ユーザーガイド

AWS リソースのタグ付け

Version 1.0

AWS リソースのタグ付け: ユーザーガイド

Table of Contents

AWS リソースのタグ付け 1
タグを追加する方法 1
ベストプラクティス 2
カテゴリのタグ付け 3
タグの命名制限と要件 4
一般的なタグ付け戦略 4
リソース整理のタグ 5
コスト配分のタグ 5
オートメーションのタグ 6
アクセス制御のタグ 6
タグ付けのガバナンス 6
詳細はこちら 7
Using Tag Editor 8
タグおよび属性ベースのアクセスコントロール 9
タグの名前に関するベストプラクティス 9
開始 11
前提条件 11
タグ付けするリソースの検索 18
選択したリソースのタグを表示および編集する 21
.csv ファイルへの結果のエクスポート 22
関連情報 23
タグの管理 23
選択したリソースにタグを追加する 24
選択したリソースのタグの編集 27
選択したリソースからタグを削除する 30
失敗したタグの変更を再試行する 32
関連情報 32
IAMポリシーでタグを使用する 33
タグに関連する条件キー 33
タグを使用する IAM ポリシーの例 34
AWS Organizations タグポリシー 35
前提条件とアクセス許可 36
アカウントのコンプライアンスの評価 40
組織全体のコンプライアンスを評価する 42

タグ変更の監視 45
タグの変更により EventBridge イベントが生成されます 45
Lambda とサーバーレス 47
チュートリアル：必須タグがない Amazon EC2 インスタンスの自動停止 47
タグ変更のトラブルシューティング 60
関連情報 60
セキュリティ 61
データ保護 61
データ暗号化 62
インターネットトラフィックのプライバシー 63
アイデンティティおよびアクセス管理 63
対象者 64
アイデンティティによる認証 64
ポリシーを使用したアクセス権の管理 68
IAM で タグエディタ を使用する方法 70
アイデンティティベースポリシーの例 74
トラブルシューティング 78
ロギングとモニタリング 79
CloudTrail 統合 79
コンプライアンス検証 82
耐障害性 83
インフラストラクチャセキュリティ 84
リファレンス 86
タグエディタのService Quotas 86
ドキュメント履歴 88
AWS 用語集 92
..... xciii

AWS リソースのタグ付け

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、AWS Secrets Manager のシークレットなどがあります。

> **Important**
>
> 個人情報 (PII) などの機密情報や秘匿性の高い情報はタグに格納しないでください。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。

タグを使用し、リソースを目的、所有者、環境などの基準別に分類できます。

AWS リソースにタグを追加する方法

AWS リソースにタグを追加する方法は 3 つあります。

- 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、AWS のサービス を直接サポートしていました。各 AWS のサービス が提供するタグ機能については、「AWSドキュメンテーションインデックス」にある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 — 一部のサービスはAWS タグエディタ コンソールによるタグ付けもサポートしています。
- リソースグループのタグ付け API — ほとんどのサービスは、AWS Resource Groups Tagging API を使用したタグ付けもサポートしています。

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。以下のサービスについては、AWS は、お客様のユースケースのタグ付けにより適した新しい代わりの AWS のサービス をお勧めします。

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	

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、次のベストプラクティスに従ってください。

- 個人情報 (PII) などの機密情報や秘匿性の高い情報はタグに追加しないようにします。タグは、多くの AWS のサービス (請求など) 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 リソースグループのタグ付け API を使用すると、プログラムによるタグの制御が可能になるため、タグとリソースの自動的な管理、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。

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	• 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、さらに精密な調査が必要なルートテーブルまたはセキュリティグループを特定する	• 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

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
• 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			

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。
- UTF-8 では、タグキーは 1 文字以上で、最大 128 文字の Unicode 文字である必要があります。
- UTF-8 では、タグ値は 0 文字以上、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、AWS のサービスごとに異なります。AWS の特定のサービスでリソースのタグ付け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。通常、使用できる文字は、UTF-8 対応の文字、数字、スペースと、 _ . : / = + - @ の文字です。
- 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。

コンテンツ

- リソース整理のタグ
- コスト配分のタグ
- オートメーションのタグ
- アクセス制御のタグ

リソース整理のタグ

タグは、AWSで AWS Management Console リソースを整理するための効果的な手段です。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 つまたは複数のタグ、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 スタック内での出現回数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 と請求明細レポートを使用すると、AWS の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、お客様、またはプロジェクトといったビジネスタグを使用して、AWS の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。次に、コスト配分レポートの例を示します。

Total Cost	user:Owner	user:Stack	user:Cost Center	user:Application
0.95	DbAdmin	Test	80432	Widget2
0.01	DbAdmin	Test	80432	Widget2
3.84	DbAdmin	Prod	80432	Widget2
6.00	DbAdmin	Test	78925	Widget1
234.63	SysEng	Prod	78925	Widget1
0.73	DbAdmin	Test	78925	Widget1
0.00	DbAdmin	Prod	80432	Portal
2.47	DbAdmin	Prod	78925	Portal

一部のサービスでは、コスト配分のために AWS によって生成された `createdBy` タグを使用すると、分類に含まれていないリソースの説明に役立ちます。`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。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 Amazon Elastic Compute Cloud (Amazon EC2) インスタンスタグを使うと、このアクションからオプトアウトするインスタンスを簡単に指定できます。古い Amazon EBS スナップショット、 out-of-date、またはローリング Amazon EBS 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によって検索条件のディメンションが追加される可能性があります。

アクセス制御のタグ

IAM ポリシーでは、タグベースの条件をサポートしています。このため、特定のタグやタグの値に基づいて IAM アクセス許可を制限できます。たとえば、IAM ユーザーまたはロールのアクセス許可に、EC2 API コールをタグに基づいて特定の環境 (開発、テスト、本番など) に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、API 呼び出しを特定の Amazon 仮想プライベートクラウド (Amazon VPC) ネットワークに制限できます。タグベースのリソースレベルの IAM アクセス許可をサポートしているかどうかは、サービスによって異なります。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。AWS リソースへの API アクセスを制御するためのタグの使用に関する詳細については、IAM ユーザーガイドの「IAM と連携する AWS のサービス」を参照してください。

タグ付けのガバナンス

効果的なタグ付け戦略を実装するには、標準化されたタグを使用し、それをプログラミングによって AWS リソース全体に一貫して適用します。AWS 環境におけるタグの管理には、リアクティブなアプローチとプロアクティブなアプローチの両方が使用できます。

- リアクティブガバナンスの目的は、リソースグループタグ付け 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ことです。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、サービス・カタログ、AWS Organizations のタグポリシー、または IAM のリソースレベルの許可などのツールを使用して、リソース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。

  たとえば、AWS CloudFormation `Resource Tags` プロパティを使用して、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。たとえば、リソースグループタグ付け API を使用して AWS 環境のタグを検索したり、不適切にタグ付けされたリソースを隔離または削除するためのスクリプトを実行したりできます。

詳細はこちら

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。AWS の特定のサービスでリソースにタグを付ける方法の詳細については、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。

- AWS Resource Groups Tagging API の詳細については、「リソースグループのタグ付け API リファレンスガイド」を参照してください。
- タグエディタ については、このガイドの 「タグエディタ」 を参照してください。
- 各 AWS のサービス が提供するタグ機能については、「AWSドキュメンテーションインデックス」にある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 ユーザーおよびロールへのアクセスとそのユーザーおよびロールのアクセスの制御」を参照してください。

Using Tag Editor

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、AWS Secrets Manager のシークレットなどがあります。ただし、タグエディタを使用して、タグをサポートされている複数のリソースに一度に追加することもできます。さまざまな種類のリソースのクエリを作成し、検索結果のリソースのタグを追加、削除、または置換します。タグベースのクエリは 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すべてのリソースが返ります。

> **Important**
>
> 個人情報 (PII) などの機密情報や秘匿性の高い情報はタグに格納しないでください。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. AWS Management Consoleにサインインします。
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: AWS タグエディタ コンソールを使用してください。

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タイプがサポートされていない場合は、コンソールウィンドウの左下隅にある フィードバックツールを選択して、AWS に通知してください。

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

- タグおよび属性ベースのアクセスコントロール
- タグの名前に関するベストプラクティス
- タグエディタ を開始します。
- タグ付けするリソースの検索
- タグエディタ によるタグの管理
- IAM アクセス許可ポリシーでタグを使用する
- AWS Organizations タグポリシー
- サーバーレスワークフローと Amazon によるタグ変更のモニタリング EventBridge
- タグ変更のトラブルシューティング

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になる可能性があります。属性ベースのアクセス制御 (ABAC) 戦略で属性としてタグを使用するには、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 ユーザーおよびロールへのアクセスとそのユーザーおよびロールのアクセスの制御を参照してください。

また、AWS Identity and Access ManagementIAM ユーザーガイドの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には、タグを使用してさまざまなプロジェクトやグループにアクセス権を付与する方法を示す包括的なチュートリアルがあります。

シングルサインインに SAML ベースの ID プロバイダー (IdP) を使用している場合、引き受け済みのロールにタグをアタッチしてユーザーにアクセス許可を付与することができます。詳細については、AWS Identity and Access Management ユーザーガイドの IAM チュートリアル: ABAC で SAML セッションタグを使用するを参照してください。

タグの名前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 タグのキー名は大文字と小文字が区別されるので、一貫して使用するようにしてください。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。

