ユーザーガイド

AWS Organizations

AWS Organizations: ユーザーガイド

Table of Contents

AWS Organizations の概要

AWS Organizations は、ユーザーが作成して一元管理する組織に、複数の AWS アカウント を統合するためのアカウント管理サービスです。AWS Organizations には、お客様のビジネスの予算、セキュリティ、コンプライアンスのニーズをより適切に満たすアカウント管理および一括請求 (コンソリデーティッドビリング) 機能が備わっています。組織の管理者は、組織内にアカウントを作成したり、既存のアカウントを組織に招待して参加させることができます。

このユーザーガイドでは、AWS Organizations の主要な概念を定義して、チュートリアルを提供し、組織を作成して管理する方法について説明します。

トピック

- AWS Organizations の機能
- AWS Organizations の料金
- AWS Organizations へのアクセス
- AWS Organizations のサポートとフィードバック

AWS Organizations の機能

AWS Organizations には以下の機能があります。

すべての AWS アカウント を一元管理

既存のアカウントを組織に結合して、アカウントを一元管理することができます。自動的に組織の一部であるアカウントを作成し、他のアカウントを組織に招待することができます。アカウントの一部または全部に影響するポリシーをアタッチすることもできます。

すべてのメンバーアカウントの一括請求

一括請求は AWS Organizations の機能です。組織の管理アカウントを使用して、すべてのメンバーアカウントを統合して支払うことができます。一括請求 (コンソリデーティッドビリング) を行う場合、管理アカウントは、組織のメンバーアカウントの請求情報、アカウント情報、アカウントアクティビティにアクセスできます。この情報は、Cost Explorer などのサービスに使用され、管理アカウントが組織のコストパフォーマンスを向上させるのに役立ちます。

予算、セキュリティ、コンプライアンスのニーズを満たすアカウントの階層的なグループ化

アカウントを組織単位 (OU) にグループ化し、各 OU に異なるアクセスポリシーをアタッチすることができます。たとえば、特定の規制要件を満たす AWS サービスにのみアクセスする必要が

あるアカウントがある場合、それらのアカウントを 1 つの OU に入れることができます。その後、それらの規制要件を満たさないサービスへのアクセスをブロックする OU にポリシーをアタッチすることができます。OU は、他の OU 内に 5 レベルまでネストできるため、アカウントグループを柔軟に構成できます。

各アカウントがアクセスできる AWS サービスと API アクションのコントロールを一元化するポリシー

組織の管理アカウントの管理者は、サービスコントロールポリシー (SCP) を使用して、組織内のメンバーアカウントに対するアクセス許可の上限を指定できます。SCP を使用すると、各メンバーアカウントのユーザーとロールがどの AWS サービスリソースおよび個々の API アクションにアクセスできるかを制限することができます。また、AWS のサービス、リソースおよび API アクションへのアクセスをいつ制限するかの条件も定義できます。これらの制限は、組織内のメンバーアカウントの管理者よりも優先されます。AWS Organizations がメンバーアカウントのサービス、リソース、または API アクションへのアクセスをブロックすると、そのアカウントのユーザーまたはロールはアクセスできません。このブロックは、メンバーアカウントの管理者が IAM ポリシーで明示的にそのようなアクセス許可を付与した場合でも有効なままです。

詳細については、「サービスコントロールポリシー (SCP)」を参照してください。

組織のアカウントのリソース間でタグを標準化するポリシー

タグポリシーを使用して、タグキーおよびタグ値の大文字と小文字の処理方法の設定など、一貫したタグを維持できます。

詳細については、タグポリシー を参照してください。

AWS 人工知能 (AI) および機械学習サービスによるデータの収集と保存の方法をコントロールするポリシー

AI サービスのオプトアウトポリシーを使用して、使用しない AWS AI サービスでのデータの収集と保存をオプトアウトできます。

詳細については、AI サービスのオプトアウトポリシー を参照してください。

組織のアカウントのリソースに対して自動バックアップを設定するポリシー

バックアップポリシーを使用して、組織のアカウント全体のリソースに対する AWS Backup プランを設定し、自動的に適用できます。

詳細については、バックアップポリシー を参照してください。

AWS Identity and Access Management (IAM) の統合とサポート

IAM により、個々のアカウントのユーザーとロールのきめ細かなコントロールが可能になります。AWS Organizations は、アカウントまたはアカウントのグループのユーザーとロールによって実行可能なことをコントロールできるようにすることで、きめ細かなコントロールをアカウントレベルに展開します。実際に付与されるアクセス許可は、AWS Organizations によるアカウントレベルの許可と、そのアカウント内のユーザーまたはロールレベルで IAM によって明示的に付与されるアクセス許可との論理的な共通部分です。つまり、ユーザーは AWS Organizations ポリシーと IAM ポリシーの両方で許可されているものだけにアクセスできます。どちらかがオペレーションをブロックすると、ユーザーはそのオペレーションにアクセスできません。

他の AWS サービスとの統合

選択した AWS サービスで、AWS Organizations で利用できるマルチアカウント管理サービスを利用して、組織のメンバーであるすべてのアカウントでタスクを実行できます。サービスのリストおよび組織全体のレベルにおける各サービスを利用する利点については、「AWS で使用できる のサービス AWS Organizations」を参照してください。

組織のメンバーアカウントで自動的にタスクを実行するために AWS サービスを有効にすると、AWS Organizations はそのサービス用のサービスにリンクされた IAM ロールを各メンバーアカウントに作成します。このサービスにリンクされたロールには、組織およびそのアカウントで特定のタスクを実行することをもう一方の AWS サービスに許可する IAM 許可が事前定義されています。これを機能させるため、組織内のすべてのアカウントにサービスにリンクされたロールが自動的に割り当てられます。このロールにより、信頼されたアクセスを有効にする AWS サービスに必要な、サービスにリンクされたロールが、AWS Organizations サービスで作成可能になります。サービスにリンクされたロールでこのように追加で作成されるものには IAM 許可ポリシーがアタッチされるため、指定されたサービスは、設定によって実行が必須とされるタスクだけを実行できるようになります。詳細については、「AWS Organizations を他のAWS サービスと併用する」を参照してください。

グローバルアクセス

AWS Organizations は、どの AWS リージョン でも使用できる単一のエンドポイントを持つグローバルサービスです。オペレーションを行うリージョンを明示的に選択する必要はありません。

結果的に整合性があるデータのレプリケーション

AWS Organizations には、他の多くの AWS のサービスと同様、結果整合性があります。AWS Organizations は、リージョン内の AWS のデータセンター内の複数のサーバーにデータを複製することにより、高可用性を実現します。何らかのデータの変更リクエストが成功すると、変更は

コミットされ、安全な場所に保管されます。ただし、変更は複数のサーバー間でレプリケートされる必要があります。詳細については、「変更がすぐに表示されない」を参照してください。

使用料無料

AWS Organizations は追加料金なしで提供される AWS アカウント の機能です。組織のアカウントから他の AWS サービスにアクセスした場合にのみ課金されます。他の AWS 製品の料金について詳しくは、Amazon Web Services の料金ページを参照してください。

AWS Organizations の料金

AWS Organizations は、追加料金なしで提供されます。メンバーアカウントのユーザーとロールが使用する AWS リソースに対してのみ課金されます。例えば、メンバーアカウントのユーザーまたはロールが使用する Amazon EC2 インスタンスの標準料金が請求されます。他の AWS サービスの料金については、AWS の料金を参照してください。

AWS Organizations へのアクセス

AWS Organizations は次のいずれかの方法で使用できます。

AWS Management Console

AWS Organizations コンソールは、組織と AWS リソースを管理するために使用できるブラウザベースのインターフェイスです。コンソールを使用して、組織内の任意のタスクを実行できます。

AWS コマンドラインツール

AWS コマンドラインツールを使用すると、システムのコマンドラインでコマンドを発行し、AWS Organizations および AWS タスクを実行できます。コマンドラインを使用すると、コンソールよりも高速かつ便利になります。コマンドラインツールは、AWS のタスクを実行するスクリプトを作成する場合に便利です。

AWS には、2 セットのコマンドラインツールが用意されています。

- AWS Command Line Interface (AWS CLI)。AWS CLI のインストールおよび使用の方法については、AWS Command Line Interface ユーザーガイドを参照してください。
- AWS Tools for Windows PowerShell。Tools for Windows PowerShell のインストールおよび使用の方法については、AWS Tools for Windows PowerShell ユーザーガイドを参照してください。

AWS SDK

AWS SDK は、さまざまなプログラミング言語とプラットフォーム (Java、Python、Ruby、.NET、iOS、Android など) のライブラリとサンプルコードで構成されています。SDK は、暗号署名によるリクエスト、エラーの管理、リクエストの自動再試行などのタスクを処理します。AWS SDK のダウンロードおよびインストール方法の詳細については、「Amazon Web Services 用ツール」を参照してください。

AWS Organizations HTTPS クエリ API

AWS Organizations HTTPS クエリ API を使用すると、AWS Organizations および AWS にプログラムでアクセスできます。HTTPS クエリ API を使用すると、HTTPS リクエストを直接サービスに発行できます。HTTPS API を使用する場合は、認証情報を使用してリクエストにデジタル署名するコードを含める必要があります。詳細については、HTTP クエリリクエストを作成して API を呼び出す、および AWS Organizations API リファレンスを参照してください。

AWS Organizations のサポートとフィードバック

ご意見をお待ちしております。feedback-awsorganizations@amazon.com にコメントを送信することができます。また、AWS Organizations サポートフォーラムにフィードバックと質問を掲載することができます。AWS Support フォーラムの詳細については、「フォーラムヘルプ」を参照してください。

その他の AWS リソース

- AWS トレーニングおよびコース - AWS に関するスキルを磨き、実践的経験を積むために役立つ、職務別の特別コースとセルフペースラボへのリンクです。
- AWS デベロッパーツール - AWS を使用して革新的なアプリケーションを構築する際に役立つ、ドキュメント、サンプルコード、リリースノート、その他の情報を提供するデベロッパーツールとリソースへのリンクです。
- AWS Support Center – AWS のサポートケースを作成して管理するためのハブです。フォーラム、技術上のよくある質問、サービスヘルスステータス、AWS Trusted Advisor などの便利なリソースへのリンクも含まれています。
- AWS Support - 1 対 1 での迅速な対応を行うサポートチャネルである AWS Support に関する情報のメインウェブページです。クラウド上のアプリケーションの構築および実行に関するサポートを受けることができます。

- お問い合わせ - AWS の請求、アカウント、イベント、不正使用、その他の問題などに関するお問い合わせの受付窓口です。
- AWS サイトの利用規約 – 当社の著作権、商標、お客様のアカウント、ライセンス、サイトへのアクセス、その他のトピックに関する詳細情報。

の開始方法 AWS Organizations

次のトピックでは、 AWS Organizationsの学習と使用の開始に役立つ情報を提供します。

詳細 ..。

AWS Organizations の用語と概念

AWS Organizationsを理解するために必要な用語と基本概念について説明します。このセクションでは、アカウントのユーザーが実行できることを新しいレベルで管理できるように、組織の各コンポーネントとそれらが連携する方法の基本情報を説明します。

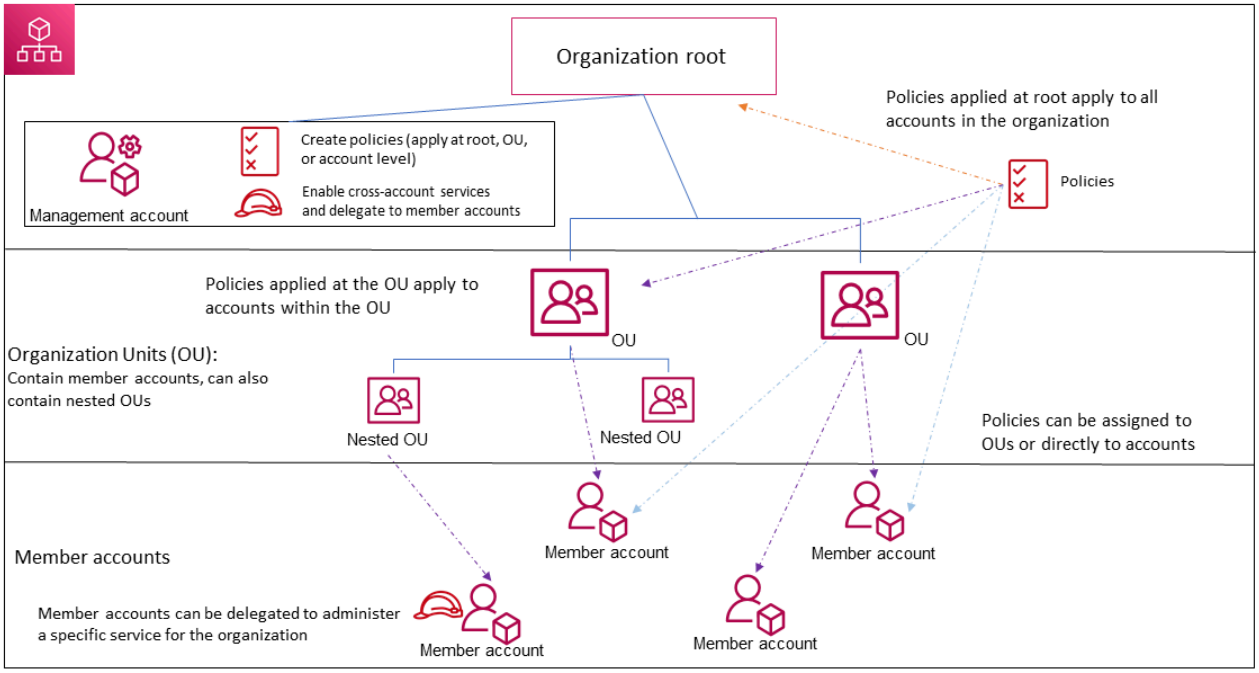
組織の一括請求

に関する主な機能の 1 つは、組織内のすべてのアカウントの請求の統合 AWS Organizations です。組織内で請求が処理される方法、および複数のアカウントで共有されるときの割引方法についての詳細を説明します。詳しくは、AWS Billing ユーザーガイドを参照してください。

AWS Organizations の用語と概念

このトピックでは AWS Organizations、 の使用を開始するために、いくつかの主要な概念について説明します。

次の図は、基本的な組織を示しています。この組織は、5 つのアカウントで構成されており、そのアカウントは、ルートを親として、4 つの組織単位 (OU) に分類されています。この組織には、複数のポリシーがあり、OU の一部、またはアカウントに直接アタッチされています。これらの各項目の詳細については、このトピックの定義を参照してください。



組織

AWS アカウントを 1 つのユニットとして管理できるようにアカウントを統合するために作成するエンティティ。AWS Organizations コンソールを使用して、組織内のすべてのアカウントを一元的に表示および管理できます。組織には、1 つの管理アカウントと、ゼロ以上のメンバーアカウントを含みます。最上部が ルートの階層ツリーを模擬した構造のアカウントや、ルート下に作られた組織単位を整理できます。各アカウントは、ルートに直接含めるか、階層内の OU のいずれかに配置することができます。組織には、有効にする機能セットによって決定された機能を含みます。

ルート

組織のすべてのアカウントが設定された親コンテナ ルートに承認ポリシーを適用すると、組織内のすべての組織単位 (OUsとメンバーアカウントに適用されます。管理ポリシーをルートに適用すると、組織内のすべての組織単位 (OUsと、組織内の管理アカウントを含むアカウントに適用されます。

> **Note**
>
> 現在、組織の作成時に root を 1 つだけ AWS Organizations 自動的に作成できます。

組織単位 (OU)

ルート内のアカウントのコンテナです。また、OU は他の OU に含めることもでき、上下反転したツリーのような階層を作成できます。最上部にはルートがあり、下に向かって OU の枝が広がり、先端にはツリーの葉であるアカウントがあります。階層内のノードのいずれかにポリシーをアタッチすると、その配下にあるすべての枝 (OU) や葉 (アカウント) に適用されます。OU は、厳密に親を 1 つ持つことができ、現在、各アカウントを厳密に 1 つの OU のメンバーにすることができます。

アカウント

Organizations のアカウントは、 AWS リソースと AWS アカウント 、それらのリソースにアクセスできる ID を含む標準です。

> **Tip**
>
> AWS アカウントはユーザーアカウントと同じではありません。AWS ユーザーは、 AWS Identity and Access Management (IAM) を使用して作成する ID で、長期の認証情報を持つ IAM ユーザーか、短期の認証情報を持つ IAM ロールのどちらかの形態をとります。1 つの AWS アカウントには多数のユーザーとロールを含めることができ、通常は含まれます。

組織は 2 種類のアカウントで構成されます。一つは管理アカウントとして指定された単一のアカウント、もう一つは 1 つ以上のメンバーアカウントです。

- 管理アカウントは、組織を作成するために使用するアカウントです。組織の管理アカウントから、以下のことができます。
  - 組織にアカウントを作成する
  - 組織に他の既存のアカウントを招待する
  - 組織からアカウントを削除する
  - 委任管理者アカウントを指定する
  - 招待を管理する
  - 組織内のエンティティ (ルート、OU、またはアカウント) にポリシーを適用する
  - サポートされている AWS サービスとの統合を有効にして、組織内のすべてのアカウントでサービス機能を提供します。

  管理アカウントには、支払いアカウントだけでなく、メンバーアカウントによって発生したすべての料金を支払う責任があります。組織の管理アカウントを変更することはできません。

- 組織内の残りのアカウントは、すべてメンバーアカウントです。アカウントが組織のメンバーになることができるのは、一度に 1 つのみです。1 つのアカウントにポリシーをアタッチして、そのアカウントのみ制御することができます。

> **Note**
>
> 一部のメンバーアカウントを委任管理者アカウントとして指定できます。下記の「委任管理者」を参照してください。

委任管理者

Organizations の管理アカウントとそのユーザーおよびロールは、そのアカウントで実行する必要のあるタスクのみに使用することをおすすめします。 AWS リソースを組織内の他のメンバーアカウントに保存し、管理アカウントからは切り離すことをおすすめします。これは、Organizations のサービスコントロールポリシー (SCP) などのセキュリティ機能は、管理アカウントのどのユーザーやどのロールにも制限を加えることができないためです。また、リソースを管理アカウントから分離することで、請求書に記載される請求額が理解しやすくなります。組織の管理アカウントから、1 つ以上のメンバーアカウントを委任管理者アカウントとして指定すると、この推奨事項を実施するうえで役立ちます。委任管理者には次の 2 種類があります。

- Organizations の委任管理者:これらのアカウントからは、組織のポリシーを管理したり、組織内のエンティティ(ルート、OU、またはアカウント)にポリシーをアタッチしたりできます。管理アカウントは、委任権限をきめ細かく制御できます。詳細については、「の委任管理者 AWS Organizations」を参照してください。
- AWS サービスの委任管理者: これらのアカウントから、Organizations と統合する AWS サービスを管理できます。管理アカウントは、必要に応じて異なるメンバーアカウントをさまざまなサービスの委任管理者として登録できます。これらのアカウントには、特定のサービスの管理者権限と、Organizations の読み取り専用アクションの権限があります。詳細については、「Organizations と連携する AWS サービスの委任管理者」を参照してください。

招待

別のアカウントを組織に招待するプロセスです。招待は、組織の管理アカウントによってのみ発行できます。招待は、招待されるアカウントに関連付けられたアカウント ID または E メールアドレスに送られます。招待済みのアカウントによって招待が承認されると、そのアカウントが、組織のメンバーアカウントになります。一括請求機能のみのサポートから、組織のすべての機能のサポートへの変更にすべてのメンバーが承認する必要が出てきた場合に、現在のすべてのメンバーアカウントに招待を送信することもできます。招待は、ハンドシェイクを交換するアカウントによって行われます。 AWS Organizations コンソールで作業している場合、ハンドシェイク

が表示されないことがあります。ただし、 AWS CLI または AWS Organizations API を使用する場合は、ハンドシェイクを直接操作する必要があります。

ハンドシェイク

二者間で情報を交換する複数ステップのプロセスです。の主な用途の 1 AWS Organizations つは、招待の基盤となる実装として機能することです。ハンドシェイクメッセージは、ハンドシェイクの開始者と受信者の間で受け渡しと応答が行われます。メッセージは、両方の当事者が現在のステータスを確実に把握できる方法で受け渡しされます。また、ハンドシェイクは、一括請求機能のみのサポートから、 で提供されるすべての機能 AWS Organizations のサポートへ組織を変更する際にも使用されます。通常、ハンドシェイクと直接やり取りする必要があるのは、 AWS Organizations API または などのコマンドラインツールを使用する場合のみです AWS CLI。

利用可能な機能セット

- すべての機能 — で使用できるデフォルトの機能セット AWS Organizations。一括請求のすべての機能だけでなく、高度な機能を使用して、組織のアカウントを詳細に制御できます。例えば、すべての機能が有効な場合、組織の管理アカウントは、メンバーアカウントで行えることを完全に制御できます。管理アカウントでは、サービスやアクションを制限する SCP を適用し、アカウントのユーザー (ルートユーザーを含む) とロールがアクセスできるサービスおよびアクションを制限できます。管理アカウントは、メンバーアカウントが組織を離れるのを防ぐこともできます。また、サポートされている AWS サービスとの統合を有効にして、それらのサービスが組織内のすべてのアカウントで機能を提供できるようにすることもできます。

  すべての機能を有効にして組織を作成するか、組織の設定を一括請求機能 (コンソリデーティッドビリング) のみのサポートからすべての機能のサポートに変更することができます。すべての機能を有効にするには、招待済みのすべてのメンバーアカウントを使用して、管理アカウントでプロセスを開始する際に送信される招待を承認し、変更を承認する必要があります。

- 一括請求 – この機能セットは共有請求機能を提供しますが、 のより高度な機能は含まれていません AWS Organizations。例えば、他の AWS のサービスが組織と統合してすべてのアカウントで動作するようにしたり、ポリシーを使用して異なるアカウントのユーザーとロールが実行できる操作を制限したりすることはできません。高度な AWS Organizations 機能を使用するには、組織内のすべての機能を有効にする必要があります。

サービスコントロールポリシー (SCP)

ユーザーやロールが、SCP による影響を受けるアカウントで使用できるサービスやアクションを指定するポリシーです。アクセス許可が付与されない点を除き、SCP は IAM 許可ポリシーと類似しています。代わりに、SCP は組織、組織単位 (OU) あるいはアカウントに最大アクセス権限を指定します。SCP を組織の root あるいは OU にアタッチすると、SCP はメンバーアカウント内のエントリのアクセス権限を制限します。

許可リストと拒否リスト

許可リストと拒否リストは、SCP を適用してアカウントで利用できるアクセス許可をフィルタ処理するための補足的な戦略です。

- 許可リスト戦略 – 許可されるアクセスを明示的に指定します。その他のアクセス権限はすべて暗黙的にブロックされます。デフォルトでは、 は、 と呼ばれる AWS 管理ポリシーFullAWSAccessをすべてのルート、OUs、および アカウントにア AWS Organizations タッチします。そのため、組織を構成する際は、設定しない限り一切ブロックされません。つまり、デフォルトではすべてのアクセス権限が許可されます。アクセス許可を制限する準備ができたら、FullAWSAccess ポリシーを、より限定的かつ適切な一連のアクセス許可だけを使用できるポリシーに変更します。影響を受けるアカウントのユーザーとロールは、IAM ポリシーではすべてのアクションが許可されている場合でも、指定されたレベルのアクセスだけが可能になります。ルートのデフォルトポリシーを変更する場合、組織内のアカウントはすべて、制限が適用されます。SCP は、アクセス許可を付与することができず、フィルタ処理のみを行うため、階層の下位レベルで追加し直すことはできません。
- 拒否リスト戦略 – 許可されないアクセスを明示的に指定します。その他のアクセス権限はすべて有効になります。このシナリオでは、明示的にブロックされる場合を除き、すべてのアクセス権限が有効です。これは のデフォルトの動作です AWS Organizations。デフォルトでは、 AWS Organizations は という AWS マネージドポリシーFullAWSAccessをすべてのルート、OUs、および アカウントにアタッチします。これにより、すべての アカウントは、 AWS Organizations課された制限なしで任意のサービスまたはオペレーションにアクセスできます。上記の許可リスト手法とは異なり、拒否リストを使用する場合は、デフォルトの FullAWSAccess ポリシー (「すべて」を許可する) をそのまま使用します。ただしその後、不要なサービスやアクションへのアクセスを明示的に拒否する追加のポリシーをアタッチします。IAM 許可ポリシーの場合と同様、サービスアクションの明示的拒否により、そのアクションに対するすべての許可は上書きされます。

人工知能 (AI) サービスのオプトアウトポリシー

組織内のすべてのアカウントで AWS AI サービスのオプトアウト設定を標準化するのに役立つタイプのポリシー。特定の AWS AI サービスは、Amazon AI サービスとテクノロジーの開発と継続的な改善のために、それらのサービスによって処理された顧客コンテンツを保存して使用できます。 AWS カスタマーは、AI サービスのオプトアウトポリシーを使用して、コンテンツの保存またはサービスの改善に使用されたことをオプトアウトできます。

バックアップポリシー

このタイプのポリシーは、組織内のアカウント全体で、リソースのバックアップ戦略を標準化、実装するのに役立ちます。リソースのバックアッププランの設定とデプロイは、バックアップポリシーで行うことができます。

タグポリシー

このタイプのポリシーは、組織のアカウント全体のリソース間でタグを標準化するのに役立ちます。タグポリシーでは、特定のリソースのタグ付けルールを指定できます。

AWS SDK AWS Organizations での の使用

AWS Software Development Kit (SDKsは、多くの一般的なプログラミング言語で使用できます。各 SDK には、デベロッパーが好みの言語でアプリケーションを簡単に構築できるようにする API、コード例、およびドキュメントが提供されています。

SDK ドキュメント	コード例
AWS SDK for C++	AWS SDK for C++ コード例
AWS CLI	AWS CLI コード例
AWS SDK for Go	AWS SDK for Go コード例
AWS SDK for Java	AWS SDK for Java コード例
AWS SDK for JavaScript	AWS SDK for JavaScript コード例
AWS SDK for Kotlin	AWS SDK for Kotlin コード例
AWS SDK for .NET	AWS SDK for .NET コード例

SDK ドキュメント	コード例
AWS SDK for PHP	AWS SDK for PHP コード例
AWS Tools for PowerShell	PowerShell コード例のツール
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) コード例
AWS SDK for Ruby	AWS SDK for Ruby コード例
AWS SDK for Rust	AWS SDK for Rust コード例
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP コード例
AWS SDK for Swift	AWS SDK for Swift コード例

可用性の例

必要なものが見つからなかった場合。このページの下側にある [Provide feedback (フィードバックを送信)] リンクから、コードの例をリクエストしてください。

AWS Organizations のチュートリアル

このセクションのチュートリアルを使用して、AWS Organizations を使用するタスクの実行方法について学びます。

チュートリアル: 組織の作成と設定

組織の作成、最初のメンバーアカウントの招待、アカウントを含む OU 階層の作成、サービスコントロールポリシー (SCP) の適用に関する手順を実行します。

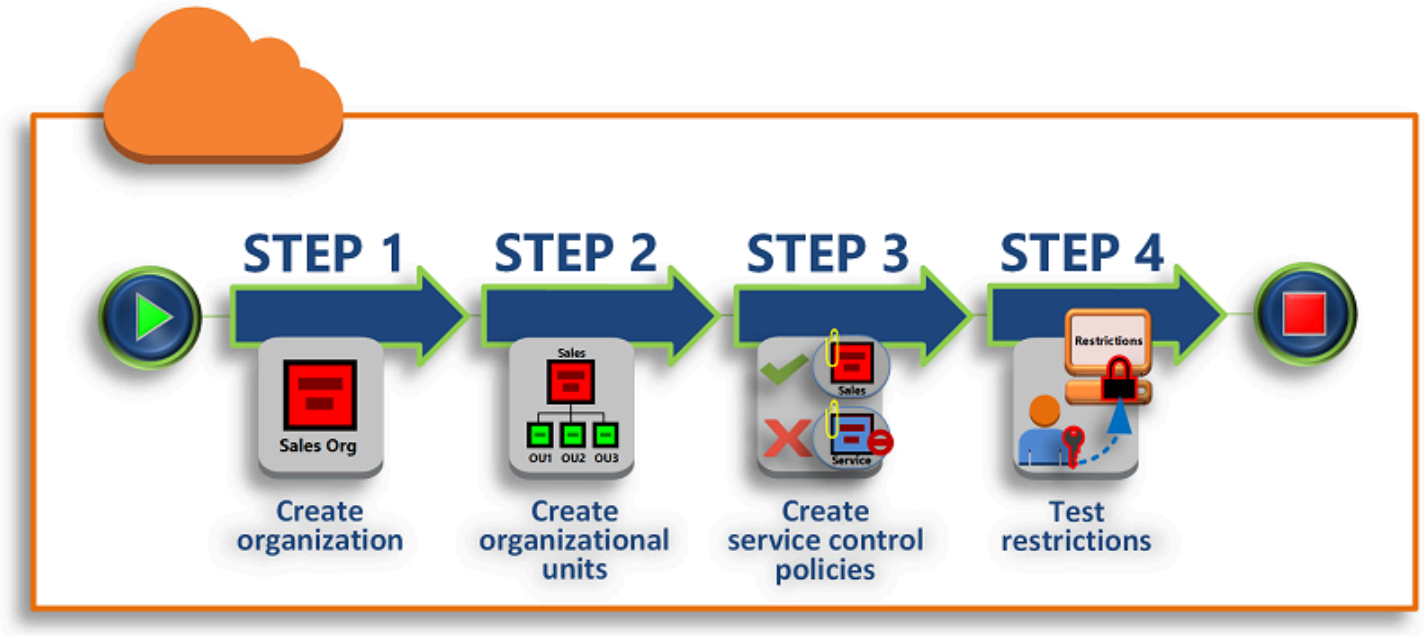
チュートリアル: Amazon EventBridge を使用して、組織の重要な変更をモニタリングする

組織内で指定したアクションが発生した場合に、E メール、SMS テキストメッセージ、または口グエントリの形式でアラームをトリガーするように Amazon EventBridge を設定し、組織での主要な変更をモニタリングします。例えば、多くの組織は、アカウントの作成日時や、アカウントが登録解除された日時を必要としています。

チュートリアル: 組織の作成と設定

このチュートリアルでは、組織を作成し、2 つの AWS メンバーアカウントで設定します。組織のメンバーアカウントのいずれかを作成し、お客様の組織に参加する他のアカウントを招待します。次に、許可リストの手法を使用して、アカウント管理者が明示的にリストされたサービスとアクションだけを委任できるように指定します。これにより、管理者は、社内の他のユーザーが使用を許可する前に、 AWS が導入する新しいサービスを検証できます。これにより、 が新しいサービス AWS を導入した場合、管理者が適切なポリシーの許可リストにサービスを追加するまで、そのサービスは禁止されたままになります。このチュートリアルでは、拒否リストを使用して、メンバーアカウントのユーザーが が AWS CloudTrail 作成する監査口グの設定を変更できないようにする方法についても説明します。

次の図は、チュートリアルの主なステップを示しています。



ステップ 1: 組織を作成する

このステップでは、現在の を管理アカウント AWS アカウント として持つ組織を作成します。また、組織に参加する AWS アカウント よう招待し、メンバーアカウントとして 2 つ目のアカウントを作成します。

ステップ 2: 組織単位を作成する

次に、新しい組織に 2 つの組織単位 (OU) を作成し、それらの OU にメンバーアカウントを配置します。

ステップ 3: サービスコントロールポリシーを作成する

サービスコントロールポリシー (SCP) を使用して、メンバーアカウントのユーザーおよびロールに委任できるアクションを制限することができます。このステップでは、2 つの SCP を作成し、それらを組織の OU にアタッチします。

ステップ 4: 組織のポリシーをテストする

各テストアカウントからユーザーとしてサインインし、SCP がアカウントに及ぼす影響を確認することができます。

このチュートリアルのどのステップでも、 AWS bill. にコストはかかりません。 は無料サービス AWS Organizations です。

前提条件

このチュートリアルでは、2 つの既存の AWS アカウント (このチュートリアルの一部として 3 つ目の を作成する) へのアクセス権があり、各 に管理者としてサインインできることを前提としています。

このチュートリアルでは、アカウントを次のように参照します。

- `111111111111` - 組織を作成するために使用するアカウント。このアカウントが管理アカウントになります。このアカウントの所有者には、E メールアドレス `OrgAccount111@example.com` があります。
- `222222222222` - メンバーアカウントとして組織に参加するように招待されたアカウント。このアカウントの所有者には、E メールアドレス `member222@example.com` があります。
- `333333333333` - 組織のメンバーとして作成するアカウント。このアカウントの所有者には、E メールアドレス `member333@example.com` があります。

上記の値をテストアカウントに関連付けられた値に置き換えます。このチュートリアルでは、本番稼働用アカウントを使用しないことをお勧めします。

ステップ 1: 組織を作成する

このステップでは、管理者としてアカウント 111111111111 にサインインし、そのアカウントを管理アカウントとして組織を作成し、メンバーアカウントとして参加するように既存カウント 222222222222 を招待します。

AWS Management Console

1. アカウント 111111111111 の管理者 AWS として にサインインし、AWS Organizations コンソール を開きます。
2. 概要ページで、[Create an organization] (組織を作成する) を選択します。
3. 確認ダイアログボックスで、[Create an organization] (組織を作成する) を選択します。

> **Note**
>
> デフォルトでは、組織はすべての機能を有効にして作成されます。また、一括請求機能のみを有効にした組織を作成することもできます。

AWS が組織を作成し、AWS アカウントページを表示します。別のページが表示されている場合は、左側のナビゲーションペインで AWS アカウント を選択します。

これまでに、ご利用のアカウントのメールアドレスが認証されたことがない場合は、管理アカウントに関連付けられたアドレスに、検証用の E メールが AWSによって自動的に送信されます。検証 E メールの受信には時間がかかる場合があります。

4. 24 時間以内に E メールアドレスを検証します。詳細については、「E メールアドレスの検証」を参照してください。

これで、メンバーだけのアカウントを持つ組織ができました。これは、組織の管理アカウントです

組織に参加するために既存のアカウントを招待する

現在組織がありますので、アカウントの入力を開始できます。このセクションのステップでは、参加する既存のアカウントを組織のメンバーとして招待します。

AWS Management Console

参加する既存のアカウントを招待するには

1. AWS アカウントページに移動し、[Add an AWS アカウント] (AWS アカウントの追加) を選択します。
2. 「 AWS アカウントの追加」ページで、「既存の を招待する AWS アカウント」を選択します。
3. [Email address or account ID of an AWS アカウント to invite] (招待する AWS アカウント の E メールアドレスまたはアカウント ID) ボックスに、招待するアカウントの所有者の E メールアドレスを **member222@example.com** のように入力します。または、 AWS アカウント ID 番号がわかっている場合は、代わりに入力できます。
4. [Message to include in the invitation email message] (招待 E メールのメッセージに含めるメッセージ) ボックスに、必要なテキストを入力します。このテキストに含まれているアカウントの所有者に E メールが送信されます。
5. 招待を送信 を選択します。 は招待をアカウント所有者 AWS Organizations に送信します。

> **Important**
>
> エラーがある場合、エラーメッセージを展開します。組織のアカウント制限を超過したことを示すエラーが発生した場合、または組織がまだ初期化中であるためアカウントを追加できない場合は、組織を作成してから 1 時間後にもう一度試してください。それでもエラーが解決しない場合は、AWS サポートまでお問い合わせください。

6. このチュートリアルでは、独自の招待を受け入れる必要があります。次のいずれかを実行して、コンソールの [Invitations] ページに移動します。

   - 管理アカウントから AWS 送信された E メールを開き、招待を受け入れるリンクを選択します。サインインするように求められたら、招待されたメンバーアカウントの管理者としてログインします。
   - AWS Organizations コンソールを開き、[Invitations] (招待) ページに移動します。

7. AWS アカウント ページで、[Accept] (許可)、[Confirm] (確認) の順に選択します。

> **Tip**
>
> 招待の送信が遅れることがあるため、招待を受信するまで待つ必要がある場合があります。

8. メンバーアカウントからサインアウトし、管理アカウントの管理者ユーザーとして再度サインインします。

メンバーアカウントを作成する

このセクションのステップでは、自動的に組織のメンバー AWS アカウント となる を作成します。このチュートリアルでは、このアカウントを 333333333333 とします。

AWS Management Console

メンバーアカウントを作成するには

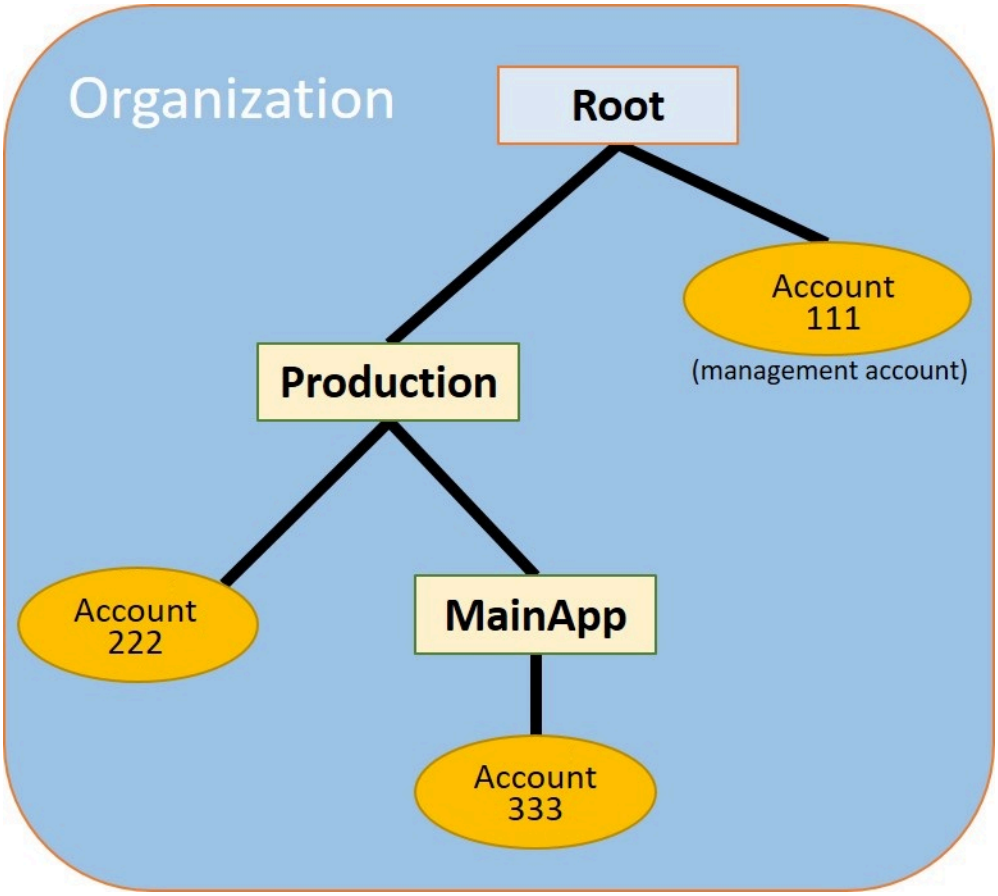
1. AWS Organizations コンソールのAWS アカウントページで、「追加 AWS アカウント」を選択します。

2. [Add an AWS アカウント] (の追加) ページで、[Create an AWS アカウント] (の作成) を選択します。
3. [AWS アカウント name] (AWS アカウント 名) には、**MainApp Account** などのアカウント名を入力します。
4. [Email address of the account's root user] (アカウントのルートユーザーの E メールアドレス) には、アカウントに代わって通信を受信する個人の E メールアドレスを入力します。この値は、グローバルで一意であることが必要です。2 つのアカウントで同じ E メールアドレスを使用することはできません。たとえば、**mainapp@example.com** のようなものを使用できます。
5. [IAM role name] の場合は、このフィールドを空白のままにしてデフォルトのロール名 `OrganizationAccountAccessRole` を自動的に使用するか、独自の名前を付けることができます。このロールを使用すると、管理アカウントで IAM ユーザーとしてサインインしたときに、新しいメンバーアカウントにアクセスできます。このチュートリアルでは、ブランクのままにして、 AWS Organizations にデフォルト名のロールを作成するよう指示します。
6. [作成] AWS アカウント を選択します。新しいアカウントが AWS アカウント ページに表示されるのを確認するには、しばらく待ってからページを更新する必要があります。

> **Important**
>
> 組織のアカウント制限を超過したことを示すエラーが発生した場合、または組織がまだ初期化中であるためアカウントを追加できない場合は、組織を作成してから 1 時間待つか、もう一度試してください。それでもエラーが解決しない場合は、AWS サポートまでお問い合わせください。

ステップ 2: 組織単位を作成する

このセクションのステップでは、組織単位 (OU) を作成し、メンバーアカウントを配置します。完了すると、階層は次の図のようになります。管理アカウントはルートのままになります。1 つのメンバーアカウントは本番稼働用 OU に移動され、もう 1 つのメンバーアカウントは本番稼働用の子である MainApp OU に移動されます。



AWS Management Console

OU を作成して設定するには

> **Note**
>
> 次の手順では、オブジェクト自体の名前またはオブジェクトの横にあるラジオボタンのいずれかを選択するため、オブジェクトを操作します。
>
> - オブジェクトの名前を選択すると、オブジェクトの詳細を表示する新しいページが開きます。
> - オブジェクトの横にあるラジオボタンをクリックすると、オプションメニューの選択など、別のアクションによって処理されるオブジェクトが表示されます。
>
> 次のステップでは、ラジオボタンをクリックしてメニューを選択し、関連するオブジェクトを処理します。

1. AWS Organizations コンソールで、AWS アカウント ページに移動します。
2. ルートコンテナの横にあるチェックボックス
   
   をオンにします。
3. Actions ドロップダウンを選択し、Organizational unit で、Create new を選択します。
4. [Create organizational unit in Root] (ルートでの組織単位の作成) ページで、[Organizational unit name] (組織単位名) に **`Production`** と入力し、[Create organizational unit] (組織単位の作成) を選択します。
5. 新しいProduction OU の横にあるチェックボックス
   
   をオンにします。
6. [Actions] (アクション) を選択し、[Organizational unit] (組織単位) で [Create new] (新規作成) を選択します。
7. [Create organizational unit in Production] (本番環境での組織単位の作成) ページで、2 番目の OU の名前として **`MainApp`** を入力し、[Create organizational unit] (組織単位の作成) を選択します。

   これで、メンバーアカウントをこれらの OU に移動できます。
8. AWS アカウントページに戻り、[Production OU] (運用 OOU) の横にある三角形
   
   をクリックして、その下のツリーを展開します。これにより、本番稼働用 MainApp の子として OU が表示されます。
9. [333333333333] の横にあるチェックボックス
   
   をオンにし (名前ではない)、[アクション] を選択してから、AWS アカウント の下の [移動] を選択します。
10. AWS アカウント 333333333333」の移動ページで、本番稼働用 の横にある三角形を選択して展開します。の横にあるMainAppラジオボタン
    
    (名前ではない) を選択し、「 の移動 AWS アカウント」を選択します。
11. [222222222222] の横にあるチェックボックス
    
    をオンにし (名前ではない)、[アクション] を選択してから、AWS アカウント の下の [移動] を選択します。

12. AWS アカウント 222222222222」の移動ページで、本番稼働用 の横にあるラジオボタン (名前ではない) を選択し、「 の移動 AWS アカウント」を選択します。

ステップ 3: サービスコントロールポリシーを作成する

このセクションのステップでは、3 つのサービスコントロールポリシー (SCP) を作成し、それらを root と OU にアタッチして、組織のアカウントのユーザーの実行範囲を制限します。最初の SCP は、メンバーアカウントのすべてのユーザーが、設定した AWS CloudTrail ログを作成または変更することを禁止します。管理アカウントは SCP の影響を受けないため、 CloudTrail SCP を適用した後、管理アカウントからログを作成する必要があります。

組織のサービスコントロールポリシータイプを有効にする

いずれかのタイプのポリシーをルートまたはルート内の任意の OU にアタッチするには、その組織のポリシータイプを有効にする必要があります。ポリシータイプは、デフォルトでは有効になっていません。このセクションのステップでは、組織のサービスコントロールポリシー (SCP) タイプを有効にする方法を示します。

AWS Management Console

組織の SCP を有効にするには

1. ポリシーページに移動し、[サービスコントロールポリシー] を選択します。
2. サービスコントロールポリシー ページで、[サービスコントロールポリシーを有効にする] を選択します。

   緑色のバナーが表示され、組織で SCP を作成できるようになりました。

SCP の作成

組織のサービスコントロールポリシーが有効になったので、このチュートリアルに必要な 3 つのポリシーを作成できます。

AWS Management Console

CloudTrail 設定アクションをブロックする最初の SCP を作成するには

1. ポリシーページに移動し、[サービスコントロールポリシー] を選択します。
2. サービスコントロールポリシーページで、[ポリシーの作成] を選択します。

3. [ポリシー名] に「**Block CloudTrail Configuration Actions**」と入力します。
4. ポリシーセクションの右側のサービスのリストで、サービスの CloudTrail を選択します。次に、、、AddTags、CreateTrail、、、DeleteTrail、RemoveTags、StopLoggingおよびStartLoggingのアクションを選択しますUpdateTrail。
5. 右側のペインで、リソースを追加 を選択し、 CloudTrailとすべてのリソース を指定します。[リソースの追加] を選択します。

   左側のポリシーステートメントは次のようになります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1234567890123",
            "Effect": "Deny",
            "Action": [
                "cloudtrail:AddTags",
                "cloudtrail:CreateTrail",
                "cloudtrail:DeleteTrail",
                "cloudtrail:RemoveTags",
                "cloudtrail:StartLogging",
                "cloudtrail:StopLogging",
                "cloudtrail:UpdateTrail"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

6. [ポリシーの作成] を選択します。

2 番目のポリシーは、Production OU のユーザーとロールが使用できるすべてのサービスとアクションの許可リストを定義します。終了したら、本稼働の OU のユーザーはリストにあるサービスとアクションのみにアクセスすることができます。

AWS Management Console

Production OU の承認されたサービスを許可する第 2 のポリシーを作成するには

1. サービスコントロールポリシーページから、[Create policy] (ポリシーの作成) を選択します。
2. [ポリシー名] に「**Allow List for All Approved Services**」と入力します。
3. カーソルの右ペインの [ポリシー] セクションに合わせ、次のようにポリシーを貼り付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1111111111111",
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "elasticloadbalancing:*",
                "codecommit:*",
                "cloudtrail:*",
                "codedeploy:*"
              ],
            "Resource": [ "*" ]
        }
    ]
}
```

4. [ポリシーの作成] を選択します。

最後のポリシーは、 MainApp OU で使用がブロックされている サービスの拒否リストを提供します。このチュートリアルでは、OU にあるすべてのアカウントで Amazon DynamoDB MainApp へのアクセスをブロックします。

AWS Management Console

MainApp OU で使用できないサービスへのアクセスを拒否する 3 番目のポリシーを作成するには

1. サービスコントロールポリシーページから、[Create policy] (ポリシーの作成) を選択します。

2. [ポリシー名] に「**Deny List for MainApp Prohibited Services**」と入力します。
3. 左側の [Policy] (ポリシー) セクションで、サービスに [Amazon DynamoDB] を選択します。アクションでは、[すべてのアクション] を選択します。
4. 引き続き左側のペインで、[Add resource](リソースの追加) を選択し、[DynamoDB] および [All Resources] (すべてのリソース) を指定します。[リソースの追加] を選択します。

   右側の更新ポリシーステートメントは次のようになります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [ "dynamodb:*" ],
      "Resource": [ "*" ]
    }
  ]
}
```

5. [ポリシーの作成] を選択して SCP を保存します。

SCP を OU にアタッチする

これで SCP はルートに対して有効になったため、ルートおよび OU にアタッチすることができます。

AWS Management Console

root と OU にポリシーをアタッチするには

1. AWS アカウント ページに移動します。
2. AWS アカウントページで、[Root] (ラジオボタンではなく名前) を選択し、詳細ページに移動します。
3. [Root] (ルート) の詳細ページで [Policies] (ポリシー) を選択してから、[Service Control Policies] (サービスコントロールポリシー) で [Attach] (アタッチ) を選択します
4. [Attach a service control policy] (サービスコントロールポリシーのアタッチ) ページで、SCP の横にある `Block CloudTrail Configuration Actions` というラジオボタンをクリックし、[Attach] (アタッチ) を選択します。このチュートリアルでは、ルートにアタッチし

て、すべてのメンバーアカウントに影響を与え、ユーザーが を設定した方法を変更できないようにします CloudTrail。

[Root] (ルート) の詳細ページの [Policies] (ポリシー) タブで、2 つの SCP (アタッチした SCP とデフォルトの `FullAWSAccess` SCP) がルートにアタッチされていることが確認できました。

5. AWS アカウントページに戻り、[Production OU] (ラジオボタンではなく名前) を選択して、詳細ページに移動します。
6. [Production OU] の詳細ページで、[Policies] (ポリシー) タブを選択します。
7. [Service Control Policies] (サービスコントロールポリシー) で [Attach] (アタッチ) を選択します。
8. [Attach a service control policy] (サービスコントロールポリシーのアタッチ) ページで、`Allow List for All Approved Services` の横にあるラジオボタンをクリックしてから [Attach] (アタッチ) を選択します。これにより、Production OUのメンバーアカウントのユーザーまたはロールが、承認されたサービスにアクセスできるようになります。
9. [ポリシー] タブを再度選択して、2 つの SCP が OU にアタッチされていることを確認します。先ほどアタッチした SCPと、デフォルトの `FullAWSAccess` SCPです。ただし、`FullAWSAccess` SCP はすべてのサービスとアクションを許可する許可リストでもあるため、承認されたサービスのみが許可されるように、今はこの SCP をデタッチする必要があります。
10. 本番稼働用 OU からデフォルトポリシーを削除するには、ラジオボタンの「フル AWSAccess」を選択し、「デタッチ」を選択し、確認ダイアログボックスの「ポリシーのデタッチ」を選択します。

    このデフォルトポリシーを削除すると、Production OU のすべてのメンバーアカウントは、直前のステップでアタッチした、許可リスト SCP にないすべてのアクションとサービスにすぐにアクセスできなくなります。Allow List for All Approved Services SCP に含まれていないアクションを使用するリクエストはすべて拒否されます。これは、アカウントの管理者が、メンバーアカウントのいずれかのユーザーに IAM アクセス許可ポリシーをアタッチして別のサービスへのアクセスを許可する場合にも当てはまります。
11. これで、 という名前の SCP をアタッチ`Deny List for MainApp Prohibited services`して、 MainApp OU 内のアカウント内の誰も制限されたサービスを使用できないようにできます。

これを行うには、AWS アカウントページに移動し、三角形のアイコンを選択して本番稼働用 OU MainApp のブランチを展開し、OU (ラジオボタンではなく名前) を選択してその内容に移動します。

12. MainApp 詳細ページで、ポリシータブを選択します。
13. 「サービスコントロールポリシー」で「アタッチ」を選択し、使用可能なポリシーのリストで MainApp 「禁止対象サービスの拒否リスト」の横にあるラジオボタンを選択し、「ポリシーのアタッチ」を選択します。

ステップ 4: 組織のポリシーをテストする

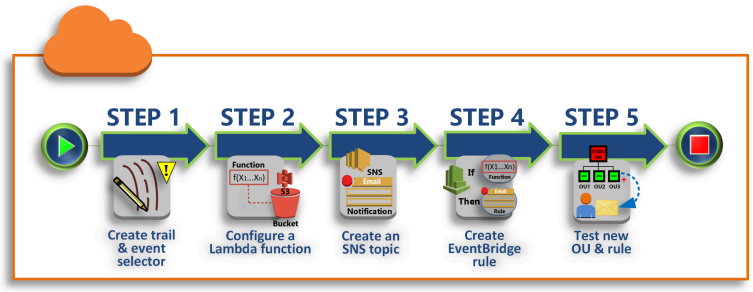
これで、メンバーアカウントのいずれかのユーザーでサインインし、さまざまな AWS アクションを実行できるようになりました。

- 管理アカウントでユーザーとしてサインインすると、IAM アクセス許可ポリシーで許可されているオペレーションを実行することができます。SCP は、アカウントがどのルートまたは OU に属していても、管理アカウントのユーザーまたはロールに影響を与えません。
- アカウント 222222222222 のユーザーとしてサインインすると、許可リストで許可されているすべてのアクションを実行できます。許可リストに含まれていないサービスでアクションを実行しようとする試みは AWS Organizations 拒否されます。また、 AWS Organizations は設定 CloudTrail アクションのいずれかの実行を拒否します。
- アカウント 333333333333 でユーザーとしてサインインすると、許可リストで許可され、拒否リストでブロックされていないアクションはすべて実行できます。 AWS Organizations は、許可リストポリシーにないアクションと拒否リストポリシーにあるアクションを実行しようとするすべての試みを拒否します。また、 AWS Organizations は設定 CloudTrail アクションのいずれかの実行を拒否します。

チュートリアル: Amazon EventBridge を使用して、組織の重要な変更をモニタリングする

このチュートリアルでは、組織の変更をモニタリングできるように Amazon EventBridge (旧 Amazon CloudWatch Events) を設定する方法を紹介します。まず、ユーザーが特定の AWS Organizations 操作を呼び出したときにトリガーされるルールを設定します。次に、ルールがトリガーされたときに AWS Lambda 関数を実行するように Amazon EventBridge を設定し、イベントに関する詳細を E メールで送信できるように Amazon SNS を設定します。

次の図は、チュートリアルの主なステップを示しています。



[ステップ 1: 証跡およびイベントセレクターを設定する]

で、証跡AWS CloudTrailと呼ばれるログを作成します。すべての API コールをキャプチャするように設定します。

[ステップ 2: Lambda 関数を設定する]

イベントに関する詳細を S3 バケットにログ記録する AWS Lambda 関数を作成します。

[ステップ 3: 受信者に E メールを送信する Amazon SNS トピックを作成する]

受信者に E メールを送信する Amazon SNS トピックを作成し、自分自身でトピックをサブスクライブします。

[ステップ 4: Amazon EventBridge ルールを作成する]

指定された API コールの詳細を Lambda 関数および SNS トピックのサブスクライバーに渡すよう Amazon EventBridge に指示するルールを作成します。

[ステップ 5: Amazon EventBridge ルールをテストする]

監視対象の操作の 1 つを実行して新しいルールをテストします。このチュートリアルでは、監視対象の操作で、組織単位 (OU) を作成しています。Lambda 関数が作成するログエントリを表示し、Amazon SNS が受信者に送信する E メールを表示します。

> **ヒント**
>
> このチュートリアルを、アカウントの作成が完了した際のメール通知を送信など、類似したオペレーションを設定するガイドとして使用することもできます。アカウントの作成は非同期オペレーションであるため、デフォルトでは完了時に通知されません。AWS CloudTrail の使用方法および AWS Organizations での Amazon EventBridge の詳細については、「[での口グ記録とモニタリング AWS Organizations]」を参照してください。

前提条件

このチュートリアルでは、次のことを前提としています。

- 組織の管理アカウントの IAM ユーザーとして AWS Management Console にサインインできます。IAM ユーザーには、CloudTrail のログ、Lambda の関数、Amazon SNS のトピック、Amazon EventBridge のルールを作成および設定するためのアクセス許可が必要です。アクセス許可を付与する方法の詳細については、IAM ユーザーガイドの「アクセス管理」または、アクセスを設定するサービスのガイドを参照してください。
- ステップ 1 で設定した CloudTrail ログを受信するための既存の Amazon Simple Storage Service (Amazon S3) バケットにアクセスできます (または、バケットを作成するアクセス許可があります)。

> **Important**
>
> 現在、AWS Organizations は、米国東部 (バージニア北部) リージョン でホストされています (グローバルに利用可能です)。このチュートリアルのステップを実行するには、そのリージョンを使用するよう AWS Management Console を設定する必要があります。

ステップ 1: 証跡およびイベントセレクターを設定する

このステップでは、管理アカウントにサインインして、AWS CloudTrail でログ (証跡と呼ばれる) を設定します。また、Amazon EventBridge がトリガーを呼び出すように、証跡でイベントセレクターを設定し、すべての読み取り/書き込み API コールをキャプチャします。

追跡を作成するには

1. 組織の管理アカウントの管理者として AWS にサインインし、https://console.aws.amazon.com/cloudtrail/ の CloudTrail コンソールを開きます。
2. コンソールウィンドウの右上隅にあるナビゲーションバーで、米国東部 (バージニア北部) リージョンを選択します。他のリージョンを選択した場合、AWS Organizations が Amazon EventBridge 設定のオプションとして表示されず、CloudTrail は AWS Organizations に関する情報をキャプチャしません。
3. ナビゲーションペインで、[Trails] (追跡) を選択します。
4. [追跡の作成]を選択します。

5. [Trail name] (証跡名) に、**`My-Test-Trail`** と入力します。
6. CloudTrail がログを配信する場所を指定するには、次のいずれかのオプションを実行します。
   - バケットを作成する必要がある場合は、[Create a new S3 bucket] (新しい S3 バケットの作成) を選択し、[Trail log bucket and folder] (Trail ログバケットとフォルダ) に新しいバケットの名前を入力します。

     > **Note**
     >
     > S3 バケット名は、グローバルに 一意である必要があります。

   - 既にバケットがある場合、[Use existing S3 bucket] (既存の S3 バケットを使用) を選択し、次に S3 バケット リストからバケット名を選択します。
7. [Next] (次へ) をクリックします。
8. [Choose log events] (ログイベントの選択) ページの [Management events] (管理イベント) セクションで、[Read] (読み取り) と [Write] (書き込み) を選択します。
9. [Next] (次へ) をクリックします。
10. 場所を確認して [Create function] (関数の作成) を選択します。

Amazon EventBridge では、アラームルールが着信 API コールと一致したときに、アラートを送信する複数の異なる方法から選択できます。このチュートリアルでは、2 つの方法について説明します。API コールをログに記録できる Lambda 関数を呼び出す方法、およびトピックの受信者へ E メールまたはテキストメッセージを送信する Amazon SNS トピックに情報を送信する方法です。次の 2 つのステップでは、必要なコンポーネントである Lambda 関数および Amazon SNS トピックを作成します。

ステップ 2: Lambda 関数を設定する

このステップでは、後で設定する Amazon EventBridge ルールによって送信される API アクティビティをログに記録する Lambda 関数を作成します。

Amazon EventBridge イベントをログに記録する Lambda 関数を作成するには

1. https://console.aws.amazon.com/lambda/ で AWS Lambda コンソールを開きます。
2. Lambda を初めて利用する場合は、ようこそページの [Get Started Now] (今すぐ始める) を選択するか、[Create function] (関数を作成) を選択します。
3. [Create function] (関数の作成) ページで、[Blueprints] (設計図) を選択します。

4. [設計図] 検索ボックスでは、フィルターに **hello** を入力し、[hello-world] 設計図を選択します。
5. [設定] を選択します。
6. [基本的な情報] ページでは、以下を実行します。

   a. [Name] (名前) テキストボックスに、Lambda 関数名として **LogOrganizationEvents** を入力します。

   b. [Role] で、[Create a new role with basic Lambda permissions] を選択します。このロールは、必要なデータにアクセスし、出力ログを書き込むために Lambda 関数にアクセス許可を付与します。
7. 次の例に示すように、 Lambda 関数のコードを編集します。

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('LogOrganizationsEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    return event.key1;  // Echo back the first key value
    // throw new Error('Something went wrong');
};
```

   このサンプルコードでは、「**LogOrganizationEvents**」マーカー文字列の後にイベントを構成する JSON 文字列を続けてイベントをログに記録します。
8. [機能の作成]を選択します。

ステップ 3: 受信者に E メールを送信する Amazon SNS トピックを作成する

このステップでは、受信者に E メールで情報を送信する Amazon SNS トピックを作成します。このトピックを、後で作成する Amazon EventBridge ルールのターゲットにします。

受信者に E メールを送信する Amazon SNS トピックを作成するには

1. https://console.aws.amazon.com/sns/v3/ で Amazon SNS コンソールを開きます。
2. ナビゲーションペインで、[トピック] を選択します。
3. [Create new topic] を選択します。

a. [トピック名] に **OrganizationsCloudWatchTopic** と入力します。

b. [Display name (表示名)] に **OrgsCWEvnt** と入力します。

c. [Create topic] (トピックの作成) を選択します。

4. トピックのサブスクリプションを作成できるようになりました。先ほど作成したトピックの ARN を選択します。

5. [Create subscription] (サブスクリプションの作成) を選択します。

a. [Create subscription] ページの [Protocol] で [Email] を選択します。

b. [エンドポイント] に E メールアドレスを入力します。

c. [Create subscription] を選択します。AWS は、前のステップで指定した E メールアドレスに E メールを送信します。E メールが送信されたら、[サブスクリプションを確認] リンクを選択して、E メールを正常に受信したことを確認します。

d. コンソールに戻り、ページを更新します。[Pending confirmation] メッセージが表示されなくなり、現在有効なサブスクリプション ID に置き換えられます。

ステップ 4: Amazon EventBridge ルールを作成する

必要な Lambda 関数がアカウントに存在するようになったので、ルールの基準が満たされた場合にそのルールを呼び出す Amazon EventBridge ルールを作成します。

EventBridge ルールを作成するには

1. https://console.aws.amazon.com/events/ で[Amazon EventBridge console] (Amazon EventBridge コンソール) を開きます。

2. コンソールを米国東部 (バージニア北部)リージョンに設定しないと、Organizations に関する情報は利用できません。コンソールウィンドウの右上隅にあるナビゲーションバーで、米国東部 (バージニア北部) リージョンを選択します。

3. ルールの作成手順については、「Amazon EventBridge ユーザーガイド」の「Getting started with Amazon EventBridge」(Amazon EventBridge の開始方法) を参照してください。

ステップ 5: Amazon EventBridge ルールをテストする

このステップでは、組織単位 (OU) を作成して Amazon EventBridge ルールを確認し、ログエントリを生成して、イベントに関する詳細を E メールで送信します。

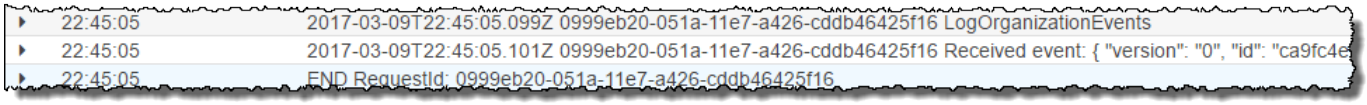
AWS Management Console

OU を作成するには

1. AWS Organizations コンソールで、[AWS アカウント] ページを開きます。
2. [☑ Root] (ルート) OU のチェックボックスをオンにし、[Action] (アクション) を選択してから、[Organizational unit] (組織単位) で [Create new] (新規作成) を選択します。
3. OU の名前では、**TestCWEOU** と入力してから、[Create organizational unit (組織単位の作成)] を選択します。

EventBridge ログエントリを表示するには

1. https://console.aws.amazon.com/cloudwatch/ にある CloudWatch コンソールを開きます。
2. ナビゲーションページで [Logs] (ログ) を選択します。
3. [Log Groups] (ロググループ) で、Lambda 関数 [/aws/lambda/LogOrganizationEvents] に関連付けられているグループを選択します。
4. 各グループには 1 つ以上のストリームがあり、今日のための 1 つのグループがあります。これを選択します。
5. ログを表示します。次のような行が表示されます。



6. エントリの中央の行を選択すると、受信したイベントの完全な JSON テキストが表示されます。API リクエストのすべての詳細は、出力の `requestParameters` および `responseElements` で確認できます。

```
2017-03-09T22:45:05.101Z 0999eb20-051a-11e7-a426-cddb46425f16 Received event:
{
    "version": "0",
    "id": "123456-EXAMPLE-GUID-123456",
    "detail-type": "AWS API Call via CloudTrail",
    "source": "aws.organizations",
    "account": "123456789012",
    "time": "2017-03-09T22:44:26Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
```

```
        "eventVersion": "1.04",
        "userIdentity": {
            ...
        },
        "eventTime": "2017-03-09T22:44:26Z",
        "eventSource": "organizations.amazonaws.com",
        "eventName": "CreateOrganizationalUnit",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "192.168.0.1",
        "userAgent": "AWS Organizations Console, aws-internal/3",
        "requestParameters": {
            "parentId": "r-exampleRootId",
            "name": "TestCWEOU"
        },
        "responseElements": {
            "organizationalUnit": {
                "name": "TestCWEOU",
                "id": "ou-exampleRootId-exampleOUId",
                "arn": "arn:aws:organizations::1234567789012:ou/o-exampleOrgId/ou-
exampleRootId-exampeOUId"
            }
        },
        "requestID": "123456-EXAMPLE-GUID-123456",
        "eventID": "123456-EXAMPLE-GUID-123456",
        "eventType": "AwsApiCall"
    }
}
```


7. E メールアカウントで、[OrgsCWEvnt] (Amazon SNS トピックの表示名) からのメッセージを確認します。E メールの本文には、前のステップで示されたログエントリと同じ JSON テキスト出力が含まれます。

クリーンアップ: 不要になったリソースを削除する

料金の発生を防ぐには、不要なこのチュートリアルの一部として作成した AWS リソースを削除する必要があります。

AWS 環境をクリーンアップするには

1. CloudTrail コンソールを使用して、ステップ 1 で作成した **My-Test-Trail** という名前の証跡を削除します。

2. ステップ 1 で Amazon S3 バケットを作成した場合は、Amazon S3 コンソールを使用して削除します。
3. Lambda コンソールを使用して、ステップ 2 で作成した **LogOrganizationEvents** という名前の関数を削除します。
4. Amazon SNS コンソールを使用して、ステップ 3 で作成した **OrganizationsCloudWatchTopic** という名前の Amazon SNS トピックを削除します。
5. CloudWatch コンソールを使用して、ステップ 4 で作成した **OrgsMonitorRule** という名前の EventBridge ルールを削除します。
6. 最後に、Organizations コンソールを使用して、ステップ 5 で作成した **TestCWEOU** という名前の OU を削除します。

これで完了です。このチュートリアルでは、組織の変更をモニタリングできるように EventBridge を設定しました。ユーザーが特定の AWS Organizations オペレーションを呼び出したときにトリガーされるルールを設定しました。ルールによって、イベントを記録した Lambda 関数が実行され、イベントに関する詳細を含む E メールが送信されました。

マルチアカウント管理のベストプラクティス

AWS Organizations でマルチアカウント環境を設定および管理する手順は、以下の事項に従って行うことをおすすめします。

トピック

アカウントを 1 つの組織内で管理する

1 つの組織を作成し、その組織内ですべてのアカウントを管理することをおすすめします。組織は、環境のアカウント間の一貫性を維持するためのセキュリティ境界です。組織のアカウントに対し、ポリシーやサービスレベルの設定を一元的に適用できます。マルチアカウント環境全体で一貫したポリシー、一元的な可視性、プログラムによる制御を実現したい場合は、単一の組織内で実現するのが最適です。

ルートユーザーには強力なパスワードを使用する

強力でユニークなパスワードを使用することをおすすめします。そのためにはさまざまなパスワードマネージャーや、複雑なパスワードを生成するアルゴリズムやツールを使用できます。詳細につい

ては、「AWS アカウントのルートユーザーのパスワードを変更する」を参照してください。ルートユーザーのパスワードの長期保存と、それへのアクセスを管理するには、会社の情報セキュリティポリシーを使用してください。パスワードは、組織のセキュリティ要件を満たすパスワードマネージャーシステムまたは同等のシステムに保存することをおすすめします。循環依存が生じるのを避けるには、保護されたアカウントでサインインする AWS サービスに依存するツールでルートユーザーのパスワードを保存しないようにします。どの方法を選択する場合でも、復元能力を優先し、保護を強化するためにこの保管庫へのアクセスには複数のアクターを必須とするようにすることがおすすめです。パスワードとその保存場所へのアクセスはすべてログに記録し、モニタリングする必要があります。ルートユーザーパスワードに関するその他の推奨事項については、「AWS アカウントのルートユーザーのベストプラクティス」を参照してください。

ルートユーザーの認証情報の使用プロセスをドキュメント化する

重要なプロセスが実行される度にドキュメント化し、各ステップに誰が関与したかの記録が残るようにします。パスワードを管理するには、安全な暗号化されたパスワードマネージャーを使用することをおすすめします。また、例外や予期しないイベントが発生する可能性がある場合は、それについてのドキュメントを用意することも重要です。詳細については、「AWS サインインユーザーガイド」の「AWS Management Console サインインのトラブルシューティング」と、「IAM ユーザーガイド」の「ルートユーザー認証情報を必要とするタスク」を参照してください。

ルートユーザーに引き続きアクセスできること、および連絡先の電話番号が有効であることのテストと妥当性確認を、少なくとも四半期ごとに実施します。このようにすることで、プロセスが機能し、ルートユーザーへのアクセスが保持されていることを、組織的に確認することができます。また、ルートアクセス担当者が、プロセスを成功させるために実行しなければならない手順を理解しているか確認することもできます。応答時間と成功率を向上させるには、プロセスに関わるすべての担当者が、アクセスが必要な場合に何をすべきかを正確に理解していることが重要です。

ルートユーザーの認証情報に対して MFA を有効にする

AWS アカウント 内の AWS アカウント ルートユーザーと IAM ユーザーに対しては、複数の多要素認証 (MFA) デバイスを有効にすることをおすすめします。これにより、AWS アカウント のセキュリティレベルを引き上げ、AWS アカウント ルートユーザーなどの権限の高いユーザーに対するアクセスの管理を簡素化できます。お客様のさまざまなニーズに対応するため、AWS は 3 種類の IAM 用の MFA デバイスをサポートしています (FIDO セキュリティキー、仮想認証アプリケーション、タイムベースドワンタイムパスワード (TOTP) ハードウェアトークン)。

オーセンティケータの種類によって、使用ケースにとって最適な物理的特性やセキュリティ特性が若干異なります。FIDO2 セキュリティキーは最高レベルの保証を提供し、フィッシング対策として有

効です。どの形式の MFA でも、パスワードのみによる認証よりもセキュリティ体制が強化されるため、アカウントに何らかの MFA を追加することを強くおすすめします。セキュリティと運用の要件に最も合ったデバイスタイプを選択してください。

TOTP ハードウェアトークンなど、プライマリ認証にバッテリー駆動のデバイスを選択する場合は、バックアップメカニズムとしてバッテリーに依存しない認証システムを登録することも検討してください。デバイスの機能を定期的にチェックし、有効期限が切れる前に交換することも、アクセスを中断させないために不可欠です。どのタイプのデバイスを選択するとしても、デバイスの紛失や障害に対する回復力を高めるために、少なくとも 2 台のデバイスを登録することをおすすめします (IAM は 1 ユーザーあたり最大 8 つの MFA デバイスをサポートします)。

MFA デバイスの保管については、組織の情報セキュリティポリシーに従ってください。MFA デバイスは、関連するパスワードとは別に保管することをおすすめします。これにより、パスワードと MFA デバイスにアクセスするのに、それぞれ異なるリソース (人、データ、ツール) が必要になります。このように分離することで、不正アクセスに対する保護をさらに強化することができます。また、MFA デバイスやその保存場所へのアクセスをすべてログに記録し、監視することをおすすめします。これにより、不正アクセスを検出して対応することができます。

詳細については、「IAM ユーザーガイド」の「多要素認証 (MFA) でルートユーザーのサインインを保護する」を参照してください。MFA を有効にする手順については、「AWS での多要素認証 (MFA) の使用」および「AWS でのユーザーの MFA デバイスの有効化」を参照してください。

ルートユーザーの認証情報へのアクセスをモニタリングするコントロールを適用する

ルートユーザーの認証情報へのアクセスは、頻繁に行われるべきではありません。管理アカウントルートユーザーの認証情報のログインと使用が通知されるよう、Amazon EventBridge などのツールを使用してアラートを作成します。通知には、少なくともルートユーザー自体に使用されているメールアドレスを含めるようにします。このアラートは重要で、見逃しにくくする必要があります。設定例については、Monitor and Notify on AWS アカウント Root User Activity を参照してください。こうした通知を受け取る担当者が、ルートユーザーのアクセスが予期されたものであることを確認する方法および、セキュリティインシデントが発生していると判断した場合、エスカレーションする方法を理解していることを確認する必要があります。詳細については、「疑わしいメールの報告」または「脆弱性レポート」を参照してください。または、AWS に問い合わせて、サポートや追加のガイダンスを求めることもできます。

連絡先の電話番号を最新の状態に保つ

AWS アカウント へのアクセスを回復するには、連絡先電話番号が有効で、テキストメッセージや電話を受信できることが重要です。AWS がアカウントのサポートや復旧のために連絡が取れるように、専用の電話番号を使用することをおすすめします。アカウントの電話番号は、AWS Management Console またはアカウント管理 API を使用して簡単に表示、管理できます。

AWS が連絡が取れるようにするための専用電話番号を取得するには、さまざまな方法があります。専用の SIM カードと電話機を入手することを強くおすすめします。電話番号をアカウントの復旧に使用できるように、電話機と SIM を長期間安全に保管します。また、携帯電話料金の請求処理を担当するチームが、この番号が長期間使用されない場合でも重要な番号であることを理解していることを確認してください。この電話番号は、さらなる保護のために、組織内で秘密にしておくことが不可欠です。

AWS 連絡先情報のコンソールページに電話番号を記録し、その詳細を組織内で知っておく必要のある特定のチームと共有します。このアプローチは、電話番号を別の SIM に転送する際に生じるリスクを最小限に抑えるのに役立ちます。電話は、既存の情報セキュリティポリシーに従って保管します。ただし、保管場所は、関連する他の認証情報と同じにしてはなりません。電話機とその保存場所へのアクセスはすべてログに記録し、モニタリングする必要があります。アカウントに関連付けられている電話番号が変更された場合は、既存のドキュメントに記載の電話番号を更新するプロセスを実施してください。

ルートアカウントにグループメールアドレスを使用する

会社が管理するメールアドレスを使用します。受信したメッセージをユーザーのグループに直接転送するメールアドレスを使用します。アクセス権の確認など、AWS からアカウントの所有者に連絡する必要が生じる事象では、メールのメッセージは複数の当事者に送信されます。こうすることで、誰かが休暇中や病欠であるか、あるいはすでに退職している場合でも、応答の遅延のリスクを軽減できます。

報告体制ではなく、ビジネス目的に基づいてワークロードをグループ化する

本番環境のワークロード環境とデータを、最上位のワークロード指向の OU に分離することをおすすめします。OU は、会社の報告体制を反映するのではなく、共通のコントロールセットに基づいて作成する必要があります。本番環境の OU とは別に、ワークロードの開発とテストに使用されるア

カウントとワークロード環境を含む非本番環境の OU を 1 つ以上定義することがおすすめです。追加のガイダンスについては、「ワークロード指向の OU を構成する」を参照してください。

複数のアカウントを使用してワークロードを整理する

AWS アカウント ではデフォルトで、AWS リソースのセキュリティ、アクセス、請求境界が提供されます。複数のアカウントを使用することには、アカウントレベルのクォータと API リクエストレートの制限を分散できるというメリットがあります。その他にも、こちらに記載されているようなメリットがあります。セキュリティ、ログ、インフラストラクチャ用のアカウントなど、組織全体の基本アカウントを複数使用することをおすすめします。ワークロードアカウントの場合、本番用ワークロードとテスト/開発用ワークロードを別々のアカウントに分ける必要があります。

サービスコンソールまたは API/CLI 操作を使用して組織レベルで AWS サービスを有効にする

ベストプラクティスとして、AWS Organizations 全体で統合したいサービスを有効または無効にするには、そのサービスのコンソール、または API 操作/CLI コマンドを使用してことをおすすめします。この方法により、必要なリソースの作成やサービスを無効にしたときのリソースのクリーンアップなど、組織に必要なすべての初期化手順を AWS サービスで実行できます。 AWS Account Management は唯一、AWS Organizations コンソールまたは API を使用して有効にする必要があるサービスです。AWS Organizations と統合されているサービスのリストを確認するには、AWS で使用できる のサービス AWS Organizations を参照してください。

請求ツールを使用してコストを追跡し、リソースの使用を最適化する

組織を管理する場合、組織内のアカウントのすべての料金が網羅的にまとめられた一括請求書が届きます。コストを可視化する必要のあるビジネスユーザーには、請求ツールやコストツールを確認するために、読み取り専用の制限つき権限ロールを管理アカウントに割り当てることができます。たとえば、請求レポートへのアクセスを許可する権限セットを作成したり、AWS Cost Explorer Service (経時的なコスト傾向を表示するツール)を使用したり、Amazon S3 ストレージレンズ、AWS Compute Optimizer などのコスト効率化サービスを使用したりできます。

組織リソース全体でのタグ付け戦略とタグの適用を計画する

アカウントやワークロードがスケールしてくると、コスト追跡、アクセス制御、リソースの整理にタグが役立ちます。タグ付けの命名方法については、「AWS リソースにタグを付ける」のガイダンスに従ってください。リソースの他にも、組織のルート、アカウント、OU、およびポリシーにタグを作成することができます。詳しくは、「タグ付け戦略の構築」を参照してください。

管理アカウントのベストプラクティス

AWS Organizations の管理アカウントのセキュリティを保護するため、以下の推奨事項に従います。これらの推奨事項は、厳密に必要とするタスクにのみルートユーザーを使用するというベストプラクティスを遵守していることを前提としています。

トピック

- 管理アカウントにアクセスできるユーザーを制限する
- 誰がアクセスできるかを確認、追跡する
- 管理アカウントは、管理アカウントが必要なタスクにのみ使用してください
- 組織の管理アカウントにワークロードをデプロイすることを避ける
- 分散化のために管理アカウント外に責任を委任する

管理アカウントにアクセスできるユーザーを制限する

管理アカウントは、アカウント管理、ポリシー、他の AWS サービスとの統合、一括請求など、前述のすべての管理タスクにとって重要です。そのため、管理アカウントへのアクセスは、組織に変更を加える権限を必要とする管理者ユーザーのみに制限する必要があります。

誰がアクセスできるかを確認、追跡する

管理アカウントへのアクセスを維持するには、そのアカウントと関連付けられたメールアドレス、パスワード、MFA、電話番号に社内の誰がアクセスできるかを定期的に確認する必要があります。確認は既存のビジネス上の手続きに則って行うことができます。担当者だけに適切なアクセスを限定できるよう、月ごとや四半期ごとにこの情報を確認してください。ルートユーザーの認証情報へのアクセスを回復またはリセットするプロセスが、特定の個人に依存しないようにしてください。すべてのプロセスは、誰かが不在だったとしても問題なく進められるよう設計します。

管理アカウントは、管理アカウントが必要なタスクにのみ使用してください

管理アカウントとそのユーザーおよびロールは、そのアカウントで実行する必要のあるタスクのみに使用することをおすすめします。すべての AWS リソースは組織内の他の AWS アカウント に保存し、管理アカウントからは切り離します。リソースを他のアカウントで保持する重要な理由の一つは、管理アカウントのユーザーとロールに対する制限に Organizations サービスコントロールポリシー (SCP) を使用できないためです。また、リソースを管理アカウントから分離することで、請求書に記載される請求額が理解しやすくなります。

組織の管理アカウントにワークロードをデプロイすることを避ける

特権操作は組織の管理アカウント内で実行でき、SCP は管理アカウントには適用されません。そのため、管理アカウントに含まれるクラウドリソースとデータは、管理アカウントで管理する必要があるものだけに制限する必要があります。

分散化のために管理アカウント外に責任を委任する

可能ならば、責任とサービスを管理アカウント外に委任することをおすすめします。チームにチーム自身の権限を提供することで、管理アカウントにアクセスすることなく、各自のアカウントから組織のニーズを管理できます。さらに、組織全体でソフトウェアを共有するための AWS Service Catalog や、スタックの作成と展開を行うための AWS CloudFormation StackSet など、この機能をサポートするサービスを利用できるよう、複数の委任管理者を登録できます。

詳細については、「セキュリティリファレンスアーキテクチャ」や、「複数のアカウントを使用した AWS 環境の構築」を参照してください。また、メンバーアカウントをさまざまな AWS サービスの委任管理者として登録する場合の推奨事項については、AWS で使用できる のサービス AWS Organizations を参照してください。委任管理者の設定の詳細については、「AWS Account Management の委任管理者アカウントの有効化」および の委任管理者 AWS Organizations を参照してください。

メンバーアカウントのベストプラクティス

組織内のメンバーアカウントのセキュリティを保護するために、以下の推奨事項に従ってください。これらの推奨事項は、厳密に必要とするタスクにのみルートユーザーを使用するというベストプラクティスを遵守していることを前提としています。

トピック

- アカウント名と属性を定義する
- 環境とアカウントの使用量を効率的にスケールする
- SCPを使用し、メンバーアカウントのルートユーザーで行えることを制限する

アカウント名と属性を定義する

メンバーアカウントには、アカウントの使用状況を反映した命名構造とメールアドレスを使用してください。たとえば、WorkloadsFooADev に「Workloads+fooA+dev@domain.com」、WorkloadsFooBDev に「Workloads+fooB+dev@domain.com」などです。組織でカスタムタグを定義している場合は、アカウントの使用方法、コストセンター、環境、およびプロジェクトを反映したアカウントにそれらのタグを割り当てることをおすすめします。これにより、アカウントの識別、整理、検索が容易になります。

環境とアカウントの使用量を効率的にスケールする

スケールする際は、新しいアカウントを作成する前に、不必要な重複を避けるために、同じようなニーズを持つアカウントが他に存在しないことを確認してください。AWS アカウント は一般的なアクセス要件に基づいている必要があります。サンドボックスアカウントや同等のアカウントを再利用する予定がある場合は、そのアカウントの不要なリソースやワークロードをクリーンアップし、後で使用できるようにアカウントを保存しておくことをおすすめします。

アカウントを閉鎖する際には、アカウント閉鎖クォータ制限が適用されることに注意してください。詳細については、「のクォータ AWS Organizations」を参照してください。可能であれば、アカウントを閉鎖して新しいアカウントを作成するのではなく、クリーンアッププロセスを実行してアカウントを再利用することを検討してください。これにより、リソースを実行することによるコストの発生や CloseAccount API の上限に達することを回避できます。

SCPを使用し、メンバーアカウントのルートユーザーで行えることを制限する

サービスコントロールポリシー (SCP) を組織内に作成して組織のルートにアタッチし、すべてのメンバーアカウントに適用されるようにすることをお勧めします。詳細については、「Organizations アカウントのルートユーザー認証情報を保護する」を参照してください。

メンバーアカウントで実行する必要がある特定のルート限定アクションを除いて、すべてのルートアクションを拒否できます。例えば、以下の SCP は、どのメンバーアカウントのルートユーザーでも、「誤って設定されており、すべてのプリンシパルへのアクセスを拒否する S3 バケットポリシー

の更新」(ルート認証情報を必要とするアクションの 1 つ) 以外は、どの AWS サービス API コールも実行できないようにします。詳細については、「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。

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

    "Statement": [

        {

            "Effect": "Deny",

            "NotAction":[

            "s3:GetBucketPolicy",

            "s3:PutBucketPolicy",

            "s3:DeleteBucketPolicy"

                 ],

            "Resource": "*",

            "Condition": {
 "StringLike": { "aws:PrincipalArn": "arn:aws:iam::*:root" }

            }

        }

    ]
 }
```

