

```
 //           and value, this function stops that instance.

 // If this event is not for the ARN of an instance in our include list, then do
nothing.
 if (InstanceList.indexOf(instanceId)<0) {
   console.log("Event not for one of the monitored instances -- no action (",
resource, ")");
   return;
 }

 console.log("Tags changed on monitored EC2 instance (",instanceId,")");

 // Check attached tags for expected tag key and value pair
 if ( tags.hasOwnProperty(ValidKeyName) && tags[ValidKeyName] == "valid-value"){
   // Required tags ARE present
   console.log("The instance has the required tag key and value -- no action");
   callback(null, "no action");
   return;
 }

 // Required tags NOT present
 console.log("This instance is missing the required tag key or value -- attempting
to stop the instance");

 var params = {
   InstanceIds: [instanceId],
   DryRun: true
 };

 // call EC2 to stop the selected instances
 ec2.stopInstances(params, function(err, data) {
   if (err && err.code === 'DryRunOperation') {
     // dryrun succeeded, so proceed with "real" stop operation
     params.DryRun = false;
     ec2.stopInstances(params, function(err, data) {
       if (err) {
         console.log("Failed to stop instance");
         callback(err, "fail");
       } else if (data) {
         console.log("Successfully stopped instance", data.StoppingInstances);
         callback(null, "Success");
       }
     });
   } else {
```

```
      console.log("Dryrun attempt failed");
      callback(err);
    }
  });
};
```

8. ディプロイを選択して変更を保存し、新しいバージョンの関数をアクティブにします。

この Lambda 関数は、 のタグ変更イベントによって報告された Amazon EC2インスタンスのタグをチェックします EventBridge。この例では、イベント内のインスタンスに必要なタグキー `valid-key` がない場合や、そのタグに `valid-value` 値がない場合、関数はインスタンスを停止しようとします。このロジカルチェックやタグ要件は、各自の使用事例に合わせて変更できます。

Lambda コンソールのウィンドウは開いたままにします。

ステップ 2. 必要なIAMアクセス許可を設定する

関数を正常に実行するには、EC2インスタンスを停止するアクセス許可を関数に付与する必要があります。- AWS が提供するロール lambda_basic_execution にはそのアクセス許可がありません。このチュートリアルでは、 という名前の関数の実行ロールにアタッチされているデフォルトのIAMアクセス許可ポリシーを変更しますAutoEC2Termination-role-*uniqueid*。このチュートリアルで最低限必要な追加権限は `ec2:StopInstances` です。

Amazon EC2固有のIAMポリシーの作成の詳細については、「 IAMユーザーガイド」の「Amazon EC2 EC2: インスタンスの起動または停止とセキュリティグループの変更をプログラムによりコンソールで許可する」を参照してください。

アクセスIAM許可ポリシーを作成して Lambda 関数の実行ロールにアタッチするには

1. 別のブラウザタブまたはウィンドウで、IAMコンソールのロールページを開きます。
2. ロール名 **AutoEC2Termination** の入力を開始し、リストに表示されたらそのロール名を選択します。
3. ロールの 概要ページで 権限タブを選択し、すでにアタッチされている 1 つのポリシーの名前を選択します。
4. ポリシーの概要ページで ポリシーの編集を選択します。
5. ビジュアルエディタタブで、さらにアクセス許可を追加する を選択します。
6. サービス で、 を選択しますEC2。

7. アクション で、 を選択しますStopInstances。検索バーで **Stop** と入力して、検索バーが表示されるタイミングで StopInstances を選択します。
8. リソースで すべてのリソースを選択し、レビューポリシーを選択し、最後に変更を保存を選択します。

   これにより、ポリシーの新しいバージョンが自動的に作成され、デフォルトとしてこのバージョンが設定されます。

   最終的なポリシーは次の例のようになります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:StopInstances",
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:us-east-1:123456789012:*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/
AutoEC2Termination:*"
        }
    ]
}
```

ステップ 3. Lambda 関数の予備テストを行います。

このステップでは、関数にテストイベントを送信します。Lambda テスト機能は、手動で提供したテストイベントを送信することで機能します。この関数は、テストイベントを からのイベントであるかのように処理します EventBridge。異なる値で複数のテストイベントを定義して、コードのさまざまな部分をすべて試すことができます。このステップでは、Amazon EC2インスタンスのタグが変更され、新しいタグに必要なタグキーと値が含まれていないことを示すテストイベントを送信します。

Lambda 関数をテストします。

1. Lambda コンソールでウィンドウまたはタブに戻り、自動EC2Termination関数のテストタブを開きます。
2. 新規イベントの作成 ()を選択します。
3. イベント名()で、**SampleBadTagChangeEvent** と入力します。
4. イベント JSONで、テキストを次のサンプルテキストに示すサンプルイベントに置き換えます。このテストイベントが正しく動作するためには、アカウント、リージョン、インスタンス ID を変更する必要はありません。

```
{
  "version": "0",
  "id": "bddcf1d6-0251-35a1-aab0-adc1fb47c11c",
  "detail-type": "Tag Change on Resource",
  "source": "aws.tag",
  "account": "123456789012",
  "time": "2018-09-18T20:41:38Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa"
  ],
  "detail": {
    "changed-tag-keys": [
      "valid-key"
    ],
    "tags": {
      "valid-key": "NOT-valid-value"
    },
    "service": "ec2",
    "resource-type": "instance",
    "version": 3
  }
```

```
}
```

5. Save (保存) を選択してから、テストを選択します。

   テストは失敗したようですが、問題ありません。

   レスポンス ()の 実行結果 ()タブに次のエラーが表示されるはずです。

```
{
  "errorType": "InvalidInstanceID.NotFound",
  "errorMessage": "The instance ID 'i-0000000aaaaaaaaaa' does not exist",
  ...
}
```

   このエラーは、テストイベントで指定されたインスタンスが存在しないために発生します。

   関数ログセクションの「実行結果」タブの情報は、Lambda 関数がEC2インスタンスの停止を正常に試行したことを示しています。 しかし、コードで最初にインスタンスを停止する DryRun 操作が試行され、インスタンス ID が無効であることが示されたため、失敗しました。