いくつかのタグは AWS により事前に定義されています。また、さまざまな AWS のサービス によって自動的に作成されます。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、 という名の架空の会社では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。

プレフィックスを付けることで、自分の組織が定義したタグだということが明確に認識でき、AWS または使用中のサードパーティーのツールにより定義されたタグではないことがわかります。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。

タグエディタ を開始します。

タグエディタ はリソースにタグを付ける方法の 1 つです。以下のセクションを参照して、使用するための前提条件を理解してください。

タグエディタ を使用するための前提条件

リソースへのタグ付け作業を開始する前に、既存のリソースを含むアクティブな AWS アカウントと、リソースをタグ付けし、グループを作成する適切な権限があることを確認します。

トピック

- にサインアップする AWS アカウント
- 管理ユーザーの作成
- リソースの作成
- アクセス許可の設定

にサインアップする AWS アカウント

がない場合は AWS アカウント、次のステップを実行して作成します。

にサインアップするには AWS アカウント

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

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

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザーが作成されます。ルートユーザーには、アカウント内のすべての AWS のサービス とリソースにアクセスできます。セキュリティのベストプラクティスとして、管理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ます。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [アカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理ユーザーの作成

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

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

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

管理ユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。
2. IAM アイデンティティセンターで、管理ユーザーに管理アクセス権を付与します。

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

管理ユーザーとしてサインインする

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

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

リソースの作成

タグ付け AWS アカウント するリソースが にある必要があります。サポートされているリソースタイプの詳細については、「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」にある 「タグエディタ のタグ付け」列を参照してください。

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。

管理者であれば、 AWS Identity and Access Management (IAM) サービスを通じ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。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のあるポリシーを適用します。IAM ポリシーの作成とアタッチについては、「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 AWS サービスコンソールおよび APIs から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。

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付けするには、 Amazon EC2CreateTags オペレーションなどの、そのサービスの API で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。

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、IAM の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。によって維持および最新の状態に保たれる AWS マネージドポリシーを追加するか AWS、独自のカスタムポリシーを作成して維持できます。