多くの場合、どのような管理タスクも、関連する管理者用アクセス許可を持つメンバーアカウントの AWS Identity and Access Management (IAM) ロールで実行することが可能です。このようなロールには、アクティビティの制限、ログ、モニタリングを行えるよう、適切なコントロールが適用されている必要があります。

組織の作成と管理

AWS Organizations コンソールを使用するか、AWS Command Line Interface (AWS CLI) コマンドまたは同等の AWS SDK API オペレーションを実行し、以下のタスクを実行できます。

- 組織を作成します。現在のアカウントを管理アカウントとして組織を作成します。組織内にメンバーアカウントを作成し、他のアカウントを組織に招待します。
- 組織内のすべての機能の有効化。AWS Organizations を使用に当たっては、すべての機能を有効にすることが推奨されます。組織を作成する際、すべての機能を有効にするか、あるいは一括請求用の機能のサブセットのオプションがあります。デフォルトでは、一括請求 (コンソリデーティッドビリング) 機能を含むすべての機能が有効になります。

  すべての機能が有効になっていると、サービスコントロールポリシー (SCP) などの高度なアカウント管理機能が AWS Organizations で利用できます。SCP を利用することで、組織内のすべてのアカウントを対象に、使用可能なアクセス許可の範囲を一元管理できるため、組織のアクセスコントロールガイドラインへの準拠をアカウント全体で徹底できるようになります。
- 組織の詳細を表示します。組織、そのルート、組織単位 (OU)、およびアカウントに関する詳細を表示します。
- 組織を削除します 。不要になった組織を削除します。

> **Note**
>
> このセクションのステップでは、タスクを実行するために必要な最小限のアクセス権限を指定します。これらは通常、API またはコマンドラインツールへのアクセスに適用されます。コンソールでタスクを実行するには、追加の権限が必要な場合があります。例えば、組織内のすべてのユーザーに読み取り専用アクセス許可を与えたり、特定のユーザーが特定のタスクを実行できるよう別のアクセス許可を与えたりすることができます。

組織の作成

を管理アカウント AWS アカウント として で始まる組織を作成できます。組織を作成する際、この組織がすべての機能をサポートする (推奨)、または一括請求機能のみをサポートするかを選択できます。

組織の作成後、管理アカウントから以下の方法で組織にアカウントを追加できます。

- 自動的に組織のメンバーアカウントとして追加される AWS アカウントを別途作成する
- メールアドレスの検証後、組織のメンバーアカウントとして既存の AWS アカウントを招待する

組織を作成する

組織を作成するには、 を使用するか、 AWS Management Console または SDK APIs AWS CLI のいずれかのコマンドを使用します。

> 最小アクセス許可
>
> 現在の で組織を作成するには AWS アカウント、次のアクセス許可が必要です。
>
> - `organizations:CreateOrganization`
> - `iam:CreateServiceLinkedRole`
>
> このアクセス許可は、サービスプリンシパルだけに制限できますorganizations.amazonaws.com。

AWS Management Console

組織を作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. デフォルトでは、組織はすべての機能を有効にして作成されます。ただし、次のいずれかのステップを選択できます。

   - すべての機能が有効な組織を作成するには、概要ページで [Create an organization] (組織を作成する) を選択します。
   - 一括請求 (コンソリデーティッドビリング) 機能のみを持つ組織を作成するには、概要ページの [Create an organization] (組織を作成する) で一括請求 (コンソリデーティッドビリング) 機能を選択してから、[Create an organization] (組織を作成する) を選択します。

   オプションの選択を間違えた場合は、すぐに設定ページに移動して [Delete organization] (組織の削除) を選択し、もう一度やり直してください。

3. 組織が作成され、AWS アカウント ページが表示されます。存在するアカウントは管理アカウントのみで、この時点ではルート組織単位 (OU) に保存されています。

   必要に応じて、Organizations により、管理アカウントに関連付けられたアドレスに検証メールが自動的に送信されます。検証 E メールの受信には時間がかかる場合があります。24 時間以内に E メールアドレスを検証します。詳細については、「E メールアドレスの検証」を参照してください。管理アカウントのメールアドレスを検証しなくても、組織内に新しいアカウントを作成することは可能です。ただし、既存のアカウントを招待するには、その前にメールの検証が完了している必要があります。

   > **Note**
   >
   > 以前にそのアカウントのメールアドレスの検証が行われていた場合は、そのアカウントを使用して組織を作成した際にメールの検証が再度求められることはありません。

AWS CLI & AWS SDKs

以下のコード例は、`CreateOrganization` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates an organization in AWS Organizations.
/// </summary>
public class CreateOrganization
{
```

```
        /// <summary>
        /// Creates an Organizations client object and then uses it to create
        /// a new organization with the default user as the administrator, and
        /// then displays information about the new organization.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var response = await client.CreateOrganizationAsync(new
CreateOrganizationRequest
            {
                FeatureSet = "ALL",
            });

            Organization newOrg = response.Organization;

            Console.WriteLine($"Organization: {newOrg.Id} Main Accoount:
{newOrg.MasterAccountId}");
        }
    }
```

- API の詳細については、「 API リファレンスCreateOrganization」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

例 1: 新しい組織を作成するには

Bill は、アカウント 111111111111 の認証情報を使用して組織を作成したいと考えています。次の例は、このアカウントが新しい組織のマスターアカウントになることを示しています。Bill は機能セットを指定していないため、新しい組織ではデフォルトですべての機能が有効になり、サービスコントロールポリシーがルート上で有効になります。

```
aws organizations create-organization
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
        "Organization": {
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ],
                "MasterAccountId": "111111111111",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/
o-exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "FeatureSet": "ALL",
                "Id": "o-exampleorgid",
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
        }
}
```

例 2: 一括決済機能のみを有効にした新しい組織を作成するには

次の例では、一括決済機能のみをサポートする組織を作成します。

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
        "Organization": {
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [],
                "Id": "o-exampleorgid",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/
o-exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "FeatureSet": "CONSOLIDATED_BILLING"
        }
}
```

詳細については、「AWS Organizations ユーザーガイド」の「Creating an Organization」を参照してください。

- API の詳細については、「 コマンドリファレンスCreateOrganization」の「」を参照してください。 AWS CLI

これで、次のような手順で組織にアカウントを追加できるようになりました。

- AWS アカウント 自動的に組織の一部になる を作成するには、 AWS 「」を参照してください組織にメンバーアカウントを作成する。
- 既存のアカウントを組織に招待するには、「AWS アカウント を組織に招待する」を参照してください。

E メールアドレスの検証

組織を作成したら、アカウントを招待する前に、組織の管理アカウントから提供されたメールアドレスを所有していることを検証する必要があります。

組織を作成するときに、管理アカウントが以前に検証されていない場合、 は指定された E メールアドレスに検証 E メール AWS を自動的に送信します。検証 E メールの受信には時間がかかる場合があります。

24 時間以内に E メール内の指示に従って、E メールアドレスを検証します。

24 時間以内に E メールアドレスを検証しない場合は、検証リクエストを再送信して、他の AWS アカウント を組織に招待できます。検証 E メールが受信されない場合には、E メールアドレスが正しいことを確認し、必要に応じて変更します。

- 管理アカウントに関連付けられたメールアドレスを確認するには、管理アカウントを使用した組織の詳細の表示 を参照してください。
- 管理アカウントに関連付けられたメールアドレスを変更するには、AWS Billing ユーザーガイドのAWS アカウントの管理を参照してください。

AWS Management Console

検証リクエストを再送信するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. 設定ページに移動し、[Send verification request] (確認リクエストの送信) を選択します。このオプションは、管理アカウントが検証されていない場合にのみ表示されます。
3. 24 時間以内に E メールアドレスを検証します。

   メールアドレスを検証したら、別の AWS アカウント を組織に招待できます。詳細については、「AWS アカウント を組織に招待する」を参照してください。

管理アカウントのメールアドレスを変更する場合、アカウントは「メール未検証」の状態に戻るため、新しいメールアドレスに対して検証プロセスを実施する必要があります。

> **Note**
>
> 管理アカウントのメールアドレスの変更前に組織への招待をアカウントに送り、その招待が未承諾の場合、管理アカウントの新しいメールアドレスを検証するまでは、招待は承諾可能になりません。前述の手順で、検証リクエストを再送信します。送られてきたメールの内容に沿ってプロセスを完了すると、招待されたアカウントで招待が承諾できるようになります。

組織内のすべての機能の有効化

AWS Organizations には 2 つの機能セットがあります。

- すべての機能 — この機能は を使用するための推奨方法であり AWS Organizations、請求機能の統合が含まれています。組織を作成する際、デフォルトではすべての機能が有効化されています。すべての機能を有効にすると、サポートされている AWS サービスとの統合や組織管理ポリシー AWS Organizations など、 で使用できる高度なアカウント管理機能を使用できます。 ???
- 一括請求 (コンソリデーティッドビリング) 機能 - この機能サブセットはすべての組織でサポートされます。これによって組織内のアカウント管理を一元化するために使用できる基本的な管理ツールが提供されます。

一括請求機能のみを使用して組織を作成する場合、後ですべての機能を有効にできます。このページでは、すべての機能を有効にするプロセスについて説明します。

すべての機能を有効にする前に

一括請求のみをサポートする組織からすべての機能をサポートする機能に変更する前に、以下の点に注意してください。

- すべての機能を有効にするプロセスを開始すると、 AWS Organizations は組織への参加を招待したすべてのメンバーアカウントにリクエストを送信します。すべての招待されたアカウントは、すべての機能を有効にするリクエストを受け入れて承認する必要があります。その場合にのみ、プロセスを完了して組織内のすべての機能を有効にします。アカウントがリクエストを拒否した場合は、組織からアカウントを削除するか、リクエストを再送信する必要があります。すべての機能を有効にするプロセスを完了する前に、リクエストを受け入れる必要があります。 を使用して作成した AWS Organizations アカウントは、追加のコントロールを承認する必要がないため、リクエストを取得しません。
- 組織へのアカウントの招待は、すべての機能が有効になっている状態でも継続できます。招待されたアカウントの所有者には、招待されている組織で有効なのは一括請求 (コンソリデーティッドビリング) だけか、それともすべての機能かが、招待時に通知されます。
  - すべての機能の有効化プロセスの進行中にアカウントを招待した場合、招待には、組織はすべての機能が有効なものとして記載されます。アカウントが招待を承諾する前に、すべての機能の有効化プロセスをキャンセルすると、その招待はキャンセルされます。一括請求 (コンソリデーティッドビリング) 機能のみの組織として、再度招待する必要があります。
  - すべての機能の有効化プロセスの開始前にアカウントを招待し、招待が未承諾の場合、その招待はプロセスの開始時にキャンセルされます。これは、招待には一括請求 (コンソリデーティッドビリング) 機能のみの組織であると記載されているためです。すべての機能が有効になっている組織として、再度招待する必要があります。
- 組織内のアカウントの作成も継続できます。このプロセスがこうした変更の影響を受けることはありません。
- AWS Organizations は、すべてのメンバーアカウントに という名前のサービスにリンクされたロールがあることを確認しますAWSServiceRoleForOrganizations。すべての機能を有効にするには、このロールがすべてのアカウントで必須です。招待したアカウントでこのロールが削除されている場合は、すべての機能を有効にするために招待を承諾するとロールが再作成されます。を使用して作成されたアカウントのロールを削除した場合 AWS Organizations、そのアカウントは特にそのロールを再作成するための招待を受け取ります。組織ですべての機能を有効にする処理を完了するには、これらの招待をすべて受諾する必要があります。

- すべての機能を有効にすると SCP を使用できるようになるため、アカウントの管理者が組織、組織単位、あるいはアカウントに SCP をアタッチすることの影響を理解していることを確認してください。SCP は、影響を受けるアカウントでユーザーだけではなく管理者が実行できる操作を制限することができます。例えば、管理アカウントは、メンバーアカウントが組織を離れるのを禁止する SCP を適用することができます。
- 管理アカウントが SCP の影響を受けることはありません。SCP を適用して、管理アカウントのユーザーとロールの機能を制限することはできません。SCP は、メンバーアカウントのみに影響します。
- 一括請求からすべての機能への移行は一方向です。すべての機能を有効にした組織を、統合された一括請求機能のみに切り替えることはできません。
- (非推奨) 組織で一括請求 (コンソリデーティッドビリング) 機能のみが有効になっている場合、メンバーアカウントの管理者は、`AWSServiceRoleForOrganizations` という名前のサービスにリンクされたロールを削除することができます。ただし、その後に組織ですべての機能を有効にすることにした場合、このロールは必須となります。そのため、招待を承諾してすべての機能の有効化に同意したすべてのアカウントに、このロールが再作成されます。がこのロール AWS Organizations を使用する方法の詳細については、「」を参照してくださいAWS Organizations とサービスにリンクされたロール。

すべての機能を有効にするプロセスの開始

組織の管理アカウントにサインインすると、すべての機能の有効化プロセスを開始できます。そのためには、以下の手順を完了します。

> **最小アクセス許可**
>
> 組織内のすべての機能を有効にするには、次のアクセス権限が必要です。
>
> - `organizations:EnableAllFeatures`
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

招待されたメンバーアカウントに、組織内のすべての機能を有効にすることに同意するよう依頼する

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [設定] ページで、[すべての機能の有効化プロセスの開始] を選択します。
3. [すべての機能の有効化] ページで、[すべての機能の有効化プロセスの開始] を選択して切り替えた後、一括請求機能のみの設定には戻れないことを理解していることを確認します。

   AWS Organizations は、組織内のすべての招待された (作成されていない) アカウントにリクエストを送信し、組織内のすべての機能を有効にするための承認を求めます。を使用して作成されたアカウントがあり AWS Organizations 、メンバーアカウント管理者が という名前のサービスにリンクされたロールを削除した場合`AWSServiceRoleForOrganizations`、はそのアカウントに対してロールの再作成リクエスト AWS Organizations を送信します。

   コンソールには、招待されたアカウントのリクエストの承認ステータスリストが表示されます。

   > **Tip**
   >
   > 後でこのページに戻るには、[Settings] (設定) ページを開き、[Request sent date] (リクエストの送信日) セクションで [View status] (ステータスを表示) を選択します。

4. [Enable all features] (すべての機能の有効化) ページには、組織内の各アカウントに対するリクエストの現在のステータスが表示されます。リクエストに同意したアカウントのステータスは、[ACCEPTED] (承諾済み) と表示されます。まだ同意していないアカウントのステータスは、[OPEN] (オープン) と表示されます。

AWS CLI & AWS SDKs

招待されたメンバーアカウントに、組織内のすべての機能を有効にすることに同意するよう依頼する

組織のすべての機能を有効にするには、次のいずれかのコマンドを使用します。

- AWS CLI: enable-all-features

次のコマンドを実行すると、すべての機能の有効化プロセスが組織で開始されます。

```
$ aws organizations enable-all-features
{
    "Handshake": {
        "Id": "h-79d8f6f114ee4304a5e55397eEXAMPLE",
        "Arn": "arn:aws:organizations::123456789012:handshake/o-aa111bb222/
enable_all_features/h-79d8f6f114ee4304a5e55397eEXAMPLE",
        "Parties": [
            {
                "Id": "a1b2c3d4e5",
                "Type": "ORGANIZATION"
            }
        ],
        "State": "REQUESTED",
        "RequestedTimestamp": "2020-11-19T16:21:46.995000-08:00",
        "ExpirationTimestamp": "2021-02-17T16:21:46.995000-08:00",
        "Action": "ENABLE_ALL_FEATURES",
        "Resources": [
            {
                "Value": "o-a1b2c3d4e5",
                "Type": "ORGANIZATION"
            }
        ]
    }
}
```

出力には、招待されたメンバーアカウントが同意する必要があるハンドシェイクの詳細が表示されます。

- AWS SDKs EnableAllFeatures

メモ

- リクエストがメンバーアカウントに送信されると、90 日のカウントダウンが開始されます。すべてのアカウントは期間内にリクエストを承認する必要があり、承認されない場合リクエストの有効期限が切れます。リクエストの有効期限が切れると、この試行に関連するすべてのリクエストがキャンセルされます。この場合はステップ 2 からやり直す必要があります。

- すべての機能の有効化をリクエストすると、受理されていない既存のアカウントへの招待はキャンセルされます。
- すべての機能の移行プロセス中でも、新しいアカウントの招待を開始したり、新しいアカウントを作成したりすることができます。

すべてのアカウントがリクエストを承認したら、プロセスを終了してすべての機能を有効化できます。組織に招待されたメンバーアカウントがない場合は、プロセスをすぐに終了することもできます。プロセスを終了するには、すべての機能を有効にするプロセスの最終処理 の手順を実施してください。

リクエストを承認してすべての機能を有効にする、またはサービスにリンクされたロールを再作成する

組織の招待されたメンバーアカウントのいずれかにサインインすると、管理アカウントからのリクエストを承認できます。アカウントが元は組織に参加するように招待されたものである場合、招待はすべての機能を有効にするものであり、必要に応じた AWSServiceRoleForOrganizations ロールの再作成の黙示的な承認を含んでいます。アカウントが代わりに を使用して作成 AWS Organizations され、AWSServiceRoleForOrganizationsサービスにリンクされたロールを削除した場合は、ロールを再作成するための招待のみが送信されます。そのためには、以下の手順を完了します。

Important

次のすべての機能を有効にすると、組織の管理アカウントはメンバーアカウントにポリシーベースのコントロールを適用できます。これらのコントロールは、ユーザーはもちろん、管理者として自分がアカウント内で実行できることも制限できます。こうした制限により、アカウントが組織を離れるのを禁止することも可能になります。

最小アクセス許可

メンバーアカウントのすべての機能を有効にするリクエストを承認するには、次のアクセス権限が必要です。

- organizations:AcceptHandshake

- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:ListHandshakesForAccount` - Organizations コンソールを使用する場合にのみ必要
- `iam:CreateServiceLinkedRole` - メンバーアカウントで `AWSServiceRoleForOrganizations` ロールを再作成する必要がある場合にのみ必要

AWS Management Console

組織内のすべての機能を有効にするためのリクエストに同意するには

1. AWS Organizations コンソール AWS Organizations でコンソールにサインインします。メンバーアカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. 組織内のすべての機能のリクエストの受け入れがあなたのアカウントのものであることを確認した後、[Accept] を選択します。このページでは、組織内のすべてのアカウントがリクエストを承認し、管理アカウントの管理者がプロセスを終了するまで、プロセスが未完了であると表示されます。

AWS CLI & AWS SDKs

組織内のすべての機能を有効にするためのリクエストに同意するには

リクエストに同意するには、`"Action": "APPROVE_ALL_FEATURES"` でハンドシェイクを承諾する必要があります。

- AWS CLI:
  - accept-handshake
  - list-handshakes-for-account

  次の例は、アカウントで使用可能なハンドシェイクを一覧表示する方法を示しています。`"Id"` の値 (出力の 4 行目) は、次のコマンドに必要な値です。

```
$ aws organizations list-handshakes-for-account
{
    "Handshakes": [
        {
```

```
            "Id": "h-a2d6ecb7dbdc4540bc788200aEXAMPLE",
            "Arn": "arn:aws:organizations::123456789012:handshake/o-aa111bb222/
approve_all_features/h-a2d6ecb7dbdc4540bc788200aEXAMPLE",
            "Parties": [
                {
                    "Id": "a1b2c3d4e5",
                    "Type": "ORGANIZATION"
                },
                {
                    "Id": "111122223333",
                    "Type": "ACCOUNT"
                }
            ],
            "State": "OPEN",
            "RequestedTimestamp": "2020-11-19T16:35:24.824000-08:00",
            "ExpirationTimestamp": "2021-02-17T16:35:24.035000-08:00",
            "Action": "APPROVE_ALL_FEATURES",
            "Resources": [
                {
                    "Value": "c440da758cab44068cdafc812EXAMPLE",
                    "Type": "PARENT_HANDSHAKE"
                },
                {
                    "Value": "o-aa111bb222",
                    "Type": "ORGANIZATION"
                },
                {
                    "Value": "111122223333",
                    "Type": "ACCOUNT"
                }
            ]
        }
    ]
}
```

次の例では、前のコマンドのハンドシェイク ID を使用して、そのハンドシェイクを受け入れます。

```
$ aws organizations accept-handshake --handshake-id h-
a2d6ecb7dbdc4540bc788200aEXAMPLE
{
    "Handshake": {
        "Id": "h-a2d6ecb7dbdc4540bc788200aEXAMPLE",
```

```
        "Arn": "arn:aws:organizations::123456789012:handshake/o-aa111bb222/
approve_all_features/h-a2d6ecb7dbdc4540bc788200aEXAMPLE",
        "Parties": [
            {
                "Id": "a1b2c3d4e5",
                "Type": "ORGANIZATION"
            },
            {
                "Id": "111122223333",
                "Type": "ACCOUNT"
            }
        ],
        "State": "ACCEPTED",
        "RequestedTimestamp": "2020-11-19T16:35:24.824000-08:00",
        "ExpirationTimestamp": "2021-02-17T16:35:24.035000-08:00",
        "Action": "APPROVE_ALL_FEATURES",
        "Resources": [
            {
                "Value": "c440da758cab44068cdafc812EXAMPLE",
                "Type": "PARENT_HANDSHAKE"
            },
            {
                "Value": "o-aa111bb222",
                "Type": "ORGANIZATION"
            },
            {
                "Value": "111122223333",
                "Type": "ACCOUNT"
            }
        ]
    }
}
```

- AWS SDKs
  - list-handshakes-for-account
  - AcceptHandshake

すべての機能を有効にするプロセスの最終処理

招待されたメンバーアカウントすべてが全機能を有効にするリクエストを承認する必要があります。組織に招待されたメンバーアカウントがない場合、[Enable all features progress] ページにはプロセスを終了できる緑色のバナーが表示されます。

最小アクセス許可

組織のすべての機能を有効にするためのプロセスを終了するには、次のアクセス権限が必要です。

- `organizations:AcceptHandshake`
- `organizations:ListHandshakesForOrganization`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

すべての機能を有効にするプロセスを終了するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. すべての機能の有効化のリクエストがすべての招待されたアカウントによって承認されると、[Settings] (設定) ページの上部に緑色のボックスが表示されます。緑色のボックスで、[Go to finalize] (終了処理に進む) を選択します。
3. [Enable all features] (すべての機能の有効化) ページで [Finalize] (終了する) を選択し、確認ダイアログボックスでもう一度 [Finalize] (終了する) を選択します。
4. 現在、組織は、すべての機能が有効になっています。

AWS CLI & AWS SDKs

すべての機能を有効にするプロセスを終了するには

プロセスを終了するには、`["Action": "ENABLE_ALL_FEATURES"]` でハンドシェイクを承諾する必要があります。

- AWS CLI:
  - list-handshakes-for-organization
  - accept-handshake

```
$ aws organizations list-handshakes-for-organization
```

```
{
    "Handshakes": [
        {
            "Id": "h-43a871103e4c4ee399868fbf2EXAMPLE",
            "Arn": "arn:aws:organizations::123456789012:handshake/o-aa111bb222/
enable_all_features/h-43a871103e4c4ee399868fbf2EXAMPLE",
            "Parties": [
                {
                    "Id": "a1b2c3d4e5",
                    "Type": "ORGANIZATION"
                }
            ],
            "State": "OPEN",
            "RequestedTimestamp": "2020-11-20T08:41:48.047000-08:00",
            "ExpirationTimestamp": "2021-02-18T08:41:48.047000-08:00",
            "Action": "ENABLE_ALL_FEATURES",
            "Resources": [
                {
                    "Value": "o-aa111bb222",
                    "Type": "ORGANIZATION"
                }
            ]
        }
    ]
}
```

次の例は、組織で使用可能なハンドシェイクを一覧表示する方法を示しています。"Id" の値 (出力の 4 行目) は、次のコマンドに必要な値です。

```
$ aws organizations accept-handshake \
    --handshake-id h-43a871103e4c4ee399868fbf2EXAMPLE
{
    "Handshake": {
        "Id": "h-43a871103e4c4ee399868fbf2EXAMPLE",
        "Arn": "arn:aws:organizations::123456789012:handshake/o-aa111bb222/
enable_all_features/h-43a871103e4c4ee399868fbf2EXAMPLE",
        "Parties": [
            {
                "Id": "a1b2c3d4e5",
                "Type": "ORGANIZATION"
            }
        ],
        "State": "ACCEPTED",
```

```
        "RequestedTimestamp": "2020-11-20T08:41:48.047000-08:00",
        "ExpirationTimestamp": "2021-02-18T08:41:48.047000-08:00",
        "Action": "ENABLE_ALL_FEATURES",
        "Resources": [
            {
                "Value": "o-aa111bb222",
                "Type": "ORGANIZATION"
            }
        ]
    }
}
```

- AWS SDKs
  - ListHandshakesForOrganization
  - AcceptHandshake

次のステップ:

- 使用するポリシータイプを有効にします。その後、ポリシーをアタッチして組織内のアカウントを管理することができます。詳細については、「でのポリシーの管理 AWS Organizations」を参照してください。
- サポートされているサービスとの統合を有効にします。詳細については、「AWS Organizations を他のAWSサービスと併用する」を参照してください。

組織に関する詳細の表示

組織の要素の詳細を表示するには、次のタスクを実行します。

トピック

- 管理アカウントを使用した組織の詳細の表示
- ルートコンテナの詳細の表示
- OU の詳細の表示
- アカウントの詳細の表示
- ポリシーの詳細の表示

管理アカウントを使用した組織の詳細の表示

AWS Organizations コンソールで組織の管理アカウントにサインインすると、組織の詳細を表示できます。

> **最小アクセス許可**
>
> 組織の詳細を表示するには、次のアクセス権限が必要です。
>
> - `organizations:DescribeOrganization`

AWS Management Console

組織の詳細を表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. 設定ページに移動します。このページには、組織 ID や、組織の管理アカウントに割り当てられているアカウント名と電子メールアドレスなどの組織の詳細が表示されます。

AWS CLI & AWS SDKs

組織の詳細を表示するには

組織の詳細を表示するには、次のいずれかのコマンドを使用します。

- AWS CLI: describe-organization

  次の例は、このコマンドの出力に含まれる情報を示しています。

```
$ aws organizations describe-organization
{
    "Organization": {
        "Id": "o-aa111bb222",
        "Arn": "arn:aws:organizations::123456789012:organization/o-aa111bb222",
        "FeatureSet": "ALL",
        "MasterAccountArn": "arn:aws:organizations::128716708097:account/o-
aa111bb222/123456789012",
        "MasterAccountId": "123456789012",
```

```
        "MasterAccountEmail": "admin@example.com",
        "AvailablePolicyTypes": [ ...DEPRECATED - DO NOT USE... ]
    }
}
```

> **Important**
>
> AvailablePolicyTypes フィールドは非推奨であり、組織で有効になっているポリシーに関する正確な情報が含まれません。組織で実際に有効になっているすべてのポリシータイプの正確な一覧を確認するには、ListRoots コマンドを使用します。詳しくは、次のセクションの AWS CLI に関する部分を参照してください。

- AWS SDK: DescribeOrganization

ルートコンテナの詳細の表示

AWS Organizations コンソールで組織の管理アカウントにサインインすると、ルートコンテナの詳細を表示できます。

> **最小アクセス許可**
>
> ルートの詳細を表示するには、次のアクセス許可が必要です。
>
> - organizations:DescribeOrganization (コンソールのみ)
> - organizations:ListRoots

ルートは、組織単位 (OU) の階層の最上位コンテナであり、通常は OU として動作します。ただし、階層の最上位にあるコンテナであるルートへの変更は、組織内にある他のすべての OU と AWS アカウント に影響します。

AWS Management Console

ルートの詳細を表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。

2. AWS アカウント ページに移動し、ルート OU (ラジオボタンではなくその名前) を選択します。
3. ルートの詳細ページが表示され、ルートの詳細を確認できます。

AWS CLI & AWS SDKs

ルートの詳細を表示するには

ルートの詳細を表示するには、次のいずれかのコマンドを使用します。

- AWS CLI: list-roots

  次の例は、組織で現在どのポリシータイプが有効になっているかなど、ルートの詳細情報を取得する方法を示しています。

```
$ aws organizations list-roots
{
    "Roots": [
        {
            "Id": "r-a1b2",
            "Arn": "arn:aws:organizations::123456789012:root/o-aa111bb222/r-a1b2",
            "Name": "Root",
            "PolicyTypes": [
                {
                    "Type": "BACKUP_POLICY",
                    "Status": "ENABLED"
                }
            ]
        }
    ]
}
```

- AWS SDKs: ListRoots

OU の詳細の表示

AWS Organizations コンソールで組織の管理アカウントにサインインすると、組織の OU の詳細を表示できます。

最小アクセス許可

組織単位 (OU) の詳細を表示するには、次のアクセス権限が必要です。

- `organizations:DescribeOrganizationalUnit`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:ListOrganizationsUnitsForParent` - Organizations コンソールを使用する場合にのみ必要
- `organizations:ListRoots` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

OU の詳細を表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページで、調べたい OU の名前 (ラジオボタンではなく) を選択します。目的の OU が別の OU の子 OU である場合は、親 OU の横にある三角形のアイコンを選択して展開すると、次の階層に子 OU が表示されます。目的の OU が表示されるまで繰り返します。

   [Organizational unit details] (組織単位の詳細) ボックスに、OU に関する情報が表示されます。

AWS CLI & AWS SDKs

OU の詳細を表示するには

次のいずれかのコマンドを使用して、OU の詳細を表示できます。

- AWS CLI、AWS SDK
  - list-roots
  - list-children
  - describe-organizational-unit

次の例は、AWS CLI を使用して OU の ID を見つける方法を示しています。OU ID を見つけるには、階層をトラバースします。はじめに `list-roots` コマンドを実行します。次に、`list-children` をまずルートで実行し、その後は子 OU で、目的の OU ID が見つかるまで繰り返し実行します。

```
$ aws organizations list-roots
{
    "Roots": [
        {
            "Id": "r-a1b2",
            "Arn": "arn:aws:organizations::123456789012:root/o-aa111bb222/r-a1b2",
            "Name": "Root",
            "PolicyTypes": []
        }
    ]
}
$ aws organizations list-children --parent-id r-a1b2 --child-type
 ORGANIZATIONAL_UNIT
{
    "Children": [
        {
            "Id": "ou-a1b2-f6g7h111",
            "Type": "ORGANIZATIONAL_UNIT"
        }
    ]
}
```

次の例は、OU の ID を見つけた後に、OU の詳細を取得する方法を示しています。

```
$ aws organizations describe-organizational-unit --organizational-unit-id ou-a1b2-
f6g7h111
{
    "OrganizationalUnit": {
        "Id": "ou-a1b2-f6g7h111",
        "Arn": "arn:aws:organizations::123456789012:ou/o-aa111bb222/ou-a1b2-
f6g7h111",
        "Name": "Production-Apps"
    }
}
```

- AWS SDK:

- ListRoots
- ListChildren
- DescribeOrganizationalUnit

アカウントの詳細の表示

AWS Organizations コンソールで組織の管理アカウントにサインインすると、アカウントの詳細を表示できます。

> **最小アクセス許可**
>
> AWS アカウント の詳細を表示するには、次のアクセス許可が必要です。
>
> - `organizations:DescribeAccount`
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:ListAccounts` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

AWS アカウント の詳細を表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページに移動し、調べたいアカウントの名前 (ラジオボタンではなく) を選択します。目的のアカウントが子 OU である場合は、親 OU の横の三角形のアイコン
   ▶
   を選択して展開し、子 OU を表示します。アカウントが見つかるまで繰り返します。

   [Account details] (アカウントの詳細) ボックスに、アカウントに関する情報が表示されます。

AWS CLI & AWS SDKs

AWS アカウント の詳細を表示するには

次のいずれかのコマンドを使用して、アカウントの詳細を表示できます。

- AWS CLI:
  - list-accounts - 組織のすべてのアカウントの詳細を一覧表示します。
  - describe-account - 指定したアカウントの詳細のみを一覧表示します。

  どちらのコマンドでも、レスポンスに含まれる詳細情報は同じです。

  次の例は、指定したアカウントの詳細を取得する方法を示しています。

```
$ aws organizations describe-account --account-id 123456789012

{
    "Account": {
        "Id": "123456789012",
        "Arn": "arn:aws:organizations::123456789012:account/o-
aa111bb222/123456789012",
        "Email": "admin@example.com",
        "Name": "Example.com Organization's Management Account",
        "Status": "ACTIVE",
        "JoinedMethod": "INVITED",
        "JoinedTimestamp": "2020-11-20T09:04:20.346000-08:00"
    }
}
```

- AWS SDK:
  - ListAccounts
  - DescribeAccount

ポリシーの詳細の表示

AWS Organizations コンソールで組織の管理アカウントにサインインすると、ポリシーの詳細を表示できます。

> **最小アクセス許可**
>
> ポリシーの詳細を表示するには、次のアクセス権限が必要です。
>
> - `organizations:DescribePolicy`

- organizations:ListPolicies

AWS Management Console

ポリシーの詳細を表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. 次のいずれかを実行します。
   - ポリシーページに移動し、調べたいポリシーのポリシータイプを選択します。
   - AWS アカウント ページに移動し、ポリシーがアタッチされている OU またはアカウントに移動します。最後に、[Policies] (ポリシー) タブを選択し、アタッチされているポリシーの一覧を表示します。
3. ポリシーの名前 (ラジオボタンではなく) を選択します。

   [Details] (詳細) ページには、JSON ポリシーテキスト、ポリシーがアタッチされている OU とアカウントの一覧など、ポリシーに関するすべての情報が表示されます。

AWS CLI & AWS SDKs

ポリシーの詳細を表示するには

ポリシーの詳細を表示するには、次のいずれかのコマンドを使用します。

- AWS CLI:
  - list-policies
  - describe-policy - 指定したポリシーの詳細のみを一覧表示します。

  次の例は、調べたいポリシーのポリシー ID を見つける方法を示しています。ポリシータイプは指定する必要があります。このコマンドは、そのタイプに一致するすべてのポリシーを返します。

```
$ aws organizations list-policies --filter BACKUP_POLICY
{
    "Policies": [
        {
```

```
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "test-backup-policy",
            "Description": "test-policy-description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        }
    ]
}
```

レスポンスには、JSON ポリシードキュメントを除くすべての詳細情報が含まれます。

次の例は、指定したポリシーの詳細のみを取得する方法を示しています。これには JSON ポリシードキュメントも含まれます。

```
$ aws organizations describe-policy --policy-id p-i9j8k7l6m5
{
    "Policies": [
        {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "test-backup-policy",
            "Description": "test-policy-description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"plans\":{\"My-Backup-Plan\":{\"regions\":{\"@@assign\":
[\"us-west-2\"]},\"rules\":{\"My-Backup-Rule\"
                    :{\"target_backup_vault_name\":{\"@@assign\":\"My-Primary-
Backup-Vault\"}}},\"selections\":{\"tags\":{
                    \"My-Backup-Plan-Resource-Assignment\":{\"iam_role_arn\":
{\"@@assign\":\"arn:aws:iam::$account:role/
                    My-Backup-Role\"},\"tag_key\":{\"@@assign\":\"Stage\"},
\"tag_value\":{\"@@assign\":[\"Production\"]}}}}}}}"
    ]
}
```

- AWS SDK:
  - ListPolicies
  - DescribePolicy

組織の削除

不要になった組織は削除できます。組織を削除しても管理アカウントは閉鎖されません。管理アカウントが組織から削除され、組織自体も削除されます。以前の管理アカウント AWS アカウント は、によって管理されなくなったスタンドアロンアカウントになります AWS Organizations。そうすると、スタンドアロンのアカウントとして引き続き使用、別の組織を作成するために使用、またはメンバーアカウントとしてその組織をアカウントに追加するために他の組織からの招待を受け入れる、といった 3 つのオプションが提示されます。

Important

- 組織を削除した場合、その組織は復元できません。組織内にポリシーを作成していた場合、それらのポリシーも削除され、復元はできません。
- 組織からすべてのメンバーアカウントを消去した場合に限り、組織を削除できます。を使用して一部のメンバーアカウントを作成した場合 AWS Organizations、それらのアカウントの削除がブロックされる可能性があります。スタンドアロンの AWS アカウントとして動作するために必要な情報すべてがある場合に限り、メンバーアカウントを削除できます。その情報を提供してアカウントを削除する方法について詳しくは、メンバーアカウントから組織を退会する を参照してください。
- 組織から削除する前にメンバーアカウントを閉鎖した場合、そのアカウントは一定期間「停止」状態になり、完全に閉鎖されるまでは組織から削除することができません。この処理には最大 90 日かかります。すべてのメンバーアカウントが完全に閉鎖されるまでは、組織を削除できない場合があります。

組織を削除することで組織から管理アカウントを削除すると、そのアカウントは次の影響を受けます。

- このアカウントは独自の料金のみを支払うことになり、他のアカウントの料金を支払う責任はなくなります。
- 他のサービスとの統合が無効になる場合があります。例えば、 AWS IAM Identity Center では組織を運用する必要があるため、IAM Identity Center をサポートする組織からアカウントを削除すると、そのアカウントのユーザーはそのサービスを使用できません。

組織の管理アカウントが SCP (サービスコントロールポリシー) の影響を受けることはないので、SCP を利用できなくなってもアクセス許可が変更されることはありません。

トピック

組織を削除する

以前の管理アカウントを によって管理されなくなったスタンドアロンに戻す組織を削除するには AWS アカウント 、次の手順に従います AWS Organizations。

> **最小アクセス許可**
>
> 組織を削除するには、管理アカウントのユーザーまたはロールとしてサインインし、次の許可を付与される必要があります。
>
> - `organizations:DeleteOrganization`
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

組織を削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. 組織を削除する前に、すべてのアカウントを組織から削除する必要があります。詳細については、「組織からのメンバーアカウントの削除」を参照してください。
3. [Settings] (設定) ページに移動し、[Delete organization] (組織の削除) を選択します。
4. [Delete organization] (組織の削除) 確認ダイアログボックスで、テキストボックスの上の行に表示されている組織の ID を入力します。次に、[Delete organization] (組織の削除) を選択します。

> **Important**
>
> この操作では管理アカウントは閉鎖されず、スタンドアロンの AWS アカウント に戻ります。アカウントを閉鎖するには、組織のメンバーアカウントの閉鎖 の手順に従ってください。

AWS CLI & AWS SDKs

以下のコード例は、DeleteOrganization の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Shows how to delete an existing organization using the AWS
    /// Organizations Service.
    /// </summary>
    public class DeleteOrganization
    {
        /// <summary>
        /// Initializes the Organizations client and then calls
        /// DeleteOrganizationAsync to delete the organization.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var response = await client.DeleteOrganizationAsync(new
DeleteOrganizationRequest());

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Successfully deleted organization.");
            }
            else
            {
```

```
                Console.WriteLine("Could not delete organization.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeleteOrganization」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

組織を削除するには

次の例は、組織を削除する方法を示しています。この操作を実行するには、組織のマスターアカウントの管理者である必要があります。この例では、組織からメンバーアカウント、OU、ポリシーをすべて削除済みであることを前提としています。

```
aws organizations delete-organization
```

- API の詳細については、「 コマンドリファレンスDeleteOrganization」の「」を参照してください。 AWS CLI

組織 AWS アカウント 内の の管理

組織は、一緒に AWS アカウント 管理する のコレクションです。組織の一部であるアカウントを管理するには、次のタスクを実行します。

- 組織のアカウントの詳細を表示します。アカウントの一意の ID 番号、Amazon リソースネーム (ARN)、アタッチされているポリシーを確認できます。
- 組織内のすべての のリストをエクスポート AWS アカウント します。組織内のすべてのアカウントのアカウントの詳細を含む.csv ファイルをダウンロードできます。
- 既存の AWS アカウント を組織 に招待します。招待の作成、作成した招待の管理、招待の承認または拒否を行います。
- 組織 AWS アカウント の一部として を作成します。組織に自動的に属 AWS アカウント する を作成してアクセスします。
- 組織内の代替連絡先を更新します。組織内の AWS アカウントの代替連絡先を更新します。
- 組織 AWS アカウント から を削除します。管理アカウントの管理者は、管理が不要になったメンバーアカウントを組織から削除します。メンバーアカウントの管理者として、組織からアカウントを削除します。管理アカウントによってポリシーがメンバーアカウントにアタッチされている場合は、アカウントを削除できない場合があります。
- AWS アカウントを削除 (または閉鎖) します。が不要になった場合は AWS アカウント、アカウントを閉鎖して、料金の使用や発生を防ぐことができます。

組織への参加に伴う影響

- 組織 AWS アカウントに参加する にはどのような影響がありますか？
- 組織で AWS アカウント作成する にはどのような影響がありますか？

組織 AWS アカウント に参加する への影響

AWS アカウント を組織に招待し、アカウントの所有者が招待を受け入れると、 は新しいメンバーアカウントに対して次の変更 AWS Organizations を自動的に行います。

- AWS Organizations は、 というサービスにリンクされたロールを作成しますAWSServiceRoleForOrganizations。組織がすべての機能をサポートする場合、アカウントにはこのロールが必要です。組織が一括請求機能セットのみをサポートしている場合は、ロール

は削除できます。ロールを削除し、後で組織内のすべての機能を有効にすると、 はアカウントのロール AWS Organizations を再作成します。

- 組織のルートや、アカウントを含む OU には、さまざまなポリシーがアタッチされている場合があります。その場合、これらのポリシーは、招待されたアカウントのすべてのユーザーとロールに即座に適用されます。
- 組織の別のサービスの AWS サービス信頼を有効にできます。これを行うと、その信頼されたサービスは、招待されたアカウントを含め、組織内の任意のメンバーアカウントで、サービスにリンクされた役割を作成したり、アクションを実行したりできます。

> **Note**
>
> 招待されたメンバーアカウントの場合、IAM AWS Organizations ロール を自動的に作成しませんOrganizationAccountAccessRole。このロールは、メンバーアカウントへの管理アクセスを管理アカウントのユーザーに付与します。招待されたアカウントにこのレベルの管理上のコントロールを有効にする場合は、ロールを手動で追加します。詳細については、「招待されたメンバーアカウント OrganizationAccountAccessRole での の作成」を参照してください。

一括請求 (コンソリデーティッドビリング) 機能のみが有効になっている組織に参加するようアカウントを招待できます。後で組織のすべての機能を有効にする場合は、招待されたアカウントが変更を承認する必要があります。

組織で AWS アカウント 作成した への影響

組織 AWS アカウント で を作成すると、 は新しいメンバーアカウントに次の変更 AWS Organizations を自動的に加えます。

- AWS Organizations は、 というサービスにリンクされたロールを作成しますAWSServiceRoleForOrganizations。組織がすべての機能をサポートする場合、アカウントにはこのロールが必要です。組織が一括請求機能セットのみをサポートしている場合は、ロールは削除できます。ロールを削除し、後で組織内のすべての機能を有効にすると、 はアカウントのロール AWS Organizations を再作成します。
- AWS Organizations は IAM ロール を作成しますOrganizationAccountAccessRole。このロールは、新しいメンバーアカウントへのアクセスを管理アカウントに付与します。このロールは削除可能ではあるものの、復旧オプションとして使用できるよう、削除しないでおくことをお勧めします。

- OU ツリーのルートにアタッチされたポリシーがある場合、それらのポリシーは即時、作成されたアカウントのすべてのユーザーおよびロールに適用されます。新しいアカウントは、デフォルトではルート OU に追加されます。
- 組織の別の AWS サービスのサービス信頼を有効にしている場合、その信頼されたサービスは、サービスにリンクされたロールを作成したり、作成したアカウントを含む組織内の任意のメンバーアカウントでアクションを実行したりできます。

AWS アカウント を組織に招待する

組織を作成し、管理アカウントに関連付けられた E メールアドレスを所有していることを確認したら、既存の AWS アカウント を組織に招待できます。

アカウントを招待すると、 AWS Organizations はアカウント所有者に招待を送信します。アカウント所有者は招待を受け入れるか拒否するかを決定します。 AWS Organizations コンソールを使用して、他のアカウントに送信する招待を開始および管理できます。別のアカウントへの招待は、組織の管理アカウントからのみ送信できます。

> **Note**
>
> すべてのアカウントの請求履歴とレポートは、組織内の支払者アカウントに残ります。アカウントを新しい Organization に移動する前に、保持するメンバーアカウントの請求履歴とレポート履歴をダウンロードしてください。これには、コストと使用状況レポート、詳細な請求レポート、または Cost Explorer サービスによって生成されたレポートが含まれることがあります。

の管理者である場合は AWS アカウント、組織からの招待を承諾または拒否することもできます。受諾する場合は、アカウントがその組織のメンバーになります。アカウントが参加できる組織は 1 つのみであるため、複数の招待を受信した場合は、1 つのみ受諾できます。

アカウントが組織への招待を受け入れた時点で、組織の管理アカウントは、新しいメンバーアカウントに発生するすべての課金に対して責任を負います。そのメンバーアカウントにそれまで適用されていた支払い方法は使用されなくなります。代わりに、メンバーアカウントに発生したすべての課金の支払いは、組織の管理アカウントに適用されている支払い方法に基づいて行われます。

招待されたアカウントが組織に参加し、組織がすべての機能モードの場合、管理アカウントは招待されたメンバーアカウントへの完全な管理アクセスと制御ができます。ただし、作成されたアカウントとは異なり、IAM `OrganizationAccountAccessRole` ロールは、管理アカウントが引き受けるア

クセス許可を持つメンバーアカウントに自動的には作成されません。招待されたアカウントがメンバーになった後にこれを作成して設定するには、ステップ に従います招待されたメンバーアカウント OrganizationAccountAccessRole での の作成。

> **Note**
>
> 既存のアカウントを招待するのではなく、組織内にアカウントを作成すると、 によって AWS Organizations 自動的に IAM ロール (`OrganizationAccountAccessRole`デフォルトでは という名前) が作成されます。このロールを使用して、管理アカウントのユーザーに、作成されたアカウントへのアクセス権を付与できます。

AWS Organizations は、 AWS Organizations と他の AWS のサービスの統合をサポートするために、招待されたメンバーアカウントにサービスにリンクされたロールを自動的に作成します。詳細については、「AWS Organizations とサービスにリンクされたロール」を参照してください。

1 日に送信できる招待の数については、最大値および最小値 を参照してください。承諾済みの招待は、このクォータに対してカウントされません。1 つの招待が承諾されるとすぐ、同じ日に別の招待を送信できます。各招待は、15 日以内に応答する必要があります。応答しない場合は期限切れとなります。

アカウントへ送られた招待は、組織内のアカウントのクォータに対してカウントされます。招待が辞退された場合や、管理アカウントによってキャンセルされた場合、または招待の有効期限が切れた場合、その招待はカウントから差し引かれます。

組織の一部としてアカウントを作成するには、「組織にメンバーアカウントを作成する」を参照してください。

> **Important**
>
> 請求上の制約により、 を招待できるのは、管理アカウントと同じ AWS 販売者 (AWS インドの場合) と AWS パーティションから AWS アカウント のみです。
>
> - 組織の管理アカウントが Amazon Web Services India Private Limited (AWS 「インド」) (以前は Amazon Internet Services Private Limited と呼ばれていました) によって作成された場合、組織内のすべてのアカウントは管理アカウントと同じ登録販売者から取得する必要があります。例えば、インド AWS の販売者として、組織に招待できるのは他の AWS India アカウントのみです。India AWS または他の AWS 販売者からのアカウントを組み合わせることはできません。

- 組織内のすべてのアカウントは、管理アカウントと同じ AWS パーティションから取得する必要があります。商用 AWS リージョン パーティションのアカウントは、中国リージョンパーティションのアカウントまたは AWS GovCloud (US) リージョンパーティションのアカウントを持つ組織内に存在することはできません。

AWS アカウントへの招待の送信

組織にアカウントを招待するには、まず管理アカウントに関連付けられたメールアドレスを所有していることを検証する必要があります。詳細については、「E メールアドレスの検証」を参照してください。メールアドレスを検証したら、次のステップを実施してアカウントを組織に招待します。

最小アクセス許可

AWS アカウント を組織に招待するには、次のアクセス許可が必要です。

- `organizations:DescribeOrganization` (コンソールのみ)
- `organizations:InviteAccountToOrganization`

AWS Management Console

別のアカウントを組織に招待するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. で E メールアドレスをすでに検証している場合は AWS、このステップをスキップします。

   まだメールアドレスを検証していない場合には、組織の作成から 24 時間以内に E メールアドレスの検証の手順を実施します。検証メールが届くまでには時間がかかる場合があります。メールアドレスの検証が完了したら、アカウントを組織に招待できます。
3. [AWS アカウント] ページに移動し、[Add an AWS account] (AWS アカウントを追加する) を選択します。
4. [Add an AWS アカウント] (AWS アカウントを追加する) ページで、[Invite an existing AWS account] (既存の AWS アカウントを招待する) を選択します。

5. 既存のページに AWS招待する の E メールアドレスまたはアカウント ID AWS アカウント で招待するには、招待するアカウントに関連付けられている E メールアドレス、またはそのアカウント ID 番号を入力します。
6. (オプション) [Message to include in the invitation email message] (招待メールに含めるメッセージ) で、招待するアカウントの所有者に送る、メールによる招待に含める文章を入力します。
7. (オプション) [Add tags] (タグの追加) セクションで、招待がアカウント管理者によって承諾されたときに自動的にアカウントに適用する 1 つ以上のタグを指定します。これを行うには、[Add tag] (タグの追加) を選択してから、キーとオプションの値を入力します。値を空白のままにすると、空の文字列が設定され、null にはなりません。1 つの AWS アカウントに最大 50 個のタグをアタッチできます。
8. [Send invitation] (招待の送信) を選択します。

> **Important**
>
> 組織内のアカウントのクォータを超過した、または組織がまだ初期化中であるためアカウントを追加できないというメッセージが表示された場合は、AWS Support までお問い合わせください。

9. コンソールは自動的に [Invitations] (招待) ページに移動します。このページで、保留中または承諾済みのすべての招待を確認できます。作成した招待がリストの上部に表示され、ステータスが OPEN に設定されます。

   AWS Organizations は、組織に招待したアカウントの所有者の E メールアドレスに招待を送信します。この E メールメッセージには、アカウント所有者が詳細を表示し、招待を承諾または辞退することを選択できる AWS Organizations コンソールへのリンクが含まれています。または、招待されたアカウントの所有者は、E メールメッセージをバイパスし、 AWS Organizations コンソールに直接移動して招待を表示し、承諾または拒否できます。

   このアカウントへの招待は、組織内で保持できるアカウントの上限数に対してすぐにカウントされます。 AWS Organizations は、アカウントが招待を承諾するまで待ちません。招待が辞退された場合は、管理アカウントはその招待をキャンセルします。招待されたアカウントが指定された期間内に応答しない場合、招待は期限切れになります。どちらの場合も、招待はクォータに対してカウントされなくなります。

AWS CLI & AWS SDKs

別のアカウントを組織に招待するには

他のアカウントを参加するよう組織を招待するには、次のいずれかのコマンドを使用します。

- AWS CLI: invite-account-to-organization

```
$ aws organizations invite-account-to-organization \
    --target '{"Type": "EMAIL", "Id": "juan@example.com"}' \
    --notes "This is a request for Juan's account to join Bill's organization."
{
    "Handshake": {
        "Action": "INVITE",
        "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/
invite/h-examplehandshakeid111",
        "ExpirationTimestamp": 1482952459.257,
        "Id": "h-examplehandshakeid111",
        "Parties": [
            {
                "Id": "o-exampleorgid",
                 "Type": "ORGANIZATION"
            },
            {
                 "Id": "juan@example.com",
                 "Type": "EMAIL"
            }
        ],
        "RequestedTimestamp": 1481656459.257,
        "Resources": [
            {
                "Resources": [
                    {
                        "Type": "MASTER_EMAIL",
                        "Value": "bill@amazon.com"
                    },
                    {
                         "Type": "MASTER_NAME",
                         "Value": "Management Account"
                    },
                    {
                         "Type": "ORGANIZATION_FEATURE_SET",
                         "Value": "FULL"
                    }
```

```
                ],
                "Type": "ORGANIZATION",
                "Value": "o-exampleorgid"
            },
            {
                "Type": "EMAIL",
                "Value": "juan@example.com"
            }
        ],
        "State": "OPEN"
    }
}
```

- AWS SDKs InviteAccountToOrganization

組織の保留中の招待の管理

管理アカウントにサインインすると、組織にリンクされているすべての AWS アカウント の表示と、保留中 (オープン) の招待のキャンセルが可能です。そのためには、以下の手順を完了します。

> **最小アクセス許可**
>
> 組織の保留中の招待を管理するには、次のアクセス権限が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:ListHandshakesForOrganization`
> - `organizations:CancelHandshake`

AWS Management Console

組織から別のアカウントへ送信される招待を表示またはキャンセルするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [Invitations] (招待) ページに移動します。

   組織から送信されたすべての招待と、各招待の現在のステータスが表示されます。

> **Note**
>
> 承諾済み、キャンセル、および拒否された招待は、30 日間リストに表示され続けます。その後は削除され、リストに表示されなくなります。

3. キャンセルする招待の横にあるラジオボタン

を選択してから、[Cancel invitation] (招待をキャンセルする) を選択します。ラジオボタンが灰色で表示されている場合、その招待はキャンセルできません。

招待のステータスが [OPEN] (オープン) から [CANCELED] (キャンセル済み) に変更されます。

AWS は、招待をキャンセルしたことを示す E メールメッセージをアカウント所有者に送信します。新しく招待を送信しない限り、対象アカウントで組織に参加することはできません。

AWS CLI & AWS SDKs

組織から別のアカウントへ送信される招待を表示またはキャンセルするには

招待を表示またはキャンセルするには、次のコマンドを使用します。

- AWS CLI: list-handshakes-for-organization、cancel-handshake
- 以下は、組織から他のアカウントに送信された招待の例です。

```
$ aws organizations list-handshakes-for-organization
{
    "Handshakes": [
        {
            "Action": "INVITE",
            "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/
invite/h-examplehandshakeid111",
            "ExpirationTimestamp": 1482952459.257,
            "Id": "h-examplehandshakeid111",
            "Parties": [
                {
                    "Id": "o-exampleorgid",
                    "Type": "ORGANIZATION"
                },
```

```
                {
                    "Id": "juan@example.com",
                    "Type": "EMAIL"
                }
            ],
            "RequestedTimestamp": 1481656459.257,
            "Resources": [
                {
                    "Resources": [
                        {
                            "Type": "MASTER_EMAIL",
                            "Value": "bill@amazon.com"
                        },
                        {
                            "Type": "MASTER_NAME",
                            "Value": "Management Account"
                        },
                        {
                            "Type": "ORGANIZATION_FEATURE_SET",
                            "Value": "FULL"
                        }
                    ],
                    "Type": "ORGANIZATION",
                    "Value": "o-exampleorgid"
                },
                {
                    "Type": "EMAIL",
                    "Value": "juan@example.com"
                },
                {
                    "Type":"NOTES",
                    "Value":"This is an invitation to Juan's account to join
 Bill's organization."
                }
            ],
            "State": "OPEN"
        },
        {
            "Action": "INVITE",
            "State":"ACCEPTED",
            "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/
invite/h-examplehandshakeid111",
            "ExpirationTimestamp": 1.471797437427E9,
            "Id": "h-examplehandshakeid222",
```

```
            "Parties": [
                {
                    "Id": "o-exampleorgid",
                    "Type": "ORGANIZATION"
                },
                {
                    "Id": "anika@example.com",
                    "Type": "EMAIL"
                }
            ],
            "RequestedTimestamp": 1.469205437427E9,
            "Resources": [
                {
                    "Resources": [
                        {
                            "Type":"MASTER_EMAIL",
                             "Value":"bill@example.com"
                        },
                        {
                            "Type":"MASTER_NAME",
                            "Value":"Management Account"
                        }
                    ],
                    "Type":"ORGANIZATION",
                    "Value":"o-exampleorgid"
                },
                {
                    "Type":"EMAIL",
                    "Value":"anika@example.com"
                },
                {
                    "Type":"NOTES",
                    "Value":"This is an invitation to Anika's account to join
 Bill's organization."
                }
            ]
        }
    ]
}
```

次の例は、アカウントへの招待をキャンセルする方法を示しています。

```
$ aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

```
{
    "Handshake": {
        "Id": "h-examplehandshakeid111",
        "State":"CANCELED",
        "Action": "INVITE",
        "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/
invite/h-examplehandshakeid111",
        "Parties": [
            {
                "Id": "o-exampleorgid",
                "Type": "ORGANIZATION"
            },
            {
                "Id": "susan@example.com",
                "Type": "EMAIL"
            }
        ],
        "Resources": [
            {
                "Type": "ORGANIZATION",
                "Value": "o-exampleorgid",
                "Resources": [
                    {
                        "Type": "MASTER_EMAIL",
                        "Value": "bill@example.com"
                    },
                    {
                        "Type": "MASTER_NAME",
                        "Value": "Management Account"
                    },
                    {
                        "Type": "ORGANIZATION_FEATURE_SET",
                        "Value": "CONSOLIDATED_BILLING"
                    }
                ]
            },
            {
                "Type": "EMAIL",
                "Value": "anika@example.com"
            },
            {
                "Type": "NOTES",
                "Value": "This is a request for Susan's account to join Bob's
 organization."
```

```
            }
        ],
        "RequestedTimestamp": 1.47008383521E9,
        "ExpirationTimestamp": 1.47137983521E9
    }
}
```

- AWS SDKs ListHandshakesForOrganization、 CancelHandshake

組織からの招待の承認または拒否

は、組織への参加の招待を受け取る AWS アカウント 場合があります。招待を承認または拒否することができます。そのためには、以下の手順を完了します。

> **Note**
>
> 組織に対するアカウントのステータスは、表示できるコストと使用状況のデータに影響します。
>
> - メンバーアカウントが組織を離れてスタンドアロンアカウントになると、そのアカウントは組織のメンバーであったときのコストと使用状況のデータにアクセスできなくなります。アカウントがアクセスできるのは、スタンドアロンアカウントとして生成されたデータのみです。
> - メンバーアカウントが組織 A を離れ組織 B に参加すると、そのアカウントは組織 A のメンバーであったときのコストと使用状況のデータにアクセスできなくなります。アカウントがアクセスできるのは、組織 B のメンバーとして生成されたデータのみです。
> - アカウントが以前に属していた組織に再参加すると、そのアカウントはコストと使用状況データの履歴へのアクセスを再び許可されます。

> **Note**
>
> メンバーアカウントおよびスタンドアロンアカウントのみが、組織への招待を承諾または拒否することができます。メンバーアカウントに招待が送信されたら、そのアカウントは現在の組織を離れてから招待を承諾する必要があります。既に AWS 組織に参加している管理アカウントに招待状が送信された場合、そのアカウントは、組織からすべてのメンバーアカウントを削除 し、組織を削除 しない限り、招待状を受け入れることができません。

最小アクセス許可

AWS 組織への招待を承諾または辞退するには、次のアクセス許可が必要です。

- `organizations:ListHandshakesForAccount` – AWS Organizations コンソールで招待のリストを表示するために必要です。
- `organizations:AcceptHandshake`.
- `organizations:DeclineHandshake`.
- `iam:CreateServiceLinkedRole` – 招待を受け入れるために、他の サービスとの統合をサポートするために、メンバーアカウントで AWS サービスにリンクされたロールを作成する必要がある場合にのみ必要です。詳細については、「AWS Organizations とサービスにリンクされたロール」を参照してください。

AWS Management Console

招待を承諾または辞退するには

1. 組織への招待は、アカウント所有者の Eメールアドレスに送信されます。アカウント所有者が招待メールを受け取ったら、そのメール内の手順に従うか、ブラウザで AWS Organizations コンソールに移動し、[Invitations] (招待) を選択します。または、[member account's Invitation] (メンバーアカウントの招待) ページに直接移動することもできます。
2. プロンプトが表示されたら、招待されたアカウントに IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、アカウントのルートユーザー (推奨されません) としてサインインします。
3. メンバーアカウントの招待ページには、保留中の組織への招待が表示されます。

   必要に応じて [Accept invitation] (招待を承諾) または [Decline invitation] (招待を辞退) を選択します。

   - 前のステップで [Accept invitation] (招待を承諾) を選択した場合、コンソールは自動的に組織の概要ページにリダイレクトされます。このページで、メンバーアカウントになることを承諾した組織についての詳細を確認できます。組織の ID および所有者の Eメールアドレスが表示されます。

> **Note**
>
> 承諾済みの招待は、30 日間リストに表示され続けます。その後は削除されリストに表示されなくなります。

AWS Organizations は、 AWS Organizations と他の のサービスの統合をサポートするために、新しいメンバーアカウントに AWS サービスにリンクされたロールを自動的に作成します。詳細については、「AWS Organizations とサービスにリンクされたロール」を参照してください。

AWS は、招待を承諾したことを示す E メールメッセージを組織の管理アカウントの所有者に送信します。また、メンバーアカウントの所有者に、組織のメンバーになったことがメールで通知されます。

- 前述のステップで辞退を選択した場合、アカウントは、その他の保留中の招待が表示されているメンバーアカウントの招待ページにそのまま表示され続けます。

  AWS は、招待を拒否したことを示す E メールメッセージを組織の管理アカウントの所有者に送信します。

> **Note**
>
> 拒否された招待は、30 日間リストに表示され続けます。その後は削除されリストに表示されなくなります。

AWS CLI & AWS SDKs

招待を承諾または辞退するには

招待を承諾または拒否するには、次のコマンドを使用します。

- AWS CLI: accept-handshake、decline-handshake

  次の例は、組織への招待を承諾する方法を示しています。

```
$ aws organizations accept-handshake --handshake-id h-examplehandshakeid111
{
    "Handshake": {
```

```
        "Action": "INVITE",
        "Arn": "arn:aws:organizations::111111111111:handshake/o-exampleorgid/
invite/h-examplehandshakeid111",
        "RequestedTimestamp": 1481656459.257,
        "ExpirationTimestamp": 1482952459.257,
        "Id": "h-examplehandshakeid111",
        "Parties": [
            {
                "Id": "o-exampleorgid",
                "Type": "ORGANIZATION"
            },
            {
                "Id": "juan@example.com",
                "Type": "EMAIL"
            }
        ],
        "Resources": [
            {
                "Resources": [
                    {
                        "Type": "MASTER_EMAIL",
                        "Value": "bill@amazon.com"
                    },
                    {
                        "Type": "MASTER_NAME",
                        "Value": "Management Account"
                    },
                    {
                        "Type": "ORGANIZATION_FEATURE_SET",
                         "Value": "ALL"
                    }
                ],
                "Type": "ORGANIZATION",
                "Value": "o-exampleorgid"
            },
            {
                "Type": "EMAIL",
                "Value": "juan@example.com"
            }
        ],
        "State": "ACCEPTED"
    }
}
```

次の例は、組織への招待を辞退する方法を示しています。

- AWS SDKs AcceptHandshake、 DeclineHandshake

組織にメンバーアカウントを作成する

組織は、一元管理 AWS アカウント される のコレクションです。このページでは、 で組織 AWS アカウント 内で を作成する方法について説明します AWS Organizations。単一の の作成については AWS アカウント、「 入門リソースセンター」を参照してください。

組織の一部であるアカウントを管理するには、次の手順を実行します。

- 組織の一部 AWS アカウント である の作成
- 管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス

メンバーアカウントを作成する前の考慮事項

Organizations は、メンバーアカウントの IAM ロールを自動的に作成します。

組織内にメンバーアカウントを作成すると、Organizations は`OrganizationAccountAccessRole`メンバーアカウントに AWS Identity and Access Management (IAM) ロールを自動的に作成します。これにより、管理アカウントのユーザーとロールは、メンバーアカウントに対して完全な管理制御を実行できます。同じ管理ポリシーにアタッチされた追加のアカウントは、ポリシーが更新されるたびに自動的に更新されます。このロールは、メンバーアカウントに適用されるすべてのサービスコントロールポリシー (SCP) の対象となります。

Organizations は、メンバーアカウントのサービスにリンクされたロールを自動的に作成します。

組織内にメンバーアカウントを作成すると、Organizations は自動的にメンバーアカウント`AWSServiceRoleForOrganizations`内にサービスにリンクされたロールを自動的に作成し、一部の AWS サービスとの統合を可能にします。統合を許可するように他のサービスを設定する必要があります。詳細については、「AWS Organizations とサービスにリンクされたロール」を参照してください。

メンバーアカウントは、スタンドアロンアカウントとして動作するために追加情報を要求できます

AWS は、メンバーアカウントがスタンドアロンアカウントとして動作するために必要なすべての情報を自動的に収集するわけではありません。組織からメンバーアカウントを削除してスタンドアロン

アカウントにするには、削除する前に、そのアカウントの情報を入力する必要があります。詳細については、「メンバーアカウントから組織を退会する」を参照してください。

メンバーアカウントは組織のルートでのみ作成されます。

組織のメンバーアカウントは、組織のルートでのみ作成でき、他の組織単位 (OUsでは作成できません。組織のメンバーアカウントのルートを作成したら、OUs 間で移動できます。詳細については、「アカウントの OU への移動と、ルートと OU 間の移動」を参照してください。

によって管理される組織のメンバーアカウントは、 で作成 AWS Control Tower する必要があります AWS Control Tower

組織が によって管理されている場合は AWS Control Tower、 AWS Control Tower コンソールの AWS Control Tower Account Factory または AWS Control Tower APIsを使用してメンバーアカウントを作成します。組織が によって管理されているときに Organizations でメンバーアカウントを作成した場合 AWS Control Tower、そのアカウントは に登録されません AWS Control Tower。詳細については、AWS Control Tower ユーザーガイドの AWS Control Tower外のリソースを参照するを参照してください。

メンバーアカウントは、マーケティング E メールの受信をオプトインする必要があります

組織の一部として作成したメンバーアカウントは、 AWS マーケティング E メールに自動的にサブスクライブされません。マーケティングメールを受信するようアカウントをオプトインするには、https://pages.awscloud.com/communication-preferences を参照してください。

組織の一部 AWS アカウント である の作成

組織の管理アカウントにサインインすると、自動的に組織に属するよう、メンバーアカウントを作成することができます。次の手順を使用してアカウントを作成すると、 は管理アカウントから新しいメンバーアカウントに次の主要連絡先情報 AWS Organizations を自動的にコピーします。

- 電話番号
- 会社名
- ウェブサイトの URL
- Address

また、管理アカウントから通信言語と Marketplace 情報 (一部の のアカウントのベンダー AWS リージョン) をコピーします。

最小アクセス許可

組織にメンバーアカウントを作成するには、次のアクセス権限が必要です。

- `organizations:CreateAccount`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `iam:CreateServiceLinkedRole` (メンバーアカウントに必要なサービスリンクロールを作成できるようにプリンシパル `organizations.amazonaws.com` に付与されます)。

AWS Management Console

自動的に組織の一部 AWS アカウント となる を作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、[Add an AWS アカウント] (を追加) を選択します。
3. [Add an AWS アカウント] (を追加) ページで、[Create an AWS アカウント] (を作成) を選択します (デフォルトで選択されています)。
4. [Create an AWS アカウント] (を作成) ページの [AWS アカウント name] (名) に、アカウントに割り当てる名前を入力します。この名前は、アカウントを組織内の他のアカウントと区別する際に役立ちます。IAM エイリアスや所有者のメールの名前とは異なります。
5. [Email address of the account's owner] (アカウント所有者のメールアドレス) に、アカウントの所有者のメールアドレスを入力します。この E メールアドレスは、アカウントのルートユーザーのユーザー名認証情報になる AWS アカウント ため、別の E メールアドレスにまだ関連付けることはできません。
6. (オプション) 新しいアカウント内に自動で作成される IAM ロールに割り当てる名前を指定します。このロールは、組織の管理アカウントに、新しく作成されたメンバーアカウントへのアクセス許可を付与します。名前を指定しない場合、 はロールにデフォルトの名前 AWS Organizations を付与します`OrganizationAccountAccessRole`。一貫性を保つため、すべてのアカウントでデフォルトの名前を使用することをお勧めします。

> **Important**
>
> このロールの名前を忘れないでください。後で、管理アカウントのユーザーおよびロールの新しいアカウントにアクセス権限を付与する際に必要になります。

7. (オプション) [タグ] セクションで、[タグの追加] を選択してキーとオプションの値を入力し、新しいアカウントに 1 つ以上のタグを追加します。値を空白のままにすると、空の文字列が設定され、null にはなりません。1 つのアカウントに最大 50 個のタグをアタッチできます。
8. [作成] AWS アカウント を選択します。

   - 組織のアカウントクォータを超えたことを示すエラーが表示された場合は、組織にアカウントを追加しようとすると「クォータを超えました」というメッセージが表示される を参照してください。
   - 組織がまだ初期化中であるため、アカウントを追加できないことを示すエラーが表示された場合は 1 時間待ってから、もう一度試してください。
   - アカウントの作成が成功したかどうかについては、 AWS CloudTrail ログを確認することもできます。詳細については、「での ログ記録とモニタリング AWS Organizations」を参照してください。
   - エラーが引き続き発生する場合は、AWS Support までお問い合わせください。

   AWS アカウント ページが表示され、新しいアカウントがリストに追加されます。
9. これで、管理アカウントのユーザーに管理者アクセスを付与する IAM ロールを持つアカウントができました。このアカウントにアクセスするには、組織のメンバーアカウントへのアクセス のステップを実施します。

> **Note**
>
> アカウントを作成すると、 は AWS Organizations 最初に長い (64 文字) の複雑なランダムに生成されたパスワードをルートユーザーに割り当てます。この初期パスワードを再び取得することはできません。root ユーザーとしてアカウントに初めてアクセスする場合は、パスワード復旧プロセスを行う必要があります。詳細については、「ルートユーザーとしてのメンバーアカウントへのアクセス」を参照してください。

AWS CLI & AWS SDKs

以下のコード例は、CreateAccount の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates a new AWS Organizations account.
/// </summary>
public class CreateAccount
{
    /// <summary>
    /// Initializes an Organizations client object and uses it to create
    /// the new account with the name specified in accountName.
    /// </summary>
    public static async Task Main()
    {
        IAmazonOrganizations client = new AmazonOrganizationsClient();
        var accountName = "ExampleAccount";
        var email = "someone@example.com";

        var request = new CreateAccountRequest
        {
            AccountName = accountName,
            Email = email,
        };

        var response = await client.CreateAccountAsync(request);
        var status = response.CreateAccountStatus;
```

```
            Console.WriteLine($"The staus of {status.AccountName} is
 {status.State}.");
        }
    }
```

- API の詳細については、「 API リファレンスCreateAccount」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

自動的に組織の一部となるメンバーアカウントを作成するには

次の例は、組織のメンバーアカウントを作成する方法を示しています。メンバーアカウントは、「プロダクションアカウント」という名前と E メールアドレス (susan@example.com) で構成されます。roleName パラメータが指定されていない OrganizationAccountAccessRole ため、Organizations は のデフォルト名を使用して IAM ロールを自動的に作成します。roleName また、アカウントの請求データにアクセスするための十分なアクセス許可を持つ IAM ユーザーまたはロールを許可する設定は、 IamUserAccessToBilling パラメータが指定されていないため、デフォルト値の ALLOW に設定されます。Organizations は、スーザンに「ようこそ」という AWS E メールを自動的に送信します。

```
aws organizations create-account --email susan@example.com --account-name
 "Production Account"
```

出力には、ステータスが現在の IN_PROGRESS 状態であることを示すリクエストオブジェクトが含まれます。

```
{
        "CreateAccountStatus": {
                "State": "IN_PROGRESS",
                "Id": "car-examplecreateaccountrequestid111"
        }
}
```

後で、 create-account-request-id パラメータの値として describe-create-account-status コマンドに Id レスポンス値を指定することで、リクエストの現在のステータスをクエリできます。

詳細については、「 Organizations ユーザーガイド AWS 」の「組織でのアカウントの作成」を参照してください。 AWS

- API の詳細については、「 コマンドリファレンスCreateAccount」の「」を参照してください。 AWS CLI

組織のメンバーアカウントへのアクセス

組織でアカウントを作成すると、ルートユーザーに加えて、OrganizationAccountAccessRole というデフォルト名の IAM ロールが AWS Organizations によって自動的に作成されます。名前は作成時に個別に指定できますが、アカウント全体で一貫性のある名前を使用することをお勧めします。このガイドでは、ロールをデフォルト名で表記します。AWS Organizations はこれ以外のユーザーまたはロールを作成することはありません。組織のアカウントにアクセスするには、次のいずれかの方法を使用する必要があります。

- 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。ルートユーザーのセキュリティに関するその他の推奨事項については、「AWS アカウントのルートユーザーのベストプラクティス」を参照してください。
- AWS Organizations の一部として提供されるツールを使用してアカウントを作成する場合、事前に設定済みの OrganizationAccountAccessRole というロールを使用してアカウントにアクセスします。このロールは、この方法で作成されるすべての新しいアカウントに存在します。詳細については、「管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス」を参照してください。
- 既存のアカウントを組織に招待し、そのアカウントによって招待が承諾されると、招待されたメンバーアカウントへのアクセスを管理アカウントに許可する IAM ロールを作成できるようになります。このロールは、AWS Organizations で作成されたアカウントに自動的に追加されるロールと

同一であることを意図しています。このロールの作成については、「招待されたメンバーアカウント OrganizationAccountAccessRole での の作成」を参照してください。ロールの作成が完了したら、「管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス」のステップを使用してアクセスできます。

- AWS IAM Identity Center を使用し、IAM Identity Center の信頼されたアクセスを AWS Organizations で有効にします。これにより、ユーザーは企業の認証情報を使用して AWS アクセスポータルにサインインし、割り当てられている管理アカウントまたはメンバーアカウント内のリソースにアクセスできます。

  詳細については、AWS IAM Identity Center ユーザーガイドのマルチアカウント許可を参照してください。IAM Identity Center への信頼されたアクセス設定については、「AWS IAM Identity Center および AWS Organizations」を参照してください。

> **最小アクセス許可**
>
> 組織の別のアカウントから AWS アカウント にアクセスするには、次のアクセス許可が必要です。
>
> - `sts:AssumeRole` - `Resource` 要素は、アスタリスク (*) に設定するか、新しいメンバーアカウントにアクセスする必要のあるユーザーが含まれるアカウントのアカウント ID 番号に設定する必要があります。

ルートユーザーとしてのメンバーアカウントへのアクセス

新しいアカウントを作成すると、AWS Organizations ではまず、文字長が最低でも 64 文字のパスワードを root ユーザーに割り当てます。すべての文字はランダムに生成され、特定の文字セットが登場する保証もありません。この初期パスワードを再び取得することはできません。root ユーザーとしてアカウントに初めてアクセスする場合は、パスワード復旧プロセスを行う必要があります。詳細については、AWS「 サインインユーザーガイド」の「 のルートユーザーパスワードを忘れてしまったAWS アカウント」を参照してください。

> **メモ**
>
> - ベストプラクティスとして、ルートユーザーは、アクセス許可を制限した他のユーザーやロールの作成にのみ使用し、それ以外のアカウントへのアクセスには使用しないことをお勧めします。復旧プロセスが完了したら、ユーザーまたはロールでサインインします。

- ルートユーザーに多要素認証 (MFA) を有効化することもお勧めします。パスワードをリセットし、root ユーザーに MFA デバイスを割り当てることができます。
- 正しくない E メールアドレスを使用して組織のメンバーアカウントを作成すると、ルートユーザーとしてアカウントにサインインできません。サポートが必要な場合は、AWS Billing and Support にお問い合わせください。

招待されたメンバーアカウント OrganizationAccountAccessRole での の作成

デフォルトでは、組織の一部としてメンバーアカウントを作成すると、そのアカウントに AWS が自動的に作成するロールにより、そのロールを引き受けることができる管理アカウントの IAM ユーザーに、管理者用のアクセス許可が付与されます。デフォルトでは、そのロールの名前は `OrganizationAccountAccessRole` です。詳細については、「管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス」を参照してください。

しかし、組織に招待するメンバーアカウントに、管理者ロールが自動的に作成されることはありません。次の手順に従って、手動で行います。これにより、ロールはコピーされ、作成されたアカウントに自動的に設定されます。一貫性と覚えやすさの点から、手動で作成したロールには、同一の名前 (`OrganizationAccountAccessRole`) を使用されることをお勧めします。

AWS Management Console

メンバーアカウントの AWS Organizations 管理者ロールを作成するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) にサインインします。メンバーアカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。使用するユーザーまたはロールには、IAM ロールとポリシーを作成するアクセス許可が必要です。
2. IAM コンソールで、ロールに移動し、ロールの作成を選択します。
3. を選択しAWS アカウント、次に別の AWS アカウントを選択します。
4. 管理者アクセス権を付与する管理アカウントの 12 桁のアカウント ID 番号を入力します。オプション で、次の点に注意してください。

   - このロールの場合、アカウントは会社内部で使用するため、[Require external ID] (外部 ID が必要) は選択しないでください。外部 ID オプションの詳細については、「IAM ユーザーガイド」の「外部 ID を使用するタイミング」を参照してください。

- MFA が有効化され設定されている場合は、オプションで、MFA デバイスを使用した認証を求めるように設定できます。MFA の詳細については、IAM ユーザーガイドの「 での多要素認証 (MFA) AWS の使用」を参照してください。

5. [次へ] をクリックします。
6. アクセス許可の追加ページで、 という名前のAWSマネージドポリシーAdministratorAccessを選択し、次へを選択します。
7. 名前、レビュー、作成 ページで、ロール名とオプションの説明を指定します。新しいアカウントのロールに割り当てられたデフォルト名との整合性を保つために、OrganizationAccountAccessRole を使用されることをお勧めします。変更をコミットするには、[ロールの作成] を選択します。
8. 新しいロールが、使用可能なロールのリストに表示されます。新しいロールの名前を選択して詳細を表示します。その際、表示されるリンクの URL に注意します。ロールへのアクセスが必要なメンバーアカウントのユーザーにこの URL を通知します。また、ステップ 15 で必要になる [ロール ARN] も書き留めます。
9. IAM コンソール (https://console.aws.amazon.com/iam/) にサインインします。今回は、ポリシーを作成し、そのポリシーをユーザーまたはグループに割り当てるアクセス許可を持つ管理アカウントのユーザーとしてサインインします。
10. ポリシーに移動し、ポリシーの作成を選択します。
11. [Service] で、[STS] を選択します。
12. [Actions] (アクション) で、[Filter] (フィルター) ボックスに「**AssumeRole**」と入力し始め、表示されたら、横のチェックボックスをオンにします。
13. リソース で、特定の が選択されていることを確認してから、ARNs の追加 を選択します。
14. AWS メンバーアカウント ID 番号を入力し、ステップ 1 ~ 8 で作成したロールの名前を入力します。[ARN を追加] を選択します。
15. 複数のメンバーアカウントのロールを引き受けるためのアクセス権限を付与する場合は、アカウントごとにステップ 14 と 15 を繰り返します。
16. [次へ] をクリックします。
17. 確認と作成ページで、新しいポリシーの名前を入力し、ポリシーの作成を選択して変更を保存します。
18. ナビゲーションペインでユーザーグループを選択し、メンバーアカウントの管理を委任するために使用するグループの名前 (チェックボックスではない) を選択します。
19. [アクセス許可] タブを選択します。

20. アクセス許可の追加を選択し、ポリシーのアタッチを選択し、ステップ 11 ~ 18 で作成したポリシーを選択します。

選択したグループのメンバーであるユーザーは、ステップ 9 で取得した URL より、各メンバーアカウントのロールにアクセスできるようになりました。こうしたメンバーアカウントには、組織内に作成したアカウントにアクセスする場合と同様にアクセスすることができます。メンバーアカウントを管理するロールの使用の詳細については、「管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス」を参照してください。

管理アカウントのアクセスロールを持つメンバーアカウントへのアクセス

AWS Organizations コンソールを使用してメンバーアカウントを作成すると、AWS Organizations によって、`OrganizationAccountAccessRole` という名前の IAM ロールがアカウントに自動的に作成されます。このロールには、メンバーアカウントの完全な管理権限が含まれます。このロールのアクセスの範囲には、管理アカウント内のすべてのプリンシパルが含まれます。これにより、ロールは組織の管理アカウントにそのアクセスを許可するように構成されます。招待されたメンバーアカウントと同一のロールを作成するには、「招待されたメンバーアカウント OrganizationAccountAccessRole での の作成」のステップを実行します。このロールを使用してメンバーアカウントにアクセスするには、ロールを引き受けるアクセス許可を持つ管理アカウントのユーザーでサインインする必要があります。このアクセス権限を設定するには、次の手順を行います。メンテナンスしやすいように、アクセス権限は、ユーザーではなくグループに付与することをお勧めします。

AWS Management Console

管理アカウントの IAM グループのメンバーにアクセス許可を付与して、ロールにアクセスするには

1. IAM コンソール (https://console.aws.amazon.com/iam/) に管理者用のアクセス許可を持つ管理アカウントのユーザーとしてサインインします。これは、メンバーアカウントのロールにアクセスするユーザーが属する IAM グループにアクセス許可を委任するために必要です。
2. まず、後で必要になる管理ポリシーを ??? で作成します。

   ナビゲーションペインで、[Policies (ポリシー)] を選択し、[Create Policy (ポリシーの作成)] を選択します。
3. [ビジュアルエディタ] タブで、[サービスの選択] を選択し、検索ボックスに **STS** と入力してリストをフィルタリングし、[STS] オプションを選択します。

4. 「アクション」セクション**assume**で、検索ボックスに「」と入力してリストをフィルタリングし、AssumeRoleオプションを選択します。
5. 「リソース」セクションで「特定の」を選択し、ARNs を追加」を選択し、前のセクションで作成したメンバーアカウント番号とロールの名前を入力します (という名前を付けることをお勧めしますOrganizationAccountAccessRole) 。
6. ダイアログボックスに正しい ARNsたら、ARN を追加 を選択します。
7. (オプション) 多要素認証 (MFA) が必要な場合や、指定された IP アドレス範囲からロールへのアクセスを制限する場合、[リクエスト条件] セクションを展開し、適用するオプションを選択します。
8. [次へ] をクリックします。
9. 確認と作成ページで、新しいポリシーの名前を入力します。例: **GrantAccessToOrganizationAccountAccessRole**。オプションとして説明を追加することもできます。
10. [ポリシーの作成] を選択してポリシーを保存します。
11. ポリシーが使用可能になったので、グループにアタッチできます。