```
START RequestId: 390c1f8d-0d9b-4b44-b087-8de64479ab44 Version: $LATEST
2022-11-30T20:17:30.427Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    Tags
 changed on monitored EC2 instance ( i-0000000aaaaaaaaaa )
2022-11-30T20:17:30.427Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    This
 instance is missing the required tag key or value -- attempting to stop the
 instance
2022-11-30T20:17:31.206Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    INFO    Dryrun
 attempt failed
2022-11-30T20:17:31.207Z    390c1f8d-0d9b-4b44-b087-8de64479ab44    ERROR   Invoke
 Error     {"errorType":"InvalidInstanceID.NotFound","errorMessage":"The instance
 ID 'i-0000000aaaaaaaaaa' does not
 exist","code":"InvalidInstanceID.NotFound","message":"The instance ID
 'i-0000000aaaaaaaaaa' does not
 exist","time":"2022-11-30T20:17:31.205Z","requestId":"a5192c3b-142d-4cec-
bdbc-685a9b7c7abf","statusCode":400,"retryable":false,"retryDelay":36.87870631147607,"stack
["InvalidInstanceID.NotFound: The instance ID 'i-0000000aaaaaaaaaa' does
 not exist","    at Request.extractError (/var/runtime/node_modules/aws-sdk/
lib/services/ec2.js:50:35)","    at Request.callListeners (/var/runtime/
node_modules/aws-sdk/lib/sequential_executor.js:106:20)","    at Request.emit
 (/var/runtime/node_modules/aws-sdk/lib/sequential_executor.js:78:10)","    at
 Request.emit (/var/runtime/node_modules/aws-sdk/lib/request.js:686:14)","    at
 Request.transition (/var/runtime/node_modules/aws-sdk/lib/request.js:22:10)","
```

```
    at AcceptorStateMachine.runTo (/var/runtime/node_modules/aws-sdk/lib/
state_machine.js:14:12)","    at /var/runtime/node_modules/aws-sdk/lib/
state_machine.js:26:10","    at Request.<anonymous> (/var/runtime/node_modules/aws-
sdk/lib/request.js:38:9)","    at Request.<anonymous> (/var/runtime/node_modules/
aws-sdk/lib/request.js:688:12)","    at Request.callListeners (/var/runtime/
node_modules/aws-sdk/lib/sequential_executor.js:116:18)"]}
END RequestId: 390c1f8d-0d9b-4b44-b087-8de64479ab44
```

6. 正しいタグが使用されてもコードがインスタンスを停止しようとしないことを確認するには、別のテストイベントを作成して送信します。

   コードソースの上にある テスト タブを選択します。コンソールに既存のSampleBadTagChangeEventテストイベントが表示されます。
7. 新規イベントの作成 ()を選択します。
8. イベント名 に、「**SampleGoodTagChangeEvent**」と入力します。
9. 17 行目で、**NOT-** を削除して値を **valid-value** に変更します。
10. テストイベントウィンドウの上部で 保存を選択し、次に テストを選択します。

    出力には以下が表示されます。これは、関数が有効なタグを認識し、インスタンスをシャットダウンしようとしないことを示しています。

```
START RequestId: 53631a49-2b54-42fe-bf61-85b9e91e86c4 Version: $LATEST
2022-12-01T23:24:12.244Z    53631a49-2b54-42fe-bf61-85b9e91e86c4    INFO    Tags
 changed on monitored EC2 instance ( i-0000000aaaaaaaaaa )
2022-12-01T23:24:12.244Z    53631a49-2b54-42fe-bf61-85b9e91e86c4    INFO    The
 instance has the required tag key and value -- no action
END RequestId: 53631a49-2b54-42fe-bf61-85b9e91e86c4
```

    ブラウザで Lambda コンソールを開いておきます。

ステップ 4. 関数を起動する EventBridge ルールを作成する

これで、イベントに一致し、Lambda 関数をポイントする EventBridge ルールを作成できます。

EventBridge ルールを作成するには

1. 別のブラウザタブまたはウィンドウで、EventBridge コンソールを開いてルールの作成ページを開きます。
2. [名前] に「**ec2-instance-rule**」と入力し、[次へ] を選択します。

3. 作成方法までスクロールし、カスタムパターン (JSON エディタ) を選択します。
4. 編集ボックスに、次のパターンテキストを貼り付け、「次へを選択します。

```
{
  "source": [
    "aws.tag"
  ],
  "detail-type": [
    "Tag Change on Resource"
  ],
  "detail": {
    "service": [
      "ec2"
    ],
    "resource-type": [
      "instance"
    ]
  }
}
```

このルールは Amazon EC2インスタンスのTag Change on Resourceイベントに一致し、次のステップでターゲットとして指定したものをすべて呼び出します。

5. 次に、ターゲットとして Lambda 関数を追加します。ターゲット 1ボックスの ターゲットの選択で、Lambda 関数を選択します。
6. 関数 で、前に作成した AutoEC2Termination 関数を選択し、次へ を選択します。
7. ログ記録の設定ページで、次へをクリックします。確認して作成ページで、ルールの作成を選択します。また、指定した Lambda 関数 EventBridge を呼び出すアクセス許可を に自動的に付与します。

Step 5. ソリューション全体をテストしてください。

最終結果をテストするには、EC2インスタンスを作成し、タグを変更したときに何が起こるかを監視します。

モニタリングソリューションを実際のインスタンスでテストするには

1. Amazon EC2コンソールを開いてインスタンスページを開きます。

2. Amazon EC2インスタンスを作成します。起動する前に、キー `valid-key` と値 `valid-value` を含むタグをアタッチしてください。インスタンスを作成して起動する方法については、「Amazon ユーザーガイド」の「ステップ 1: インスタンスを起動する」を参照してください。EC2 「インスタンスを起動するには」手順のステップ 3 で、名前タグを入力し、その他のタグを追加 を選択し、タグを追加を選択してから、 **valid-key** の キーと **valid-value** の値を入力します。このインスタンスがこのチュートリアルのみを目的としており、完了後にこのインスタンスを削除する予定がある場合は、キーのペアなしで続行できます。ステップ 1 が終わったら、このチュートリアルに戻ってください。ステップ 2: インスタンスに接続する必要はありません。
3. コンソールInstanceIdから をコピーします。
4. Amazon EC2コンソールから Lambda コンソールに切り替えます。AutoEC2Termination 関数を選択し、Code タブを選択し、index.js タブを選択してコードを編集します。
5. Amazon EC2コンソールからコピーした値を貼り付け`InstanceList`て、 の 2 番目のエントリを変更します。`RegionToMonitor` 値が、貼り付けたインスタンスを含むリージョンと一致することを確認してください。
6. ディプロイを選択して変更を有効にします。これで、指定したリージョンのインスタンスへのタグ変更によって関数を有効化する準備が整いました。
7. Lambda コンソールから Amazon EC2コンソールに切り替えます。
8. valid-key を削除するか、そのキーの値を変更して、インスタンスにアタッチされている タグを変更します。

   > **Note**
   >
   > 実行中の Amazon EC2インスタンスのタグを変更する方法については、「Amazon ユーザーガイド」の「個々のリソースのタグの追加と削除EC2」を参照してください。

9. 数秒間待ってから、コンソールを更新します。インスタンスは 、インスタンスの状態を 停止中に変更し、次に 停止済みに変更する必要があります。
10. 関数を使用して Amazon EC2コンソールから Lambda コンソールに切り替え、モニタータブを選択します。
11. ログ タブを選択し、最近の呼び出し テーブルで、 LogStream列の最新のエントリを選択します。

    Amazon CloudWatch コンソールが開き、Lambda 関数の最後の呼び出しのログイベントページが表示されます。最後のエントリーは次のように表示されます。