タグエディタ のアクセス許可に AWS マネージドポリシーを使用する

タグエディタは、ユーザーに事前定義されたアクセス許可セットを提供するために使用できる以下の AWS マネージドポリシーをサポートしています。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、 AWS Resource Groups とタグエディタの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、アタッチされた IAM ロールまたはユーザーに付与します。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、次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、Resource Groups のオペレーションとタグエディタ の読み取り· 書き込みオペレーションを呼び出すアクセス許可を、アタッチされた IAM ロールまたはユーザーに付与します。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、次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 管理ポリシーは、すべてのサービスのリソースに対する読み取り専用オペレーションへのアクセス許可ReadOnlyAccessを付与します。 は、このポリシーが利用可能になると、新しい で AWS 自動的にこのポリシーを最新の状態に保ち AWS のサービス ます。
- 多くの サービスは、サービス固有の読み取り専用 AWS 管理ポリシーを提供します。このポリシーを使用して、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できます。たとえば、Amazon EC2 は AmazonEC2ReadOnlyAccess を提供しています。
- 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 (ARN) によって一覧表示される特定のリソースへのアクセスを拒否します。

```
{
    "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、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。
> 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、タグエディタでリソースを一覧表示できます。

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

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

1. IAM コンソールの「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. (ポリシーの作成) を選択します。
6. JSON タブに、以下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** です。
9. ポリシーの作成を選択します。

ポリシーが IAM に保存され、ロール、グループ、ユーザーなど他のプリンシパルにアタッチできるようになりました。プリンシパルにポリシーをアタッチする方法の詳細については、「IAM ユーザーガイド」の 「IAM アイデンティティの許可の追加および削除」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。
- ポリシーにおけるリソースレベルでのアクセス許可 – 多くのサービスでは ARN を使用してポリシーで個々のリソースを指定できます。
- 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、「 AWS Identity and Access Management ユーザーガイド」の「 の ABAC とは AWS」を参照してください。

- 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 AWS Identity and Access Management (IAM) を統合する方法の詳細については、AWS Identity and Access Management 「 ユーザーガイド」の以下のトピックを参照してください。