    ナビゲーションペインで、ユーザーグループを選択し、メンバーアカウントでロールを引き受けることができるグループの名前 (チェックボックスではない) を選択します。必要に応じて、新しいグループを作成できます。
12. [アクセス許可] タブを選択し、[アクセス許可の追加] を選択してから、[ポリシーの添付] を選択します。
13. (オプション) [検索] ボックスに、ポリシー名を入力し始めると、「Step 2」から「Step 10」で作成したポリシーの名前が表示されるまで、リストをフィルタリングできます。すべてのタイプの を選択し、カスタマーAWS管理の を選択して、すべての管理ポリシーを除外することもできます。
14. ポリシーの横にあるチェックボックスをオンにし、ポリシーのアタッチ を選択します。

これで、グループのメンバーである IAM ユーザーに、次の手順に従って AWS Organizations コンソールで新しいロールに切り替えられるアクセス許可が付与されました。

AWS Management Console

メンバーのアカウントのロールに切り替えるには

ロールを使用する際、ユーザーは、新しいメンバーアカウントの管理者権限が付与されます。グループのメンバーである IAM ユーザーに、以下を実行して新しいロールに切り替えるように指示します。

1. AWS Organizations コンソールの右上隅で、現在のサインイン名が表示されたリンクをクリックし、[Switch Role] (ロールの切り替え) を選択します。
2. 管理者から提供されたアカウント ID 番号とロール名を入力します。
3. [表示名] で、ロール使用時にユーザー名の代わりに右上隅のナビゲーションバーに表示する文字列を入力します。オプションで色を選択することもできます。
4. [Switch Role] (ロールの切り替え) を選択します。これで、実行するアクションはすべて、切り替えたロールに付与されているアクセス権限で行われるようになりました。切り替えを戻さない限り、元の IAM ユーザーに関連付けられているアクセス許可を使用することはできません。
5. ロールのアクセス許可を必要とするアクションの実行が完了したら、通常の IAM ユーザーにもう一度切り替えることができます。右上隅 (ディスプレイ名として指定したもの) のロール名を選択し、「 に戻る***UserName***」を選択します。

追加リソース

- ロールを切り替えるアクセス許可の付与の詳細については、「IAM ユーザーガイド」の「ロールを切り替えるアクセス許可をユーザーに付与する」を参照してください。
- 引き受けるアクセス許可が付与されたロールの使用の詳細については、IAM ユーザーガイドの「ロールへの切り替え (コンソール)」を参照してください。
- クロスアカウントアクセスにロールを使用する方法のチュートリアルについては、IAM ユーザーガイドの「チュートリアル: 間の IAM ロールAWS アカウントを使用したアクセスの委任」を参照してください。
- AWS アカウント の閉鎖については、組織のメンバーアカウントの閉鎖 を参照して ください。

組織の AWS アカウント の詳細をエクスポートする

AWS Organizations を使用すると、組織の管理アカウントユーザーおよび委任管理者は、組織内のすべてのアカウントの詳細を含む .csv ファイルをエクスポートできます。その結果、組織管理者は、

アカウントを簡単に表示し、ステータス (ACTIVE、SUSPENDED、または PENDING) でフィルターできます。組織に多数のアカウントがある場合、.csv ファイルのダウンロードオプションを使用すると、スプレッドシートのアカウントの詳細の表示と並べ替えを容易に行うことができます。

以前は、アカウントは、AWS Organizations コンソールでアカウント階層またはリスト表示でなければ表示できませんでした。

> **Note**
>
> アカウントリストをダウンロードできるのは、管理アカウントのプリンシパルのみです。

組織のすべての AWS アカウント のリストのエクスポート

組織の管理アカウントにサインインすると、組織に属するすべてのアカウントのリストを .csv ファイルとして取得できます。リストには個々のアカウントの詳細が含まれますが、アカウントが属する組織単位 (OU) は示されません。

.csv ファイルには、各アカウントの次の情報が含まれています。

- アカウント ID - 数値のアカウント識別子。例: 123456789012。
- ARN - アカウントの Amazon リソースネーム。例: `arn:aws:organizations::123456789012account/o-o1gb0d1234/123456789012`
- Email - アカウントに関連付けられている E メールアドレス。例: marymajor@example.com。
- 名前 - アカウント作成者によって提供されたアカウント名。例: stage testing account。
- ステータス - 組織内のアカウントのステータス。値は、PENDING、ACTIVE、または SUSPENDED です。
- 参加方法 - アカウントがどのように作成されたかを示します。値は INVITED または CREATED です。
- 参加タイムスタンプ - アカウントが組織に参加した日付と時刻。

> **最小アクセス許可**
>
> 組織内のすべてのメンバーアカウントを含む .csv ファイルをエクスポートするには、以下のアクセス許可が必要です。
>
> - `organizations:DescribeOrganization`

- `organizations:ListAccounts`

AWS Management Console

組織のすべての AWS アカウントを含む .csv ファイルをエクスポートするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [アクション] を選択し、[AWS アカウント] で [アカウントリストをエクスポート] を選択します。ページ上部の青いバナーは、エクスポートが進行中であることを示します。
3. ファイルの準備ができたら、バナーが緑色に変わり、ダウンロードの準備ができたことを示します。[Download CSV] を選択します。ファイル `Organization_accounts_information.csv` がデバイスにダウンロードされます。

AWS CLI & AWS SDKs

アカウントの詳細を含む .csv ファイルをエクスポートする唯一の方法は、AWS Management Consoleを使用することです。AWS CLI を使用してアカウントリストの .csv ファイルをエクスポートすることはできません。

組織からのメンバーアカウントの削除

組織のアカウントの管理の一環として、不要になったメンバーアカウントを削除します。メンバーアカウントを削除してもアカウントは閉鎖されません。そのメンバーアカウントは組織から削除されます。それまでのメンバーアカウントは AWS Organizations の管理対象から外れ、スタンドアロンの AWS アカウント になります。その後、そのアカウントにはポリシーが適用されなくなり、請求書の支払いはそのアカウントの責任となります。アカウントが組織から削除された後は、そのアカウントで発生した費用については、組織の管理アカウントには請求されなくなります。

管理アカウントの削除については、組織の削除 を参照してください。

トピック

- 組織のアカウントを削除する前に考慮する事項
- 組織からメンバーアカウントを削除する
- メンバーアカウントから組織を退会する

組織のアカウントを削除する前に考慮する事項

アカウントを削除する前に、以下を考慮することが重要です。

- アカウントがスタンドアロンアカウントとして動作するために必要な情報を持っている場合に限り、組織からアカウントを削除できます。AWS Organizations コンソール、API、AWS CLI コマンドを使用して組織内にアカウントを作成しても、スタンドアロンアカウントの必須情報がすべて自動的に収集されるわけではありません。スタンドアロンとして使用する各アカウントについて、サポートプランを選択し、必須の連絡先情報を入力および検証して、現在の支払い方法を入力する必要があります。この支払い方法は、アカウントが組織に関連付けられていない間に発生する課金対象 (AWS 無料利用枠外) の AWS アクティビティに対して課金するために AWS によって使用されます。この情報がないアカウントを削除するには、「メンバーアカウントから組織を退会する」の手順に従ってください。
- 組織内に作成したアカウントを削除するには、アカウントが作成されてから 7 日以上が経過している必要があります。招待されたアカウントの場合には、7 日間待つ必要はありません。
- アカウントが正常に組織を離れた時点で、その AWS アカウント の所有者は、新たに発生するすべての AWS コストと、アカウントに適用される支払い方法についての責任を負います。そのとき以降、組織の管理アカウントが責任を負うことはありません。
- 削除するアカウントは、組織の AWS サービスの代理管理者アカウントであってはなりません。アカウントが代理管理者である場合は、まずその代理管理者アカウントを組織に残る別のアカウントに変更する必要があります。AWS サービスの代理管理者アカウントの無効化または変更の方法について詳しくは、当該サービスのドキュメントを参照してください。
- 組織内から作成されたアカウント (AWS Organizations コンソールまたは `CreateAccount` API を使用して作成されたアカウント) を削除した後も、(i) 作成されたアカウントには、作成元の管理アカウントの契約条項が適用されます。また、(ii) 作成元の管理アカウントは、作成されたアカウントによって行われるいかなるアクションについても連帯責任を負います。お客様と当社間の契約、その契約に基づく権利と義務は、当社が事前に同意しない限り、他に割り当てまたは転移することはできません。当社の同意を得るには、AWS にお問い合わせください。
- メンバーアカウントが組織を離れると、そのアカウントは組織のメンバーであったときのコストと使用状況のデータにアクセスできなくなります。ただし、組織の管理アカウントは引き続きデータにアクセスできます。再度組織に加わった場合、アカウントは再びデータにアクセスできます。
- メンバーアカウントが組織を離れると、そのアカウントにアタッチされていたタグはすべて削除されます。
- メンバーアカウントを組織から削除しても、組織の管理アカウントによるアクセスを有効にするために作成された IAM ロールは自動的に削除されません。以前の組織の管理アカウントからアクセスされないようにするには、その IAM ロールを手動で削除する必要があります。ロールを削除

する方法について詳しくは、IAM ユーザーガイドのロールまたはインスタンスプロファイルの削除を参照してください。

組織からアカウントを消去した場合の影響

組織からアカウントを削除する場合、そのアカウントに直接的な変更は適用されません。ただし、次の間接的な影響があります。

- このアカウントは独自の料金を支払う責任を担い、アカウントにアタッチ済みの有効な支払い方法が必要となります。
- アカウントのプリンシパルは組織で適用されていたポリシーの影響を受けなくなります。つまり、SCP によって課せられていた制限がなくなり、そのアカウントのユーザーとロールに以前より多くのアクセス許可が付与される可能性があります。その他の組織ポリシータイプが適用または処理されることはなくなります。
- ポリシーに `aws:PrincipalOrgID` 条件キーを使用し、組織の AWS アカウント のユーザーとロールだけにアクセスを限定している場合は、そのメンバーアカウントを削除する前にそのポリシーを確認し、必要に応じて更新します。ポリシーを更新していない状態でアカウントが組織を離れると、そのアカウントのユーザーとロールがリソースにアクセスできなくなる可能性があります。
- 他のサービスとの統合が無効になる場合があります。AWS サービスとの統合が有効になっている組織からアカウントを削除すると、アカウントのユーザーはそのサービスを使用できなくなります。

組織からメンバーアカウントを削除する

組織の管理アカウントにサインインすると、不要になったメンバーアカウントを組織から削除できます。これを行うには、以下の手順を完了します。この手順はメンバーアカウントのみに適用されます。管理アカウントを削除するには、組織を削除する必要があります。

> **Note**
>
> メンバーアカウントが組織から削除されると、そのメンバーアカウントは組織契約の対象範囲ではなくなります。管理アカウントの管理者は、メンバーアカウントが必要に応じて新しい契約を用意できるよう、メンバーアカウントを組織から削除する前に、そのことをメンバーアカウントに通達するべきです。有効な組織契約の内容は、AWS Artifact コンソールの [AWS Artifact Organization Agreements] (AWS Artifact 組織契約) ページで確認できます。

最小アクセス許可

組織から 1 つ以上のメンバーアカウントを削除するには、次の許可がある管理アカウントのユーザーまたはロールとしてサインインする必要があります。

- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:RemoveAccountFromOrganization`

ステップ 5 でメンバーアカウントのユーザーまたはロールとしてサインインを選択した場合、そのユーザーまたはロールには次の許可が必要となります。

- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要です。
- `organizations:LeaveOrganization` - 組織の管理者は、このアクセス許可を削除するポリシーをアカウントに適用し、組織からアカウントを削除することを禁止できます。
- IAM ユーザーとしてサインインし、アカウントに支払い情報がない場合、そのユーザーには `aws-portal:ModifyBilling` 許可と `aws-portal:ModifyPaymentMethods` 許可 (アカウントがきめ細かな許可に移行されていない場合)、または `payments:CreatePaymentInstrument` 許可と `payments:UpdatePaymentPreferences` 許可 (アカウントがきめ細かな許可に移行されている場合) のどちらか一方が必要になります。また、メンバーアカウントでは、請求への IAM ユーザーアクセスが有効になっている必要があります。有効になっていない場合は、AWS Billing ユーザーガイドの Billing and Cost Management コンソールへのアクセスを有効にするを参照してください。

AWS Management Console

組織からメンバーアカウントを削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページで、組織から削除するメンバーアカウントを探し、その横にあるチェックボックス
   ☑

をオンにします。OU の階層を移動するか、[View AWS アカウント only] (AWS アカウントだけを表示する) をオンにして OU 構造のないフラットリストでアカウントを表示します。アカウントが多い場合、削除対象をすべて見つけるにはリスト下部の [Load more accounts in 'ou-name'] ('ou-name' のアカウントをさらに読み込む) を選択する必要がある場合があります。

AWS アカウント ページで、組織から削除するメンバーアカウントの名前を探し、選択します。場合によっては、目的のアカウントを見つけるには OU を展開 (▶ を選択) する必要があります。

3. [Actions] (アクション) を選択し、[AWS アカウント] で [Remove from organization] (組織から削除する) を選択します。
4. [Remove account 'account-name' (#account-id-num) from organization?] (アカウント 'account-name' (#account-id-num) を組織から削除しますか?) ダイアログボックスで、[Remove account] (アカウントを削除する) を選択します。
5. AWS Organizations が 1 つ以上のアカウントを削除できなかった場合、その原因は、通常、アカウントがスタンドアロンで動作するのに必要な情報の一部が指定されていないためです。以下のステップを実行します。

   a. 失敗したアカウントにサインインします。メンバーアカウントには、[Copy link] を選択してから、新しい incognito ブラウザウィンドウのアドレスバーに貼り付けてサインインすることをお勧めします。シークレットウィンドウを使用しない場合は、管理アカウントからサインアウトされ、このダイアログボックスに戻ることができなくなります。

   b. ブラウザを使用すると、このアカウントで実行していなかったステップを完了するためのサインアッププロセスが直接表示されます。示されたすべてのステップを実行します。これには次のものが含まれます。

      - 連絡先情報を指定する
      - 有効な支払い方法の指定
      - 電話番号を検証する
      - サポートプランオプションを選択する

   c. 最後のサインアップステップを完了すると、AWS によりブラウザがメンバーアカウントの AWS Organizations コンソールに自動的にリダイレクトされます。[Leave organization] を選択し、確認ダイアログボックスで、その選択を確認します。AWS Organizations コンソールの [Getting Started] ページにリダイレクトされます。そこで、招待が保留中のアカウントを表示して、他の組織に参加することができます。

d. アカウントへのアクセスを付与する IAM ロールを組織から削除します。

> **Important**
>
> 組織で作成されたアカウントの場合、組織の管理アカウントによるアクセスを有効にするための IAM ロールが、Organizations によってアカウントに自動的に作成されています。招待されたアカウントの場合、Organizations によってこのようなロールが自動で作成されることはありませんが、ご自身または別の管理者が同じメリットを得るためにロールを作成している可能性があります。いずれの場合も、組織からアカウントを削除した場合に、このようなロールは自動的に削除されません。以前の組織の管理アカウントからアクセスされないようにするには、この IAM ロールを手動で削除する必要があります。ロールを削除する方法について詳しくは、IAM ユーザーガイドのロールまたはインスタンスプロファイルの削除を参照してください。

AWS CLI & AWS SDKs

組織からメンバーアカウントを削除するには

メンバーアカウントを作成するには、次のいずれかのコマンドを使用します。

- AWS CLI: remove-account-from-organization

```
$ aws organizations remove-account-from-organization \
    --account-id 123456789012
```

このコマンドが成功した場合、出力は生成されません。

- AWS SDK: RemoveAccountFromOrganization

メンバーアカウントが組織から削除されたら、アカウントへのアクセスを付与する IAM ロールを組織から削除します。

> **Important**
>
> 組織で作成されたアカウントの場合、組織の管理アカウントによるアクセスを有効にするための IAM ロールが、Organizations によってアカウントに自動的に作成されています。招待されたアカウントの場合、Organizations によってこのようなロールが自動で作成さ

れることはありませんが、ご自身または別の管理者が同じメリットを得るためにロールを作成している可能性があります。いずれの場合も、組織からアカウントを削除した場合に、このようなロールは自動的に削除されません。以前の組織の管理アカウントからアクセスされないようにするには、この IAM ロールを手動で削除する必要があります。ロールを削除する方法について詳しくは、IAM ユーザーガイドのロールまたはインスタンスプロファイルの削除を参照してください。

メンバーアカウントは、代わりに Leave-Organization を使用して自身のアカウントを削除することができます。詳細については、「メンバーアカウントから組織を退会する」を参照してください。

メンバーアカウントから組織を退会する

メンバーアカウントにサインインすると、そのアカウントを組織から削除できます。これを行うには、以下の手順を完了します。この手順はメンバーアカウントのみに適用されます。管理アカウントは、この方法で組織を離れることはできません。管理アカウントを削除するには、組織を削除する必要があります。

> **Note**
>
> 組織に対するアカウントのステータスは、表示できるコストと使用状況のデータに影響します。
>
> - メンバーアカウントが組織を離れてスタンドアロンアカウントになると、そのアカウントは組織のメンバーであったときのコストと使用状況のデータにアクセスできなくなります。アカウントがアクセスできるのは、スタンドアロンアカウントとして生成されたデータのみです。
> - メンバーアカウントが組織 A を離れ組織 B に参加すると、そのアカウントは組織 A のメンバーであったときのコストと使用状況のデータにアクセスできなくなります。アカウントがアクセスできるのは、組織 B のメンバーとして生成されたデータのみです。
> - アカウントが以前に属していた組織に再参加すると、そのアカウントはコストと使用状況データの履歴へのアクセスを再び許可されます。

> **Important**
>
> 組織を離れると、メンバーアカウントを代表して組織の管理アカウントによって受諾されていた組織契約は適用されなくなります。こうした組織契約の内容は、AWS Artifact コンソールの [AWS Artifact Organization Agreements] (AWS Artifact 組織契約) ページで確認できます。組織を離れる前に、(必要に応じて、法務、プライバシー、またはコンプライアンスチームの支援を得て) 新しい契約を結ぶ必要があるかどうか判断してください。

> **最小アクセス許可**
>
> AWS 組織を登録解除する場合は、次のアクセス権限が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要です。
> - `organizations:LeaveOrganization` - 組織の管理者は、このアクセス許可を削除するポリシーをアカウントに適用し、組織からアカウントを削除することを禁止できます。
> - IAM ユーザーとしてサインインし、アカウントに支払い情報がない場合、そのユーザーには `aws-portal:ModifyBilling` 許可と `aws-portal:ModifyPaymentMethods` 許可 (アカウントがきめ細かな許可に移行されていない場合)、または `payments:CreatePaymentInstrument` 許可と `payments:UpdatePaymentPreferences` 許可 (アカウントがきめ細かな許可に移行されている場合) のどちらか一方が必要になります。また、メンバーアカウントでは、請求への IAM ユーザーアクセスが有効になっている必要があります。有効になっていない場合は、AWS Billing ユーザーガイドの Billing and Cost Management コンソールへのアクセスを有効にするを参照してください。

AWS Management Console

メンバーアカウントから組織を退会するには

1. AWS Organizations コンソールで、AWS Organizations コンソールにサインインします。メンバーアカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。

   デフォルトでは、AWS Organizations を使用して作成されたメンバーアカウントでは root ユーザーのパスワードへのアクセス権はありません。必要に応じて、ルートユーザーとして

のメンバーアカウントへのアクセス の手順に従って root ユーザーのパスワードを回復します。

2. Organizations ダッシュボードページで、[この組織を離れる] を選択します。
3. [組織を離れることを確認する] ダイアログボックスで、[組織を離れる] を選択します。プロンプトが表示されたら、アカウントの削除を確認します。AWS Organizations コンソールの [開始方法] ページにリダイレクトされます。このページでは、他の組織に参加するための保留中の招待が表示されます。

   [まだ組織を離れることはできません] というメッセージが表示される場合、アカウントにはスタンドアロンアカウントとして運営するために必要なすべての情報が揃っていません。この場合は、次のステップに進みます。
4. [組織を離れることを確認する] ダイアログボックスに [まだ組織を離れることはできません] というメッセージが表示される場合、[アカウントのサインアップステップを完了する] というリンクを選択します。
5. [AWS にサインアップする] ページで、スタンドアロンアカウントになるために必要な情報をすべて入力します。これには、以下の種類の情報が含まれる場合があります。

   - 連絡先名と住所
   - 有効な支払い方法
   - 電話番号による確認
   - サポートプランオプション
6. サインアッププロセスが完了したことを示すダイアログボックスが表示されたら、[Leave organization] を選択します。

   確認のダイアログボックスが表示されます。アカウントを削除するには、その選択を確認します。AWS Organizations コンソールの [Getting Started] ページにリダイレクトされます。そこで、招待が保留中のアカウントを表示して、他の組織に参加することができます。
7. アカウントへのアクセスを付与する IAM ロールを組織から削除します。

> **Important**
>
> 組織で作成されたアカウントの場合、組織の管理アカウントによるアクセスを有効にするための IAM ロールが、Organizations によってアカウントに自動的に作成されています。招待されたアカウントの場合、Organizations によってこのようなロールが自動で作成されることはありませんが、ご自身または別の管理者が同じメリットを得るためにロールを作成している可能性があります。いずれの場合も、組織からアカウ

ントを削除した場合に、このようなロールは自動的に削除されません。以前の組織の管理アカウントからアクセスされないようにするには、この IAM ロールを手動で削除する必要があります。ロールを削除する方法について詳しくは、IAM ユーザーガイドのロールまたはインスタンスプロファイルの削除を参照してください。

AWS CLI & AWS SDKs

メンバーアカウントとして組織を離れるには

組織を登録解除するには、次のいずれかのコマンドを使用します。

- AWS CLI: leave-organization

  次の例では、アカウントの認証情報を使用し、組織を離れるコマンドを実行しています。

  ```
  $ aws organizations leave-organization
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS SDK: LeaveOrganization

メンバーアカウントが組織を離れたら、アカウントへのアクセスを付与する IAM ロールを組織から削除します。

> **Important**
>
> 組織で作成されたアカウントの場合、組織の管理アカウントによるアクセスを有効にするための IAM ロールが、Organizations によってアカウントに自動的に作成されています。招待されたアカウントの場合、Organizations によってこのようなロールが自動で作成されることはありませんが、ご自身または別の管理者が同じメリットを得るためにロールを作成している可能性があります。いずれの場合も、組織からアカウントを削除した場合に、このようなロールは自動的に削除されません。以前の組織の管理アカウントからアクセスされないようにするには、この IAM ロールを手動で削除する必要があります。ロールを削除する方法について詳しくは、IAM ユーザーガイドのロールまたはインスタンスプロファイルの削除を参照してください。

代わりの方法として、メンバーアカウントは、管理アカウントのユーザーが remove-account-from-Organization を使用して削除することもできます。詳細については、「組織からメンバーアカウントを削除する」を参照してください。

組織のメンバーアカウントの閉鎖

組織内でメンバーアカウントが不要になった場合は、このセクションの指示に従ってAWS Organizations コンソールからアカウントを閉じることができます。組織がすべての機能モードの場合にのみ、 AWS Organizations コンソールを使用してメンバーアカウントを閉鎖できます。

ルートユーザーとしてサインイン AWS Management Console した後、 のアカウントページ AWS アカウント から直接 を閉じることもできます。 step-by-step 手順については、「 AWS アカウント管理ガイドAWS アカウント」の「 を閉じる」を参照してください。

管理アカウントを解約するには、「」を参照してください組織内の管理アカウントの閉鎖。

メンバーアカウントを閉鎖する方法

組織の管理アカウントにサインインすると、組織に属しているメンバーアカウントを閉鎖できます。そのためには、以下の手順を完了します。

> **Important**
>
> メンバーアカウントを閉鎖する前に、考慮事項を確認し、アカウントを閉鎖した場合の影響を理解しておくことを強くお勧めします。詳細については、「 アカウント管理ガイド」の「 アカウントを閉鎖する前に知っておくべきこと」および「 アカウントを閉鎖した後の予定」を参照してください。 AWS

AWS Management Console

AWS Organizations コンソールからメンバーアカウントを解約するには

1. AWS Organizations コンソール にサインインします。
2. AWS アカウント ページで、閉鎖するメンバーアカウントの名前を探し、選択します。OU の階層を移動するか、OU 構造のないアカウントのフラットリストを表示できます。
3. ページの上部のアカウント名の横にある [Close] (閉じる) をクリックします。一括請求モードの組織は、コンソールで閉じるボタンを表示できません。一括請求モードでアカウント

を解約するには、「アカウントAWS 管理ガイド」の「スタンドアロンアカウント」タブの「アカウントを解約する方法」の手順に従います。

4. 必要なアカウント閉鎖ステートメントをすべて承認するために、各チェックボックスをオンにします。
5. メンバーアカウント ID を入力し、アカウントを閉じる を選択します。

> **Note**
>
> 閉鎖したメンバーアカウントには、 AWS Organizations コンソールのアカウント名の横にSUSPENDEDラベルが表示されます。

アカウントページからメンバーアカウントを閉鎖するには

オプションで、 のアカウントページから AWS メンバーアカウントを直接閉鎖できます AWS Management Console。ガイダンスについては、「 step-by-stepAWS アカウント管理ガイド」の AWS アカウント「 を閉じる」の指示に従ってください。

AWS CLI & AWS SDKs

を閉じるには AWS アカウント

次のいずれかのコマンドを使用して AWS アカウントを閉鎖できます。

- AWS CLI: close-account

```
$ aws organizations close-account \
    --account-id 123456789012
```

このコマンドが成功した場合、出力は生成されません。

- AWS SDKs CloseAccount

メンバーアカウントが閉鎖されないように保護する

メンバーアカウントが誤って閉鎖されないように保護したい場合、IAM ポリシーを作成して、閉鎖されないようにするアカウントを指定することができます。これらのポリシーで保護されているメンバーアカウントは閉鎖されません。SCP では、管理アカウントのプリンシパルに影響しないため、この操作を実行できません。

アカウントの閉鎖を拒否する IAM ポリシーは、次の 2 つの方法のいずれかで作成できます。

- Resource エレメントに arn を含めることにより、ポリシーで保護するアカウントを明示的にリストする。例については、「このポリシーに記載されているメンバーアカウントが閉鎖されないようにする」を参照してください。
- 個々のアカウントにタグを付けて、アカウントが閉鎖されないようにする。ポリシー内で aws:ResourceTag タググローバル条件キーを使用して、タグを付けたアカウントが閉鎖されないようにします。アカウントにタグを付ける方法については、「Organizations リソースのタグ付け」を参照してください。例については、「タグ付きのメンバーアカウントが閉鎖されないようにする」を参照してください。

メンバーアカウントが閉鎖されないようにする IAM ポリシーの例

次のコード例は、メンバーアカウントによるアカウントの閉鎖を制限するために使用できる 2 つの異なる方法を示しています。

タグ付きのメンバーアカウントが閉鎖されないようにする

管理アカウントの ID に次のポリシーをアタッチできます。このポリシーにより、管理アカウントのプリンシパルは、aws:ResourceTag タググローバル条件キー、AccountType キー、および Critical タグ値がタグ付けされているメンバーアカウントを閉鎖できなくなります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PreventCloseAccountForTaggedAccts",
            "Effect": "Deny",
            "Action": "organizations:CloseAccount",
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/AccountType": "Critical"}
            }
        }
    ]
}
```

このポリシーに記載されているメンバーアカウントが閉鎖されないようにする

管理アカウントの ID に次のポリシーをアタッチできます。このポリシーにより、管理アカウントのプリンシパルは、Resource エレメントで明示的に指定されたメンバーアカウントを閉鎖できなくなります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PreventCloseAccount",
            "Effect": "Deny",
            "Action": "organizations:CloseAccount",
            "Resource": [
                "arn:aws:organizations::555555555555:account/
o-12345abcdef/123456789012",
                "arn:aws:organizations::555555555555:account/
o-12345abcdef/123456789014"
            ]
        }
    ]
}
```

組織内の管理アカウントの閉鎖

組織内の管理アカウントを閉鎖するには、まず組織???内のすべてのメンバーアカウントを閉鎖または削除する必要があります。管理アカウントを閉鎖する行為は、閉鎖後期間が終了した後に、その組織内で作成した のインスタンス AWS Organizations とポリシーも削除します。

管理アカウントを閉鎖する方法

管理アカウントを閉鎖するには、次の手順に従います。

> **Important**
>
> 管理アカウントを閉鎖する前に、考慮事項を確認し、アカウントを閉鎖した場合の影響を理解しておくことを強くお勧めします。詳細については、「アカウント管理ガイド」の「アカウントを閉鎖する前に知っておくべきこと」および「アカウントを閉鎖した後の予定」を参照してください。 AWS

AWS Management Console

アカウントページから管理アカウントを閉鎖するには

> **Note**
> コンソールから AWS Organizations 管理アカウントを直接閉鎖することはできません。

1. 閉鎖する管理アカウントのルートユーザー AWS Management Console として にサインインします。IAM ユーザーまたはロールとしてサインインしている間は、アカウントを閉じることはできません。
2. 組織にアクティブなメンバーアカウントが残っていないことを確認します。これを行うには、AWS Organizations コンソール に移動し、すべてのメンバーアカウントがアカウント名のSuspended横に表示されていることを確認します。まだアクティブなメンバーアカウントがある場合は、次のステップに進む組織のメンバーアカウントの閉鎖前に、「」に記載されているガイダンスに従う必要があります。
3. 右上隅のナビゲーションバーで、アカウント名または番号を選択し、アカウント を選択します。
4. アカウントページ で、ページの下部までスクロールし、アカウントを閉じるセクションに移動します。アカウント閉鎖プロセスを読み、理解していることを確認します。
5. アカウント閉鎖ボタンを選択して、アカウント閉鎖プロセスを開始します。
6. 数分以内に、アカウントが閉鎖されたことを示す確認メールが届きます。

AWS CLI & AWS SDKs

このタスクは、 AWS CLI または AWS SDKsオペレーションではサポートされていません。このタスクは、 を使用してのみ実行できます AWS Management Console。

メンバーアカウントのルートユーザーの E メールアドレスの更新

セキュリティと管理のレジリエンスを高めるために、管理アカウント (必要な IAM アクセス許可を持つ) の IAM プリンシパルは、各アカウントに個別にサインインすることなく、メンバーアカウントのルートユーザーの E メールアドレス (プライマリ E メールアドレスとも呼ばれます) を一元的に更新できます。これにより、管理アカウント (または委任された管理者アカウント) の管理者は、メンバーアカウントをより詳細に制御できます。また、元のルートユーザーの E メールアドレスや管

理者認証情報にアクセスできなくなった場合でも、 全体の任意のメンバーアカウントのルートユーザーの E メールアドレスを最新の状態に保つ AWS Organizations ことができます。

管理アカウント管理者がルートユーザーの E メールアドレスを一元的に変更した場合、パスワードと MFA 設定の両方が変更前と同じままになります。MFA は、アカウントのルートユーザーの E メールアドレスと主要連絡先の電話番号を制御できるユーザーがバイパスできることに注意してください。

組織内のメンバーアカウントのルートユーザーの E メールアドレスを更新するには、以前にすべての機能モードを有効にしておく必要があります。統合請求モードまたは組織の一部ではないアカウント AWS Organizations では、 はルートユーザーの E メールアドレスを一元的に更新できません。API でサポートされていないアカウントのルートユーザーの E メールアドレスを変更したいユーザーは、引き続き請求コンソールを使用してルートユーザーの E メールアドレスを管理する必要があります。

メンバーアカウントのルートユーザーの E メールアドレスを一元的に更新する方法

ルートユーザーの E メールアドレスを更新するには、次の手順に従います。

AWS Management Console

> **メモ**
>
> - 組織内の管理アカウントまたは委任された管理者アカウントからメンバーアカウントに対してこの手順を実行するには、アカウント管理サービス の信頼されたアクセスを有効にする必要があります。
> - この手順を使用して、オペレーションの呼び出しに使用している組織とは異なる組織のアカウントにアクセスすることはできません。

AWS Organizations コンソールを使用してメンバーアカウントのルートユーザーの E メールアドレスを更新するには

1. 組織内の管理アカウントのルートユーザー (または同等の IAM アクセス許可) としてAWS Organizations コンソールにサインインします。
2. AWS アカウント ページで、ルートユーザーの E メールアドレスを更新するメンバーアカウントを選択します。

3. アカウントの詳細 セクションで、アクション ボタンを選択し、E メールアドレスの更新 を選択します。
4. E メール で、ルートユーザーの新しい E メールアドレスを入力し、保存 を選択します。これにより、新しい E メールアドレスにワンタイムパスワード (OTP) が送信されます。

> **Note**
>
> コードを待っている間に Organizations コンソールでこのページを閉じる必要がある場合は、コードが送信されてから 24 時間以内に OTP プロセスを戻して終了できます。これを行うには、アカウントの詳細ページでアクションボタンを選択し、E メール更新の完了を選択します。

5. 検証コード で、前のステップで新しい E メールアドレスに送信されたコードを入力し、確認 を選択します。これにより、アカウントのルートユーザーの E メールアドレスに更新がコミットされます。

AWS CLI & AWS SDKs

ルートユーザーの E メールアドレス (プライマリ E メールアドレスとも呼ばれます) を取得または更新するには、次の AWS CLI コマンドまたは AWS SDK と同等のオペレーションを使用します。

- GetPrimaryEmail
- StartPrimaryEmailUpdate
- AcceptPrimaryEmailUpdate

> **メモ**
>
> - 組織内の管理アカウントまたは委任された管理者アカウントからメンバーアカウントに対してこれらのオペレーションを実行するには、アカウント管理サービス の信頼されたアクセスを有効にする必要があります。
> - 操作の呼び出しに使用する組織と異なる組織のアカウントにアクセスすることはできません。

最小アクセス許可

各操作については、その操作に対応するアクセス許可が必要です。

- `account:GetPrimaryEmail`
- `account:StartPrimaryEmailUpdate`
- `account:AcceptPrimaryEmailUpdate`

これらの個々のアクセス許可を使用する場合、一部のユーザーにルートユーザーの E メールアドレス情報のみを読み取る権限を付与し、他のユーザーに読み取りと書き込みの両方の権限を付与できます。

ルートユーザーの E メール更新プロセスを完了するには、以下の例に示す順序でプライマリ E APIs を一緒に使用する必要があります。

Example **`GetPrimaryEmail`**

次の例では、組織内の指定されたメンバーアカウントからルートユーザーの E メールアドレスを取得します。使用される認証情報は、組織の管理アカウント、またはアカウント管理の委任管理者アカウントのいずれかから取得する必要があります。

```
$ aws account get-primary-email --account-id 123456789012
```

Example **`StartPrimaryEmailUpdate`**

次の例では、ルートユーザーの E メールアドレスの更新プロセスを開始し、新しい E メールアドレスを識別して、組織内の指定されたメンバーアカウントの新しい E メールアドレスにワンタイムパスワード (OTP) を送信します。使用される認証情報は、組織の管理アカウント、またはアカウント管理の委任管理者アカウントのいずれかから取得する必要があります。

```
$ aws account start-primary-email-update --account-id 123456789012 --primary-email
 john@examplecorp.com
```

Example **`AcceptPrimaryEmailUpdate`**

次の例では、OTP コードを受け入れ、新しい E メールアドレスを組織内の指定されたメンバーアカウントに設定します。使用される認証情報は、組織の管理アカウント、またはアカウント管理の委任管理者アカウントのいずれかから取得する必要があります。

```
$ aws account accept-primary-email-update --account-id 123456789012 --otp 12345678
 --primary-email john@examplecorp.com
```

組織の代替連絡先の更新

組織内のアカウントの代替連絡先を更新する場合は、AWS Organizations コンソールを使用するか、AWS CLI または AWS SDK を使用してプログラムで行います。代替連絡先の更新方法については、「AWS アカウント管理リファレンス」の「代替連絡先へのアクセスまたは更新」を参照してください。

組織の主な連絡先情報の更新

AWS 組織コンソールを使用するか、AWS CLI または AWS SDK を使用してプログラムで、組織内のアカウントの主要な連絡先情報を更新できます。主要連絡先情報を更新する方法については、「AWS Account Management Reference」(アカウント管理リファレンス) の「Accessing or updating the primary account contact」(主要アカウント連絡先へのアクセスと更新) を参照してください。

組織内で有効な AWS リージョン の更新

AWS Organizations コンソールを使用して、組織内のアカウントで有効な AWS リージョン を更新できます。有効な AWS リージョン を更新する方法については、「AWS アカウント管理リファレンス」の「Specifying which AWS リージョン your account can use」(アカウントで使用できる AWS リージョン の指定) を参照してください。

でのポリシーの管理 AWS Organizations

のポリシー AWS Organizations を使用すると、組織 AWS アカウント 内の に追加の管理タイプを適用できます。組織ですべての機能が有効にされている場合、ポリシーを使用できます。

AWS Organizations コンソールには、各ポリシータイプの有効または無効のステータスが表示されます。[Organize accounts (アカウントの整理)] タブの左側のナビゲーションペインで、Root を選択します。画面の右側の詳細ペインには、使用可能なすべてのポリシータイプが表示されます。リストには、その組織ルートで有効になっているものと、無効になっているものが示されます。タイプを [Enable (有効)] にするオプションが存在する場合、そのタイプは現在無効であることを意味します。タイプを [Disable (無効)] にするオプションが存在する場合、そのタイプは現在有効であることを意味します。

ポリシータイプ

Organizations のポリシータイプは、次の 2 つのカテゴリに大別されます。

承認ポリシー

承認ポリシーは、組織内の AWS アカウント のセキュリティを一元管理するのに役立ちます。

- サービスコントロールポリシー (SCP) では、組織のすべてのアカウントで使用可能な最大アクセス許可を一元的に制御できます。

管理ポリシー

管理ポリシーを使用すると、 AWS のサービスとその機能を一元的に設定および管理できます。

- 人工知能 (AI) サービスのオプトアウトポリシーを使用すると、組織のすべてのアカウントでの AWS AI サービス用のデータ収集をコントロールできます。
- バックアップポリシーは、バックアッププランを一元管理し、組織のアカウント全体の AWS リソースに適用するのに役立ちます。
- タグポリシーは、組織のアカウントの AWS リソースにアタッチされたタグを標準化するのに役立ちます。

次の表は、各ポリシータイプの主な特性をまとめたものです。これらのポリシータイプのその他の特徴については、のクォータ AWS Organizations を参照してください。

ポリシータイプ	管理アカウントに影響するか	アタッチの最大数 (ルート、OU、アカウントの合計)	最大サイズ	OU またはアカウントの有効なポリシーを表示可能か
SCP	 いいえ	5	5120 文字	 いいえ
AI サービスのオプトアウトポリシー	 はい	5	2500 文字	 はい
バックアップポリシー	 はい	10	10,000 文字	 はい
タグポリシー	 はい	10	10,000 文字	 はい

組織内でのポリシーの使用

- ポリシータイプの有効化と無効化
- 組織のポリシーに関する情報の取得
- の委任管理者 AWS Organizations
- 管理ポリシー
- サービスコントロールポリシー (SCP)

ポリシータイプの有効化と無効化

ポリシータイプの有効化

ポリシーを作成して組織にアタッチする前に、そのポリシータイプを有効にする必要があります。ポリシータイプの有効化は、組織ルートで行う 1 回限りのタスクです。ポリシータイプの有効化は、組織の管理アカウントからのみ行うことができます。

最小アクセス許可

ポリシータイプを有効にするには、以下のアクションを実行するアクセス許可が必要です。

- `organizations:EnablePolicyType`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:ListRoots` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

ポリシータイプを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. [Policies] (ポリシー) ページで、有効化するポリシータイプの名前を選択します。
3. ポリシータイプのページで [Enable ***policy type***] (ポリシータイプの有効化) を選択します。

   ページは、指定したタイプの使用可能なポリシーのリストに置き換えられます。

AWS CLI & AWS SDKs

ポリシータイプを有効にするには

次のいずれかのコマンドを使用して、ポリシータイプを有効にできます。

- AWS CLI: enable-policy-type

次の例は、組織のバックアップポリシーを有効にする方法を示しています。組織のルートの ID を指定する必要があることに注意してください。

```
$ aws organizations enable-policy-type \
    --root-id r-a1b2 \
    --policy-type BACKUP_POLICY
{
    "Root": {
        "Id": "r-a1b2",
        "Arn": "arn:aws:organizations::123456789012:root/o-aa111bb222/r-a1b2",
        "Name": "Root",
        "PolicyTypes": [
            {
                "Type": "BACKUP_POLICY",
                "Status": "ENABLED"
            }
        ]
    }
}
```

出力の PolicyTypes のリストには、指定したポリシータイプが ENABLED の Status で含まれるようになります。

- AWS SDK: EnablePolicyType

ポリシータイプの無効化

組織内で特定のポリシータイプを使用しなくなった場合は、誤って使用されないように、そのタイプを無効にすることができます。ポリシータイプの無効化は、組織の管理アカウントからのみ行うことができます。

> **Important**
>
> - ポリシータイプを無効にすると、指定されたされたタイプのすべてのポリシーは、組織ルート内のすべてのエンティティから自動的にデタッチされます。ポリシーの削除は行われません。
> - (サービスコントロールポリシータイプのみ) 後で SCP ポリシータイプを再び有効にした場合、組織ルート内のすべてのエンティティはデフォルトの FullAWSAccess SCP にのみアタッチされた状態になります。組織で SCP が無効になった時点で、エンティティへ

の SCP のアタッチは解除されます。後から SCP を再度有効にした場合は、必要に応じて組織のルート、OU、アカウントにアタッチし直す必要があります。

最小アクセス許可

SCP を無効にするには、以下のアクションを実行する権限が必要です。

- `organizations:DisablePolicyType`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:ListRoots` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

ポリシータイプを無効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. ポリシーページで、無効にするポリシータイプの名前を選択します。
3. ポリシータイプのページで [Disable ***policy type***] (ポリシータイプの無効化) を選択します。
4. 確認ダイアログボックスで、「**disable**」と入力してから、[Disable] (無効化する) を選択します。

   使用可能なポリシーの一覧に、指定したタイプが表示されなくなります。

AWS CLI & AWS SDKs

ポリシータイプを無効にするには

ポリシータイプを無効にするには、次のコマンドを使用します。

- AWS CLI: disable-policy-type

次の例は、組織のバックアップポリシーを無効にする方法を示しています。組織のルートの ID を指定する必要があることに注意してください。

```
$ aws organizations disable-policy-type \
    --root-id r-a1b2 \
    --policy-type BACKUP_POLICY
{
    "Root": {
        "Id": "r-a1b2",
        "Arn": "arn:aws:organizations::123456789012:root/o-aa111bb222/r-a1b2",
        "Name": "Root",
        "PolicyTypes": []
    }
}
```

出力の PolicyTypes のリストには、指定したポリシータイプが含まれなくなります。

- AWS SDK: DisablePolicyType

組織のポリシーに関する情報の取得

このセクションでは、組織内のポリシーの詳細を取得するさまざまな方法について説明します。これらの手順は、すべてのポリシータイプに適用されます。あるタイプのポリシーを組織ルート内のエンティティにアタッチする前に、その組織ルートでそのポリシータイプを有効にする必要があります。

すべてのポリシーの一覧表示

最小アクセス許可

組織内のポリシーを一覧表示するには、次のアクセス権限が必要です。

- organizations:ListPolicies

組織内のポリシーは、 または AWS Command Line Interface （ AWS CLI) コマンドまたは AWS SDK オペレーション AWS Management Console を使用して表示できます。

AWS Management Console

組織のすべてのポリシーを一覧表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ポリシーページで、一覧表示するポリシーを選択します。

   指定したポリシータイプが有効な場合、コンソールには、組織で現在利用できる同様のタイプのポリシーの一覧が表示されます。
3. ポリシーページに戻り、ポリシータイプごとにこれを繰り返します。

AWS CLI & AWS SDKs

以下のコード例は、`ListPolicies` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to list the AWS Organizations policies associated with an
/// organization.
/// </summary>
public class ListPolicies
{
    /// <summary>
    /// Initializes an Organizations client object, and then calls its
```

```
        /// ListPoliciesAsync method.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            // The value for the Filter parameter is required and must must be
            // one of the following:
            //     AISERVICES_OPT_OUT_POLICY
            //     BACKUP_POLICY
            //     SERVICE_CONTROL_POLICY
            //     TAG_POLICY
            var request = new ListPoliciesRequest
            {
                Filter = "SERVICE_CONTROL_POLICY",
                MaxResults = 5,
            };

            var response = new ListPoliciesResponse();
            try
            {
                do
                {
                    response = await client.ListPoliciesAsync(request);
                    response.Policies.ForEach(p => DisplayPolicies(p));
                    if (response.NextToken is not null)
                    {
                        request.NextToken = response.NextToken;
                    }
                }
                while (response.NextToken is not null);
            }
            catch (AWSOrganizationsNotInUseException ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        /// <summary>
        /// Displays information about the Organizations policies associated
        /// with an organization.
        /// </summary>
        /// <param name="policy">An Organizations policy summary to display
```

```
        /// information on the console.</param>
        private static void DisplayPolicies(PolicySummary policy)
        {
            string policyInfo = $"{policy.Id}
 {policy.Name}\t{policy.Description}";

            Console.WriteLine(policyInfo);
        }
    }
```

- API の詳細については、「 API リファレンスListPolicies」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

特定のタイプの組織のすべてのポリシーのリストを取得するには

次の例は、フィルターパラメータで指定された SCP のリストを取得する方法を示しています。

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

出力には、ポリシーのリストと概要情報が含まれます。

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllS3Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid111",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Enables account admins to delegate
 permissions for any S3 actions to users and roles in their accounts."
                },
                {
                        "Type": "SERVICE_CONTROL_POLICY",
```

```
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                },
                {
                        "AwsManaged": true,
                        "Description": "Allows access to every operation",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-FullAWSAccess",
                        "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
                        "Name": "FullAWSAccess"
                }
        ]
}
```

- API の詳細については、「 コマンドリファレンスListPolicies」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def list_policies(policy_filter, orgs_client):
    """
    Lists the policies for the account, limited to the specified filter.

    :param policy_filter: The kind of policies to return.
    :param orgs_client: The Boto3 Organizations client.
    :return: The list of policies found.
    """
```

```
    try:
        response = orgs_client.list_policies(Filter=policy_filter)
        policies = response["Policies"]
        logger.info("Found %s %s policies.", len(policies), policy_filter)
    except ClientError:
        logger.exception("Couldn't get %s policies.", policy_filter)
        raise
    else:
        return policies
```

- API の詳細については、 ListPolicies AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ルート、OU、またはアカウントにアタッチされているポリシーを一覧表示する

最小アクセス許可

組織内のルート、組織単位 (OU)、またはアカウントにアタッチされたポリシーを一覧表示するには、次のアクセス許可が必要です。

- 指定されたターゲット (または "*") の Amazon リソースネーム (ARN) を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:ListPoliciesForTarget`。

AWS Management Console

指定したルート、OU、またはアカウント に直接アタッチされているすべてのポリシーを一覧表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. リポジトリの AWS アカウント ページで、ポリシーを表示するルート、OU、またはアカウントの名前を選択します。必要な OU を見つけるために、OUを展開する

(▶
を選択する) 必要がある場合があります。

3. ルート、OU、またはアカウントページで、[Policies] (ポリシー) タブを選択します。

   [Policies] (ポリシー) タブでは、そのルート、OU、またはアカウントにアタッチされているすべてのポリシーが、ポリシータイプ別にグループ化されて表示されます。

AWS CLI & AWS SDKs

指定したルート、OU、またはアカウント に直接アタッチされているすべてのポリシーを一覧表示するには

エンティティにアタッチされているポリシーを一覧表示するには、次のいずれかのコマンドを使用します。

- AWS CLI: list-policies-for-target

  次の例では、指定された OU にアタッチされているすべてのサービスコントロールポリシーを一覧表示します。ルート、OU、またはアカウントの ID、および一覧表示するポリシーのタイプを指定する必要があります。

```
$ aws organizations list-policies-for-target \
    --target-id ou-a1b2-f6g7h222 \
    --filter SERVICE_CONTROL_POLICY
{
    "Policies": [
        {
            "Id": "p-FullAWSAccess",
            "Arn": "arn:aws:organizations::aws:policy/service_control_policy/p-
FullAWSAccess",
            "Name": "FullAWSAccess",
            "Description": "Allows access to every operation",
            "Type": "SERVICE_CONTROL_POLICY",
            "AwsManaged": true
        }
    ]
}
```

- AWS SDKs ListPoliciesForTarget

ポリシーがアタッチされているすべての root、OU、およびアカウントの一覧表示

> **最小アクセス許可**
>
> ポリシーがアタッチされているエンティティを一覧表示するには、次のアクセス権限が必要です。
>
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:ListTargetsForPolicy`。

AWS Management Console

特定のポリシーがアタッチされているすべてのルート、OU、およびアカウントを一覧表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ポリシーページで、ポリシータイプを選択してから、添付ファイルを確認するポリシーの名前を選択します。
3. [Targets] (ターゲット) タブを選択し、選択したポリシーがアタッチされているすべてのルート、OU、およびアカウントのテーブルを表示します。

AWS CLI & AWS SDKs

特定のポリシーがアタッチされているすべてのルート、OU、およびアカウントを一覧表示するには

ポリシーを含むエンティティを一覧表示するには、次のいずれかのコマンドを使用します。

- AWS CLI: list-targets-for-policy

  次の例は、指定されたポリシーのルート、OU、およびアカウントに対するすべての添付ファイルを示しています。

```
$ aws organizations list-targets-for-policy \
```

```
    --policy-id p-FullAWSAccess
{
    "Targets": [
        {
            "TargetId": "ou-a1b2-f6g7h111",
            "Arn": "arn:aws:organizations::123456789012:ou/o-aa111bb222/ou-a1b2-
f6g7h111",
            "Name": "testou2",
            "Type": "ORGANIZATIONAL_UNIT"
        },
        {
            "TargetId": "ou-a1b2-f6g7h222",
            "Arn": "arn:aws:organizations::123456789012:ou/o-aa111bb222/ou-a1b2-
f6g7h222",
            "Name": "testou1",
            "Type": "ORGANIZATIONAL_UNIT"
        },
        {
            "TargetId": "123456789012",
            "Arn": "arn:aws:organizations::123456789012:account/o-
aa111bb222/123456789012",
            "Name": "My Management Account (bisdavid)",
            "Type": "ACCOUNT"
        },
        {
            "TargetId": "r-a1b2",
            "Arn": "arn:aws:organizations::123456789012:root/o-aa111bb222/r-a1b2",
            "Name": "Root",
            "Type": "ROOT"
        }
    ]
}
```

- AWS SDKs ListTargetsForPolicy

ポリシーの詳細の取得

最小アクセス許可

ポリシーの詳細を表示するには、次のアクセス権限が必要です。

- 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:DescribePolicy`。

AWS Management Console

ポリシーの詳細を取得するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ポリシーページで、確認するポリシーのポリシータイプを選択してから、ポリシーの名前を選択します。

   ポリシーページには、ARN、説明、アタッチメントなど、ポリシーに関する利用可能な情報が表示されます。

   - [Content] (コンテンツ) タブには、ポリシーの現在の内容が JSON 形式で表示されます。
   - [Targets] (ターゲット) タブには、ポリシーがアタッチされているルート、OU、およびアカウントの一覧が表示されます。
   - [Tags] (タグ) タブには、ポリシーにアタッチされたタグが表示されます。注: [Tags] (タグ) タブは、 AWS 管理ポリシーでは使用できません。

   ポリシーを編集するには、[Edit policy] (ポリシーの編集) を選択します。編集要件はポリシータイプごとに異なるため、指定したポリシータイプのポリシーの作成および更新手順を参照してください。

AWS CLI & AWS SDKs

以下のコード例は、`DescribePolicy` の使用方法を示しています。

CLI

AWS CLI

ポリシーに関する情報を取得するには

次の例は、ポリシーに関する情報をリクエストする方法を示しています。

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

出力には、ポリシーの詳細を含むポリシーオブジェクトが含まれます。

```
{
        "Policy": {
                "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement
\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n
 \"Resource\": \"*\"\n    }\n  ]\n}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::111111111111:policy/o-
exampleorgid/service_control_policy/p-examplepolicyid111",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-examplepolicyid111",
                        "AwsManaged": false,
                        "Name": "AllowAllS3Actions",
                        "Description": "Enables admins to delegate S3
 permissions"
                }
        }
}
```

- API の詳細については、「 コマンドリファレンスDescribePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def describe_policy(policy_id, orgs_client):
    """
    Describes a policy.

    :param policy_id: The ID of the policy to describe.
```

```
    :param orgs_client: The Boto3 Organizations client.
    :return: The description of the policy.
    """
    try:
        response = orgs_client.describe_policy(PolicyId=policy_id)
        policy = response["Policy"]
        logger.info("Got policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't get policy %s.", policy_id)
        raise
    else:
        return policy
```

- API の詳細については、 DescribePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

の委任管理者 AWS Organizations

AWS Organizations 管理アカウントとそのユーザーとロールは、そのアカウントで実行する必要があるタスクにのみ使用することをお勧めします。また、すべての AWS リソースを組織内の他のメンバーアカウントに保存し、管理アカウントからは切り離すことをおすすめします。これは、Organizations のサービスコントロールポリシー (SCP) などのセキュリティ機能は、管理アカウントのユーザーやロールに制限を加えることができないためです。

組織の管理アカウントから、ポリシー管理を Organizations の指定のメンバーアカウントに委任して、デフォルトでは管理アカウントのみが使用できるポリシーアクションを実行できます。

リソースベースの委任ポリシーを作成または更新する

管理アカウントから、組織のリソースベースの委任ポリシーを作成または更新し、ポリシーに対してアクションを実行できるメンバーアカウントを指定するステートメントを追加します。ポリシーに複数のステートメントを追加して、メンバーアカウントにさまざまなアクセス許可を示すことができます。

最小アクセス許可

リソースベースの委任ポリシーを作成または更新するには、次のアクションを実行するアクセス許可が必要です。

- `organizations:PutResourcePolicy`
- `organizations:DescribeResourcePolicy`

また、必要なアクションに対応する IAM アクセス許可を委任された管理者アカウントのロールとユーザーに付与する必要があります。IAM アクセス許可がない場合、呼び出し元のプリンシパルには AWS Organizations ポリシーを管理するために必要なアクセス許可がないと見なされます。

AWS Management Console

次のいずれかの方法を使用して、 AWS Management Console のリソースベースの委任ポリシーにステートメントを追加します。

- JSON ポリシー — リソースベースの委任ポリシーの例を貼り付けてカスタマイズしてアカウントで使用するか、JSON エディタでの独自の JSON ポリシードキュメントを入力します。
- ビジュアルエディタ — ビジュアルエディタで新しい委任ポリシーを作成します。これにより、JSON 構文を記述せずに委任ポリシーを作成できます。

JSON ポリシーエディタを使用して委任ポリシーを作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [設定] を選択します。
3. [AWS Organizations の委任管理者]セクションで、[委任] を選択して Organizations 委任ポリシーを作成します。既存の委任ポリシーを更新するには、[Edit] (編集) を選択します。
4. JSON ポリシードキュメントを入力するか貼り付けます。IAM ポリシー言語の詳細については、「IAM JSON ポリシーリファレンス」を参照してください。
5. ポリシーの検証中に生成されたセキュリティ警告、エラー、または一般的な警告を解決し、[Create policy] (ポリシーの作成) を選択して作業を保存します。

ビジュアルエディタを使用して、委任ポリシーを作成または更新します。

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [設定] を選択します。
3. [AWS Organizations の委任管理者]セクションで、[委任] を選択して Organizations 委任ポリシーを作成します。既存の委任ポリシーを更新するには、[Edit] (編集) を選択します。
4. [Create Delegation policy] (委任ポリシーの作成) ページで、[Add new statement] (新しいステートメントを追加)を選択します。
5. [Effect] (効果) を `Allow` に設定します。
6. `Principal` を追加して委任したいメンバーアカウントを定義します。構文の詳細については、「リソースベースの委任ポリシーの例」を参照してください。
7. [Actions] (アクション) のリストから、委任するアクションを選択します。[Filter actions] (アクションのフィルタ)を使用して選択を絞り込むことができます。
8. 委任されたメンバーアカウントが組織ルートまたは組織単位 (OU) にポリシーをアタッチできるかどうかを指定するには、`Resources` を設定します。また、リソースタイプとして `policy` を選択する必要があります。詳細については、「リソースベースの委任ポリシーの例」を参照してください。リソースは次の方法で指定できます。

   - [Add a resource] (リソースの追加) を選択し、ダイアログボックスのプロンプトに従って Amazon リソースネーム (ARN) を作成します。
   - エディタでリソース ARN を手動で一覧表示します。ARN 構文の詳細については、「 AWS 全般のリファレンスガイド」の「Amazon リソースネーム (ARN)」を参照してください。ポリシーのリソース要素で ARN を使用する方法については、「IAM JSON ポリシー要素: Resource」を参照してください。
9. 委任するポリシータイプなど、他の条件を指定するには、[Add a condition] (条件の追加) を選択します。条件の [Condition key] (条件キー)、[Tag key] (タグキー)、[Qualifier] (修飾子)、[Operator] (演算子) を選択し、**`Value`** を入力します。詳細については、「リソースベースの委任ポリシーの例」を参照してください。完了したら、[Add condition] (条件の追加) を選択します。条件要素の詳細については、「IAM JSON ポリシーリファレンス」の「IAM JSON ポリシーの要素: 条件」を参照してください。
10. さらにアクセス許可ブロックを追加するには、[Add new statement] (新しいステートメントを追加) を選択します。各ブロックに対して、ステップ 5 から 9 を繰り返します。

11. ポリシーの検証で生成されたセキュリティ警告、エラー、または一般的な警告を解決し、[Create policy] (ポリシーの作成) を選択して作業を保存します。

AWS CLI & AWS SDKs

委任ポリシーを作成または更新する

以下のコマンドを使用して委任ポリシーを作成または更新できます。

- AWS CLI: put-resource-policy

以下は委任ポリシーを作成または更新する例です。

```
$ aws organizations put-resource-policy --content
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Fully_manage_backup_policies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "135791357913"
            },
            "Action": [
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:CreatePolicy",
                "organizations:DescribePolicy",
                "organizations:UpdatePolicy",
                "organizations:DeletePolicy",
                "organizations:AttachPolicy",
                "organizations:DetachPolicy"
            ],
            "Resource": [
                "arn:aws:organizations::246802468024:root/o-abcdef/r-pqrstu",
                "arn:aws:organizations::246802468024:ou/o-abcdef/*",
                "arn:aws:organizations::246802468024:account/o-abcdef/*",
                "arn:aws:organizations::246802468024:organization/policy/
backup_policy/*",
            ],
            "Condition": {
                "StringLikeIfExists": {
                    "organizations:PolicyType": [
```

```
                        "BACKUP_POLICY"
                    ]
                }
            }
        }
    ]
}
```

- AWS SDK: PutResourcePolicy

サポート対象の委任ポリシーのアクション

委任ポリシーでは、次のアクションがサポートされています。

- AttachPolicy
- CreatePolicy
- DeletePolicy
- DescribeAccount
- DescribeCreateAccountStatus
- DescribeEffectivePolicy
- DescribeHandshake
- DescribeOrganization
- DescribeOrganizationalUnit
- DescribePolicy
- DescribeResourcePolicy
- DetachPolicy
- DisablePolicyType
- EnablePolicyType
- ListAccounts
- ListAccountsForParent
- ListAWSServiceAccessForOrganization
- ListChildren

- ListCreateAccountStatus
- ListDelegatedAdministrators
- ListDelegatedServicesForAccount
- ListHandshakesForAccount
- ListHandshakesForOrganization
- ListOrganizationalUnitsForParent
- ListParents
- ListPolicies
- ListPoliciesForTarget
- ListRoots
- ListTagsForResource
- ListTargetsForPolicy
- TagResource
- UntagResource
- UpdatePolicy

サポートされている条件キー

委任ポリシーに使用できるのは AWS Organizations 、 でサポートされている条件キーのみです。詳細については、「サービス認証リファレンス」の「 の条件キー AWS Organizations」を参照してください。

リソースベースの委任ポリシーを表示する

管理アカウントから、組織のリソースベースの委任ポリシーを表示して、どの委任管理者がどのポリシータイプを管理できるかを把握できます。

> **最小アクセス許可**
>
> リソースベースのデリゲーションポリシーを表示するには、organizations:DescribeResourcePolicy のアクションを実行するアクセス許可が必要です。

AWS Management Console

委任ポリシーを表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [設定] を選択します。
3. [AWS Organizations の委任管理者]セクションをスクロールして、委任ポリシー全体を表示します。

AWS CLI & AWS SDKs

委任ポリシーを表示する

以下のコマンドを使用して委任ポリシーを表示できます。

- AWS CLI: describe-resource-policy

  以下はポリシーを取得する例です。

  ```
  $ aws organizations describe-resource-policy
  ```

- AWS SDK: DescribeResourcePolicy

リソースベースの委任ポリシーを削除する

組織内のポリシーの管理を委任する必要がなくなった場合、リソースベースの委任ポリシーを組織の管理アカウントから削除できます。

> **Important**
>
> リソースベースの委任ポリシーを削除した場合、回復できません。

> **最小アクセス許可**
>
> リソースベースのデリゲーションポリシーを削除するには、`organizations:DeleteResourcePolicy` のアクションを実行するアクセス許可が必要です。

AWS Management Console

委任ポリシーを削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [設定] を選択します。
3. [AWS Organizations の委任管理者]セクションで、[削除]を選択します。
4. [Detach policy] (ポリシーの削除) のダイアログボックスで、 **`delete`** を入力します。[Delete policy] (ポリシーの削除) を選択します。

AWS CLI & AWS SDKs

委任ポリシーを削除する

以下のコマンドを使用して委任ポリシーを削除できます。

- AWS CLI: delete-resource-policy

  以下はポリシーを削除する例です。

  ```
  $ aws organizations delete-resource-policy
  ```

- AWS SDK: DeleteResourcePolicy

リソースベースの委任ポリシーの例

以下はリソースベースの委任ポリシーを使用する方法の例です。

例

- 例: 組織、OU、アカウント、ポリシーの表示
- 例: 組織のバックアップポリシーを管理するための一括アクセス許可

例: 組織、OU、アカウント、ポリシーの表示

ポリシーの管理を委任する前に、組織の階層構造を移動したり、組織単位 (OU)、アカウント、およびそれらにアタッチされているポリシーを表示するために、アクセス許可を委任する必要があります。

以下は、これらのアクセス許可をメンバーアカウント *AccountId* のリソースベースの委任ポリシーに含める方法の例です。

> **Important**
>
> このポリシーを使用して Organizations 読み取り専用アクションを委任することは可能ですが、この例で示されているように、必要最小限のアクションのみにアクセス許可を付与することをおすすめします。

このポリシーでは、AWS API または AWS CLI から、アクションをプログラムで完了するために必要なアクセス許可を付与します。この委任ポリシーを使用するには、*AccountId* の AWS プレースメントフォルダテキストを独自の情報に置き換えてください。次に、の委任管理者 AWS Organizations の指示に従ってください。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DelegatingNecessaryDescribeListActions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::AccountId:root"
      },
      "Action": [
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",
        "organizations:DescribeAccount",
        "organizations:DescribePolicy",
        "organizations:DescribeEffectivePolicy",
```

```
        "organizations:ListRoots",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListChildren",
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListPolicies",
        "organizations:ListPoliciesForTarget",
        "organizations:ListTargetsForPolicy",
        "organizations:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

例: 組織のバックアップポリシーを管理するための一括アクセス許可

以下は create、read、update、delete アクションや attach、detach のポリシーアクションなど、組織内のバックアップポリシーを管理するために必要なすべてのアクセス許可を管理アカウントが委任できるようにする、リソースベースの委任ポリシーを作成する方法の例です。各アクション、リソース、および条件の重要性を理解するには、「リソースベースの委任ポリシーの例」を参照してください。

> **Important**
>
> このポリシーにより、委任管理者は管理アカウントを含む組織内の任意のアカウントで作成したポリシーに対して指定されたアクションを実行できます。

この委任ポリシーの例では、 AWS API または からプログラムでアクションを完了するために必要なアクセス許可を付与します AWS CLI。この委任ポリシーを使用するには、、*MemberAccountId*、*ManagementAccountIdOrganizationId*、 のプレースホルダーテキストを AWS ユーザー自身の情報に置き換え*RootId*ます。次に、の委任管理者 AWS Organizations の指示に従ってください。

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

```
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::MemberAccountId:root"
      },
      "Action": [
            "organizations:DescribeOrganization",
            "organizations:DescribeOrganizationalUnit",
            "organizations:DescribeAccount",
            "organizations:ListRoots",
            "organizations:ListOrganizationalUnitsForParent",
            "organizations:ListParents",
            "organizations:ListChildren",
            "organizations:ListAccounts",
            "organizations:ListAccountsForParent",
            "organizations:ListTagsForResource"
        ],
      "Resource": "*"
    },
    {
      "Sid": "DelegatingNecessaryDescribeListActionsForSpecificPolicyType",
      "Effect": "Allow",
      "Principal": {
            "AWS": "arn:aws:iam::MemberAccountId:root"
      },
      "Action": [
            "organizations:DescribePolicy",
            "organizations:DescribeEffectivePolicy",
            "organizations:ListPolicies",
            "organizations:ListPoliciesForTarget",
            "organizations:ListTargetsForPolicy"
      ],
      "Resource": "*",
      "Condition": {
            "StringLikeIfExists": {
                "organizations:PolicyType": "BACKUP_POLICY"
            }
      }
    },
    {
      "Sid": "DelegatingAllActionsForBackupPolicies",
      "Effect": "Allow",
      "Principal": {
      "AWS": "arn:aws:iam::MemberAccountId:root"
    },
```

```
      "Action": [
            "organizations:CreatePolicy",
            "organizations:UpdatePolicy",
            "organizations:DeletePolicy",
            "organizations:AttachPolicy",
            "organizations:DetachPolicy",
            "organizations:EnablePolicyType",
            "organizations:DisablePolicyType"
      ],
      "Resource": [
            "arn:aws:organizations::ManagementAccountId:root/o-OrganizationId/
r-RootId",
            "arn:aws:organizations::ManagementAccountId:ou/o-OrganizationId/*",
            "arn:aws:organizations::ManagementAccountId:account/o-OrganizationId/*",
            "arn:aws:organizations::ManagementAccountId:policy/o-OrganizationId/
backup_policy/*"
      ],
      "Condition": {
            "StringLikeIfExists": {
                "organizations:PolicyType": "BACKUP_POLICY"
            }
      }
    }
  ]
}
```

管理ポリシー

管理ポリシーを使用すると、AWS サービスとその機能を一元的に設定および管理できます。ポリシーが OU とそれを継承するアカウントに与える影響は、AWS Organizations に適用した管理ポリシーの種類によって異なります。管理ポリシーに関連する用語や概念を理解するには、このセクションのトピックを確認してください。

トピック

- 管理ポリシーの継承を理解する
- AI サービスのオプトアウトポリシー
- バックアップポリシー
- タグポリシー

管理ポリシーの継承を理解する

> **Note**
>
> SCP は IAM アクションの許可と拒否の両方を管理するから、このセクションの情報は SCP には当てはありません。SCP はルート、OU、アカウントに関連付けられていますが、アクションを許可するには、ルートからアカウントへのダイレクトパスにある各 OU (ターゲットアカウント自体を含む) のすべてのレベルで SCP での明示的な `allow` ステートメントが必要です。AWS Organizations 階層での SCP の動作の詳細については、「SCP 評価」を参照してください。

組織内の組織エンティティ (組織ルート、組織単位 (OU)、またはアカウント) に管理ポリシーをアタッチできます。

- 管理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 OU およびアカウントがそのポリシーを継承します。
- 特定の OU に管理ポリシーをアタッチすると、その OU または子 OU の直下にあるアカウントがポリシーを継承します。
- 特定のアカウントに管理ポリシーをアタッチすると、そのアカウントにのみ影響します。

組織内の複数のレベルに管理ポリシーをアタッチできるため、アカウントは複数のポリシーを継承できます。

このセクションでは、親ポリシーと子ポリシーがアカウントの有効なポリシーにどのように処理されるかを説明します。

トピック

継承用語

このトピックでは、管理ポリシーの継承について説明するときに、次の用語を使用します。

ポリシーの継承

組織の最上位ルートから、組織単位 (OU) 階層、個々のアカウントへと移行する、組織のさまざまなレベルでのポリシーの相互作用です。

ポリシーは、組織ルート、OU、個々のアカウント、およびこれらの組織エンティティの任意の組み合わせにアタッチできます。ポリシーの継承とは、組織ルートまたは OU にアタッチされた管理ポリシーを指します。管理ポリシーがアタッチされている組織ルートまたは OU のメンバーであるすべてのアカウントは、そのポリシーを継承します。

例えば、管理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がそのポリシーを継承します。これは、組織内のすべてのアカウントが常に組織ルートの下にあるためです。特定の OU にポリシーをアタッチすると、その OU または子 OU の直下にあるアカウントがそのポリシーを継承します。組織内の複数のレベルにポリシーをアタッチできるため、アカウントは 1 つのポリシータイプに対して複数のポリシードキュメントを継承する場合があります。

親ポリシー

組織ツリーで下位のエンティティにアタッチされているポリシーよりも上位にアタッチされているポリシー。

例えば、管理ポリシー A を組織ルートにアタッチすると、それは単なるポリシーです。ここでポリシー B を そのルートの下にある OU にアタッチすると、ポリシー A はポリシー B の親ポリシーとなり、ポリシー B はポリシー A の子ポリシーとなります。ポリシー A とポリシー B がマージされ、OU のアカウントの有効なタグポリシーになります。

子ポリシー

組織ツリーで親ポリシーよりも下位レベルでアタッチされているポリシー。

有効なポリシー

アカウントに適用されるルールを指定する、最終的な 1 つのポリシードキュメント。有効なポリシーは、アカウントが継承するすべてのポリシーと、アカウントに直接アタッチされたポリシーが集約されたものです。例えば、タグポリシーを使用すると、アカウントに適用される有効なタグポリシーを表示できます。詳細については、「有効なタグポリシーの表示」を参照してください。

継承演算子

継承されたポリシーを 1 つの有効なポリシーにマージする方法を制御する演算子。これらの演算子は、アドバンスト機能とみなされます。経験豊富なポリシー作成者は、ポリシーを使用して、

子ポリシーがどのような変更を行うことができるか、ポリシーの設定がどのようにマージされるかを制限できます。詳細については、「継承演算子」を参照してください。

管理ポリシータイプのポリシー構文と継承

ポリシーが OU とそれを継承するアカウントに与える影響は、選択した管理ポリシーの種類によって異なります。管理ポリシーには、次のタイプがあります。

- 人工知能 (AI) サービスのオプトアウトポリシー
- バックアップポリシー
- タグポリシー

この管理ポリシータイプの構文には、継承演算子 が含まれています。継承演算子を使用すると、適用する親ポリシーの要素や、子 OU とアカウントが継承する際に上書きまたは変更できる要素を細かく指定できます。

有効なポリシーは、組織ルートと OU から継承されるルールのセットと、アカウントに直接アタッチされたルールのセットです。有効なポリシーは、アカウントに適用される最終的なルールセットを指定します。適用されたポリシー内のすべての継承演算子の効果を含む、アカウントの有効なポリシーを表示できます。詳細については、「有効なタグポリシーの表示」を参照してください。

継承演算子

継承演算子は、アカウントの有効なポリシーが作成される際に、継承されたポリシーとアカウントポリシーがどのようにマージされるかを制御します。これらの演算子には、値設定演算子と子制御演算子が含まれます。

AWS Organizations コンソールでビジュアルエディタを使用する場合は、`@@assign` 演算子のみを使用できます。他の演算子は、アドバンスト機能とみなされます。他の演算子を使用するには、JSON ポリシーを手動で作成する必要があります。経験豊富なポリシーの作成者は、継承演算子を使用して、有効なポリシーに適用するタグ値を制御し、子ポリシーがどのような変更を行うことができるかを制限できます。

値設定演算子

次の値設定演算子を使用して、ポリシーと親ポリシーとの相互作用を制御できます。

- `@@assign` - 継承されたポリシー設定を指定した設定で上書きします。指定した設定が継承されていない場合、この演算子はその設定を有効なポリシーに追加します。この演算子は、任意のタイプのポリシー設定に適用できます。
  - 単一値の設定の場合、この演算子は、継承された値を指定された値に置き換えます。
  - 複数値設定 (JSON 配列) の場合、この演算子は、継承された値をすべて削除し、このポリシーで指定された値に置き換えます。
- `@@append` - 継承された設定に、指定した設定を (一切削除せずに) 追加します。指定した設定が継承されていない場合、この演算子はその設定を有効なポリシーに追加します。この演算子は、複数値の設定でのみ使用できます。
  - この演算子は、指定された値を継承された配列内の任意の値に追加します。
- `@@remove` - 継承された指定の設定を有効なポリシーから削除します (存在する場合)。この演算子は、複数値の設定でのみ使用できます。
  - この演算子は、親ポリシーから継承された値の配列から、指定された値のみを削除します。他の値は配列内に引き続き存在することができ、子ポリシーによって継承できます。

子制御演算子

制御演算子の使用はオプションです。`@@operators_allowed_for_child_policies` 演算子を使用して、子ポリシーで使用できる値設定演算子を制御できます。すべての演算子、一部の演算子を許可するか、または演算子を一切許可しないという選択肢があります。デフォルトでは、すべての演算子 (`@@all`) が許可されます。

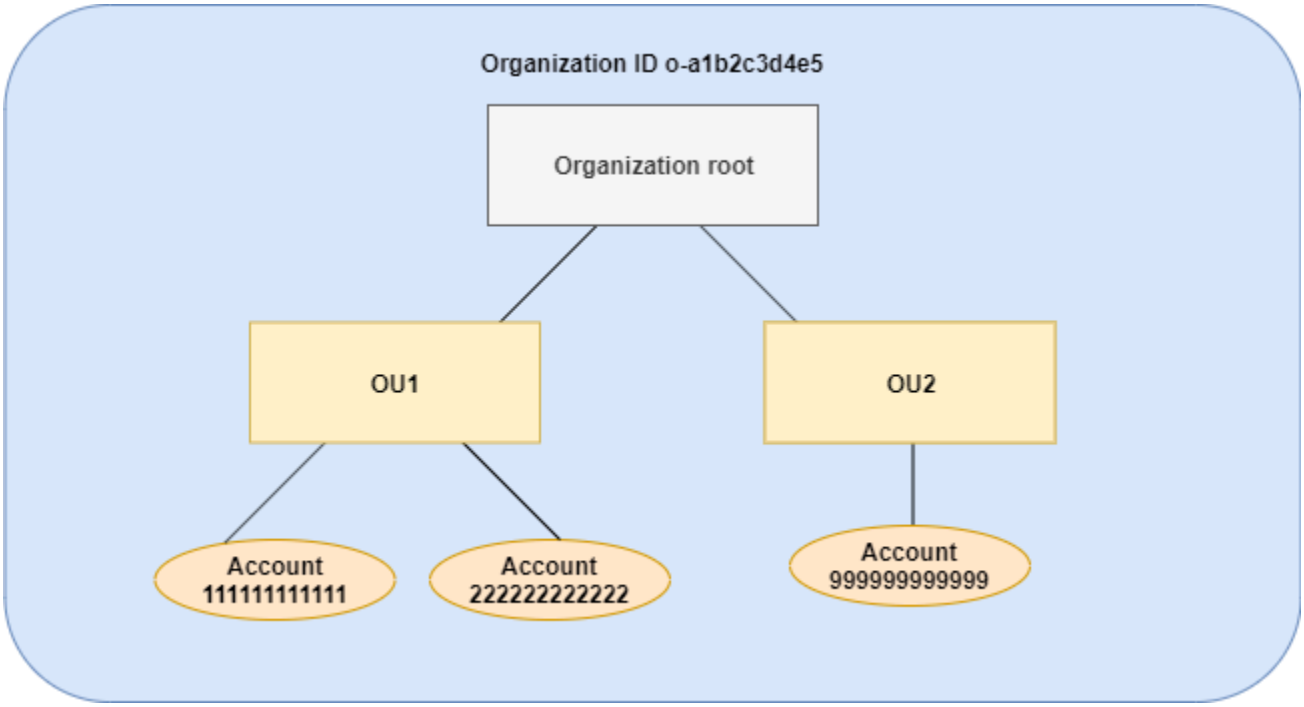
- `"@@operators_allowed_for_child_policies": ["@@all"]` - 子 OU とアカウントは、ポリシーの任意の演算子を使用できます。デフォルトでは、子ポリシーですべての演算子が許可されます。
- `"@@operators_allowed_for_child_policies": ["@@assign", "@@append", "@@remove"]` - 子 OU とアカウントは、子ポリシーで指定された演算子のみを使用できます。この子制御演算子では、1 つ以上の値設定演算子を指定できます。
- `"@@operators_allowed_for_child_policies": ["@@none"]` - 子 OU とアカウントは、ポリシーの演算子を使用できません。この演算子を使用して、子ポリシーがこれらの値を追加、付加、または削除できないように、親ポリシーで定義されている値で効果的に口ックできます。

> **Note**
>
> 継承された子制御演算子によって演算子の使用が制限されている場合、子ポリシーでそのルールを元に戻すことはできません。親ポリシーに子制御演算子を含めると、すべての子ポリシーの値設定演算子が制限されます。

継承の例

これらの例は、親タグポリシーと子タグポリシーがマージされてアカウントの有効なタグポリシーになるのを示すことにより、ポリシーの継承がどのように機能するかを示しています。

この例では、次の図に示す組織構造があることを前提としています。



例

- 例 1: 子ポリシーによるタグ値のみの上書きを許可する
- 例 2: 継承されたタグに新しい値を追加する
- 例 3: 継承されたタグから値を削除する
- 例 4: 子ポリシーへの変更を制限する
- 例 5: 子制御演算子との競合
- 例 6: 同じ階層レベルで値を追加した場合の競合

例 1: 子ポリシーによるタグ値のみの上書きを許可する

次のタグポリシーは、CostCenter タグキーと 2 つの許容値 (Development および Support) を定義します。組織ルートにアタッチすると、タグポリシーは組織内のすべてのアカウントで有効になります。

ポリシー A - 組織ルートのタグポリシー

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@assign": [
                    "Development",
                    "Support"
                ]
            }
        }
    }
}
```

OU1 のユーザーにキーに別のタグ値を使用してもらい、特定のリソースタイプに対してタグポリシーを適用するようにしたいとします。ポリシー A では、許可される子制御演算子が指定されていないため、すべての演算子が許可されます。@@assign 演算子を使用して、次のようなタグポリシーを作成し、OU1 にアタッチできます。

ポリシー B - OU1 タグポリシー

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@assign": [
                    "Sandbox"
                ]
            },
```

```
            "enforced_for": {
                "@@assign": [
                    "redshift:*",
                    "dynamodb:table"
                ]
            }
        }
    }
}
```

タグの @@assign 演算子を指定すると、ポリシー A とポリシー B が統合されてアカウントの有効なタグポリシーが形成された場合に、以下のようになります。

- ポリシー B は、親ポリシーであるポリシー A で指定された 2 つのタグ値を上書きします。その結果、Sandbox は、CostCenter タグキーの準拠値のみとなります。
- enforced_for を追加することで、すべての Amazon Redshift リソースと Amazon DynamoDB テーブルで、指定されたタグ値として CostCenter タグを使用する必要があることが指定されます。

図に示すように、OU1 には 2 つのアカウント (111111111111 と 222222222222) が含まれています。

アカウント 111111111111 および 222222222222 に対して有効な、結果として生じるタグポリシー

> **Note**
>
> 表示された有効なポリシーの内容を、新しいポリシーの内容として直接使用することはできません。構文には、他の子ポリシーと親ポリシーのマージを制御するために必要な演算子は含まれていません。有効なポリシーの表示は、マージの結果を把握することのみを目的としています。

```
{
    "tags": {
        "costcenter": {
            "tag_key": "CostCenter",
            "tag_value": [
                "Sandbox"
            ],
            "enforced_for": [
```

```
                "redshift:*",
                "dynamodb:table"
            ]
        }
    }
}
```

例 2: 継承されたタグに新しい値を追加する

組織内のすべてのアカウントで、許容値の短いリストでタグキーを指定する必要がある場合が考えられます。1 つの OU のアカウントでは、リソースの作成時にそれらのアカウントのみが指定できる追加の値を許可することをおすすめします。この例では、@@append 演算子を使用してその方法を指定します。@@append 演算子はアドバンスト機能です。

例 1 と同様、この例も組織ルートのタグポリシーのポリシー A から始まります。

ポリシー A - 組織ルートのタグポリシー

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@assign": [
                    "Development",
                    "Support"
                ]
            }
        }
    }
}
```

この例では、ポリシー C を OU2 にアタッチします。この例の違いは、ポリシー C で @@append 演算子を使用すると、許容可能な値のリストと enforced_for ルールが上書きされるのではなく、追加されることです。

ポリシー C - 値を追加するための OU2 タグポリシー

```
{
    "tags": {
```

```
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@append": [
                    "Marketing"
                ]
            },
            "enforced_for": {
                "@@append": [
                    "redshift:*",
                    "dynamodb:table"
                ]
            }
        }
    }
}
```

ポリシー C を OU2 にアタッチすると、ポリシー A とポリシー C がマージしてアカウントの有効なタグポリシーを形成した時点で次のような効果があります。

- ポリシー C には @@append 演算子が含まれているため、ポリシー A で指定されている許容タグ値のリストを上書きするのではなく、追加できます。
- ポリシー B では、enforced_for を追加することにより、すべての Amazon Redshift リソースと Amazon DynamoDB テーブルで、指定されたタグ値として CostCenter タグを使用する必要があることが指定されます。上書き (@@assign) と追加 (@@append) は、子ポリシーが指定できるものを制限する子制御演算子が親ポリシーに含まれていない場合、同じ効果があります。

図に示すように、OU2 には 1 つのアカウント (999999999999) が含まれています。ポリシー A とポリシー C をマージして、アカウント 999999999999 に対する有効なタグポリシーを作成します。

アカウント 999999999999 に対する有効なタグポリシー

> **Note**
>
> 表示された有効なポリシーの内容を、新しいポリシーの内容として直接使用することはできません。構文には、他の子ポリシーと親ポリシーのマージを制御するために必要な演算子は含まれていません。有効なポリシーの表示は、マージの結果を把握することのみを目的としています。

```
{
    "tags": {
        "costcenter": {
            "tag_key": "CostCenter",
            "tag_value": [
                "Development",
                "Support",
                "Marketing"
            ],
            "enforced_for": [
                "redshift:*",
                "dynamodb:table"
            ]
        }
    }
}
```

例 3: 継承されたタグから値を削除する

組織にアタッチされたタグポリシーで、アカウントで使用するよりも多くのタグ値が定義されている場合があります。この例では、@@remove 演算子を使用してタグポリシーを変更する方法について説明します。@@remove はアドバンスト機能です。