```
2022-11-30T12:03:57.544-08:00    START RequestId: b5befd18-2c41-43c8-
a320-3a4b2317cdac Version: $LATEST
2022-11-30T12:03:57.548-08:00    2022-11-30T20:03:57.548Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO Tags changed on monitored EC2 instance ( arn:aws:ec2:us-
west-2:123456789012:instance/i-1234567890abcdef0 )
2022-11-30T12:03:57.548-08:00    2022-11-30T20:03:57.548Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO This instance is missing the required tag key or value --
 attempting to stop the instance
2022-11-30T12:03:58.488-08:00    2022-11-30T20:03:58.488Z b5befd18-2c41-43c8-
a320-3a4b2317cdac INFO Successfully stopped instance [ { CurrentState: { Code: 64,
 Name: 'stopping' }, InstanceId: 'i-1234567890abcdef0', PreviousState: { Code: 16,
 Name: 'running' } } ]
2022-11-30T12:03:58.546-08:00    END RequestId: b5befd18-2c41-43c8-
a320-3a4b2317cdac
```

チュートリアルの概要

このチュートリアルでは、Amazon EC2インスタンスのリソースイベントでタグの変更と照合する EventBridge ルールを作成する方法を示しました。このルールは、必要なタグがない場合にインスタンスを自動的にシャットダウンする Lambda 関数を指していました。

でのタグ変更に対する Amazon EventBridge サポート AWS リソースは、多くの でイベント駆動型のオートメーションを構築する可能性を開きます。 AWS のサービス。 この機能と の組み合わせ AWS Lambda は、 にアクセスするサーバーレスソリューションを構築するためのツールを提供します。 AWS リソースは安全、オンデマンドでスケーリングでき、費用対効果も高くなります。

tag-change-on-resource EventBridge イベントで考えられるその他のユースケースは次のとおりです。

- 誰かが通常とは異なる IP アドレスからリソースにアクセスした場合に警告を表示する — タグを使用して、リソースにアクセスする各訪問者のソース IP アドレスを保存します。タグを変更すると、 CloudWatch イベントが生成されます。このイベントを使用して、ソース IP アドレスを有効な IP アドレスのリストと比較し、ソース IP アドレスが有効でない場合は警告メールをアクティブ化できます。
- リソースのタグベースのアクセスコントロールに変更があるかどうかを監視する – 属性 (タグ) ベースのアクセスコントロール (ABAC) を使用してリソースへのアクセスを設定している場合は、タグの変更によって生成されたイベントを使用して EventBridge、セキュリティチームによる監査を促すことができます。

タグ変更のトラブルシューティング

タグ付けするリソースを見つける クエリの結果で選択したリソースにタグを適用または変更しようとしたときにエラーが発生した場合は、次のチェックリストが役立ちます。

- リソースタグの最大数がすでにある場合があります。通常、リソースには最大 50 個のユーザー定義タグを含めることができます。 AWS が生成したタグは、最大 50 タグにはカウントされません。他のユーザーも同じリソースに同時にタグを追加している可能性があります。これにより、リソースのタグが最大になる可能性があります。
- 一部のサービスでは、タグを作成するために異なる文字セットを使用できます (または許可されている文字セットを制限します)。特殊文字を使用してタグを追加または変更した場合は、リソースのサービスドキュメントでタグの要件を調べて、それらの文字がサービスで許可されていることを確認してください。
- リソースのタグを変更するためのアクセス許可がない可能性があります。リソース上の既存のタグを表示する権限がない場合は、リソースのタグを変更することはできません。
- リソースを変更するための権限がない可能性があります。リソースのメタデータに対する変更は、他の管理者によって制限されている可能性があります。
- リソースが別のユーザーまたはプロセスによって編集または削除された可能性があります。たとえば、 AWS CloudFormation スタック作成の一環としてリソースが起動されたと仮定します。スタックが削除されるか、アクティブな状態ではなくなった場合、そのリソースは使用できなくなる可能性があります。
- リソースがオフラインであるか終了している場合、またはリソースへの他の更新 (ソフトウェアのアップグレードなど) が進行中の場合は、タグを変更できない可能性があります。
- タグの変更が完了する前にブラウザタブを閉じたりページを変更したりすると、タグの変更が失敗する可能性があります。ページを離れる前に、タグの変更が終了したら、成功または失敗のバナーがページに表示されるのを待ちます。
- にはレート制限がありますが AWS Resource Groups Tagging API、タグ付けするサービスによって、Resource Groups のタグ付け制限の前にヒットする別のAPI制限が課される場合があります。

失敗したタグの変更を再試行する

選択したリソースの少なくとも 1 つでタグの変更に失敗した場合、タグエディタのページ下部に赤いバナーが表示されます。バナーには、発生した障害の種類ごとにエラーメッセージが表示されます。エラーごとに、バナーはタグエディタがタグを変更できなかった特定のリソースを識別します。

エラーを確認してトラブルシューティングを行った後、リソースで失敗したタグの変更を再試行するを選択して、タグの変更に失敗したリソースでのみ変更を再試行します。

タグエディタ のセキュリティ

AWS ではクラウドセキュリティが最優先事項です。セキュリティを最も重視する組織の要件を満たすために構築された AWS のデータセンターとネットワークアーキテクチャは、お客様に大きく貢献します。

セキュリティは、AWS と顧客の間の責任共有です。責任共有モデルでは、この責任がクラウドのセキュリティおよびクラウド内のセキュリティとして説明されています。

- クラウドのセキュリティ — AWS は、AWS クラウド で AWS のサービス を実行するインフラストラクチャを保護する責任を負います。また AWS は、お客様が使用するサービスを安全に提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。タグエディタ に適用されるコンプライアンスプログラムの詳細については、「AWS コンプライアンスプログラムによる対象範囲内のサービス」を参照してください。
- クラウド内のセキュリティ — お客様の責任は、使用する AWS のサービス に応じて異なります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、タグエディタ 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。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するようにタグエディタ を設定する方法について説明します。

トピック

タグエディタ でのデータ保護

- AWS 責任共有モデル 、タグエディタでのデータ保護に適用されます。このモデルで説明されているように、 AWS は、すべての を実行するグローバルインフラストラクチャを保護する責任があ

ります。 AWS クラウド。 お客様は、このインフラストラクチャでホストされているコンテンツの制御を維持する責任があります。また、 のセキュリティ設定と管理タスクについても責任を負います。 AWS のサービス 使用する 。データプライバシーの詳細については、「データプライバシー FAQ」を参照してください。欧州でのデータ保護の詳細については、「」を参照してください。 AWS の責任共有モデルとGDPRブログ記事 AWS セキュリティブログ 。

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

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して と通信する AWS リソースの使用料金を見積もることができます。1TLS.2 が必要で、1.3 TLS をお勧めします。
- で APIとユーザーアクティビティの ログ記録を設定する AWS CloudTrail。 CloudTrail 証跡を使用してキャプチャする方法については、「」を参照してください。 AWS アクティビティ、「」の「証 CloudTrail 跡の使用」を参照してください。 AWS CloudTrail ユーザーガイド。