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

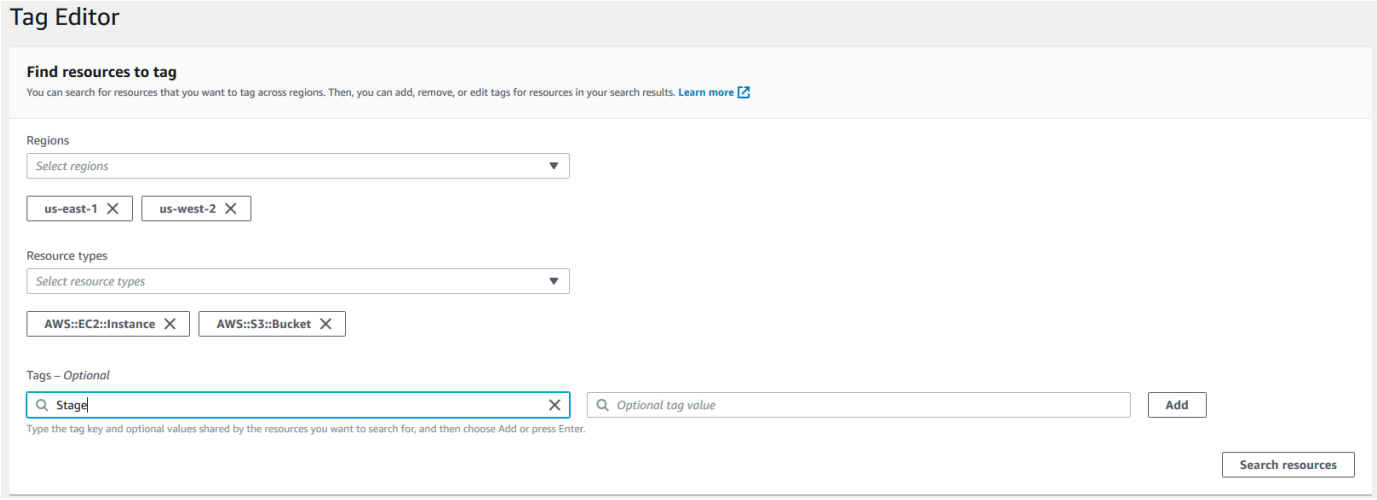
タグ付けするリソースの検索

タグエディタを使用して、タグ付けに使用できる 1 つまたは複数の AWS リージョン でリソースを検索するためのクエリを作成します。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. 「オプショナル」タグ付けするリソースを検索する AWS リージョン 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と を選択しますAWS::S3::Bucket。



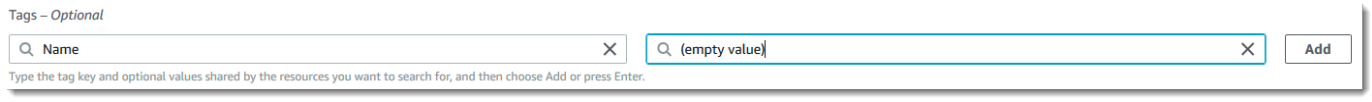
4. 「オプショナル」タグフィールドで、タグキーまたはタグのキーと値のペアを指定して、現在の AWS リージョン内のリソースを指定された値でタグ付けされたもののみに制限します。タグキーを入力すると、現在のリージョンで一致するタグキーが以下のリストに表示さます。リストからタグキーを選択できます。既存のキーと一致する十分な文字を入力すると、タグエディタがタグキーを自動補完します。タグ付けが完了したら、追加 を選択するか、Enter キーを押します。この例では、ステージ のタグキーを含むリソースをフィルタリングします。タグ値はオプションですが、クエリの結果を絞り込むことができます。さらにタグを追加するには、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。

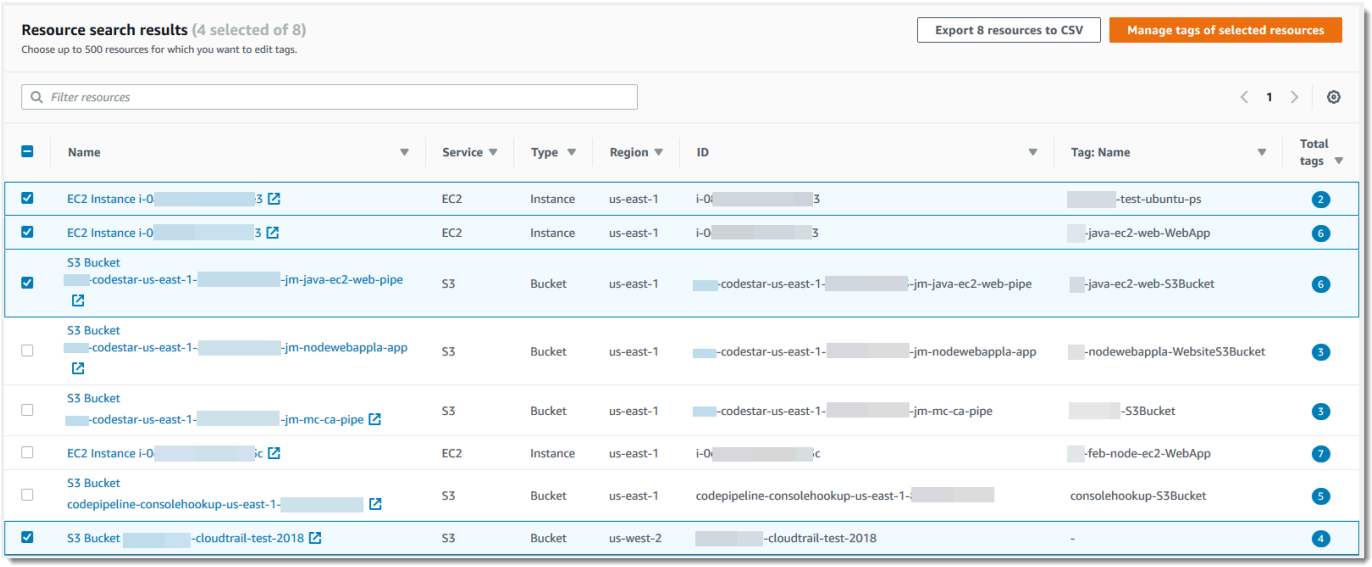
Tags (タグ) ボックスを空のままにして、選択された AWS リージョン で指定されたタイプのすべてのリソースを見つけます。このクエリは、任意のタグがあるリソースを返し、これにはタグがないリソースも含まれます。クエリからタグを削除するには、タグのラベルで X を選択します。

タグがあっても空の値を持つリソースを検索するには、以下に示されるようにタグ値ボックスにカーソルがあるときに、(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. (オプショナル)リソースの検索結果で タグエディタ に表示される列を設定するには、(リソースの検索結果) で 環境設定歯車アイコン

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。

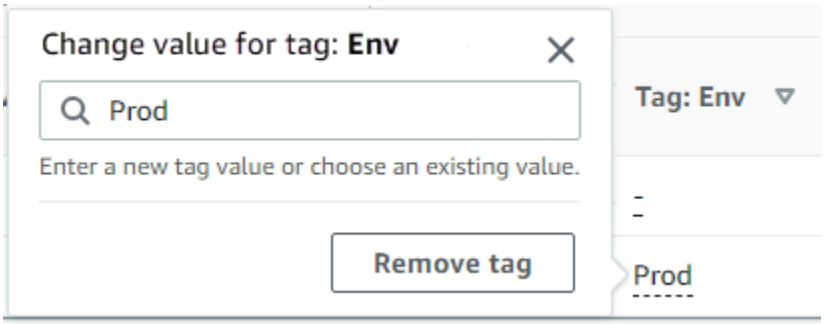
検索結果テーブルでタグをインラインで編集するには

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。

以下の例では、タグが Env で、現在の値が Prod である列が選択されています。



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

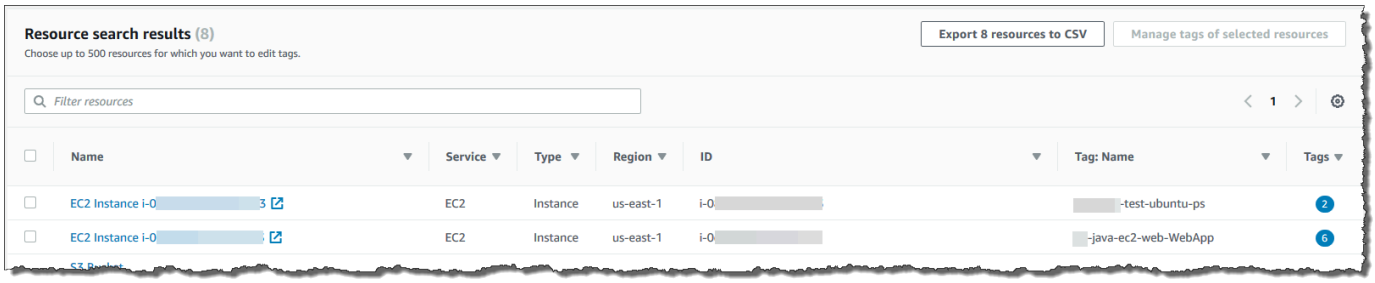
1. タグ付けするリソースを検索 クエリの結果で、既存のタグを表示するリソースの Tags (タグ) 列で数字を選択します。タグ 列でダッシュの付いたリソースには既存のタグがありません。
2. リソースタグ で既存のタグを表示します。「タグの管理」ページでタグを変更または削除するときに、「選択したリソースのタグを管理」を選択してこのウィンドウを開くこともできます。

> **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 ID 、タグの合計数、および収集内の一意のタグキーそれぞれの列が記載されています。.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. タグ付けするリソースを検索クエリの結果で、CSV にエクスポート を選択します。



2. ブラウザでプロンプトが表示されたら、 .csv ファイルを開くことを選択するか、または都合のよい場所に保存します。

関連情報

- 「AWS Billing ユーザーガイド」の「コスト配分タグの使用」