他の例と同様、この例は組織ルートのタグポリシーのポリシー A から始まります。

ポリシー A - 組織ルートのタグポリシー

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@assign": [
                    "Development",
                    "Support"
                ]
            }
        }
    }
}
```

この例では、ポリシー D をアカウント 999999999999 にアタッチします。

ポリシー D - 値を削除するためのアカウント 999999999999 のタグポリシー

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@remove": [
                    "Development",
                    "Marketing"
                ],
                "enforced_for": {
                    "@@remove": [
                        "redshift:*",
                        "dynamodb:table"
                    ]
                }
            }
        }
    }
}
```

ポリシー D をアカウント 999999999999 にアタッチすると、ポリシー A、ポリシー C、およびポリシー D をマージして有効なタグポリシーを形成した時点で、次の効果があります。

- 前述の例をすべて実行した場合、ポリシー B、C、および C は A の子ポリシーになっています。ポリシー B は OU1 にのみアタッチされるため、アカウント 9999999999999 には影響しません。
- アカウント 9999999999999 の場合、CostCenter タグキーの許容値は Support のみです。
- CostCenter タグキーに対してコンプライアンスは強制されません。

アカウント 999999999999 の新しい有効なタグポリシー

> **Note**
>
> 表示された有効なポリシーの内容を、新しいポリシーの内容として直接使用することはできません。構文には、他の子ポリシーと親ポリシーのマージを制御するために必要な演算子は

含まれていません。有効なポリシーの表示は、マージの結果を把握することのみを目的としています。

```
{
    "tags": {
        "costcenter": {
            "tag_key": "CostCenter",
            "tag_value": [
                "Support"
            ]
        }
    }
}
```

後で OU2 にアカウントを追加すると、追加したアカウントの有効なタグポリシーはアカウント 999999999999 とは異なるものになります。これは、より制限の厳しいポリシー D がアカウントレベルでのみアタッチされ、OU にはアタッチされていないためです。

例 4: 子ポリシーへの変更を制限する

子ポリシーの変更を制限する必要がある場合があります。この例では、子制御演算子を使用してそれを行う方法について説明します。

この例では、新しい組織ルートタグポリシーから開始し、タグポリシーがまだ組織エンティティにアタッチされていないことを前提としています。

ポリシー E – 子ポリシーの変更を制限する組織ルートのタグポリシー

```
{
    "tags": {
        "project": {
            "tag_key": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "@@assign": "Project"
            },
            "tag_value": {
                "@@operators_allowed_for_child_policies": ["@@append"],
                "@@assign": [
                    "Maintenance",
                    "Escalations"
```

```
                ]
            }
        }
    }
}
```

ポリシー E を組織ルートにアタッチすると、子ポリシーが Project タグキーを変更できなくなります。ただし、子ポリシーはタグ値を上書きまたは追加できます。

その後、次のポリシー F を OU にアタッチすると仮定します。

ポリシー F - OU のタグポリシー

```
{
    "tags": {
        "project": {
            "tag_key": {
                "@@assign": "PROJECT"
            },
            "tag_value": {
                "@@append": [
                    "Escalations - research"
                ]
            }
        }
    }
}
```

ポリシー E とポリシー F をマージした時点で、OU のアカウントに次のような影響があります。

- ポリシー F は、ポリシー E の子ポリシーです。
- ポリシー F は大文字と小文字の取り扱いを変更しようとしますが、できません。これは、ポリシー E にタグキーの "@@operators_allowed_for_child_policies": ["@@none"] 演算子が含まれているためです。
- ただし、ポリシー F はキーのタグ値を追加できます。これは、ポリシー E にタグ値の "@@operators_allowed_for_child_policies": ["@@append"] が含まれているためです。

OU のアカウントに対する有効なポリシー

> **Note**
>
> 表示された有効なポリシーの内容を、新しいポリシーの内容として直接使用することはできません。構文には、他の子ポリシーと親ポリシーのマージを制御するために必要な演算子は含まれていません。有効なポリシーの表示は、マージの結果を把握することのみを目的としています。

```
{
    "tags": {
        "project": {
            "tag_key": "Project",
            "tag_value": [
                "Maintenance",
                "Escalations",
                "Escalations - research"
            ]
        }
    }
}
```

例 5: 子制御演算子との競合

子制御演算子は、組織階層の同じレベルでアタッチされたタグポリシー内に存在することができます。この場合、ポリシーがマージされてアカウントの有効なポリシーを形成するときに、許可された演算子の共通部分が使用されます。

ポリシー G とポリシー H が組織ルートにアタッチされていると仮定します。

ポリシー G - 組織ルートのタグポリシー 1

```
{
    "tags": {
        "project": {
            "tag_value": {
                "@@operators_allowed_for_child_policies": ["@@append"],
                "@@assign": [
                    "Maintenance"
                ]
            }
        }
```

```
    }
}
```

ポリシー H - 組織ルートのタグポリシー 2

```
{
    "tags": {
        "project": {
            "tag_value": {
                "@@operators_allowed_for_child_policies": ["@@append", "@@remove"]
            }
        }
    }
}
```

この例では、組織ルートの 1 つのポリシーで、タグキーの値のみを追加できるということが定義されています。組織ルートにアタッチされたもう 1 つのポリシーは、子ポリシーが値の追加と削除の両方を行うことを許可します。これら 2 つのアクセス許可の共通部分が子ポリシーに使用されます。その結果、子ポリシーは値を追加できますが、値は削除できません。したがって、子ポリシーはタグ値のリストに値を追加できますが、Maintenance の値を削除することはできません。

例 6: 同じ階層レベルで値を追加した場合の競合

各組織エンティティに複数のタグポリシーをアタッチできます。これを行うと、同じ組織エンティティにアタッチされているタグポリシーに競合する情報が含まれる場合があります。ポリシーは、組織エンティティにアタッチされた順序に基づいて評価されます。最初に評価されるポリシーを変更するには、ポリシーをデタッチしてから再度アタッチします。

ポリシー J が最初に組織ルートにアタッチされ、その次にポリシー K が組織ルートにアタッチされると仮定します。

ポリシー J - 組織ルートにアタッチされた最初のタグポリシー

```
{
    "tags": {
        "project": {
            "tag_key": {
                "@@assign": "PROJECT"
            },
            "tag_value": {
                "@@append": ["Maintenance"]
            }
```

```
        }
    }
}
```

ポリシー K - 組織ルートにアタッチされた 2 番目のタグポリシー

```
{
    "tags": {
        "project": {
            "tag_key": {
                "@@assign": "project"
            }
        }
    }
}
```

この例では、PROJECT タグキーを定義したポリシーが最初に組織ルートにアタッチされたため、このタグキーが有効なタグポリシーで使用されます。

ポリシー JK - アカウントの有効なタグポリシー

アカウントの有効なポリシーは次のようになります。

> **Note**
>
> 表示された有効なポリシーの内容を、新しいポリシーの内容として直接使用することはできません。構文には、他の子ポリシーと親ポリシーのマージを制御するために必要な演算子は含まれていません。有効なポリシーの表示は、マージの結果を把握することのみを目的としています。

```
{
    "tags": {
        "project": {
            "tag_key": "PROJECT",
            "tag_value": [
                "Maintenance"
            ]
        }
    }
}
```

AI サービスのオプトアウトポリシー

AWS Amazon Rekognition 、Amazon 、Amazon Transcribe CodeWhisperer、Contact Lens for Amazon Connect などの人工知能 (AI) サービスは、他のサービスの開発と継続的な改善のために、これらのサービスによって処理された顧客コンテンツを保存して使用する場合があります AWS 。お客様は AWS 、サービスの改善のためにコンテンツを保存または使用することをオプトアウトできます。

組織が使用する各 に対してこの設定 AWS アカウント を個別に設定する代わりに、組織のメンバーであるすべてのアカウントに対して設定の選択を適用する組織ポリシーを設定できます。コンテンツの保存と使用をオプトアウトする AI サービスは、個別に指定することも、該当するサービスすべてを一律に指定することも可能です。各アカウントに適用される有効なポリシーを問い合わせ、設定の効果を確認できます。

> **Note**
>
> AWS 人工知能 (AI) サービスは、サービスの改善のためにデータ AWS の使用をオプトアウトした場合でも、サービスを提供するためにコンテンツを保存する必要がある場合があります。詳細については、ご使用の AI サービスのドキュメントを参照してください。

AI サービスのオプトアウトポリシーを使用する際の考慮事項

オプトアウトは、 AWS リージョン を除くすべての に適用されます。 AWS GovCloud (US)

サービスのオプトインまたはオプトアウト設定を指定すると、その設定はグローバルになり、 AWS リージョン を除くすべての に適用されます AWS GovCloud (US) Regions。1 つの AWS リージョン内で行った値の設定は、他のすべてのリージョンにレプリケートされます。

オプトアウトすると、関連するすべての履歴コンテンツが削除されます。

AWS AI サービスによるコンテンツの使用をオプトアウトすると、そのサービスは、オプションを設定する AWS 前に と共有された関連する履歴コンテンツをすべて削除します。この削除は、サービス機能の提供に必要のない保存データに限定する必要があります。

AI サービスのオプトアウトポリシーの利用開始

人工知能 (AI) サービスのオプトアウトポリシーの利用を開始するステップは以下のとおりです。

1. AI サービスのオプトアウトポリシーを組織で有効にする。

2. AI サービスのオプトアウトポリシーを作成する。
3. 組織ルート、OU、またはアカウントに AI サービスのオプトアウトポリシーをアタッチする。
4. アカウントに適用される集約された有効な AI サービスのオプトアウトポリシーを確認する。

これらのすべてのステップでは、 AWS Identity and Access Management 組織の管理アカウントで (IAM) ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザー (推奨されません) としてサインインします。

その他の情報

- AI サービスのオプトアウトポリシーの構文と例を確認する

AI サービスのオプトアウトポリシーの作成、更新、削除

このトピックの内容

- 組織の AI サービスのオプトアウトポリシーを有効化すると、ポリシーの作成が可能になります。
- オプトアウト要件を変更した場合は、既存のポリシーを更新します。
- ポリシーが不要になった場合、すべての組織単位 (OU) およびアカウントからポリシーをデタッチした後、ポリシーを削除できます。

AI サービスのオプトアウトポリシーの作成

> **最小アクセス許可**
>
> AI サービスのオプトアウトポリシーを作成するには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:CreatePolicy`

AWS Management Console

AI サービスのオプトアウトポリシーを作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。

2. AI サービスのオプトアウトポリシーページで、[Create policy] (ポリシーの作成) を選択します。
3. [Create new AI services opt-out policy] (新しい AI サービスのオプトアウトポリシーの作成) ページで、ポリシー名とポリシーの説明を入力します。
4. (オプション) [Add tag] (タグの追加) を選択してキーとオプションの値を入力することで、ポリシーに 1 つ以上のタグを追加できます。値を空白のままにすると、空の文字列が設定され、null にはなりません。1 つのポリシーに最大 50 個のタグをアタッチできます。詳細については、「AWS Organizations リソースのタグ付け」を参照してください。
5. [JSON] タブで、ポリシーテキストを入力するか貼り付けます。AI サービスのオプトアウトポリシーの構文について詳しくは、AI サービスのオプトアウトポリシーの構文と例 を参照してください。開始点として使用できるサンプルポリシーについては、AI サービスのオプトアウトポリシーの例 を参照してください。
6. ポリシーの編集が完了したら、ページの右下隅の [Create policy] (ポリシーの作成) を選択します。

AWS CLI & AWS SDKs

AI サービスのオプトアウトポリシーを作成するには

次のいずれかを使用して、タグポリシーを作成できます。

- AWS CLI: create-policy

1. 次のような AI サービスのオプトアウトポリシーを作成し、テキストファイルとして保存します。「optOut」と「optIn」では大文字と小文字が区別されます。

```
{
    "services": {
        "default": {
            "opt_out_policy": {
                "@@assign": "optOut"
            }
        },
        "rekognition": {
            "opt_out_policy": {
                "@@assign": "optIn"
            }
        }
    }
```

```
}
```

この AI サービスのオプトアウトポリシーは、ポリシーの影響を受けるすべてのアカウントが、Amazon Rekognition を除くすべての AI サービスからオプトアウトされるように指定します。

2. JSON ポリシーファイルをインポートして、組織内に新しいポリシーを作成します。この例では、先に扱った JSON ファイルは `policy.json` という名前になっています。

```
$ aws organizations create-policy \
    --type AISERVICES_OPT_OUT_POLICY \
    --name "MyTestPolicy" \
    --description "My test policy" \
    --content file://policy.json
{
    "Policy": {
        "Content": "{\"services\":{\"default\":{\"opt_out_policy\":{\"@@assign
\":\"optOut\"}},\"rekognition\":{\"opt_out_policy\":{\"@@assign\":\"optIn
\"}}}}",
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5"
            "Arn": "arn:aws:organizations::o-aa111bb222:policy/
aiservices_opt_out_policy/p-i9j8k7l6m5",
            "Description": "My test policy",
            "Name": "MyTestPolicy",
            "Type": "AISERVICES_OPT_OUT_POLICY"
        }
    }
}
```

- AWS SDKs CreatePolicy

次のステップ

AI サービスのオプトアウトポリシーを作成したら、オプトアウト設定を有効にすることができます。そのためには、組織ルート、組織単位 (OUs)、組織 AWS アカウント 内、またはこれらすべての組み合わせにポリシーをアタッチできます。

AI サービスのオプトアウトポリシーの更新

> **最小アクセス許可**
>
> AI サービスのオプトアウトポリシーを更新するには、次のアクションを実行するアクセス許可が必要です。
>
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:UpdatePolicy`。
> - 指定したポリシー (または "*") の Amazon リソースネーム (ARN) を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:DescribePolicy`。

AWS Management Console

AI サービスのオプトアウトポリシーを更新するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AI サービスのオプトアウトポリシーページで、更新するポリシーの名前を選択します。
3. ポリシーの詳細ページで、[Edit policy] (ポリシーの編集) を選択します。
4. 新しいポリシー名とポリシーの説明を入力するか、JSON ポリシーテキストを編集します。AI サービスのオプトアウトポリシーの構文について詳しくは、AI サービスのオプトアウトポリシーの構文と例 を参照してください。開始点として使用できるサンプルポリシーについては、AI サービスのオプトアウトポリシーの例 を参照してください。
5. ポリシーの更新が完了したら、[変更を保存] を選択します。

AWS CLI & AWS SDKs

ポリシーを更新するには

次のいずれかを使用して、ポリシーを更新できます。

- AWS CLI: update-policy

  次の例では、AI サービスのオプトアウトポリシーの名前を変更しています。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --name "Renamed policy"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
aiservices_opt_out_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Type": "AISERVICES_OPT_OUT_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"services\":{\"default\":{\"opt_out_policy\":
  ....TRUNCATED FOR BREVITY...   :{\"@@assign\":\"optIn\"}}}}"
    }
}
```

次の例では、AI サービスのオプトアウトポリシーの説明を追加、変更しています。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
aiservices_opt_out_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "AISERVICES_OPT_OUT_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"services\":{\"default\":{\"opt_out_policy\":
  ....TRUNCATED FOR BREVITY...   :{\"@@assign\":\"optIn\"}}}}"
    }
}
```

次の例では、AI サービスのオプトアウトポリシーにアタッチされた JSON ポリシードキュメントを変更しています。この例では、次のようなテキストを含む policy.json というファイルから、内容が取得されています。

```
{
    "services": {
        "default": {
            "opt_out_policy": {
                "@@assign": "optOut"
            }
        },
        "comprehend": {
            "opt_out_policy": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "@@assign": "optOut"
            }
        },
        "rekognition": {
            "opt_out_policy": {
                "@@assign": "optIn"
            }
        }
    }
}
```

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --content file://policy.json
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
aiservices_opt_out_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "AISERVICES_OPT_OUT_POLICY",
            "AwsManaged": false
        },
         "Content": "{\n\"services\": {\n\"default\": {\n\"   ....TRUNCATED FOR
 BREVITY....    ": \"optIn\"\n}\n}\n}\n}\n"}
```

```
}
```

- AWS SDKs UpdatePolicy

AI サービスのオプトアウトポリシーにアタッチされたタグの編集

組織の管理アカウントにサインインすると、AI サービスのオプトアウトポリシーにアタッチされたタグの追加と削除を行うことができます。タグ付けの詳細については、「AWS Organizations リソースのタグ付け」を参照してください。

> **最小アクセス許可**
>
> AWS 組織の AI サービスのオプトアウトポリシーにアタッチされたタグを編集するには、次のアクセス許可が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:DescribePolicy` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:TagResource`
> - `organizations:UntagResource`

AWS Management Console

AI サービスのオプトアウトポリシーにアタッチされたタグを編集するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AI サービスのオプトアウトポリシーページで、タグを編集するポリシーの名前を選択します。
3. 選択したポリシーの詳細ページで、[Tags] (タグ) タブ、[Manage tags] (タグ管理) の順に選択します。
4. このページでは次のアクションを実行できます。

- 古い値に上書きして新しい値を入力し、任意のタグの値を編集します。キーは変更できません。キーを変更するには、古いキーを持つタグを削除し、新しいキーを持つタグを追加する必要があります。
- [Remove] (削除) を選択すると、既存のタグが削除されます。
- 新しいタグのキーと値のペアを追加します。[Add tag] (タグの追加) を選択し、表示されたボックスに新しいキー名とオプションの値を入力します。[Value] (値) ボックスを空白のままにすると、値は空の文字列に設定され、`null` にはなりません。

5. 必要な追加、削除、編集をすべて終えたら、[Save changes] (変更の保存) を選択します。

AWS CLI & AWS SDKs

AI サービスのオプトアウトポリシーにアタッチされたタグを編集するには

AI サービスのオプトアウトポリシーにアタッチされたタグを編集するには、次のいずれかのコマンドを使用します。

- AWS CLI: tag-resource および untag-resource
- AWS SDKsTagResourceおよび UntagResource

AI サービスのオプトアウトポリシーの削除

組織の管理アカウントにサインインすると、組織に不要になったポリシーを削除できます。

ポリシーを削除するには、まずそのポリシーをすべての添付エンティティからデタッチする必要があります。

> 最小アクセス許可
>
> ポリシーを削除するには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:DescribePolicy` (コンソールのみ - ポリシーに移動するために使用)
> - `organizations:DeletePolicy`

AWS Management Console

AI サービスのオプトアウトポリシーを削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AI サービスのオプトアウトポリシーページで、削除するポリシーの名前を選択します。
3. 削除するポリシーは、まず、すべてのルート、OU、アカウントからデタッチする必要があります。[Targets] (ターゲット) タブを選択し、[Targets] (ターゲット) リストの各ルート、OU、アカウントの横にあるラジオボタンをクリックしてから、[Detach] (デタッチ) を選択します。確認ダイアログボックスで、[Detach] (デタッチ) を選択します。すべてのターゲットを削除するまで繰り返します。
4. ページの上部で、[Delete] (削除) を選択します。
5. 確認ダイアログボックスで、ポリシーの名前を入力し、[Delete] (削除) を選択します。

AWS CLI & AWS SDKs

AI サービスのオプトアウトポリシーを削除するには

以下のコード例は、`DeletePolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Deletes an existing AWS Organizations policy.
```

```
    /// </summary>
    public class DeletePolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and then uses it to
        /// delete the policy with the specified policyId.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var policyId = "p-00000000";

            var request = new DeletePolicyRequest
            {
                PolicyId = policyId,
            };

            var response = await client.DeletePolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted Policy: {policyId}.");
            }
            else
            {
                Console.WriteLine($"Could not delete Policy: {policyId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeletePolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDeletePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def delete_policy(policy_id, orgs_client):
    """
    Deletes a policy.

    :param policy_id: The ID of the policy to delete.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.delete_policy(PolicyId=policy_id)
        logger.info("Deleted policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_id)
        raise
```

- API の詳細については、 DeletePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AI サービスのオプトアウトポリシーのアタッチとデタッチ

人工知能 (AI) サービスのオプトアウトポリシーは、組織全体、組織単位 (OU)、個々のアカウントで使用できます。AI サービスのオプトアウトポリシーの適用の内容は、アタッチする組織要素によって異なります。

- AI サービスのオプトアウトポリシーを組織のルート にアタッチすると、そのポリシーは管理アカウントを含むすべての OUs とアカウントに適用されます。
- AI サービスのオプトアウトポリシーを OU にアタッチすると、ポリシーはその OU またはその子 OU に属するアカウントに適用されます。これらのアカウントには、組織ルートにアタッチされたバックアップポリシーも適用されます。
- AI サービスのオプトアウトポリシーをアカウントにアタッチすると、ポリシーはそのアカウントにのみ適用されます。このアカウントは、組織のルートにアタッチされたポリシーと、そのアカウントが属する OU にも適用されます。

アカウントがルートおよび親 OU から継承する AI サービスのオプトアウトポリシーと、アカウントに直接アタッチされたポリシーを集約したものが、有効なポリシーとなります。ポリシーを有効なポリシーにマージする方法については、「管理ポリシーの継承を理解する」を参照してください。

> **最小アクセス許可**
>
> AI サービスのオプトアウトポリシーをアタッチするには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:AttachPolicy`

AWS Management Console

AI サービスのオプトアウトポリシーのアタッチには、ポリシーに移動する方法と、ポリシーをアタッチするルート、OU、またはアカウントに移動する方法があります。

ルート、OU、またはアカウントに移動して AI サービスのオプトアウトポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。

2. AWS アカウント ページで、ポリシーをアタッチするルート、OU、またはアカウントを見つけ、名前を選択します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
3. [Policies] (ポリシー) タブの [AI service opt-out policies] (AI サービスのオプトアウトポリシー) の項目で、[Attach] (アタッチ) を選択します。
4. 目的のポリシーを見つけて [Attach policy] (ポリシーのアタッチ) を選択します。

   [Policies] (ポリシー) タブで、アタッチされている AI サービスのオプトアウトポリシーの一覧が更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動して AI サービスのオプトアウトポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AI サービスのオプトアウトポリシーページで、アタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで [Attach] (アタッチ) を選択します。
4. ポリシーをアタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
5. Attach policy] (ポリシーのアタッチ) を選択します。

   [Targets] (ターゲット) タブで、アタッチされている AI サービスのオプトアウトポリシーの一覧が更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

組織のルート、OU、またはアカウントから AI サービスのオプトアウトポリシーをアタッチするには

以下のコード例は、`AttachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to attach an AWS Organizations policy to an organization,
/// an organizational unit, or an account.
/// </summary>
public class AttachPolicy
{
    /// <summary>
    /// Initializes the Organizations client object and then calls the
    /// AttachPolicyAsync method to attach the policy to the root
    /// organization.
    /// </summary>
    public static async Task Main()
    {
        IAmazonOrganizations client = new AmazonOrganizationsClient();
        var policyId = "p-00000000";
        var targetId = "r-0000";

        var request = new AttachPolicyRequest
        {
            PolicyId = policyId,
            TargetId = targetId,
        };

        var response = await client.AttachPolicyAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
```

```
                Console.WriteLine($"Successfully attached Policy ID {policyId} to
 Target ID: {targetId}.");
            }
            else
            {
                Console.WriteLine("Was not successful in attaching the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスAttachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシーを OU にアタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- API の詳細については、「 コマンドリファレンスAttachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def attach_policy(policy_id, target_id, orgs_client):
    """
    Attaches a policy to a target. The target is an organization root, account,
 or
    organizational unit.

    :param policy_id: The ID of the policy to attach.
    :param target_id: The ID of the resources to attach the policy to.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.attach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Attached policy %s to target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't attach policy %s to target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 AttachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに有効になります。

AI サービスのオプトアウトポリシーのデタッチ

組織の管理アカウントにサインインすると、組織ルート、OU、またはアカウントから、アタッチされている AI サービスのオプトアウトポリシーをデタッチすることができます。エンティティから AI

サービスのオプトアウトポリシーをデタッチすると、そのポリシーは、デタッチしたエンティティの影響下にあったいかなるアカウントに対しても適用されなくなります。ポリシーをデタッチするには、次のステップを実行します。

> **最小アクセス許可**
>
> 組織ルート、OU、またはアカウントから AI サービスのオプトアウトポリシーをデタッチするには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:DetachPolicy`

AWS Management Console

AI サービスのオプトアウトポリシーのデタッチには、ポリシーに移動する方法と、ポリシーをデタッチするルート、OU、またはアカウントに移動する方法があります。

ポリシーがアタッチされているルート、OU、またはアカウントに移動して AI サービスのオプトアウトポリシーをデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをデタッチするルート、OU、またはアカウントに移動します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。ルート、OU、またはアカウントの名前を選択します。
3. [Policies] (ポリシー) タブで、デタッチする AI サービスのオプトアウトポリシーの横にあるラジオボタンを選択し、[Detach] (デタッチ) を選択します。
4. 確認ダイアログボックスで、[Detach policy] (ポリシーのデタッチ) を選択します。

   アタッチされている AI サービスのオプトアウトポリシーの一覧が更新されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動して AI サービスのオプトアウトポリシーをデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AI サービスのオプトアウトポリシーページで、ルート、OU、またはアカウントからデタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで、ポリシーをデタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。
4. [Detach] (デタッチ) を選択します。
5. 確認ダイアログボックスで、[Detach] (デタッチ) を選択します。

   アタッチされている AI サービスのオプトアウトポリシーの一覧が更新されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

組織のルート、OU、またはアカウントから AI サービスのオプトアウトポリシーをデタッチするには

以下のコード例は、`DetachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;
```

```
    /// <summary>
    /// Shows how to detach a policy from an AWS Organizations organization,
    /// organizational unit, or account.
    /// </summary>
    public class DetachPolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and uses it to call
        /// DetachPolicyAsync to detach the policy.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new DetachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.DetachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully detached policy with Policy Id:
{policyId}.");
            }
            else
            {
                Console.WriteLine("Could not detach the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDetachPolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111
 --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDetachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def detach_policy(policy_id, target_id, orgs_client):
    """
    Detaches a policy from a target.

    :param policy_id: The ID of the policy to detach.
    :param target_id: The ID of the resource where the policy is currently
 attached.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.detach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Detached policy %s from target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 DetachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに反映されます。

有効な AI サービスのオプトアウトポリシーの表示

組織内のアカウントに対して有効な、人工知能 (AI) サービスの有効なオプトアウトポリシーを特定します。

有効な AI サービスのオプトアウトポリシーとは

有効な AI サービスのオプトアウトポリシーとは、AWS アカウント に適用される最終的なルールのことです。これは、アカウントが継承するすべての AI サービスのオプトアウトポリシーと、アカウントに直接アタッチされているすべての AI サービスのオプトアウトポリシーを集約したものです。組織ルートに AI サービスのオプトアウトポリシーをアタッチすると、組織内のすべてのアカウントに適用されます。OU に AI サービスのオプトアウトポリシーをアタッチすると、その OU に属するすべてのアカウントと OU に適用されます。アカウントに直接ポリシーをアタッチすると、その AWS アカウント にのみ適用されます。

例えば、組織ルートにアタッチされた AI サービスのオプトアウトポリシーで、すべての AWS 機械学習サービスによるコンテンツの使用を組織内のすべてのアカウントがオプトアウトするよう指定したとします。一方、あるメンバーアカウントに直接アタッチされた別の AI サービスのオプトアウトポリシーでは、Amazon Rekognition によるコンテンツの使用だけはオプトインするよう指定されています。これらの AI サービスのオプトアウトポリシーが組み合わされ、有効な AI サービスのオプトアウトポリシーが構成されます。結果として、組織内のすべてのアカウントは、すべての AWS サービスからオプトアウトされ、例外として、1 つのアカウントでのみ Amazon Rekognition をオプトインします。

複数のポリシーが集約され、最終的に有効なポリシーが構成される仕組みについては、管理ポリシーの継承を理解する を参照してください。

有効な AI サービスのオプトアウトポリシーを表示する方法

アカウントの有効な AI サービスのオプトアウトポリシーは、AWS Management Console、AWS API 、AWS Command Line Interface のいずれかを使用して表示できます。

最小アクセス許可

アカウントの有効な AI サービスのオプトアウトポリシーを表示するには、次のアクションを実行するアクセス許可が必要です。

- `organizations:DescribeEffectivePolicy`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

アカウントの有効な AI サービスのオプトアウトポリシーを表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページで、有効な AI サービスのオプトアウトポリシーを表示するアカウントの名前を選択します。場合によっては、目的のアカウントを見つけるには OU を展開 (▶ を選択) する必要があります。
3. [Policies] (ポリシー) タブの [AI services opt-out policies] (AI サービスのオプトアウトポリシー) セクションで、[View the effective AI policy for this AWS アカウント] (この の有効な AI ポリシーを表示する) を選択します。

   指定したアカウントに適用されている有効なポリシーがコンソールに表示されます。

   Note

   有効なポリシーをコピーアンドペーストして、大きな変更を加えずに別の AI サービスのオプトアウトポリシーの JSON として使用することはできません。AI サービスのオプトアウトポリシードキュメントには、各設定を最終的な有効なポリシーにマージする方法を指定する継承演算子を含める必要があります。

AWS CLI & AWS SDKs

アカウントの有効な AI サービスのオプトアウトポリシーを表示するには

次のいずれかを使用して、有効な AI サービスのオプトアウトポリシーを表示できます。

- AWS CLI: describe-effective-policy

  次の例は、アカウントの有効な AI サービスのオプトアウトポリシーを示しています。

```
$ aws organizations describe-effective-policy \
    --policy-type AISERVICES_OPT_OUT_POLICY \
    --target-id 123456789012
{
    "EffectivePolicy": {
        "PolicyContent": "{\"services\":{\"comprehend\":{\"opt_out_policy\":
\"optOut\"},   ....TRUNCATED FOR BREVITY....   "opt_out_policy\":\"optIn\"}}}",
        "LastUpdatedTimestamp": "2020-12-09T12:58:53.548000-08:00",
        "TargetId": "123456789012",
        "PolicyType": "AISERVICES_OPT_OUT_POLICY"
    }
}
```

- AWS SDK: DescribeEffectivePolicy

AI サービスのオプトアウトポリシーの構文と例

このトピックでは、人工知能 (AI) サービスのオプトアウトポリシーの構文を例を挙げて説明します。

AI サービスのオプトアウトポリシーの構文

AI サービスのオプトアウトポリシーは、JSON のルールに従って構造化されたプレーンテキストファイルです。AI サービスのオプトアウトポリシーの構文は、管理ポリシータイプの構文に従います。この構文の詳しい説明については、「管理ポリシーの継承を理解する」を参照してください。このトピックでは、一般的な構文を AI サービスのオプトアウトポリシータイプの特定の要件に適用することを重点的に扱っています。

> **Important**
>
> このセクションでは、値の大文字と小文字の区別が重要な点として取り上げられます。トピックで説明されたとおりに大文字と小文字を区別し、値を入力するようにしてください。大文字と小文字の区別が不適切だと、ポリシーは機能しません。

次のポリシーは、AI サービスのオプトアウトポリシーの基本構文を示しています。この例がアカウントに直接アタッチされている場合、そのアカウントは、あるサービスで明示的にオプトアウトされ、もう 1 つのサービスではオプトインされます。他のサービスのオプトインとオプトアウトに関しては、より高いレベル (OU またはルートのポリシー) から継承したポリシーに依存します。

```
{
    "services": {
        "rekognition": {
            "opt_out_policy": {
                "@@assign": "optOut"
            }
        },
        "lex": {
            "opt_out_policy": {
                "@@assign": "optIn"
            }
        }
    }
}
```

組織のルートに次のサンプルポリシーポリシーがアタッチされていると仮定します。これにより、組織はデフォルトですべての AI サービスをオプトアウトするよう設定されています。明示的に除外されない限りすべての AI サービスが自動的にこの対象になります。これには、 AWS によって将来デプロイされる AI サービスも例外なく含まれます。子ポリシーを OU、または直接アカウントにアタッチすることで、Amazon Comprehend 以外のすべての AI サービスに関し、この設定を上書きできます。次の例の 2 番目のエントリでは、`@@operators_allowed_for_child_policies` を none に設定して使用し、オーバーライドされないようにする必要があります。例の 3 番目のエントリでは、Amazon Rekognition の除外を組織全体に設定しています。そのサービスに対して組織全体がオプトインされますが、このポリシーでは、子ポリシーによる適切な上書きを許可しています。

```
{
    "services": {
        "default": {
            "opt_out_policy": {
                "@@assign": "optOut"
            }
        },
        "comprehend": {
            "opt_out_policy": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "@@assign": "optOut"
```

```
            }
        },
        "rekognition": {
            "opt_out_policy": {
                "@@assign": "optIn"
            }
        }
    }
}
```

AI サービスのオプトアウトポリシーの構文には、次の要素が含まれます。

- services 要素。AI サービスのオプトアウトポリシーは、この固定名により、JSON を含む最も外側の要素として識別されます。

  AI サービスのオプトアウトポリシーには、services 要素の下に 1 つ以上のステートメントを含めることができます。各ステートメントには以下の要素が含まれます。

  - AWS AI サービスを識別するサービス名キー。次のキー名は、このフィールドにおいて有効な値です。
    - **default** - 現在利用可能なすべての AI サービスを表します。将来追加される AI サービスも、暗黙的かつ自動的に含まれます。
    - awssupplychain
    - chimesdkvoiceanalytics
    - cloudwatch
    - codeguruprofiler
    - codewhisperer
    - comprehend
    - connectamd
    - connectoptimization
    - contactlens
    - datazone
    - entityresolution
    - frauddetector
    - glue
    - guardduty

- lex
- polly
- q
- quicksightq
- rekognition
- securitylake
- textract
- transcribe
- translate

サービス名キーによって識別される各ポリシーステートメントには、次の要素を含めることができます。

- opt_out_policy キー。このキーは必須の要素です。サービス名キーの下に配置できる唯一のキーです。

  opt_out_policy キーには、次のいずれかの値の @@assign 演算子だけを含めることが可能です。

  - optOut - 指定した AI サービスによるコンテンツの使用のオプトアウトを選択します。
  - optIn - 指定した AI サービスによるコンテンツの使用のオプトインを選択します。

  > **メモ**
  >
  > - @@append および @@remove 継承演算子は AI サービスのオプトアウトポリシーに使用できません。
  > - @@enforced_for 演算子は AI サービスのオプトアウトポリシーに使用できません。

- どのレベルでも、@@operators_allowed_for_child_policies 演算子を使用し、親ポリシーによる設定を上書きして子ポリシーが設定できることをコントロールできます。次のいずれかの値を指定できます。
  - @@assign - このポリシーの子ポリシーは、@@assign 演算子を使用して継承された値を別の値で上書きします。
  - @@none - このポリシーの子ポリシーは値を変更できません。

@@operators_allowed_for_child_policies がどのように動作するかは、配置される場所によって異なります。以下の場所を使用できます。

- services キーの下 - 有効なポリシーのサービスリストの追加と変更を子ポリシーに許可するかどうかをコントロールします。
- 特定の AI サービスのキーまたは default キーの下 - この特定のエントリの下にあるキーリストの追加と変更を子ポリシーに許可するかどうかをコントロールします。
- 特定のサービスの opt_out_policies キーの下 - この特定のサービスの設定の変更を子ポリシーに許可するかどうかをコントロールします。

AI サービスのオプトアウトポリシーの例

次のポリシーの例は、情報提供のみを目的としています。

例 1: 組織内のすべてのアカウントで、すべての AI サービスをオプトアウトする

次の例は、組織内のアカウントで AI サービスをオプトアウトするよう、組織のルートにアタッチできるポリシーを示しています。

> **Tip**
>
> 例の右上隅にあるコピーボタンを使用して例をコピーした場合、行番号は除外され、そのまま貼り付けることができます。

```
    | {
    |     "services": {
[1] |         "@@operators_allowed_for_child_policies": ["@@none"],
    |         "default": {
[2] |             "@@operators_allowed_for_child_policies": ["@@none"],
    |             "opt_out_policy": {
[3] |                 "@@operators_allowed_for_child_policies": ["@@none"],
    |                 "@@assign": "optOut"
    |             }
    |         }
    |     }
    | }
```

- [1] - services の下の "@@operators_allowed_for_child_policies": ["@@none"] により、すべての子ポリシーに対し、個別のサービス用に新しいセクションを追加することを禁止しています。存在できるのは、すでにある default セクションだけです。Default は、「すべての AI サービス」を表すプレースホルダです。
- [2] - default の下の "@@operators_allowed_for_child_policies": ["@@none"] により、すべてのポリシーに対し、新しいセクションを追加することを禁止しています。存在できるのは、すでにある opt_out_policy セクションだけです。
- [3] - opt_out_policy の下の "@@operators_allowed_for_child_policies": ["@@none"] により、子ポリシーによる optOut 設定の値の変更、および設定の追加を禁止しています。

例 2: 組織のデフォルト設定をすべてのサービスに適用しつつ、子ポリシーによるサービスごとの設定の上書きを許可する

次のサンプルポリシーでは、すべての AI サービスを対象に、組織全体のデフォルトを設定しています。default の値により、子ポリシーによるサービス default (すべての AI サービスのプレースホルダ) の optOut 値の変更を禁止しています。このポリシーをルートまたは OU にアタッチして親ポリシーとして適用した場合、子ポリシーでは、2 つ目のポリシーに示すように、サービスごとにオプトアウト設定を変更できます。

- services キーの下に "@@operators_allowed_for_child_policies": ["@@none"] を配置していないため、個々のサービス用に新しいセクションを追加することを子ポリシーに許可しています。
- default の下の "@@operators_allowed_for_child_policies": ["@@none"] により、すべてのポリシーに対し、新しいセクションを追加することを禁止しています。存在できるのは、すでにある opt_out_policy セクションだけです。
- opt_out_policy の下の "@@operators_allowed_for_child_policies": ["@@none"] により、子ポリシーによる optOut 設定の値の変更、および設定の追加を禁止しています。

組織ルートのユーザー AI サービスのオプトアウト親ポリシー

```
{
    "services": {
        "default": {
            "@@operators_allowed_for_child_policies": ["@@none"],
            "opt_out_policy": {
                "@@operators_allowed_for_child_policies": ["@@none"],
```

```
                "@@assign": "optOut"
            }
        }
    }
}
```

次のサンプルポリシーは、先に挙げたサンプルポリシーが組織ルートまたは親 OU にアタッチされており、その親ポリシーの影響を受けるアカウントに、このサンプルがアタッチされることを前提としています。デフォルトのオプトアウト設定を上書きし、Amazon Lex サービスのみに明示的にオプトインします。

AI サービスのオプトアウトポリシー (子)

```
{
    "services": {
        "lex": {
            "opt_out_policy": {
                "@@assign": "optIn"
            }
        }
    }
}
```

結果として得られる の有効なポリシー AWS アカウント は、アカウントが Amazon Lex にのみオプトインし、親ポリシーから継承されたオプトアウト設定のために他のすべての AWS AI サービスをdefaultオプトアウトすることです。

例 3: 単一のサービスに対して組織全体の AI サービスのオプトアウトポリシーを定義する

以下の例では、AI サービスのオプトアウトポリシーで単一の AI サービスに対する optOut 設定を定義しています。このポリシーが組織のルートにアタッチされている場合、すべての子ポリシーに対し、このサービスの optOut 設定の上書きが禁止されます。その他のサービスはこのポリシーの適用対象外ですが、他の OU またはアカウントの子ポリシーの影響を受ける可能性はあります。

```
{
    "services": {
        "rekognition": {
            "opt_out_policy": {
                "@@assign": "optOut",
                "@@operators_allowed_for_child_policies": ["@@none"]
            }
```

```
        }
    }
}
```

バックアップポリシー

AWS Backup では、 AWS リソースのバックアップ方法を定義するバックアッププランを作成できます。プランのルールには、バックアップの頻度、バックアップが発生する時間枠、バックアップするリソース AWS リージョン を含む 、バックアップを保存するボールトなど、さまざまな設定が含まれます。その後、タグを使用して識別された AWS リソースのグループにバックアッププランを適用できます。また、ユーザーに代わってバックアップオペレーションを実行するアクセス許可を付与 AWS Backup する AWS Identity and Access Management (IAM) ロールも特定する必要があります。

のバックアップポリシー AWS Organizations は、これらのすべての要素を JSON テキストドキュメントに結合します。バックアップポリシーは、ルート、組織単位 (OU)、個々のアカウントなど、組織構造の任意の要素にアタッチできます。Organizations は継承ルールに基づき、組織のルート、親 OU、アカウントにアタッチされたポリシーを集約します。これにより、各アカウントに対して 有効なバックアップポリシー が生成されます。この有効なポリシーは、 リソースを自動的にバックアップ AWS Backup する方法を指示します AWS 。

バックアップポリシーを使用すると、組織が必要とするあらゆるレベルでリソースのバックアップをきめ細かく制御できます。例えば、組織のルートにアタッチされたポリシーで、すべての Amazon DynamoDB テーブルをバックアップするよう指定できます。このポリシーには、デフォルトのバックアップ頻度を含めることができます。その後、各 OU の要件に従ってバックアップ頻度を上書きするバックアップポリシーを OU にアタッチできます。例えば、`Developers` OU ではバックアップ頻度を週に 1 回指定し、`Production` OU では 1 日に 1 回指定することができます。

リソースを正常にバックアップするために必要な情報の一部だけを個別に含む部分的なバックアップポリシーを作成できます。これらのポリシーを下位レベルの OU とアカウントによって継承されることを意図して、ルートや親 OU などの組織ツリーのさまざまな部分にアタッチできます。Organizations の継承ルールに基づき、アカウントのすべてのポリシーを組み合わせて構成される有効なポリシーは、必要な要素をすべて備えている必要があります。そうしないと、ポリシーが有効ではなく、影響を受けるリソースがバックアップされない AWS Backup と見なされます。

> **Important**
>
> AWS Backup は、すべての必須要素を含む完全な有効なポリシーによって呼び出された場合にのみ、成功したバックアップを実行できます。

前述の部分ポリシー戦略は機能しますが、アカウントの有効なポリシーが不完全な場合は、エラーになるか、リソースが正常にバックアップされなくなります。代替戦略として、すべてのバックアップポリシーが単独で完全かつ有効であることが必要であることを検討してください。階層の上位にアタッチされたポリシーによって提供される デフォルト値を使用し、継承子制御演算子を含めることによって、子ポリシーで必要に応じてオーバーライドします。

組織 AWS アカウント 内の各 の有効なバックアッププランは、そのアカウントのイミュータブルなプランとしてコンソールに表示されます AWS Backup 。表示することはできますが、変更することはできません。

がポリシーによって作成されたバックアッププランに基づいてバックアップ AWS Backup を開始すると、 AWS Backup コンソールでバックアップジョブのステータスを確認できます。メンバーアカウントのユーザーは、そのメンバーアカウントのバックアップジョブのステータスとエラーを確認できます。で信頼されたサービスアクセスも有効にすると AWS Backup、組織の管理アカウントのユーザーは、組織内のすべてのバックアップジョブのステータスとエラーを確認できます。詳細については、AWS Backup デベロッパーガイドのクロスアカウント管理の有効化を参照してください。

バックアップポリシーの使用開始

バックアップポリシーの使用を開始するには、次のステップを実行します。

1. バックアップポリシータスクの実行に必要なアクセス許可について説明します
2. バックアップポリシーを使用する際に推奨されるベストプラクティスについて説明します。
3. 組織のバックアップポリシーを有効にします。
4. バックアップポリシーを作成します。
5. バックアップポリシーを組織のルート、OU、またはアカウントにアタッチします。
6. アカウントに適用される有効なバックアップポリシーを組み合わせて表示します。

これらすべてのステップでは、組織の管理アカウントの IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザー (推奨されません) としてサインインします。

その他の情報

- バックアップポリシーの構文について説明し、ポリシーの例を参照します

バックアップポリシーを管理するための前提条件とアクセス許可

このページでは、AWS Organizations のバックアップポリシーを管理するための前提条件と必要なアクセス許可について説明します。

トピック

バックアップポリシーを管理するための前提条件

組織内のバックアップポリシーを管理するには、次のことが必要です。

- 組織で、すべての機能が有効になっている必要があります。
- 組織の管理アカウントにサインインする必要があります。
- AWS Identity and Access Management (IAM) ユーザーまたはロールには、次のセクションに記載されているアクセス許可が必要です。

バックアップポリシーを管理するためのアクセス許可

次のサンプル IAM ポリシーは、組織内のバックアップポリシーを全面的に管理するためのアクセス許可を提供します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ManageBackupPolicies",
            "Effect": "Allow",
            "Action": [
                "organizations:AttachPolicy",
                "organizations:CreatePolicy",
                "organizations:DeletePolicy",
                "organizations:DescribeAccount",
                "organizations:DescribeCreateAccountStatus",
                "organizations:DescribeEffectivePolicy",
                "organizations:DescribeOrganization",
                "organizations:DescribeOrganizationalUnit",
                "organizations:DescribePolicy",
                "organizations:DetachPolicy",
```

```
                "organizations:DisableAWSServiceAccess",
                "organizations:DisablePolicyType",
                "organizations:EnableAWSServiceAccess",
                "organizations:EnablePolicyType",
                "organizations:ListAccounts",
                "organizations:ListAccountsForParent",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListCreateAccountStatus",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:ListParents",
                "organizations:ListPolicies",
                "organizations:ListPoliciesForTarget",
                "organizations:ListRoots",
                "organizations:ListTargetsForPolicy",
                "organizations:UpdatePolicy"
            ],
            "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。

バックアップポリシーを使用する場合のベストプラクティス

AWS では、バックアップポリシーを使用する際に次のベストプラクティスを推奨しています。

バックアップポリシー戦略を決定する

継承およびマージされた不完全な部分でバックアップポリシーを作成し、各メンバーアカウントの完全なポリシーを作成できます。これを行う場合、あるレベルで、そのレベルより低いすべてのアカウントに対する変更の影響を慎重に考慮せずに変更を加えると、不完全な効果的なポリシーになる危険性があります。これを回避するために、すべてのレベルで実装するバックアップポリシーが単独で完全であるようにすることをお勧めします。親ポリシーは、子ポリシーで指定された設定によってオーバーライドできるデフォルトポリシーとして扱います。これにより、子ポリシーが存在しない場合でも、継承されたポリシーは完全であり、デフォルト値が使用されます。子制御継承演算子を使用して、子ポリシーで追加、変更、または削除できる設定を制御できます。

GetEffectivePolicy を使用してバックアップポリシーの変更を検証する

バックアップポリシーを変更した後、変更を行ったレベルより低い代表アカウントの有効なポリシーを確認します。有効なポリシーを表示するには、AWS Management Console を使用する

か、GetEffectivePolicy API 操作または AWS CLI あるいは AWS SDK バリアントのいずれかを使用します。加えた変更が、有効なポリシーに意図した影響を与えていることを確認します。

単純なものから始めて、小さな変更を加える

デバッグを簡素化するには、単純なポリシーから開始し、一度に 1 つの項目を変更します。次の変更を行う前に、各変更の動作と影響を検証します。こうすることで、エラーや予期しない結果が発生した場合に考慮する必要がある変数が減ります。

組織内の他の AWS リージョン とアカウントにバックアップのコピーを保存する

バックアップのコピーを保存しておくと、災害対策の強化につながります。

- 別のリージョン - バックアップのコピーを別の AWS リージョン に追加で保存することで、元のリージョンで偶発的な破損や削除からバックアップを保護できます。ポリシーの `copy_actions` セクションを使用し、バックアッププランが実行されるアカウントの 1 つ以上のリージョンにボールトを指定します。これを行うには、バックアップのコピーを保存するバックアップボールトの ARN を指定する際に `$account` 変数を使用し、アカウントを特定します。`$account` 変数は、バックアップポリシーが実行されているアカウント ID に、実行時に自動的に置き換えられます。
- 別のアカウント - バックアップのコピーを別の AWS アカウント に追加で保存することで、アカウントを侵害する悪意のある人物に対するセキュリティの障壁を追加し、保護を強化できます。ポリシーの `copy_actions` セクションを使用し、組織内の 1 つ以上のアカウントにボールトを指定します。バックアッププランを実行するアカウントとは別にする必要があります。これを行うには、バックアップのコピーを保存するバックアップボールトの ARN を指定する際に実際のアカウント ID 番号を使用し、アカウントを特定します。

ポリシーごとのプラン数を制限する

複数のプランを含むポリシーは、すべてを検証する必要がある多数の出力のため、トラブルシューティングが複雑になります。デバッグとトラブルシューティングを簡素化するために、各ポリシーにはバックアッププランを 1 つだけ含めるようにします。その後、他の要件を満たすために、他のプランにポリシーを追加できます。こうすることで、プランに関する問題が 1 つのポリシーに分離され、他のポリシーとそのプランに関する問題のトラブルシューティングが複雑になるのを防ぐことができます。

スタックセットを使用して必要なバックアップボールトと IAM ロールを作成する

AWS CloudFormation スタックセットの Organizations との統合を使用し、組織内の各メンバーアカウントに必要なバックアップボールトと AWS Identity and Access Management (IAM) ロールを自動

的に作成します。組織内のすべての AWS アカウント で自動的に利用可能になるリソースを含むスタックセットを作成できます。こうすることで、依存関係がすでに満たされていることが保証された状態でバックアッププランを実行できます。詳細については、AWS CloudFormation ユーザーガイドのセルフマネージド型のアクセス許可を持つスタックセットの作成を参照してください。

各アカウントで作成された最初のバックアップを確認して、結果をチェックします。

ポリシーを変更するときは、その変更後に作成された次のバックアップをチェックして、変更が目的の影響を与えたことを確認します。こうすることで、有効なポリシーを確認すると同時に、確実に意図したとおりに AWS Backup がポリシーを解釈してバックアッププランを実装するようにできます。

バックアップポリシーの作成、更新、削除

このトピックの内容

- 組織のバックアップポリシーを有効化すると、ポリシーの作成が可能になります。
- バックアップ要件が変更されると、既存のポリシーを更新できます。
- ポリシーが不要になった場合、すべての組織単位 (OU) およびアカウントからポリシーをデタッチした後、ポリシーを削除できます。

バックアップポリシーの作成

> **最小アクセス許可**
>
> バックアップポリシーを作成するには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:CreatePolicy`

AWS Management Console

バックアップポリシーは、次の 2 つの方法のいずれか AWS Management Console で に作成できます。

- オプションを選択し、JSON ポリシーテキストを生成できるビジュアルエディタ。
- JSON ポリシーテキストを直接作成できるテキストエディタ。

ビジュアルエディタを使用すると、プロセスが簡単になりますが、柔軟性は制限されます。これは、最初のポリシーを作成し、使用に慣れるのに最適な方法です。これらの機能の仕組みを理解し、ビジュアルエディタが提供するものによって制限され始めたら、JSON ポリシーテキストを自分で編集して、ポリシーに高度な機能を追加できます。ビジュアルエディタは、@@assign 値設定演算子のみを使用し、子制御演算子へのアクセスは提供しません。子制御演算子は、JSON ポリシーテキストを手動で編集した場合にのみ追加できます。

バックアップポリシーを作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページで、[Create policy] (ポリシーの作成) を選択します。
3. [Create policy] (ポリシーの作成) ページで、ポリシー名と、オプションでポリシーの説明を入力します。
4. (オプション) [Add tag] (タグの追加) を選択してキーとオプションの値を入力することで、ポリシーに 1 つ以上のタグを追加できます。値を空白のままにすると、空の文字列が設定され、null にはなりません。1 つのポリシーに最大 50 個のタグをアタッチできます。タグ付けの詳細については、「AWS Organizations リソースのタグ付け」を参照してください。
5. この手順で説明するように、[ビジュアルエディタ] を使用してポリシーを構築できます。また、[JSON] タブにポリシーテキストを入力または貼り付けることもできます。バックアップポリシーの構文については、バックアップポリシーの構文と例 を参照してください。

   [ビジュアルエディタ] を使用する場合は、シナリオに適したバックアップオプションを選択します。バックアッププランは 3 つの部分で構成されます。こうしたバックアッププランの要素について詳しくは、AWS Backup デベロッパーガイドのバックアッププランの作成、およびリソースの割り当てを参照してください。

   a. バックアッププランの全般的な説明

      - [バックアッププラン名] には、英数字、ハイフン、下線のみを使用できます。
      - リストから少なくとも 1 つの [バックアッププランリージョン] を選択する必要があります。プランは、選択した でのみリソースをバックアップできます AWS リージョン。

   b. AWS Backup の動作方法とタイミングを指定する 1 つ以上のバックアップルール。各バックアップルールは、次の項目を定義します。

      - バックアップの頻度、およびバックアップを実行できるタイムウィンドウを含むスケジュール。

- 使用するバックアップボールトの名前。[バックアップボールト名] は、英数字、ハイフン、下線のみで構成できます。プランを正常に実行するには、バックアップボールトが存在している必要があります。 AWS Backup コンソールまたは AWS CLI コマンドを使用してボールトを作成します。
- (オプション) 1 つ以上のリージョンにコピールールで、バックアップを他の AWS リージョンのボールトにもコピーします。
- このバックアッププランを実行するたびに作成されるバックアップリカバリポイントに関連付ける 1 つ以上のタグキーと値のペア。
- バックアップがコールドストレージに移行するタイミングとバックアップの期限を指定するライフサイクルオプション。

[Add rule] (ルールの追加) を選択し、必要な各ルールをプランに追加します。

バックアップルールの詳細については、AWS Backup デベロッパーガイドのバックアップルールを参照してください。

c. このプランで AWS Backup がバックアップするリソースを指定するリソース割り当て。割り当ては、 AWS Backup がリソースを検索して照合するために使用するタグペアを指定することによって行われます。

- [リソースの割り当て名] には、英数字、ハイフン、下線のみを使用できます。
- AWS Backup 用の [IAM role] (IAM ロール) を指定します。バックアップはこの名前で実行されます。

  コンソールでは、Amazon リソースネーム (ARN) の全体は指定しません。ロール名とロールのタイプを指定するプレフィックスの両方を含める必要があります。通常、プレフィックスは `role` または `service-role` で、ロール名とはスラッシュ (「/」) で区切られます。例えば、`role/MyRoleName` または `service-role/MyManagedRoleName` と入力します。これは、基本となる JSON に保存される際に完全な ARN に自動で変換されます。

  > **Important**
  >
  > 指定した IAM ロールは、ポリシーが適用されるアカウントにすでに存在している必要があります。存在しない場合、バックアッププランはバックアップジョブを正常に開始する可能性がありますが、それらのバックアップジョブは失敗します。

- [Resource tag key] (リソースタグキー) と [Tag values] (タグ値) のペアを 1 つ以上指定し、バックアップするリソースを特定します。複数のタグ値がある場合は、値をカンマで区切ります。

[Add assignment] (割り当てを追加) を選択し、バックアッププランに設定した各リソース割り当てを追加します。

詳細については、AWS Backup デベロッパーガイドのバックアッププランへのリソースの割り当てを参照してください。

6. ポリシーの作成が完了したら、[Create policy] (ポリシーの作成) を選択します。使用可能なバックアップポリシーのリストにポリシーが表示されます。

AWS CLI & AWS SDKs

バックアップポリシーを作成するには

次のいずれかを使用して、バックアップポリシーを作成できます。

- AWS CLI: create-policy

バックアッププランを次のような JSON テキストとして作成し、テキストファイルとして保存します。構文のすべてのルールについては、バックアップポリシーの構文と例 を参照してください。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": { "@@assign": [ "ap-northeast-2", "us-east-1", "eu-
north-1" ] },
            "rules": {
                "Hourly": {
                    "schedule_expression": { "@@assign": "cron(0 5/1 ? * * *)" },
                    "start_backup_window_minutes": { "@@assign": "480" },
                    "complete_backup_window_minutes": { "@@assign": "10080" },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": { "@@assign": "180" },
                        "delete_after_days": { "@@assign": "270" }
                    },
                    "target_backup_vault_name": { "@@assign": "FortKnox" },
                    "copy_actions": {
```

```
                        "arn:aws:backup:us-east-1:$account:backup-vault:secondary-
vault": {
                            "lifecycle": {
                                "move_to_cold_storage_after_days": { "@@assign":
 "10" },
                                "delete_after_days": { "@@assign": "100" }
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": { "@@assign": "arn:aws:iam::$account:role/
MyIamRole" },
                        "tag_key": { "@@assign": "dataType" },
                        "tag_value": { "@@assign": [ "PII" ] }
                    }
                }
            }
        }
    }
}
```

このバックアッププランでは、指定された にあり、値が AWS アカウント dataTypeの タグを持つ、影響を受ける 内のすべてのリソースを AWS Backup AWS リージョン がバックアップするように指定しますPII。

次に、JSON ポリシーファイルをインポートして、組織内に新しいポリシーを作成します。出力のポリシー ARN の末尾にあるポリシー ID を書き留めます。

```
$ aws organizations create-policy \
    --name "MyBackupPolicy" \
    --type BACKUP_POLICY \
    --description "My backup policy" \
    --content file://policy.json{
    "Policy": {
        "PolicySummary": {
            "Arn": "arn:aws:organizations::o-aa111bb222:policy/backup_policy/p-
i9j8k7l6m5",
            "Description": "My backup policy",
```

```
            "Name": "MyBackupPolicy",
            "Type": "BACKUP_POLICY"
        }
        "Content": "...a condensed version of the JSON policy document you
 provided in the file...",
    }
}
```

- AWS SDKs CreatePolicy

次のステップ

バックアップポリシーを作成したら、ポリシーを有効にできます。そのためには、組織ルート、組織単位 (OUs)、組織 AWS アカウント 内、またはこれらすべての組み合わせにポリシーをアタッチできます。

バックアップポリシーの更新

組織の管理アカウントにサインインすると、組織内で変更が必要なポリシーを編集できます。

> 最小アクセス許可
>
> バックアップポリシーを更新するには、次のアクションを実行するアクセス許可が必要です。
>
> - 更新するポリシー (または "*") の ARN を含む同じポリシーステートメントの Resource 要素を持つ organizations:UpdatePolicy。
> - 更新するポリシー (または "*") の ARN を含む同じポリシーステートメントの Resource 要素を持つ organizations:DescribePolicy。

AWS Management Console

バックアップポリシーを更新するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページで、更新するポリシーの名前を選択します。
3. [Edit policy] (ポリシーの編集) を選択します。

4. 新しいポリシー名とポリシーの説明を入力できます。ポリシーの内容を変更するには、[Visual editor] (ビジュアルエディタ) を使用するか、直接 JSON を編集します。
5. ポリシーの更新が完了したら、[変更を保存] を選択します。

AWS CLI & AWS SDKs

バックアップポリシーを更新するには

次のいずれかを使用して、バックアップポリシーを更新できます。

- AWS CLI: update-policy

次の例では、バックアップポリシーの名前を変更しています。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --name "Renamed policy"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
    }
}
```

次の例では、バックアップポリシーの説明を追加、変更しています。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
```

```
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
       "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
    }
}
```

次の例では、バックアップポリシーにアタッチされた JSON ポリシードキュメントを変更しています。この例では、次のようなテキストを含む policy.json というファイルから、内容が取得されています。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": { "@@assign": [ "ap-northeast-2", "us-east-1", "eu-
north-1" ] },
            "rules": {
                "Hourly": {
                    "schedule_expression": { "@@assign": "cron(0 5/1 ? * * *)" },
                    "start_backup_window_minutes": { "@@assign": "480" },
                    "complete_backup_window_minutes": { "@@assign": "10080" },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": { "@@assign": "180" },
                        "delete_after_days": { "@@assign": "270" }
                    },
                    "target_backup_vault_name": { "@@assign": "FortKnox" },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:backup-vault:secondary-
vault": {
                            "lifecycle": {
                                "move_to_cold_storage_after_days": { "@@assign":
 "10" },
                                "delete_after_days": { "@@assign": "100" }
                            }
                        }
                    }
                },
```

```
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": { "@@assign": "arn:aws:iam::$account:role/
MyIamRole" },
                        "tag_key": { "@@assign": "dataType" },
                        "tag_value": { "@@assign": [ "PII" ] }
                    }
                }
            }
        }
    }
}
```

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --content file://policy.json
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
backup_policy/p-i9j8k7l6m5",
            "Name": "Renamed policy",
            "Description": "My new description",
            "Type": "BACKUP_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"plans\":{\"TestBackupPlan\":{\"regions\":{\"@@assign\":
  ....TRUNCATED FOR BREVITY....   "@@assign\":[\"Yes\"]}}}}}}"
}
```

- AWS SDKs UpdatePolicy

バックアップポリシーにアタッチされたタグの編集

組織の管理アカウントにサインインすると、バックアップポリシーにアタッチされたタグの追加と削除を行うことができます。タグ付けの詳細については、「AWS Organizations リソースのタグ付け」を参照してください。

最小アクセス許可

AWS 組織内のバックアップポリシーにアタッチされたタグを編集するには、次のアクセス許可が必要です。

- `organizations:DescribeOrganization` (コンソールのみ - ポリシーに移動するために使用)
- `organizations:DescribePolicy` (コンソールのみ - ポリシーに移動するために使用)
- `organizations:TagResource`
- `organizations:UntagResource`

AWS Management Console

バックアップポリシーにアタッチされたタグを編集するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページ
3. 編集するタグを含むポリシーの名前を選択します。

   ポリシーの詳細ページが表示されます。
4. [Tags (タグ)] タブで、[Manage tags (タグ管理)] を選択します。
5. このページでは次のアクションを実行できます。

   - 古い値に上書きして新しい値を入力し、任意のタグの値を編集します。キーは変更できません。キーを変更するには、古いキーを持つタグを削除し、新しいキーを持つタグを追加する必要があります。
   - [Remove] (削除) を選択すると、既存のタグが削除されます。
   - 新しいタグのキーと値のペアを追加します。[Add tag] (タグの追加) を選択し、表示されたボックスに新しいキー名とオプションの値を入力します。[Value] (値) ボックスを空白のままにすると、値は空の文字列に設定され、`null` にはなりません。
6. 必要な追加、削除、編集をすべて終えたら、[Save changes] (変更の保存) を選択します。

AWS CLI & AWS SDKs

バックアップポリシーにアタッチされたタグを編集するには

バックアップポリシーにアタッチされたタグを編集するには、次のいずれかのコマンドを使用します。

- AWS CLI: tag-resource および untag-resource
- AWS SDKsTagResourceおよび UntagResource

バックアップポリシーの削除

組織の管理アカウントにサインインすると、組織に不要になったポリシーを削除できます。

ポリシーを削除するには、まずそのポリシーをすべての添付エンティティからデタッチする必要があります。

> **最小アクセス許可**
>
> ポリシーを削除するには、次のアクションを実行するアクセス許可が必要です。
>
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:DeletePolicy`。

AWS Management Console

バックアップポリシーを削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページから、削除するバックアップポリシーの名前を選択します。
3. 削除するバックアップポリシーは、まず、すべてのルート、OU、アカウントからデタッチする必要があります。[Targets] (ターゲット) タブを選択し、[Targets] (ターゲット) リストの各ルート、OU、アカウントの横にあるラジオボタンをクリックしてから、[Detach] (デタッチ) を選択します。確認ダイアログボックスで、[Detach] (デタッチ) を選択します。すべてのターゲットを削除するまで繰り返します。

4. ページの上部で、[Delete] (削除) を選択します。
5. 確認ダイアログボックスで、ポリシーの名前を入力し、[Delete] (削除) を選択します。

AWS CLI & AWS SDKs

バックアップポリシーを削除するには

以下のコード例は、DeletePolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Deletes an existing AWS Organizations policy.
/// </summary>
public class DeletePolicy
{
    /// <summary>
    /// Initializes the Organizations client object and then uses it to
    /// delete the policy with the specified policyId.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var policyId = "p-00000000";

        var request = new DeletePolicyRequest
```

```
            {
                PolicyId = policyId,
            };

            var response = await client.DeletePolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted Policy: {policyId}.");
            }
            else
            {
                Console.WriteLine($"Could not delete Policy: {policyId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeletePolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDeletePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def delete_policy(policy_id, orgs_client):
    """
    Deletes a policy.

    :param policy_id: The ID of the policy to delete.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.delete_policy(PolicyId=policy_id)
        logger.info("Deleted policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_id)
        raise
```

- API の詳細については、 DeletePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

バックアップポリシーのアタッチとデタッチ

バックアップポリシーは、組織全体、組織単位 (OU)、個々のアカウントで使用できます。以下の点に注意してください。

- バックアップポリシーを組織ルートにアタッチすると、ポリシーはルートのすべてのメンバー OU とアカウントに適用されます。
- バックアップポリシーを OU にアタッチすると、そのポリシーは OU またはその子 OU に属するアカウントに適用されます。これらのアカウントには、組織ルートにアタッチされたバックアップポリシーも適用されます。

- バックアップポリシーをアカウントにアタッチすると、ポリシーはそのアカウントにのみ適用されます。このアカウントは、組織のルートにアタッチされたポリシーと、そのアカウントが属する OU にも適用されます。

アカウントがルートおよび親 OU から継承するバックアップポリシーと、アカウントに直接アタッチされたポリシーを集約したものが、有効なポリシーとなります。ポリシーを有効なポリシーにマージする方法については、「管理ポリシーの継承を理解する」を参照してください。

バックアップポリシーのアタッチ

組織の管理アカウントにサインインすると、バックアップポリシーを組織のルート、OU、または直接アカウントにアタッチできます。

> **最小アクセス許可**
>
> バックアップポリシーをアタッチするには、次のアクションを実行するアクセス許可が必要です。
>
> - `organizations:AttachPolicy`

AWS Management Console

バックアップポリシーのアタッチには、ポリシーに移動する方法と、ポリシーをアタッチするルート、OU、またはアカウントに移動する方法があります。

ルート、OU、またはアカウントに移動してバックアップポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをアタッチするルート、OU、またはアカウントを見つけ、名前を選択します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
3. [Policies] (ポリシー) タブの [Backup policies] (バックアップポリシー) の項目で、[Attach] (アタッチ) を選択します。
4. 目的のポリシーを見つけて [Attach policy] (ポリシーのアタッチ) を選択します。

[Policies] (ポリシー) タブで、アタッチされているバックアップポリシーの一覧が更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動してバックアップポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページで、アタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで [Attach] (アタッチ) を選択します。
4. ポリシーをアタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
5. Attach policy] (ポリシーのアタッチ) を選択します。

   [Targets] (ターゲット) タブで、アタッチされているバックアップポリシーの一覧が更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

バックアップポリシーを組織のルート、OU、またはアカウントにアタッチするには

以下のコード例は、`AttachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
```

```
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Shows how to attach an AWS Organizations policy to an organization,
    /// an organizational unit, or an account.
    /// </summary>
    public class AttachPolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and then calls the
        /// AttachPolicyAsync method to attach the policy to the root
        /// organization.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();
            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new AttachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.AttachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully attached Policy ID {policyId} to
Target ID: {targetId}.");
            }
            else
            {
                Console.WriteLine("Was not successful in attaching the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスAttachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシーを OU にアタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- API の詳細については、「 コマンドリファレンスAttachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def attach_policy(policy_id, target_id, orgs_client):
```

```
    """
    Attaches a policy to a target. The target is an organization root, account,
or
    organizational unit.

    :param policy_id: The ID of the policy to attach.
    :param target_id: The ID of the resources to attach the policy to.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.attach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Attached policy %s to target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't attach policy %s to target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 AttachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに有効になります。

バックアップポリシーのデタッチ

組織の管理アカウントにサインインすると、組織ルート、OU、またはアカウントから、アタッチされているバックアップポリシーをデタッチすることができます。エンティティからバックアップポリシーをデタッチすると、そのポリシーは、現在デタッチされたエンティティによって以前影響を受けたすべてのアカウントに適用されなくなります。ポリシーをデタッチするには、次のステップを実行します。

最小アクセス許可

組織ルート、OU、またはアカウントからバックアップポリシーをデタッチするには、以下のアクションを実行するアクセス許可が必要です。

- `organizations:DetachPolicy`

AWS Management Console

バックアップポリシーのデタッチには、ポリシーに移動する方法と、ポリシーをデタッチするルート、OU、またはアカウントに移動する方法があります。

ポリシーがアタッチされているルート、OU、またはアカウントに移動してバックアップポリシーをデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをデタッチするルート、OU、またはアカウントに移動します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。ルート、OU、またはアカウントの名前を選択します。
3. [Policies] (ポリシー) タブで、デタッチするバックアップポリシーの横にあるラジオボタンを選択し、[Detach] (デタッチ) を選択します。
4. 確認ダイアログボックスで、[Detach policy] (ポリシーのデタッチ) を選択します。

   アタッチされているバックアップポリシーの一覧が更新されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動してバックアップポリシーをデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. バックアップポリシーページで、ルート、OU、またはアカウントからデタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで、ポリシーをデタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。
4. [Detach] (デタッチ) を選択します。
5. 確認ダイアログボックスで、[Detach] (デタッチ) を選択します。

アタッチされているバックアップポリシーの一覧が更新されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

組織のルート、OU、またはアカウントからバックアップポリシーをデタッチするには

以下のコード例は、DetachPolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to detach a policy from an AWS Organizations organization,
/// organizational unit, or account.
/// </summary>
public class DetachPolicy
{
    /// <summary>
    /// Initializes the Organizations client object and uses it to call
    /// DetachPolicyAsync to detach the policy.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var policyId = "p-00000000";
        var targetId = "r-0000";
```

```
            var request = new DetachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.DetachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully detached policy with Policy Id:
{policyId}.");
            }
            else
            {
                Console.WriteLine("Could not detach the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDetachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111
 --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDetachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def detach_policy(policy_id, target_id, orgs_client):
    """
    Detaches a policy from a target.

    :param policy_id: The ID of the policy to detach.
    :param target_id: The ID of the resource where the policy is currently
 attached.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.detach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Detached policy %s from target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 DetachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに反映されます。

有効なバックアップポリシーの表示

アカウントの有効なバックアップポリシーは、AWS マネジメントコンソール、AWS API、AWS Command Line Interface から確認できます。以下のセクションでは、有効なバックアップポリシーのおおまかな概要と例を示します。

有効なバックアップポリシーはどのようなものですか。

有効なバックアップポリシーでは、AWS アカウント に適用される最終的なバックアッププラン設定を指定します。これは、アカウントが継承するすべてのバックアップポリシーと、アカウントに直接アタッチされているバックアップポリシーが集約されたものです。組織ルートにバックアップポリシーをアタッチすると、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) にバックアップ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。アカウントに直接ポリシーをアタッチすると、その AWS アカウント にのみ適用されます。

例えば、組織のルートにアタッチされたバックアップポリシーでは、組織内のすべてのアカウントが、デフォルトのバックアップ頻度 (週に 1 回) ですべての Amazon DynamoDB テーブルをバックアップするように指定できます。テーブル内の重要な情報を持つ 1 つのメンバーアカウントに直接アタッチされた個別のバックアップポリシーは、1 日に 1 回の値で頻度を上書きできます。これらのバックアップポリシーの組み合わせは、有効なバックアップポリシーで構成されます。有効なバックアップポリシーは、組織内のアカウントごとに個別に決定されます。この例では、例外的に毎日テーブルをバックアップする 1 つのアカウントを除き、組織内のすべてのアカウントが 1 週間に 1 回ずつ DynamoDB テーブルをバックアップします。

複数のバックアップポリシーが集約され、最終的に有効なバックアップポリシーが構成される仕組みについては、管理ポリシーの継承を理解する を参照してください。

有効なバックアップポリシーの表示

アカウントの有効なバックアップポリシーは、AWS Management Console、AWS API、AWS Command Line Interface のいずれかを使用して表示できます。

最小アクセス許可

アカウントの有効なバックアップポリシーを表示するには、以下のアクションを実行するアクセス許可が必要です。

- `organizations:DescribeEffectivePolicy`
- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要

AWS Management Console

アカウントの有効なバックアップポリシーを表示するには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページで、有効なバックアップポリシーを確認するアカウントの名前を選択します。場合によっては、目的のアカウントを見つけるには OU を展開 (
を選択) する必要があります。
3. [Policies] (ポリシー) タブの [Backup policies] (バックアップポリシー) セクションで、[View the effective backup policy for this AWS アカウント] (この の有効なバックアップポリシーを表示する) を選択します。

   指定したアカウントに適用されている有効なポリシーがコンソールに表示されます。

   > **Note**
   >
   > 有効なポリシーをコピーアンドペーストして、大きな変更を加えずに別のバックアッププポリシーの JSON として使用することはできません。バックアップポリシードキュメントには、各設定を最終的な有効なポリシーにマージする方法を指定する継承演算子を含める必要があります。

AWS CLI & AWS SDKs

アカウントの有効なバックアップポリシーを表示するには

次のいずれかのコマンドを使用して、有効なバックアップポリシーを表示できます。

- AWS CLI: describe-effective-policy

  次の例では、バックアップポリシーの詳細を表示しています。

```
$ aws organizations describe-effective-policy \
--policy-type BACKUP_POLICY \
--target-id 123456789012{
    "EffectivePolicy": {
        "LastUpdatedTimestamp": "2020-06-22T14:31:50.748000-07:00",
```

```
        "TargetId": "123456789012",
        "PolicyType": "BACKUP_POLICY",
        "PolicyContent": "{\"plans\":{\"pii_backup_plan\":{\"regions\":[\"ap-
northeast-2\",\"us-east-1\",\"eu-north-1\"],\
"selections\":{\"tags\":{\"datatype\":{\"iam_role_arn\":\"arn:aws:iam::
$account:role/MyIamRole\",\"tag_value\":[\"PII\"],\
"tag_key\":\"dataType\"}}},\"rules\":{\"hourly\":{\"complete_backup_window_minutes
\":\"10080\",\"target_backup_vault_name\
":\"FortKnox\",\"start_backup_window_minutes\":\"480\",\"schedule_expression\":
\"cron(0 5/1 ? * * *)\",\"lifecycle\":{\"mo
ve_to_cold_storage_after_days\":\"180\",\"delete_after_days\":\"270\"},
\"copy_actions\":{\"arn:aws:backup:us-east-1:$accou
nt:backup-vault:secondary-vault\":{\"lifecycle\":
{\"move_to_cold_storage_after_days\":\"10\",\"delete_after_days\":\"100\"
}}}}}}}"
    }
}
```

- AWS SDK: DescribeEffectivePolicy

AWS CloudTrail イベントを使用して組織のバックアップポリシーを監視する

AWS CloudTrail イベントを使用して、AWS 組織内の任意のアカウントでバックアップポリシーが作成、更新、削除された日時や、組織のバックアップ計画が無効になっているかを監視できます。詳細については、「AWS Backup デベロッパーガイド」の「クロスアカウント管理イベントのログ」を参照してください。

バックアップポリシーの構文と例

このページでは、バックアップポリシーの構文について説明し、例を示します。

バックアップポリシーの構文

バックアップポリシーは、JSON のルールに従って構造化されたプレーンテキストファイルです。バックアップポリシーの構文は、すべての管理ポリシータイプの構文に従います。この構文の詳細については、「ポリシー構文と管理ポリシータイプの継承」を参照してください。このトピックでは、一般的な構文をバックアップポリシータイプの特定の要件に適用することを重点的に扱っています。

バックアップポリシーの大部分は、バックアッププランとそのルールで構成されます。バックアップポリシー内の AWS Organizations バックアッププランの構文は、 で使用される構文と構造的に同じですが AWS Backup、キー名は異なります。以下のポリシーキー名の説明には、それぞれに同等

の AWS Backup プランキー名が含まれています。 AWS Backup プランの詳細については、「 AWS Backup デベロッパーガイドCreateBackupPlan」の「」を参照してください。

> **Note**
>
> JSON を使用すると、重複するキー名は拒否されます。1 つのポリシーに複数のプラン、ルール、または選択を含める場合は、各キーの名前が一意であることを確認してください。

完全であり、かつ機能するには、有効なバックアップポリシーに、スケジュールとルールを備えたバックアッププラン以上のものを含める必要があります。このポリシーでは、バックアップする AWS リージョン とリソース、およびバックアップの実行に AWS Backup が使用できる AWS Identity and Access Management (IAM) ロールも特定する必要があります。

次の機能的に完全なポリシーは、基本的なバックアップポリシーの構文を示します。この例がアカウントに直接アタッチされている場合、 または の値dataTypeの タグを持つ us-east-1および eu-north-1リージョンで、そのアカウントのすべてのリソースをバックアップ AWS Backup しますPIIRED。これらのリソースを毎日午前 5:00 に My_Backup_Vault にバックアップし、My_Secondary_Vault にコピーを保存します。これらのボールトは両方ともリソースと同じアカウントにあります。また、明示的に指定された別の My_Tertiary_Vault にバックアップのコピーを保存します。ボールトは、有効なポリシーを受け取る各 AWS リージョン の指定された各 に既に存在 AWS アカウント している必要があります。バックアップされたリソースに EC2 インスタンスが含まれている場合、それらのインスタンスのバックアップに対し、Microsoft ボリュームシャドウコピーサービス (VSS) のサポートが有効になります。バックアップは、各復旧ポイントにタグ Owner:Backup を適用します。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "rules": {
                "My_Hourly_Rule": {
                    "schedule_expression": {"@@assign": "cron(0 5 ? * * *)"},
                    "start_backup_window_minutes": {"@@assign": "60"},
                    "complete_backup_window_minutes": {"@@assign": "604800"},
                    "enable_continuous_backup": {"@@assign": false},
                    "target_backup_vault_name": {"@@assign": "My_Backup_Vault"},
                    "recovery_point_tags": {
                        "Owner": {
                            "tag_key": {"@@assign": "Owner"},
                            "tag_value": {"@@assign": "Backup"}
```

```
                        }
                    },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": {"@@assign": "180"},
                        "delete_after_days": {"@@assign": "270"}
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-west-2:$account:backup-
vault:My_Secondary_Vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-west-2:$account:backup-
vault:My_Secondary_Vault"
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": {"@@assign": "180"},
                                "delete_after_days": {"@@assign": "270"}
                            }
                        },
                        "arn:aws:backup:us-east-1:$account:backup-
vault:My_Tertiary_Vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-
east-1:111111111111:backup-vault:My_Tertiary_Vault"
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": {"@@assign": "180"},
                                "delete_after_days": {"@@assign": "270"}
                            }
                        }
                    }
                }
            },
            "regions": {
                "@@append": [
                    "us-east-1",
                    "eu-north-1"
                ]
            },
            "selections": {
                "tags": {
                    "My_Backup_Assignment": {
                        "iam_role_arn": {"@@assign": "arn:aws:iam::$account:role/
MyIamRole"},
                        "tag_key": {"@@assign": "dataType"},
```

```
                        "tag_value": {
                            "@@assign": [
                                "PII",
                                "RED"
                            ]
                        }
                    }
                }
            },
            "advanced_backup_settings": {
                "ec2": {
                    "windows_vss": {"@@assign": "enabled"}
                }
            },
            "backup_plan_tags": {
                "stage": {
                    "tag_key": {"@@assign": "Stage"},
                    "tag_value": {"@@assign": "Beta"}
                }
            }
        }
    }
}
```

バックアップポリシー構文には、次のコンポーネントが含まれます。

- $account 変数 - ポリシーの特定のテキスト文字列では、$account 変数を使用して現在の AWS アカウントを表すことができます。が有効なポリシーで計画 AWS Backup を実行すると、この変数は AWS アカウント 有効なポリシーとその計画が実行されている現在の に自動的に置き換えられます。

  > **Important**
  >
  > $account 変数は、Amazon リソースネーム (ARN) を含めることができるポリシー要素でのみ使用できます。例えば、バックアップを格納するバックアップボールトを指定したり、バックアップを実行するアクセス許可を持つ IAM ロールを指定したりできるポリシー要素です。

  例えば、以下では、ポリシーが適用される各 に という名前 AWS アカウント のボールトMy_Vaultが存在する必要があります。