- 使用アイテム AWS 暗号化ソリューションと 内のすべてのデフォルトのセキュリティコントロール AWS のサービス.
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- アクセス時に FIPS 140-3 検証済みの暗号化モジュールが必要な場合 AWS コマンドラインインターフェイスまたは を介してAPI、FIPSエンドポイントを使用します。使用可能なFIPSエンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これは、タグエディタ または他の を使用する場合も同様です。 AWS のサービス コンソール、API、 AWS CLI、または AWS SDKs。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。URL を外部サーバーに提供する場合は、そのサーバーへのリクエストを検証URLするために認証情報を に含めないことを強くお勧めします。

データ暗号化

タグ情報は暗号化されません。タグには暗号化されていませんが、セキュリティ戦略の一部として使用される情報が含まれる場合があるため、リソースのタグにアクセスできるユーザーを管理すること

が重要です。タグを変更できるユーザーを管理することは特に重要です。なぜなら、そのようなアクセスは権限の昇格に利用される可能性があるからです。

保管中の暗号化

タグエディタ 固有のサービスまたはネットワークトラフィックを分離するその他の方法はありません。該当する場合は、 を使用します。 AWS 特定の分離。仮想プライベートクラウド (VPC) でタグエディタAPIとコンソールを使用すると、プライバシーとインフラストラクチャのセキュリティを最大化できます。

転送中の暗号化

タグエディタ データは、転送中に暗号化され、サービスの内部データベースにバックアップされます。これはユーザーが設定できません。

キー管理

タグエディタ は現在 と統合されていません AWS Key Management Service および は をサポートしていません AWS KMS keys.

インターネットトラフィックのプライバシー

タグエディタ は、タグエディタ ユーザーと 間のすべての送信HTTPSに を使用します。 AWS。 タグエディタ はトランスポートレイヤーセキュリティ (TLS) 1.3 を使用しますが、1.2 TLS もサポートしています。

タグエディタ の Identity and Access Management

AWS Identity and Access Management （ IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰にタグエディタリソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は追加料金なしで AWS のサービス 使用できる です。

トピック

- タグエディタ アイデンティティベースポリシーの例
- タグエディタ アイデンティティとアクセスのトラブルシューティング

対象者

AWS Identity and Access Management （ IAM) の使用方法は、タグエディタで行う作業によって異なります。

サービスユーザー – ジョブを実行するために タグエディタ サービスを使用する場合は、管理者から必要なアクセス許可と認証情報が与えられます。作業を実行するためにさらに多くの タグエディタ の機能を使用するとき、追加の許可が必要になる場合があります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。タグエディタ の機能にアクセスできない場合は、「タグエディタ アイデンティティとアクセスのトラブルシューティング」を参照してください。

サービス管理者 – 社内の タグエディタ リソースを担当している場合は、通常、タグエディタ へのフルアクセスがあります。サービスのユーザーがどの タグエディタ 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、サービスユーザーのアクセス許可を変更するリクエストをIAM管理者に送信する必要があります。このページの情報を確認して、 の基本概念を理解しますIAM。タグエディタIAMで を使用する方法の詳細については、「」を参照してくださいタグエディタと の連携方法 IAM。

IAM 管理者 – IAM管理者の場合は、タグエディタへのアクセスを管理するポリシーの作成方法の詳細を知りたい場合があります。で使用できるタグエディタのアイデンティティベースのポリシーの例を表示するにはIAM、「」を参照してくださいタグエディタ アイデンティティベースポリシーの例。

アイデンティティを使用した認証

認証は、アイデンティティ認証情報 AWS を使用して にサインインする方法です。として、IAMユーザーとして AWS アカウントのルートユーザー、またはIAMロールを引き受けることで、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook 認証情報は、フェデレーティッド ID の例です。フェデレーティッド ID としてサインインすると、管理者は以前にIAMロールを使用して ID フェデレーションをセットアップしていました。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

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

AWS プログラムで にアクセスする場合、 はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号化して署名します。 AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM「 ユーザーガイド」の「リクエストの署名 AWS API」を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 は、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、AWS IAM Identity Center 「 ユーザーガイド」の「多要素認証の使用」および「 ユーザーガイド」の「多要素認証の使用 (MFA) AWS」を参照してください。 IAM

AWS アカウント ルートユーザー

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

ユーザーとグループ

IAM ユーザーとは、1 人のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を持つIAMユーザーを作成する代わりに、一時的な認証情報に依存することをお勧めします。ただし、IAMユーザーとの長期的な認証情報を必要とする特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、IAM「 ユーザーガイド」の「長期的な認証情報を必要とするユースケースのアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAMユーザーのコレクションを指定する ID です。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、

という名前のグループがありIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

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

ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内の ID です。ユーザーと似ていますがIAM、特定の人物には関連付けられていません。IAM ロール を切り替える AWS Management Console ことで、 でロールを一時的に引き受けることができます。または AWS API オペレーションを AWS CLI 呼び出すか、カスタム を使用してロールを引き受けることができますURL。ロールを使用する方法の詳細については、 IAM ユーザーガイドの「ロールを引き受ける方法」を参照してください。

IAM 一時的な認証情報を持つ ロールは、以下の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、 IAM ユーザーガイドの「サードパーティー ID プロバイダーのロールの作成」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証された後にアクセスできる内容を制御するために、IAMIdentity Center はアクセス許可セットを のロールに関連付けますIAM。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。
- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、特定のタスクに対して異なるアクセス許可を一時的に引き受けるIAMロールを引き受けることができます。