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、それらのタグがどのように タグエディタ に追加されたか、つまりリソースのサービスコンソールによるものか、または API を使用したことによるのもかについても表示されます。

> **Important**
>
> 個人情報 (PII) などの機密情報や秘匿性の高い情報はタグに格納しないでください。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、AWS Management Console における タグエディタ を使用したリソースのタグ付けについて説明します。ただし、以下のツールを使用して AWS リソースのタグを管理することもできます。

- AWS Command Line Interface (AWS CLI) で `resourcegroupstaggingapi` コマンドを使用することで、シェルプロンプトでコマンドを入力またはスクリプト化することができます。
- AWS Tools for PowerShell Core で AWS Resource Groupsタグ付け API を使用することで、 PowerShell スクリプトを作成および実行することができます。
- リソースグループタグ付け API python 用の API のタグ付け や java 用のタグ付け API) などを使用することで、利用可能な AWS SDK を使用してプログラムを作成および実行することができます。

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。

トピック

- 選択したリソースにタグを追加する
- 選択したリソースのタグの編集
- 選択したリソースからタグを削除する
- 失敗したタグの変更を再試行する
- 関連情報

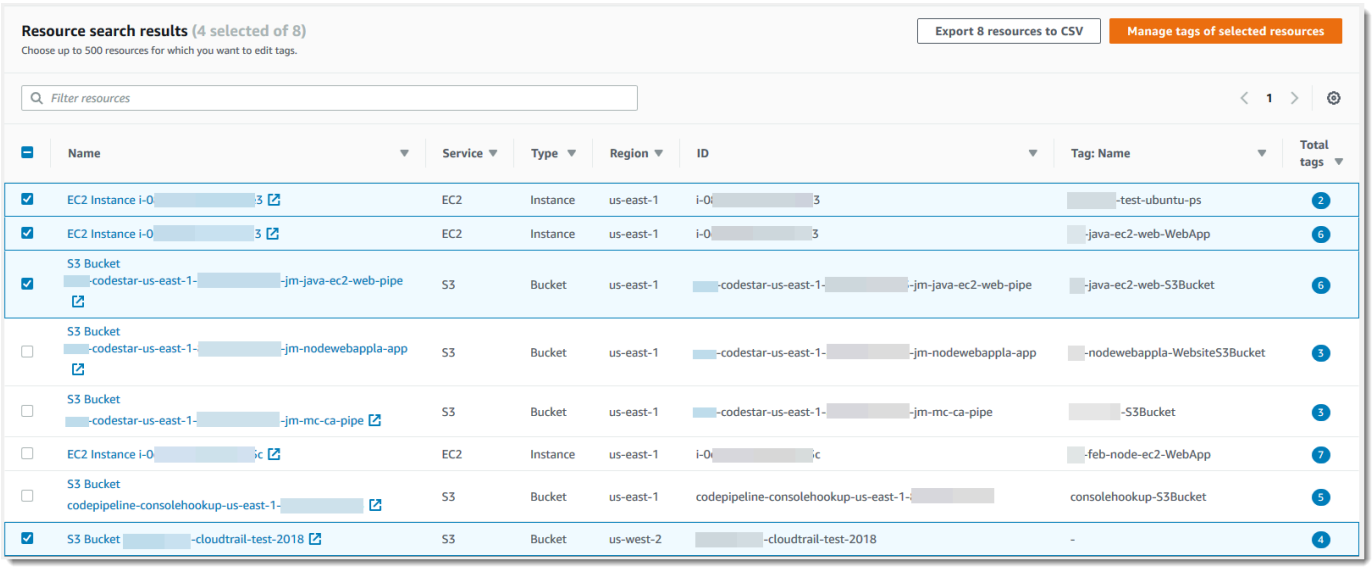
選択したリソースにタグを追加する

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。

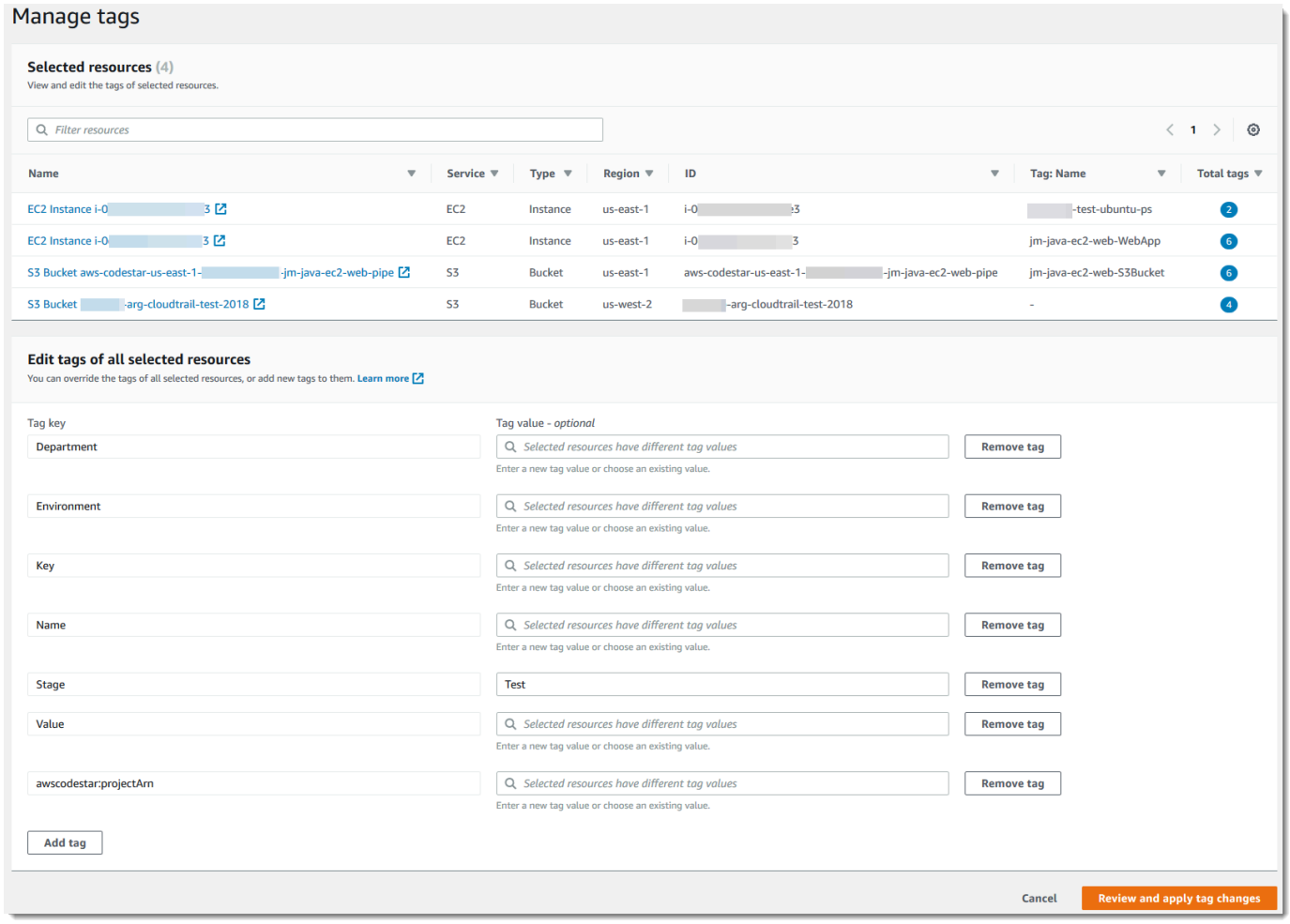
1. タグエディタコンソール を開き、タグ付けしたい複数のリソースを返すクエリを送信します。

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。次の例では、リストの最初にある EC2 インスタンスには既に 2 つのタグがあります。

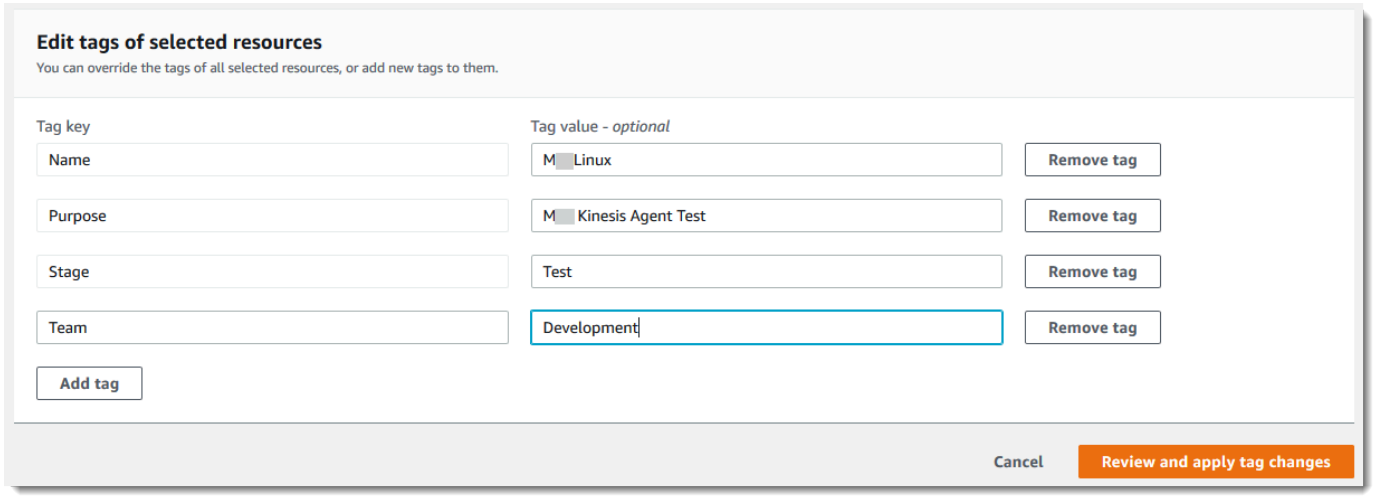


3. 1 つ以上のリソースのチェックボックスを選択して、選択したリソースのタグの管理 () を選択します。

4. 以下に示される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 個近くなると、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 生成されたタグは 50 の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。

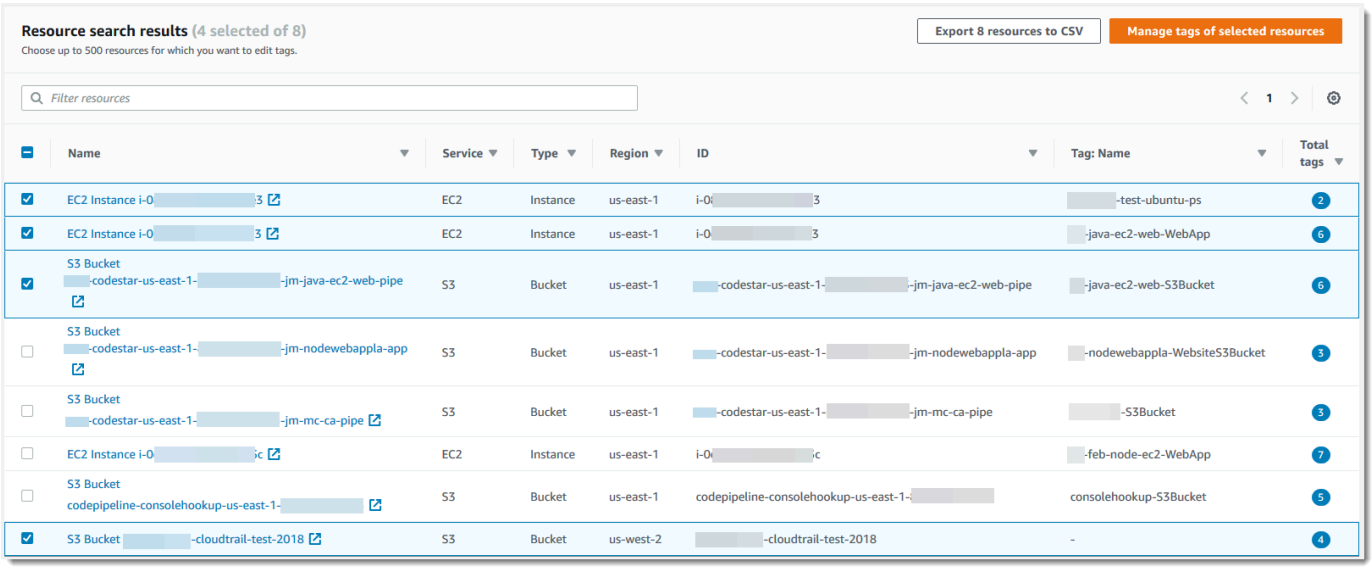
   一部またはすべてのリソースに対するタグの変更が成功しなかった場合は、「タグ変更のトラブルシューティング」を参照してください。失敗したタグの変更 「アクセス権の不足など」を解決した後は、タグの変更に失敗したリソースでタグの変更を再試行できます。詳細については、「the section called “失敗したタグの変更を再試行する”」を参照してください。

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。

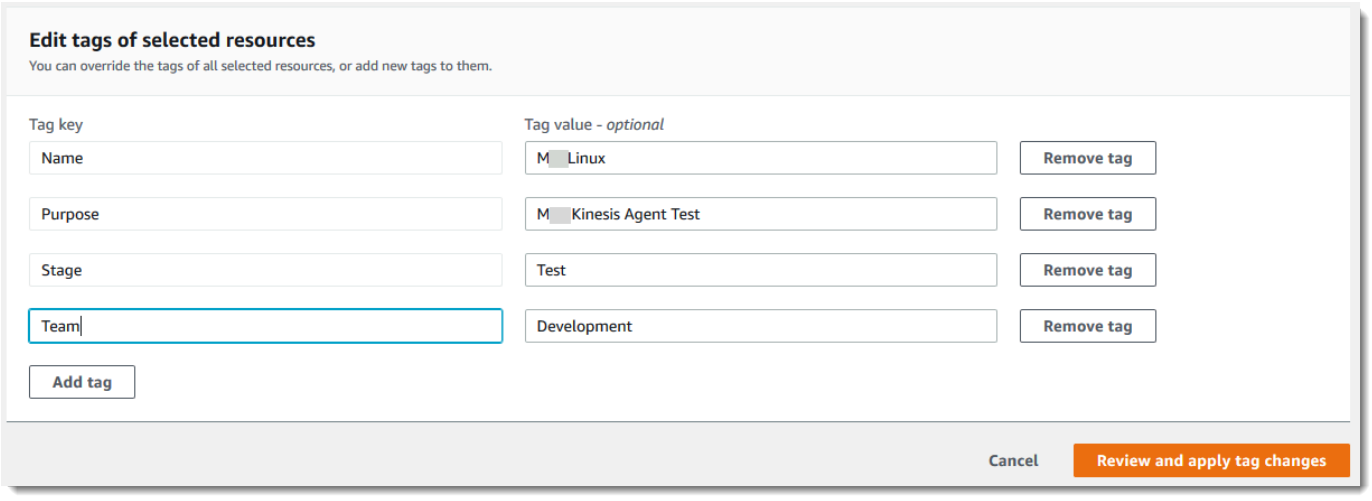
> **Important**
>
> 個人情報 (PII) などの機密情報や秘匿性の高い情報はタグに格納しないでください。タグを使用して、課金および管理サービスを提供します。タグは、プライベートデータや機密データに使用することを意図していません。

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。次の例では、最初に選択した EC2 インスタンスには既に 2 つのタグがあります。

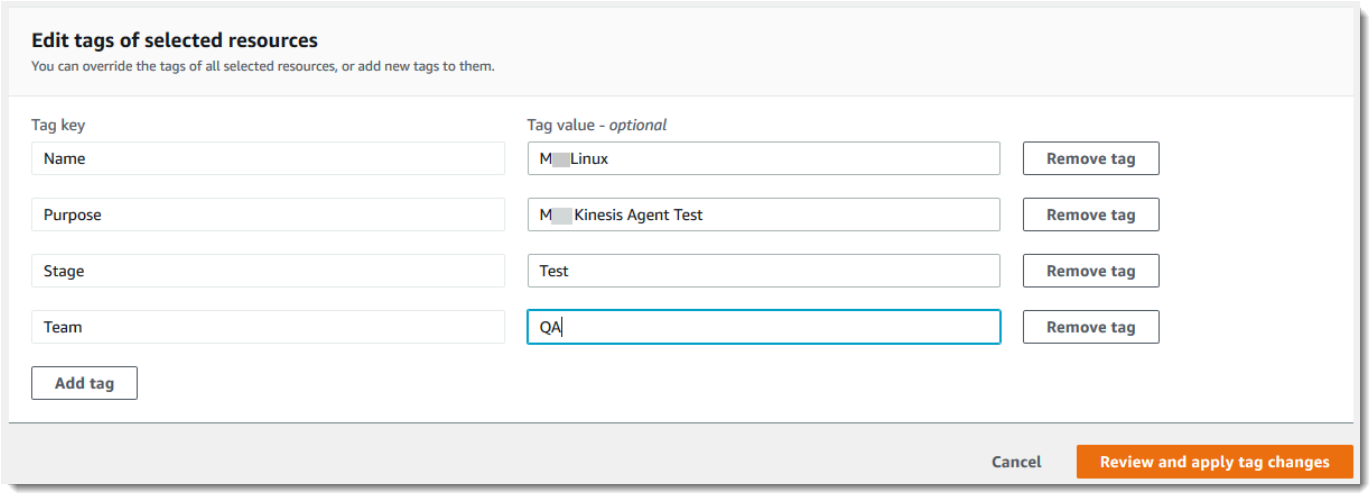