```
arn:aws:backup:us-west-2:$account:vault:My_Vault"
```

AWS CloudFormation スタックセットとその Organizations との統合を使用して、組織内の各メンバーアカウントのバックアップボールトと IAM ロールを自動的に作成および設定することをお勧めします。詳細については、AWS CloudFormation ユーザーガイドのセルフマネージド型のアクセス許可を持つスタックセットの作成を参照してください。

- 継承演算子 - バックアップポリシーでは、継承値設定演算子および子制御演算子の両方を使用できます。
- plans

  ポリシーの最上位レベルにあるキーは plans キーです。バックアップポリシーは、常にポリシーファイルの先頭にこの固定キー名のある状態で開始する必要があります。このキーの下に、1 つ以上のバックアッププランを配置できます。
- plans 最上位キーの下にある各プランには、ユーザーが割り当てたバックアッププラン名で構成されるキー名があります。前の例では、バックアッププラン名は PII_Backup_Plan です。1 つのポリシーには複数のプランを含めることができ、それぞれに独自の rules、regions、selections、tags を設定できます。

  バックアップポリシーのこのバックアッププランキー名は、 AWS Backup プランのBackupPlanNameキーの値にマッピングされます。

  各プランには次の要素を含めることができます。

  - rules - このキーには、ルールのコレクションが格納されます。各ルールは、開始時刻とウィンドウがスケジュールされたタスクに変換され、そのスケジュールに基づき、有効なバックアップポリシーの selections 要素と regions 要素によって識別されるリソースをバックアップします。
  - regions — このキーには、このポリシーでバックアップできるリソース AWS リージョン を持つ の配列リストが含まれています。
  - selections - このキーには、指定した rules によってバックアップされる (指定した regions 内にある) リソースのコレクションが 1 つ以上格納されます。
  - advanced_backup_settings - このキーには、特定のリソースで実行されるバックアップに固有の設定が格納されます。
  - backup_plan_tags - これは、バックアッププラン自体にアタッチされるタグを指定します。
- rules

rules ポリシーキーは、 AWS Backup プラン内の Rules キーにマッピングされます。rules キーの下に 1 つ以上のルールを設定できます。各ルールは、選択したリソースのバックアップを実行するスケジュールされたタスクになります。

各ルールにはキーがあり、そのキー名がルールの名前です。前の例では、ルール名は「My_Hourly_Rule」です。ルールキーの値は、ルール要素の次のコレクションです。

- schedule_expression – このポリシーキーは、 AWS Backup プランのScheduleExpressionキーにマッピングされます。

  バックアップの開始時刻を指定します。このキーには、@@assign継承値演算子と、 AWS Backup がバックアップジョブを開始するタイミングを指定する CRON 式を含む文字列値が含まれます。CRON 文字列の一般的な形式は「cron()」です。それぞれが、数字またはワイルドカードです。例えば、cron(0 5 ? * 1,3,5 *) は、毎週月曜日、水曜日、金曜日の午前 5 時にバックアップを開始します。cron(0 0/1 ? * * *) は、1 週間毎日、毎正時にバックアップを開始します。

- target_backup_vault_name – このポリシーキーは、 AWS Backup プランのTargetBackupVaultNameキーにマッピングされます。

  バックアップを保存するバックアップボールトの名前を指定します。を使用して値を作成します AWS Backup。このキーには、@@assign 継承値演算子とボールト名を持つ文字列値が格納されます。

  > **Important**
  >
  > バックアッププランを最初に起動するときには、ボールトがすでに存在している必要があります。 AWS CloudFormation スタックセットとその Organizations との統合を使用して、組織内の各メンバーアカウントのバックアップボールトと IAM ロールを自動的に作成および設定することをお勧めします。詳細については、AWS CloudFormation ユーザーガイドのセルフマネージド型のアクセス許可を持つスタックセットの作成を参照してください。

- start_backup_window_minutes - このポリシーキーは、 AWS Backup プランのStartWindowMinutesキーにマッピングされます。

  (オプション) 正常に開始しないジョブをキャンセルするまでの待機時間を分単位で指定します。このキーには、@@assign 継承値演算子と、整数の時間 (分) を持つ値が格納されます。

- complete_backup_window_minutes - このポリシーキーは、 AWS Backup プランの CompletionWindowMinutes キーにマッピングされます。

  (オプション) バックアップジョブが正常に開始してから完了するか、 AWS Backupによってキャンセルされるまでの時間 (分) を指定します。このキーには、@@assign 継承値演算子と、整数の時間 (分) を持つ値が格納されます。

- enable_continuous_backup - このポリシーキーは、 AWS Backup プランのEnableContinuousBackupキーにマッピングされます。

  (オプション) が継続的バックアップ AWS Backup を作成するかどうかを指定します。Trueは、 AWS Backup が point-in-time 復元可能な継続的バックアップ (PITR) を作成することになり、 False (または指定されていない) は、 AWS Backup がスナップショットバックアップを作成することを表します。

  > **Note**
  >
  > PITR 対応のバックアップは最大 35 日間保持されるため、次のいずれかのオプションを設定する場合は、False を選択するか、値を未指定にする必要があります。
  >
  > - delete_after_days を 35 より大きい値に設定する。
  > - move_to_cold_storage_after_days を任意の値に設定する。

  継続的バックアップの詳細については、「 AWS Backup デベロッパーガイド」の「P oint-in-time 復旧」を参照してください。

- lifecycle - このポリシーキーは、 AWS Backup プランのLifecycleキーにマッピングされます。

  (オプション) がこのバックアップをコールドストレージ AWS Backup に移行するタイミングと、期限切れになるタイミングを指定します。

  - move_to_cold_storage_after_days - このポリシーキーは、 AWS Backup プランのMoveToColdStorageAfterDaysキーにマッピングされます。

    バックアップが実行されてから、 AWS Backup が復旧ポイントをコールドストレージに移動するまでの日数を指定します。このキーには、@@assign 継承値演算子と、整数の日数を持つ値が格納されます。

  - delete_after_days - このポリシーキーは、 AWS Backup プランのDeleteAfterDaysキーにマッピングされます。

バックアップが実行されてから、 AWS Backup が復旧ポイントを削除するまでの日数を指定します。このキーには、@@assign 継承値演算子と、整数の日数を持つ値が格納されます。バックアップをコールドストレージに移行する場合は、その場所に 90 日以上保持する必要があります。したがって、この値は move_to_cold_storage_after_days の値よりも 90 日以上大きくする必要があります。

- copy_actions - このポリシーキーは、 AWS Backup プランのCopyActionsキーにマッピングされます。

  (オプション) がバックアップを 1 つ以上の追加の場所にコピー AWS Backup することを指定します。各バックアップコピーの場所は次のように記述されます。

  - そのコピーアクションを一意に識別する名前を持つキー。現時点で、キー名はバックアップボールトの Amazon リソースネーム (ARN) である必要があります。このキーには、2 つのエントリが含まれます。

    - target_backup_vault_arn - このポリシーキーは、 AWS Backup プランの DestinationBackupVaultArn キーにマッピングされます。

      (オプション) がバックアップの追加のコピー AWS Backup を保存するボールトを指定します。このキーの値には、@@assign 継承値演算子と、ボールトの ARN が格納されます。

      - バックアップポリシーが実行され AWS アカウント ている でボールトを参照するには、アカウント ID 番号の代わりに ARN の $account変数を使用します。がバックアッププラン AWS Backup を実行すると、変数 はポリシーが実行され AWS アカウント ているのアカウント ID 番号に自動的に置き換えられます。これにより、バックアップポリシーが組織内の複数のアカウントに適用される場合でも、バックアップが正常に実行されます。
      - 同じ組織内の異なる AWS アカウント のボールトを参照するには、実際のアカウント ID 番号を ARN に使用します。

      > **Important**
      >
      > - このキーがない場合、親キーの名前をすべて小文字で表記したものが ARN に使用されます。ARN では大文字と小文字が区別されるため、この文字列がボールトの実際の ARN と一致せず、プランが失敗する可能性があります。このため、このキーと値は常に指定することをお勧めします。
      > - バックアップのコピー先にするバックアップボールトは、最初にバックアッププランを起動する時点ですでに存在している必要があります。組織内の各メンバーアカウントのバックアップボールトと IAM ロールを自動的に作成および設定する

には、 AWS CloudFormation StackSets、およびその Organizations との統合を使用することをお勧めします。詳細については、AWS CloudFormation ユーザーガイドのセルフマネージド型のアクセス許可を持つスタックセットの作成を参照してください。

- `lifecycle` – このポリシーキーは、 AWS Backup プランの `Lifecycle` キーの下にある `CopyAction` キーにマッピングされます。

  (オプション) がこのバックアップのコピーをコールドストレージ AWS Backup に移行するタイミングと、有効期限を指定します。

  - `move_to_cold_storage_after_days` - このポリシーキーは、 AWS Backup プランの `MoveToColdStorageAfterDays` キーにマッピングされます。

    がリカバリポイントをコールドストレージ AWS Backup に移動するまでのバックアップ発生からの日数を指定します。このキーには、@@assign 継承値演算子と、整数の日数を持つ値が格納されます。

  - `delete_after_days` - このポリシーキーは、 AWS Backup プランの `DeleteAfterDays` キーにマッピングされます。

    がリカバリポイント AWS Backup を削除するまでのバックアップ発生からの日数を指定します。このキーには、@@assign 継承値演算子と、整数の日数を持つ値が格納されます。バックアップをコールドストレージに移行する場合は、その場所に 90 日以上保持する必要があります。したがって、この値は `move_to_cold_storage_after_days` の値よりも 90 日以上大きくする必要があります。

- `recovery_point_tags` - このポリシーキーは、 AWS Backup プランのRecoveryPointTagsキーにマッピングされます。

  (オプション) このプランから作成する各バックアップに がア AWS Backup タッチするタグを指定します。このキーの値には、以下の要素が 1 つ以上含まれます。

  - このキー名と値のペアの識別子。`recovery_point_tags` の下の各要素のこの名前は、`tag_key` で大文字と小文字の取り扱いが異なる場合でも、すべて小文字のタグキー名です。この識別子では、大文字と小文字は区別されません。前の例では、このキーペアは名前 `Owner` で識別されました。各キーペアには、以下の要素が含まれます。

    - `tag_key` - バックアッププランにアタッチするタグキー名を指定します。このキーには、@@assign 継承値演算子および文字列値が格納されます。 値では、大文字と小文字が区別されます。

    - tag_value - バックアッププランに適用され、tag_key に関連付けられている値を指定します。このキーには、継承値演算子、有効なポリシーの置換、追加、または削除する 1 つ以上の値が格納されます。これらの値は大文字と小文字が区別されます。

- regions

  regions ポリシーキーは、 AWS リージョン が AWS Backup selectionsキーの条件に一致するリソースを検索するためにどの を検索するかを指定します。このキーには、継承値演算子のいずれかと、 AWS リージョン コードの 1 つ以上の文字列値が含まれます。例: ["us-east-1", "eu-north-1"]。

- selections

  selections ポリシーキーは、このポリシーのプランルールによってバックアップされるリソースを指定します。このキーは、 の BackupSelection オブジェクトにほぼ対応しています AWS Backup。リソースは、タグキー名と値をマッチングするためのクエリによって指定されます。selections キーの下には、tags という 1 つのキーが含まれています。

  - tags - リソースを識別するタグと、リソースに対してクエリおよびバックアップを実行するアクセス許可を持つ IAM ロールを指定します。このキーの値には、以下の要素が 1 つ以上含まれます。
    - このタグ要素の識別子。tags の下にあるこの識別子は、クエリするタグの大文字と小文字の取り扱いが異なる場合でも、すべて小文字のタグキー名です。この識別子では、大文字と小文字は区別されません。前の例では、1 つの要素が名前 My_Backup_Assignment で識別されました。tags の下の各識別子には、以下の要素が含まれています。
      - iam_role_arn - regions キーで指定された AWS リージョン で、タグクエリによって識別されるリソースへのアクセス許可を持つ IAM ロールを指定します。この値には、@@assign継承値演算子と role. AWS Backup uses の ARN を含む文字列値が含まれます。 はこのロールを使用して、リソースをクエリして検出し、バックアップを実行します。

        アカウント ID 番号の代わりに、ARN の $account 変数を使用できます。バックアッププランが によって実行されると AWS Backup、変数 はポリシーが実行され AWS アカウントている の実際のアカウント ID 番号に自動的に置き換えられます。

        > **Important**
        >
        > このロールは、バックアッププランを最初に起動するときにすでに存在している必要があります。 AWS CloudFormation スタックセットとその Organizations との統合を使用して、組織内の各メンバーアカウントのバックアップボールトと IAM

ロールを自動的に作成および設定することをお勧めします。詳細については、AWS CloudFormation ユーザーガイドのセルフマネージド型のアクセス許可を持つスタックセットの作成を参照してください。

  - tag_key - 検索するタグキー名を指定します。このキーには、@@assign 継承値演算子および文字列値が格納されます。 値では、大文字と小文字が区別されます。
  - tag_value – と一致するキー名に関連付ける必要がある値を指定しますtag_key。tag_keyと の両方tag_valueが一致する場合にのみ、 はバックアップにリソース AWS Backup を含めます。このキーには、継承値演算子、有効なポリシーの置換、追加、または削除する 1 つ以上の値が格納されます。これらの値は大文字と小文字が区別されます。

- advanced_backup_settings - 特定のバックアップシナリオの設定を指定します。このキーには 1 つ以上の設定が含まれます。各設定は、次の要素を持つ JSON オブジェクト文字列です。
  - オブジェクトキー名 - 次の詳細設定が適用されるリソースのタイプを指定する文字列。
  - オブジェクト値 - 関連付けられたリソースタイプに固有のバックアップ設定を 1 つ以上含む JSON オブジェクト文字列。

  現時点でサポートされている高度なバックアップ設定で利用可能なのは、Amazon EC2 インスタンスで実行される Windows または SQL Server の Microsoft ボリュームシャドウコピーサービス (VSS) バックアップだけです。キー名は "ec2" リソースタイプであるる必要があります。この値により、Amazon EC2 インスタンスで実行されるバックアップに対する "windows_vss" サポートを enabled にするか disabled にするかを指定できます。この機能について詳しくは、AWS Backup デベロッパーガイドの Creating a VSS-Enabled Windows Backup を参照してください。

```
"advanced_backup_settings": {
    "ec2": {
        "windows_vss": {
            "@@assign": "enabled"
        }
    }
}
```

- backup_plan_tags - バックアッププラン自体にアタッチされるタグを指定します。これは、ルールまたは選択で指定されたタグには影響しません。

  (オプション) バックアッププランにタグをアタッチできます。このキーの値は、要素のコレクションです。

backup_plan_tags の下の各要素のキー名は、クエリするタグの大文字と小文字の取り扱いが異なる場合でも、すべて小文字のタグキー名です。この識別子では、大文字と小文字は区別されません。これらの各エントリの値は、次のキーで構成されます。

- tag_key - バックアッププランにアタッチするタグキー名を指定します。このキーには、@@assign 継承値演算子および文字列値が格納されます。この値では、大文字と小文字が区別されます。
- tag_value - バックアッププランに適用され、tag_key に関連付けられている値を指定します。このキーには、@@assign 継承値演算子および文字列値が格納されます。この値では、大文字と小文字が区別されます。

バックアップポリシーの例

次のサンプルバックアップポリシーは、情報提供のみを目的としています。以下の例の一部では、スペースを節約するために JSON の空白書式が圧縮される場合があります。

例 1: 親ノードに割り当てられたポリシー

次の例は、アカウントの親ノードの 1 つに割り当てられているバックアップポリシーを示しています。

親ポリシー - このポリシーは、組織のルート、またはすべての対象アカウントの親である任意の OU にアタッチできます。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": {
                "@@assign": [
                    "ap-northeast-2",
                    "us-east-1",
                    "eu-north-1"
                ]
            },
            "rules": {
                "Hourly": {
                    "schedule_expression": {
                        "@@assign": "cron(0 5/1 ? * * *)"
                    },
                    "start_backup_window_minutes": {
                        "@@assign": "480"
```

```
                    },
                    "complete_backup_window_minutes": {
                        "@@assign": "10080"
                    },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": {
                            "@@assign": "180"
                        },
                        "delete_after_days": {
                            "@@assign": "270"
                        }
                    },
                    "target_backup_vault_name": {
                        "@@assign": "FortKnox"
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:backup-
vault:secondary_vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-east-1:$account:backup-
vault:secondary_vault"
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": {
                                    "@@assign": "30"
                                },
                                "delete_after_days": {
                                    "@@assign": "120"
                                }
                            }
                        },
                        "arn:aws:backup:us-west-1:111111111111:backup-
vault:tertiary_vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-
west-1:111111111111:backup-vault:tertiary_vault"
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": {
                                    "@@assign": "30"
                                },
                                "delete_after_days": {
                                    "@@assign": "120"
                                }
```

```
                            }
                        }                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": {
                            "@@assign": "arn:aws:iam::$account:role/MyIamRole"
                        },
                        "tag_key": {
                            "@@assign": "dataType"
                        },
                        "tag_value": {
                            "@@assign": [
                                "PII",
                                "RED"
                            ]
                        }
                    }
                }
            },
            "advanced_backup_settings": {
                "ec2": {
                    "windows_vss": {
                        "@@assign": "enabled"
                    }
                }
            }
        }
    }
}
```

アカウントに他のポリシーが継承またはアタッチされていない場合、該当する各 でレンダリングされる有効なポリシーは次の例の AWS アカウント ようになります。CRON 式を使用すると、バックアップは毎正時に実行されます。アカウント ID 123456789012 は、各アカウントの実際のアカウント ID になります。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": [
                "us-east-1",
```

```
                "ap-northeast-3",
                "eu-north-1"
            ],
            "rules": {
                "hourly": {
                    "schedule_expression": "cron(0 0/1 ? * * *)",
                    "start_backup_window_minutes": "60",
                    "target_backup_vault_name": "FortKnox",
                    "lifecycle": {
                        "to_delete_after_days": "2",
                        "move_to_cold_storage_after_days": "180"
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:secondary_vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-east-1:
$account:vault:secondary_vault"
                            },
                            "lifecycle": {
                                "to_delete_after_days": "28",
                                "move_to_cold_storage_after_days": "180"
                            }
                        },
                        "arn:aws:backup:us-west-1:111111111111:vault:tertiary_vault": {
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-
west-1:111111111111:vault:tertiary_vault"
                            },
                            "lifecycle": {
                                "to_delete_after_days": "28",
                                "move_to_cold_storage_after_days": "180"
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": "arn:aws:iam::123456789012:role/MyIamRole",
                        "tag_key": "dataType",
                        "tag_value": [
                            "PII",
                            "RED"
```

```
                        ]
                    }
                }
            },
            "advanced_backup_settings": {
                "ec2": {
                    "windows_vss": "enabled"
                }
            }
        }
    }
}
```

例 2: 親ポリシーが子ポリシーとマージされる

次の例では、継承された親ポリシーと子ポリシーが継承されたか、 AWS アカウント マージに直接アタッチされて有効なポリシーを形成しています。

親ポリシー - このポリシーは、組織のルートまたは任意の親 OU にアタッチできます。

```
{
    "plans": {
       "PII_Backup_Plan": {
            "regions": { "@@append":[ "us-east-1", "ap-northeast-3", "eu-north-1" ] },
            "rules": {
                "Hourly": {
                    "schedule_expression": { "@@assign": "cron(0 0/1 ? * * *)" },
                    "start_backup_window_minutes": { "@@assign": "60" },
                    "target_backup_vault_name": { "@@assign": "FortKnox" },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": { "@@assign": "28" },
                        "to_delete_after_days": { "@@assign": "180" }
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:secondary_vault" : {
                            "target_backup_vault_arn" : {
                                "@@assign" : "arn:aws:backup:us-east-1:
$account:vault:secondary_vault"
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": { "@@assign":
 "28" },
                                "to_delete_after_days": { "@@assign": "180" }
```

```
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": { "@@assign": "arn:aws:iam::$account:role/
MyIamRole" },
                        "tag_key": { "@@assign": "dataType" },
                        "tag_value": { "@@assign": [ "PII", "RED" ] }
                    }
                }
            }
        }
    }
}
```

子ポリシー - このポリシーは、アカウントに直接アタッチすることも、親ポリシーがアタッチされているより低いレベルの OU にアタッチすることもできます。

```
{
    "plans": {
       "Monthly_Backup_Plan": {
            "regions": {
                "@@append":[ "us-east-1", "eu-central-1" ] },
            "rules": {
                "Monthly": {
                    "schedule_expression": { "@@assign": "cron(0 5 1 * ? *)" },
                    "start_backup_window_minutes": { "@@assign": "480" },
                    "target_backup_vault_name": { "@@assign": "Default" },
                    "lifecycle": {
                        "move_to_cold_storage_after_days": { "@@assign": "30" },
                        "to_delete_after_days": { "@@assign": "365" }
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:Default" : {
                            "target_backup_vault_arn" : {
                                "@@assign" : "arn:aws:backup:us-east-1:
$account:vault:Default"
                            },
                            "lifecycle": {
```

```
                                "move_to_cold_storage_after_days": { "@@assign":
 "30" },
                                "to_delete_after_days": { "@@assign": "365" }
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "MonthlyDatatype": {
                        "iam_role_arn": { "@@assign": "arn:aws:iam::$account:role/
MyMonthlyBackupIamRole" },
                        "tag_key": { "@@assign": "BackupType" },
                        "tag_value": { "@@assign": [ "MONTHLY", "RED" ] }
                    }
                }
            }
        }
    }
}
```

最終的に適用される有効なポリシー - アカウントに適用される有効なポリシーには、2 つのプランがあり、それぞれに独自のルールのセットとルールを適用するリソースのセットがあります。

```
{
    "plans": {
       "PII_Backup_Plan": {
            "regions": [ "us-east-1", "ap-northeast-3", "eu-north-1" ],
            "rules": {
                "hourly": {
                    "schedule_expression": "cron(0 0/1 ? * * *)",
                    "start_backup_window_minutes": "60",
                    "target_backup_vault_name": "FortKnox",
                    "lifecycle": {
                        "to_delete_after_days": "2",
                        "move_to_cold_storage_after_days": "180"
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:secondary_vault" : {
                            "target_backup_vault_arn" : {
                                "@@assign" : "arn:aws:backup:us-east-1:
$account:vault:secondary_vault"
```

```
                            },
                            "lifecycle": {
                                "move_to_cold_storage_after_days": "28",
                                "to_delete_after_days": "180"
                            }
                        }
                    }
                },
                "selections": {
                    "tags": {
                        "datatype": {
                            "iam_role_arn": "arn:aws:iam::$account:role/MyIamRole",
                            "tag_key": "dataType",
                            "tag_value": [ "PII", "RED" ]
                        }
                    }
                }
            },
            "Monthly_Backup_Plan": {
                "regions": [ "us-east-1", "eu-central-1" ],
                "rules": {
                    "monthly": {
                        "schedule_expression": "cron(0 5 1 * ? *)",
                        "start_backup_window_minutes": "480",
                        "target_backup_vault_name": "Default",
                        "lifecycle": {
                            "to_delete_after_days": "365",
                            "move_to_cold_storage_after_days": "30"
                        },
                        "copy_actions": {
                            "arn:aws:backup:us-east-1:$account:vault:Default" : {
                                "target_backup_vault_arn": {
                                    "@@assign" : "arn:aws:backup:us-east-1:
$account:vault:Default"
                                },
                                "lifecycle": {
                                    "move_to_cold_storage_after_days": "30",
                                    "to_delete_after_days": "365"
                                }
                            }
                        }
                    },
```

```
            "selections": {
                "tags": {
                    "monthlydatatype": {
                        "iam_role_arn": "arn:aws:iam::&ExampleAWSAccountNo3;:role/
MyMonthlyBackupIamRole",
                        "tag_key": "BackupType",
                        "tag_value": [ "MONTHLY", "RED" ]
                    }
                }
            }
        }
    }
}
```

例 3: 親ポリシーが子ポリシーによる変更を防止する

次の例では、継承された親ポリシーが子制御演算子を使用してすべての設定を強制し、子ポリシーによって変更またはオーバーライドされないようにします。

親ポリシー - このポリシーは、組織のルートまたは任意の親 OU にアタッチできます。ポリシーのすべてのノードに "@@operators_allowed_for_child_policies": ["@@none"] が存在するということは、子ポリシーがどのような変更もプランに加えられないことを意味します。また、子ポリシーにより有効なポリシーにプランを追加することもできません。このポリシーは、関連付けられている OU の下にあるすべての OU およびアカウントに対して有効なポリシーになります。

```
{
    "plans": {
        "@@operators_allowed_for_child_policies": ["@@none"],
        "PII_Backup_Plan": {
            "@@operators_allowed_for_child_policies": ["@@none"],
            "regions": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "@@append": [
                    "us-east-1",
                    "ap-northeast-3",
                    "eu-north-1"
                ]
            },
            "rules": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "Hourly": {
                    "@@operators_allowed_for_child_policies": ["@@none"],
                    "schedule_expression": {
```

```
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "@@assign": "cron(0 0/1 ? * * *)"
                    },
                    "start_backup_window_minutes": {
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "@@assign": "60"
                    },
                    "target_backup_vault_name": {
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "@@assign": "FortKnox"
                    },
                    "lifecycle": {
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "move_to_cold_storage_after_days": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "@@assign": "28"
                        },
                        "to_delete_after_days": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "@@assign": "180"
                        }
                    },
                    "copy_actions": {
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "arn:aws:backup:us-east-1:$account:vault:secondary_vault": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "target_backup_vault_arn": {
                                "@@assign": "arn:aws:backup:us-east-1:
$account:vault:secondary_vault",
                                "@@operators_allowed_for_child_policies": ["@@none"]
                            },
                            "lifecycle": {
                                "@@operators_allowed_for_child_policies": ["@@none"],
                                "to_delete_after_days": {
                                    "@@operators_allowed_for_child_policies":
 ["@@none"],
                                    "@@assign": "28"
                                },
                                "move_to_cold_storage_after_days": {
                                    "@@operators_allowed_for_child_policies":
 ["@@none"],
                                    "@@assign": "180"
                                }
                            }
```

```
                        }
                    }
                }
            },
            "selections": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "tags": {
                    "@@operators_allowed_for_child_policies": ["@@none"],
                    "datatype": {
                        "@@operators_allowed_for_child_policies": ["@@none"],
                        "iam_role_arn": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "@@assign": "arn:aws:iam::$account:role/MyIamRole"
                        },
                        "tag_key": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "@@assign": "dataType"
                        },
                        "tag_value": {
                            "@@operators_allowed_for_child_policies": ["@@none"],
                            "@@assign": [
                                "PII",
                                "RED"
                            ]
                        }
                    }
                }
            },
            "advanced_backup_settings": {
                "@@operators_allowed_for_child_policies": ["@@none"],
                "ec2": {
                    "@@operators_allowed_for_child_policies": ["@@none"],
                    "windows_vss": {
                        "@@assign": "enabled",
                        "@@operators_allowed_for_child_policies": ["@@none"]
                    }
                }
            }
        }
    }
}
```

最終的に適用される有効なポリシー - 子バックアップポリシーが存在する場合、それらは無視され、親ポリシーが有効なポリシーになります。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": [
                "us-east-1",
                "ap-northeast-3",
                "eu-north-1"
            ],
            "rules": {
                "hourly": {
                    "schedule_expression": "cron(0 0/1 ? * * *)",
                    "start_backup_window_minutes": "60",
                    "target_backup_vault_name": "FortKnox",
                    "lifecycle": {
                        "to_delete_after_days": "2",
                        "move_to_cold_storage_after_days": "180"
                    },
                    "copy_actions": {
                        "target_backup_vault_arn": "arn:aws:backup:us-
east-1:123456789012:vault:secondary_vault",
                        "lifecycle": {
                            "move_to_cold_storage_after_days": "28",
                            "to_delete_after_days": "180"
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": "arn:aws:iam::123456789012:role/MyIamRole",
                        "tag_key": "dataType",
                        "tag_value": [
                            "PII",
                            "RED"
                        ]
                    }
                }
            },
            "advanced_backup_settings": {
```

```
                "ec2": {"windows_vss": "enabled"}
            }
        }
    }
}
```

例 4: 親ポリシーが、子ポリシーによる 1 つのバックアッププランへの変更を防止する

次の例では、継承された親ポリシーが子制御演算子を使用して単一のプランの設定を強制し、子ポリシーによって変更またはオーバーライドされないようにします。子ポリシーがプランを追加することはできます。

親ポリシー - このポリシーは、組織のルートまたは任意の親 OU にアタッチできます。この例は、plans 最上位レベルを除き、すべての子継承演算子がブロックされた前の例に似ています。このレベルでの @@append の設定により、子ポリシーが、有効なポリシーのコレクションに他の計画を追加できます。継承されたプランに対する変更は引き続きブロックされます。

プラン内のセクションは、わかりやすくするために切り捨てられています。

```
{
    "plans": {
        "@@operators_allowed_for_child_policies": ["@@append"],
        "PII_Backup_Plan": {
            "@@operators_allowed_for_child_policies": ["@@none"],
            "regions": { ... },
            "rules": { ... },
            "selections": { ... }
        }
    }
}
```

子ポリシー - このポリシーは、アカウントに直接アタッチすることも、親ポリシーがアタッチされているより低いレベルの OU にアタッチすることもできます。この子ポリシーは、新しいプランを定義します。

プラン内のセクションは、わかりやすくするために切り捨てられています。

```
{
    "plans": {
        "MonthlyBackupPlan": {
            "regions": { ... },
            "rules": { ... },
```

```
            "selections": { … }
        }
    }
}
```

最終的に適用される有効なポリシー - 有効なポリシーには、両方のプランが含まれます。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": { ... },
            "rules": { ... },
            "selections": { ... }
        },
        "MonthlyBackupPlan": {
            "regions": { ... },
            "rules": { ... },
            "selections": { … }
        }
    }
}
```

例 5: 子ポリシーが親ポリシーの設定をオーバーライドする

次の例では、子ポリシーが値設定演算子を使用して、親ポリシーから継承された設定の一部をオーバーライドしています。

親ポリシー - このポリシーは、組織のルートまたは任意の親 OU にアタッチできます。どの設定も子ポリシーによりオーバーライドできます。これは、デフォルトの動作を禁止する子制御演算子がない場合は、子ポリシーに @@assign、@@append、または@@remove が許可されるためです。親ポリシーには、有効なバックアッププランに必要なすべての要素が含まれているため、そのまま継承されていればリソースが正常にバックアップされます。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": {
                "@@append": [
                    "us-east-1",
                    "ap-northeast-3",
                    "eu-north-1"
                ]
```

```
            },
            "rules": {
                "Hourly": {
                    "schedule_expression": {"@@assign": "cron(0 0/1 ? * * *)"},
                    "start_backup_window_minutes": {"@@assign": "60"},
                    "target_backup_vault_name": {"@@assign": "FortKnox"},
                    "lifecycle": {
                        "to_delete_after_days": {"@@assign": "2"},
                        "move_to_cold_storage_after_days": {"@@assign": "180"}
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:t2": {
                            "target_backup_vault_arn": {"@@assign": "arn:aws:backup:us-
east-1:$account:vault:t2"},
                            "lifecycle": {
                                "move_to_cold_storage_after_days": {"@@assign": "28"},
                                "to_delete_after_days": {"@@assign": "180"}
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": {"@@assign": "arn:aws:iam::$account:role/
MyIamRole"},
                        "tag_key": {"@@assign": "dataType"},
                        "tag_value": {
                            "@@assign": [
                                "PII",
                                "RED"
                            ]
                        }
                    }
                }
            }
        }
    }
}
```

子ポリシー - 子ポリシーには、継承された親ポリシーとは異なる必要がある設定のみが含まれます。有効なポリシーにマージするときは、その他の必須設定を提供する、継承された親ポリシーが必要で

す。それがないと、有効なバックアップポリシーには無効なバックアッププランが格納され、期待どおりにリソースがバックアップされません。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": {
                "@@assign": [
                    "us-west-2",
                    "eu-central-1"
                ]
            },
            "rules": {
                "Hourly": {
                    "schedule_expression": {"@@assign": "cron(0 0/2 ? * * *)"},
                    "start_backup_window_minutes": {"@@assign": "80"},
                    "target_backup_vault_name": {"@@assign": "Default"},
                    "lifecycle": {
                        "move_to_cold_storage_after_days": {"@@assign": "30"},
                        "to_delete_after_days": {"@@assign": "365"}
                    }
                }
            }
        }
    }
}
```

最終的に適用される有効なポリシー - 有効なポリシーには両方のポリシーの設定が含まれます。子ポリシーによって提供される設定が、親ポリシーから継承された設定をオーバーライドします。この例では、以下の変更が発生します。

- リージョンのリストは、まったく別のリストに置き換えられます。継承リストにリージョンを追加する場合、子ポリシーで `@@assign` の代わりに `@@append` を使用します。
- AWS Backup は 1 時間ごとではなく、2 時間ごとに実行します。
- AWS Backup では、バックアップが開始されるまでに 60 分ではなく 80 分かかります。
- AWS Backup は`Default`、 の代わりに ボールトを使用します`FortKnox`。
- ライフサイクルは、コールドストレージへの転送とバックアップの最終的な削除の両方について延長されます。

```
{
    "plans": {
        "PII_Backup_Plan": {
            "regions": [
                "us-west-2",
                "eu-central-1"
            ],
            "rules": {
                "hourly": {
                    "schedule_expression": "cron(0 0/2 ? * * *)",
                    "start_backup_window_minutes": "80",
                    "target_backup_vault_name": "Default",
                    "lifecycle": {
                        "to_delete_after_days": "365",
                        "move_to_cold_storage_after_days": "30"
                    },
                    "copy_actions": {
                        "arn:aws:backup:us-east-1:$account:vault:secondary_vault": {
                            "target_backup_vault_arn": {"@@assign": "arn:aws:backup:us-
east-1:$account:vault:secondary_vault"},
                            "lifecycle": {
                                "move_to_cold_storage_after_days": "28",
                                "to_delete_after_days": "180"
                            }
                        }
                    }
                }
            },
            "selections": {
                "tags": {
                    "datatype": {
                        "iam_role_arn": "arn:aws:iam::$account:role/MyIamRole",
                        "tag_key": "dataType",
                        "tag_value": [
                            "PII",
                            "RED"
                        ]
                    }
                }
            }
        }
    }
}
```

タグポリシー

タグポリシーを使用して、タグキーおよびタグ値の大文字と小文字の処理方法の設定など、一貫したタグを維持できます。

タグとは

タグは、 AWS リソースに割り当てる AWS カスタム属性ラベルです。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。
- タグ値として知られるオプションのフィールド (例： `111122223333` または `Production`)。タグ値を省略すると、空の文字列を使用した場合と同じになります。タグキーと同様に、タグ値では大文字と小文字が区別されます。

このページの残りの部分では、タグポリシーについて説明します。タグの詳細については、次のソースを参照してください。

- 命名規則や使用規則など、タグ付けに関する一般的な情報については、AWS 「 リソースのタグ付けユーザーガイド」を参照してください。
- タグの使用をサポートするサービスのリストについては、「リResource Groups のタグ付け API リファレンス」を参照してください。
- タグを使用してリソースを分類する方法については、AWS 「 リソースのタグ付けのベストプラクティス」ホワイトペーパーを参照してください。
- Organizations リソースのタグ付けについては、「AWS Organizations リソースのタグ付け」を参照してください。
- 他のサービスのリソースのタグ付けについては AWS 、そのサービスのドキュメントを参照してください。

タグポリシーとは

タグポリシーはポリシーの一種で、組織のアカウント内のリソース間でタグを標準化するのに役立ちます。タグポリシーでは、リソースのタグ付けの際に適用されるタグ付けルールを指定します。

例えば、タグポリシーでは、`CostCenter` タグがリソースにアタッチされる際に、タグポリシーで定義されている大文字小文字の処理とタグ値を使用する必要があることを指定できます。タグポリ

シーは、指定したリソースタイプで非準拠のタグ付け操作を強制するように指定することもできます。つまり、指定されたリソースタイプで非準拠のタグ付けリクエストは完了できません。タグなしリソースまたはタグポリシーで定義されていないタグは、タグポリシーに準拠しているかどうか評価されません。

タグポリシーの使用には、複数の AWS サービスの使用が含まれます。

- AWS Organizations を使用してタグポリシーを管理します。組織の管理アカウントにサインインする場合、Organizations を使用してタグポリシー機能を有効にします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。次に、タグポリシーを作成して組織エンティティにアタッチし、タグ付けルールを有効にすることができます。
- AWS Resource Groups を使用して、タグポリシーへの準拠を管理します。組織のアカウントにサインインする場合は、Resource Groups を使用してアカウント内のリソースで非準拠のタグを検索します。リソースを作成した AWS サービスで非準拠のタグを修正できます。タグエディタ と Resource Groups Tagging API を使用して、複数の サービスからリソースにタグを付けたりタグを解除したりすることもできます。

  組織の管理アカウントにサインインすると、組織のすべてのアカウントのコンプライアンス情報を表示できます。

タグポリシーは、すべての機能が有効になっている組織でのみ使用できます。タグポリシーを使用するための詳しい要件については、「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」を参照してください。

> **Important**
>
> タグポリシーの使用を開始するには、 では、より高度なタグポリシーに進むタグポリシーの開始方法前に、「」で説明されているサンプルワークフローに従うことを AWS 強くお勧めします。OU または組織全体にタグポリシーを展開する前に、単純なタグポリシーを単一のアカウントにアタッチした場合の影響を理解しておくことをお勧めします。タグポリシーへの準拠を強制する前に、タグポリシーの影響を理解することは特に重要です。タグポリシーの開始方法 ページの表には、より高度なポリシー関連タスクの手順へのリンクも記載されています。

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

このページでは、AWS Organizations のタグポリシーを管理するための前提条件と必要なアクセス許可について説明します。

トピック

タグポリシーを管理するための前提条件

タグポリシーを使用するには、次が必要です。

- 組織で、すべての機能が有効になっている必要があります。
- 組織の管理アカウントにサインインする必要があります。
- 「タグポリシーを管理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。

タグポリシーへの準拠を評価するには、AWS Resource Groups を使用します 。コンプライアンスを評価するための要件については、AWS Resource Groups ユーザーガイドの「前提条件とアクセス許可」を参照してください。。

タグポリシーを管理するためのアクセス許可

以下の IAM ポリシーの例では、タグポリシーを管理するためのアクセス許可を提供します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ManageTagPolicies",
            "Effect": "Allow",
            "Action": [
                "organizations:ListPoliciesForTarget",
                "organizations:ListTargetsForPolicy",
                "organizations:DescribeEffectivePolicy",
                "organizations:DescribePolicy",
                "organizations:ListRoots",
                "organizations:DisableAWSServiceAccess",
                "organizations:DetachPolicy",
                "organizations:DeletePolicy",
```

```
                "organizations:DescribeAccount",
                "organizations:DisablePolicyType",
                "organizations:ListAWSServiceAccessForOrganization",
                "organizations:ListPolicies",
                "organizations:ListAccountsForParent",
                "organizations:ListAccounts",
                "organizations:EnableAWSServiceAccess",
                "organizations:ListCreateAccountStatus",
                "organizations:DescribeOrganization",
                "organizations:UpdatePolicy",
                "organizations:EnablePolicyType",
                "organizations:DescribeOrganizationalUnit",
                "organizations:AttachPolicy",
                "organizations:ListParents",
                "organizations:ListOrganizationalUnitsForParent",
                "organizations:CreatePolicy",
                "organizations:DescribeCreateAccountStatus"
            ],
            "Resource": "*"
        }
    ]
}
```

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

タグポリシー使用のベストプラクティス

AWS では、タグポリシーを使用する際に以下のベストプラクティスを推奨しています。

タグの大文字小文字に関する方針を決定する

タグを大文字小文字どちらにするかを決め、その方針をすべてのリソースタイプに一貫して実装します。例えば、Costcenter、costcenter、CostCenter のいずれを使用するかを決定し、すべてのタグに同じ規則を使用します。コンプライアンスレポートの結果に一貫性を持たせるには、大文字と小文字が異なる類似タグを使用しないでください。この方針は、組織のタグポリシーを定義するのに役立ちます。

推奨されるワークフローを使用する

単純なタグポリシーを作成して、小規模なものから始めます。次に、テスト目的で使用できるメンバーアカウントにアタッチします。「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を使用します。

タグ付けルールを決定する

これは、組織のニーズによって異なります。例えば、CostCenterタグが AWS Secrets Manager シークレットにアタッチされるときに、指定された大文字と小文字の処理を使用するように指定できます。準拠タグを定義するタグポリシーを作成し、タグ付けルールを有効にする組織エンティティにアタッチします。

アカウント管理者を教育する

タグポリシーの使用範囲を広げる準備ができたら、アカウント管理者を次のように教育します。

- タグ付け方針を伝えます。
- 管理者は特定のリソースタイプでタグを使用する必要があることを強調します。

  タグなしリソースは、コンプライアンス結果で非準拠と表示されないため、これが重要となります。
- タグポリシーへの準拠を確認するためのガイダンスを提供します。リソースのタグ付けユーザーガイドの「 アカウントのコンプライアンスの評価」で説明されている手順を使用して、 アカウントのリソースの非準拠のタグを検索して修正するように管理者に指示します。 AWS コンプライアンスをチェックする頻度を知らせます。

コンプライアンスの強制には注意が必要です。

コンプライアンスの強制によって、組織のアカウントのユーザーが必要なリソースにタグ付けできなくなる可能性があります。「強制について」の情報を確認します。「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も参照してください。

リソース作成リクエストにガードレールを設定するための SCP の作成を検討する

タグが付けられたことがないリソースは、レポートで非準拠として表示されません。アカウント管理者は、それでもタグなしリソースを作成できます。場合によっては、サービスコントロールポリシー (SCP) を使用して、リソース作成リクエストの周囲にガードレールを設定できます。SCP の例については、「 作成される特定のリソースにタグを要求する」を参照してください。 AWS のサービスがタグを使用したアクセスの制御をサポートしているかどうかを確認するには、AWS IAM ユーザーガイドの「IAM と連携する のサービス」を参照してください。[Authorization based on tags] (タグに基づく認可) 列で [Yes] (はい) が表示されているサービスを探します。サービスの名前を選択すると、そのサービスの認証とアクセスコントロールに関するドキュメントが表示されます。

タグポリシーの開始方法

タグポリシーの使用には、複数の AWS サービスの使用が含まれます。開始するには、次のページを参照してください。次に、このページのワークフローに従って、タグポリシーとその影響について理解を深めます。

- 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
- タグポリシー使用のベストプラクティス

初めてのタグポリシーの使用

タグポリシーを初めて使用するには、次の手順に従います。

タスク	サインインするアカウント	AWS を使用する サービスコンソール
ステップ 1: 組織のタグポリシーを有効にします。	組織の管理アカウント[1]	AWS Organizations
ステップ 2: タグポリシーを作成します。 最初のタグポリシーはシンプルにするよう心がけます。使用する大文字小文字の処理に 1 つのタグキーを入力し、その他のオプションはすべてデフォルトのままにします。	組織の管理アカウント[1]	AWS Organizations
ステップ 3: テストに使用できる単一のメンバーアカウントにタグポリシーをアタッチします。 次のステップでは、このアカウントにサインインする必要があります。	組織の管理アカウント[1]	AWS Organizations

タスク	サインインするアカウント	AWS を使用する サービスコンソール
ステップ 4: 準拠タグを持つリソースと、非準拠タグを持つリソースを作成します。	テスト目的で使用するメンバーアカウント。	使い慣れた AWS サービス。例えば、AWS Secrets Manager を使って「基本的なシークレットを作成する」の手順に従い、準拠したシークレットと非準拠のシークレットを持つシークレットを作成できます。
ステップ 5: 有効なタグポリシーを表示し、アカウントのコンプライアンス状況を評価します。	テスト目的で使用するメンバーアカウント。	Resource Groups と、リソースが作成された AWS サービス。 準拠タグと非準拠タグを使用してリソースを作成した場合、結果に非準拠タグが表示されます。
ステップ 6: テストアカウントのリソースがタグポリシーに準拠するまで、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します。	テスト目的で使用するメンバーアカウント。	Resource Groups とリソースが作成された AWS サービス。
いつでも、組織全体のコンプライアンスを評価できます。	組織の管理アカウント[1]	リソースグループ

[1] 組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。

タグポリシーの使用の拡大

次のタスクを任意の順序で実行すると、タグポリシーの使用を拡張できます。

高度なタスク	サインインするアカウント	AWS を使用する サービスコンソール
より高度なタグポリシーを作成します。 初回ユーザーと同じプロセスを実行しますが、他のタスクを試します。例えば、追加のキーや値を定義したり、タグキーに対して異なる大文字小文字の処理を指定したりします。 「管理ポリシーの継承を理解する」および「タグポリシー構文」の情報を使用して、より詳細なタグポリシーを作成できます。	組織の管理アカウント[1]	AWS Organizations
タグポリシーを追加のアカウントまたは OU にアタッチします。 アカウントまたはアカウントがメンバーである OU にさらにポリシーをアタッチした後、そのアカウントの有効なタグポリシーを確認します。	組織の管理アカウント[1]	AWS Organizations
新しいリソースが作成されたら、タグをリクエストする SCP を作成します。例については、 作成される特定のリソースにタグを要求するを参照してください。	組織の管理アカウント[1]	AWS Organizations

高度なタスク	サインインするアカウント	AWS を使用する サービスコンソール
有効なタグポリシーが変更されるたびに、アカウントのコンプライアンスステータスが引き続き評価します。非準拠タグを修正します。	有効なタグポリシーを持つメンバーアカウント。	Resource Groups と、リソースが作成された AWS サービス。
組織全体のコンプライアンスを評価します。	組織の管理アカウント[1]	リソースグループ

[1] 組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。

初めてのタグポリシーの強制

初めてタグポリシーを強制するには、タグポリシーを初めて使用する場合と同じワークフローに従い、テストアカウントを使用します。

> **Warning**
>
> コンプライアンスの強制には注意が必要です。タグポリシーを使用した場合の影響を理解し、推奨されるワークフローに従ってください。テストアカウントで強制の仕組みをテストしてから、他のアカウントに展開します。そうしないと、組織のアカウントのユーザーが必要なリソースにタグ付けできなくなる可能性があります。詳細については、「強制について」を参照してください。

強制タスク	サインインするアカウント	AWS を使用する サービスコンソール
ステップ 1: タグポリシーを作成します。 最初に強制するタグポリシーはシンプルにするよう心がけます。使用する大文字小文字	組織の管理アカウント[1]	AWS Organizations

強制タスク	サインインするアカウント	AWS を使用する サービスコンソール
の処理にタグキーを 1 つ入力し、[Prevent noncompliant operations for this tag (このタグに対する非準拠操作を防止する)] オプションを選択します。次に、適用するリソースタイプを 1 つ指定します。前述の例の続きとして、Secrets Manager シークレットに強制するよう選択します。		
ステップ 2: タグポリシーを単一のテストアカウントにアタッチします。	組織の管理アカウント[1]	AWS Organizations
ステップ 3: 準拠タグを持つリソースと、非準拠タグを持つリソースを作成してみます。タグポリシーで指定されたタイプのリソースに、非準拠のタグを作成することはできません。	テスト目的で使用するメンバーアカウント。	使い慣れた AWS サービス。例えば、AWS Secrets Manager を使って「基本的なシークレットを作成する」の手順に従い、準拠したシークレットと非準拠のシークレットを持つシークレットを作成できます。
手順 4: 有効なタグポリシーに対してアカウントのコンプライアンスステータスを評価し、非準拠タグを修正します。	テスト目的で使用するメンバーアカウント。	Resource Groups と、リソースが作成された AWS サービス。
ステップ 5: テストアカウントのリソースがタグポリシーに準拠するまで、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します。	テスト目的で使用するメンバーアカウント。	Resource Groups と、リソースが作成された AWS サービス。

強制タスク	サインインするアカウント	AWS を使用する サービスコンソール
いつでも、組織全体のコンプライアンスを評価できます。	組織の管理アカウント[1]	リソースグループ

[1] 組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。

タグポリシーの作成、更新、削除

このトピックの内容

- 組織のタグポリシーを有効にすると、ポリシーを作成できます。
- タグ要件が変更されると、既存のポリシーを更新できます。
- ポリシーが不要になった場合、すべての組織単位 (OU) およびアカウントからポリシーをデタッチした後、ポリシーを削除できます。

> **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。

タグポリシーの作成

> **最小アクセス許可**
> タグポリシーを作成するには、次のアクションを実行するためのアクセス権限が必要です。
>
> - `organizations:CreatePolicy`

でタグポリシーを作成するには、次の 2 AWS Management Console つの方法のいずれかがあります。

- オプションを選択し、JSON ポリシーテキストを生成できるビジュアルエディタ。
- JSON ポリシーテキストを直接作成できるテキストエディタ。

ビジュアルエディタを使用すると、プロセスが簡単になりますが、柔軟性は制限されます。これは、最初のポリシーを作成し、使用に慣れるのに最適な方法です。これらの機能の仕組みを理解し、ビジュアルエディタが提供するものによって制限され始めたら、JSON ポリシーテキストを自分で編集して、ポリシーに高度な機能を追加できます。ビジュアルエディタは、@@assign 値設定演算子のみを使用し、子制御演算子へのアクセスは提供しません。子制御演算子は、JSON ポリシーテキストを手動で編集した場合にのみ追加できます。

AWS Management Console

タグポリシーを作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. タグポリシーページで、[Create policy] (ポリシーの作成) を選択します。
3. [Create policy] (ポリシーの作成) ページで、ポリシー名と、オプションでポリシーの説明を入力します。
4. (オプション) ポリシーオブジェクト自体には 1 つ以上のタグを追加できます。これらのタグはポリシーの一部ではありません。これを行うには、[Add tag] (タグの追加) を選択してから、キーとオプションの値を入力します。値を空白のままにすると、空の文字列が設定され、`null` にはなりません。1 つのポリシーに最大 50 個のタグをアタッチできます。詳細については、「AWS Organizations リソースのタグ付け」を参照してください。
5. この手順で説明するように、ビジュアルエディタを使用してタグポリシーを構築できます。[JSON] タブでタグポリシーを入力またはペーストすることもできます。タグポリシーの構文については、タグポリシー構文 を参照してください。

   [New tag key 1] (新しいタグキー 1) で、追加するタグキーの名前を指定します。
6. [Tag key capitalization compliance] (タグキーの大文字と小文字のコンプライアンス) で、このオプションをオフ (デフォルト) のままにして、継承された親タグポリシーが存在する場合にタグキーの大文字と小文字の処理を定義するように指定します。

   このポリシーを使用してタグキーの大文字と小文字を区別する場合は、このオプションを有効にします。このオプションを選択すると、[タグキー] に指定した大文字と小文字は、継承された親ポリシーで指定された大文字と小文字の処理より優先されます。

親ポリシーが存在せず、このオプションを有効にしない場合、タグキーがすべて小文字のものだけが準拠していると見なされます。親ポリシーからの継承の詳細については、「管理ポリシーの継承を理解する」を参照してください。

> **Tip**
>
> タグキーとその大文字小文字の処理を定義するタグポリシーを作成する際のガイドとして、「例 1: 組織全体のタグキーの大文字小文字取り扱いの定義」に示すタグポリシーの例を使用することを検討してください。組織のルートにアタッチします。後で追加のタグポリシーを作成し、OU またはアカウントにアタッチして、追加のタグ付けルールを作成できます。

7. [タグ値コンプライアンス] で、このタグキーに許可される値を親ポリシーから継承された値に追加する場合は、このオプションを有効にします。

   デフォルトでは、このオプションはオフになっています。つまり、親ポリシーで定義され、親ポリシーから継承された値だけが準拠していると見なされます。親ポリシーが存在しない場合、またはタグ値を指定しない場合、すべての値 (値なしの場合を含む) が準拠していると見なされます。

   受け入れ可能なタグ値のリストを更新するには、[Specify allowed values for this tag key] (このタグキーに許可される値を指定する) を選択し、[Specify values] (値を指定) を選択します。プロンプトが表示されたら、新しい値を入力し (ボックスごとに 1 つの値)、[Save changes] (変更の保存) を選択します。

8. [Prevent noncompliant operations for this tag] (このタグの非準拠操作を防止します) については、タグポリシーの使用経験がない場合は、このオプションをオフ (デフォルト) のままにしておくことをお勧めします。「強制について」の推奨事項を確認し、完全なテストを実施してください。そうしないと、組織のアカウントのユーザーが必要なリソースにタグ付けできなくなる可能性があります。

   このタグキーへの準拠を強制する場合は、チェックボックスをオンにしてから [Specify resource types] (リソースタイプを指定) を選択します。プロンプトが表示されたら、ポリシーに含めるリソースタイプを選択します。次に、変更の保存を選択します。

> **Important**
>
> このオプションを選択すると、指定したタイプのリソースのタグを操作するオペレーションは、そのオペレーションの結果としてポリシーに準拠するタグが得られた場合にのみ成功します。

9. (オプション) このタグポリシーに別のタグキーを追加するには、[Add tag key] を選択します。次に、ステップ 6 ~ 9 を実行してタグキーを定義します。
10. タグポリシーの構築が完了したら、[Save changes] (変更を保存) を選択します。

AWS CLI & AWS SDKs

タグポリシーを作成するには

次のいずれかを使用して、タグポリシーを作成できます。

- AWS CLI: create-policy

タグポリシーの作成には任意のテキスト工ディタを使用できます。JSON 構文を使用し、タグポリシーを任意の名前と拡張子を持つファイルとして任意の場所に保存します。タグポリシーには、スペースを含めて最大 2,500 文字を使用できます。タグポリシーの構文については、タグポリシー構文 を参照してください。

タグポリシーを作成するには

1. 以下のようなタグポリシーのテキストファイルを作成します。

   testpolicy.json の内容:

```
{
    "tags": {
        "CostCenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            }
        }
    }
}
```

このタグポリシーは、CostCenter タグキーを定義します。タグは任意の値を受け入れることができますが、値を受け入れなくても構いません。このようなポリシーは、値の有無にかかわらず CostCenter タグがアタッチされたリソースが準拠していることを意味します。

2. ファイルにあるポリシーの内容を含むポリシーを作成します。出力が読みやすくなるように、余分な余白は切り詰められています。

```
$ aws organizations create-policy \
    --name "MyTestTagPolicy" \
    --description "My Test policy" \
    --content file://testpolicy.json \
    --type TAG_POLICY
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-a1b2c3d4e5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
tag_policy/p-a1b2c3d4e5",
            "Name": "MyTestTagPolicy",
            "Description": "My Test policy",
            "Type": "TAG_POLICY",
            "AwsManaged": false
        },
        "Content": "{\n\"tags\":{\n\"CostCenter\":{\n\"tag_key\":{\n\"@@assign
\":\"CostCenter\"\n}\n}\n}\n}\n\n"
    }
}
```

- AWS SDKs CreatePolicy

次のステップ

タグポリシーを作成したら、タグ付けルールを有効にできます。そのためには、組織ルート、組織単位 (OUs)、組織 AWS アカウント 内、または組織エンティティの組み合わせにポリシーをアタッチします。

タグポリシーの更新

> **最小アクセス許可**
>
> タグポリシーを更新するには、次のアクションを実行する権限が必要です。
>
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:UpdatePolicy`。
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:DescribePolicy`。

AWS Management Console

タグポリシーを更新するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [Tag policies] (タグポリシー) ページで、更新するタグポリシーを選択します。
3. [Edit policy] (ポリシーの編集) を選択します。
4. 新しいポリシー名とポリシーの説明を入力できます。ポリシーの内容を変更するには、ビジュアルエディタを使用するか、JSON を編集します。
5. タグポリシーの更新が完了したら、[Save changes] (変更を保存) を選択します。

AWS CLI & AWS SDKs

ポリシーを更新するには

次のいずれかを使用して、ポリシーを更新できます。

- AWS CLI: update-policy

  次の例では、タグポリシーの名前を変更します。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --name "Renamed tag policy"
{
```

```
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
tag_policy/p-i9j8k7l6m5",
            "Name": "Renamed tag policy",
            "Type": "TAG_POLICY",
            "AwsManaged": false
        },
        "Content": "{\n\"tags\":{\n\"CostCenter\":{\n\"tag_key\":{\n\"@@assign\":
\"CostCenter\"\n}\n}\n}\n}\n\n"
    }
}
```

次の例では、タグポリシーの説明を追加または変更します。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new tag policy description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
tag_policy/p-i9j8k7l6m5",
            "Name": "Renamed tag policy",
            "Description": "My new tag policy description",
            "Type": "TAG_POLICY",
            "AwsManaged": false
        },
       "Content": "{\n\"tags\":{\n\"CostCenter\":{\n\"tag_key\":{\n\"@@assign\":
\"CostCenter\"\n}\n}\n}\n}\n\n"
    }
}
```

次の例では、AI サービスのオプトアウトポリシーにアタッチされた JSON ポリシードキュメントを変更しています。この例では、次のようなテキストを含む policy.json というファイルから、内容が取得されています。

```
{
  "tags": {
    "Stage": {
```

```
      "tag_key": {
        "@@assign": "Stage"
      },
      "tag_value": {
        "@@assign": [
          "Production",
          "Test"
        ]
      }
    }
  }
}
```

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --content file://policy.json
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
tag_policy/p-i9j8k7l6m5",
            "Name": "Renamed tag policy",
            "Description": "My new tag policy description",
            "Type": "TAG_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"tags\":{\"Stage\":{\"tag_key\":{\"@@assign\":\"Stage
\"},\"tag_value\":{\"@@assign\":[\"Production\",\"Test\"]},\"enforced_for\":
{\"@@assign\":[\"ec2:instance\"]}}}}"
}
```

- AWS SDKs UpdatePolicy

タグポリシーにアタッチされたタグを編集する

組織の管理アカウントにサインインすると、タグポリシーにアタッチされたタグを追加または削除できます。そのためには、以下の手順を完了します。

最小アクセス許可

AWS 組織内のタグポリシーにアタッチされたタグを編集するには、次のアクセス許可が必要です。

- `organizations:DescribeOrganization` (コンソールのみ - ポリシーに移動するために使用)
- `organizations:DescribePolicy` (コンソールのみ - ポリシーに移動するために使用)
- `organizations:TagResource`
- `organizations:UntagResource`

AWS Management Console

AI サービスのオプトアウトポリシーにアタッチされたタグを編集するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [Tag policies] (タグポリシー) ページで、タグを編集するポリシーの名前を選択します。
3. 選択したポリシーの詳細ページで、[Tags] (タグ) タブ、[Manage tags] (タグ管理) の順に選択します。
4. このページでは次のアクションを実行できます。
   - 古い値に上書きして新しい値を入力し、任意のタグの値を編集します。キーは変更できません。キーを変更するには、古いキーを持つタグを削除し、新しいキーを持つタグを追加する必要があります。
   - [Remove] (削除) を選択すると、既存のタグが削除されます。
   - 新しいタグのキーと値のペアを追加します。[Add tag] (タグの追加) を選択し、表示されたボックスに新しいキー名とオプションの値を入力します。[Value] (値) ボックスを空白のままにすると、値は空の文字列に設定され、`null` にはなりません。
5. 必要な追加、削除、編集をすべて終えたら、[Save changes] (変更の保存) を選択します。

AWS CLI & AWS SDKs

タグポリシーにアタッチされたタグを編集するには

次のいずれかのコマンドを使用して、タグポリシーにアタッチされたタグを編集できます。

- AWS CLI: tag-resource および untag-resource
- AWS SDKsTagResourceおよび UntagResource

タグポリシーの削除

組織の管理アカウントにサインインすると、組織に不要になったポリシーを削除できます。

ポリシーを削除するには、まずそのポリシーをすべての添付エンティティからデタッチする必要があります。

> **最小アクセス許可**
>
> タグポリシーを削除するには、次のアクションを実行する権限が必要です。
>
> - `organizations:DeletePolicy`

AWS Management Console

タグポリシーを削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2.
3. [Tag policies] (タグポリシー) ページで、削除するタグポリシーを選択します。
4. 削除するポリシーは、まず、すべてのルート、OU、アカウントからデタッチする必要があります。[Targets] (ターゲット) タブで、[Targets] (ターゲット) リストに表示されている各ルート、OU、またはアカウントの横にあるラジオボタンをクリックしてから、[Detach] (デタッチ) を選択します。確認ダイアログボックスで、[Detach] (デタッチ) を選択します。
5. ページの上部で、[Delete] (削除) を選択します。
6. 確認ダイアログボックスで、ポリシーの名前を入力し、[Delete] (削除) を選択します。

AWS CLI & AWS SDKs

バックアップポリシーを削除するには

以下のコード例は、DeletePolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Deletes an existing AWS Organizations policy.
/// </summary>
public class DeletePolicy
{
    /// <summary>
    /// Initializes the Organizations client object and then uses it to
    /// delete the policy with the specified policyId.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var policyId = "p-00000000";

        var request = new DeletePolicyRequest
        {
            PolicyId = policyId,
        };

        var response = await client.DeletePolicyAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
```

```
                Console.WriteLine($"Successfully deleted Policy: {policyId}.");
            }
            else
            {
                Console.WriteLine($"Could not delete Policy: {policyId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeletePolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDeletePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def delete_policy(policy_id, orgs_client):
```

```
    """
    Deletes a policy.

    :param policy_id: The ID of the policy to delete.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.delete_policy(PolicyId=policy_id)
        logger.info("Deleted policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_id)
        raise
```

- API の詳細については、 DeletePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

タグポリシーをアタッチおよびデタッチする

タグポリシーは、組織全体、組織単位 (OU)、個々のアカウントで使用できます。

- タグポリシーを組織ルートにアタッチすると、タグポリシーはルートのすべてのメンバー OU とアカウントに適用されます。
- OU にタグポリシーをアタッチすると、そのタグポリシーは OU に属するアカウントに適用されます。これらのアカウントには、組織ルートにアタッチされたタグポリシーも適用されます。
- タグポリシーをアカウントにアタッチすると、そのタグポリシーがアカウントに適用されます。さらに、そのアカウントには、組織ルートにアタッチされたタグポリシーと、そのアカウントが属する OU にアタッチされたタグポリシーが適用されます。

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が、有効なタグポリシーになります。詳細については、「管理ポリシーの継承を理解する」を参照してください。

> **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。

最小アクセス許可

タグポリシーをアタッチするには、次のアクションを実行する権限が必要です。

- `organizations:AttachPolicy`

AWS Management Console

タグポリシーをアタッチするには、ポリシーをアタッチするルート、OU、またはアカウントに移動します。

ルート、OU、またはアカウントに移動してタグポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをアタッチするルート、OU、またはアカウントを見つけ、名前を選択します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
3. [Policies] (ポリシー) タブの [Tag policies] (タグポリシー) で、[Attach] (アタッチ) を選択します。
4. 目的のポリシーを見つけて [Attach policy] (ポリシーのアタッチ) を選択します。

   [Policies] (ポリシー) タブで、アタッチされているタグポリシーのリストが更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動してタグポリシーをアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. タグポリシーページで、アタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで [Attach] (アタッチ) を選択します。

4. ポリシーをアタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。
5. Attach policy] (ポリシーのアタッチ) を選択します。

   [Targets] (ターゲット) タブで、アタッチされているタグポリシーのリストが更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

組織のルート、OU、またはアカウントにタグポリシーをアタッチするには

以下のコード例は、`AttachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to attach an AWS Organizations policy to an organization,
/// an organizational unit, or an account.
/// </summary>
public class AttachPolicy
{
    /// <summary>
    /// Initializes the Organizations client object and then calls the
    /// AttachPolicyAsync method to attach the policy to the root
    /// organization.
```

```
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();
            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new AttachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.AttachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully attached Policy ID {policyId} to
Target ID: {targetId}.");
            }
            else
            {
                Console.WriteLine("Was not successful in attaching the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスAttachPolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシーを OU にアタッチする方法を示しています。

```
aws organizations attach-policy
```

```
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- API の詳細については、「 コマンドリファレンスAttachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def attach_policy(policy_id, target_id, orgs_client):
    """
    Attaches a policy to a target. The target is an organization root, account,
 or
    organizational unit.

    :param policy_id: The ID of the policy to attach.
    :param target_id: The ID of the resources to attach the policy to.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.attach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Attached policy %s to target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
```

```
            "Couldn't attach policy %s to target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 AttachPolicy AWS SDK for Python (Boto3) API リファレンス の「」を参照してください。

ポリシーの変更はすぐに有効になります。

次のステップ

タグポリシーをアタッチすると、リソースがそのタグポリシーにどの程度準拠しているかを確認できます。これを行うには、Resource Groups コンソールを使用します。詳細については、「 リソースのタグ付けユーザーガイド」の「 アカウントのコンプライアンスの評価」を参照してください。AWS

タグポリシーのデタッチ

組織の管理アカウントにサインインすると、アタッチされている組織ルート、OU、またはアカウントからタグポリシーをデタッチすることができます。エンティティからタグポリシーをデタッチすると、そのポリシーは、現在デタッチされたエンティティによって影響を受けたすべてのアカウントに適用されなくなります。ポリシーをデタッチするには、次のステップを実行します。

> 最小アクセス許可
>
> 組織ルート、OU、またはアカウントからタグポリシーをデタッチするには、以下のアクションを実行する権限が必要です。
>
> - `organizations:DetachPolicy`

AWS Management Console

タグポリシーをデタッチするには、ポリシーをデタッチするルート、OU、またはアカウントに移動します。

タグポリシーがアタッチされているルート、OU、またはアカウントに移動してデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをデタッチするルート、OU、またはアカウントに移動します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。ルート、OU、またはアカウントの名前を選択します。
3. [Policies] (ポリシー) タブで、デタッチするタグポリシーの横にあるラジオボタンをクリックし、[Detach] (デタッチ) を選択します。
4. 確認ダイアログボックスで、[Detach policy] (ポリシーのデタッチ) を選択します。

   アタッチされているタグポリシーのリストが更新されます。ポリシーの変更はすぐに反映されます。

ポリシーに移動してタグポリシーをデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. タグポリシーページで、ルート、OU、またはアカウントからデタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで、ポリシーをデタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (
を選択) する必要があります。
4. [Detach] (デタッチ) を選択します。
5. 確認ダイアログボックスで、[Detach] (デタッチ) を選択します。

   アタッチされているタグポリシーのリストが更新されます。ポリシーの変更はすぐに反映されます。

AWS CLI & AWS SDKs

組織のルート、OU、またはアカウントからタグポリシーをデタッチするには

以下のコード例は、DetachPolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to detach a policy from an AWS Organizations organization,
/// organizational unit, or account.
/// </summary>
public class DetachPolicy
{
    /// <summary>
    /// Initializes the Organizations client object and uses it to call
    /// DetachPolicyAsync to detach the policy.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var policyId = "p-00000000";
        var targetId = "r-0000";

        var request = new DetachPolicyRequest
        {
            PolicyId = policyId,
```

```
                TargetId = targetId,
            };

            var response = await client.DetachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully detached policy with Policy Id:
{policyId}.");
            }
            else
            {
                Console.WriteLine("Could not detach the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDetachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111
 --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDetachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def detach_policy(policy_id, target_id, orgs_client):
    """
    Detaches a policy from a target.

    :param policy_id: The ID of the policy to detach.
    :param target_id: The ID of the resource where the policy is currently
 attached.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.detach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Detached policy %s from target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 DetachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに反映されます。

有効なタグポリシーの表示

アカウント内のタグ付きリソースのコンプライアンス状態の確認を開始する前に、まずアカウントの有効なタグポリシーを特定しておくと便利です。

有効なタグポリシーとは

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。これは、アカウントが継承するすべてのタグポリシーと、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。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。

例えば、組織ルートにアタッチされたタグポリシーで、4 つの準拠値を持つ `CostCenter` タグを定義するとします。この場合、アカウントにアタッチされた別のタグポリシーで、 `CostCenter` キーを 4 つの準拠値のうち 2 つだけに制限できます。これらのタグポリシーを組み合わせることにより、有効なタグポリシーが構成されます。その結果、組織ルートのタグポリシーで定義されている 4 つの準拠タグ値のうち 2 つだけがアカウントで準拠することになります。

有効なタグポリシーの生成方法の詳細と詳しい例については、「管理ポリシーの継承を理解する」を参照してください 。

有効なタグポリシーの表示方法

アカウントの有効なタグポリシーは、AWS Management Console、AWS API 、または AWS Command Line Interface から表示できます。

> **最小アクセス許可**
>
> アカウントの有効なタグポリシーを表示するには、以下のアクションを実行する権限が必要です。
>
> - `organizations:DescribeEffectivePolicy`
> - `organizations:DescribeOrganization`

AWS Management Console

アカウントの有効なポリシーを表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. AWS アカウント ページで、有効なタグポリシーを表示するアカウントの名前を選択します。場合によっては、目的のアカウントを見つけるには OU を展開

(▶
を選択) する必要があります。

3. [Policies] (ポリシー) タブの [Tag policies] (タグポリシー) セクションで、[View the effective tag policy for this AWS アカウント] (この AWS アカウント の有効なタグポリシーを表示する) を選択します。

   指定したアカウントに適用されている有効なポリシーがコンソールに表示されます。

   > **Note**
   >
   > 有効なポリシーをコピーして貼り付けて、大きな変更を加えずに別のタグポリシーの JSON として使用することはできません。タグポリシードキュメントには、各設定を最終的な有効なポリシーにマージする方法を指定する 継承演算子 を含める必要があります。

AWS CLI & AWS SDKs

アカウントの有効なポリシーを表示するには

次のいずれかの方法を使用して、有効なタグポリシーを表示できます。

- AWS CLI: describe-effective-policy

  アカウントによって継承された、またはアカウントにアタッチされているタグ付けルールを決定するには、アカウントから次のコマンドを実行し、結果をファイルに保存します。

```
$ aws organizations describe-effective-policy \
    --policy-type TAG_POLICY
{
    "EffectivePolicy": {
        "PolicyContent": "{\"tags\":{\"costcenter\":{\"tag_value\":[\"*\"],
\"tag_key\":\"CostCenter\"}}}",
        "LastUpdatedTimestamp": "2020-06-09T08:34:25.103000-07:00",
        "TargetId": "123456789012",
        "PolicyType": "TAG_POLICY"
    }
}
```

タグポリシーがアカウントとルート、またはすべての OU にアタッチされている場合、アカウントの有効なタグポリシーは継承されたすべてのポリシーの組み合わせによって定義されます。このような場合、アカウントから `describe-effective-policy` を実行すると、アカウントの階層内のすべてのタグポリシーのマージされた内容が返されます。

- AWS SDK: DescribeEffectivePolicy

Amazon EventBridge を使用した非準拠タグのモニタリング

Amazon EventBridge (旧 Amazon CloudWatch Events) を使用すると、非準拠タグが導入された場合にモニタリングできます。次のイベント例では、`tag-policy-compliant` の `"false"` 値が、新しいタグが有効なタグポリシーに準拠していないことを示します。

```
{
  "detail-type": "Tag Change on Resource",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa"
  ],
  "detail": {
    "changed-tag-keys": [
      "a-new-key"
    ],
    "service": "ec2",
    "resource-type": "instance",
    "version": 3,
    "tag-policy-compliant": "false",
    "tags": {
      "a-new-key": "tag-value-on-new-key-just-added"
    }
  }
}
```

イベントにサブスクライブし、監視する文字列またはパターンを指定できます。EventBridge の詳細は、「Amazon EventBridge ユーザーガイド」を参照してください。

強制について

タグポリシーは、指定したリソースタイプで非準拠のタグ付けオペレーションを強制するように指定できます。つまり、指定されたリソースタイプで非準拠のタグ付けリクエストは完了できません。

> **Important**
>
> 強制は、タグなしで作成されたリソースには影響しません。

タグポリシーへのコンプライアンスを強制するには、タグポリシーを作成するときに、次のいずれかの操作を行います。

- [Visual editor (ビジュアルエディタ)] タブで、[Prevent noncompliant operations for this tag (このタグに対する非準拠のオペレーションを禁止する)] を選択します。
- [JSON] タブで、`enforced_for` フィールドを使用します。タグポリシーの構文については、「タグポリシーの構文と例」を参照してください。

タグポリシーへの準拠を強制するには、次のベストプラクティスに従います。

- コンプライアンスの強制には注意が必要です - タグポリシーを使用した場合の影響を理解し、「タグポリシーの開始方法」で説明されている推奨ワークフローに従ってください。テストアカウントで強制の仕組みをテストしてから、他のアカウントに展開します。そうしないと、組織のアカウントのユーザーが必要なリソースにタグ付けできなくなる可能性があります。
- 強制できるリソースタイプに注意する - サポートされているリソースタイプに対してのみ、タグポリシーへの準拠を強制できます。コンプライアンスの強制をサポートするリソースタイプは、ビジュアルエディタを使用してタグポリシーを構築するときに一覧表示されます。
- 一部の サービスとのやり取りを理解する – 一部の AWS サービスには、自動的にリソースを作成するコンテナのようなリソースのグループがあり、タグはあるサービスのリソースから別のサービスに伝播できます。例えば、Amazon EC2 Auto Scaling グループと Amazon EMR クラスターのタグは、それら含む Amazon EC2 インスタンスに自動的に伝播します。Auto Scaling グループまたは EMR クラスターよりも厳しい Amazon EC2 のタグポリシーがある場合があります。強制を有効にすると、タグポリシーによってリソースにタグ付けできなくなり、動的なスケーリングおよびプロビジョニングがブロックされる可能性があります。

次のセクションでは、非準拠のリソースを見つけ、準拠するものに修正する方法を説明します。

アカウントの非準拠リソースの検索

各アカウントについて、非準拠のリソースに関する情報を取得できます。このコマンドは、アカウントにリソースがあるすべてのリージョンから実行する必要があります。

タグポリシーを持つアカウントの非準拠リソースを検索するには、次のコマンドを実行して結果をファイルに保存します。

```
$ aws resourcegroupstaggingapi get-resources  --region us-east-1 \
    --include-compliance-details \
    --exclude-compliant-resources > outputfile.txt
```

リソース内の非準拠タグの修正

非準拠タグを見つけたら、以下のいずれかの方法を使用して修正します。非準拠タグの付いたリソースがあるアカウントにサインインする必要があります。

- 非準拠リソースを作成した AWS サービスのコンソールまたはタグ付け API オペレーションを使用します。
- および AWS Resource Groups TagResourcesUntagResourcesオペレーションを使用して、有効なポリシーに準拠するタグを追加したり、非準拠のタグを削除したりできます。

その他の非準拠問題の検出と修正

コンプライアンスの問題を見つけて修正することは、反復的なプロセスです。関心のあるリソースがタグポリシーに準拠するまで、前の 2 つのセクションのステップを繰り返します。

組織全体のコンプライアンスレポートの生成

組織全体の でタグ付けされたすべてのリソースを一覧表示するレポートは、いつでも生成 AWS アカウント できます。レポートには、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が表示されます。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反映されるまで、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ます。

us-east-1 リージョンの組織の管理アカウントが Amazon S3 バケットにアクセスできる場合、このアカウントからレポートを生成できます。「Amazon S3 Bucket 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。レポートを生成するには、次のコマンドを実行します。

```
$ aws resourcegroupstaggingapi start-report-creation --region us-east-1
```

一度に生成できるレポートは 1 つです。

このレポートは完了までに時間がかかる場合があります。ステータスを確認するには、次のコマンドを実行します。