- クロスアカウントアクセス — IAMロールを使用して、別のアカウントの誰か (信頼できるプリンシパル) が自分のアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、IAM「 ユーザーガイド」の「 のクロスアカウントリソースアクセスIAM」を参照してください。
- クロスサービスアクセス — 他の の機能 AWS のサービス を使用するものもあります AWS のサービス。例えば、サービスで呼び出しを行うと、そのサービスが Amazon でアプリケーションを実

行EC2したりAmazon S3にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。

- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストを使用します。FAS リクエストは、サービスが他の AWS のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「アクセスセッションの転送」を参照してください。
- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受けるIAMロールです。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、IAM「 ユーザーガイド」の「 にアクセス許可を委任するロールの作成 AWS のサービス」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示することはできますが、編集することはできません。

- Amazon で実行されているアプリケーション EC2 – IAMロールを使用して、EC2インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2インスタンス内にアクセスキーを保存するよりも望ましいです。EC2インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルには ロールが含まれており、EC2インスタンスで実行されているプログラムが一時的な認証情報を取得できるようにします。詳細については、IAM「 ユーザーガイド」のIAM「ロールを使用して Amazon EC2インスタンスで実行されているアプリケーションにアクセス許可を付与する」を参照してください。

IAM ロールとIAMユーザーのどちらを使用するかについては、 IAM ユーザーガイドの (ユーザーではなく) IAMロールを作成するタイミングを参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成して AWS ID またはリソースにアタッチします。ポリシーは AWS 、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する オブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーはJSONドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「 ユーザーガイド」のJSON「ポリシーの概要」を参照してください。 IAM

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

デフォルトでは、ユーザーやロールに権限はありません。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するには、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行に使用する方法に関係なく、アクションのアクセス許可を定義します。例えば、`iam:GetRole`アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS からロール情報を取得できますAPI。

アイデンティティベースのポリシー

ID ベースのポリシーは、IAMユーザー、ユーザーのグループ、ロールなどの ID にアタッチできるJSONアクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。ID ベースのポリシーを作成する方法については、「 ユーザーガイド」のIAM「ポリシーの作成」を参照してください。 IAM

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

リソースベースのポリシー

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

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーIAMでは、 から AWS 管理ポリシーを使用することはできません。

アクセスコントロールリスト (ACLs)

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

Amazon S3、および Amazon VPCは AWS WAF、 をサポートするサービスの例ですACLs。の詳細についてはACLs、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) 概要」を参照してください。

その他のポリシータイプ

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

- アクセス許可の境界 – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できる最大アクセス許可を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM「 ユーザーガイド」のIAM「エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定するJSONポリシーです AWS Organizations。 AWS Organizations は、ビジネスが

所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) をアカウントのいずれかまたはすべてに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可SCPを制限します AWS アカウントのルートユーザー。Organizations と の詳細については SCPs、AWS Organizations 「 ユーザーガイド」の「サービスコントロールポリシー」を参照してください。

- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「 ユーザーガイド」の「セッションポリシー」を参照してください。 IAM

複数のポリシータイプ

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

タグエディタ と の連携方法 IAM

IAM を使用してタグエディタへのアクセスを管理する前に、タグエディタで使用できるIAM機能を理解しておく必要があります。タグエディタおよびその他の が と AWS のサービス 連携する方法の概要を把握するにはIAM、 IAM ユーザーガイドの「 AWS のサービス と連携する IAM 」を参照してください。

トピック

- タグエディタ のアイデンティティベースのポリシー
- リソースベースのポリシー
- タグに基づく認可
- タグエディタ IAMロール

タグエディタ のアイデンティティベースのポリシー

IAM アイデンティティベースのポリシーでは、アクションが許可または拒否される条件に加えて、許可または拒否されるアクションとリソースを指定できます。タグエディタ は、特定のアクション、リソース、および条件キーをサポートしています。JSON ポリシーで使用するすべての要素については、「 ユーザーガイド」の「 IAMJSONポリシー要素のリファレンスIAM」を参照してください。

アクション

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

JSON ポリシーの Action要素は、ポリシーでアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションの名前は通常、関連する AWS APIオペレーションと同じです。一致するAPIオペレーションがないアクセス許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

タグエディタ のポリシーアクションは、アクションの前にプレフィックスを使用します: tag:。タグエディタのアクションはコンソールで完全に実行されますが、ログエントリにプレフィックス tag が付けられます。

例えば、リソースに tag:TagResourcesAPIオペレーションでタグ付けするアクセス許可を付与するには、ポリシーに tag:TagResourcesアクションを含めます。ポリシーステートメントには、Action または NotAction 要素を含める必要があります。タグエディタ は、このサービスで実行できるタスクを記述する独自のアクションのセットを定義します。

単一のステートメントに複数のタグ付けアクションを指定するには、次のようにコンマで区切ります。