2. 選択したリソースのタグの管理 を選択します。

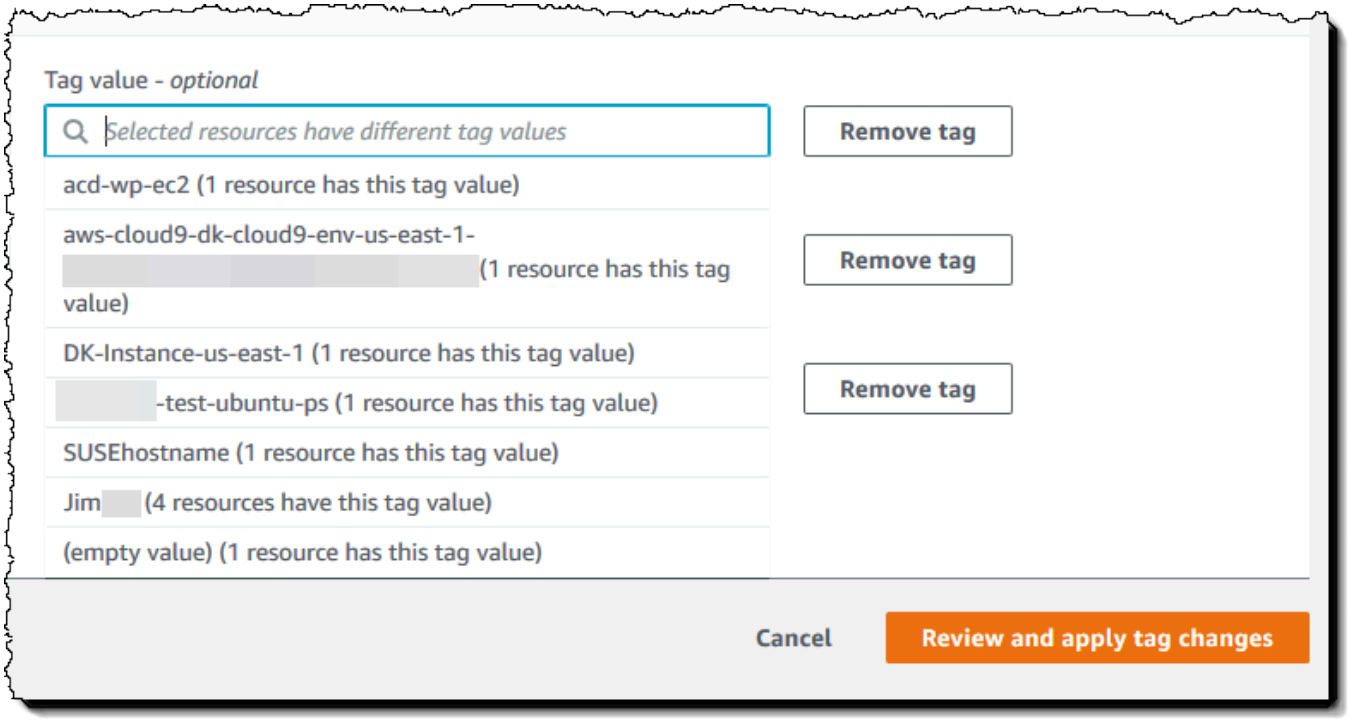
3. タグの管理 ページの 選択したリソースのタグの編集 で、選択したリソースのタグを表示します。元のクエリはより多くのリソースを返したかもしれませんが、ステップ 1 で選択したリソースのタグのみを変更しています。



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。



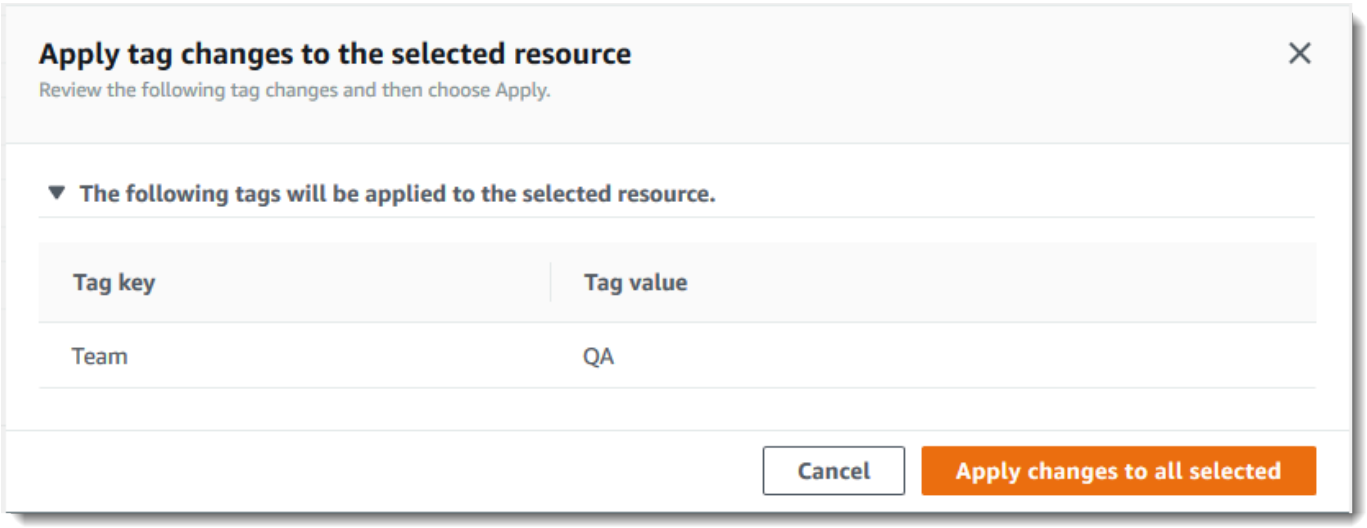
選択したリソースが同じキーに対して異なる値を持つ場合、選択したリソースのタグ値は異なります が タグ値 フィールドに表示されます。この場合、ボックス内にカーソルを置くと、選択したリソース内のこのタグキーに使用できるすべての値のドロップダウンリストが開きます。



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。



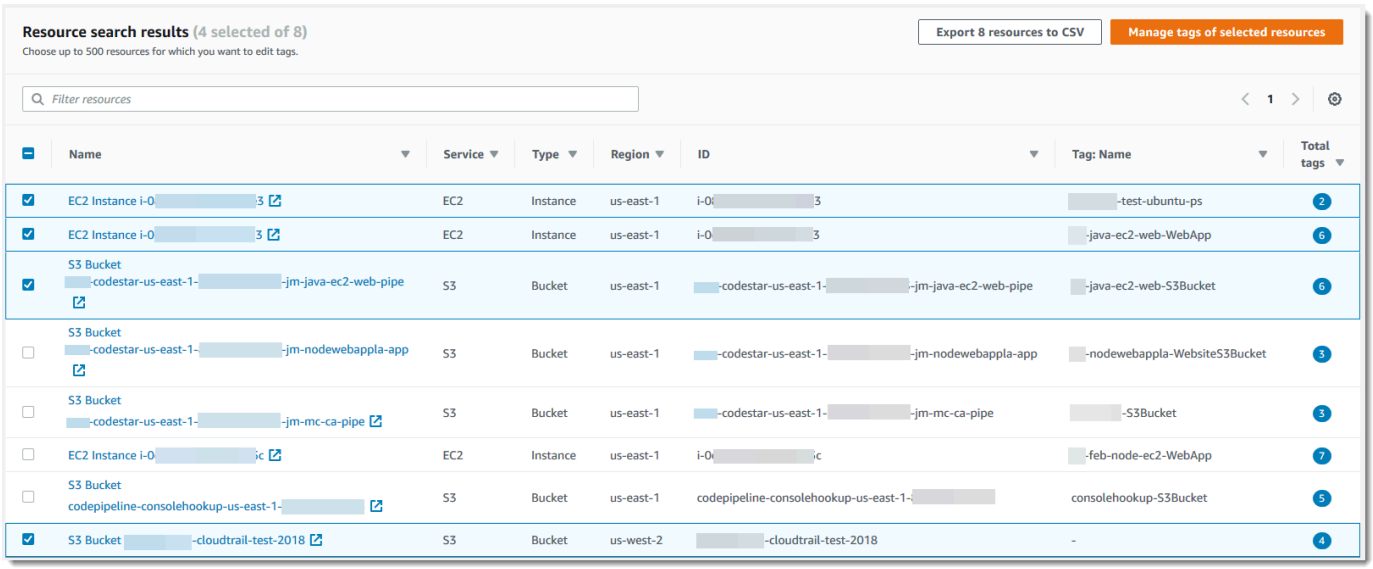
7. 選択したリソースの数によっては、タグの編集には数分かかることがあります。同じブラウザタブでページを離れたり、別のページを開いたりしないでください。変更が成功した場合は、緑色の成功バナーがページ上部に表示されます。続行する前に、成功または失敗のバナーがページに表示されるのを待ちます。

   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。

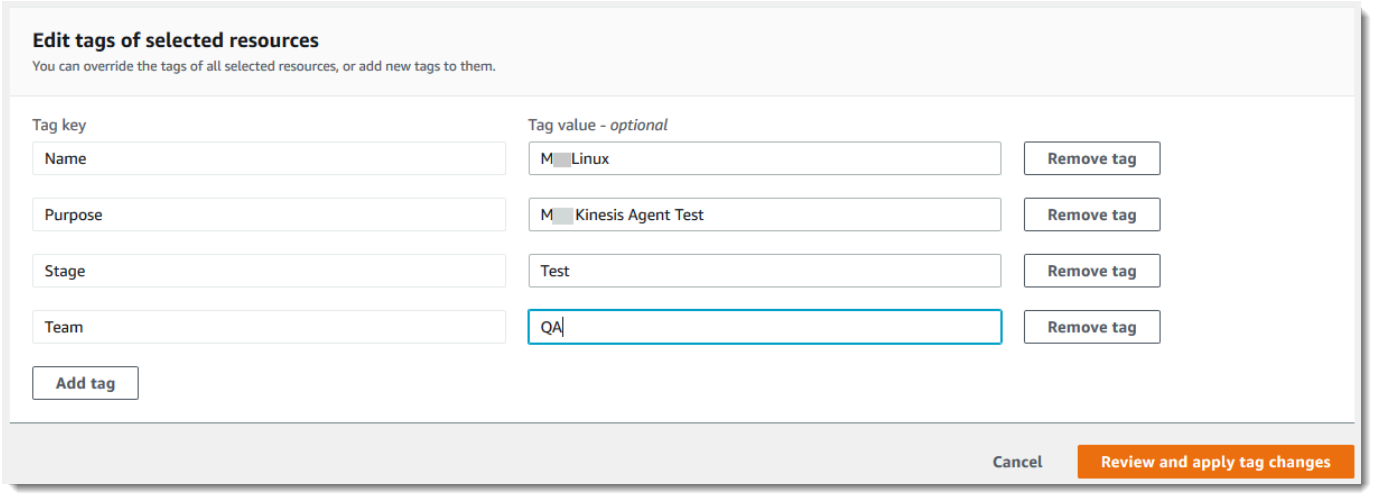
選択したリソースからタグを削除する

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。

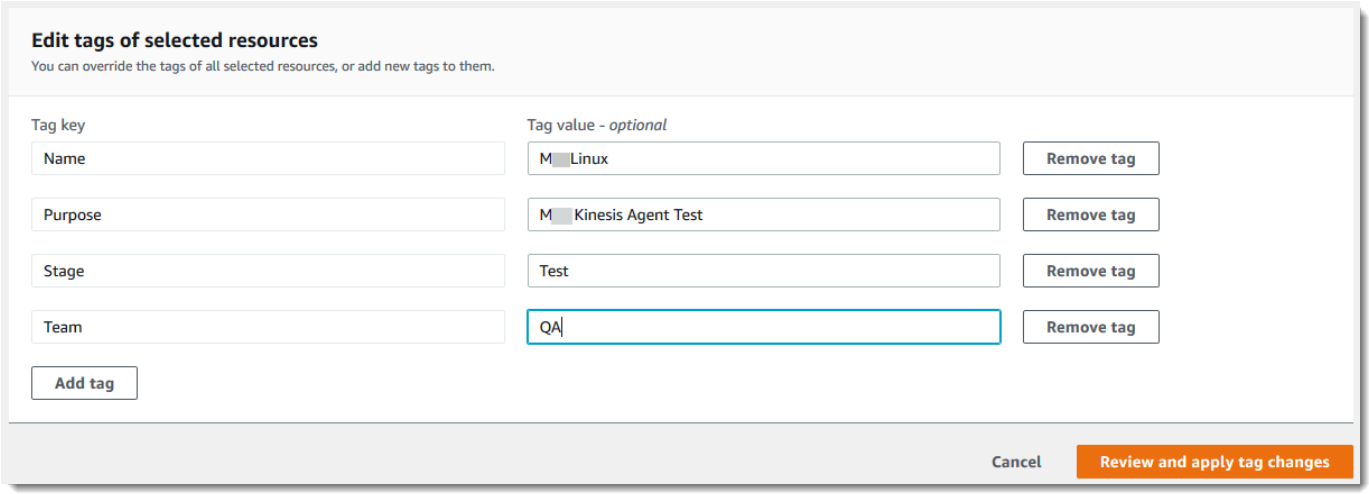


2. 選択したリソースのタグの管理 を選択します。
3. タグの管理 ページの、選択したリソースのタグの管理で、選択したリソースのタグを表示します。元のクエリはより多くのリソースを返したかもしれませんが、ステップ 1 で選択したリソースのタグのみを変更しています。



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。この例では、タグの値に関係なく、現在 **Team** タグを持っているすべての選択されたリソースから **Team** タグを削除します。



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。

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。

関連情報

- 「AWS Billing ユーザーガイド」の「コスト配分タグの使用」

IAM アクセス許可ポリシーでタグを使用する

AWS Identity and Access Management (IAM) は、誰が AWS リソースにアクセスできるのか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される AWS のサービス です。AWS サービスへのアクセスや AWS リソースの読み取り/書き込みの試行はすべて、IAM ポリシーによってアクセス制御されます。

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ために、IAM アクセス許可ポリシーで使用できる条件キーを説明しています。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。

条件キー名	説明
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 Owner が設定されたタグを含めるように強制する方法を示しています。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、呼び出し元プリンシパルがそのインスタンスと同じ project タグの値でタグ付けされている場合にのみ、実行中の Amazon EC2 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}"}
      }
    }
  ]
}
```