```
$ aws resourcegroupstaggingapi describe-report-creation --region us-east-1
{
    "Status": "SUCCEEDED"
}
```

上記のコマンドが SUCCEEDED を返したら、Amazon S3 バケットのレポートを開くことができます。

強制をサポートするサービスとリソースタイプ

タグポリシーへのコンプライアンスをサポートするサービスとリソースタイプは以下のとおりです。

サービス名	リソースタイプ	JSON 構文
Amazon API Gateway	• API キー • ドメイン名 • REST API オペレーション • ステージ	• "apigateway:apikeys" • "apigateway:domainnames" • "apigateway:restapis" • "apigateway:restapis/stages"
AWS Amplify	• コンポーネント • テーマ	• "amplifyuibuilder:app/environment/components" • "amplifyuibuilder:app/environment/themes"
AWS AppConfig	• アプリケーション • 設定プロファイル • デプロイ • デプロイ戦略 • 環境	• "appconfig:application" • "appconfig:application/configurationprofile" • "appconfig:application/environment/deployment" • "appconfig:deploymentstrategy" • "appconfig:application/environment"

サービス名	リソースタイプ	JSON 構文
AWS App Mesh	• すべて • ゲートウェイルート • [Mesh] (メッシュ) • ルート • 仮想ゲートウェイ • 仮想ノード • 仮想ルーター • 仮想サービス	• "appmesh:*" • "appmesh:mesh/virtualGateway/gatewayRoute" • "appmesh:mesh" • "appmesh:mesh/virtualRouter/route" • "appmesh:mesh/virtualGateway" • "appmesh:mesh/virtualNode" • "appmesh:mesh/virtualRouter" • "appmesh:mesh/virtualService"
Amazon Athena	• すべて • Workgroup	• "athena:*" • "athena:workgroup"
AWS Audit Manager	• [評価] • 評価フレームワーク • コントロール	• "auditmanager:assessment " • "auditmanager:assessmentFramework " • "auditmanager:control "
AWS Backup	• バックアッププラン • ボールト • ゲートウェイ • ハイパーバイザー • VM	• "backup:backup-plan" • "backup:backup-vault" • "backup-gateway:gateway" • "backup-gateway:hypervisor" • "backup-gateway:vm"
AWS Batch	• ジョブ • ジョブ定義 • ジョブキュー	• "batch:job" • "batch:job-definition" • "batch:job-queue"
AWS BugBust	• イベント	• "bugbust:event"

サービス名	リソースタイプ	JSON 構文
AWS Certificate Manager	• すべて • 証明書 • Private Certificate Authority	• "acm:*" • "acm:certificate" • "acm-pca:certificate-authority"
Amazon Chime	• アプリケーションインスタンス • Channel • メディアパイプライン • 会議 • SIP メディアアプリケーション • ユーザーアプリケーションインスタンス • Voice Connect	• "chime:app-instance" • "chime:app-instance/channel" • "chime:media-pipeline" • "chime:meeting" • "chime:sma" • "chime:app-instance/user" • "chime:vc"
AWS Clean Rooms	• コラボレーション • 設定済みテーブル • メンバーシップ • 設定済みのテーブル関連付け	• "cleanrooms:collaboration" • "cleanrooms:configuredtable" • "cleanrooms:membership" • "cleanrooms:membership/configuredtableassociation"
AWS Cloud9	• 環境	• "cloud9:environment"
Amazon CloudFront	• すべて • ディストリビューション • ストリーミング配信	• "cloudfront:*" • "cloudfront:distribution" • "cloudfront:streaming-distribution"

サービス名	リソースタイプ	JSON 構文
AWS CloudTrail	• すべて • 追跡	• "cloudtrail:*" • "cloudtrail:trail"
Amazon CloudWatch	• すべて • アラーム • Contributor Insights のルール • メトリクスストリーム	• "cloudwatch:*" • "cloudwatch:alarm" • "cloudwatch:insight-rule" • "cloudwatch:metric-stream"
Amazon CloudWatch Internet Monitor	• モニター	• "internetmonitor:monitor"
Amazon CloudWatch Logs	• デスティネーション • ロググループ	• "logs:destination" • "logs:log-group"
Amazon CloudWatch Observability Access Manager	• リンク • シンク	• "oam:link" • "oam:sink"
AWS CodeBuild	• すべて • プロジェクト	• "codebuild:*" • "codebuild:project"
Amazon CodeCatalyst	• 接続	• "codecatalyst:connections"
AWS CodeCommit	• すべて • リポジトリ	• "codecommit:*" • "codecommit:repository"
AWS CodePipeline	• すべて • アクションタイプ • パイプライン • ウェブフック	• "codepipeline:*" • "codepipeline:actiontype" • "codepipeline:pipeline" • "codepipeline:webhook"

サービス名	リソースタイプ	JSON 構文
Amazon Cognito ID	• すべて • ID プール	• "cognito-identity:*" • "cognito-identity:identitypool"
Amazon Cognito ユーザープール	• すべて • ユーザープール	• "cognito-idp:*" • "cognito-idp:userpool"
Amazon Comprehend	• すべて • ドキュメント分類子 • エンティティ認識機能	• "comprehend:*" • "comprehend:document-classifier" • "comprehend:entity-recognizer"
AWS Config	• すべて • 集計の承認 • Config アグリゲータ • Config ルール	• "config:*" • "config:aggregation-authorization" • "config:config-aggregator" • "config:config-rule"
Amazon CodeGuru Reviewer	• 関連付け	• "codeguru-reviewer:association"
Amazon CodeGuru セキュリティ	• Scan	• "codeguru-security:scans"
CodeConnections	• Connection • ホスト	• "codestar-connections:connection" • "codestar-connections:host"

サービス名	リソースタイプ	JSON 構文
Amazon Connect	• 問い合わせフロー • 統合アソシエーション • キュー • クイック接続 • ルーティングプロファイル • ユーザー	• "connect:instance/contact-flow" • "connect:instance/integration-association" • "connect:instance/queue" • "connect:instance/transfer-destination" • "connect:instance/routing-profile" • "connect:instance/agent"
Amazon Connect Wisdom	• Assistant • 関連付け • コンテンツ • ナレッジベース • セッション	• "wisdom:assistant" • "wisdom:association" • "wisdom:content" • "wisdom:knowledge-base" • "wisdom:session"
AWS Database Migration Service	• すべて • エンドポイント • ES • Rep • Subgrp • タスク	• "dms:*" • "dms:endpoint" • "dms:es" • "dms:rep" • "dms:subgrp" • "dms:task"
Amazon Data Lifecycle Manager	• ポリシー	• "dlm:policy"
AWS Diode	• マッピング	• "diode-messaging:mapping"

サービス名	リソースタイプ	JSON 構文
AWS Direct Connect	• すべて • Dxcon • Dxlag • Dxvif	• "directconnect:*" • "directconnect:dxcon" • "directconnect:dxlag" • "directconnect:dxvif"
Amazon DynamoDB	• すべて • テーブル	• "dynamodb:*" • "dynamodb:table"
Amazon EC2	• キャパシティーの予約 • キャパシティ予約フリート • キャリアゲートウェイ	• "ec2:capacity-reservation" • "ec2:capacity-reservation-fleet" • "ec2:carrier-gateway"
	• クライアント VPN エンドポイント • CoIP プール • カスタマーゲートウェイ	• "ec2:client-vpn-endpoint" • "ec2:coip-pool" • "ec2:customer-gateway"
	• 専用ホスト • DHCP オプション • Egress-only インターネットゲートウェイ	• "ec2:dedicated-host" • "ec2:dhcp-options" • "ec2:egress-only-internet-gateway"

サービス名	リソースタイプ	JSON 構文
	• Elastic IP • イベントウィンドウ • イメージのエクスポートタスク • インスタンスのエクスポートタスク • フリート	• "ec2:elastic-ip" • "ec2:instance-event-window" • "ec2:export-image-task" • "ec2:export-instance-task" • "ec2:fleet"
	• FPGA イメージ • ホスト予約 • イメージ	• "ec2:fpga-image" • "ec2:host-reservation" • "ec2:image"
	• イメージのインポートタスク • スナップショットのインポートタスク • インスタンス • インターネットゲートウェイ • IP Address Manager	• "ec2:import-image-task" • "ec2:import-snapshot-task" • "ec2:instance" • "ec2:internet-gateway" • "ec2:ipam"
	• IP Address Manager プール • IP Address Manager スコープ • IPv4 プール	• "ec2:ipam-pool" • "ec2:ipam-scope" • "ec2:ipv4pool-ec2"

サービス名	リソースタイプ	JSON 構文
	• キーペア • 起動テンプレート • ローカルゲートウェイルートテーブル	• "ec2:key-pair" • "ec2:launch-template" • "ec2:local-gateway-route-table"
	• ローカルゲートウェイルートテーブル仮想インターフェイスグループの関連付け • ローカルゲートウェイルートテーブル VPC の関連付け • NAT ゲートウェイ	• "ec2:local-gateway-route-table-virtual-interface-group-association" • "ec2:local-gateway-route-table-vpc-association" • "ec2:natgateway"
	• ネットワーク ACL • ネットワークインターフェイス • Network Insights アクセススコープ	• "ec2:network-acl" • "ec2:network-interface" • "ec2:network-insights-access-scope"
	• Network Insights アクセススコープ分析 • Network Insights 分析 • Network Insights パス	• "ec2:network-insights-access-scope-analysis" • "ec2:network-insights-analysis" • "ec2:network-insights-path"

サービス名	リソースタイプ	JSON 構文
	• プレイスメントグループ • プレフィックスリスト • ルートボリュームの置き換えタスク	• "ec2:placement-group" • "ec2:prefix-list" • "ec2:replace-root-volume-task"
	• リザーブドインスタンス • ルートテーブル • セキュリティグループ	• "ec2:reserved-instances" • "ec2:route-table" • "ec2:security-group"
	• Snapshot • スポットフリートリクエスト • スポットインスタンスリクエスト • サブネット	• "ec2:snapshot" • "ec2:spot-fleet-request" • "ec2:spot-instances-request" • "ec2:subnet"
	• サブネット CIDR 予約 • Traffic mirror フィルター • Traffic mirror セッション	• "ec2:subnet-cidr-reservation" • "ec2:traffic-mirror-filter" • "ec2:traffic-mirror-session"

サービス名	リソースタイプ	JSON 構文
	• Traffic mirror ターゲット • トランジットゲートウェイ • トランジットゲートウェイアタッチメント	• "ec2:traffic-mirror-target" • "ec2:transit-gateway" • "ec2:transit-gateway-attachment"
	• Transit Gateway Connect ピア • Transit Gateway マルチキャストドメイン • トランジットゲートウェイポリシーテーブル	• "ec2:transit-gateway-connect-peer" • "ec2:transit-gateway-multicast-domain" • "ec2:transit-gateway-policy-table"
	• トランジットゲートウェイルートテーブル • トランジットゲートウェイルートテーブルのお知らせ • Verified Access エンドポイント • Verified Access グループ	• "ec2:transit-gateway-route-table" • "ec2:transit-gateway-route-table-announcement" • "ec2:verified-access-endpoint" • "ec2:verified-access-group"

サービス名	リソースタイプ	JSON 構文
	• Verified Access インスタンス • Verified Access 信頼プロバイダー • ボリューム	• "ec2:verified-access-instance" • "ec2:verified-access-trust-provider" • "ec2:volume"
	• VPC フローログ • VPC • VPC エンドポイント	• "ec2:vpc-flow-log" • "ec2:vpc" • "ec2:vpc-endpoint"
	• VPC エンドポイントサービス • VPC ピア接続 • VPN 接続 • VPN ゲートウェイ	• "ec2:vpc-endpoint-service" • "ec2:vpc-peering-connection" • "ec2:vpn-connection" • "ec2:vpn-gateway"
Amazon EC2 のごみ箱	• ルール	• "rbin:rule"
AWS Elastic Beanstalk	• アプリケーション • アプリケーションバージョン • 設定テンプレート • プラットフォーム	• "elasticbeanstalk:application" • "elasticbeanstalk:applicationversion" • "elasticbeanstalk:configurationtemplate" • "elasticbeanstalk:platform"
Amazon Elastic Container Registry	• リポジトリ	• "ecr:repository"

サービス名	リソースタイプ	JSON 構文
Amazon Elastic Container Service	• キャパシティープロバイダー • クラスター • サービス • タスク定義 • タスクセット	• "ecs:capacity-provider" • "ecs:cluster" • "ecs:service" • "ecs:task-definition" • "ecs:task-set"
Amazon Elastic File System	• すべて • ファイルシステム	• "elasticfilesystem:*" • "elasticfilesystem:file-system"
Amazon Elastic Inference	• アクセラレーター	• "elastic-inference:elastic-inference-accelerator"
Amazon Elastic Kubernetes Service	• クラスター	• "eks:cluster"
Amazon Elastic Search	• [ドメイン]	• "es:domain"
Amazon EMR	• クラスター • [Editor] (エディタ)	• "elasticmapreduce:cluster" • "elasticmapreduce:editor"
Amazon EMR Serverless	• アプリケーション	• "emr-serverless:applications"
AWS エンティティ解決	• マッチングワークフロー • スキーママッピング	• "entityresolution:matchingworkflow" • "entityresolution:schemamapping"
Amazon ElastiCache	• クラスター	• "elasticache:cluster"

サービス名	リソースタイプ	JSON 構文
Amazon EventBridge	• すべて • イベントバス • ルール	• "events:*" • "events:event-bus" • "events:rule"
Amazon EventBridge Pipes	• パイプ	• "pipes:pipe"
Amazon ス EventBridge ケジューラ	• スケジュールグループ	• "scheduler:schedule-group"
Amazon Fraud Detector	• ディテクター • ディテクターバージョン • モデル • ルール • 変数	• "frauddetector:detector" • "frauddetector:detector-version" • "frauddetector:model" • "frauddetector:rule" • "frauddetector:variable"
Amazon Global Accelerator	• アクセラレーター	• "globalaccelerator:accelerator"
Elastic Load Balancing	• すべて • Listener • リスナールール • ロードバランサー • ターゲットグループ	• "elasticloadbalancing:*" • "elasticloadbalancing:listener" • "elasticloadbalancing:listener-rule" • "elasticloadbalancing:loadbalancer" • "elasticloadbalancing:targetgroup"
Amazon FSx	• すべて • バックアップ • ファイルシステム	• "fsx:*" • "fsx:backup" • "fsx:file-system"

サービス名	リソースタイプ	JSON 構文
Amazon GuardDuty	• ディテクター • フィルター • IP セット • 脅威インテリジェンスセット	• "guardduty:detector" • "guardduty:detector/filter" • "guardduty:detector/ipset" • "guardduty:detector/threatintelset"
AWS HealthLake	• データストア	• "healthlake:datastore "
AWS HealthOmics	• アノテーションストア • アノテーションストアのバージョン • リファレンスストア • リファレンス • 実行 • 実行グループ • シーケンスストア • リードセット • バリアントストア • ワークフロー	• "omics:annotationStore" • "omics:annotationStore/version" • "omics:referenceStore" • "omics:referenceStore/reference" • "omics:run" • "omics:runGroup" • "omics:sequenceStore" • "omics:sequenceStore/readSet" • "omics:variantStore" • "omics:workflow"
Amazon Inspector	• フィルター	• "inspector2:filter "
AWS Identity and Access Management	• インスタンスプロファイル • MFA • OIDC プロバイダー • ポリシー • SAML プロバイダー • サーバー証明書	• "iam:instance-profile" • "iam:mfa" • "iam:oidc-provider" • "iam:policy" • "iam:saml-provider" • "iam:server-certificate"

サービス名	リソースタイプ	JSON 構文
AWS IoT Analytics	• すべて • Channel • データセット • データストア • パイプライン	• "iotanalytics:*" • "iotanalytics:channel" • "iotanalytics:dataset" • "iotanalytics:datastore" • "iotanalytics:pipeline"
AWS IoT Events	• すべて • 検出器モデル • 入力	• "iotevents:*" • "iotevents:detectorModel" • "iotevents:input"
AWS IoT Fleet Hub	• アプリケーション	• "iotfleethub:application"
AWS IoT SiteWise	• [アセット] • アセットモデル	• "iotsitewise:asset" • "iotsitewise:asset-model "
AWS IoT Greengrass	• 一括デプロイ • コネクタの定義 • コア定義 • デバイスの定義 • 関数の定義 • ロガー定義 • リソースの定義 • サブスクリプションの定義	• "greengrass:bulk" • "greengrass:connectorsDefinition" • "greengrass:coresDefinition" • "greengrass:devicesDefinition" • "greengrass:functionsDefinition" • "greengrass:loggersDefinition" • "greengrass:resourcesDefinition" • "greengrass:subscriptionsDefinition"
AWS Key Management Service	• すべて • キー	• "kms:*" • "kms:key"

サービス名	リソースタイプ	JSON 構文
Amazon Kinesis	• すべて • アプリケーション	• "kinesisanalytics:*" • "kinesisanalytics:application"
Amazon Data Firehose	• すべて • 配信ストリーム	• "firehose:*" • "firehose:deliverystream"
AWS Lambda	• すべて • 機能	• "lambda:*" • "lambda:function"
Amazon Macie	• カスタムデータ識別子	• "macie2:custom-data-identifier"
Amazon MediaStore	• コンテナ	• "mediastore:container"
Amazon MQ	• ブローカー • 構成	• "mq:broker" • "mq:configuration"
Amazon Network Firewall	• ファイアウォール • ファイアウォールポリシー • ステートフルルールグループ • ステートレスルールグループ	• "network-firewall:firewall" • "network-firewall:firewall-policy" • "network-firewall:stateful-rulegroup" • "network-firewall:stateless-rulegroup"
Amazon OpenSearch Serverless	• 収集	• "aoss:collection"
AWS Organizations	• アカウント • 部門名 • ポリシー • ルート	• "organizations:account" • "organizations:ou" • "organizations:policy" • "organizations:root"

サービス名	リソースタイプ	JSON 構文
Amazon Pinpoint SMS Voice V2	• 設定セット • オプトアウトリスト • 電話番号 • プール • 送信者 ID	• "sms-voice:configuration-set" • "sms-voice:opt-out-list" • "sms-voice:phone-number" • "sms-voice:pool" • "sms-voice:sender-id"
Amazon RDS	• クラスターパラメータグループ • クラスターエンドポイント • イベントサブスクリプション • DB オプショングループ • DB パラメータグループ • DB プロキシ • RDS プロキシエンドポイント • リザーブド DB インスタンス • DB セキュリティグループ • DB サブネットグループ • ターゲットグループ	• "rds:cluster-pg" • "rds:cluster-endpoint" • "rds:es" • "rds:og" • "rds:pg" • "rds:db-proxy" • "rds:db-proxy-endpoint" • "rds:ri" • "rds:secgrp" • "rds:subgrp" • "rds:target-group"

サービス名	リソースタイプ	JSON 構文
Amazon Redshift	• すべて • クラスター • DB グループ • DB name • DB ユーザー • イベントサブスクリプション • HSM クライアント証明書 • HSM の設定 • パラメータグループ • Snapshot • スナップショットコピー権限 • スナップショットスケジュール • サブネットグループ	• `"redshift:*"` • `"redshift:cluster"` • `"redshift:dbgroup"` • `"redshift:dbname"` • `"redshift:dbuser"` • `"redshift:eventsubscription"` • `"redshift:hsmclientcertificate"` • `"redshift:hsmconfiguration"` • `"redshift:parametergroup"` • `"redshift:snapshot"` • `"redshift:snapshotcopygrant"` • `"redshift:snapshotschedule"` • `"redshift:subnetgroup"`
Amazon Redshift Serverless	• 名前空間 • Workgroup	• `"redshift-serverless:namespace"` • `"redshift-serverless:workgroup"`
AWS Resource Access Manager	• すべて • リソースの共有	• `"ram:*"` • `"ram:resource-share"`
AWS Resource Groups	• すべて • グループ	• `"resource-groups:*"` • `"resource-groups:group"`
Amazon Route 53	• ホストゾーン	• `"route53:hostedzone"`

サービス名	リソースタイプ	JSON 構文
Amazon Route 53 Resolver	• すべて • リゾルバーエンドポイント • リゾルバールール	• "route53resolver:*" • "route53resolver:resolver-endpoint" • "route53resolver:resolver-rule"
Amazon S3	• バケット • Storage Lens • Storage Lens グループ	• "s3:bucket" • "s3:storage-lens" • "s3:storage-lens-group"
Amazon SageMaker	• アプリケーションイメージ構成 • アーティファクト • Context • トレーニングジョブ • ジョブの処理中 • パッケージグループのモデル • ヒューマンタスク UI • モデルパッケージ • アクション • パイプライン • 実験 • フロー定義 • プロジェクト	• "sagemaker:app-image-config" • "sagemaker:artifact" • "sagemaker:context" • "sagemaker:training-job" • "sagemaker:processing-job " • "sagemaker:model-package-group" • "sagemaker:human-task-ui" • "sagemaker:model-package" • "sagemaker:action" • "sagemaker:pipeline" • "sagemaker:experiment" • "sagemaker:flow-definition" • "sagemaker:project"
AWS Secrets Manager	• すべて • シークレット	• "secretsmanager:*" • "secretsmanager:secret"

サービス名	リソースタイプ	JSON 構文
AWS Security Lake	• データレイク • サブスクライバー	• "securitylake:data-lake" • "securitylake:subscriber"
AWS Service Catalog	• アプリケーション • 属性グループ • ポートフォリオ • 製品	• "servicecatalog:applications" • "servicecatalog:attribute-groups " • "catalog:portfolio " • "catalog:product "
「Amazon Simple Notification Service (SNS)」	• トピック	• "sns:topic"
Amazon Simple Queue Service (SQS)	• キュー	• "sqs:queue"
Amazon ステートメント言語	• すべて • アクティビティ • [State Machine] (ステートマシン)	• "states:*" • "states:activity " • "states:stateMachine "
AWS Step Functions	• アクティビティ	• "states:activity"
AWS Storage Gateway	• すべて • ゲートウェイ • 共有 • テープ • ボリューム	• "storagegateway:*" • "storagegateway:gateway" • "storagegateway:share" • "storagegateway:tape" • "storagegateway:gateway/volume"

サービス名	リソースタイプ	JSON 構文
AWS Systems Manager	• 関連付け • 自動化の実行 • ドキュメント • メンテナンスウィンドウ • マネージドインスタンス • Ops item • パッチベースライン • セッション • 問い合わせ	• "ssm:association" • "ssm:automation-execution" • "ssm:document" • "ssm:maintenancewindow" • "ssm:managed-instance" • "ssm:opsitem" • "ssm:patchbaseline" • "ssm:session" • "ssm-contacts:contact"
Amazon Textract	• アダプター • バージョン	• "textract:adapters" • "textract:adapters/versions"
AWS Transfer Family	• [サーバー] • ユーザー • ワークフロー	• "transfer:server" • "transfer:user" • "transfer:workflow"
Amazon Well-Architected	• ワークロード	• "wellarchitected:workload"
AWS Wickr	• ネットワーク	• "wickr:network"

サービス名	リソースタイプ	JSON 構文
Amazon WorkSpaces	• すべて • 接続エイリアス • ディレクトリ • WorkSpace • WorkSpaces バンドル • WorkSpaces イメージ • WorkSpaces IP グループ	• "workspaces:*" • "workspaces:connectionalias" • "workspaces:directory" • "workspaces:workspace" • "workspaces:workspacebundle" • "workspaces:workspaceimage" • "workspaces:workspaceipgroup"
Amazon WorkLink	• フリート	• "worklink:fleet"

タグポリシーの構文と例

このページでは、タグポリシーの構文について説明し、例を示します。

タグポリシー構文

タグポリシーは、JSON のルールに従って構造化されたプレーンテキストファイルです。タグポリシーの構文は、すべての管理ポリシータイプの構文に従います。この構文の詳しい説明については、「管理ポリシーの継承を理解する」を参照してください。このトピックは、一般的な構文をタグアップポリシータイプの特定の要件に適用することに焦点を当てています。

次のタグポリシーは、基本的なタグポリシーの構文を示しています。

```
{
    "tags": {
        "costcenter": {
            "tag_key": {
                "@@assign": "CostCenter"
            },
            "tag_value": {
                "@@assign": [
                    "100",
                    "200"
```

```
                ]
            },
            "enforced_for": {
                "@@assign": [
                    "secretsmanager:*"
                ]
            }
        }
    }
}
```

タグポリシー構文には、次の要素が含まれます。

- tags フィールドキーの名前。タグポリシーは、常にこの固定キー名で始まります。上記のポリシーの例では一番上の行です。
- ポリシーステートメントを一意に識別するポリシーキー。大文字小文字の処理を除き、タグキーの値と一致する必要があります。タグキー (次に説明します) とは異なり、ポリシー値では大文字と小文字が区別されません。

  この例では、costcenter がポリシーキーです。
- リソースを準拠させる大文字小文字表記を持つ許容タグキーを指定するタグキーを 1 つ以上指定します。大文字小文字の処理が定義されていない場合、タグキーのデフォルトは小文字です。タグキーの値は、ポリシーキーの値と一致する必要があります。ただし、ポリシーのキーバリューでは大文字と小文字が区別されないため、大文字と小文字が異なる可能性があります。

  この例では、CostCenter がタグキーです。これは、タグポリシーに準拠するために必要な大文字と小文字の処理です。このタグキーに対して大文字と小文字の処理が異なるリソースは、タグポリシーに準拠していません。

  タグポリシーでは、複数のタグキーを定義できます。
- (オプション) タグキーに対して受け入れ可能な 1 つ以上のタグ値のリスト。タグポリシーがタグキーのタグ値を指定しない場合、すべての値 (値なしの場合を含む) が準拠していると見なされます。

  この例では、CostCenter タグキーの許容値は 100 と 200 です。
- (オプション) 指定されたサービスおよびリソースに対する非準拠のタグ付け操作を防止するかどうかを示す enforced_for オプション。コンソールでは、タグポリシーを作成するためのビジュアルエディタの [Prevent noncompliant operations for this tag (このタグに対する非準拠操作を防止する)] オプションです。このオプションのデフォルト設定は null です。

タグポリシーの例では、すべての AWS Secrets Manager リソースに渡されるCostCenterタグがこのポリシーに準拠している必要があることを指定します。

> **Warning**
>
> タグポリシーの使用経験がある場合にのみ、このオプションをデフォルトから変更してください。そうしないと、組織のアカウントのユーザーが必要なリソースを作成できなくなる可能性があります。

- タグポリシーを組織ツリーの他のタグポリシーとマージして、アカウントの有効なタグポリシーを作成する方法を指定する演算子。この例では、@@assign を使用して、tag_key、tag_value、および enforced_for に文字列を割り当てます。演算子の詳細については、「継承演算子」を参照してください。
- - タグ値と [*] フィールドには、ワイルドカード enforced_forを使用できます。
  - タグ値ごとに、ワイルドカードを 1 つのみ使用できます。例えば、*@example.com は許可されますが、*@*.com は許可されません。
  - enforced_for では、一部のサービスで <service>:* を使用して、該当するサービスのすべてのリソースに対して強制を適用できます。enforced_for がサポートするサービスとリソースタイプのリストについては、「強制をサポートするサービスとリソースタイプ」を参照してください。

    ワイルドカードを使用してすべてのサービスを指定したり、すべてのサービスのリソースを指定したりすることはできません。

タグポリシーの例

次のタグポリシーの例は、情報提供のみを目的としています。

> **Note**
>
> 組織でこれらのタグポリシーの例を使用する前に、次の点に注意してください。
>
> - タグポリシーの使用を開始するための推奨ワークフローを実行したことを確認します。
> - 固有の要件に合わせて、これらのタグポリシーを慎重に確認し、カスタマイズする必要があります。
> - タグポリシーのすべての文字には、上限サイズが適用されます。このガイドの例では、読みやすさを向上させるため、空白文字を追加してフォーマットされたタグポリシーを示し

ています。ただし、ポリシーのサイズが上限サイズに近づいたら、スペースを節約するために空白を削除できます。空白の例としては、空白文字や引用符の外側の改行などがあります。

- 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。

例 1: 組織全体のタグキーの大文字小文字取り扱いの定義

次の例は、組織内のアカウントに標準化させる 2 つのタグキーと大文字小文字のみを定義するタグポリシーを示しています。

ポリシー A - 組織ルートのタグポリシー

```
{
    "tags": {
        "CostCenter": {
            "tag_key": {
                "@@assign": "CostCenter",
                "@@operators_allowed_for_child_policies": ["@@none"]
            }
        },
        "Project": {
            "tag_key": {
                "@@assign": "Project",
                "@@operators_allowed_for_child_policies": ["@@none"]
            }
        }
    }
}
```

このタグポリシーは、CostCenter と Project という 2 つのタグキーを定義します。このタグポリシーを組織のルートにアタッチすると、次の効果があります。

- 組織内のすべてのアカウントは、このタグポリシーを継承します。
- 組織内のすべてのアカウントは、準拠のために、定義された大文字と小文字の処理を使用する必要があります。CostCenter タグと Project とタグの付いたリソースは準拠しています。タグキーに対して大文字小文字の処理が異なるリソース (costcenter、Costcenter、COSTCENTER など) は準拠していません。

- `@@operators_allowed_for_child_policies": ["@@none"]` 行はタグキーをロックダウンします。組織ツリーの下位にアタッチされたタグポリシー (子ポリシー) は、大文字と小文字の処理を含め、値設定演算子を使用してタグキーを変更することはできません。
- すべてのタグポリシーと同じように、タグなしリソースまたはタグポリシーで定義されていないタグは、タグポリシーに準拠しているかどうか評価されません。

AWS では、使用するタグキーの同様のタグポリシーを作成する際のガイドとして、この例を使用することをお勧めします。組織のルートにアタッチします。次に、下の例のようなタグポリシーを作成します。この例では、定義されたタグキーの許容値のみを定義します。

次のステップ: 値の定義

組織ルートに先ほどのタグポリシーをアタッチしたと仮定します。次に、次のようなタグポリシーを作成し、アカウントにアタッチできます。このポリシーは、`CostCenter` および `Project` タグキーの許容値を定義します。

ポリシー B - アカウントのタグポリシー

```
{
    "tags": {
        "CostCenter": {
            "tag_value": {
                "@@assign": [
                    "Production",
                    "Test"
                ]
            }
        },
        "Project": {
            "tag_value": {
                "@@assign": [
                    "A",
                    "B"
                ]
            }
        }
    }
}
```

ポリシー A を組織ルートにアタッチし、ポリシー B をアカウントにアタッチすると、ポリシーが結合され、次の有効なタグポリシーがアカウントに対して作成されます。

ポリシー A + ポリシー B = アカウントの有効なタグポリシー

```
{
    "tags": {
        "Project": {
            "tag_value": [
                "A",
                "B"
            ],
            "tag_key": "Project"
        },
        "CostCenter": {
            "tag_value": [
                "Production",
                "Test"
            ],
            "tag_key": "CostCenter"
        }
    }
}
```

継承演算子の動作例や有効なタグポリシーの例など、ポリシー継承の詳細については、「管理ポリシーの継承を理解する」を参照してください。

例 2: タグキーの使用を禁止する

タグキーの使用を禁止するには、次のようなタグポリシーを組織エンティティにアタッチします。

このサンプルポリシーは、Color タグキーに対して使用できる値がないことを指定します。また、子タグポリシーで演算子を使用できないことも指定します。したがって、影響を受けるアカウントのリソース上の Color タグは、非準拠と見なされます。ただし enforced_for オプションは、実際には、影響を受けるアカウントが Color タグを使用して Amazon DynamoDB テーブルのみにタグ付けできないようにします。

```
{
    "tags": {
        "Color": {
            "tag_key": {
                "@@operators_allowed_for_child_policies": [
                    "@@none"
                ],
                "@@assign": "Color"
            },
```

```
            "tag_value": {
                "@@operators_allowed_for_child_policies": [
                    "@@none"
                ],
                "@@assign": []
            },
            "enforced_for": {
                "@@assign": [
                    "dynamodb:table"
                ]
            }
        }
    }
}
```

サポートされるリージョン

タグポリシー機能は、次のリージョンで使用できます。

リージョン名	リージョンパラメータ
米国東部 (バージニア北部) リージョン[1]	**us-east-1**
米国東部 (オハイオ) リージョン	us-east-2
米国西部 (北カリフォルニア) リージョン	us-west-1
米国西部 (オレゴン) リージョン	us-west-2
アフリカ (ケープタウン) リージョン[2]	af-south-1
アジアパシフィック (香港) リージョン[2]	ap-east-1
アジアパシフィック (ムンバイ) リージョン	ap-south-1
アジアパシフィック (ハイデラバード)2	ap-south-2
アジアパシフィック (東京) リージョン	ap-northeast-1
アジアパシフィック (ソウル) リージョン	ap-northeast-2
アジアパシフィック (大阪) リージョン	ap-northeast-3

リージョン名	リージョンパラメータ
アジアパシフィック (シンガポール) リージョン	ap-southeast-1
アジアパシフィック (シドニー) リージョン	ap-southeast-2
アジアパシフィック (ジャカルタ) リージョン2	ap-southeast-3
アジアパシフィック (メルボルン)2	ap-southeast-4
カナダ西部 (カルガリー)2	ca-west-1
カナダ (中部) リージョン	ca-central-1
欧州 (フランクフルト) リージョン	eu-central-1
欧州 (チューリッヒ) リージョン2	eu-central-2
欧州 (ミラノ) リージョン[2]	eu-south-1
欧州 (スペイン)2	eu-south-2
欧州 (アイルランド) リージョン	eu-west-1
欧州 (ロンドン) リージョン	eu-west-2
欧州 (パリ) リージョン	eu-west-3
欧州 (ストックホルム) リージョン	eu-north-1
中東 (アラブ首長国連邦) リージョン2	me-central-1
中東 (バーレーン) リージョン[2]	me-south-1
南米 (サンパウロ) リージョン	sa-east-1
イスラエル (テルアビブ)2	il-central-1

[1]次の Organizations オペレーションを呼び出す場合は、**us-east-1** リージョンを指定する必要があります。

- DeletePolicy
- DisablePolicyType
- EnablePolicyType
- など、組織ルートに対するその他のオペレーションListRoots。

タグポリシー機能の一部である次のリソースグループタグ付け API 操作を呼び出す場合も、**us-east-1** リージョンを指定する必要があります。

- DescribeReportCreation
- GetComplianceSummary
- StartReportCreation

> **Note**
>
> 組織全体でのタグポリシーへの準拠を評価するには、レポートを保存するため、米国東部 (バージニア北部) リージョンの Amazon S3 バケットにもアクセスできる必要があります。詳細については、「 リソースのタグ付けユーザーガイド」の「レポートストレージ用の Amazon S3 バケットポリシー」を参照してください。 AWS

[2]これらのリージョンは、手動で有効にする必要があります。の有効化と無効化の詳細については AWS リージョン、「 アカウント管理リファレンスガイド」の AWS リージョン 「アカウントで使用できる を指定する」を参照してください。 AWS Resource Groups コンソールは、これらのリージョンでは利用できません。

サービスコントロールポリシー (SCP)

サービスコントロールポリシー (SCP) は、組織のアクセス許可の管理に使用できる組織ポリシーの一種です。SCP を使用すると、組織内の IAM ユーザーと IAM ロールが使用できる最大権限を一元管理できます。SCP は、アカウントが組織のアクセスコントロールガイドラインに従っていることを確認するのに役立ちます。SCP は、すべての機能が有効になっている 組織でのみ使用できます。組織が一括請求機能のみを有効にしている場合、SCP は使用できません。SCP を有効にする方法については、「ポリシータイプの有効化と無効化」を参照してください。

SCP は組織内の IAM ユーザーと IAM ロールにアクセス権限を付与しません。SCP によってアクセス許可を付与することはできません。SCP は、組織内の IAM ユーザーと IAM ロールが実行できるア

クションについて、権限ガードレールを定義したり、制限を設定したりします。アクセス権限を付与するには、管理者は IAM ユーザーと IAM ロールにアタッチされる ID ベースのポリシーや、アカウント内のリソースにアタッチされるリソースベースのポリシーなど、アクセスを制御するポリシーをアタッチする必要があります。有効なアクセス権限とは、SCP で許可されているものと、ID やリソースベースのポリシーで許可されているものを論理的に組み合わせたものです。

> **Important**
>
> SCP は、管理アカウントのユーザーやロールには影響を与えません。SCP は、組織内のメンバーアカウントにのみ影響を与えます。

このページのトピック

SCP の効果をテストする

AWS ポリシーがアカウントに及ぼす影響を徹底的にテストしない限り、SCP を組織の根幹に結び付けないことを強くお勧めします。代わりに、お客様のアカウントを一度に 1 つずつ、または少なくとも少人数ずつ移動できる OU を作成し、誤って主要なサービスからユーザーを締め出すことのないようにします。アカウントでサービスが使用されているかどうかを判断する方法の 1 つは、IAM のサービスの最終アクセス時間データを調べることです。もう 1 つの方法は、AWS CloudTrail を使用して API レベルでサービスの使用状況を記録することです。

> **Note**
>
> 変更するか、AWSAccess許可されたアクションを含む別のポリシーに置き換えない限り、フルポリシーを削除しないでください。そうしないと、 AWS メンバーアカウントからのすべてのアクションが失敗します。

SCP の上限サイズ

SCP 内のすべての文字は、その上限サイズに対してカウントされます。このガイドの例では、読みやすさを向上させるため、空白文字を追加してフォーマットされた SCP を示します。ただし、ポリシーサイズが上限サイズに近づいている場合は、スペースを節約するために、引用符の外側にあるすべての空白文字 (スペースや改行など) を削除できます。

> **Tip**
>
> ビジュアルエディタを使用して SCP を構築します。これによって、よけいな空白が自動的に削除されます。

SCP を組織内のさまざまなレベルにアタッチする

SCP の動作の詳細な説明については、「SCP 評価」を参照してください。

アクセス許可における SCP 効果

SCP は AWS Identity and Access Management (IAM) アクセス権限ポリシーに似ており、構文もほぼ同じです。ただし、SCP がアクセス権限を付与することはありません。代わりに、SCP は組織内の IAM ユーザーと IAM ロールの最大権限を指定する JSON ポリシーです。詳細については、IAM ユーザーガイドのポリシーの評価論理を参照してください。

- SCP は、組織の一部であるアカウントが管理する IAM ユーザーとロールのみに影響します。SCP はリソースベースのポリシーには直接影響しません。また、組織外のアカウントに属するユーザーやロールにも影響しません。組織内のアカウント A が所有する Amazon S3 バケットについて考えてみます。このバケットポリシー (リソースベースのポリシー) は、組織外のアカウント B に属するユーザーにアクセスを許可します。アカウント A には SCP がアタッチされています。この SCP はアカウント B の外部ユーザーには適用されません。SCP は組織内のアカウント A が管理するユーザーにのみ適用されます。

- SCP は、メンバーアカウントのルートユーザーを含む、メンバーアカウントの IAM ユーザーとロールのアクセス許可を制限します。すべてのアカウントには、その上位のすべての親で許可されている権限のみがあります。アクセス許可が、暗黙的に (`Allow` ポリシーステートメントに含まれない)、または明示的に (`Deny` ポリシーステートメントに含まれる)、アカウントのレベルでブロックされている場合、影響を受けるアカウントのユーザーまたはロールは、アカウント管理者が `*/*` アクセス許可を持つ `AdministratorAccess` IAM ポリシーをユーザーにアタッチしても、そのアクセス許可を使用することはできません。
- SCP は、組織内のメンバーアカウントのみに影響します。管理アカウントのユーザーやロールには影響しません。
- ユーザーとロールには、適切な IAM アクセス許可ポリシーを使用してアクセス許可を付与する必要があります。IAM アクセス許可ポリシーがアタッチされていないユーザーは、たとえ適用される SCP によりすべてのサービスとアクションが許可されても、いずれのサービスもアクセスも許可されません。
- アクションへのアクセスを付与する IAM アクセス許可ポリシーがユーザーまたはロールに付与されており、そのアクションが適用可能な SCP によって許可されている場合、ユーザーまたはロールはそのアクションを実行できます。
- アクションへのアクセスを許可する IAM アクセス許可ポリシーがユーザーまたはロールに付与されており、そのアクションが適用可能な SCP によって許可されていないか、または明示的に拒否されている場合、ユーザーまたはロールがそのアクションを実行することはできません。
- SCP は、アタッチされたアカウントのすべてのユーザーやロール (ルートユーザーを含む) に影響します。唯一の例外は、「SCP によって制限されないタスクおよびエンティティ」で説明されているものです。
- SCP はサービスにリンクされたロールに影響しません。サービスにリンクされたロールにより、AWS 他のサービスを SCP AWS Organizations と統合できるようになり、SCP によって制限されることはありません。
- ルートの SCP ポリシータイプを無効にすると、すべての SCP はそのルートのすべてのエンティティから自動的に切り離されます。 AWS Organizations AWS Organizations エンティティには、組織単位、組織、アカウントが含まれます。そのルートの SCP を再度有効にすると、そのルートはすべてのエンティティに自動的にアタッチされたデフォルトの `FullAWSAccess` ポリシーに戻ります。SCP の無効化の前に AWS Organizations にアタッチされていたすべての SCP は失われ、自動的には復旧されません。ただし、手動で再度アタッチできます。
- アクセス許可の境界 (高度な IAM 機能) と SCP が両方存在する場合は、アクセス許可の境界、SCP、およびアイデンティティのポリシーによって、すべてのアクションが許可されます。

アクセスデータを使用して SCP を改善する

管理アカウントの認証情報を使用してサインインすると、IAM AWS OrganizationsAWS Organizationsコンソールのセクションでエンティティまたはポリシーのサービスの最終アクセス時間データを表示できます。IAM の AWS Command Line Interface (AWS CLI) または AWS API を使用して、サービスの最終アクセス日データを取得することもできます。このデータには、 AWS Organizations アカウント内の IAM ユーザーとロールが最後にアクセスを試みた許可されたサービスと、いつアクセスしようとしたかに関する情報が含まれます。この情報を使用して不要なアクセス許可を識別し、最小権限の原則により良く準拠するように SCP を改良できます。

たとえば、3 つのサービスへのアクセスを禁止する拒否リスト SCP があるとします。 AWS SCP の Deny ステートメントにリストされていないすべてのサービスが許可されます。IAM のサービスの最終アクセス日データから、SCP AWS で許可されているが使用されていないサービスがわかります。この情報により、SCP を更新して、必要でないサービスへのアクセスを拒否できます。

詳細については、IAM ユーザーガイドにある下記のトピックを参照してください。

- 組織の組織サービスの最終アクセス データの表示
- データを使用した組織単位のアクセス権限の調整

SCP によって制限されないタスクおよびエンティティ

SCP を使用して次のタスクを制限することはできません。

- 管理アカウントによって実行されるすべてのアクション
- サービスにリンクされたロールにアタッチされたアクセス許可を使用して実行されるすべてのアクション。
- root ユーザーとして Enterprise サポートプランに登録する
- root AWS ユーザーとしてサポートレベルを変更します。
- CloudFront プライベートコンテンツに信頼された署名者機能を提供する。
- Amazon Lightsail メールサーバーおよび Amazon EC2 インスタンスの逆引き DNS をルートユーザーとして設定する
- AWS一部の関連サービスのタスク:
  - Alexa Top Sites
  - Alexa Web Information Service

- Amazon Mechanical Turk
- Amazon Product Marketing API

サービスコントロールポリシーの作成、更新、削除

組織の管理アカウントにサインインすると、サービスコントロールポリシー (SCP) を作成および更新することができます。指定するサービスおよびアクションへのアクセスを拒否または許可するステートメントを構築して、SCP を作成します。

SCP を使用するためのデフォルトの設定では、「ブロックリスト」戦略を使用します。この戦略では、アクセスを拒否するステートメントを作成することで、ブロックするアクションを除くすべてのアクションが暗黙的に許可されます。deny ステートメントでは、ステートメントのリソースと条件を指定し、 NotAction要素を使用できます。許可ステートメントでは、サービスおよびアクションのみを指定できます。アクセスを拒否するステートメントとアクセスを許可するステートメントの詳細については、「SCP 評価」を参照してください。

> **Tip**
>
> IAM のサービスの最終アクセス時間データを、SCP を更新するためのデータポイントとして使用し、必要な AWS のサービスのみへのアクセスを制限できます。詳細については、IAM ユーザーガイドの「組織の Organizations サービスの最終アクセス時間データを表示する」を参照してください。

このトピックの内容

- 組織のサービスコントロールポリシーを有効にしたあと、ポリシーを作成することができます。
- SCP の要件が変更された場合は、既存のポリシーを更新できます。
- ポリシーが不要になった場合、すべての組織単位 (OU) およびアカウントからポリシーをデタッチした後、ポリシーを削除できます。

SCP を作成する

> **最小アクセス許可**
>
> SCP を作成するには、以下のアクションを実行する権限が必要です。

- `organizations:CreatePolicy`

AWS Management Console

サービスコントロールポリシーを作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、[Create policy] (ポリシーの作成) を選択します。
3. [Create new service control policy] (新しいサービスコントロールポリシーの作成) ページで、[Policy name] (ポリシー名) とオプションの [Policy description] (ポリシーの説明) に入力します。
4. (オプション) [Add tag] (タグの追加) を選択してキーとオプションの値を入力し、1 つ以上のタグを追加します。値を空白のままにすると、空の文字列が設定され、`null` にはなりません。1 つのポリシーに最大 50 個のタグをアタッチできます。詳細については、「AWS Organizations リソースのタグ付け」を参照してください。

   > **Note**
   >
   > この後のほとんどのステップでは、JSON エディタの右側にあるコントロールを使用して、要素ごとにポリシーを構築する方法について説明します。また、ウィンドウの左側にある JSON エディタには、いつでもテキストを入力することもできます。直接入力することも、コピーアンドペーストを使用することもできます。

5. ポリシーを構築するための次のステップは、アクセスを拒否または許可するステートメントを追加するかどうかに応じて異なります。詳細については、「SCP 評価」を参照してください。`Deny` ステートメントを使用する場合、特定のリソースへのアクセスを制限し、SCPs が有効になるタイミングの条件を定義し、 NotAction要素を使用できるため、追加の制御が可能です。構文の詳細については、「SCP 構文」を参照してください。

   アクセスを拒否するステートメントを追加するには

   a. エディタの右側の「ステートメントの編集」ペインの「アクションの追加」で、 AWS サービスを選択します。

右側のオプションを選択すると、JSON エディタが更新され、左側に対応する JSON ポリシーが表示されます。

b. サービスを選択すると、そのサービスで使用可能なアクションが記載されたリストが開きます。[All actions] (すべてのアクション) または、拒否する 1 つ以上の個別のアクションを選択します。

左側の JSON が更新され、選択したアクションが表示されます。

> **Note**
>
> 個別のアクションを選択したら、戻って [All actions] (すべてのアクション) を選択すると、予定される *servicename*/* のエントリが JSON に追加されますが、以前に選択した個別のアクションは JSON に残ったまま削除されません。

c. 追加のサービスからアクションを追加したい場合は、[Statement] (ステートメント) ボックスの上部にある [All services] (すべてのサービス) を選択し、必要に応じて前の 2 つのステップを繰り返します。

d. ステートメントに含めるリソースを指定します。

- [リソースの追加] の横にある [追加] を選択します。
- [Add a resource] (リソースの追加) ダイアログで、リソースを制御するサービスをリストから選択します。前のステップで選択したサービスからのみ選択できます。
- [Resource type] (リソースタイプ) で、制御するリソースのタイプを選択します。
- 最後に、[Resource ARN] (リソース ARN) で Amazon リソースネーム (ARN) を入力し、アクセスをコントロールするリソースを特定します。中括弧 {} で囲まれたすべてのプレースホルダーを置き換える必要があります。そのリソースタイプの ARN 構文で許可されているワイルドカード (*) を指定できます。ワイルドカードを使用できる場所については、特定のリソースタイプに関するドキュメントを参照してください。
- [Add a resource] (リソースの追加) を選択して、ポリシーへの追加を保存します。JSON の `Resource` 要素に、追加や変更が反映されます。[Resource] (リソース) 要素が必要です。

> **Tip**
>
> 選択したサービスのすべてのリソースを指定する場合は、リストの [All resource] (すべてのリソース) のオプションを選択するか、JSON で `Resource` ステートメントを直接編集して `"Resource":"*"` を読み取ります。

e. (オプション) ポリシーステートメントが有効なときに制限する条件を指定するには、[条件を追加] の横にある [追加] を選択します。

- 条件キー – リストから、すべての AWS サービスで使用できる任意の条件キー (例: `aws:SourceIp`) 、またはこのステートメントで選択した 1 つのサービスのみのサービス固有のキーを選択できます。
- 限定条件 - (オプション) 条件に複数の値を入力する場合 (指定した条件キーに応じて)、値に対するリクエストをテストする限定条件を指定できます。
  - デフォルト値 — ポリシーの条件キーバリューに対する、リクエスト内の単一の値をテストします。リクエスト値がポリシーの値と一致する場合、条件は true を返します。ポリシーで複数の値を指定した場合、それらは「or」のテストとして扱われ、リクエスト値がポリシーの値のいずれかに一致すると、条件は true を返します。
  - リクエスト内の任意の値 — リクエストに複数の値を含めることができる場合、このオプションでは、リクエスト値の少なくとも 1 つが、ポリシーの少なくとも 1 つの条件キーバリューと一致するかどうかをテストします。リクエスト内のキーバリューのいずれかがポリシーの条件値のいずれかと一致する場合に true が返されます。一致するキーまたは空のデータセットがない場合、条件は false を返します。
  - リクエスト内のすべての値 - リクエストに複数の値を含めることができる場合、このオプションは、すべてのリクエスト値がポリシーの条件キーバリューと一致するかどうかをテストします。リクエストのすべてのキーバリューがポリシーの 1 つ以上の値と一致する場合、条件は true を返します。また、リクエストにキーがない場合、またはキーバリューが空の文字列などの null データセットに解決される場合は true を返します。
- 演算子 — 演算子は、比較するタイプを指定します。表示されるオプションは、条件キーのデータ型によって異なります。例えば、`aws:CurrentTime` グローバル条件キーを使用すると、任意の日付比較演算子または `Null` から選択でき、それを使用してリクエスト内に値が存在するかどうかをテストできます。

Null テスト以外の条件演算子については、 IfExistsオプションを選択できます。

- 値 — (オプション) リクエストをテストする 1 つ以上の値を指定します。

[条件を追加] を選択します。

条件キーの詳細については、IAM ユーザーガイドの「IAM JSON ポリシーの要素: Condition」を参照してください。

f. (オプション) NotAction 要素を使用して、指定したアクションを除くすべてのアクションへのアクセスを拒否するには、左側のペインにある Action を NotAction 要素の直後に表示される "Effect": "Deny", で置き換えます。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素： NotAction」を参照してください。

6. アクセスを許可するステートメントを追加するには

a. 左側の JSON エディタで、行 "Effect": "Deny" を "Effect": "Allow" に変更します。

右側のオプションを選択すると、JSON エディターが更新され、対応する JSON ポリシーが左側に表示されます。

b. サービスを選択すると、そのサービスで使用可能なアクションが記載されたリストが開きます。[All actions] (すべてのアクション) または、許可する 1 つ以上のアクションを個別に選択できます。

左側の JSON が更新され、選択したアクションが表示されます。

> **Note**
>
> 個別のアクションを選択したら、戻って [All actions] (すべてのアクション) を選択すると、予定される *servicename*/* のエントリが JSON に追加されますが、以前に選択した個別のアクションは JSON に残ったまま削除されません。

c. 追加のサービスからアクションを追加したい場合は、[Statement] (ステートメント) ボックスの上部にある [All services] (すべてのサービス) を選択し、必要に応じて前の 2 つのステップを繰り返します。

7. (オプション) ポリシーに別のステートメントを追加するには、[Add statement] (ステートメントを追加) を選択し、ビジュアルエディタを使用して次のステートメントを構築します。

8. ステートメントの追加が終了したら、[ポリシーの作成] を選択して完了した SCP を保存します。

新しい SCP は組織のポリシーのリストに表示されます。SCP をルート、OU、またはアカウントにアタッチできるようになりました。

AWS CLI & AWS SDKs

サービスコントロールポリシーを作成するには

SCP を作成するには、次のいずれかのコマンドを使用します。

- AWS CLI: create-policy

  次の例は、JSON ポリシーのテキストを含む Deny-IAM.json いう名前のファイルがあることを前提としたものです。このファイルを使用して、新しいサービスコントロールポリシーを作成します。

```
$ aws organizations create-policy \
    --content file://Deny-IAM.json \
    --description "Deny all IAM actions" \
    --name DenyIAMSCP \
    --type SERVICE_CONTROL_POLICY
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
service_control_policy/p-i9j8k7l6m5",
            "Name": "DenyIAMSCP",
            "Description": "Deny all IAM actions",
            "Type": "SERVICE_CONTROL_POLICY",
            "AwsManaged": false
        },
         "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"Statement1\",\"Effect\":\"Deny\",\"Action\":[\"iam:*\"],\"Resource\":[\"*\"]}]}"
    }
}
```

- AWS SDKs CreatePolicy

> **Note**
>
> SCP は、管理アカウントやその他のいくつかの状況では有効になりません。詳細については、「SCP によって制限されないタスクおよびエンティティ」を参照してください。

SCP を更新する

組織の管理アカウントにサインインすると、ポリシーの名前または内容を変更することができます。SCP の内容を変更すると、すぐにアタッチされているすべてのアカウントの任意のユーザー、グループ、およびロールに影響します。

> **最小アクセス許可**
>
> SCP を更新するには、次のアクションを実行する権限が必要です。
>
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:UpdatePolicy`。
> - 指定されたポリシー (または "*") の ARN を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:DescribePolicy`。

AWS Management Console

ポリシーを更新するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、更新するポリシーの名前を選択します。
3. ポリシーの詳細ページで、[Edit policy] (ポリシーの編集) を選択します。
4. 次の変更のいずれか、またはすべてを行います。
   - ポリシーの名前を変更するには、[Policy name] (ポリシー名) に新しい名前を入力します。
   - 説明を変更するには、[Policy description] (ポリシーの説明) に新しいテキストを入力します。
   - 左側のペインでポリシーを JSON 形式で編集すると、ポリシーテキストを編集できます。または、右側のエディタでステートメントを選択し、コントロールを使用して要素を変更

することもできます。各コントロールの詳細については、このトピックで前述した SCP の作成手順を参照してください。

5. 完了したら、[変更の保存] を選択します。

AWS CLI & AWS SDKs

ポリシーを更新するには

ポリシーを更新するには、以下のいずれかのコマンドを使用します。

- AWS CLI: update-policy

次の例では、ポリシーの名前を変更します。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --name "MyRenamedPolicy"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
service_control_policy/p-i9j8k7l6m5",
            "Name": "MyRenamedPolicy",
            "Description": "Blocks all IAM actions",
            "Type": "SERVICE_CONTROL_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"Statement1\",\"Effect\":\"Deny\",\"Action\":[\"iam:*\"],\"Resource\":[\"*\"]}]}"
    }
}
```

次の例では、サービスコントロールポリシーの説明を追加または変更します。

```
$ aws organizations update-policy \
    --policy-id p-i9j8k7l6m5 \
    --description "My new policy description"
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
```

```
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
service_control_policy/p-i9j8k7l6m5",
            "Name": "MyRenamedPolicy",
            "Description": "My new policy description",
            "Type": "SERVICE_CONTROL_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"Statement1\",\"Effect\":\"Deny\",\"Action\":[\"iam:*\"],\"Resource\":[\"*\"]}]}"
    }
}
```

次の例では、新しい JSON ポリシーテキストを含むファイルを指定して、SCP のポリシードキュメントを変更します。

```
$ aws organizations update-policy \
    --policy-id p-zlfw1r64
    --content file://MyNewPolicyText.json
{
    "Policy": {
        "PolicySummary": {
            "Id": "p-i9j8k7l6m5",
            "Arn": "arn:aws:organizations::123456789012:policy/o-aa111bb222/
service_control_policy/p-i9j8k7l6m5",
            "Name": "MyRenamedPolicy",
            "Description": "My new policy description",
            "Type": "SERVICE_CONTROL_POLICY",
            "AwsManaged": false
        },
        "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"AModifiedPolicy\",\"Effect\":\"Deny\",\"Action\":[\"iam:*\"],\"Resource\":[\"*
\"]}]}"
    }
}
```

- AWS SDKs UpdatePolicy

詳細情報

SCP の作成の詳細については、以下のトピックを参照してください。

- サービスコントロールポリシーの例

- SCP 構文

SCP にアタッチされたタグを編集する

組織の管理アカウントにサインインすると、SCP にアタッチされるタグを追加または削除できます。タグ付けの詳細については、「AWS Organizations リソースのタグ付け」を参照してください。

> **最小アクセス許可**
>
> AWS 組織内の SCP にアタッチされたタグを編集するには、次のアクセス許可が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:DescribePolicy` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:TagResource`
> - `organizations:UntagResource`

AWS Management Console

SCP にアタッチされたタグを編集するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、編集するタグがアタッチされたポリシーの名前を選択します。
3. ポリシーの詳細ページで、[Tags] (タグ) タブ、[Manage tags] (タグ管理) の順に選択します。
4. 次の変更のいずれか、またはすべてを行います。
   - 古い値を新しい値で上書きして、タグの値を変更します。タグキーを直接変更することはできません。キーを変更するには、古いキーを持つタグを削除してから、新しいキーを持つタグを追加する必要があります。
   - [Remove] (削除) を選択すると、既存のタグが削除されます。

- 新しいタグのキーと値のペアを追加します。[Add tag] (タグの追加) を選択し、表示されたボックスに新しいキー名とオプションの値を入力します。[Value] (値) ボックスを空白のままにすると、値は空の文字列に設定され、null にはなりません。

5. 完了したら、[変更の保存] を選択します。

AWS CLI & AWS SDKs

SCP にアタッチされたタグを編集するには

SCP にアタッチされたタグを編集するには、次のいずれかのコマンドを使用します。

- AWS CLI: tag-resource および untag-resource
- AWS SDKsTagResourceおよび UntagResource

SCP を削除する

組織の管理アカウントにサインインすると、組織に不要になったポリシーを削除できます。

> **メモ**
>
> - ポリシーを削除するには、まずそのポリシーをすべての添付エンティティからデタッチする必要があります。
> - という名前の SCP などの AWS マネージド SCP は削除できませんFullAWSAccess。

> **最小アクセス許可**
>
> SCP を削除するには、次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。
>
> - organizations:DeletePolicy

AWS Management Console

SCP を削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、削除する SCP の名前を選択します。
3. 削除するポリシーは、まず、すべてのルート、OU、アカウントからデタッチする必要があります。[Targets] (ターゲット) タブを選択し、[Targets] (ターゲット) リストの各ルート、OU、アカウントの横にあるラジオボタンをクリックしてから、[Detach] (デタッチ) を選択します。確認ダイアログボックスで、[Detach] (デタッチ) を選択します。すべてのターゲットを削除するまで繰り返します。
4. ページの上部で、[Delete] (削除) を選択します。
5. 確認ダイアログボックスで、ポリシーの名前を入力し、[Delete] (削除) を選択します。

AWS CLI & AWS SDKs

SCP を削除するには

以下のコード例は、`DeletePolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Deletes an existing AWS Organizations policy.
```

```
    /// </summary>
    public class DeletePolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and then uses it to
        /// delete the policy with the specified policyId.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var policyId = "p-00000000";

            var request = new DeletePolicyRequest
            {
                PolicyId = policyId,
            };

            var response = await client.DeletePolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully deleted Policy: {policyId}.");
            }
            else
            {
                Console.WriteLine($"Could not delete Policy: {policyId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeletePolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDeletePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def delete_policy(policy_id, orgs_client):
    """
    Deletes a policy.

    :param policy_id: The ID of the policy to delete.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.delete_policy(PolicyId=policy_id)
        logger.info("Deleted policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_id)
        raise
```

- API の詳細については、 DeletePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

サービスコントロールポリシーのアタッチとデタッチ

組織の管理アカウントにサインインすると、前に作成したサービスコントロールポリシー (SCP) をアタッチできます。SCP は、組織ルート、組織単位 (OU)、または直接アカウントにアタッチすることができます。SCP を作成するには、次の手順を実行します。

最小アクセス許可

SCP をルート、OU、またはアカウントにアタッチするには、次のアクションを実行する権限が必要です。

- 特定のポリシーの "*" または Amazon リソースネーム (ARN) を含む同じポリシーステートメントの `Resource` 要素を持つ `organizations:AttachPolicy`、およびポリシーをアタッチするルート、OU、またはアカウントの ARN。

AWS Management Console

SCP をアタッチするには、ポリシーをアタッチするルート、OU、またはアカウントに移動する必要があります。

ルート、OU、またはアカウントに移動して SCP をアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、SCP をアタッチするルート、OU、またはアカウントの横にあるチェックボックスに移動してオンにします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
3. [Policies] (ポリシー) タブの [Service control policies] (サービスコントロールポリシー) で、[Attach] (アタッチ) を選択します。
4. 目的のポリシーを見つけて [Attach policy] (ポリシーのアタッチ) を選択します。

   [Policies] (ポリシー) タブで、アタッチされている SCP のリストが更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに有効になり、アタッチされたアカウントや、アタッ

チされたルートまたは OU の下のすべてのアカウントの IAM ユーザーとロールのアクセス許可に影響します。

ポリシーに移動して SCP をアタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、アタッチするポリシーの名前を選択します。
3. [Targets] (ターゲット) タブで [Attach] (アタッチ) を選択します。
4. ポリシーをアタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
5. Attach policy] (ポリシーのアタッチ) を選択します。

   [Targets] (ターゲット) タブで、アタッチされている SCP のリストが更新され、新たに追加したものが表示されます。ポリシーの変更はすぐに有効になり、アタッチされたアカウントや、アタッチされたルートまたは OU の下のすべてのアカウントの IAM ユーザーとロールのアクセス許可に影響します。

AWS CLI & AWS SDKs

ルート、OU、またはアカウントに移動して SCP をアタッチするには

以下のコード例は、`AttachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
```

```
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Shows how to attach an AWS Organizations policy to an organization,
    /// an organizational unit, or an account.
    /// </summary>
    public class AttachPolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and then calls the
        /// AttachPolicyAsync method to attach the policy to the root
        /// organization.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();
            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new AttachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.AttachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully attached Policy ID {policyId} to
Target ID: {targetId}.");
            }
            else
            {
                Console.WriteLine("Was not successful in attaching the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスAttachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシーを OU にアタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- API の詳細については、「 コマンドリファレンスAttachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def attach_policy(policy_id, target_id, orgs_client):
```

```
    """
    Attaches a policy to a target. The target is an organization root, account,
or
    organizational unit.

    :param policy_id: The ID of the policy to attach.
    :param target_id: The ID of the resources to attach the policy to.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.attach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Attached policy %s to target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't attach policy %s to target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 AttachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに有効になり、アタッチされたアカウントや、アタッチされたルートまたは OU の下のすべてのアカウントの IAM ユーザーとロールのアクセス許可に影響します。

組織ルート、OU、またはアカウントからの SCP のデタッチ

組織の管理アカウントにサインインすると、アタッチされている組織ルート、OU、またはアカウントから SCP をデタッチすることができます。SCP をエンティティからデタッチすると、その SCP は、現在デタッチされたエンティティの影響を受けた IAM ユーザーと IAM ロールには適用されません。SCP をデタッチするには、次の手順を実行します。

> **Note**
>
> ルート、OU、またはアカウントから最後の SCP をデタッチすることはできません。すべてのルート、OU、アカウントには常に少なくとも 1 つの SCP がアタッチされている必要があります。

> **最小アクセス許可**
>
> ルート、OU、またはアカウントから SCP をデタッチするには、以下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。
>
> - `organizations:DetachPolicy`

AWS Management Console

SCP をデタッチするには、ポリシーまたはそのポリシーをデタッチするルート、OU、アカウントに移動する必要があります。

SCP がアタッチされているルート、OU、またはアカウントに移動して SCP をデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、ポリシーをデタッチするルート、OU、またはアカウントに移動します。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。ルート、OU、またはアカウントの名前を選択します。
3. [Policies] (ポリシー) タブで、デタッチする SCP の横にあるラジオボタンを選択して、[Detach] (デタッチ) を選択します。
4. 確認ダイアログボックスで、[Detach policy] (ポリシーのデタッチ) を選択します。

   アタッチされている SCP のリストが更新されます。SCP のデタッチによるポリシーの変更はすぐに有効になります。例えば、SCP をデタッチすると、以前にアタッチされた 1 つ以上のアカウントの IAM ユーザーおよびロールのアクセス許可によって、以前にアタッチされた組織ルートまたは OU にすぐに影響します。

ポリシーに移動して SCP をデタッチするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスコントロールポリシーページで、ルート、OU、またはアカウントからデタッチするポリシーの名前を選択します。

3. [Targets] (ターゲット) タブで、ポリシーをデタッチするルート、OU、またはアカウントの横にあるラジオボタンをクリックします。場合によっては、目的の OU またはアカウントを表示するため、OU を展開 (▶ を選択) する必要があります。
4. [Detach] (デタッチ) を選択します。
5. 確認ダイアログボックスで、[Detach] (デタッチ) を選択します。

アタッチされている SCP のリストが更新されます。SCP のデタッチによるポリシーの変更はすぐに有効になります。例えば、SCP をデタッチすると、以前にアタッチされた 1 つ以上のアカウントの IAM ユーザーおよびロールのアクセス許可によって、以前にアタッチされた組織ルートまたは OU にすぐに影響します。

AWS CLI & AWS SDKs

ルート、OU、またはアカウントから SCP をデタッチするには

以下のコード例は、`DetachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to detach a policy from an AWS Organizations organization,
/// organizational unit, or account.
/// </summary>
public class DetachPolicy
```

```
    {
        /// <summary>
        /// Initializes the Organizations client object and uses it to call
        /// DetachPolicyAsync to detach the policy.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new DetachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.DetachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully detached policy with Policy Id:
{policyId}.");
            }
            else
            {
                Console.WriteLine("Could not detach the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDetachPolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111
 --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDetachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def detach_policy(policy_id, target_id, orgs_client):
    """
    Detaches a policy from a target.

    :param policy_id: The ID of the policy to detach.
    :param target_id: The ID of the resource where the policy is currently
 attached.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.detach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Detached policy %s from target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 DetachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

ポリシーの変更はすぐに有効になり、アタッチされたアカウントや、アタッチされたルートまたは OU の下のすべてのアカウントの IAM ユーザーとロールのアクセス許可に影響します。

SCP 評価

> **Note**
>
> このセクションの情報は、AI サービスのオプトアウトポリシー、バックアップポリシー、タグポリシーなどの管理ポリシータイプには適用されません。詳細については、「管理ポリシーの継承を理解する」を参照してください。

AWS Organizationsではさまざまなレベルで複数のサービスコントロールポリシー (SCP) を関連付けることができるため、SCP の評価方法を理解しておくと、正しい結果をもたらす SCP を作成するのに役立ちます。

トピック

- SCP と Allow の連携の仕組み
- SCP と Deny の連携の仕組み
- SCP の使用戦略

SCP と Allow の連携の仕組み

特定のアカウントに対してアクセス許可を許可するには、アカウント (ターゲット アカウント自体を含む) への直接パスのルートから各 OU までのすべてのレベルで明示的な **`Allow`** ステートメントが必要です。このため、SCPs、 は Full という名前の AWS マネージド SCP AWSAccessポリシーをアAWS Organizations タッチし、すべてのサービスとアクションを許可します。このポリシーが組織のどのレベルでも削除され、置き換えられない場合、そのレベルのすべての OU とアカウントはいかなるアクションも実行できなくなります。

例えば、図 1 と図 2 のシナリオを見ていきましょう。アカウント B でアクセス権限またはサービスを許可するには、そのアクセス権限またはサービスを許可する SCP を Production OU であるルート、およびアカウント B 自体にアタッチする必要があります。

SCP 評価は deny-by-default モデルに従います。つまり、SCPs で明示的に許可されていないアクセス許可は拒否されます。ルート、Production OU、アカウント B などのどのレベルでも SCP に許可ステートメントが存在しない場合、アクセスは拒否されます。

メモ

- SCP 内の `Allow` ステートメントにより、`Resource` 要素に `"*"` エントリのみを含めることが許可されます。
- SCP 内の `Allow` ステートメントには、`Condition` 要素を含めることは一切できません。

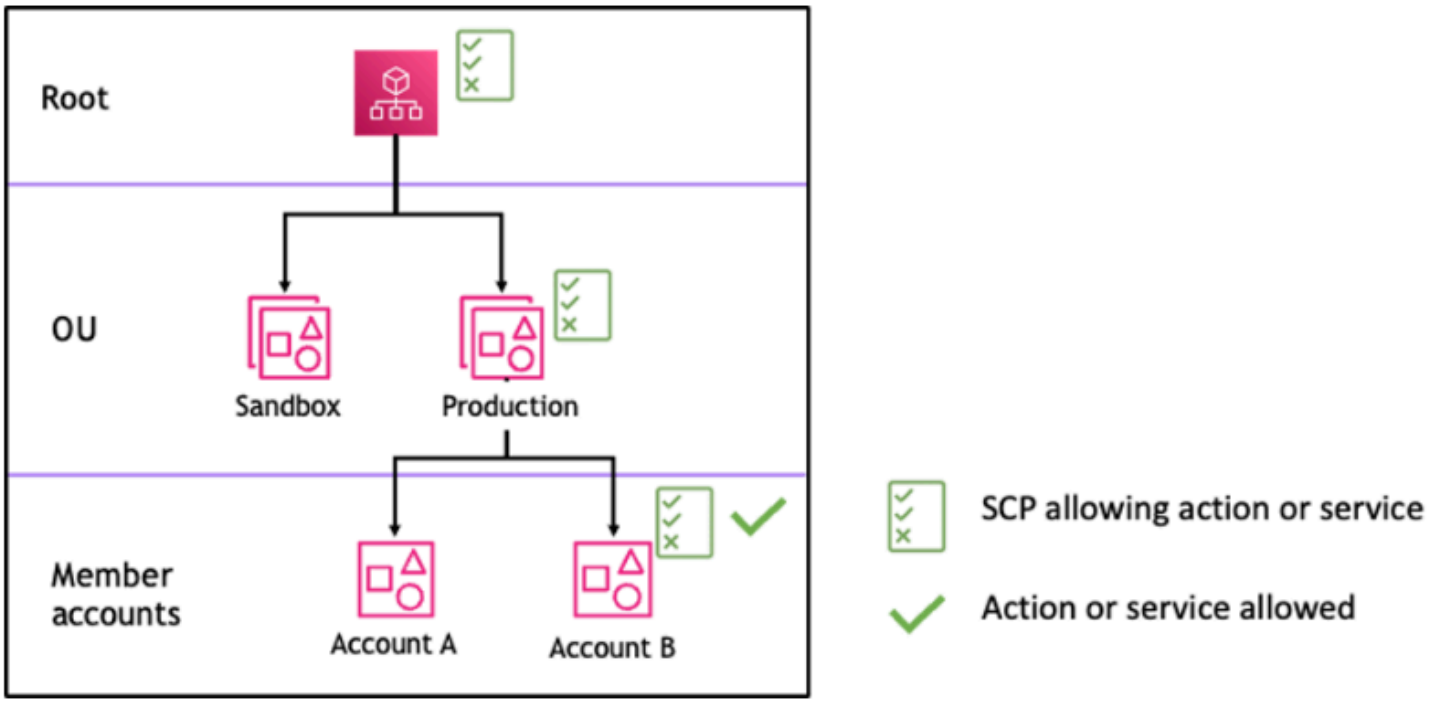


図 1: ルート、Production OU、アカウント B に `Allow` ステートメントがアタッチされた組織構造の例

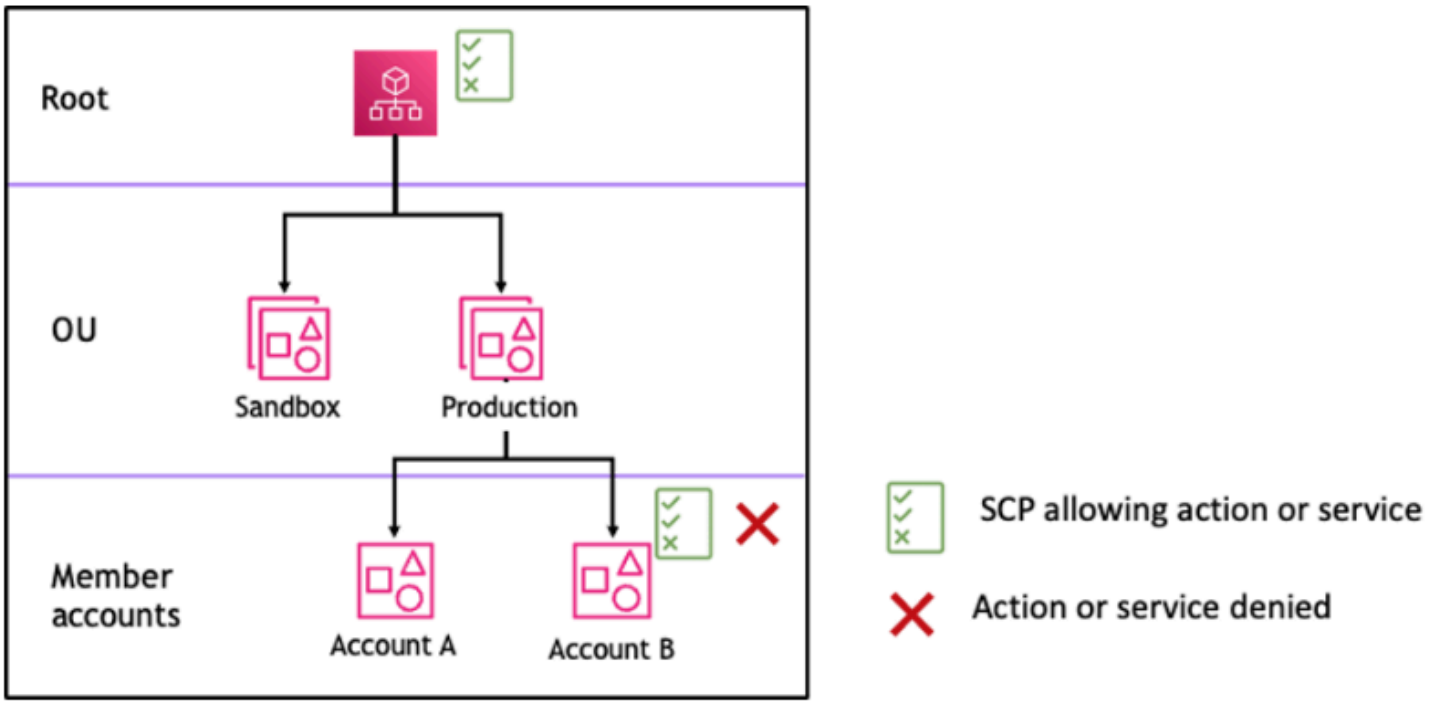


図 2: Production OU で `Allow` ステートメントが欠落している組織構造の例と、アカウント B への影響

SCP と Deny の連携の仕組み

特定のアカウントに対するアクセス許可が拒否される場合、ルートからアカウント (ターゲット アカウント自体を含む) への直接パス内の各 OU を経由するすべての SCPがそのアクセス許可を拒否できます。

例えば、Production OU にアタッチされている SCP に、特定のサービスに対して明示的な `Deny` ステートメントが指定されているとします。また、図 3 に示すように、同じサービスへのアクセスを明示的に許可する別の SCP がルートとアカウント B にアタッチされている場合もあります。その結果、組織内のどのレベルにも適用された拒否ポリシーが、その下にあるすべての OU とメンバーアカウントに対して評価されるため、アカウント A とアカウント B の両方がサービスへのアクセスを拒否されます。

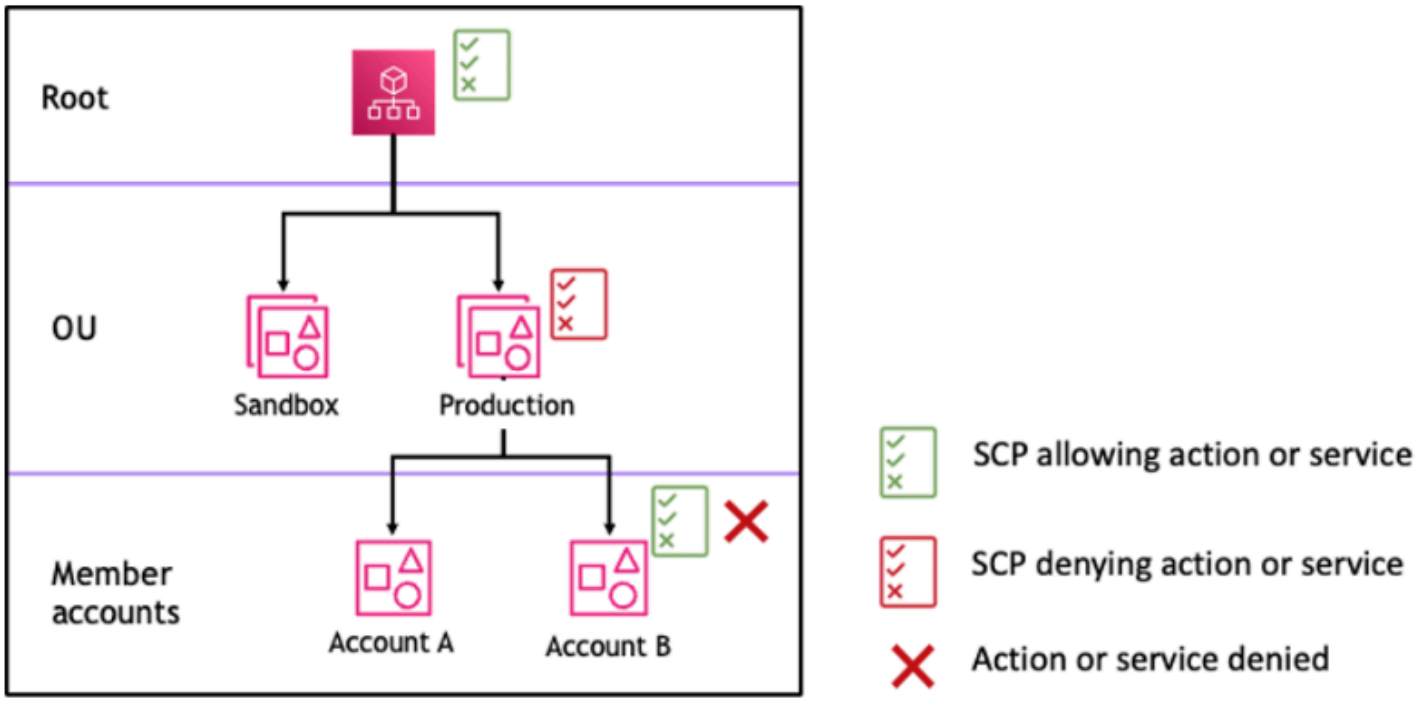


図 3: Production OU で Deny ステートメントがアタッチされた組織構造の例と、アカウント B への影響

SCP の使用戦略

SCP を作成する際、Allow と Deny ステートメントを組み合わせて使用することで、組織内で意図したアクションやサービスを実現できます。Deny ステートメントは、ルートレベルまたは OU レベルで適用されると、その下にあるすべてのアカウントに影響するため、組織や OU のより広い範囲に適用すべき制限を実装する強力な方法です。

例えば、メンバーアカウントが組織を離れるのを禁止するルートレベルで へのDenyステートメントを使用してポリシーを実装できます。これは、組織内のすべてのアカウントに対して有効になります。Deny ステートメントは、例外の作成に役立つ条件要素もサポートしています。

> **Tip**
>
> IAM のサービスの最終アクセスデータを使用して SCPs、必要な AWS サービスのみにアクセスを制限できます。詳細については、IAM ユーザーガイドの「組織の Organizations サービスの最終アクセス時間データを表示する」を参照してください。

AWS Organizations は、作成時にすべてのルート、OU、アカウントに FullAWSAccess という名前の AWS マネージド SCP をアタッチします。このポリシーはすべてのサービスとアクションを許可します。FullAWSAccess を一連のサービスのみを許可するポリシーに置き換えて、SCPsを更新することで明示的に許可されない限り、新しい AWS サービスを許可しないようにすることができます。例えば、組織内で一部のサービスの使用のみを許可したい場合は、Allow ステートメントを使用して特定のサービスのみを許可できます。

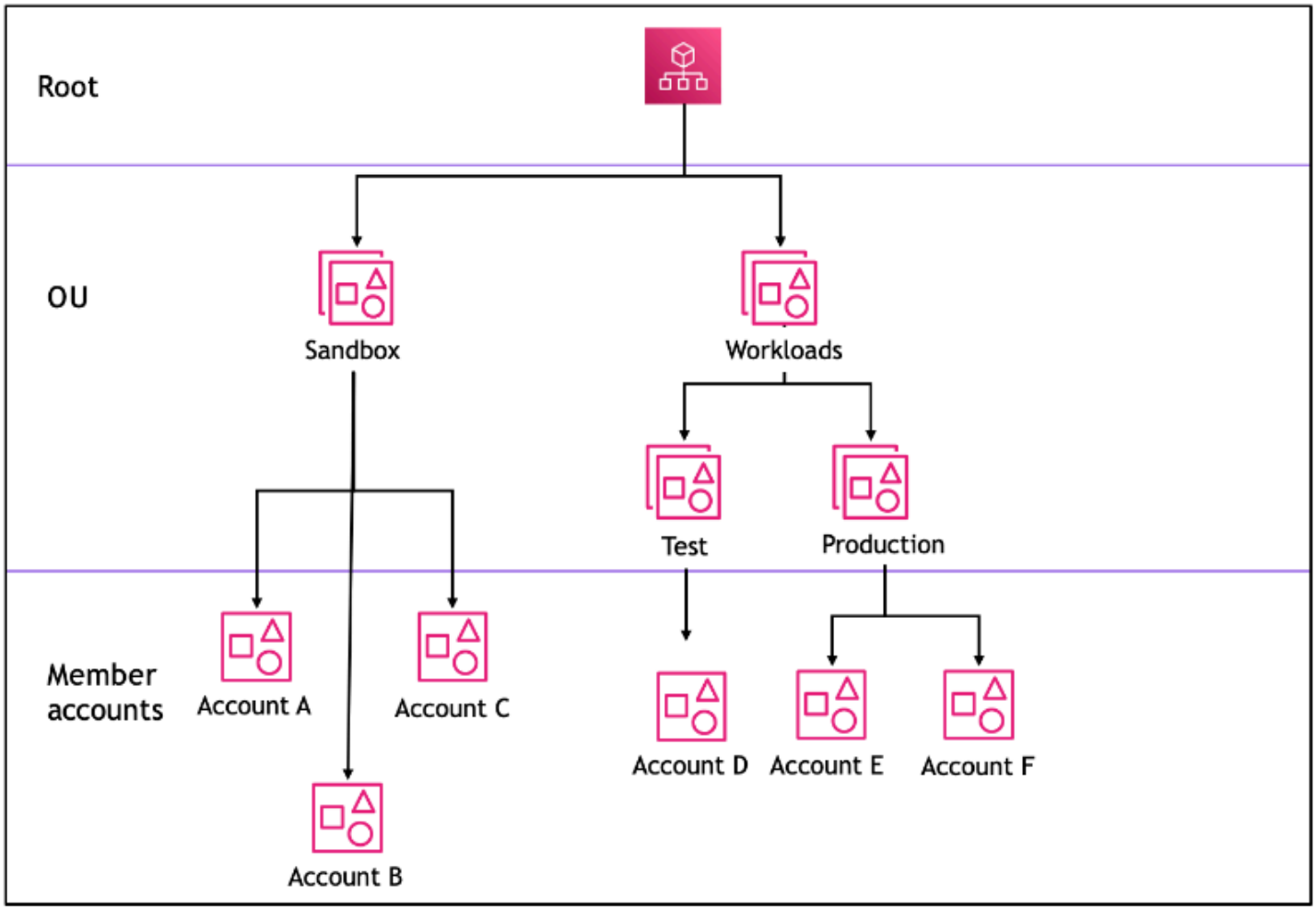
```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "cloudwatch:*",
                "organizations:*"
            ],
            "Resource": "*"
        }
    ]
}
```

この 2 つのステートメントを組み合わせたポリシーは、次の例のようになります。このポリシーでは、メンバーアカウントが組織から退出することを防ぎ、必要な AWS サービスの使用を許可します。組織管理者は、代わりにフルAWSAccessポリシーをデタッチし、これをアタッチできます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:*",
                "cloudwatch:*",
                "organizations:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action":"organizations:LeaveOrganization",
            "Resource": "*"
```

```
        }
    ]
}
```

次に、以下のサンプル組織構造を検討して、組織内のさまざまなレベルで複数の SCP を適用する方法を理解してください。



次の表は、サンドボックス OU で有効なポリシーを示しています。

シナリオ	[ルート] における SCP	[サンドボックス] OU における SCP	[アカウント A] における SCP	[アカウント A] における適用されるポリシー	[アカウント B] と [アカウント C] における適用されるポリシー
1	フル AWS アクセス	フル AWS アクセス + S3 アクセス拒否	フル AWS アクセス + EC2 アクセス拒否	S3 も EC2 もアクセスなし	S3 アクセスなし
2	フル AWS アクセス	EC2 アクセスを許可する	EC2 アクセスを許可する	フル AWS アクセス	フル AWS アクセス
3	S3 アクセスを拒否する	S3 アクセスを許可する	フル AWS アクセス	S3 アクセスなし	S3 アクセスなし

次の表は、ワークロード OU で有効なポリシーを示しています。

シナリオ	[ルート] における SCP	[ワークロード] OU における SCP	[テスト] OU におけるSCP	[アカウント D] における適用されるポリシー	[Production] OU、[アカウント E]、[アカウント F] における適用されるポリシー
1	フル AWS アクセス	フル AWS アクセス	フル AWS アクセス + EC2 アクセス拒否	EC2 アクセスなし	フル AWS アクセス
2	フル AWS アクセス	フル AWS アクセス	EC2 アクセスを許可する	フル AWS アクセス	フル AWS アクセス
3	S3 アクセスを拒否する	フル AWS アクセス	S3 アクセスを許可する	S3 アクセスなし	S3 アクセスなし

SCP 構文

サービスコントロールポリシー (SCPs、 AWS Identity and Access Management (IAM) アクセス許可ポリシーおよびリソースベースのポリシー (Amazon S3 バケットポリシーなど) で使用される構文と同様の構文を使用します。IAM ポリシーの詳細とその構文については、IAM ユーザーガイドの「IAM ポリシーの概要」を参照してください。

SCP は、JSON のルールに従って構造化されたプレーンテキストファイルです。このトピックで説明されている要素を使用します。

> **Note**
>
> SCP 内のすべての文字は、その上限サイズに対してカウントされます。このガイドの例では、読みやすさを向上させるため、空白文字を追加してフォーマットされた SCP を示します。ただし、ポリシーサイズが上限サイズに近づいている場合は、スペースを節約するために、引用符の外側にあるすべての空白文字 (スペースや改行など) を削除できます。

SCP に関する一般情報については、「サービスコントロールポリシー (SCP)」を参照してください。

要素の概要

次の表には、SCP で使用できるポリシー要素を要約しています。一部のポリシー要素はアクションを拒否する SCP のみで使用できます。[Supported Effects] (サポートされる効果) の列には、SCP の各ポリシー要素で使用できる効果のタイプが一覧表示されています。

要素	目的	サポートされる効果
Version	ポリシーの処理に使用する言語構文ルールを指定します。	`Allow, Deny`
Statement	ポリシー要素の	`Allow, Deny`

要素	目的	サポートされる効果
	コンテナとして機能します。SCP には複数のステートメントを含めることができます。	
Statement ID (Sid)	(オプション) ステートメントにわかりやすい名前を付けます。	`Allow`, `Deny`

要素	目的	サポートされる効果
[Effect] (効果)	SCP ステートメントがプリンシパルおよびアカウント内の IAM ユーザーとロールへのアクセスを許可するか拒否するかを定義します。	`Allow`, `Deny`
[アクション]	SCP が許可または拒否する AWS サービスとアクションを指定します。	`Allow`, `Deny`

要素	目的	サポートされる効果
NotAction	SCP から除外される AWS サービスとアクションを指定します。Actior 要素の代わりに使用します。	Deny
リソース	SCP が適用される AWS リソースを指定します。	Deny
条件	ステートメントを実行するタイミングの条件を指定します。	Deny

次のセクションでは、SCP でポリシー要素がどのように使用されるかについての詳細および例を提供しています。

Version 要素

すべての SCP には、Version値を持つ要素"2012-10-17"が含まれる必要があります。これは、IAM アクセス許可ポリシーの最新バージョンと同じバージョンの値です。

```
    "Version": "2012-10-17",
```

詳細については、IAM ユーザーガイドの「IAM JSON ポリシー要素: Version」を参照してください。

Statement 要素

SCP は、1 つ以上の Statement 要素で構成されます。ポリシーには Statement キーワードを 1 つだけ含めることができますが、値は、ステートメントの JSON 配列 ([ ] の文字で囲まれる) もあります。

以下の例は、単一の Effect 要素、Action 要素、Resource 要素で構成される単一のステートメントを示しています。

```
    "Statement": {
        "Effect": "Allow",
        "Action": "*",
        "Resource": "*"
    }
```

次の例では、1 つの Statement 要素内に、配列リストとして 2 つのステートメントが含まれています。最初のステートメントではすべてのアクションが許可されますが、2 つ目のステートメントではすべての EC2 アクションが拒否されます。結果的に、アカウントの管理者は、Amazon Elastic Compute Cloud (Amazon EC2) 以外のアクセス許可をすべて委譲できます。

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "ec2:*",
            "Resource": "*"
        }
    ]
```

詳細については、IAM ユーザーガイドの「IAM JSON ポリシー要素: Statement」を参照してください。

ステートメント ID (**Sid**) 要素

Sid は、ポリシーステートメントに提供するオプションの識別子です。Sid 値は、ステートメント配列内の各ステートメントに割り当てることができます。次の SCP の例は、Sid ステートメントのサンプルを示しています。

```
{
    "Statement": {
        "Sid": "AllowsAllActions",
        "Effect": "Allow",
        "Action": "*",
        "Resource": "*"
    }
}
```

詳細については、IAM ユーザーガイドの「IAM JSON ポリシー要素: ID」を参照してください。

Effect 要素

各ステートメントには必ず Effect を 1 つ含める必要があります。この値は Allow または Deny となります。これは、同じステートメントにリストされたアクションが影響を受けます。

詳細については、IAM ユーザーガイドの「IAM JSON ポリシーの要素: Effect」を参照してください。

"Effect": "Allow"

次の例では、Allow の値を持つ Effect 要素を含むステートメントを持つ SCP を示しています。これにより、アカウントユーザーが Amazon S3 サービスのアクションを実行できるようになります。この例は、許可リスト戦略を使用する組織で役立ちます (デフォルトの FullAWSAccess ポリシーがすべてデタッチされているため、デフォルトでアクセス許可は暗黙的に拒否されます)。その結果、このステートメントでは、アタッチされているすべてのアカウントに対する Amazon S3 のアクセスが許可されます。

```
{
    "Statement": {
        "Effect": "Allow",
        "Action": "s3:*",
        "Resource": "*"
    }
```

```
}
```

このステートメントは IAM アクセス許可ポリシーに同じ `Allow` 値を使用しますが、SCP では実際に何かを実行するためのアクセス許可をユーザーに付与しません。代わりに、SCPs組織内の IAM ユーザーと IAM ロールの最大アクセス許可を指定するフィルターとして機能します。前述の例では、アカウント内のユーザーで `AdministratorAccess` 管理ポリシーがアタッチされている場合でも、SCP は影響を受けるアカウントのすべてのユーザーによるアクションを Amazon S3 アクションのみに制限します。

`"Effect": "Deny"`

また、`Effect` 要素に `Deny` の値があるステートメントでは、SCP の有効時に特定のリソースへのアクセスを制限したり、条件を定義することもできます。

次の例は、拒否ステートメントで条件キーを使用する方法の例を示しています。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "ec2:RunInstances",
        "Resource": "arn:aws:ec2:*:*:instance/*",
        "Condition": {
            "StringNotEquals": {
                "ec2:InstanceType": "t2.micro"
            }
        }
    }
}
```

SCP のこのステートメントは、影響が及ぶアカウント (SCP がアカウント自体にアタッチされているか、アカウントがある組織ルートあるいは OU にアタッチされている場合) が、Amazon EC2 インスタンスが `t2.micro` に設定されていないときに Amazon EC2 インスタンスを起動しないようにするガードレールを設定します。このアクションを許可する IAM ポリシーがアカウントにアタッチされている場合でも、SCP によって作成されたガードレールはこれを許可しません。

`Action` および `NotAction` 要素

各ステートメントには、次のいずれかが含まれている必要があります。

- 許可あるいは拒否ステートメントの `Action` 要素。

- 拒否ステートメントのみ (Effect 要素の値が Deny の場合) では、Action または NotAction 要素。

Action または NotAction要素の値は、ステートメントによって許可または拒否される AWS サービスとアクションを識別する文字列のリスト (JSON 配列) です。

各文字列は、サービスの略称 (「s3」、「ec2」、「iam」、「organizations」など) で構成されており、すべて小文字で、コロンと、その後にサービスのアクションが続きます。アクションおよび注釈は大文字と小文字が区別され、各サービスのドキュメントに示されているとおりに入力する必要があります。一般的にこれらはすべて、大文字で始まり、残りは小文字の各単語で入力されます。例: "s3:ListAllMyBuckets"。

SCP でアスタリスク (*) や疑問符 (?) などのワイルドカード文字を使用することもできます。

- 名前の一部を共有する複数のアクションを検索するには、アスタリスクをワイルドカードとして使用します。値 "s3:*" は、Amazon S3 サービス内のすべてのアクションを意味します。値 "ec2:Describe*" は「Describe」で始まる EC2 アクションのみに一致します。
- 単一の文字を検索する場合は疑問符 (?) を使用します。

> **Note**
>
> SCP では、Action または NotAction 要素のワイルドカード文字 (* および ?) は、要素自身、または文字列の末尾にのみ使用できます。文字列の先頭または中間には表示されません。そのため、"servicename:action*" は有効ですが、"servicename:*action" と "servicename:some*action" はいずれも、SCP で無効です。