```
"Action": [
      "tag:action1",
      "tag:action2",
      "tag:action3"
```

ワイルドカード *を使用して複数のアクションを指定することができます。例えば、Get という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "tag:Get*"
```

タグエディタ のアクションのリストについては、「サービス認証リファレンス」の「タグエディタのアクション、リソース、および条件キー」を参照してください。

リソース

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

Policy ResourceJSON要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、Resource または NotResource 要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

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

タグエディタ には独自のリソースはありません。代わりに、他の AWS のサービスが作成したリソースにアタッチされたメタデータ (タグ) を操作します。

条件キー

管理者はポリシーを使用して AWS JSON、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルが、どのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条

件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば、リソースにIAMユーザー名でタグ付けされている場合にのみ、リソースへのアクセス許可をIAMユーザーに付与できます。詳細については、「 ユーザーガイド」のIAM「ポリシー要素: 変数とタグIAM」を参照してください。

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

タグエディタ は、サービス固有の条件キーを定義しません。

例

タグエディタ のアイデンティティベースのポリシーの例を表示するには、「タグエディタ アイデンティティベースポリシーの例」を参照してください。

リソースベースのポリシー

タグエディタ は独自のリソースを定義しないため、リソースベースのポリシーはサポートされていません。

タグに基づく認可

タグに基づいた認可は、属性ベースのアクセスコントロール () と呼ばれるセキュリティ戦略の一部ですABAC。

タグに基づいてリソースへのアクセスを管理するには、`aws:ResourceTag/`*`key-name`*、`aws:RequestTag/`*`key-name`*、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素 でタグ情報を提供します。リソースを作成または更新するときに、リソースにタグを適用することができます。

リソースのタグに基づいてリソースへのアクセスを制限するためのアイデンティティベースポリシーの例を表示するには、「タグに基づいたグループの表示」を参照してください。属性ベースのアクセスコントロール (ABAC) の詳細については、「 IAMユーザーガイド」の「 とは AWS ABAC」を参照してください。

タグエディタ IAMロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内のエンティティです。タグエディタにはサービスロールがないか、または使用しません。

タグエディタ での一時的な認証情報の使用

タグエディタ では、一時的な認証情報を使用して、フェデレーションでサインインしたり、 IAM ロールを引き受けたり、クロスアカウントロールを引き受けたりすることができます。一時的なセキュリティ認証情報を取得するには、 AssumeRoleや などのオペレーションを呼び出し AWS STS APIますGetFederationToken。

サービスリンクロール

サービスにリンクされたロールを使用すると AWS のサービス 、 は他の サービスのリソースにアクセスして、ユーザーに代わってアクションを実行できます。

タグエディタ にはサービスにリンクされたロールがないか、または使用しません。

サービスロール

この機能により、ユーザーに代わってサービスがサービスロールを引き受けることが許可されます。

タグエディタ にはサービスロールがないか、または使用しません。

タグエディタ アイデンティティベースポリシーの例

デフォルトでは、ロールやユーザーなどの IAM プリンシパルには、タグを作成または変更するアクセス許可はありません。AWS Management Console や AWS Command Line Interface (AWS CLI)又はAWS API を使用してタスクを実行することもできません。IAM 管理者は、プリンシパルに必要な、指定されたリソースで特定の API オペレーションを実行するアクセス許可をプリンシパルに付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらのアクセス許可が必要なプリンシパルに、そのポリシーをアタッチしなければなりません。

これらの JSON ポリシードキュメント例を使用して IAM のアイデンティティベースポリシーを作成する手順については、「IAM ユーザーガイド」の「JSON タブでのポリシーの作成」を参照してください。

トピック

- ポリシーのベストプラクティス
- タグエディタ コンソールと リソースグループのタグ付け API を使用する
- ユーザーが自分のアクセス許可を表示できるようにする方法
- タグに基づいたグループの表示

ポリシーのベストプラクティス

アイデンティティベースポリシーは、ユーザーのアカウントで誰かが タグエディタ リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、AWS アカウント に料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

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

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

タグエディタ コンソールと リソースグループのタグ付け API を使用する

タグエディタ コンソールおよび リソースグループのタグ付け API にアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、AWS アカウント のリソースにアタッチされたタグの詳細をリスト化および表示できます。最小限必要な許可よりも制限されたアイデンティティベースのポリシーを作成すると、そのポリシーを持つ IAM プリンシパルに対しては、コンソールおよび API コマンドが意図したとおりに機能しません。

これらのプリンシパルがまだ タグエディタ を使用できるように、エンティティに次のポリシー (または次のポリシーに記載されているアクセス許可を含むポリシー) をアタッチします。詳細については、IAM ユーザーガイド」の「ユーザーへの許可の追加」を参照してください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources",
        "tag:TagResources",
        "tag:UntagResources",
        "tag:getTagKeys",
        "tag:getTagValues",
        "resource-explorer:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

タグエディタ および リソースグループのタグ付け API へのアクセス権限を付与する方法については、 タグエディタ を使用するためのアクセス許可を付与する を参照してください。

ユーザーが自分のアクセス許可を表示できるようにする方法

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI か AWS API を使用してプログラム的に、このアクションを完了する権限が含まれています。

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

タグに基づいたグループの表示

アイデンティティベースのポリシーの条件を使用して、タグに基づいて タグエディタ リソースへのアクセスをコントロールできます。この例では、リソースを表示できるポリシーを作成する方法、この場合はリソースグループについて表示します 。ただし、アクセス許可が付与されるのは、project グループタグが、呼び出し元のプリンシパルに付けられた project タグと同じ値を持つ場合のみです。