この例では「属性ベースのアクセス制御 (ABAC)」の例を示します。IAM ポリシーを使用したタグベースのアクセス制御戦略を実装する方法の詳細および追加の例については、「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 ユーザーおよびロールへのアクセスとそのユーザーおよびロールのアクセスの制御
- 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では、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にアクセス権を付与する方法を説明します。

AWS Organizations タグポリシー

タグポリシーは、AWS Organizations で作成するポリシーのタイプです。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間でタグを標準化できます。タグポリシーを使用するには、「AWS Organizationsユーザーガイド」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。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。

トピック

- 前提条件とアクセス許可
- アカウントのコンプライアンスの評価
- 組織全体のコンプライアンスを評価する

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。

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。

- 最初に、AWS Organizations で機能を有効にし、タグポリシーを作成してアタッチする必要があります。詳細については、AWS Organizations ユーザーガイドの以下のページを参照してください。
  - 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、AWS リージョン 米国東部 (バージニア北部) からのみリクエストできます。

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、`ec2:CreateTags` のアクセス許可が必要です。
- `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 EC2 のリソースのタグを解除するには、`ec2:DeleteTags` のアクセス許可が必要です。

次の AWS Identity and Access Management IAM ポリシーの例では、アカウントのタグ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ています。

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

IAM ポリシーおよび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。

次の IAM ポリシーの例では、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ています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EvaluateOrgCompliance",
            "Effect": "Allow",
            "Action": [
                "organizations:DescribeEffectivePolicy",
                "tag:GetComplianceSummary",
                "tag:StartReportCreation",
                "tag:DescribeReportCreation"
            ],
            "Resource": "*"
        }
    ]
}
```

IAM ポリシーおよび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、レポートを保存するため、タグポリシーのサービスプリンシパルに対して、米国東部 (バージニア北部) リージョンの Amazon Simple Storage Service (Amazon S3) バケットへのアクセス許可を付与する必要があります。以下のバケットポリシーをバケットに添付し、各########を独自の情報に置き換えます。

- S3 バケット名
- 組織の ID 番号
- ポリシーを適用する組織の管理アカウントのアカウント ID 番号。

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

```
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "tagpolicies.tag.amazonaws.com"
                 ]
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::<your-bucket-name>",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "<organization-management-account-id>",
                    "aws:SourceArn": "arn:aws:tag:us-east-1:<organization-management-
account-id>:*"
                }
            }
        },
        {
            "Sid": "TagPolicyBucketDelivery",
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "tagpolicies.tag.amazonaws.com"
                 ]
            },
            "Action": [
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": "arn:aws:s3:::<your-bucket-name>/AwsTagPolicies/<your-
organization-id>/*",
            "Condition": {
                "StringLike": {
                    "aws:SourceAccount": "<organization-management-account-id>",
                    "aws:SourceArn": "arn:aws:tag:us-east-1:<organization-management-
account-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付けされていないリソースを検索するには、 **tag:none** を使用するクエリで AWS Resource Explorer を使用します。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。組織単位 (OU) にタグポリシーをアタッチすると、OU に属するすべてのアカウントと OU 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で、非準拠のタグを検索する AWS リージョン 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。1 ページごと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。

非準拠のタグを検出するには (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

AWS CLI でタグポリシーを使用する完全な手順については、AWS Organizations ユーザーガイドの AWS CLI でのタグポリシーの使用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付けされていないリソースを検索するには、 **tag:none** を使用するクエリで AWS Resource Explorer を使用します。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。

us-east-1 AWS リージョン にある組織の管理アカウントからのみレポートを生成できます。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:StartReportCreation`
- `tag:DescribeReportCreation`
- `tag:GetComplianceSummary`

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。

以下にレポートの抜粋を示します。

	A	B	C	D	E	F	G	H	I	J
1	AccountId	Region	ResourceType	ComplianceStatus	NoncompliantKeys	KeysWithNoncompliantVa	ResourceARN	Tags	LastUpdated	PolicyLastUpdated
2	111122223333	ap-southeast-1	s3:bucket	TRUE			arn:aws:s3:::bucket	{"Name":"bucket","TestKey":"TestValue"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
3	444455556666	ap-southeast-1	ec2:route-table	TRUE			arn:aws:ec2:ap-southeast-1:444455556666:route-table/table	{"Name":"route-table"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
4	123456789012	ap-southeast-2	ec2:route-table	TRUE			arn:aws:ec2:ap-southeast-2:123456789012:route-table/table-2	{"Name":"route-table"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
5	777788889999	ap-southeast-2	ec2:instance	TRUE		Name,CostCenter	arn:aws:ec2:ap-southeast-2:777788889999:instance/i-123	{"Name":"instance2","CostCenter":"0002"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
6	234567890123	us-west-1	ec2:instance	TRUE		Name	arn:aws:ec2:us-west-1:234567890123:instance/i-1234	{"Name":"instan"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
7	111111111111	us-west-1	ec2:subnet	TRUE			arn:aws:ec2:us-west-1:111111111111:subnet/subnet-	{"Name":"subnet"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
8	222222222222	us-west-2	s3:bucket	TRUE			arn:aws:s3:::bucket-3	{"Name":"bucket"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
9	333333333333	us-west-2	s3:bucket	TRUE			arn:aws:s3:::bucket-2	{"Name":"bucket"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
10	444444444444	us-east-1	ec2:elastic-ip	TRUE			arn:aws:ec2:us-east-1:444444444444:elastic-ip/eip	{"Name":"elastic-ip"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
11	555555555555	us-east-1	elasticmapreduce:cluster	TRUE		Name	arn:aws:elasticmapreduce:us-east-1:555555555555:cluster/c-1	{"Name":"cluster-2"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
12	666666666666	us-east-1	ec2:natgateway	TRUE			arn:aws:ec2:us-east-1:666666666666:natgateway/nat-1	{"Name":"natgateway"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
13	777777777777	us-east-1	ec2:natgateway	TRUE			arn:aws:ec2:us-east-1:777777777777:natgateway/nat-2	{"Name":"natgateway"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
14	888888888888	us-east-2	ec2:subnet	TRUE			arn:aws:ec2:us-east-2:888888888888:subnet/subnet-1	{"Name":"subnet"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
15	999999999999	us-east-2	ec2:route-table	TRUE	name	Name	arn:aws:ec2:us-east-2:999999999999:route-table/table-3	{"name":"route-table","Name":"route-tab"}	2023-11-01T20:50:48Z	2023-05-24T19:35:22Z
16										
17										
18										

> **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、大文字と小文字の扱いが正しくないか、または非準拠の値を持つ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一覧表示されます。
> - 組織のメンバーだった AWS アカウント をクローズしても、タグコンプライアンスレポートには最大 90 日間表示され続ける可能性があります。

組織全体のコンプライアンスレポートを生成するには (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

  AWS CLI でタグポリシーを使用する完全な手順については、AWS Organizations ユーザーガイドの AWS CLI でのタグポリシーの使用を参照してください。
- AWS API:
  - StartReportCreation
  - DescribeReportCreation

- GetComplianceSummary

サーバーレスワークフローと Amazon によるタグ変更のモニタリング 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作成できます EventBridge。たとえば、ターゲットは自動化されたワークフローを呼び出す AWS Lambda 関数かもしれません。このトピックでは、Lambda を使用して AWS リソース上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、 AWSリソースの変更を示す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おり、それらはユーザー定義のカスタム属性で、AWS リソースを簡単に整理して分類できます。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、AWSリソースのタグの状態を追跡できます。これまでは、同様の機能を実現するために API を継続的にポーリングし、複数の呼び出しをオーケストレーションしていたかもしれません。今では、個々のサービス API、タグエディタ 、Tagging API を含むタグに変更を加えると、リソースイベント時に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 配列はイベントにかかわるリソースを識別する Amazon リソースネーム (ARN) を含むみます 。これはタグが変更されたリソースです。
- detail — JSON オブジェクトであり、その内容はイベントタイプによって異なります。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 `ec2` 、つまり Amazon EC2 です。
  - Resource type – サービスのリソースタイプ。この例では、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、指定したタグがインスタンスにアタッチされていない場合に、 Amazon EC2 インスタンスを自動的に停止する方法についても説明します。 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、サーバーレスのパラダイムに従い、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。ここでいうサーバーレスとは、コードの実行に使用するサーバーをプロビジョニング、設定、管理する必要がないということです。 AWS がすべてを自動的に行うため、ユーザーは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：必須タグがない Amazon EC2 インスタンスの自動停止

トピック

- ステップ 1。Lambda 関数を作成する
- ステップ 2。必要な IAM アクセス権限をセットアップする
- ステップ 3。Lambda 関数の予備テストを行います。
- ステップ 4。関数を起動する EventBridge ルールを作成する
- ステップ 5。ソリューション全体をテストしてください。
- [概要]

管理する AWS リソースおよび AWS アカウント のプールが増えたら、タグを使用してリソースを簡単に分類できます。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。AWS リソースを効果的に管理するには、リソースに一貫したタグを付ける必要があります。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 インスタンスのタグ変更を監視して、指定したインスタンスに必要なタグが引き続き存在す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、効果的なコスト配分を実現したり、属性ベースのアクセス制御 (ABAC) に基づくセキュリティを信頼したりするために、すべてのリソースに企業のタグポリシーに従ってタグが付けられるようになります。