AWS Organizations SCPs「サービス認証リファレンス」の AWS 「 サービスのアクション、リソース、および条件キー」を参照してください。

詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素: アクション」および「IAM JSON ポリシー要素： NotAction」を参照してください。

Action 要素の例

次の例では、アカウント管理者に、アカウント内の EC2 インスタンスの許可を記述、開始、停止、終了する権限を委譲することを許可するステートメントを持つ SCP を示しています。これは、許可リストの例であり、デフォルトではアクセス許可を暗黙的に拒否するためにデフォルトの

Allow * ポリシーがアタッチされていない場合に便利です。デフォルトの Allow * ポリシーがルート、OU、次のポリシーがアタッチされるアカウントに引き続きアタッチされている場合は、ポリシーの効果はありません。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
          "ec2:DescribeInstances", "ec2:DescribeImages", "ec2:DescribeKeyPairs",
          "ec2:DescribeSecurityGroups", "ec2:DescribeAvailabilityZones",
 "ec2:RunInstances",
          "ec2:TerminateInstances", "ec2:StopInstances", "ec2:StartInstances"
        ],
        "Resource": "*"
    }
}
```

次の例では、アタッチされたアカウントで使用されたくないサービスへのアクセスを拒否する方法を示しています。デフォルトの "Allow *" SCP がすべての OU および root にまだアタッチされていることを前提としています。このポリシーの例では、アタッチされたアカウントのアカウント管理者は、IAM、Amazon EC2、Amazon RDS サービスにアクセス許可を移譲することはできません。他のサービスからのあらゆるアクションは、移譲を拒否する他のポリシーがアタッチされていない限り移譲できます。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": [ "iam:*", "ec2:*", "rds:*" ],
        "Resource": "*"
    }
}
```

NotAction 要素の例

次の例は、 NotAction要素を使用してポリシーの影響から AWS サービスを除外する方法を示しています。

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

```
  "Statement": [
    {
      "Sid": "LimitActionsInRegion",
      "Effect": "Deny",
      "NotAction": "iam:*",
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": "us-west-1"
        }
       }
     }
   ]
}
```

このステートメントでは、影響を受けるアカウントは、IAM アクションを使用する場合を除き AWS リージョン、指定された でアクションを実行することに限定されます。

`Resource` 要素

`Effect` 要素に `Allow` の値があるステートメントでは、SCP の `Resource` 要素の「`*`」のみを指定することができます。個々のリソースの Amazon リソースネーム (ARN) を指定することはできません。

リソース要素でアスタリスク (*) や疑問符 (?) などのワイルドカード文字を使用することもできます。

- 名前の一部を共有する複数のアクションを検索するには、アスタリスクをワイルドカードとして使用します。
- 単一の文字を検索する場合は疑問符 (?) を使用します。

`Effect` 要素に `Deny` の値があるステートメントでは、次の例に示すように、個々の ARN を指定することができます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessToAdminRole",
      "Effect": "Deny",
      "Action": [
```

```
        "iam:AttachRolePolicy",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
      ],
      "Resource": [
        "arn:aws:iam::*:role/role-to-deny"
      ]
    }
  ]
}
```

この SCP は、影響を受けるアカウントの IAM ユーザーとロールが、組織内のすべてのアカウントに作成された共通の管理 IAM ロールに変更を加えることを制限します。

詳細については、IAM ユーザーガイドの「IAM JSON ポリシー要素: Resource」を参照してください。

`Condition` 要素

SCP の拒否ステートメントに `Condition` 要素を指定することができます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAllOutsideEU",
            "Effect": "Deny",
            "NotAction": [
                "cloudfront:*",
                "iam:*",
                "route53:*",
                "support:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
```

```
                    "aws:RequestedRegion": [
                        "eu-central-1",
                        "eu-west-1"
                    ]
                }
            }
        }
    ]
}
```

この SCP は、リストされたサービスを除く、eu-central-1 および eu-west-1 リージョンの外部のすべてのオペレーションへのアクセスを拒否します。

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。

サポートされていない要素

以下の要素は SCP ではサポートされていません。

- Principal
- NotPrincipal
- NotResource

サービスコントロールポリシーの例

このトピックで表示されているサービスコントロールポリシー (SCP) の例は、情報提供のみを目的としています。

> **これらの例を使用する前に**
>
> 組織内でこれらの SCP の例を使用する前に、以下のことを実行します。
>
> - お客様の固有要件に応じて SCP を慎重にレビューし、カスタマイズします。
> - 使用する AWS のサービスを使用して、環境内の SCP の完全なテストを実施します。
>
> このセクションのポリシーの例では、SCP の実装と使用について説明します。これらの例は、公式な AWS 推奨事項やベストプラクティスとして解釈されることを意図したものではありません。拒否ベースのポリシーが、お客様の環境のビジネス要件を解決するために適切であるかどうかを慎重にテストする責任はお客様にあります。拒否ベースのサービ

スコントロールポリシーでは、必要な例外をポリシーに追加しない限り、意図せず AWS のサービスの利用を制限またはブロックしてしまうことがあります。このような例外の例については、不要な AWS リージョン へのアクセスをブロックするルールで、グローバルサービスを除外する 1 つ目の例を参照してください。

- SCP は、ルートユーザーを含む、それが関連付けられているすべてのアカウントのすべてのユーザーとロールに影響することに注意してください。

> **Tip**
>
> IAM でサービスの最終アクセス時間データを使用して SCP を更新し、必要な AWS サービスのみへのアクセスを制限できます。詳細については、IAM ユーザーガイドの「組織の Organizations サービスの最終アクセス時間データを表示する」を参照してください。

次の各種ポリシーは、「拒否リストポリシー」戦略の例です。拒否リストポリシーは、影響を受けるアカウントの承認済みアクションを許可する他のポリシーと合わせてアタッチする必要があります。例えば、デフォルトの `FullAWSAccess` ポリシーは、アカウントによるすべてのサービスの使用を許可します。このポリシーは、デフォルトによって、ルート、すべての組織単位 (OU)、およびすべてのアカウントにアタッチされます。実際、アクセス許可は付与されません。付与する SCP がないためです。代わりに、そのアカウントの管理者は、アカウントのユーザーやロール、またはグループに標準の AWS Identity and Access Management (IAM) アクセス許可ポリシーをアタッチすることで、それらのアクションへのアクセスを委譲することができます。これらの各拒否リストポリシーによって、指定されたサービスまたはアクションへのアクセスがブロックされ、ポリシーはすべて上書きされます。

例

- 一般的な例
  - リクエスト AWS された に基づいて へのアクセスを拒否する AWS リージョン
  - IAM ユーザーとロールによる特定の変更を禁止する
  - 指定された管理者ロール以外の IAM ユーザーとロールが特定の変更を行うことを禁止する
  - API アクションの実行に MFA を要求する
  - ルートユーザーによるサービスへのアクセスをブロックする
  - メンバーアカウントが組織を離れるのを禁止する
- Amazon CloudWatch の SCP の例

- ユーザーによる CloudWatch の無効化または設定の変更を禁止する
- AWS Config の SCP の例
  - ユーザーによる AWS Config の無効化またはルールの変更を禁止する
- Amazon Elastic Compute Cloud (Amazon EC2) の SCP の例
  - 指定するタイプを使用するよう Amazon EC2 インスタンスに要求する
  - IMDSv2 なしで EC2 インスタンスが起動することを禁止する
  - デフォルトの Amazon EBS 暗号化の無効化を禁止する
- Amazon GuardDuty の SCP の例
  - ユーザーによる GuardDuty の無効化または設定の変更を禁止する
- の SCPsの例 AWS Resource Access Manager
  - 外部共有の禁止
  - 特定のアカウントが、指定したリソースタイプのみを共有できるようにする
  - 組織または組織単位 (OU) との共有を禁止する
  - 指定した IAM ユーザーおよびロールのみとの共有を許可する
- Amazon Route 53 Application Recovery Controller の SCP 例
  - ユーザーが Route 53 ARC ルーティング制御状態を更新できないようにする
- Amazon S3 の SCP の例
  - Amazon S3 の暗号化されていないオブジェクトのアップロードを禁止する
- リソースのタグ付けの SCP の例
  - 作成される特定のリソースにタグを要求する
  - 許可されたプリンシパル以外のタグが変更されないようにする
- Amazon Virtual Private Cloud (Amazon VPC) の SCP の例
  - ユーザーによる VPC フローログの削除を禁止する
  - インターネットアクセスに接続されていない VPC を使用した取得を禁止する

一般的な例

リクエスト AWS された に基づいて へのアクセスを拒否する AWS リージョン

この SCP は、指定されたリージョン外でのオペレーションへのアクセスを拒否します。`eu-central-1` と を AWS リージョン 、使用する `eu-west-1`に置き換えます。これにより、承認され

たグローバルサービスでオペレーションが除外されます。この例では、指定した 2 つの管理者ロールのいずれかによるリクエストを除外する方法についても説明します。

> **Note**
>
> でリージョン拒否 SCP を使用するには AWS Control Tower、「 AWS Control Tower コントロールリファレンスガイド」の「リクエスト AWS された に基づいて へのアクセスを拒否 AWS リージョンする」を参照してください。

このポリシーは、Deny 効果を使用して、承認された 2 つのリージョン (eu-central-1 および eu-west-1) のいずれかをターゲットとしないオペレーションに対するリクエストへのアクセスを拒否します。NotAction 要素を使用すると、オペレーション (または個々のオペレーション) がこの制限から除外されているサービスを一覧表示できます。グローバルサービスには us-east-1 リージョンによって物理的にホストされるエンドポイントがあるため、この方法で除外する必要があります。このように構成された SCP では、リクエストされたサービスが NotAction 要素に含まれている場合、us-east-1 リージョン内のグローバルサービスに対するリクエストが許可されます。us-east-1 リージョン内のサービスに対するその他のリクエストは、このポリシー例によって拒否されます。

> **Note**
>
> この例では、最新のグローバル AWS サービスまたはオペレーションがすべて含まれていない場合があります。サービスとオペレーションのリストを、組織内のアカウントによって使用されるグローバルサービスで置換えてください。
>
> > **ヒント**
> >
> > IAM コンソールでサービスの最終アクセスデータを表示すると、組織が使用しているグローバルサービスを確認できます。IAM ユーザー、グループ、またはロールの詳細ページの [Access Advisor] (アクセスアドバイザー) タブには、そのエンティティによって使用された AWS のサービスが最新のアクセス順に表示されます。

考慮事項

- AWS KMS および はリージョンエンドポイント AWS Certificate Manager をサポートします。ただし、Amazon などのグローバルサービスで使用する場合は、次の SCP 例のグローバルサービス除外リストに含める CloudFront 必要があります。Amazon などのグローバルサービスでは、 CloudFront 通常、同じリージョンの AWS KMS と ACM にアクセスする必要があります。グローバルサービスの場合は、米国東部 (バージニア北部) リージョン () ですus-east-1。
- デフォルトでは、 AWS STS はグローバルサービスであり、グローバルサービス除外リストに含める必要があります。ただし、 を有効に AWS STS して、単一のグローバルエンドポイントの代わりにリージョンエンドポイントを使用できます。これを行うと、次の SCP の例にあるグローバルサービスの免除リストから STS を削除できます。詳細については、「AWS STS での の管理 AWS リージョン」を参照してください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAllOutsideEU",
            "Effect": "Deny",
            "NotAction": [
                "a4b:*",
                "acm:*",
                "aws-marketplace-management:*",
                "aws-marketplace:*",
                "aws-portal:*",
                "budgets:*",
                "ce:*",
                "chime:*",
                "cloudfront:*",
                "config:*",
                "cur:*",
                "directconnect:*",
                "ec2:DescribeRegions",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeVpnGateways",
                "fms:*",
                "globalaccelerator:*",
                "health:*",
```

```
                "iam:*",
                "importexport:*",
                "kms:*",
                "mobileanalytics:*",
                "networkmanager:*",
                "organizations:*",
                "pricing:*",
                "route53:*",
                "route53domains:*",
                "route53-recovery-cluster:*",
                "route53-recovery-control-config:*",
                "route53-recovery-readiness:*",
                "s3:GetAccountPublic*",
                "s3:ListAllMyBuckets",
                "s3:ListMultiRegionAccessPoints",
                "s3:PutAccountPublic*",
                "shield:*",
                "sts:*",
                "support:*",
                "trustedadvisor:*",
                "waf-regional:*",
                "waf:*",
                "wafv2:*",
                "wellarchitected:*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestedRegion": [
                        "eu-central-1",
                        "eu-west-1"
                    ]
                },
                "ArnNotLike": {
                    "aws:PrincipalARN": [
                        "arn:aws:iam::*:role/Role1AllowedToBypassThisSCP",
                        "arn:aws:iam::*:role/Role2AllowedToBypassThisSCP"
                    ]
                }
            }
        }
    ]
}
```

IAM ユーザーとロールによる特定の変更を禁止する

この SCP は、IAM ユーザーとロールが、組織内のすべてのアカウントで作成した特定の IAM ロールに変更を加えることを制限します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessToASpecificRole",
      "Effect": "Deny",
      "Action": [
        "iam:AttachRolePolicy",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
      ],
      "Resource": [
        "arn:aws:iam::*:role/name-of-role-to-deny"
      ]
    }
  ]
}
```

指定された管理者ロール以外の IAM ユーザーとロールが特定の変更を行うことを禁止する

この SCP は、前述の例において、管理者の例外を追加します。これにより、影響を受けるアカウントの IAM ユーザーやロールが、指定されたロールを使用する、組織内の管理者以外のすべてのアカウントで作成された共通の管理用 IAM ロールに変更を加えられないようにします。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAccessWithException",
      "Effect": "Deny",
      "Action": [
```

```
        "iam:AttachRolePolicy",
        "iam:DeleteRole",
        "iam:DeleteRolePermissionsBoundary",
        "iam:DeleteRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePermissionsBoundary",
        "iam:PutRolePolicy",
        "iam:UpdateAssumeRolePolicy",
        "iam:UpdateRole",
        "iam:UpdateRoleDescription"
      ],
      "Resource": [
        "arn:aws:iam::*:role/name-of-role-to-deny"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/name-of-admin-role-to-allow"
        }
      }
    }
  ]
}
```

API アクションの実行に MFA を要求する

以下のような SCP を使用して、IAM ユーザーまたはロールがアクションを実行する前に多要素認証 (MFA) を要求するようにします。この例では、アクションは Amazon EC2 インスタンスを停止することです。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyStopAndTerminateWhenMFAIsNotPresent",
      "Effect": "Deny",
      "Action": [
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*",
      "Condition": {"BoolIfExists": {"aws:MultiFactorAuthPresent": false}}
    }
  ]
```

```
}
```

ルートユーザーによるサービスへのアクセスをブロックする

以下のポリシーでは、メンバーアカウントのルートユーザーに対して、指定したアクションへのすべてのアクセスを制限します。アカウントで特定の方法のルート認証情報を使用できないようにするには、このポリシーに独自のアクションを追加します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RestrictEC2ForRoot",
      "Effect": "Deny",
      "Action": [
        "ec2:*"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringLike": {
          "aws:PrincipalArn": [
            "arn:aws:iam::*:root"
          ]
        }
      }
    }
  ]
}
```

メンバーアカウントが組織を離れるのを禁止する

次のポリシーでは、LeaveOrganization API オペレーションを使用して、メンバーアカウントの管理者が組織からアカウントを削除できないようにします。

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

```
                "organizations:LeaveOrganization"
            ],
            "Resource": "*"
        }
    ]
}
```

Amazon CloudWatch の SCP の例

このカテゴリの例

- ユーザーによる CloudWatch の無効化または設定の変更を禁止する

ユーザーによる CloudWatch の無効化または設定の変更を禁止する

CloudWatch の下位レベルのオペレーターについては、ダッシュボードとアラームをモニタリングする必要があります。ただし、オペレーターが、上級者が設置したダッシュボードやアラームを削除または変更できないようにする必要があります。この SCP では、影響を受けるアカウントのユーザーまたはロールは、ダッシュボードまたはアラームを削除または変更する可能性のある CloudWatch コマンドを実行できなくなります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "cloudwatch:DeleteAlarms",
        "cloudwatch:DeleteDashboards",
        "cloudwatch:DisableAlarmActions",
        "cloudwatch:PutDashboard",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:SetAlarmState"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS Config の SCP の例

このカテゴリの例

- ユーザーによる AWS Config の無効化またはルールの変更を禁止する

ユーザーによる AWS Config の無効化またはルールの変更を禁止する

この SCP は、影響を受けるアカウントのユーザーまたはロールは、AWS Config の無効化や、ルールまたはトリガーの変更を無効にする可能性のある AWS Config オペレーションを実行できなくなります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "config:DeleteConfigRule",
        "config:DeleteConfigurationRecorder",
        "config:DeleteDeliveryChannel",
        "config:StopConfigurationRecorder"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon Elastic Compute Cloud (Amazon EC2) の SCP の例

このカテゴリの例

- 指定するタイプを使用するよう Amazon EC2 インスタンスに要求する
- IMDSv2 なしで EC2 インスタンスが起動することを禁止する
- デフォルトの Amazon EBS 暗号化の無効化を禁止する

指定するタイプを使用するよう Amazon EC2 インスタンスに要求する

この SCP を使用すると、`t2.micro` インスタンスタイプを使用していないすべてのインスタンスの起動は拒否されます。

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

```
      "Sid": "RequireMicroInstanceType",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "ec2:InstanceType": "t2.micro"
        }
      }
    }
  ]
}
```

IMDSv2 なしで EC2 インスタンスが起動することを禁止する

以下のポリシーは、すべてのユーザーが IMDSv2 なしで EC2 インスタンスを起動することを制限します。

```
[
   {
      "Effect":"Deny",
      "Action":"ec2:RunInstances",
      "Resource":"arn:aws:ec2:*:*:instance/*",
      "Condition":{
         "StringNotEquals":{
            "ec2:MetadataHttpTokens":"required"
         }
      }
   },
   {
      "Effect":"Deny",
      "Action":"ec2:RunInstances",
      "Resource":"arn:aws:ec2:*:*:instance/*",
      "Condition":{
         "NumericGreaterThan":{
            "ec2:MetadataHttpPutResponseHopLimit":"3"
         }
      }
   },
   {
      "Effect":"Deny",
```

```
      "Action":"*",
      "Resource":"*",
      "Condition":{
         "NumericLessThan":{
            "ec2:RoleDelivery":"2.0"
         }
      }
   },
   {
      "Effect":"Deny",
      "Action":"ec2:ModifyInstanceMetadataOptions",
      "Resource":"*"
   }
]
```

以下のポリシーは、すべてのユーザーが IMDSv2 なしで EC2 インスタンスを起動することを制限しますが、特定の IAM ID がインスタンスのメタデータオプションを変更することを許可します。

```
[
  {
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "StringNotEquals": {
        "ec2:MetadataHttpTokens": "required"
      }
    }
  },
  {
    "Effect": "Deny",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
      "NumericGreaterThan": {
        "ec2:MetadataHttpPutResponseHopLimit": "3"
      }
    }
  },
  {
    "Effect": "Deny",
    "Action": "*",
    "Resource": "*",
```

```
    "Condition": {
      "NumericLessThan": {
        "ec2:RoleDelivery": "2.0"
      }
    }
  },
  {
    "Effect": "Deny",
    "Action": "ec2:ModifyInstanceMetadataOptions",
    "Resource": "*",
    "Condition": {
      "StringNotLike": {
        "aws:PrincipalARN": [
          "arn:aws:iam::{ACCOUNT_ID}:{RESOURCE_TYPE}/{RESOURCE_NAME}"
        ]
      }
    }
  }
]
```

デフォルトの Amazon EBS 暗号化の無効化を禁止する

以下のポリシーは、すべてのユーザーがデフォルトの Amazon EBS 暗号化を無効にすることを制限します。

```
{
  "Effect": "Deny",
  "Action": [
    "ec2:DisableEbsEncryptionByDefault"
  ],
  "Resource": "*"
}
```

Amazon GuardDuty の SCP の例

このカテゴリの例

- ユーザーによる GuardDuty の無効化または設定の変更を禁止する

ユーザーによる GuardDuty の無効化または設定の変更を禁止する

この SCP は、影響を受けるアカウントのユーザーまたはロールが、コマンドとして直接、またはコンソールから GuardDuty を無効にしたり、設定を変更したりするのを防ぎます。これにより、GuardDuty の情報とリソースへの読み取り専用アクセスを効果的に実現できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "guardduty:AcceptInvitation",
                "guardduty:ArchiveFindings",
                "guardduty:CreateDetector",
                "guardduty:CreateFilter",
                "guardduty:CreateIPSet",
                "guardduty:CreateMembers",
                "guardduty:CreatePublishingDestination",
                "guardduty:CreateSampleFindings",
                "guardduty:CreateThreatIntelSet",
                "guardduty:DeclineInvitations",
                "guardduty:DeleteDetector",
                "guardduty:DeleteFilter",
                "guardduty:DeleteInvitations",
                "guardduty:DeleteIPSet",
                "guardduty:DeleteMembers",
                "guardduty:DeletePublishingDestination",
                "guardduty:DeleteThreatIntelSet",
                "guardduty:DisassociateFromMasterAccount",
                "guardduty:DisassociateMembers",
                "guardduty:InviteMembers",
                "guardduty:StartMonitoringMembers",
                "guardduty:StopMonitoringMembers",
                "guardduty:TagResource",
                "guardduty:UnarchiveFindings",
                "guardduty:UntagResource",
                "guardduty:UpdateDetector",
                "guardduty:UpdateFilter",
                "guardduty:UpdateFindingsFeedback",
                "guardduty:UpdateIPSet",
                "guardduty:UpdatePublishingDestination",
                "guardduty:UpdateThreatIntelSet"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

の SCPsの例 AWS Resource Access Manager

このカテゴリの例

- 外部共有の禁止
- 特定のアカウントが、指定したリソースタイプのみを共有できるようにする
- 組織または組織単位 (OU) との共有を禁止する
- 指定した IAM ユーザーおよびロールのみとの共有を許可する

外部共有の禁止

以下の SCP の例では、組織の一部ではない IAM ユーザーおよびロールとの共有を許可するリソース共有を、ユーザーが作成できないようにします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ram:CreateResourceShare",
                "ram:UpdateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "ram:RequestedAllowsExternalPrincipals": "true"
                }
            }
        }
    ]
}
```

特定のアカウントが、指定したリソースタイプのみを共有できるようにする

以下の SCP では、アカウント 111111111111 と 222222222222 が、プレフィックスリストを共有するリソース共有を作成し、プレフィックスリストを既存のリソース共有に関連付けることができます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "OnlyNamedAccountsCanSharePrefixLists",
            "Effect": "Allow",
            "Action": [
                "ram:AssociateResourceShare",
                "ram:CreateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:PrincipalAccount": [
                        "111111111111",
                        "222222222222"
                    ]
                },
                "StringEquals": {
                    "ram:RequestedResourceType": "ec2:PrefixList"
                }
            }
        }
    ]
}
```

組織または組織単位 (OU) との共有を禁止する

次の SCP は、ユーザーが AWS Organization または OUs。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ram:CreateResourceShare",
```

```
                "ram:AssociateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "ram:Principal": [
                        "arn:aws:organizations::*:organization/*",
                        "arn:aws:organizations::*:ou/*"
                    ]
                }
            }
        }
    ]
}
```

指定した IAM ユーザーおよびロールのみとの共有を許可する

以下の SCP の例では、ユーザーは組織 o-12345abcdef、組織単位 ou-98765fedcba、およびアカウント 111111111111 のみとリソースを共有できます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ram:AssociateResourceShare",
                "ram:CreateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "ram:Principal": [
                        "arn:aws:organizations::123456789012:organization/
o-12345abcdef",
                        "arn:aws:organizations::123456789012:ou/o-12345abcdef/
ou-98765fedcba",
                        "111111111111"
                    ]
                }
            }
        }
    ]
```

```
}
```

Amazon Route 53 Application Recovery Controller の SCP 例

このカテゴリの例

- ユーザーが Route 53 ARC ルーティング制御状態を更新できないようにする

ユーザーが Route 53 ARC ルーティング制御状態を更新できないようにする

下位レベルの Route 53 ARC オペレーターは、ダッシュボードをモニタリングして Route 53 ARC 情報を確認する必要があります。ただし、上級オペレーターが許可されているときと同様に、オペレーターはルーティング制御を更新し、1 つの AWS リージョン から別のものでアプリケーションをフェイルオーバーできないようにする必要があります。この SCP は、影響されたアカウントのユーザーまたはロールが、Route 53 ARC ルーティング制御を更新する Route 53 ARC 操作の実行を防止します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyAll",
            "Effect": "Deny",
            "Action": [
                "route53-recovery-cluster:UpdateRoutingControlState",
                "route53-recovery-cluster:UpdateRoutingControlStates"
            ],
            "Resource": "*",
            "Condition": {
                "ArnNotLike": {
                    "aws:PrincipalARN": [
                        "arn:aws:iam::*:role/Role1AllowedToBypassThisSCP",
                        "arn:aws:iam::*:role/Role2AllowedToBypassThisSCP"
                    ]
                }
            }
        }
    ]
}
```

Amazon S3 の SCP の例

> **Note**
>
> Amazon Simple Storage Service (Amazon S3) は、別の暗号化オプションを指定しない限り、新しいオブジェクトごとにサーバー側の暗号化 (SSE-S3) を自動的に適用します。詳細については、Amazon S3ユーザーガイド」の「Amazon S3 がすべての新しいオブジェクトを自動的に暗号化するようになりました」を参照してください。 Amazon S3

このカテゴリの例

- Amazon S3 の暗号化されていないオブジェクトのアップロードを禁止する

Amazon S3 の暗号化されていないオブジェクトのアップロードを禁止する

次のポリシーは、すべてのユーザーに対して、暗号化されていないオブジェクトを S3 バケットへのアップロードを制限します。

```
{
  "Effect": "Deny",
  "Action": "s3:PutObject",
  "Resource": "*",
  "Condition": {
    "Null": {
      "s3:x-amz-server-side-encryption": "true"
    }
  }
}
```

次のポリシーは、すべてのユーザーに対して、暗号化されていないオブジェクトを S3 バケットにアップロードすることを制限し、バケットへのオブジェクトのアップロードに指定された暗号化タイプ (AES256 または aws:kms) を適用します。

```
[
  {
    "Effect": "Deny",
    "Action": "s3:PutObject",
    "Resource": "*",
    "Condition": {
      "Null": {
```

```
        "s3:x-amz-server-side-encryption": "true"
      }
    }
  },
  {
    "Effect": "Deny",
    "Action": "s3:PutObject",
    "Resource": "*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "AES256"
      }
    }
  }
]
```

リソースのタグ付けの SCP の例

このカテゴリの例

- 作成される特定のリソースにタグを要求する
- 許可されたプリンシパル以外のタグが変更されないようにする

作成される特定のリソースにタグを要求する

次の SCP では、リクエストに指定されたタグが含まれていない場合、影響を受けるアカウントの IAM ユーザーとロールが特定のリソースタイプを作成できないようにします。

> **Important**
>
> お客様の環境で使用するサービスを使用して、拒否ベースのポリシーを必ずテストしてください。以下の例は、タグ付けされていないシークレットを作成したり、タグ付けされていない Amazon EC2 インスタンスを実行したりする単純なブロックであり、例外は含まれていません。
>
> 次のポリシーの例は、そのままでは AWS CloudFormation との互換性がありません。これは、そのサービスがシークレットを作成し、それを 2 つの個別のステップとしてタグ付けするためです。このポリシーの例では、AWS CloudFormation がスタックの一部としてシークレットを作成することを効果的にブロックします。それらのアクションは、短期間であっても要求されたとおりにタグ付けされていないシークレットとなるためです。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCreateSecretWithNoProjectTag",
      "Effect": "Deny",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    },
    {
      "Sid": "DenyRunInstanceWithNoProjectTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    },
    {
      "Sid": "DenyCreateSecretWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/CostCenter": "true"
        }
      }
    },
    {
      "Sid": "DenyRunInstanceWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
```

```
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/CostCenter": "true"
        }
      }
    }
  ]
}
```

AWS Organizations SCP および IAM アクセス許可ポリシーをサポートするすべてのサービスとアクションのリストについては、IAM ユーザーガイドの「AWS のサービスのアクション、リソース、および条件キー」を参照してください。

許可されたプリンシパル以外のタグが変更されないようにする

次の SCP では、ポリシーによって、許可されたプリンシパルのみがリソースにアタッチされたタグを変更できるようにする方法を示します。これは、属性ベースのアクセスコントロール (ABAC) を AWS クラウドセキュリティ戦略の一環として使用する際の重要なポイントです。このポリシーでは、発信者は、認可タグ (この例では access-project)がリクエストを行ったユーザーまたはロールに付けられた認可タグと完全に一致するリソースのみでタグを修正することができます。また、このポリシーでは、許可されたユーザーが、認可に使用されるタグの値を変更することを防ぎます。呼び出し元のプリンシパルが変更を行うには、認可タグが必要になります。

このポリシーは、権限のないユーザーによるタグの変更のみをブロックします。このポリシーでブロックされずに認可されたユーザーには、関連する API のタグ付けの際に Allow アクセス許可を明示的に付与する別の IAM ポリシーが必要です。例えば、ユーザーが管理者ポリシーで Allow */* (すべてのサービスとすべてのオペレーションを許可) を設定している場合、この組み合わせによって、管理者ユーザーは、ユーザーのプリンシパルにアタッチされた認可タグと一致する認可タグの値を含むタグのみの変更が許可されることになります。これは、このポリシーの明示的な Deny が、管理者ポリシーの明示的な Allow よりも優先されるためです。

> **Important**
>
> これはポリシーによる完全な解決策ではないため、ここで説明したとおりに使用することはできません。この例は、ABAC 戦略の一部を説明することを目的としているため、本番環境向けにカスタマイズし、テストする必要があります。

その仕組みの詳細な分析を含む完全なポリシーについては、「AWS Organizations のサービスコントロールポリシーを使用して、認可に使用されるリソースのタグを保護する」を参照してください。
お客様の環境で使用するサービスを使用して、拒否ベースのポリシーを必ずテストしてください。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyModifyTagsIfResAuthzTagAndPrinTagDontMatch",
            "Effect": "Deny",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "ec2:ResourceTag/access-project": "${aws:PrincipalTag/access-
project}",
                    "aws:PrincipalArn": "arn:aws:iam::123456789012:role/org-admins/iam-
admin"
                },
                "Null": {
                    "ec2:ResourceTag/access-project": false
                }
            }
        },
        {
            "Sid": "DenyModifyResAuthzTagIfPrinTagDontMatch",
            "Effect": "Deny",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": [
                "*"
            ],
```

```
            "Condition": {
                "StringNotEquals": {
                    "aws:RequestTag/access-project": "${aws:PrincipalTag/access-
project}",
                    "aws:PrincipalArn": "arn:aws:iam::123456789012:role/org-admins/iam-
admin"
                },
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": [
                        "access-project"
                    ]
                }
            }
        },
        {
            "Sid": "DenyModifyTagsIfPrinTagNotExists",
            "Effect": "Deny",
            "Action": [
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringNotEquals": {
                    "aws:PrincipalArn": "arn:aws:iam::123456789012:role/org-admins/iam-
admin"
                },
                "Null": {
                    "aws:PrincipalTag/access-project": true
                }
            }
        }
    ]
}
```

Amazon Virtual Private Cloud (Amazon VPC) の SCP の例

このカテゴリの例

- ユーザーによる VPC フローログの削除を禁止する
- インターネットアクセスに接続されていない VPC を使用した取得を禁止する

ユーザーによる VPC フローログの削除を禁止する

この SCP では、影響を受けるアカウントのユーザーまたはロールは Amazon Elastic Compute Cloud (Amazon EC2) フローログ、CloudWatch ロググループまたはログストリームを削除できなくなります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:DeleteFlowLogs",
        "logs:DeleteLogGroup",
        "logs:DeleteLogStream"
      ],
      "Resource": "*"
    }
  ]
 }
```

インターネットアクセスに接続されていない VPC を使用した取得を禁止する

この SCP では、影響を受けるアカウントのユーザーまたはロールは、Amazon EC2 仮想プライベートクラウド (VPC) の設定を変更して、インターネットへの直接アクセスを許可しないようにします。既存の直接アクセスや、オンプレミスネットワーク環境を経由するアクセスはブロックされません。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ec2:AttachInternetGateway",
        "ec2:CreateInternetGateway",
        "ec2:CreateEgressOnlyInternetGateway",
        "ec2:CreateVpcPeeringConnection",
        "ec2:AcceptVpcPeeringConnection",
        "globalaccelerator:Create*",
        "globalaccelerator:Update*"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

組織単位の管理

組織単位 (OU) を使用すると、アカウントをまとめてグループ化し、単一の単位として管理できます。その結果、アカウントの管理は大幅に簡略化されます。たとえば、ポリシーベースの制御を OU に付与すると、OU 内のすべてのアカウントに自動的に OU ポリシーが継承されます。OU は、1 つの組織に複数作成することができるため、その他の OU 内にも OU を作成することができます。各 OU には、複数のアカウントを含めることができるだけでなく、OU から別の OU へアカウントを移動することもできます。ただし、OU の名前は、親 OU またはルート内で一意である必要があります。

> **Note**
>
> 組織内にはルートが 1 つあり、組織を最初にセットアップするときに AWS Organizations によって作成されます。

トピック

- ルートおよび OU 階層の操作
- OU の作成
- OU の名前変更
- OU にアタッチされたタグの編集
- アカウントの OU への移動と、ルートと OU 間の移動
- OU の削除

組織全体のすべての OU を確認することもできます。詳細については、「OU の詳細の表示」を参照してください。

ルートおよび OU 階層の操作

アカウントの移動またはポリシーのアタッチの際に、別の OU またはルートに移動するには、デフォルトの「ツリー」ビューを使用します。

AWS Management Console

「ツリー」ビューで組織内を移動するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウントページの [Organization] (組織) セクションの上部で [List] (リスト) ではなく、[Hierarchy] (階層) を選択します。
3. 初期状態のツリーにはルートと、階層の最初の子 OU および
アカウントのみが表示されます。ツリーを展開してより深い階
層を表示するには、親エンティティの横にある展開アイコン
(▶)
を選択します。表示をシンプルにするためにツリーのブランチを折りた
たむには、展開された親エンティティの横にある折りたたみアイコン
(▼)
を選択します。
4. 詳細の確認や、特定の操作を行うには、OU またはルートの名前を選択します。または、名前の横にあるラジオボタンを選択し、[Actions] (アクション) メニューからエンティティに対する特定の操作を行うこともできます。

組織内のアカウントのみの一覧を表形式ビューで表示することもでき、特定のアカウントを見つけるために逐一 OU に移動する必要はありません。このビューでは、OU の表示や、OU にアタッチされたポリシーの操作はできません。

AWS Management Console

アカウントのフラットリストとして階層なしで組織を表示するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、組織セクションの上部にある「切り替
え AWS アカウント のみを表示」アイコンを選択してオンにします。

3. アカウントの一覧が階層なしで表示されます。

OU の作成

組織の管理アカウントにサインインすると、組織のルートに OU を作成できます。OU は、最大 5 レベルの深さまでネストできます。OU を作成するには、次のステップを完了します。

> **Important**
>
> この組織が で管理されている場合は AWS Control Tower、 AWS Control Tower コンソールまたは API を使用して OUs を作成します。 APIs Organizations で OU を作成する場合、その OU は に登録されません AWS Control Tower。詳細については、AWS Control Tower ユーザーガイドの AWS Control Tower外のリソースを参照するを参照してください。

> **最小アクセス許可**
>
> 組織のルート内に OU を作成するには、次のアクセス権限が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:CreateOrganizationalUnit`

AWS Management Console

OU を作成するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページに移動します。

   ルート OU とその内容がコンソールに表示されます。初めてルートにアクセスするときは、コンソールの最上位のビューにすべての AWS アカウント が表示されます。以前に OU を作成してその OU にアカウントを移動している場合、コンソールには最上位の OU と、OU に移動していないアカウントのみ表示されます。
3. (オプション) 既存の OU 内に OU を作成する場合は、子 OU に移動します。移動するには、子 OU の名前 (チェックボックスではなく) を選択するか、ツリービューで OU の横の

を選択して目的の OU が表示されるまで展開し、目的の子 OU の名前を選択します。

4. 階層内の正しい親 OU を選択したら、[Actions] (アクション) メニューの [Organizational Unit] (組織単位) で [Create new] (新規作成) を選択します。
5. [Create organizational unit] (組織単位の作成) ダイアログボックスで、作成する OU の名前を入力します。
6. (オプション) [Add tag] (タグの追加) を選択してキーとオプションの値を入力し、1 つ以上のタグを追加します。値を空白のままにすると、空の文字列が設定され、null にはなりません。1 つの OU に最大 50 個のタグをアタッチできます。
7. 最後に、[Create organizational unit] (組織単位の作成) を選択します。

親 OU 内に新しい OU が作成されます。これで、この OU にアカウントを移動する、またはポリシーをアタッチすることができるようになりました。

AWS CLI & AWS SDKs

OU を作成するには

以下のコード例は、`CreateOrganizationalUnit` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates a new organizational unit in AWS Organizations.
/// </summary>
```

```
    public class CreateOrganizationalUnit
    {
        /// <summary>
        /// Initializes an Organizations client object and then uses it to call
        /// the CreateOrganizationalUnit method. If the call succeeds, it
        /// displays information about the new organizational unit.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var orgUnitName = "ProductDevelopmentUnit";

            var request = new CreateOrganizationalUnitRequest
            {
                Name = orgUnitName,
                ParentId = "r-0000",
            };

            var response = await client.CreateOrganizationalUnitAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully created organizational unit:
{orgUnitName}.");
                Console.WriteLine($"Organizational unit {orgUnitName} Details");
                Console.WriteLine($"ARN: {response.OrganizationalUnit.Arn} Id:
{response.OrganizationalUnit.Id}");
            }
            else
            {
                Console.WriteLine("Could not create new organizational unit.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ルート OU または親 OU に OU を作成するには

次の例は、AccountingOU という名前の OU を作成する方法を示しています。

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

出力には、新しい OU に関する詳細を含む organizationalUnit オブジェクトが含まれます。

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- API の詳細については、「 コマンドリファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS CLI

OU の名前変更

組織の管理アカウントにサインインすると、OU の名前を変更することができます。そのためには、以下の手順を完了します。

最小アクセス許可

AWS 組織内のルート内の OU の名前を変更するには、次のアクセス許可が必要です。

- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:UpdateOrganizationalUnit`

AWS Management Console

OU の名前を変更するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウントページで、名前を変更する OU に移動し、次のいずれかのステップを実施します。
   - 名前を変更する OU の横にあるラジオボタン
     
     を選択します。次に、[Actions] (アクション) メニューの [Organizational Unit] (組織単位) で、[Rename] (名前の変更) を選択します。
   - OU の名前を選択し、OU の詳細ページにアクセスします。ページの上部で、[Rename] (名前の変更) を選択します。
3. [Rename organizational unit] (組織単位名の変更) ダイアログボックスで新しい名前を入力し、[Save changes] (変更の保存) を選択します。

AWS CLI & AWS SDKs

OU の名前を変更するには

OU の名前を変更するには、次のいずれかのコマンドを使用します。

- AWS CLI: update-organizational-unit

  次の例は、OU の名前を変更する方法を示しています。

```
$ aws organizations update-organizational-unit \
    --organizational-unit-id ou-a1b2-f6g7h222 \
    --name "Renamed-OU"
{
    "OrganizationalUnit": {
        "Id": "ou-a1b2-f6g7h222",
        "Arn": "arn:aws:organizations::123456789012:ou/o-aa111bb222/ou-a1b2-
f6g7h222",
        "Name": "Renamed-OU"
    }
```

```
}
```

- AWS SDKs UpdateOrganizationalUnit

OU にアタッチされたタグの編集

組織の管理アカウントにサインインすると、OU にアタッチされるタグを追加または削除できます。そのためには、以下の手順を完了します。

> 最小アクセス許可
>
> AWS 組織内のルート内の OU にアタッチされたタグを編集するには、次のアクセス許可が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:DescribeOrganizationalUnit` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:TagResource`
> - `organizations:UntagResource`

AWS Management Console

OU にアタッチされたタグを編集するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、編集する OU に移動して名前を選択します。
3. OU の詳細ページで、[Tags] (タグ) タブを選択し、[Manage tags] (タグ管理) を選択します。
4. このタブでは次のアクションが実行可能です。
   - 古い値に上書きして新しい値を入力し、任意のタグの値を編集します。タグキーは変更できません。キーを変更するには、古いキーを持つタグを削除し、新しいキーを持つタグを追加する必要があります。
   - 削除するタグの横にある [Remove] (削除) を選択し、既存のタグを削除します。

- 新しいタグのキーと値のペアを追加します。[Add tag] (タグの追加) を選択し、表示されたボックスに新しいキー名とオプションの値を入力します。[Value] (値) ボックスを空白のままにすると、値は空の文字列に設定され、null にはなりません。

5. 必要な追加、削除、編集をすべて終えたら、[Save changes] (変更の保存) を選択します。

AWS CLI & AWS SDKs

OU にアタッチされたタグを編集するには

OU にアタッチされたタグを変更するには、次のいずれかのコマンドを使用します。

- AWS CLI: tag-resource および untag-resource

  次の例では、タグ "Department"="12345" を OU にアタッチしています。Key と Value では大文字と小文字が区別されます。

```
$ aws organizations tag-resource \
    --resource-id ou-a1b2-f6g7h222 \
    --tags Key=Department,Value=12345
```

  このコマンドが成功した場合、出力は生成されません。

  次の例では、Department タグを OU から削除しています。

```
$ aws organizations untag-resource \
    --resource-id ou-a1b2-f6g7h222 \
    --tag-keys Department
```

  このコマンドが成功した場合、出力は生成されません。

- AWS SDKsTagResourceおよび UntagResource

アカウントの OU への移動と、ルートと OU 間の移動

組織の管理アカウントにサインインすると、ルートと OU の間、または OU どうしの間で組織のアカウントを移動させることができます。OU 内にアカウントを配置すると、親 OU、およびその OU からルートまでの間にあるすべての OU にアタッチされているポリシーが適用されます。OU に属していないアカウントには、ルート、およびそのアカウントに直接アタッチされているポリシーのみが適用されます。アカウントを移動させるには、次のステップを実施します。

最小アクセス許可

OU 階層の新しい場所にアカウントを移動させるには、次のアクセス許可が必要です。

- `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
- `organizations:MoveAccount`

AWS Management Console

アカウントを OU に移動させるには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、移動させるアカウントを見つけます。OU の階層を移動するか、[View AWS アカウント only] (AWS アカウント だけを表示する) を有効にして OU 構造のないフラットリストでアカウントを表示します。アカウントが多い場合、削除対象をすべて見つけるにはリスト下部の [Load more accounts in 'ou-name'] ('ou-name' のアカウントをさらに読み込む) を選択する必要がある場合があります。
3. 移動させるアカウントの名前の横にあるチェックボックス
   ☑
   を選択します。
4. [Actions] (アクション) メニューの [AWS アカウント] で、[Move] (移動) を選択します。
5. [Move AWS アカウント] (AWS アカウントの移動) ダイアログボックスで、アカウントの移動先の OU またはルートを見つけて選択し、[Move AWS アカウント] (AWS アカウントの移動) を選択します。

AWS CLI & AWS SDKs

アカウントを OU に移動させるには

アカウントを移動するには、次のいずれかのコマンドを使用します。

- AWS CLI: move-account

次の例では、 をルート AWS アカウント から OU に移動します。ソースコンテナと宛先コンテナの両方の ID を指定する必要があります。

```
$ aws organizations move-account \
    --account-id 111122223333 \
    --source-parent-id r-a1b2 \
    --destination-parent-id ou-a1b2-f6g7h111
```

このコマンドが成功した場合、出力は生成されません。

- AWS SDKs MoveAccount

OU の削除

組織の管理アカウントにサインインすると、不要になった OU を削除できます。

子 OU を削除するには、まず OU とその子 OU 内のアカウントをすべて移動させる必要があります。

> **最小アクセス許可**
>
> OU を削除するには、次のアクセス権限が必要です。
>
> - `organizations:DescribeOrganization` - Organizations コンソールを使用する場合にのみ必要
> - `organizations:DeleteOrganizationalUnit`

AWS Management Console

OU を削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. AWS アカウント ページで、削除する OU を探し、その OU 名の横にあるチェックボックス ☑ をオンにします。

3. [Actions] (アクション) を選択し、[Organizational unit] (組織単位) で [Delete] (削除) を選択します。
4. その OU の削除を確定するには、OU の名前 (削除対象が 1 つだけの場合) または「delete」という文字列 (削除対象が複数ある場合) を入力してから、[Delete] (削除) を選択します。

   AWS Organizations は OUsし、リストから削除します。

AWS CLI & AWS SDKs

OU を削除するには

以下のコード例は、`CreateOrganizationalUnit` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates a new organizational unit in AWS Organizations.
/// </summary>
public class CreateOrganizationalUnit
{
    /// <summary>
    /// Initializes an Organizations client object and then uses it to call
    /// the CreateOrganizationalUnit method. If the call succeeds, it
    /// displays information about the new organizational unit.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
```

```
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var orgUnitName = "ProductDevelopmentUnit";

            var request = new CreateOrganizationalUnitRequest
            {
                Name = orgUnitName,
                ParentId = "r-0000",
            };

            var response = await client.CreateOrganizationalUnitAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully created organizational unit:
{orgUnitName}.");
                Console.WriteLine($"Organizational unit {orgUnitName} Details");
                Console.WriteLine($"ARN: {response.OrganizationalUnit.Arn} Id:
{response.OrganizationalUnit.Id}");
            }
            else
            {
                Console.WriteLine("Could not create new organizational unit.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ルート OU または親 OU に OU を作成するには

次の例は、AccountingOU という名前の OU を作成する方法を示しています。

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

出力には、新しい OU に関する詳細を含む organizationalUnit オブジェクトが含まれます。

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- API の詳細については、「 コマンドリファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS CLI

AWS Organizations リソースのタグ付け

タグは、AWS リソースに追加して、リソースの識別、整理、検索を容易にできるカスタム属性ラベルです。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`)。タグキーの長さは最大 128 文字で、大文字と小文字は区別されます。
- タグ値 (例: `111122223333` または `Production`)。タグ値の長さは最大 256 文字で、タグキーと同様に大文字と小文字は区別されます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。タグ値を省略すると、空の文字列を使用した場合と同じになります。

タグのキーまたは値で許可される文字の詳細についてはResource Groups のタグ付け API リファレンスの「タグ付け API の タグパラメータ」を参照してください。

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。詳細については、AWS「リソースのタグ付けのベストプラクティス」を参照してください。

> **Tip**
>
> 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の実装を標準化できます。

現時点で、AWS Organizations は、管理アカウントにログインしているときに次のタグ付けオペレーションをサポートします。

- 以下のタイプの組織のリソースにタグを追加できます。
  - AWS アカウント
  - 組織単位
  - 組織のルート
  - ポリシー

タグは次のタイミングで追加できます。

- リソースを作成するとき — Organizations コンソールでタグを指定するか、`Create` API オペレーションの 1 つで `Tags` パラメータを使用します。これは、組織のルートには適用されません。
- リソースを作成した後 — Organizations コンソールを使用するか、`TagResource` オペレーションを呼び出します。

コンソールを使用するか、`ListTagsForResource` オペレーションを呼び出すことで、AWS Organizations 内のタグ付け可能な任意のリソースのタグを表示できます。

リソースからタグを削除するには、コンソールを使用して削除するキーを指定するか、`UntagResource` オペレーションを呼び出します。

タグの使用

タグを使用すると、役に立つカテゴリ別にリソースをグループ化でき、組織内のリソースリソースを整理できます。例えば、所有部門を追跡する「部門」タグを割り当てることができます。「Environment」タグを割り当てると、特定のリソースがアルファ、ベータ、ガンマ、または本番環境の一部であるかどうかを追跡できます。

タグを使用して以下のこともできます。

- リソースにタグ付け基準を適用します。
- 誰がリソースにアクセスできるかを制御します。

タグの追加、更新、削除

組織の管理アカウントにサインインすると、組織のリソースにタグを追加できます。

リソースの作成時にタグを追加する

> **最小アクセス許可**
>
> リソースの作成時にリソースにタグを追加するには、次のアクセス許可が必要です。
>
> - 特定のタイプのリソースを作成するためのアクセス許可
> - `organizations:TagResource`
> - `organizations:ListTagsForResource` - Organizations コンソールを使用する場合にのみ必要

作成時に、以下のリソースにアタッチされたタグキーと値を含めることができます。

- AWS アカウント
  - 作成したアカウント
  - 招待されたアカウント
- 組織単位 (OU)
- ポリシー
  - AI サービスのオプトアウトポリシー
  - バックアップポリシー
  - サービスコントロールポリシー
  - タグポリシー

組織ルートは、最初に組織を作成する際に作成されるため、既存のリソースとしてのみタグを追加できます。

既存のリソースにタグを追加または更新する

新しいタグを追加したり、既存のリソースにアタッチされているタグの値を更新することもできます。

最小アクセス許可

組織のリソースにタグを追加または更新するには、次のアクセス許可が必要です。

- `organizations:TagResource`
- `organizations:ListTagsForResource` - Organizations コンソールを使用する場合にのみ必要

組織のリソースからタグを削除するには、次のアクセス許可が必要です。

- `organizations:UntagResource`

AWS Management Console

既存のリソースのタグを追加、更新、または削除するには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. アカウント、Root、OU、またはポリシーに移動して選択し、名前をクリックして詳細ページを開きます。
3. [Tags (タグ)] タブで、[Manage tags (タグ管理)] を選択します。
4. 新しいタグの追加、既存のタグの値の変更、またはタグの削除ができます。

   (オプション) タグを追加するには、[タグを追加] を選択してキーを入力し、オプションでタグの値を入力します。

   タグを削除するには、[削除] を選択します。

   タグのキーと値では、大文字と小文字が区別されます。大文字と小文字の区別は標準化して使用します。適用されるタグポリシーの要件を順守することも必要です。
5. 前のステップを必要な回数繰り返します。
6. [変更の保存] をクリックします。

AWS CLI & AWS SDKs

既存のリソースにタグを追加または更新するには

組織のタグ付け可能なリソースにタグを追加するには、次のいずれかのコマンドを使用します。

- AWS CLI: タグリソース
- AWS SDKs TagResource

組織のリソースからタグを削除するには

タグを削除するには、次のいずれかのコマンドを使用します。

- AWS CLI: タグなしリソース
- AWS SDKs UntagResource

AWS Organizations を他のAWSサービスと併用する

信頼されたアクセスを使用して、指定したサポートされている AWS サービス (信頼されたサービスと呼ばれる) を有効にできます。これにより、組織とそのアカウントのタスクを代理で実行できるようになります。これには、信頼されたサービスに許可を付与する必要がありますが、ユーザーまたはロールの許可に影響はありません。アクセスを有効にすると、信頼されたサービスは、組織の各アカウントにサービスにリンクされたロールと呼ばれる IAM ロール を必要なときに作成できるようになります。このロールには、信頼されたサービスを使用して、該当サービスのドキュメントに記載されているタスクの実行を可能にするアクセス許可ポリシーが含まれています。これにより、信頼されたサービスを使用して、ユーザーに代わって組織のアカウントで管理する設定や構成の詳細を指定できます。信頼されたサービスは、アカウントに対して管理アクションを実行する必要がある場合にのみ、サービスにリンクされたロールを作成します。必ずしも組織のすべてのアカウントで管理アクションを実行する必要はありません。

Important

選択肢がある場合、信頼されたアクセスの有効化と無効化には、信頼されたサービスのコンソールか、その AWS CLI または API のオペレーションのみを使用することを強くおすすめします。これにより、信頼できるアクセスの有効化に必要なすべての初期化処理が信頼できるサービスによって実行可能になります。例えば、必要なリソースの作成や、信頼できるアクセスの無効にする際のリソースのクリーンアップなどです。
信頼されたサービスを使用し、信頼されたサービスによる組織へのアクセスを有効または無効にする方法については、AWS で使用できる のサービス AWS Organizations の [Supports Trusted Access] (信頼されたアクセスをサポート) 列の [Learn more] (詳細はこちら) リンクを参照してください。
Organizations コンソール、CLI コマンド、API オペレーションを使用してアクセスを無効にした場合の結果は次のようになります。

- そのサービスでは、サービスにリンクされたロールを組織のアカウントに作成できなくなります。つまり、組織の新しいアカウントに対するオペレーションをサービスがユーザーに代わって実行できなくなります。そのサービスによる AWS Organizations のクリーンアップが完了するまでは、古いアカウントに対するオペレーションは引き続き実行可能です。
- そのサービスでは、ロールにアタッチされている IAM ポリシーによって明示的に許可されていない限り、組織のメンバーアカウントのタスクを実行できなくなります。これには、

メンバーアカウントから管理アカウントまたは委任管理者アカウント (該当する場合) へのデータ集約が含まれます。

- 一部のサービスはこれを検出し、統合に関連する残りのデータやリソースをクリーンアップします。一方、組織へのアクセスを停止するものの、統合を再び有効にする場合のために履歴データと設定を残しておくサービスもあります。

そうしたサービスであっても、コンソールまたはコマンドを使用して統合を無効にすると、その統合以外に用途のないリソースがクリーンアップされるようになります。組織のアカウントのリソースをクリーンアップする仕組みは、サービスによって異なります。詳しくは、AWS の他のサービスのドキュメントを参照してください。

信頼されたアクセスを有効にするために必要なアクセス許可

信頼されたアクセスを使用するには、2 つのサービス (AWS Organizations と信頼されたサービス) のアクセス権限が必要です。信頼されたアクセスを有効にするには、次のいずれかのシナリオを選択します。

- AWS Organizations と信頼されたサービスの両方にアクセス許可がある認証情報が設定されている場合、信頼されたサービスによって提供されているツール (コンソールまたは AWS CLI) を使用してアクセスを有効化します。これにより、そのサービスには、ユーザーに代わって AWS Organizations で信頼されたアクセスを有効にすること、および組織でのオペレーションに必要なリソースを作成することが可能になります。

  これらの認証情報に必要な最小限のアクセス権限は次のとおりです。

  - `organizations:EnableAWSServiceAccess`。また、このオペレーションに `organizations:ServicePrincipal` 条件キーを使用し、承認されたサービスプリンシパル名のリストに対してオペレーションが行うリクエストを制限することもできます。詳細については、「条件キー」を参照してください。
  - `organizations:ListAWSServiceAccessForOrganization` - AWS Organizations コンソールを使用する場合は必須。
  - 信頼されたサービスで必要な最小限のアクセス権限は、サービスによって異なります。詳細については、信頼されたサービスのドキュメントを参照してください。
- AWS Organizations のアクセス権限を含む認証情報が設定されているユーザーと、信頼されたサービスのアクセス権限を含む認証情報が設定されているユーザーがいる場合は、以下のステップをこの順序で実行します。

1. AWS Organizations のアクセス権限を含む認証情報がユーザーに設定されている場合は、AWS Organizations コンソール、AWS CLI、または AWS SDK を使用して、信頼されたサービスの信頼されたアクセスを有効にします。これにより、次のステップ (ステップ 2) を実行すると、組織の必要な設定を実行するためのアクセス権限が他のサービスに付与されます。

   AWS Organizations の最小限のアクセス権限は次のとおりです。

   - `organizations:EnableAWSServiceAccess`
   - `organizations:ListAWSServiceAccessForOrganization` - AWS Organizations コンソールを使用する場合のみ必須

   AWS Organizations で信頼されたアクセスを有効にする特定のステップについては、「信頼されたアクセスを有効または無効にする方法」を参照してください。
2. 信頼されたサービスのアクセス許可を含む認証情報をユーザーに設定すると、そのサービスで AWS Organizations を操作できます。これにより、信頼されたサービスを使用して、組織で操作するために必要なリソースの作成など、必要な初期化を行うようサービスに指示されます。詳細については、サービス固有の手順 (AWS で使用できる のサービス AWS Organizations) を参照してください。

信頼されたアクセスを無効にするために必要なアクセス許可

信頼されたサービスを使用して、組織またはそのアカウントで操作する必要がなくなった場合は、次のいずれかのシナリオを選択します。

> **Important**
>
> サービスへの信頼されたアクセスを無効にすると、適切なアクセス権限を含むユーザーやロールは、そのサービスを使用できなくなります。ユーザーとロールが AWS サービスにアクセスするのを完全にブロックするには、そのアクセスを許可する IAM アクセス許可を削除するか、AWS Organizations でサービス コントロール ポリシー (SCP) を使用します。SCP はメンバーアカウントにのみ適用できます。SCP は管理アカウントには適用されません。管理アカウントではサービスを実行しないことをお勧めします。代わりに、SCP を使用してセキュリティを制御できるメンバーアカウントで実行します。

- AWS Organizations と信頼されたサービスの両方にアクセス許可がある認証情報が設定されている場合には、信頼されたサービスで利用できるツール (コンソールまたは AWS CLI) を使用してアクセスを無効化します。無効になると、サービスは、ユーザーの代わりに、不要になったリソース

を削除し、AWS Organizations のサービスの信頼されたアクセスを無効にしてクリーンアップします。

これらの認証情報に必要な最小限のアクセス権限は次のとおりです。

- `organizations:DisableAWSServiceAccess`。また、このオペレーションに `organizations:ServicePrincipal` 条件キーを使用し、承認されたサービスプリンシパル名のリストに対してオペレーションが行うリクエストを制限することもできます。詳細については、「条件キー」を参照してください。
- `organizations:ListAWSServiceAccessForOrganization` - AWS Organizations コンソールを使用する場合は必須。
- 信頼されたサービスで必要な最小限のアクセス権限は、サービスによって異なります。詳細については、信頼されたサービスのドキュメントを参照してください。

- AWS Organizations のアクセス権限を含む認証情報と、信頼されたサービスのアクセス権限を含む認証情報が異なる場合は、以下のステップをこの順序で実行します。

  1. まず、信頼されたサービスのアクセス権限を含むユーザーを使用して、このサービスを使用するアクセスを無効にします。これにより、信頼されたアクセスに必要なリソースを削除してクリーンアップするよう、信頼されたサービスに指示されます。詳細については、サービス固有の手順 (AWS で使用できる のサービス AWS Organizations) を参照してください。
  2. これで、AWS Organizations のアクセス権限を含むユーザーは、AWS Organizations コンソール、AWS CLI、または AWS SDK を使用して、信頼されたサービスのアクセスを無効にできるようになります。これにより、信頼されたサービスのアクセス許可は、組織やそのアカウントより削除されます。

     AWS Organizations の最小限のアクセス権限は次のとおりです。

     - `organizations:DisableAWSServiceAccess`
     - `organizations:ListAWSServiceAccessForOrganization` - AWS Organizations コンソールを使用する場合のみ必須

     AWS Organizations で信頼されたアクセスを無効にする特定のステップについては、信頼されたアクセスを有効または無効にする方法を参照してください。

信頼されたアクセスを有効または無効にする方法

AWS Organizations のアクセス許可のみ付与されており、他の AWS サービスの管理者の代わりに、組織への信頼されたアクセスを有効または無効にする場合は、次の手順を使用します。

Important

選択肢がある場合、信頼されたアクセスの有効化と無効化には、信頼されたサービスのコンソールか、その AWS CLI または API のオペレーションのみを使用することを強くおすすめします。これにより、信頼できるアクセスの有効化に必要なすべての初期化処理が信頼できるサービスによって実行可能になります。例えば、必要なリソースの作成や、信頼できるアクセスの無効にする際のリソースのクリーンアップなどです。
信頼されたサービスを使用し、信頼されたサービスによる組織へのアクセスを有効または無効にする方法については、AWS で使用できる のサービス AWS Organizations の [Supports Trusted Access] (信頼されたアクセスをサポート) 列の [Learn more] (詳細はこちら) リンクを参照してください。
Organizations コンソール、CLI コマンド、API オペレーションを使用してアクセスを無効にした場合の結果は次のようになります。

- そのサービスでは、サービスにリンクされたロールを組織のアカウントに作成できなくなります。つまり、組織の新しいアカウントに対するオペレーションをサービスがユーザーに代わって実行できなくなります。そのサービスによる AWS Organizations のクリーンアップが完了するまでは、古いアカウントに対するオペレーションは引き続き実行可能です。
- そのサービスでは、ロールにアタッチされている IAM ポリシーによって明示的に許可されていない限り、組織のメンバーアカウントのタスクを実行できなくなります。これには、メンバーアカウントから管理アカウントまたは委任管理者アカウント (該当する場合) へのデータ集約が含まれます。
- 一部のサービスはこれを検出し、統合に関連する残りのデータやリソースをクリーンアップします。一方、組織へのアクセスを停止するものの、統合を再び有効にする場合のために履歴データと設定を残しておくサービスもあります。

そうしたサービスであっても、コンソールまたはコマンドを使用して統合を無効にすると、その統合以外に用途のないリソースがクリーンアップされるようになります。組織のアカウントのリソースをクリーンアップする仕組みは、サービスによって異なります。詳しくは、AWS の他のサービスのドキュメントを参照してください。

AWS Management Console

信頼されたサービスのアクセスを有効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. サービスページで、有効にするサービスの行を探し、その名前を選択します。
3. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
4. 確認ダイアログボックスで、[Show the option to enable trusted access] (信頼されたアクセスを有効にするオプションを表示する) チェックボックスをオンにし、ボックスに「**enable**」と入力してから、[Enable trusted access] (信頼されたアクセスを有効にする) を選択します。
5. アクセスを有効にする場合は、他のサービスで AWS Organizations を操作できるようになったことを他の AWS サービスの管理者に伝えます。

信頼されたサービスのアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする (推奨されません) 必要があります。
2. サービスページで、無効にするサービスの行を探し、その名前を選択します。
3. もう一方のサービスの管理者から、サービスが無効になり、そのリソースのクリーンアップが完了したことが知らされるまで待ちます。
4. 確認ダイアログボックスで、ボックスに「**disable**」と入力してから、[Disable trusted access] (信頼されたアクセスを無効にする) を選択します。

AWS CLI, AWS API

信頼されたサービスのアクセスを有効または無効にするには

サービスへの信頼されたアクセスを有効または無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: AWS organizations enable-aws-service-access
- AWS CLI: AWS organizations disable-aws-service-access

- AWS API: EnableAWSServiceAccess
- AWS API: DisableAWSServiceAccess

AWS Organizations とサービスにリンクされたロール

AWS Organizations では、サービスにリンクされた IAM ロールを使用して、信頼されたサービスが組織のメンバーアカウントで自動的にタスクを実行できるようにします。信頼されたサービスを設定して、組織との統合のためにそのサービスを承認すると、サービスにリンクされたロールをメンバーアカウントに作成するようにそのサービスから AWS Organizations にリクエストできます。信頼されたサービスによって必要に応じて非同期的に行われますが、組織のすべてのアカウントで必ずしも同時に必要とは限りません。サービスにリンクされたロールには IAM 許可が事前定義されており、信頼されたサービスは、そのアカウント内の特定のタスクだけを実行する許可が与えられます。一般的に、サービスにリンクされたロールはすべて AWS によって管理されます。つまり、通常、ロールまたはアタッチされたポリシーを変更することはできません。

こうした変更を行えるようにするため、組織内にアカウントを作成するとき、または組織への既存のアカウントの招待が承諾されたときに、AWS Organizations は、サービスにリンクされたロール (`AWSServiceRoleForOrganizations`) を使用してメンバーアカウントをプロビジョニングします。このロールは、AWS Organizations サービス自体のみ引き受けることができます。また、このロールに含まれるアクセス許可により、AWS Organizations は、サービスにリンクされたロールを他の AWS サービス用にも作成できるようになります。このサービスにリンクされたロールは、すべての組織に存在します。

組織で一括請求機能のみ有効になっている場合、サービスにリンクされたロール (`AWSServiceRoleForOrganizations`) は使用されないため、削除できます。ただし、推奨はされません。組織のすべての機能を後に有効にする場合はこのロールが必要になるため、復元する必要があります。次のチェックは、すべての機能を有効にするプロセスを開始するときに実行されます。

- 組織に参加するように招待された各メンバーアカウント - アカウント管理者には、すべての機能を有効にすることへの同意を求めるリクエストが送信されます。サービスにリンクされたロール (`AWSServiceRoleForOrganizations`) が存在しない場合に適切にリクエストに同意するには、`organizations:AcceptHandshake` 許可および `iam:CreateServiceLinkedRole` 許可の両方がアカウント管理者に必要です。`AWSServiceRoleForOrganizations` ロールが既に存在する場合、管理者がリクエストに同意するには、`organizations:AcceptHandshake` アクセス権限のみが管理者に必要です。管理者がリクエストに同意すると、サービスにリンクされたロールが存在しない場合でも、AWS Organizations によって自動的に作成されます。

- 組織に作成された各メンバーアカウント - サービスにリンクされたロールの再作成リクエストがアカウント管理者に送信されます。(メンバーアカウントの管理者には、すべての機能を有効にするリクエストが届きません。これは、管理アカウント (旧称は「マスターアカウント」) の管理者が、作成されたメンバーアカウントの所有者と見なされるためです)。サービスにリンクされたロールは、メンバーアカウント管理者がリクエストに同意すると AWS Organizations によって作成されます。ハンドシェイクを適切に承諾するには、`organizations:AcceptHandshake` アクセス権限と `iam:CreateServiceLinkedRole` アクセス権限の両方が管理者に必要です。

組織内のすべての機能を有効にすると、サービスにリンクされたロール `AWSServiceRoleForOrganizations` はどのアカウントからも削除できなくなります。

> **Important**
>
> AWS Organizations の SCP がサービスにリンクされたロールに影響を及ぼすことはありません。これらのロールは、SCP 制限の対象外であるためです。

AWS で使用できる のサービス AWS Organizations

を使用すると、複数の を 1 つの組織に統合することで、アカウント管理アクティビティを大規模 AWS アカウント に実行 AWS Organizations できます。アカウントを統合すると、他の AWS のサービスの使用方法が簡素化されます。で利用可能なマルチアカウント管理サービスを一部の AWS サービス AWS Organizations で活用して、組織のメンバーであるすべてのアカウントでタスクを実行できます。

次の表 AWS に、 で使用できるサービスと AWS Organizations、組織全体レベルで各サービスを使用する利点を示します。

信頼されたアクセス – 互換性のある AWS サービスを有効にして、組織内のすべての AWS アカウント でオペレーションを実行できます。詳細については、「AWS Organizations を他のAWSサービスと併用する」を参照してください。

AWS サービスの委任管理者 – 互換性のある AWS サービスは、組織内の AWS メンバーアカウントを、そのサービス内の組織のアカウントの管理者として登録できます。詳細については、「Organizations と連携する AWS サービスの委任管理者」を参照してください。

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Account Management AWS アカウント 組織のすべての の詳細とメタデータを管理します。	組織内のすべてのアカウントの代替連絡先情報を作成、更新、削除できます。	 はい 詳細はこちら	 はい 詳細はこちら	
AWS Application Migration Service AWS Application Migration Service では、互換性の問題、パフォーマンスの中断、または長いカットオーバー期間なしに、AWS 多数の物理サーバー、仮想サーバー、またはクラウドサーバー lift-and-	複数のアカウントにわたる、大規模な移行を管理できます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
shift に接続できます。				
AWS Artifact ISO レポートや PCI レポートなどの AWS セキュリティコンプライアンスレポートをダウンロードします。	契約を組織内のすべてのアカウントに代わって受諾できます。	 はい 詳細はこちら	 いえ	いい
AWS Audit Manager クラウドサービスの使用の監査に役立てられるよう、エビデンスの継続的な収集を自動化します。	組織内の複数のアカウントで AWS 使用を継続的に監査し、リスクとコンプライアンスの評価方法を簡素化します。	 はい 詳細はこちら	 はい 詳細はこちら	はい

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Backup 組織内のすべてのアカウントのバックアップを管理およびモニタリングします。	組織全体、または組織単位 (OU) におけるアカウントのグループのバックアッププランを設定および管理できます。すべてのアカウントのバックアップを一元的にモニタリングできます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Billing and Cost Management AWS クラウド財務管理データの概要と、より迅速かつ情報に基づいた意思決定を行うのに役立ちます。	該当する場合は、分割コスト配分データで AWS Organizations 情報を取得し、オプトインした分割コスト配分データサービスのテレメトリデータを収集できるようにします。 詳細については、請求情報とコスト管理ユーザーガイドの「 AWS Billing	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	and Cost Management とは」を参照してください。			
AWS CloudFormation スタック・セット 1 回の操作で、複数のアカウントとリージョンにまたがるスタックを作成、更新、または削除できます。	管理アカウントまたは委任管理者アカウントのユーザーは、組織内のアカウントにスタックインスタンスを配備する、サービスによって管理されたアクセス許可を持つスタックセットを作成できます。	 はい 詳細はこちら	 はい 詳細はこちら	は

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS CloudTrail アカウントのガバナンス、コンプライアンス、および運用とリスクの監査を実行できます。	管理アカウントまたは委任管理者アカウントのユーザーは、組織のアカウントに関するすべてのイベントを口グに記録するイベントデータストアまたは組織の証跡を作成できます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Compute Optimizer AWS コンピューティング最適化の推奨事項を取得します。	組織のアカウントにあるすべてのリソースを分析して、最適化の推奨事項を取得できます。 詳細については、AWS Compute Optimizer ユーザーガイドの Compute Optimizer によってサポートされるアカウントを参照してください。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Config AWS リソースの設定を診断、監査、評価します。	コンプライアンスステータスに関する組織全体のビューを取得できます。AWS Config API オペレーションを使用して、組織内のすべての AWS アカウントで AWS Config ルールとコンフォーマンスパックを管理することもできます。	 はい 詳細はこちら	 はい 詳細はこちら: 設定 ルール コンフォーマンスパック マルチアカウント、マルチリージョンのデータ集計	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	委任管理者アカウントを使用し、 AWS Organizationsの組織のメンバーアカウント全体を対象に、リソース構成とコンプライアンス・データを集約できます。詳細については、 AWS Config デベロッパーガイドの委任された管理者の登録を参照			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	してください。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Control Tower 基準に準拠した安全な AWS マルチアカウント環境をセットアップして管理します。	ランディングゾーンは、すべての AWS リソースのマルチアカウント環境として設定できます。この環境には、組織および組織の エンティーティが含まれています。この環境を使用して、すべての にコンプライアンス規制を適用できます AWS アカウント。	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	詳細については、AWS Control Tower ユーザーガイドのAWS Control Towerの仕組みおよびAWS Organizations によるアカウントの管理を参照してください。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Cost Optimization Hub AWS 最適化製品全体でコストに関する推奨事項を収集します。	AWS Organizations メンバーアカウントと AWS リージョン全体で、AWS コスト最適化のレコメンデーションを簡単に特定、フィルタリング、集計できます。 詳細については、Cost Optimization Hub ユーザーガイドの「Cost Optimization Hub」を参照し	 はい 詳細はこちらa	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート
	てください。		
Amazon Detective ログデータから可視化を生成して、セキュリティに関する検出結果や疑わしいアクティビティの根本原因の分析、調査、および迅速な特定を行います。	Amazon Detective を と統合 AWS Organizations して、Detective 動作グラフがすべての組織アカウントのアクティビティを可視化できるようにします。	 はい 詳細はこちら	 はい 詳細はこちら

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon DevOpsGuru 運用データとアプリケーションのメトリクスおよびイベントを分析し、通常の運用パターンから逸脱する動作を特定します。DevOpsGuru が運用上の問題またはリスクを検出すると、ユーザーに通知されます。	と統合 AWS Organizations して、組織全体のすべてのアカウントからのインサイトを管理できます。管理者を委任すると、すべてのアカウントから取得したインサイトを表示、ソート、およびフィルタリングし、すべての監視対象アプリケーションの組織全体	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	の健全性を取得できます。			
AWS Directory Service AWS クラウドでディレクトリを設定して実行するか、既存のオンプレミス Microsoft Active Directory にリソースを接続します AWS 。	AWS Directory Service を と統合 AWS Organizations すると、リージョン内の複数のアカウントと任意の VPC 間でディレクトリをシームレスに共有できます。	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon EventBridge AWS リソースと で実行しているアプリケーションを AWS リアルタイムでモニタリングします。	組織内のすべてのアカウントで、以前の Amazon Events であるすべての Amazon CloudWatch EventBridge イベントの共有を有効にできます。 詳細については、「Amaz ユーザーガイド」の「 間での Amazon EventBridge イベントの送受信 AWS	 いいえ	 いいえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	アカウント」を参照してください。EventBridge			
AWS Firewall Manager アカウントやアプリケーション間でウェブアプリケーションのファイアウォールルールを一元的に設定、管理します。	組織内のアカウント全体で AWS WAF ルールを一元的に設定および管理できます。	 はい 詳細はこちら	 はい 詳細はこちら	は

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon GuardDuty GuardDuty は、さまざまなデータソースからの情報を分析および処理する継続的なセキュリティモニタリングサービスです。脅威インテリジェンスフィードおよび機械学習を使用して、AWS 環境内の予期しないアクティビティ、不正の可能性があるアクティビティ、悪意のあるアクティビティを識別します。	組織内のすべてのアカウント GuardDuty について、表示および管理するメンバーアカウントを指定できます。メンバーアカウントを追加すると、選択した のそれらのアカウント GuardDuty に対して が自動的に有効になります AWS リージョン。組織に追加さ	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	れた新しいアカウントの GuardDuty アクティベーションを自動化することもできます。 詳細については、「Amaz GuardDuty ユーザーガイド」のGuardDut「」および「組織」を参照してください。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Health AWS サービスのリソースパフォーマンスや可用性の問題に影響を与える可能性のあるイベントを可視化します。	組織内のアカウント間で AWS Health イベントを集約できます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Identity and Access Management AWS リソースへのアクセスを安全に制御します。	IAM のサービスの最後にアクセスされたデータを使用することで、組織全体の AWS アクティビティをより詳しく把握できます。このデータを使用してサービスコントロールポリシー (SCP) を作成、更新し、組織のアカウントが使用する AWS サービスだけにアクセ	 い いえ	 い いえ	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	スを限定することができます。 設定例については、IAM ユーザーガイドの情報を使用した組織単位のアクセス許可の調整を参照してください。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
IAM Access Analyzer AWS 環境内のリソースベースのポリシーを分析して、信頼ゾーン外のプリンシパルにアクセスを許可するポリシーを特定します。	IAM Access Analyzer の管理者には、メンバーアカウントを指定できます。 詳細については、IAM ユーザーガイド の Access Analyzer の有効化を参照してください。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon Inspector AWS ワークロードの脆弱性を自動的にスキャンして、Amazon ECR に存在する Amazon EC2 インスタンスとコンテナイメージを検出し、ソフトウェアの脆弱性や意図しないネットワークへの露出を検出します。	管理者を委任することで、メンバーアカウントのスキャンの有効化または無効化、組織全体の集約された調査結果データの表示、抑制ルールの作成および管理などが可能になります。 詳細については、「Amazon Inspector ユーザーガイド」の「AWS Organizat	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	ionsで複数のアカウントを管理する」を参照してください。			
AWS License Manager ソフトウェアライセンスをクラウドに移動するプロセスを効率化します。	組織全体でコンピューティングリソースのクロスアカウントの検出を有効にすることができます。	 はい 詳細はこちら	 はい 詳細はこちら	は

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon Macie 機械学習を使用してビジネスクリティカルなコンテンツを検出および分類し、データのセキュリティとプライバシーの要件を満たすのに役立ちます。Amazon S3 に保存されているコンテンツを継続的に評価し、潜在的な問題があれば通知します。	組織内のすべてのアカウントに対して Amazon Macie を設定し、指定された Macie 管理者アカウントのすべてのアカウントで、Amazon S3 のすべてのデータを一括表示することができます。組織の成長に合わせて新しいアカウントのリソースを自動的に保護す	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	るように Macie を設定できます。組織全体の S3 バケット間でポリシーの設定ミスを修正するよう求めるアラートが生成されます。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート
AWS Marketplace 厳選されたデジタルカタログで、ここからサードパーティーのソフトウェア、データ、サービスを検索、購入、配置、管理し、ソリューションの構築やビジネス運営に活用することができます。	AWS Marketplace サブスクリプションと購入のライセンスは、組織内のアカウント間で共有できます。	 はい 詳細はこちら	 いえ

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Marketplace Private Marketplace で利用可能な製品の幅広いカタログと AWS Marketplace、それらの製品のきめ細かな制御を提供します。	組織全体、1 つ以上の OUs、または組織内の 1 つ以上のアカウントに関連付けられた複数のプライベートマーケットプレイスエクスペリエンスを作成し、それぞれに独自の承認済み製品のセットを作成できます。AWS 管理者は、会社またはチームの口	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	ゴ、メッセージング、およびカラースキームに関する各プライベートマーケットプレイスエクスペリエンスに会社のブランドを適用することもできます。			

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Network Manager AWS アカウント、リージョン、オンプレミスロケーション間で、AWS Cloud WAN コアネットワークと AWS Transit Gateway ネットワークを一元管理できます。	組織 AWS 内の複数のアカウントで、トランジットゲートウェイとそのアタッチされたリソースを使用して、グローバルネットワークを一元管理およびモニタリングできます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon Q Developer Amazon Q Developer は、生成人工知能 (AI) を活用した会話アシスタントで、AWS アプリケーションの理解、構築、拡張、運用に役立ちます。	Amazon Q Developer の有料サブスクリプションバージョンには、Organizations の統合が必要です。	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Resource Access Manager 所有している指定された AWS リソースを他の アカウントと共有します。	組織内で、追加の招待を交換することなくリソースを共有できます。共有できるリソースには、Route 53 リゾルバーのルール、オンデマンドキャパシティーの予約などがあります。 キャパシティ予約の共有については、Amazon EC2 ユーザーガイド」また	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	はAmazon EC2 ユーザーガイド」を参照してください。 共有可能なリソースの一覧については、AWS RAM ユーザーガイドの共有可能なリソースを参照してください。			
AWS Resource Explorer インターネット検索エンジンのようなエクスペリエンスを使用してリソースを調べます。	マルチアカウント検索を有効にします。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Security Hub でセキュリティ状態を表示 AWS し、セキュリティ業界標準とベストプラクティスに照らして環境をチェックします。	Security Hub は、追加した新しいアカウントを含む組織のすべてのアカウントに対し、自動的に有効にすることができます。これにより、Security Hub のチェックと検出がより広範囲に行えるようになり、セキュリティ体制の全体像をより正確に把握できます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon S3 Storage Lens Amazon S3 ストレージの使用状況とアクティビティに関するメトリクスを可視化し、ストレージを最適化する実用的な推奨事項を提供します。	Amazon S3 Storage Lens を設定することで、Amazon S3 ストレージの使用状況とアクティビティの傾向を可視化するとともに、組織のすべてのメンバーアカウントに向けた推奨事項を取得することが可能になります。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon Security Lake Amazon Security Lake は、クラウド、オンプレミス、カスタムソースのセキュリティデータを、アカウントに保存されているデータレイクに一元化します。	アカウント全体からログとイベントを収集するデータレイクを作成します。	はい 詳細はこちら	はい 詳細はこちら	は
AWS Service Catalog AWSでの使用が承認されたIT サービスのカタログを作成および管理します。	ポートフォリオ ID を共有することなく、アカウント間でより簡単にポートフォリオの共有と製品のコピーができます。	はい 詳細はこちら	はい 詳細はこちら	は

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Service Quotas 一元的な場所から、サービスクォータ (制限とも呼ばれます) を表示および管理します。	クォータリクエストテンプレートを作成して、組織のアカウントの作成時に自動的にクォータの引き上げをリクエストできます。	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS IAM Identity Center すべてのアカウントとクラウドアプリケーションに対してシングルサインオンサービスを提供します。	ユーザーは、企業の認証情報を使用して AWS アクセスポータルにサインインし、割り当てられた管理アカウントまたはメンバーアカウントのリソースにアクセスできます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Systems Manager AWS リソースの可視性と制御を有効にします。	Systems Manager Explorer を使用して、組織内のすべての AWS アカウント でオペレーションデータを同期できます。 また、Systems Manager Change Manager を使用すれば、委任管理者アカウントから、組織内のすべてのメンバーアカウントを対象に、変更	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
	のテンプレート、承認、レポート作成を管理できます。			
タグポリシー 組織のアカウントのリソース間でタグを標準化するのに役立ちます。	タグポリシーを作成して特定のリソースおよびリソースタイプのタグ付けルールを定義し、そのポリシーを組織とアカウントにアタッチして適用できます。	 はい 詳細はこちら	 いえ	い

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Trusted Advisor Trusted Advisor は AWS 環境を検査し、コスト削減、システムの可用性とパフォーマンスの向上、セキュリティギャップの解消に役立つ機会が存在する場合にレコメンデーションを行います。	組織 AWS アカウント 内のすべての の Trusted Advisor チェックを実行します。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
AWS Well-Architected Tool AWS Well-Architected Tool は、ワークロードの状態を文書化し、最新の AWS アーキテクチャのベストプラクティスと比較するのに役立ちます。	AWS WA Tool と Organizations の両方のお客様が、組織の他のメンバーと AWS WA Tool リソースを共有するプロセスを簡素化できるようにします。	 はい 詳細はこちら	 いえ	いい

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon VPC IP Address Manager (IPAM) IPAM は、AWS ワークロードの IP アドレスの計画、追跡、モニタリングを容易にする VPC 機能です。	組織全体の IP アドレスの使用状況をモニタリングし、メンバーアカウント間で IP アドレスプールを共有します。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS サービス	で を使用する利点 AWS Organizations	信頼されたアクセスをサポート	委任管理者をサポート	
Amazon VPC Reachability Analyzer Reachability Analyzer は、仮想プライベートクラウド (VPC) 内のソースリソースと送信先リソース間の接続テストを実行できるようにする設定分析ツールです。	組織内のアカウント間のパスを追跡できます。	 はい 詳細はこちら	 はい 詳細はこちら	

AWS Account Management および AWS Organizations

AWS Account Management は、組織 AWS アカウント 内のすべての のアカウント情報とメタデータを管理するのに役立ちます。組織の各メンバーアカウントの代替連絡先情報を設定、変更、または削除できます。詳細については、AWS Account Management ユーザーガイドの「組織で AWS Account Management を使用する」を参照してください。

以下の情報は、 AWS Account Management との統合に役立ちます AWS Organizations。

Account Management で信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

アカウント管理では、組織のこのサービスの委任管理者としてメンバーアカウントを指定する AWS Organizations 前に、 への信頼されたアクセスが必要です。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Account Managementで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Account Management」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Account Management に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Account Management として を有効にできます。

```
$ aws organizations enable-aws-service-access \
```

```
    --service-principal account.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

Account Management で信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

で信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです AWS Account Management。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Account Managementで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Account Management」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Account Management に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Account Management として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal account.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