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

```
        {
            "Effect": "Allow",
            "Action": "resource-groups:ListGroups",
            "Resource": "arn:aws:resource-groups::region:account_ID:group/group_name"
        },
        {
            "Effect": "Allow",
            "Action": "resource-groups:ListGroups",
            "Resource": "arn:aws:resource-groups::region:account_ID:group/group_name",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/project": "${aws:PrincipalTag/
project}"}
            }
        }
    ]
}
```

このポリシーをアカウントの ユーザーにアタッチできます。projectalphaタグキーとタグ値を持つユーザーがリソースグループを表示しようとした場合、そのグループにもタグを付ける必要がありますproject=alpha。それ以外の場合、ユーザーはアクセスを拒否されます。条件キー名では大文字と小文字が区別されないため、条件タグキー project は Project と project の両方に一致します。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素：条件) を参照してください。

タグエディタ アイデンティティとアクセスのトラブルシューティング

次の情報は、タグエディタ と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- タグエディタ でアクションを実行する権限がない
- iam を実行する権限がありません。PassRole

タグエディタ でアクションを実行する権限がない

AWS Management Console から、アクションを実行する権限がないと通知された場合は、管理者に問い合わせてサポートを依頼する必要があります。管理者とは、サインイン認証情報を提供した担当者です。

以下の例のエラーは、mateojackson ユーザーがコンソールを使用して、 リソースのタグを表示しようとしているが、 tag:GetTagKeys のアクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 tag:GetTagKeys on resource: arn:aws:resource-groups::us-west-2:123456789012:resource-
type/my-test-resource
```

この場合、Mateo は、tag:GetTagKeys アクションを使用して my-test-resource リソースへのアクセスが許可されるように、管理者にポリシーの更新を依頼します。

iam を実行する権限がありません。PassRole

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して タグエディタ にロールを渡すことができるようにする必要があります。

一部の AWS のサービスでは、新しいサービスロールやサービスリンクロールを作成せずに、既存のロールをサービスに渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して タグエディタ でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。Mary には、ロールをサービスに渡す権限がありません。

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

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

サポートが必要な場合は、AWS 管理者に問い合わせてください。サインイン資格情報を提供した担当者が管理者です。

タグエディタ でのログ記録とモニタリング

タグエディタ でのすべてのアクションは AWS CloudTrail にログ記録されます。

でのタグエディタ API コールのログ記録 CloudTrail

タグエディタ はAWS CloudTrail、ユーザー、ロール、またはAWS のサービスタグエディタの によって実行されたアクションを記録するサービスである と統合されています。 は、タグエディタ

コンソールからの呼び出しや Resource Groups Tagging API へのコード呼び出しを含む、タグエディタのすべての API コールをイベントとして CloudTrail キャプチャします。証跡を作成する場合は、タグエディタの CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。証跡を設定しない場合でも、コンソールのイベント履歴 で最新の CloudTrail イベントを表示できます。 で収集された情報を使用して CloudTrail、タグエディタに対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

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

のタグエディタ情報 CloudTrail

CloudTrail アカウントを作成するAWS アカウントと、 は で有効になります。タグエディタ またはタグエディタコンソールでアクティビティが発生すると、そのアクティビティはイベント履歴 CloudTrailの他のAWS の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 を使用した CloudTrail イベントの表示」を参照してください。

タグエディタ のイベントなどの AWS アカウント のイベントを継続的に記録するには、証跡を作成します。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、それに基づく対応を行うAWS のサービスように他の を設定できます。詳細については、以下のリソースを参照してください。

- AWS アカウント の追跡の作成
- CloudTrail でサポートされるサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

すべてのタグエディタ アクションは によってログに記録 CloudTrail され、「タグエディタ API リファレンス」に記載されています。コンソールのタグエディタアクションは によってログに記録され CloudTrail、 `tagging.amazonaws.com`として を持つイベントとして表示されます`eventSource`。

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。同一性情報は次の判断に役立ちます。

- リクエストが、ルートと 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ションユーザーの一時的なセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービス によって送信されたかどうか。

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

タグエディタ のログファイルエントリの概要

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。 CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、アクション を示す CloudTrail ログエントリを示していますTagResources。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661372702",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661372702",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-24T20:25:03Z",
```

```
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-08-24T20:27:14Z",
    "eventSource": "tagging.amazonaws.com",
    "eventName": "TagResources",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.65",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resourcegroupstaggingapi.tag-resources",
    "requestParameters": {
        "resourceARNList": [
            "arn:aws:events:us-east-1:123456789012:rule/SecretsManagerMonitorRule"
        ],
        "tags": {
            "owner": "alice"
        }
    },
    "responseElements": {
        "failedResourcesMap": {}
    },
    "requestID": "8f9ea891-4125-460c-802f-26c11EXAMPLE",
    "eventID": "b2c9322a-aad7-424b-8f0b-423daEXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "tagging.us-east-1.amazonaws.com"
    }
}
```


タグエディタ のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスHIPAAのセキュリティとコンプライアンスのためのアーキテクチャ – このホワイトペーパーでは、企業が AWS を使用して HIPAA対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべての AWS のサービス がHIPAA対象となるわけではありません。詳細については、HIPAA「対象サービスリファレンス」を参照してください。

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド — コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST) 、Payment Card Industry Security Standards Council ()、PCI国際標準化機構 (ISO) など) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、Security Hub のコントロールリファレンスを参照してください。
- Amazon GuardDuty — これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。 GuardDuty は、特定のコンプライアンスフレームワーク

で義務付けられている侵入検知要件を満たすことでDSS、 PCI などのさまざまなコンプライアンス要件への対応に役立ちます。

- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

タグエディタ における耐障害性

タグエディタ は、内部サービスリソースへの自動バックアップを実行します。これらのバックアップはユーザーが設定できません。バックアップは、保管時と転送中のいずれも暗号化されます。タグエディタ は Amazon DynamoDB に顧客データを保存します。

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。

タグを誤って削除した場合は、AWS Support センターにお問い合わせください。

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。

タグエディタ でのインフラストラクチャセキュリティ

タグエディタ には、サービスまたはネットワークトラフィックを分離するその他の方法はありません。該当する場合は、AWS 固有の分離を使用してください。仮想プライベートクラウド (VPC) でタグエディタ API とコンソールを使用することで、プライバシーとインフラストラクチャのセキュリティを最大限に高めることができます。