> **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では、このシナリオの影響をインスタンス ID のリストにあるインスタンスのみに意図的に制限しています。テストのためにシャットダウンしてもよいインスタンス ID でリストを更新する必要があります。これにより、 AWS アカウントのリージョンにおいて、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。
> 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。その後、リスト上のインスタンス ID のみに機能を制限しているコードを削除できます。

この例では JavaScript と Node.js の 16.x バージョンを使用しています。この例では、例 AWS アカウント ID 123456789012 AWS リージョン と米国東部 (バージニア北部)(`us-east-1`)を使用しています。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。

ステップ 1。Lambda 関数を作成する

Lambda 関数を作成するには

1. AWS Lambdaマネジメントコンソールを開きます。
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 のタブが開いていない場合は、ナビゲーションウィンドウの AutoEC2Terminator フォルダにある index.js ファイルを右クリックします。次に、Open を選択します。
7. index.js タブの工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 アクセス権限をセットアップする

関数を正常に実行するには、EC2 インスタンスを停止する権限を関数に付与する必要があります。AWS が提供したロール lambda_basic_execution にはその権限がありません。このチュートリアルでは、`AutoEC2Termination-role-`*`uniqueid`* 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 アクセス権限ポリシーを変更します。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 ポリシーの作成に関する詳細情報は、「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、およびセキュリティグループの変更を、プログラムによりおよびコンソールで許可する」を参照してください。

IAM アクセス権限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 Roles (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 EC2 を選択します。
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。関数は、テストイベントを からのものと同じように処理します 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ましたが、新しいタグには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 コンソールのウィンドウまたはタブに戻り、「AutoEC2Termination 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 ()内のテキストを、次のテキスト例に示されている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 インスタンスを正常に停止しようと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 まで下にスクロールし、カスタムパターン (JSON エディター)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` イベントを照合し、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。これにより、 が指定された Lambda 関数 EventBridge を呼び出すためのアクセス許可も自動的に付与されます。

ステップ 5。ソリューション全体をテストしてください。

EC2 インスタンスを作成し、タグを変更するとどうなるかを確認することで、最終結果をテストできます。

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 コンソールの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。インスタンスの作成と起動の詳細については、「Linux インスタンス用の Amazon EC2 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。「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。AutoEC2 Termination関数を選択し、コードタブを選択し、次に 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て、 `InstanceList` 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、「Linux インスタンス用 Amazon EC2 ユーザーガイド」の「個々のリソースのタグの追加と削除」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. Amazon EC2 コンソールから関数を使用して Lambda コンソールに切り替え、監視 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
```

[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のタグ変更と照合する 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。

AWS リソースでのタグ変更に対する Amazon EventBridge サポートにより、多くの でイベント駆動型のオートメーションを構築できますAWS のサービス。この機能を AWS Lambda と組み合わせることで、AWS リソースに安全にアクセスし、オンデマンドでスケーリングでき、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ツールが得られます。

tag-change-on-resource EventBridge イベントのその他のユースケースには、次のようなものがあります。

- 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モニタリングする – 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設定できます。AWS 生成されたタグは、最大 50 個の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。
- AWS Resource Groups Tagging API にはレート制限がありますが、タグ付けするサービスによって別の制限が課される場合があります。この制限には、リソースグループのタグ付け API の制限よりも前に到達する可能性があります。

関連情報

- 「AWS Billing ユーザーガイド」の「コスト配分タグの使用」

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
- タグエディタ の Identity and Access Management
- 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 のコンプライアンス検証
- タグエディタ における耐障害性
- 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 でのデータ保護

AWS 責任共有モデル は、タグエディタ における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、AWS クラウド のすべてを実行するグローバルインフラストラクチャを

保護するがあります。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービスのセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、「AWS セキュリティブログ」に投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみを各ユーザーに付与できます。また、次の方法でデータを保護することをおすすめします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- AWS CloudTrail で API とユーザーアクティビティロギングをセットアップします。
- AWS のサービス内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API 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。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

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

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、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。

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、AWS Key Management Service と統合されておらず、 AWS KMS keys はサポートされません。

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 AWS の間のすべての転送に HTTPS を使用します。タグエディタ はTransport Layer Security (TLS) 1.3 を使用しますが、TLS 1.2 もサポートします。

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するために役立つ AWS のサービス です。IAM 管理者は、誰が認証(サインイン) され、タグエディタ リソースを使用する認可を受ける (許可がある) ことができるかを制御します。IAM は、追加費用なしで使用できる AWS のサービスです。

トピック

- 対象者
- 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
- IAM で タグエディタ を使用する方法
- 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) の用途は、Tag Edutir 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、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。自社で タグエディタ で IAM を使用する方法の詳細については、「IAM で タグエディタ を使用する方法」を参照ください。

IAM 管理者 – IAM 管理者は、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について確認する場合があります。IAM で使用できる タグエディタ 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。

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。

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。

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

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには、例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあります。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制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーションユーザー、または AWS のサービスを含めることができます。

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

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための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。

その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 ユーザーガイドのポリシーの評価ロジックを参照してください。

IAM で タグエディタ を使用する方法

タグエディタ へのアクセスを管理するために IAM を使用する前に、タグエディタ でどの IAM 機能が使用できるか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 IAM と AWS のサービス 連携する方法の概要を把握するには、IAM ユーザーガイドのAWS のサービス 「IAM と連携する 」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
- タグエディタの 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 のアイデンティティベースのポリシーでは、アクションを許可または拒否する条件に加えて、許可または拒否す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、「IAM ユーザーガイド」の「IAM JSON ポリシーエレメントのリファレンス」を参照してください。

アクション

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

JSON ポリシーのAction要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じです。一致する API オペレーションのない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アクセス許可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。

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

Resource JSON ポリシーの要素は、オブジェクトあるい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は、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、リソースレベルのアクセス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、リソースレベルのアクセス許可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 AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

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

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS 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演算を使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

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

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

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

タグに基づく認可は、属性ベースのアクセス制御 (ABAC) と呼ばれるセキュリティ戦略の一部です。

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「 の ABAC とは AWS」を参照してください。

タグエディタの 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する、IAM ロールを引き受ける、またはクロスアカウントロールを引き受けることができます。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 AWS STS API オペレーションを呼び出しますGetFederationToken。

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により 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完了できます。

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別の範囲」の「AWS のサービス」と「」の「AWS のサービス」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てください。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。

AWS Artifact 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。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

AWS のサービスを使用する際のユーザーのコンプライアンス責任は、ユーザーのデータの機密性や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。AWS では、コンプライアンスに役立つ次のリソースを提供しています。

- 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を AWS にデプロイするための手順を示します。
- 「Amazon Web Services での 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を説明しています。

> **Note**
>
> すべての AWS のサービスが HIPAA 適格であるわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS コンプライアンスのリソース - このワークブックおよびガイドのコレクションは、顧客の業界と拠点に適用されるものである場合があります。
- AWS Customer Compliance Guide — コンプライアンスの観点から見た責任共有モデルを理解できます。このガイドは、AWS のサービスを保護するためのベストプラクティスを要約したものであり、複数のフレームワーク (米国標準技術研究所 (NIST)、ペイメントカード業界セキュリティ標準評議会 (PCI)、国際標準化機構 (ISO) など) にわたるセキュリティ統制へのガイダンスがまとめられています。
- 「AWS Config デベロッパーガイド」の「ルールでのリソースの評価」 - AWS Config サービスは、自社のプラクティス、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 のサービスは、AWS 内のセキュリティ状態の包括的なビューを提供します。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 Audit Manager - この AWS のサービスは、AWS の使用状況を継続的に監査して、リスクの管理方法や、規制および業界標準へのコンプライアンスの管理方法を簡素化するために役立ちます。

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。

タグエディタ の参照情報

タグエディタの参照情報には、該当するService Quotasが含まれます。

タグエディタの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	デフォルト値
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 で最低 1 文字、最大 128 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。 `_ . : / = + - @` そのプレフィックスは AWS 用に予約`aws:`されているため、キー名を で始めることはできません。 **Note** 一部の AWS のサービス には、文字または長さの制限がいくつかあります。詳細につい

名前	デフォルト値
	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。
タグ値	UTF-8 で最小 0 文字、最大 256 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、文字または長さの制限がいくつかあります。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 オペレーションを呼び出すレート	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
AWS 全般のリファレンス のタグ付けに関する内容はこのガイドに移動しました	AWS リソースのタグ付けに関するトピックは、AWS 全般のリファレンス からこのガイドに移動しました。	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沿ってガイドを更新しました。詳細については、「IAM 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。	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、「AWS Resource Groups ユーザーガイド」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。	2022 年 12 月 13 日
タグポリシーへの準拠を確認	AWS Organizations を使用してタグポリシーを作成し、アカウントにアタッチした後は、組織のアカウント内のリソースで非準拠タグを見つけることができます。	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 AWS Systems Manager コンソールから移動	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。システム・マネージャ の左側のナビゲーションバーには、 タグエディタ コンソールへのポインタがまだありますが、 タグエディタ コンソールは、AWS Management Console の左上のドロップダウンメニューから直接開くことができます。	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、昔ながらの、従来のタグエディタのメンションは削除されています。これらのツールは、AWS では利用できなくなりました。代わりに、タグエディタ を使用できます。	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、クエリ結果の CSV へのエクスポートがサポートされるようになりました	タグ付けするリソースを検索ページでクエリの結果を CSV 形式のファイル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

AWS 用語集

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

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。