タグエディタ には、AWS が公開した API コールを使用してネットワーク経由でアクセスします。クライアントは以下をサポートする必要があります:

- トランスポート層セキュリティ (TLS) TSL 1.2 および TSL 1.3 をお勧めします。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストは、アクセスキー ID と、AWS Identity and Access Management (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。

タグエディタ では、リソースベースのポリシーをサポートしません。

タグエディタ API オペレーションは任意のネットワークの場所から呼び出すことができますが、 タグエディタ ではリソースベースのアクセスポリシーがサポートされているため、ソース IP アドレスに基づく制限を含めることができます。また、 タグエディタ ポリシーを使用して、特定の Amazon Virtual Private Cloud (Amazon VPC) エンドポイントまたは特定の VPC からのアクセスを制御することもできます。このアプローチにより、実質的に AWS ネットワーク内の特定の VPC からの特定のリソースへのネットワークアクセスが分離されます。

Service Quotas

次の表に、タグエディタ のService Quotasに関する情報を示します。

現在、これらのクォータは Service Quotasコンソールでは調整できません。AWS Support に問い合わせる。

名前	デフォルト
リソースごとに添付されたタグ	50 個のユーザー定義タグ (AWS 生成されたタグはこの制限にはカウントされません)。
タグキー名	UTF-8 の最小 1、最大 128 文字の Unicode 文字。 使用可能な文字は、文字、数字、スペース、および以下の文字です。 `_ . : / = + - @` キー名は で始めることはできません。このプレフィックスは AWS 使用のために予約aws:されているためです。 **Note** 一部の AWS のサービス には、追加の文字または長さの制限があります。詳細については、特定のサービスのドキュメントを参照してください。

名前	デフォルト	
タグ値	最小値は 0、UTF-8 では最大 256 文字の Unicode 文字。 使用可能な文字は、文字、数字、スペース、および以下の文字です。 `_ . : / = + - @` **Note** 一部の AWS のサービス には、追加の文字または長さの制限があります。詳細については、特定のサービスのドキュメントを参照してください。	
GetResources API オペレーションの呼び出しレート	1 秒あたりの 15 コールの最大数	
次のAPIオペレーションを呼び出すレート： • TagResources • UntagResources • GetTagKeys • GetTagValues	1 秒あたりの 5 コールの最大数	

タグエディタ のドキュメント履歴

変更	説明	日付
組織全体のコンプライアンスを評価するためのアクセス許可を更新	組織全体のコンプライアンスを評価するためのアクセス許可を更新し、コンプライアンスレポートへのアクセスを支援するアクセス許可を含めました。	2024 年 8 月 28 日
更新された内容	読みやすさと検出可能性を向上させるために、トピックタイトルを更新し、コンテンツを再編成しました。	2024 年 7 月 25 日
からのコンテンツのタグ付け AWS 全般のリファレンス このガイドに移動しました	のタグ付けに関するトピック AWS リソースが から移動されました AWS 全般のリファレンス このガイドの「」を参照してください。	2023 年 3 月 24 日
IAM ベストプラクティスの更新	IAM ベストプラクティスに合わせてガイドを更新しました。詳細については、「 のセキュリティのベストプラクティスIAM」を参照してください。	2023 年 1 月 3 日
タグエディタ のドキュメントを独自のガイドに移動する	タグエディタ のドキュメントは、 の一部ではなく、独自のユーザーガイドで提供されるようになりました。 AWS Resource Groups ユーザーガイド。	2022 年 12 月 13 日

タグポリシーへの準拠を確認	を使用してタグポリシーを作成してアカウントにアタッチした後 AWS Organizationsでは、組織のアカウントのリソースに非準拠のタグがあります。	2019 年 11 月 26 日
タグエディタでタグ付けされていないリソースの検索がサポート	タグエディタでは、特定のタグキーに適用されるタグ値を持たないリソースを検索することができるようになりました。	2019 年 6 月 18 日
タグエディタ コンソールがから移動する AWS Systems Manager コンソール	タグエディタ コンソールは、システム・マネージャコンソールから独立しました。Systems Manager の左側のナビゲーションバーでタグエディタコンソールへのポインタは引き続き確認できますが、タグエディタコンソールは、 の左上にあるドロップダウンメニューから直接開くことができます。 AWS Management Console.	2019 年 6 月 5 日
古い、従来の タグエディタ のツールは利用できなくなりました	古いタグエディタ、クラシックタグエディタ、またはレガシータグエディタに関する言及は削除されました。これらのツールは では使用できなくなりました。 AWS。 代わりにタグエディタを使用します。	2019 年 5 月 14 日

タグエディタでは、複数のリージョン間でリソースへのタグ付けがサポートされるようになりました	タグエディタで、複数のリージョンにまたがるリソースのタグを検索および管理することができ、現在のリージョンがデフォルトでリソースクエリに追加されます。	2019 年 5 月 2 日
タグエディタ で、クエリ結果を にエクスポートできるようになりました。 CSV	クエリの結果は、「タグ付けするリソースの検索」ページで CSV形式のファイルにエクスポートできます。新しいリージョン列はタグエディタのクエリ結果に表示されます。タグエディタで、特定のタグキーに対して空白でない値を持つリソースを検索することができるようになりました。既存のキー間にある固有の値を入力すると、タグキーの値が自動入力されます。	2019 年 4 月 2 日

タグエディタで、クエリへのすべてのリソースタイプの追加がサポートされるようになりました	1 回のオペレーションで最大 20 の個々のリソースタイプにタグを適用することができ、すべてのリソースタイプ を選択して、リージョンのすべてのリソースタイプにクエリを実行することもできます。リソース間でタグキーを一貫して有効にするために役立つ、自動補完がクエリの タグのキー フィールドに追加されました。一部のリソースでタグの変更が失敗した場合、タグの変更に失敗したリソースのみでタグの変更を再試行できます。	2019 年 3 月 19 日
タグエディタで、複数のリソースタイプが検索でサポートされるようになりました	1 回のオペレーションで最大 20 のリソースタイプにタグを適用することができます。検索結果に表示された列を選択することもでき、これには検索結果で検出された固有の各タグキーの列または結果から選択されたリソースも含まれます。	2019 年 2 月 26 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。