Account Management 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、指定されたアカウントのユーザーおよびロールは組織のその他のメンバーアカウントの AWS アカウント メタデータを管理できるようになります。委任された管理者アカウントを有効にしない場合、これらのタスクは組織の管理アカウントによってのみ実行できます。この手法は、組織の管理からアカウントの詳細の管理を分離するのに有効です。

> 最小アクセス許可
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Account Management の委任管理者としてメンバーアカウントを設定できます

委任ポリシーを設定する方法については、「リソースベースの委任ポリシーを作成または更新する」を参照してください。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

  ```
  $  aws organizations register-delegated-administrator \
  ```

```
    --account-id 123456789012 \
    --service-principal account.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスプリンシパルをパラメータ`account.amazonaws.com`として識別します。

AWS Application Migration Service（ アプリケーション移行サービス) および AWS Organizations

AWS Application Migration Service は、アプリケーションを に移行する際のコストを簡素化、迅速化、削減します AWS。Organizations と統合することで、グローバルビュー機能を使用して複数のアカウントにまたがる大規模な移行を管理できます。詳細については、「Application Migration Service ユーザーガイド AWS Organizations」の「 のセットアップ」を参照してください。

AWS Application Migration Service との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Application Migration Service は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、Application Migration Service と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

- `AWSServiceRoleForApplicationMigrationService`

Application Migration Service で使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Application Migration Service で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセスを許可します。

- `mgn.amazonaws.com`

Application Migration Service での信頼されたアクセスの有効化

Application Migration Service で信頼されたアクセスを有効にすると、グローバルビュー機能を使用できます。これにより、複数のアカウント間で大規模な移行を管理できます。グローバルビューは、さまざまな AWS アカウントのソースサーバー、アプリ、ウェーブに対して特定のアクションを実行するための可視性と機能を提供します。詳細については、「 AWS Application Migration Service ユーザーガイド」のAWS 「Organizations のセットアップ」を参照してください。

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼された AWS Organizations アクセスは、 AWS Application Migration Service コンソールまたはコンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Application Migration Service コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Application Migration Service を実行できます。ここに示す手順は、統合の有効化に AWS Application Migration Serviceが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
>
> AWS Application Migration Service コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Application Migration Serviceで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。

5. 「 の信頼されたアクセスを有効にする AWS Application Migration Service」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Application Migration Service に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼できるサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Application Migration Service として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal mgn.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

Application Migration Service での信頼されたアクセスの無効化

アプリケーション移行サービスでの信頼されたアクセスを無効にすることができるのは、Organizations 管理アカウントの管理者のみです。

信頼されたアクセスは、 AWS Application Migration Service または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Application Migration Service コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、 は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS

Application Migration Service を実行できます。ここに示す手順は、統合の無効化に AWS Application Migration Serviceが提供するツールを使用できない場合にのみ実施してください。
AWS Application Migration Service コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Application Migration Serviceで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Application Migration Service」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Application Migration Service に、コンソールまたはツールを使用してそのサービスを無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Application Migration Service として を無効にできます。

```
$ aws organizations disable-aws-service-access \
    --service-principal mgn.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Application Migration Service の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーとロールは、Application Migration Service の管理アクションを実行できます。それ以外の場合は、組織の管理アカウントのユーザーまたはロールのみが実行できます。これにより、組織の管理と Application Migration Service の管理を分離できます。詳細については、「Application Migration Service ユーザーガイド AWS Organizations」の「 のセットアップ」を参照してください。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内のアプリケーション移行サービスの委任管理者としてメンバーアカウントを設定できます。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

```
$ aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal mgn.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスをパラメータ`mgn.amazonaws.com`として識別します。

Application Migration Service の委任管理者の無効化

アプリケーション移行サービスの委任された管理者を削除できるのは、Organizations 管理アカウントの管理者のみです。Organizations の `DeregisterDelegatedAdministrator` CLI または SDK オペレーションを使用して、委任された管理者を削除できます。

AWS Artifact および AWS Organizations

AWS Artifact は、ISO レポートや PCI レポートなどの AWS セキュリティコンプライアンスレポートをダウンロードできるサービスです。を使用すると AWS Artifact、組織の管理アカウントのユーザーは、新しいレポートやアカウントが追加されても、組織内のすべてのメンバーアカウントに代わって契約を自動的に承諾できます。メンバーアカウントのユーザーは、契約を表示およびダウンロードできます。詳細については、「 ユーザーガイド」の AWS 「アーティファクトでの複数のアカウントの契約の管理AWS Artifact 」を参照してください。

以下の情報は、 AWS Artifact との統合に役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより AWS Artifact 、 は、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 AWS Artifact と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

組織からメンバーアカウントを削除すると、このロールを削除または変更できますが、お勧めしません。

サービス間での混乱した代理などのセキュリティ上の問題を引き起こす可能性があるため、ロールの変更は推奨されません。混乱した使節に対する保護の詳細については、「AWS Artifact ユーザーガイド」の「Cross-service deputy prevention」(サービス間での使節の防止) を参照してください。

- `AWSServiceRoleForArtifact`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス AWS Artifact を許可します。

- `artifact.amazonaws.com`

AWS Artifactとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Artifactで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Artifact」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 コンソールを使用してそのサービスを有効に AWS Artifact して を操作できるようになったことを 管理者に伝えます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Artifact として を有効にできます。

```
$ aws organizations enable-aws-service-access \
    --service-principal artifact.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

AWS Artifactとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

で信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです AWS Artifact。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

AWS Artifact では、組織契約と連携 AWS Organizations するために との信頼されたアクセスが必要です。組織契約 AWS Artifact で を使用している AWS Organizations ときに を使用して信頼されたアクセスを無効にすると、組織にアクセスできないため、機能しなくなります。で承諾した組織契約は AWS Artifact そのまま残りますが、 からはアクセスできません AWS Artifact。が AWS Artifact 作成する AWS Artifact ロールは残ります。その後、信頼されたアクセスを再度有効にすると、 AWS Artifact は以前のように動作し続けます。サービスを再設定する必要はありません。

スタンドアロンアカウントは組織から削除され、組織契約にアクセスできなくなります。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Artifactで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。

5. 「 の信頼されたアクセスを無効にする AWS Artifact」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Artifact に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にできます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Artifact として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal artifact.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

AWS Audit Manager および AWS Organizations

AWS Audit Manager を使用すると、AWS の使用状況を継続的に監査し、リスク評価、および規制や業界標準とのコンプライアンスの評価の方法を簡素化できます。Audit Manager はエビデンスの収集を自動化するため、ポリシー、手順、アクティビティが効果的に機能しているかどうかの評価が容易になります。監査が実施される際には、Audit Manager を使用し、コントロールについてのステークホルダーレビューを管理できます。また、監査用のレポートの作成に費やす手間を大幅に削減できます。

Audit Manager をAWS Organizations と統合すれば、評価対象になっている組織の AWS アカウントを複数含めることで、より広範な情報源からエビデンスを収集できます。

詳細については、Audit Manager ユーザーガイドの Enable AWS Organizations を参照してください。

AWS Audit Manager と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Audit Manager はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Audit Manager と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合です。

Audit Manager でこのロールを使用する方法について詳しくは、AWS Audit Manager ユーザーガイドの Using service-linked roles を参照してください。

- `AWSServiceRoleForAuditManager`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Audit Manager によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `auditmanager.amazonaws.com`

Audit Manager との信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

組織の委任管理者にするメンバーアカウントを指定するにあたり、Audit Manager には AWS Organizations への信頼されたアクセスが必要です。

AWS Audit Manager コンソールまたは AWS Organizations コンソールのいずれかを使用して、信頼されたアクセスを有効にできます。

> **Important**
>
> Organizations との統合の有効化には、可能な場合は常に AWS Audit Manager コンソールまたはツールを使用することを強くお勧めします。そうすることで、サービスに必要なリソー

スの作成など、必要な構成が AWS Audit Manager で実行可能になります。ここに示す手順は、統合の有効化に AWS Audit Manager が提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
AWS Audit Manager コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実施する必要はありません。

Audit Manager コンソールを使用して信頼されたアクセスを有効にするには

信頼されたアクセスを有効にする手順については、AWS Audit Manager ユーザーガイドのセットアップを参照してください。

> **Note**
>
> AWS Audit Manager コンソールを使用して委任管理者を設定する場合は、信頼されたアクセスは AWS Audit Manager によって自動的に有効になります。

信頼されたアクセスの有効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、AWS Audit Manager を Organizations で信頼されたサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal auditmanager.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: EnableAWSServiceAccess

Audit Manager との信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

AWS Organizations 管理アカウントの管理者だけが AWS Audit Manager との信頼されたアクセスを無効にできます。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、AWS Audit Manager を Organizations で信頼されたサービスとして無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal auditmanager.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

Audit Manager 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、Audit Manager の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Audit Manager の管理を分離するのに有効です。

> **最小アクセス許可**
>
> 次の許可を持つ Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Audit Manager の委任管理者としてメンバーアカウントを設定できます。`audit-manager:RegisterAccount`

Audit Manager 用に委任管理者アカウントを有効にする手順については、AWS Audit Manager ユーザーガイドのセットアップを参照してください。

AWS Audit Manager コンソールを使用して委任管理者を設定すると、信頼されたアクセスが Audit Manager によって自動的に有効になります。

AWS CLI, AWS API

AWS CLI または AWS SDK を使用して委任管理者アカウントを設定するには、次のコマンドを使用します。

- AWS CLI:

```
$  aws audit-manager register-account \
    --delegated-admin-account 123456789012
```

- AWS SDK: 管理者アカウントを委任するには、`RegisterAccount` オペレーションを呼び出して `delegatedAdminAccount` をパラメーターとして渡します。

AWS Backup および AWS Organizations

AWS Backup は、組織内の AWS Backup ジョブを管理およびモニタリングできるサービスです。組織の管理アカウントのユーザーとしてサインインすると、AWS Backup を使用して組織全体のバックアップ保護とモニタリングを有効にできます。バックアップポリシーを使用して、組織内のすべてのアカウントのリソースに AWS Backup 計画を一元的に適用することで、コンプライアンスを達成できます。AWS Backup と AWS Organizations を一緒に使用すると、次の利点が得られます。

保護

組織でバックアップポリシータイプを有効にし、バックアップポリシーを作成して、組織のルート、OU、またはアカウントにアタッチできます。バックアップポリシーによって、AWS Backup プランと、そのプランを自動的にアカウントに適用するために必要なその他の詳細情報を組み合

わせることができます。アカウントに直接アタッチされたポリシーが、組織のルートおよび親 OU から継承されたポリシーとマージされて、アカウントに適用できる有効なポリシーを作成します。ポリシーには、アカウントのリソースで AWS Backup を実行するアクセス許可を持つ IAM ロールの ID が含まれます。AWS Backup はその IAM ロールを使用し、有効なポリシーのバックアッププランで指定されているとおりに、ユーザーに代わってバックアップを実行します。

モニタリング

組織で AWS Backupに対して信頼されたアクセスを有効にすると、AWS Backup コンソールを使用して、組織内の任意のアカウントのバックアップ、復元、およびコピージョブの詳細を表示できます。詳細については、AWS Backup デベロッパーガイドの Monitor your backup jobs を参照してください。

AWS Backup の詳細については、「AWS Backup デベロッパーガイド」を参照してください。

AWS Backup と AWS Organizations の統合には、次の情報を参考にしてください。

AWS Backup との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

AWS Backup コンソールまたは AWS Organizations コンソールのいずれかを使用して、信頼されたアクセスを有効にできます。

> **Important**
>
> Organizations との統合の有効化には、可能な場合は常に AWS Backup コンソールまたはツールを使用することを強くお勧めします。そうすることで、サービスに必要なリソースの作成など、必要な構成が AWS Backup で実行可能になります。ここに示す手順は、統合の有効化に AWS Backup が提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Backup コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実施する必要はありません。

AWS Backup を使用して信頼されたアクセスを有効にするには、AWS Backup デベロッパーガイドの Enabling backup in multiple AWS アカウント を参照してください。

AWS Backup との信頼されたアクセスの無効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

組織のアカウント全体でジョブのバックアップ、復元、コピーのモニタリングを有効にするにあたり、AWS Backup には AWS Organizations との信頼されたアクセスが必要です。信頼されたアクセス AWS Backup を無効にすると、現在のアカウント以外のジョブを表示できなくなります。AWS Backup で作成される AWS Backup ロールは残ります。信頼されたアクセスを後で再度有効にすると、AWS Backup は以前のように動作し続けます。サービスを再設定する必要はありません。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、AWS Backup を Organizations で信頼されたサービスとして無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal backup.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

AWS Backup 用の委任管理者アカウントの有効化

「AWS Backup デベロッパーガイド」の「委任管理者」を参照してください。

AWS Billing and Cost Management および AWS Organizations

AWS Billing and Cost Management には、請求のセットアップ、請求書の取得と支払い、コストの分析、整理、計画、最適化に役立つ一連の機能が用意されています。Billing and Cost Management を で使用する AWS Organizations と、該当する場合は分割コスト配分データによる AWS Organizations 情報の取得と、オプトインした分割コスト配分データサービスのテレメトリデータの収集が可能になります。

以下の情報は、 AWS Billing and Cost Management との統合に役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、請求情報とコスト管理は、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、請求情報とコスト管理および組織間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

詳細については、「 Billing and Cost Management ユーザーガイド」の「請求とコスト管理のサービスにリンクされたロールのアクセス許可」を参照してください。

- `AWSServiceRoleForSplitCostAllocationData`

請求情報とコスト管理で使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。請求情報とコスト管理で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセスを許可します。

請求情報とコスト管理では、 `billing-cost-management.amazonaws.com`サービスプリンシパルを使用します。

請求情報とコスト管理による信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

管理アカウント経由で信頼されたアクセスを有効にすると、お客様は請求情報とコスト管理のコスト配分データの分割機能を利用できます。お客様が Amazon Managed Service for Prometheus で

Amazon Elastic Kubernetes Service のコスト配分データの分割を有効にすると、信頼されたアクセスが呼び出され、組織内のすべてのメンバーアカウントのサービスにリンクされたロールが作成されます。これにより、分割コスト配分データは、顧客の Amazon Managed Service for Prometheus ワークスペースからテレメトリデータを収集し、それらのメトリクスに基づいてコスト配分を実行できます。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Billing and Cost Managementで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Billing and Cost Management」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Billing and Cost Management に、コンソールを使用してそのサービスを有効にして と連携できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Billing and Cost Management として を有効にできます。

```
$ aws organizations enable-aws-service-access \
    --service-principal  billing-cost-management.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、 Organizations の信頼されたサービス AWS Billing and Cost Management として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal  billing-cost-management.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

AWS CloudFormation StackSets および AWS Organizations

AWS CloudFormation StackSets を使用すると、複数の AWS アカウント にまたがるスタックを作成、更新、または削除できます AWS リージョン 。 と StackSets の統合 AWS Organizations によ

り、各メンバーアカウントで関連するアクセス許可を持つサービスにリンクされたロールを使用して、サービス管理アクセス許可を持つスタックセットを作成できます。これにより、組織内のメンバーアカウントにスタックインスタンスをデプロイできるようになります。必要な AWS Identity and Access Management ロールを作成する必要はありません。 は、ユーザーに代わって各メンバーアカウントに IAM ロール StackSets を作成します。

また、将来組織に追加されるアカウントへの自動デプロイを有効にすることもできます。自動デプロイを有効にすると、今後その OU に追加されるすべてのアカウントにロールと関連するスタックセットインスタンスのデプロイが自動的に追加されるようになります。

StackSets と Organizations 間の信頼されたアクセスを有効にすると、管理アカウントには組織のスタックセットを作成および管理するためのアクセス許可が付与されます。管理アカウントは、委任管理者として最大 5 つのメンバーアカウントを登録できます。信頼されたアクセスが有効になっていると、組織のスタックセットを作成および管理するためのアクセス許可が委任管理者にも付与されます。サービスマネージド型のアクセス許可を持つスタックセットは、委任された管理者によって作成されたスタックセットを含む、管理アカウントに作成されます。

> **Important**
>
> 委任された管理者は、組織内のアカウントにデプロイするための完全なアクセス許可を持っています。管理アカウントでは、特定の OU にデプロイしたり、特定のスタックセットの操作を実行したりする、委任された管理者のアクセス許可を制限することはできません。

Organizations StackSets との統合の詳細については、「 AWS CloudFormation ユーザーガイド」の AWS CloudFormation StackSets「 の使用」を参照してください。

AWS CloudFormation StackSets との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、 AWS CloudFormation スタックセットは、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 AWS CloudFormation StackSets と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- 管理アカウント: `AWSServiceRoleForCloudFormationStackSetsOrgAdmin`

サービスにリンクされたロール `AWSServiceRoleForCloudFormationStackSetsOrgMember` を組織内のメンバーアカウントに作成するには、始めに管理アカウントにスタックセットを作成する必要があります。これにより、スタックセットインスタンスが作成され、メンバーアカウントにロールが作成されます。

- メンバーアカウント: `AWSServiceRoleForCloudFormationStackSetsOrgMember`

スタックセットの作成の詳細については、「 ユーザーガイド」の AWS CloudFormation StackSets「 の使用AWS CloudFormation 」を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。 AWS CloudFormation Stacksets で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセスを許可します。

- 管理アカウント: `stacksets.cloudformation.amazonaws.com`

  このロールを変更または削除できるのは、 StackSets と Organizations 間の信頼されたアクセスを無効にした場合のみです。
- メンバーアカウント: `member.org.stacksets.cloudformation.amazonaws.com`

  アカウントからこのロールを変更または削除できるのは、 StackSets と Organizations 間の信頼されたアクセスを最初に無効にした場合、またはターゲット組織または組織単位 (OU) からアカウントを最初に削除した場合のみです。

AWS CloudFormation StackSets との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations 管理アカウントの管理者のみが、別の AWS サービスへの信頼されたアクセスを有効にするアクセス許可を持っています。 AWS CloudFormation コンソールまたは Organizations コンソールを使用して、信頼されたアクセスを有効にできます。

信頼されたアクセスは、 のみを使用して有効にできます AWS CloudFormation StackSets。

AWS CloudFormation Stacksets コンソールを使用して信頼されたアクセスを有効にするには、 AWS CloudFormation ユーザーガイドの「 で信頼されたアクセスを有効にする AWS Organizations」を参照してください。

AWS CloudFormation StackSets との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations 管理アカウントの管理者のみが、別の AWS サービスへの信頼されたアクセスを無効にするアクセス許可を持っています。信頼されたアクセスの無効化には、Organizations コンソールを使用する必要があります。の使用中に Organizations との信頼されたアクセスを無効にすると StackSets、以前に作成したすべてのスタックインスタンスが保持されます。ただし、サービスにリンクされたロールのアクセス許可を使用してデプロイされたスタックセットは、Organizations によって管理されるアカウントへのデプロイを実行できなくなります。

信頼されたアクセスは、 AWS CloudFormation コンソールまたは Organizations コンソールを使用して無効にできます。

> **Important**
>
> 信頼されたアクセスをプログラムで無効にした場合 (例: AWS CLI または API で) 、アクセス許可が削除されることに注意してください。 AWS CloudFormation コンソールで信頼されたアクセスを無効にすることをお勧めします。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS CloudFormation StackSetsで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。

5. 「 の信頼されたアクセスを無効にする AWS CloudFormation StackSets」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 のコンソールまたはツールを使用してそのサービスを無効にできるようになった AWS CloudFormation StackSets ことを の管理者に伝えます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、 Organizations の信頼されたサービス AWS CloudFormation StackSets として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal stacksets.cloudformation.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

AWS CloudFormation Stacksets の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 AWS CloudFormation StackSets の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。これにより、組織の管理と AWS CloudFormation Stacksets の管理を分離できます。

メンバーアカウントを組織の AWS CloudFormation StackSets の委任管理者として指定する手順については、AWS CloudFormation ユーザーガイドの委任された管理者の登録を参照してください。

AWS CloudTrail および AWS Organizations

AWS CloudTrail は、 のガバナンス、コンプライアンス、運用およびリスクの監査を可能にする AWS のサービスです AWS アカウント。を使用すると AWS CloudTrail、管理アカウントのユーザー

は、その組織 AWS アカウント 内のすべての のすべてのイベントを記録する組織の証跡を作成できます。組織の証跡は、組織内のすべてのメンバーアカウントに自動的に適用されます。メンバーアカウントは組織の証跡を表示できますが、これを変更または削除することはできません。デフォルトでは、メンバーアカウントは Amazon S3 バケット内にある組織の証跡のログファイルにはアクセスできません。これにより、組織内のすべてのアカウントに対してイベントのログ記録戦略を一律に適用および実施できます。

組織の証跡については、AWS CloudTrail ユーザーガイドの組織の証跡の作成を参照してください。

AWS CloudTrail との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより CloudTrail 、 は、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 CloudTrail と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

- `AWSServiceRoleForCloudTrail`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス CloudTrail を許可します。

- `cloudtrail.amazonaws.com`

CloudTrail との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

AWS CloudTrail コンソールから証跡を作成して信頼されたアクセスを有効にすると、信頼されたアクセスが自動的に設定されます (推奨) 。 AWS Organizations コンソールを使用して信頼されたアクセスを有効にすることもできます。組織の証跡を作成するには、 AWS Organizations 管理アカウントでサインインする必要があります。

AWS CLI または AWS API を使用して組織の証跡を作成する場合は、信頼されたアクセスを手動で設定する必要があります。詳細については、「 ユーザーガイド」の「 を信頼されたサービス CloudTrail として有効に AWS Organizationsするの AWS CloudTrail 」を参照してください。

> **Important**
>
> 可能な限り、 AWS CloudTrail コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。

信頼されたアクセスを有効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼できるサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS CloudTrail として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal cloudtrail.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

CloudTrail との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

AWS CloudTrail では、組織の証跡や組織のイベントデータストアを操作する AWS Organizations ために、 との信頼されたアクセスが必要です。の使用中に を使用して AWS Organizations 信頼されたアクセスを無効にすると AWS CloudTrail、 は組織にアクセス CloudTrail できないため、メン

バーアカウントのすべての組織証跡が削除されます。すべての管理アカウントの組織の証跡と組織のイベントデータストアは、アカウントレベルの証跡とイベントデータストアに変換されます。CloudTrail と の統合用に作成されたAWSServiceRoleForCloudTrailロールは、アカウント内に AWS Organizations 残ります。信頼されたアクセスを再度有効にすると、 CloudTrail は既存の証跡とイベントデータストアに対してアクションを実行しません。管理アカウントは、アカウントレベルの証跡とイベントデータストアを更新して、組織に適用する必要があります。

アカウントレベルの証跡またはイベントデータストアを組織の証跡または組織のイベントデータストアに変換するには、次の手順を実行します。

- CloudTrail コンソールから、証跡またはイベントデータストアを更新し、組織内のすべてのアカウントに対して有効にするオプションを選択します。
- から AWS CLI、次の操作を行います。
  - 証跡を更新するには、 update-trail コマンドを実行し、 `--is-organization-trail`パラメータを含めます。
  - イベントデータストアを更新するには、 update-event-data-store コマンドを実行し、 `--organization-enabled`パラメータを含めます。

で信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです AWS CloudTrail。信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS CloudTrailで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS CloudTrail」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。

6. のみの管理者である場合は AWS Organizations、 の管理者 AWS CloudTrail に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS CloudTrail として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal cloudtrail.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

の委任管理者アカウントの有効化 CloudTrail

Organizations CloudTrail で を使用すると、組織内の任意のアカウントを登録して、組織に代わって組織の証跡とイベントデータストアを管理する CloudTrail ための委任管理者として行動できます。委任管理者は、 で管理アカウント CloudTrail と同じ管理タスクを実行できる組織のメンバーアカウントです。

> **最小アクセス許可**
>
> Organizations 管理アカウントの管理者のみが、 の委任管理者を登録できます CloudTrail。

委任された管理者アカウントは、 CloudTrail コンソールを使用するか、Organizations CLI または SDK `RegisterDelegatedAdministrator` オペレーションを使用して登録できます。 CloudTrail コンソールを使用して委任管理者を登録するには、 CloudTrail 「委任管理者の追加」を参照してください。

の委任管理者を無効にする CloudTrail

Organizations 管理アカウントの管理者のみが、 の委任された管理者を削除できます CloudTrail。委任された管理者は、 CloudTrail コンソールを使用するか、Organizations CLI または SDK `DeregisterDelegatedAdministrator` オペレーションを使用して削除できます。 CloudTrail コンソールを使用して委任された管理者を削除する方法については、 CloudTrail 「委任された管理者の削除」を参照してください。

AWS Compute Optimizer および AWS Organizations

AWS Compute Optimizer は、 AWS リソースの設定と使用率のメトリクスを分析するサービスです。リソースの例には、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスや Auto Scaling グループがあります。Compute Optimizer は、リソースが最適かどうかを報告します。また、コストを削減し、ワークロードのパフォーマンスを向上させるための最適化に関する推奨事項を生成します。Compute Optimizer について詳しくは、AWS Compute Optimizer ユーザーガイドを参照してください。

AWS Compute Optimizer との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Compute Optimizer はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Compute Optimizer と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForComputeOptimizer`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Compute Optimizer によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `compute-optimizer.amazonaws.com`

Compute Optimizer との信頼されたアクセスを有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼された AWS Organizations アクセスは、 AWS Compute Optimizer コンソールまたは コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Compute Optimizer コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Compute Optimizer を実行できます。ここに示す手順は、統合の有効化に AWS Compute Optimizerが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Compute Optimizer コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

Compute Optimizer コンソールを使用して信頼されたアクセスを有効にするには

組織の管理アカウントを使用して Compute Optimizer コンソールにサインインする必要があります。組織を代表してオプトインするには、AWS Compute Optimizer ユーザーガイドのアカウントにオプトインするの指示に従います。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Compute Optimizerで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。

5. 「 の信頼されたアクセスを有効にする AWS Compute Optimizer」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Compute Optimizer に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、 を Organizations の信頼されたサービス AWS Compute Optimizer として有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal compute-optimizer.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

Compute Optimizer との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

で信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです AWS Compute Optimizer。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Compute Optimizer として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal compute-optimizer.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Compute Optimizer の委任された管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、指定されたアカウントのユーザーおよびロールは組織のその他のメンバーアカウントの AWS アカウント メタデータを管理できるようになります。委任された管理者アカウントを有効にしない場合、これらのタスクは組織の管理アカウントによってのみ実行できます。この手法は、組織の管理からアカウントの詳細の管理を分離するのに有効です。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Computer Optimizer の委任管理者としてメンバーアカウントを設定できます

Compute Optimizer の委任された管理者アカウントを有効にする手順については、AWS Compute Optimizer ユーザーガイドの https://docs.aws.amazon.com/compute-optimizer/latest/ug/delegate-administrator-account.html を参照してください。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

```
$  aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal compute-optimizer.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスプリンシパルをパラメータ`account.amazonaws.com`として識別します。

Compute Optimizer の委任された管理者の無効化

Compute Optimizer の委任された管理者を設定できるのは、組織管理アカウントの管理者だけです。

Compute Optimizer コンソールを使用して、委任された管理者 Compute Optimizer アカウントを無効にするには、AWS Compute Optimizer ユーザーガイドの https://docs.aws.amazon.com/compute-optimizer/latest/ug/delegate-administrator-account.html を参照してください。

を使用して委任された管理者を削除するには AWS AWS CLI、 コマンドリファレンスのderegister-delegated-administrator「」を参照してください。 AWS AWS CLI

AWS Config および AWS Organizations

のマルチアカウント、マルチリージョンのデータ集約 AWS Config を使用すると、複数のアカウントと の AWS Config データを AWS リージョン 1 つのアカウントに集約できます。マルチアカウント、マルチリージョンのデータ集約は、中央の IT 管理者がエンタープライズの複数の AWS アカウント のコンプライアンスをモニタリングするうえで役立ちます。アグリゲータは、複数のソースアカウントとリージョンから AWS Config データを収集 AWS Config する のリソースタイプです。集計 AWS Config データを表示するリージョンにアグリゲータを作成します。アグリゲータの作成中、個々のアカウント ID、または組織の追加を選択できます。の詳細については AWS Config、「 AWS Config デベロッパーガイド」を参照してください。

AWS Config APIs、組織内のすべての AWS アカウント のルールを管理する AWS Config こともできます。詳細については、「 AWS Config デベロッパーガイド」の「組織内のすべてのアカウントで AWS Config ルールを有効にする」を参照してください。

以下の情報は、 AWS Config との統合に役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織のアカウントに作成されます。このロールにより AWS Config 、 は組織内のアカウント内でサポートされているオペレーションを実行できます。

- `AWSServiceRoleForConfig`

このロールは、マルチアカウント aggregator. AWS Config asks を作成して組織 AWS Config で を有効にしたときに作成され、ロールを選択または作成し、名前を指定します。名前は自動生成されません。

このロールを削除または変更できるのは、 AWS Config と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

AWS Configとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼された AWS Organizations アクセスは、 AWS Config コンソールまたは コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Config コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Config を実行できます。ここに示す手順は、統合の有効化に AWS Config が提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Config コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS Config コンソールを使用して信頼されたアクセスを有効にするには

AWS Organizations を使用して への信頼されたアクセスを有効にするには AWS Config、マルチアカウントアグリゲータを作成し、組織を追加します。マルチアカウントアグリゲータの設定方法につい

ては、AWS Config デベロッパーガイドのコンソールを使用したアグリゲータのセットアップを参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Configで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Config」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 コンソールを使用してそのサービスを有効に AWS Config して を操作できるようにすることを管理者に伝えてください AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Config として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal config.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

AWS Configとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Config として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal config.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

AWS Cost Optimization Hub および AWS Organizations

AWS Cost Optimization Hub は、 AWS アカウントと AWS リージョン全体でコスト最適化のレコメンデーションを統合して優先順位を付けるのに役立つ AWS 請求情報とコスト管理機能です。これにより、 AWS 支出を最大限に活用できます。Cost Optimization Hub を で使用すると、Organizations メンバーアカウントと AWS リージョン全体で AWS コスト最適化のレコメンデーション AWS Organizations を簡単に特定、フィルタリング、集計できます。

詳細については、「 ユーザーガイド」の「Cost Optimization HubAWS Cost Management 」を参照してください。

AWS Cost Optimization Hub との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Cost Optimization Hub は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、Cost Optimization Hub と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

詳細については、「 ユーザーガイド」の「Cost Optimization Hub のサービスにリンクされたロールのアクセス許可AWS Cost Management 」を参照してください。

- `AWSServiceRoleForCostOptimizationHub`

Cost Optimization Hub で使用されるサービスプリンシパル

Cost Optimization Hub は `cost-optimization-hub.bcm.amazonaws.com`サービスプリンシパルを使用します。

Cost Optimization Hub で信頼されたアクセスを有効にする

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

組織の管理アカウントを使用してオプトインし、組織内のすべてのメンバーアカウントを含めると、Cost Optimization Hub の信頼されたアクセスが組織アカウントで自動的に有効になります。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。

3. サービスのリストAWS Cost Optimization Hubで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Cost Optimization Hub」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Cost Optimization Hub に、コンソールを使用してそのサービスを有効にして を操作できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Cost Optimization Hub として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal cost-optimization-hub.bcm.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

> **Important**
>
> オプトイン後に Cost Optimization Hub の信頼されたアクセスを無効にすると、Cost Optimization Hub は組織のメンバーアカウントのレコメンデーションへのアクセスを拒否し

ます。さらに、組織内のメンバーアカウントは Cost Optimization Hub にオプトインされません。詳細については、「 ユーザーガイド」の「Cost Optimization Hub and Organizations の信頼されたアクセス」を参照してください。 AWS Cost Management

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Cost Optimization Hub として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal cost-optimization-hub.bcm.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

AWS Control Tower および AWS Organizations

AWS Control Tower は、規範的なベスト プラクティスに従って、AWS マルチアカウント環境をセットアップおよび管理するための簡単な方法を提供します。AWS Control Tower オーケストレーションは AWS Organizations の機能を拡張します。 AWS Control Tower は、組織やアカウントがベストプラクティスから逸脱 (ドリフト) しないようにするために、予防的および発見的な制御 (ガードレール) を適用します。

AWS Control Tower オーケストレーションは、AWS Organizations の機能を拡張します。

詳細については、AWS Control Tower ユーザーガイドを参照してください。

AWS Control Tower と AWS Organizations の統合には、次の情報を参考にしてください。

統合に必要な役割

AWSControlTowerExecution ロールは、登録されたすべてのアカウントに存在する必要があります。これにより AWS Control Tower が個々のアカウントを管理し、それらのアカウントに関する情報を監査アカウントおよびログアーカイブアカウントに報告できるようにするものです。

AWS Control Tower で使用されるロールの詳細については、「AWS Control Tower がロールを使用してアカウントを作成および管理する方法」および「AWS Control Tower 用のアイデンティティベースのポリシー (IAM ポリシー) の使用」を参照してください。

AWS Control Tower によって使用されるサービスプリンシパル

AWS Control Tower は、controltower.amazonaws.com サービスプリンシパルを使用します。

AWS Control Tower との信頼されたアクセスの有効化

AWS Control Tower は、信頼できるアクセスを使用して、予防管理のためのドリフトを検出し、ドリフトを引き起こすアカウントと OU の変更を追跡します。

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

Organizations コンソールから信頼されたアクセスを有効にするには、AWS Control Tower の隣の **Enable access** を選択します。

信頼されたアクセスの有効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、AWS Control Tower を Organizations で信頼されたサービスとして有効にすることができます。

```
$ aws organizations enable-aws-service-access \
    --service-principal controltower.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: EnableAWSServiceAccess

AWS Control Tower との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

> **Important**
>
> AWS Control Tower の信頼できるアクセスを無効にすると、AWS Control Tower ランディングゾーンにドリフトが生じます。ドリフトを修正する唯一の方法は、AWS Control Tower のランディングゾーンの修理を利用することです。Organizations での信頼できるアクセスを再度有効にしても、ドリフトは解決しません。ドリフトの詳細については、「AWS Control Tower ユーザーガイド」を参照してください。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、AWS Control Tower を Organizations で信頼されたサービスとして無効にすることができます。

```
$ aws organizations disable-aws-service-access \
```

```
    --service-principal controltower.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

Amazon Detective と AWS Organizations

Amazon Detective がログデータを使用して可視化を生成することにより、セキュリティに関する検出結果や疑わしいアクティビティの根本原因を分析、調査、および特定できるようになります。

AWS Organizations を使用すると、Detective の動作グラフですべての組織アカウントのアクティビティを可視化できます。

Detective への信頼されたアクセスを許可すると、組織のメンバーシップに変更があった場合、Detective サービスが自動的に対応します。委任管理者が、任意の組織アカウントを動作グラフのメンバーアカウントとして有効にできます。Detective では、新しい組織アカウントをメンバーアカウントとして自動的に有効化することもできます。組織アカウントの動作グラフとの関連付けを解除することはできません。

詳細については、「Amazon Detective 管理ガイド」の「Using Amazon Detective in your organization」(組織内で Amazon Detective を使用する) を参照してください。

Amazon Detective を と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Detective はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Detective と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForDetective`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Detective によって使用さ

れるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `detective.amazonaws.com`

Detective との信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

> **Note**
>
> Amazon Detective の委任管理者を指定すると、組織の Detective に対する信頼されたアクセスが自動的に有効になります。
> Detective では、組織のこのサービスの委任管理者となるメンバーアカウントを指定する AWS Organizations 前に、 への信頼されたアクセスが必要です。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

AWS Organizations コンソールを使用して、信頼されたアクセスを有効にできます。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon Detective を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. Amazon Detective の信頼されたアクセスを有効にするダイアログボックスで、確認のために有効化と入力し、信頼されたアクセスを有効にするを選択します。
6. のみの管理者である場合は AWS Organizations、Amazon Detective の管理者に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

Detective との信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Amazon Detective との信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

AWS Organizations コンソールを使用して、信頼されたアクセスを無効にすることができます。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon Detective を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. Amazon Detective の信頼されたアクセスを無効にするダイアログボックスで、無効にして確認します。次に、信頼されたアクセスを無効にするを選択します。
6. のみの管理者の場合は AWS Organizations、コンソールまたはツールを使用してそのサービスを無効にできるようになったことを Amazon Detective の管理者に知らせます AWS Organizations。

Detective 用の委任管理者アカウントの有効化

Detective 用の委任管理者アカウントは、Detective 動作グラフの管理者アカウントになります。委任管理者は、その動作グラフのメンバーアカウントとして有効または無効にする組織アカウントを決定します。委任管理者は、新しい組織アカウントが組織に追加されたときに、メンバーアカウントとして自動的に有効にするように Detective を設定できます。委任管理者が組織アカウントを管理する方法については、「Amazon Detective 管理ガイド」の「組織アカウントをメンバーアカウントとして管理する」を参照してください。

Detective 用の委任管理者を設定できるのは、組織管理アカウントの管理者だけです。

委任管理者アカウントを指定する場合は、Detective コンソールまたは API を介して、あるいは Organizations CLI または SDK オペレーションを使用して行います。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Detective の委任管理者としてメンバーアカウントを設定できます

Detective コンソールまたは API を使用して委任管理者を設定するには、「Amazon Detective 管理ガイド」の「組織の Detective 管理者アカウントの指定」を参照してください。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

```
$ aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal detective.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスプリンシパルをパラメータ`account.amazonaws.com`として識別します。

Detective 用の委任管理者の無効化

委任管理者アカウントを削除する場合は、Detective コンソールまたは API を使用して、あるいは Organizations `DeregisterDelegatedAdministrator` CLI または SDK オペレーションを使用して行います。Detective コンソールまたは API、あるいは Organizations API を使用して委任管理者を削除する方法については、「Amazon Detective 管理ガイド」の「組織の Detective 管理者アカウントの指定」を参照してください。

Amazon DevOpsGuru と AWS Organizations

Amazon DevOpsGuru は、運用データとアプリケーションのメトリクスとイベントを分析し、通常の運用パターンから逸脱する動作を特定します。 DevOpsGuru が運用上の問題またはリスクを検出すると、ユーザーに通知されます。

DevOpsGuru を使用すると、 でのマルチアカウントサポートが有効になるため AWS Organizations、組織全体のインサイトを管理するメンバーアカウントを指定できます。この委任管理者は、組織内のすべてのアカウントから取得したインサイトを表示、ソート、フィルタリングして、追加のカスタマイズを必要とせずに、組織内のすべてのモニタリング対象アプリケーションの状態に関する全体像を作成できます。

詳細については、「Amazon DevOpsGuru ユーザーガイド」の「組織全体のアカウントのモニタリング」を参照してください。

Amazon DevOpsGuru を と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、 DevOpsGru は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 DevOpsGru と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

- `AWSServiceRoleForDevOpsGuru`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。 DevOpsGuru が使用するサービスにリンクされたロールは、以下のサービスプリンシパルへのアクセスを許可します。

- `devops-guru.amazonaws.com`

詳細については、「Amazon DevOps Guru ユーザーガイド」の「サービスにリンクされたロールを Guru で使用する」を参照してください。 DevOps

DevOpsGuru で信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

> **Note**
>
> Amazon DevOpsGuru の委任管理者を指定すると、 DevOpsGuru は組織の DevOpsGuru の信頼されたアクセスを自動的に有効にします。
> DevOpsGuru では、組織のこのサービスの委任管理者となるメンバーアカウントを指定する AWS Organizations 前に、 への信頼されたアクセスが必要です。

> **Important**
>
> 可能な限り、Amazon DevOpsGuru コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、Amazon DevOpsGuru はサービスに必要なリソースの作成など、必要な設定を実行できます。これらのステップは、Amazon DevOpsGuru が提供するツールを使用して統合を有効にできない場合にのみ実行してください。詳細については、この注意を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールまたは DevOpsGuru コンソールを使用します。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. サービスページで、Amazon DevOpsGuru の行を見つけ、サービスの名前を選択し、信頼されたアクセスを有効にするを選択します。
3. 確認ダイアログボックスで、[Show the option to enable trusted access] (信頼されたアクセスを有効にするオプションを表示する) を有効にし、ボックスに「**`enable`**」と入力してから、[Enable trusted access] (信頼されたアクセスを有効にする) を選択します。
4. のみの管理者である場合は AWS Organizations、Amazon DevOpsGuru の管理者に、コンソールを使用してそのサービスを有効にして と連携できるようにしました AWS Organizations。

DevOps Guru console

DevOpsGuru コンソールを使用して信頼されたサービスアクセスを有効にするには

1. 管理アカウントで管理者としてサインインし、 DevOpsGuru コンソールを開きます。Amazon DevOpsGuru コンソール
2. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。

DevOpsGuru との信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Amazon DevOpsGuru との信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

AWS Organizations コンソールを使用して、信頼されたアクセスを無効にすることができます。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon DevOpsGuru を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. Amazon DevOpsGuru の信頼されたアクセスを無効にするダイアログボックスで、無効にして確認します。次に、信頼されたアクセスを無効にするを選択します。
6. のみの管理者である場合は AWS Organizations、コンソールまたはツールを使用してそのサービスを無効にできるようになったことを Amazon DevOpsGuru の管理者に伝えます AWS Organizations。

DevOpsGuru の委任管理者アカウントの有効化

DevOpsGuru の委任管理者アカウントは、組織から DevOpsGuru にオンボーディングされているすべてのメンバーアカウントのインサイトデータを表示できます。委任管理者が組織アカウントを管理する方法については、「Amazon DevOpsGuru ユーザーガイド」の「組織全体のアカウントのモニタリング」を参照してください。

DevOpsGuru の委任管理者を設定できるのは、組織管理アカウントの管理者のみです。

委任管理者アカウントは、 DevOpsGru コンソールから、または Organizations CLI または SDK `RegisterDelegatedAdministrator` オペレーションを使用して指定できます。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内の DevOpsGuru の委任管理者としてメンバーアカウントを設定できます。

DevOps Guru console

DevOpsGuru コンソールで委任管理者を設定するには

1. 管理アカウントで管理者としてサインインし、 DevOpsGuru コンソールを開きます。Amazon DevOpsGuru コンソール
2. [Register delegated administrator (委任管理者の登録)] を選択します。管理アカウントまたは任意のメンバーアカウントのいずれかを、委任管理者として選択できます。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

```
$ aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal devops-guru.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスプリンシパルをパラメータ`account.amazonaws.com`として識別します。

DevOpsGuru の委任管理者を無効にする

委任された管理者は、 DevOpsGru コンソール、または Organizations CLI または SDK `DeregisterDelegatedAdministrator` オペレーションを使用して削除できます。 DevOpsGuru コンソールを使用して委任された管理者を削除する方法については、「Amazon DevOpsGuru ユーザーガイド」の「組織全体のアカウントのモニタリング」を参照してください。

AWS Directory Service および AWS Organizations

AWS Directory Service for Microsoft Active Directory または を使用すると AWS Managed Microsoft AD、Microsoft Active Directory (AD) をマネージドサービスとして実行できます。 AWS Directory Service は、 AWS クラウドでディレクトリを簡単にセットアップして実行したり、 AWS リソースを既存のオンプレミスの Microsoft Active Directory と接続したりできます。 AWS Managed Microsoft AD また、 と緊密に統合 AWS Organizations されているため、リージョン内の複数の VPC AWS アカウント や任意の VPC 間でディレクトリをシームレスに共有できます。詳細については、AWS Directory Service 管理ガイドを参照してください。

AWS Directory Service との統合には、以下の情報を参考にしてください AWS Organizations。

AWS Directory Serviceとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Directory Service コンソールまたは AWS Organizations コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Directory Service コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Directory Service を実行できます。ここに示す手順は、統合の有効化に AWS Directory Serviceが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。

AWS Directory Service コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS Directory Service コンソールを使用して信頼されたアクセスを有効にするには

ディレクトリを共有するには、AWS Directory Service 管理ガイドのディレクトリの共有を参照してください。ディレクトリを共有すると、信頼されたアクセスが自動的に有効になります。 step-by-step 手順については、「チュートリアル: AWS Managed Microsoft AD Directory の共有」を参照してください。

AWS Organizations コンソールを使用して、信頼されたアクセスを有効にできます。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Directory Serviceで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Directory Service」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Directory Service に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS Directory Serviceとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

の使用 AWS Organizations 中に を使用して信頼されたアクセスを無効にすると AWS Directory Service、以前に共有されたディレクトリはすべて通常どおり動作し続けます。ただし、信頼された

アクセスを再度有効化するまでは、組織内で新しいディレクトリを共有することはできなくなります。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

AWS Organizations コンソールを使用して、信頼されたアクセスを無効にすることができます。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Directory Serviceで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Directory Service」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Directory Service に、コンソールまたはツールを使用してそのサービスを無効にできるようになったことを知らせます AWS Organizations。

AWS Firewall Manager および AWS Organizations

AWS Firewall Manager は、組織内の および AWS アカウント アプリケーション全体でファイアウォールルールやその他の保護を一元的に設定および管理するために使用するセキュリティ管理サービスです。Firewall Manager を使用すると、 AWS WAF ルールのロールアウト、保護の作成 AWS Shield Advanced 、Amazon Virtual Private Cloud (Amazon VPC) セキュリティグループの設定と監査、 AWS Network Firewallのデプロイを行うことができます。Firewall Manager を使用すれば、保護を一度設定するだけで、新しいリソースやアカウントが追加されているかどうかにかかわらず、組織内のすべてのアカウントとリソースに保護が自動的に適用されます。の詳細については AWS Firewall Manager、「 AWS Firewall Manager デベロッパーガイド」を参照してください。

AWS Firewall Manager との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Firewall Manager はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Firewall Manager と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForFMS`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Firewall Manager によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `fms.amazonaws.com`

Firewall Manager との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Firewall Manager コンソールまたは AWS Organizations コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Firewall Manager コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Firewall Manager を実行できます。ここに示す手順は、統合の有効化に AWS Firewall Managerが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Firewall Manager コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS Organizations 管理アカウントでサインインし、組織内のアカウントを AWS Firewall Manager 管理者アカウントとして設定する必要があります。詳細については、AWS Firewall Manager デベロッパーガイドの Set the AWS Firewall Manager Administrator Account を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Firewall Managerで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Firewall Manager」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Firewall Manager に、コンソールを使用してそのサービスを有効にして を操作できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼できるサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Firewall Manager としてを有効にできます。

```
$ aws organizations enable-aws-service-access \
    --service-principal fms.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

Firewall Manager との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Firewall Manager または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Firewall Manager コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、 は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS Firewall Manager を実行できます。ここに示す手順は、統合の無効化に AWS Firewall Managerが提供するツールを使用できない場合にのみ実施してください。
> AWS Firewall Manager コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

Firewall Manager コンソールを使用して信頼されたアクセスを無効にするには

AWS Firewall Manager 管理者アカウントを変更または取り消すには、「 AWS Firewall Manager デベロッパーガイド」の AWS Firewall Manager 「管理者アカウントとして別のアカウントを指定する」の手順に従います。

管理者アカウントを取り消す場合は、 AWS Organizations 管理アカウントにサインインし、 の新しい管理者アカウントを設定する必要があります AWS Firewall Manager。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Firewall Managerで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Firewall Manager」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 のコンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになった AWS Firewall Manager ことを管理者に伝えます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、 Organizations の信頼されたサービス AWS Firewall Manager としてを無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal fms.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Firewall Manager 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 Firewall Manager の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Firewall Manager の管理を分離するのに有効です。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Firewall Manager の委任管理者としてメンバーアカウントを設定できます。

メンバーアカウントを組織の Firewall Manager 管理者として指定する方法については、「 AWS Firewall Manager デベロッパーガイド」のAWS Firewall Manager 「管理者アカウントの設定」を参照してください。

Amazon GuardDuty と AWS Organizations

Amazon GuardDuty は、さまざまなデータソースを分析および処理する継続的セキュリティモニタリングサービスです。脅威インテリジェンスフィードおよび機械学習を使用して、AWS 環境内の予期しないアクティビティ、不正の可能性があるアクティビティ、悪意のあるアクティビティを識別します。これには、権限のエスカレーション、公開された認証情報の使用、悪意のある IP アドレス、URL、またはドメインとの通信、Amazon Elastic Compute Cloud インスタンスおよびコンテナワークロードでのマルウェアの存在などの問題が含まれる場合があります。

Organizations を使用し、組織のすべてのアカウントを対象に GuardDuty を管理することで、GuardDuty の管理を簡素化できます。

詳細については、Amazon GuardDuty ユーザーガイドの Managing GuardDuty accounts with AWS Organizations を参照してください。

Amazon GuardDuty と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、GuardDuty はサポートされているオペレー

ションを組織内の組織アカウントで実行できます。このロールを削除できるのは、GuardDuty と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForAmazonGuardDuty` サービスにリンクされたロールは、GuardDuty をOrganizations と統合したアカウントで自動的に作成されます。詳細については、Amazon GuardDuty ユーザーガイドの Managing GuardDuty accounts with Organizations を参照してください。
- `AmazonGuardDutyMalwareProtectionServiceRolePolicy` サービスにリンクされたロールは、GuardDuty Malware Protection が有効になっているアカウントで自動的に作成されます。詳細については、Amazon GuardDuty ユーザーガイドの GuardDuty マルウェア保護のためのサービスにリンクされたロールの許可を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

- `guardduty.amazonaws.com`、`AWSServiceRoleForAmazonGuardDuty` サービスにリンクされたロールによって使用される。
- `malware-protection.guardduty.amazonaws.com`、`AmazonGuardDutyMalwareProtectionServiceRole` サービスにリンクされたロールによって使用される。

GuardDuty との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Amazon GuardDuty だけで、信頼されたアクセスを有効にできます。

組織の GuardDuty の委任管理者にするメンバーアカウントを指定するにあたり、Amazon GuardDuty には AWS Organizations への信頼されたアクセスが必要です。GuardDuty コンソールを使用して委任管理者を設定すると、信頼されたアクセスが GuardDuty によって自動的に有効になります。

ただし、AWS CLI または AWS SDK を使用して委任管理者アカウントを設定する場合は、明示的に EnableAWSServiceAccess オペレーションを呼び出し、サービスプリンシパルをパラメーターとして渡します。次に、EnableOrganizationAdminAccount を呼び出し、GuardDuty の管理者アカウントを委任します。

GuardDuty との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、Organizations で信頼されたサービスとして Amazon GuardDuty を無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal guardduty.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

GuardDuty 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、GuardDuty の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から GuardDuty の管理を分離するのに有効です。

> **最小アクセス許可**
>
> メンバーアカウントを委任管理者として指定するために必要なアクセス許可については、Amazon GuardDuty ユーザーガイドの委任された管理者の指定に必要な権限を参照してください。

GuardDuty の委任管理者としてメンバーアカウントを指定するには

Designate a delegated administrator and add member accounts (console)、および Designate a delegated administrator and add member accounts (API) を参照してください。

AWS Health および AWS Organizations

AWS Health は、リソースのパフォーマンスと、 AWS サービスとアカウントの可用性を継続的に可視化します。 は、 AWS リソースとサービスが問題の影響を受けた場合、または今後の変更の影響を受ける場合にイベント AWS Health を提供します。組織ビューを有効にすると、組織の管理アカウントのユーザーは、組織内のすべてのアカウントで AWS Health イベントを集約できます。組織ビューには、機能が有効になった後に配信された AWS Health イベントのみが表示され、90 日間保持されます。

組織ビューを有効にするには、 AWS Health コンソール、 (AWS CLI)、 AWS Command Line Interface または AWS Health API を使用します。

詳細については、「 AWS Health ユーザーガイド」の AWS Health 「イベントの集約」を参照してください。

AWS Health との統合には、以下の情報を参考にしてください AWS Organizations。

統合用のサービスにリンクされたロール

`AWSServiceRoleForHealth_Organizations` サービスにリンクされたロールにより AWS Health 、 は、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールは、 EnableHealthServiceAccessForOrganization API オペレーションを呼び出して信頼されたアクセスを有効にすると、組織の管理アカウントに自動的に作成されます。それ以外の場合は、「IAM ユーザーガイド」の「サービスにリンクされたロールの作成」の説明に従って、 AWS Health コンソール、API、または CLI を使用してロールを作成します。 https://docs.aws.amazon.com/IAM/latest/UserGuide/using-service-linked-roles.html#create-service-linked-role

このロールを削除または変更できるのは、 AWS Health と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス AWS Health を許可します。

- `health.amazonaws.com`

AWS Healthとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

の組織ビュー機能を有効にすると AWS Health、信頼されたアクセスも自動的に有効になります。

信頼された AWS Organizations アクセスは、 AWS Health コンソールまたは コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Health コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Health を実行できます。ここに示す手順は、統合の有効化に AWS Health が提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Health コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS Health コンソールを使用して信頼されたアクセスを有効にするには

と次のいずれかのオプションを使用して AWS Health 、信頼されたアクセスを有効にできます。

- AWS Health コンソールを使用します。詳細については、AWS Health ユーザーガイドの組織ビュー (コンソール) を参照してください。
- AWS CLIを使用します。詳細については、AWS Health ユーザーガイドの組織ビュー (CLI) を参照してください。

- EnableHealthServiceAccessForOrganization API オペレーションを呼び出します。

信頼されたアクセスを有効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Health として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal health.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

AWS Healthとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

組織ビュー機能を無効にすると、 は組織内の他のすべてのアカウントのイベントの集約を AWS Health 停止します。また、信頼されたアクセスは自動的に無効になります。

信頼されたアクセスは、 AWS Health または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Health コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、 は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS Health を実行できます。ここに示す

手順は、統合の無効化に AWS Healthが提供するツールを使用できない場合にのみ実施してください。
AWS Health コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

AWS Health コンソールを使用して信頼されたアクセスを無効にするには

信頼されたアクセスを無効にする方法には次のようなものがあります。

- AWS Health コンソールを使用します。詳細については、AWS Health ユーザーガイドの組織ビューの無効化 (コンソール) を参照してください。
- AWS CLIを使用します。詳細については、AWS Health ユーザーガイドの組織ビューの無効化 (CLI) を参照してください。
- DisableHealthServiceAccessForOrganization API オペレーションを呼び出します。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Health として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal health.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

の委任管理者アカウントの有効化 AWS Health

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 AWS Health の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、 AWS Healthの管理から 組織の管理を分離するのに有効です。

メンバーアカウントを AWS Health の委任管理者として指定するには

「組織ビューに委任管理者を登録する」を参照してください。

AWS Health の委任管理者を削除するにほ

「組織ビューから委任管理者を削除する」を参照してください。

Amazon Inspector および AWS Organizations

Amazon Inspector は、Amazon EC2 とコンテナのワークロードを継続的にスキャンして、ソフトウェアの脆弱性や意図しないネットワークの公開 (ネットワークエクスポージャー) を検出する、自動化された脆弱性管理サービスです。

Amazon Inspector を使用して Amazon Inspector 用の管理者アカウントを委任するだけで、AWS Organizations を介して関連付けられた複数のアカウントを管理できます。委任管理者は、組織の Amazon Inspector を管理し、組織に代わって次のようなタスクを実行するための特別なアクセス許可が付与されます。

- メンバーアカウントへのスキャンを有効または無効にする
- 組織全体の集約された調査結果データを表示する
- 抑制ルールを作成して管理する

詳細については、「Amazon Inspector ユーザーガイド」の「AWS Organizations で複数のアカウントを管理する」を参照してください。

Amazon Inspector と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Amazon Inspector は、サポートされているオペレーションを組織内の組織のアカウントで実行できます。

このロールを削除または変更できるのは、Amazon Inspector と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForAmazonInspector2`

詳細については、「Amazon Inspector ユーザーガイド」の「Amazon Inspector でのサービスにリンクされたロールの使用」を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Amazon Inspector によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `inspector2.amazonaws.com`

Amazon Inspector との信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

組織でこのサービスの委任管理者にするメンバーアカウントを指定するにあたり、Amazon Inspector には AWS Organizations への信頼されたアクセスが必要です。

Amazon Inspector の委任管理者を指定すると、組織の Amazon Inspector に対する信頼されたアクセスが自動的に有効になります。

ただし、AWS CLI または AWS SDK のいずれかを使用して委任管理者アカウントを設定する場合は、明示的に `EnableAWSServiceAccess` オペレーションを呼び出し、サービスプリンシパルをパラメータとして指定します。次に、`EnableDelegatedAdminAccount` を呼び出して Inspector 管理者アカウントを委任します。

信頼されたアクセスの有効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、Amazon Inspector を Organizations で信頼されたサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal inspector2.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: EnableAWSServiceAccess

> **Note**
>
> `EnableAWSServiceAccess` API を使用している場合、`EnableDelegatedAdminAccount` も呼び出して Inspector 管理者アカウントを委任する必要があります。

Amazon Inspector との信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

AWS Organizations 管理アカウントの管理者だけが Amazon Inspector との信頼されたアクセスを無効にできます。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、Organizations で信頼されたサービスとして Amazon Inspector を無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal inspector2.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: DisableAWSServiceAccess

Amazon Inspector 用の委任管理者アカウントの有効化

Amazon Inspector では、AWS Organizations サービスで委任された管理者を使用して、組織内の複数のアカウントを管理できます。

AWS Organizations 管理アカウントは、組織内のアカウントを Amazon Inspector 用の委任管理者アカウントとして指定します。委任管理者は、組織の Amazon Inspector を管理し、組織に代わってタスクを実行するための特別なアクセス許可が付与されます。タスクには、メンバーアカウントのスキャンの有効化または無効化、組織全体の集約された調査結果データの表示、抑制ルールの作成および管理などが含まれます

委任管理者が組織アカウントを管理する方法については、「Amazon Inspector ユーザーガイド」の「管理者とメンバーアカウントの関係について」を参照してください。

Amazon Inspector 用の委任管理者を設定できるのは、組織管理アカウントの管理者だけです。

委任管理者アカウントを指定する場合は、Amazon Inspector コンソールまたは API を介して、あるいは Organizations CLI または SDK オペレーションを使用して行います。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Amazon Inspector の委任管理者としてメンバーアカウントを設定できます

Amazon Inspector コンソールを使用して委任管理者を設定するには、「Amazon Inspector ユーザーガイド」の「ステップ 1: Amazon Inspector を有効にする – Multi-account environment」を参照してください。

> **Note**
>
> Amazon Inspector を使用する各リージョンで、`inspector2:enableDelegatedAdminAccount` を呼び出す必要があります。

AWS CLI, AWS API

AWS CLI または AWS SDK を使用して委任管理者アカウントを設定するには、次のコマンドを使用します。

- AWS CLI:

```
$ aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal inspector2.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator` オペレーションおよびメンバーアカウントの ID 番号を呼び出して、アカウントサービスプリンシパル `account.amazonaws.com` をパラメータとして識別します。

Amazon Inspector 用の委任管理者の無効化

組織から委任管理者アカウントを削除できるのは、AWS Organizations 管理アカウントの管理者だけです。

委任管理者を削除する場合は、Amazon Inspector コンソールまたは API を介して、あるいは Organizations `DeregisterDelegatedAdministrator` CLI または SDK オペレーションを使用して行います。Amazon Inspector コンソールを使用して委任管理者を削除するには、「Amazon Inspector ユーザーガイド」の「委任管理者の削除」を参照してください。

AWS License Manager および AWS Organizations

{AWS License Manager}は、ソフトウェアベンダーのライセンスをクラウドに移動するプロセスを効率化します。AWS でクラウドインフラストラクチャを構築するときに、Bring-Your-Own-License (BYOL) による機会を使用してコストを削減できます。つまり、クラウドリソースで使用するように既存のライセンスインベントリの用途を変更します。管理者は、ライセンスの使用でルールベースのコントロールを使用することにより、新規および既存のクラウドのデプロイにソフト制限またはハード制限を設定し、準拠していないサーバーの使用を事前に停止できます。

License Manager について詳しくは、License Manager ユーザーガイドを参照してください。

License Manager を AWS Organizations とリンクすると、次のことが可能になります。

- 組織全体でコンピューティングリソースのクロスアカウントの検出を有効にすることができます。
- 所有して AWS に実行している商用 Linux サブスクリプションを表示および管理します。詳細については、「AWS License Manager の Linux サブスクリプション」を参照してください。

AWS License Manager と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、次の「サービスにリンクされたロール」が組織の管理アカウントに自動的に作成されます。これらのロールにより、License Manager はサポートされているオペレーションを組織内のアカウントで実行できます。

ロールを削除または変更できるのは、License Manager と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合のみです。

- `AWSLicenseManagerMasterAccountRole`
- `AWSLicenseManagerMemberAccountRole`
- `AWSServiceRoleForAWSLicenseManagerRole`
- `AWSServiceRoleForAWSLicenseManagerLinuxSubscriptionsService`

詳細については、「License Manager における管理アカウントロール」、「License Manager におけるメンバーアカウントロール」、「License Manager における Linux サブスクリプションロール」を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。License Manager によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `license-manager.amazonaws.com`
- `license-manager.member-account.amazonaws.com`

- `license-manager-linux-subscriptions.amazonaws.com`

License Manager との信頼されたアクセスの有効化

AWS License Manager だけで、信頼されたアクセスを有効にできます。

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

License Manager との信頼されたアクセスを有効にするには

AWS Organizations 管理アカウントを使用して License Manager コンソールにサインインし、その管理アカウントを License Manager アカウントに関連付ける必要があります。詳細については、「AWS License Manager の設定」を参照してください。

License Manager との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

Organizations の AWS CLI コマンドを実行するか、いずれかの AWS SDK で Organizations API オペレーションを呼び出すことにより、信頼されたアクセスを無効にできます。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、AWS License Manager を Organizations で信頼されたサービスとして無効にすることができます。

```
$ aws organizations disable-aws-service-access \
    --service-principal license-manager.amazonaws.com
```

  このコマンドが成功した場合、出力は生成されません。

Linux サブスクリプション用の信頼されたアクセスを無効にする手順は、次のとおりです。

```
$  aws organizations disable-aws-service-access \
    --service-principal license-manager-linux-subscriptions.amazonaws.com
```

- AWS API: DisableAWSServiceAccess

License Manager 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 License Manager の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から License Manager の管理を分離するのに有効です。

License Manager の管理者をメンバーアカウントに委任するには、License Manager ユーザーガイドの委任された管理者の登録を参照してください。

Amazon Macie と AWS Organizations

Amazon Macie は、フルマネージド型のデータセキュリティおよびデータプライバシーサービスです。機械学習とパターンマッチングを使用して、Amazon Simple Storage Service (Amazon S3) 内の機密データを検出、モニタリングし、適切な保護を支援します。Macie は、組織が Amazon S3 に保存している個人を特定できる情報 (PII) や知的財産などの機密データを詳細に把握できるよう、そうしたデータの検出を自動化します。

詳細については、Amazon Macie ユーザーガイドの「Managing Amazon Macie accounts with AWS Organizations」を参照してください。

Amazon Macie と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の委任された Macie 管理アカウントに自動的に作成されます。このロールにより、Macie はサポートされているオペレーションを組織内のアカウントに対して実行できます。

このロールを削除できるのは、Macie と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleRorAmazonMacie`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Macie によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `macie.amazonaws.com`

Macie との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Amazon Macie コンソールまたは AWS Organizations コンソールを使用して、信頼されたアクセスを有効にできます。

> **Important**
>
> Organizations との統合の有効化には、可能な場合は常に Amazon Macie のコンソールまたはツールを使用することを強くお勧めします。そうすることで、サービスに必要なリソースの作成など、必要な構成が Amazon Macie で実行可能になります。ここに示す手順は、統合の有効化に Amazon Macie が提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> Amazon Macie のコンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実施する必要はありません。

Macie コンソールを使用して信頼されたアクセスを有効にするには

組織の Amazon Macie の委任管理者にするメンバーアカウントを指定するにあたり、Amazon Macie には AWS Organizations への信頼されたアクセスが必要です。Macie マネジメントコンソールを使用して委任管理者を設定すると、信頼されたアクセスが Macie によって自動的に有効になります。

詳細については、Amazon Macie ユーザーガイドの「Integrating and configuring an organization in Amazon Macie」を参照してください。

信頼されたアクセスの有効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、Amazon Macie を Organizations で信頼されたサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal macie.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: EnableAWSServiceAccess

Macie 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 Macie の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Macie の管理を分離するのに有効です。

最小アクセス許可

次の許可を持つ Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Macie の委任管理者としてメンバーアカウントを設定できます。

- `organizations:EnableAWSServiceAccess`
- `macie:EnableOrganizationAdminAccount`

メンバーアカウントを Macie の委任管理者として指定するには

組織の Amazon Macie の委任管理者にするメンバーアカウントを指定するにあたり、Amazon Macie には AWS Organizations への信頼されたアクセスが必要です。Macie マネジメントコンソールを使用して委任管理者を設定すると、信頼されたアクセスが Macie によって自動的に有効になります。

詳細については、「https://docs.aws.amazon.com/macie/latest/user/macie-organizations.html#register-delegated-admin」を参照してください

AWS Marketplace および AWS Organizations

AWS Marketplace は、ソリューションの構築とビジネスの実行に必要なサードパーティーのソフトウェア、データ、サービスを検索、購入、デプロイ、管理するために使用できる厳選されたデジタルカタログです。

AWS Marketplace は、 での購入 AWS License Manager に を使用してライセンスを作成および管理します AWS Marketplace。組織内の他のアカウントとライセンスを共有 (アクセスを許可) すると、AWS Marketplace はそのアカウント用に新しいライセンスを作成し、管理します。

詳細については、AWS Marketplace 購入者ガイドの Service-linked roles for AWS Marketplace を参照してください。

AWS Marketplace との統合には、次の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより AWS Marketplace 、 は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 AWS Marketplace と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForMarketplaceLicenseManagement`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス AWS Marketplace を許可します。

- `license-management.marketplace.amazonaws.com`

AWS Marketplaceとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Marketplace コンソールまたは AWS Organizations コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Marketplace コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Marketplace を実行できます。ここに示す手順は、統合の有効化に AWS Marketplaceが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Marketplace コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS Marketplace コンソールを使用して信頼されたアクセスを有効にするには

「AWS Marketplace 購入者ガイド」の「AWS Marketplaceのサービスにリンクされたロールの作成」を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Marketplaceで を選択します。

4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Marketplace」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Marketplace に、コンソールを使用してそのサービスを有効にし、 と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Marketplace として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal license-management.marketplace.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

AWS Marketplaceとの信頼されたアクセスの無効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Marketplace として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal license-management.marketplace.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

AWS Marketplace Private Marketplace と AWS Organizations

AWS Marketplace は、ソリューションの構築とビジネスの実行に必要なサードパーティーのソフトウェア、データ、サービスを検索、購入、デプロイ、管理するために使用できる厳選されたデジタルカタログです。プライベートマーケットプレイスでは、 で利用可能な製品の幅広いカタログと AWS Marketplace、それらの製品のきめ細かな制御が可能です。

AWS Marketplace Private Marketplace を使用すると、組織全体、1 つ以上の OUs、または組織内の 1 つ以上のアカウントに関連付けられた複数のプライベートマーケットプレイスエクスペリエンスを作成し、それぞれに独自の承認済み製品のセットを作成できます。 AWS 管理者は、会社またはチームのロゴ、メッセージング、カラースキームに関する各プライベートマーケットプレイスエクスペリエンスに会社のブランドを適用することもできます。

詳細については、「 AWS Marketplace 購入者ガイド」の「 でロールを使用して Private Marketplace を設定する AWS Marketplace」を参照してください。

Private Marketplace を AWS Marketplace と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

AWS Marketplace Private Marketplace コンソールを使用して信頼されたアクセスを有効にすると、次のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、 Private Marketplace は、組織内の組織のアカウント内でサポートされているオペレーション

を実行できます。Private Marketplace と Organizations AWS Marketplace 間の信頼されたアクセスを無効にし、組織内のすべてのプライベートマーケットプレイスエクスペリエンスの関連付けを解除する場合にのみ、このロールを削除または変更できます。

Organizations コンソール、CLI、または SDK から直接信頼されたアクセスを有効にした場合、サービスにリンクされたロールは自動的に作成されません。

- `AWSServiceRoleForPrivateMarketplaceAdmin`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Private Marketplace で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセスを許可します。

- `private-marketplace.marketplace.amazonaws.com`

Private Marketplace での信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Private Marketplace コンソールまたは AWS Marketplace コンソールを使用して、信頼された AWS Organizations アクセスを有効にできます。

> **Important**
>
> 可能な限り、 Private Marketplace AWS Marketplace コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 AWS Marketplace Private Marketplace は、サービスに必要なリソースの作成など、必要な設定を実行できます。 AWS Marketplace Private Marketplace が提供するツールを使用して統合を有効にできない場合にのみ、これらのステップを実行します。詳細については、この注意を参照してください。
>
> AWS Marketplace Private Marketplace コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

Private Marketplace コンソールを使用して信頼されたアクセスを有効にするには

AWS Marketplace 購入者ガイドの「Private Marketplace の開始方法」を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで AWS Marketplace Private Marketplace を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. AWS Marketplace Private Marketplace の信頼されたアクセスを有効にするダイアログボックスで、確認のために enable と入力し、信頼されたアクセスを有効にする を選択します。
6. のみの管理者である場合は AWS Organizations、Private Marketplace AWS Marketplace の管理者に、コンソールを使用してそのサービスを有効にして と連携できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、 Private Marketplace AWS Marketplace を Organizations の信頼されたサービスとして有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal private-marketplace.marketplace.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

Private Marketplace での信頼されたアクセスの無効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にできます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービスとして AWS Marketplace Private Marketplace を無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal private-marketplace.marketplace.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Private Marketplace の委任管理者アカウントの有効化

管理アカウント管理者は、委任された管理者と呼ばれる指定されたメンバーアカウントに Private Marketplace の管理権限を委任できます。アカウントをプライベートマーケットプレイスの委任管理者として登録するには、管理アカウント管理者が信頼されたアクセスとサービスにリンクされたロールが有効になっていることを確認し、新しい管理者の登録 を選択し、12 桁の AWS アカウント番号を指定し、送信 を選択する必要があります。

管理アカウントと委任された管理者アカウントは、エクスペリエンスの作成、ブランド設定の更新、対象者の関連付けまたは関連付け解除、製品の追加または削除、保留中のリクエストの承認または拒否などの Private Marketplace 管理タスクを実行できます。

Private Marketplace コンソールを使用して委任管理者を設定するには、「 AWS Marketplace 購入者ガイド」の「プライベートマーケットプレイスの作成と管理」を参照してください。

Organizations `RegisterDelegatedAdministrator` API を使用して、委任された管理者を設定することもできます。詳細については、 RegisterDelegatedAdministrator「 Organizations コマンドリファレンス」の「」を参照してください。

Private Marketplace の委任管理者を無効にする

Private Marketplace の委任管理者を設定できるのは、組織管理アカウントの管理者のみです。

委任された管理者は、Private Marketplace コンソールまたは API を使用するか、Organizations CLI または SDK `DeregisterDelegatedAdministrator` オペレーションを使用して削除できます。

Private Marketplace コンソールを使用して委任管理者 Private Marketplace アカウントを無効にするには、「 AWS Marketplace 購入者ガイド」の「プライベートマーケットプレイスの作成と管理」を参照してください。

AWS Network Manager と AWS Organizations

Network Manager を使用すると、 AWS アカウント、リージョン、オンプレミスロケーション間で AWS Cloud WAN コアネットワークと Transit Gateway ネットワークを AWS 一元管理できます。マルチアカウントサポートを使用すると、任意の AWS アカウントに対して単一のグローバルネットワークを作成し、Network Manager コンソールを使用して複数のアカウントからグローバルネットワークにトランジットゲートウェイを登録できます。

Network Manager と Organizations 間の信頼されたアクセスを有効にすると、登録された委任管理者と管理アカウントは、メンバーアカウントに展開されたサービスリンクの役割を利用して、グローバルネットワークに接続されたリソースを説明できます。Network Manager コンソールから、登録された委任管理者と管理アカウントは、メンバーアカウントに展開されたカスタム IAM ロール (マルチアカウントの監視とイベント、およびマルチアアクウントリソースの表示と管理のためのコンソールスイッチのロールアクセス) を引き受けることができます。

> **Important**
>
> - Network Manager コンソールを使用してマルチアカウント設定 (信頼されたアクセスの有効化/無効化、委任された管理者の登録/解除)を管理することを強くお勧めします。コンソールからこれらの設定を管理すると、マルチアカウント・アクセスに必要なメンバーア

カウントに、必要なすべてのサービスにリンクされたロールとカスタム IAM ロールが自動的に配備され、管理されます。

- Network Manager コンソールで Network Manager の信頼されたアクセスを有効にすると、コンソールでも AWS CloudFormation StackSets サービスが有効になります。Network Manager は を使用して StackSets 、マルチアカウント管理に必要なカスタム IAM ロールをデプロイします。

Network Manager とOrganizations の統合の詳細については、「Amazon VPC User Guide」の「ネットワークマネージャで複数のアカウントを管理する」を参照してください。

AWS Network Manager を と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、リストされた組織アカウントに以下のようなサービスにリンクされたロールが自動的に作成されます。これらのロールにより、Nettwork Managerは組織のアカウント内でサポートされている操作を実行できます。信頼されたアクセスを無効にした場合、Network Manager は組織内のアカウントからこれらのロールを削除しません。IAM コンソールを使用して手動で削除できます。

管理アカウント

- `AWSServiceRoleForNetworkManager`
- `AWSServiceRoleForCloudFormationStackSetsOrgAdmin`
- `AWSServiceRoleForCloudWatchCrossAccount`

メンバーアカウント

- `AWSServiceRoleForNetworkManager`
- `AWSServiceRoleForCloudFormationStackSetsOrgMember`

メンバーアカウントを委任された管理者として登録すると、委任された管理者アカウントに次の追加ロールが自動的に作成されます。

- `AWSServiceRoleForCloudWatchCrossAccount`

サービスにリンクされたロールで使用されるサービスプリンシパル

サービスにリンクされたロールは、そのロールに定義された信頼関係によって承認されたサービスプリンシパルのみが担うことができる。

- ロールの場合、アクセス権を持つ唯一のサービスプリンシパルです。
- サービスにリンクされたロールの場合、アクセス権を持つ唯一のサービスプリンシパルです。
- サービスにリンクされたロールの場合、アクセス権を持つ唯一のサービスプリンシパルです。
- サービスにリンクされたロールの場合、アクセス権を持つ唯一のサービスプリンシパルです。

これらのロールを削除すると、ネットワーク・マネージャのマルチ・アカウント機能が損なわれます。

ネットワーク・マネージャーで信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations 管理アカウントの管理者のみが、別の AWS サービスで信頼されたアクセスを有効にするアクセス許可を持っています。権限の問題を回避するために、ネットワーク・マネージャを必ず使用して、信頼されたアクセスを有効にします。詳細については、「Amazon VPC ユーザーガイド」の 「Manage multiple accounts in Network Manager with」を参照してください。

Network Manager との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Organizations 管理アカウントの管理者のみが、別の AWS サービスへの信頼されたアクセスを無効にするアクセス許可を持っています。

> **Important**
>
> 信頼されたアクセスを無効にするには、Network Manager コンソールを使用することを強くお勧めします。、API、 AWS CloudFormation コンソールなど AWS CLIで信頼されたアクセスを無効にすると、デプロイされた IAM ロール AWS CloudFormation StackSets とカスタム IAM ロールが適切にクリーンアップされない可能性があります。信頼されたサービスのアクセスを無効にするには、Network Manager コンソールにサイン・インします。

Network Manager 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、 Network Manager の管理アクションを実行できるようになります。それ以外の場合は、この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、Network Manager の管理を組織の管理から分離するのに有効です。

メンバーアカウントを組織の StackSets の委任管理者として指定する手順については、 ユーザーガイドの委任された管理者の登録を参照してください。

Amazon Q デベロッパーと AWS Organizations

Amazon Q Developer は、生成人工知能 (AI) を活用した会話アシスタントで、 AWS アプリケーションの理解、構築、拡張、運用に役立ちます。また、コードのレコメンデーションをリアルタイムで提供する、汎用の機械学習によるコードジェネレーターでもあります。Amazon Q Developer の有料サブスクリプションバージョンには、Organizations の統合が必要です。詳細については、「Amazon Q ユーザーガイド」の「アカウント、IAM Identity Center、および Organizations の設定」を参照してください。

Amazon Q Developer を と統合するには、次の情報を使用します AWS Organizations。

サービスリンクロール

`AWSServiceRoleForAmazonQDeveloper` サービスにリンクされたロールにより、Amazon Q Developer はサポートされているオペレーションを組織内で実行できます。IAM ユーザーガイドの「サービスにリンクされたロールの作成」の説明に従って、Amazon Q デベロッパーコンソール、API、または CLI を使用してロールを作成します。 https://docs.aws.amazon.com/IAM/latest/UserGuide/

メンバーアカウントを使用している場合は、Amazon Q Developer and Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合にのみ、このロールを削除または変更できます。

Amazon Q Developer で使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Amazon Q Developer で使用されるサービスにリンクされたロールは、以下のサービスプリンシパルへのアクセスを許可します。

- q.amazonaws.com

Amazon Q Developer での信頼されたアクセスの有効化

Amazon Q Developer Pro は、信頼されたアクセスを使用して、Organizations 管理アカウントで行われた設定を同じ組織のメンバーアカウントと共有します。

例えば、Organizations 管理アカウントで作業する Amazon Q Developer Pro 管理者は、コード参照を使用して提案を有効にすることができます。信頼されたアクセスが有効になっている場合、その組織内のすべてのメンバーアカウントに対してコード参照を含む提案も有効になります。

Amazon Q Developer のみを使用して、信頼されたアクセスを有効にできます。

Amazon Q Developer の信頼されたアクセスを有効にするには、この手順を使用します。

1. Amazon Q デベロッパー設定ページのメンバーアカウント設定 で、編集 を選択します。
2. ポップアップウィンドウで、 を選択します。
3. [保存] を選択します。

詳細については、「Amazon Q デベロッパーユーザーガイド」の「信頼されたアクセスの有効化」を参照してください。

Amazon Q Developer での信頼されたアクセスの無効化

Amazon Q デベロッパーツールのみを使用して、信頼されたアクセスを無効にできます。

Amazon Q Developer の信頼されたアクセスを無効にするには、この手順を使用します。

1. Amazon Q デベロッパー設定 ページのメンバーアカウント設定 で、編集 を選択します。
2. ポップアップウィンドウで、オフ を選択します。
3. [保存] を選択します。

詳細については、「Amazon Q デベロッパーユーザーガイド」の「信頼されたアクセスの有効化」を参照してください。

AWS Resource Access Manager および AWS Organizations

AWS Resource Access Manager （ AWS RAM) を使用すると、所有している指定された AWS リソースを他の と共有できます AWS アカウント。これは、複数のアカウント間でさまざまなタイプ

の AWS リソースを共有するための一貫したエクスペリエンスを提供する一元化されたサービスです。

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

AWS Resource Access Manager との統合には、次の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより AWS RAM 、 は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 AWS RAM と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForResourceAccessManager`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス AWS RAM を許可します。

- `ram.amazonaws.com`

AWS RAMとの信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Resource Access Manager コンソールまたは AWS Organizations コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Resource Access Manager コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Resource Access Manager を実行できます。

ここに示す手順は、統合の有効化に AWS Resource Access Managerが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。AWS Resource Access Manager コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS RAM コンソールまたは CLI を使用して信頼されたアクセスを有効にするには

AWS RAM ユーザーガイドの Enable Sharing with AWS Organizations を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Resource Access Managerで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Resource Access Manager」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Resource Access Manager に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼できるサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

次のコマンドを実行して、Organizations の信頼されたサービス AWS Resource Access Manager として を有効にできます。

```
$ aws organizations enable-aws-service-access \
    --service-principal ram.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

AWS RAMとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Resource Access Manager または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Resource Access Manager コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、 は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS Resource Access Manager を実行できます。ここに示す手順は、統合の無効化に AWS Resource Access Managerが提供するツールを使用できない場合にのみ実施してください。AWS Resource Access Manager コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

AWS Resource Access Manager コンソールまたは CLI を使用して信頼されたアクセスを無効にするには

AWS RAM ユーザーガイドの Enable Sharing with AWS Organizations を参照してください。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Resource Access Managerで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Resource Access Manager」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Resource Access Manager に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Resource Access Manager として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal ram.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

AWS Resource Explorer および AWS Organizations

AWS Resource Explorer は、リソースの検索および検出サービスです。Resource Explorer では、インターネット検索エンジンのようなエクスペリエンスを使用して、Amazon Elastic Compute Cloud インスタンス、Amazon Kinesis Data Streams、または Amazon DynamoDB テーブルなどのリソースを調べることができます。リソースは、名前、タグ、および ID などのリソースメタデータを使用して検索できます。Resource Explorer は、クロスリージョンワークロードをシンプル化するために、アカウント内の AWS リージョン全体で動作します。

Resource Explorer をAWS Organizations に統合すると、評価対象になっている組織の AWS アカウントを複数含めることで、より広範な情報源からエビデンスを収集できます。

AWS Resource Explorer と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールは、組織内のアカウントでサポートされている操作を Resource Explorer が実行できるようにします。

このロールを削除または変更できるのは、Resource Explorer と Organizations 間における信頼されたアクセスを無効にした場合か、組織からメンバーアカウントを削除した場合のみです。

Resource Explorer がこのロールを使用する方法の詳細については、「AWS Resource Explorer ユーザーガイド」の「サービスリンクロールの使用」を参照してください。

- `AWSServiceRoleForResourceExplorer`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Resource Explorer が使用するサービスリンクロールは、次のサービスプリンシパルにアクセス許可を付与します。

- `resource-explorer-2.amazonaws.com`

AWS Resource Explorer との信頼されたアクセスを有効にする

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Resource Explorer では、組織の委任管理者になるメンバーアカウントを指定する前に、AWS Organizations への信頼されたアクセスが必要になります。

信頼されたアクセスは、Resource Explorer コンソールまたは Organizations コンソールを使用して有効にできます。可能な場合は常に、Resource Explorer のコンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。そうすることで、サービスに必要なリソースの作成など、必要な構成が AWS Resource Explorer で実行可能になります。

Resource Explorer コンソールを使用して信頼されたアクセスを有効にする

信頼されたアクセスを有効にする手順については、「AWS Resource Explorer ユーザーガイド」の「Resource Explorer の使用に対する前提条件」を参照してください。

> **Note**
>
> AWS Resource Explorer コンソールを使用して委任管理者を設定する場合は、信頼されたアクセスは AWS Resource Explorer によって自動的に有効になります。

信頼されたアクセスの有効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、AWS Resource Explorer を Organizations で信頼されたサービスとして有効にすることができます。

```
$ aws organizations enable-aws-service-access \
```

```
    --service-principal resource-explorer-2.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: EnableAWSServiceAccess

Resource Explorer との信頼されたアクセスを無効にする

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

AWS Organizations 管理アカウントの管理者だけが AWS Resource Explorer との信頼されたアクセスを無効にできます。

AWS Resource Explorer または AWS Organizations ツールを使用して信頼されたアクセスを無効にできます。

> **Important**
>
> Organizations との統合の無効化には、可能な場合は常に AWS Resource Explorer コンソールまたはツールを使用することを強くお勧めします。そうすることで、AWS Resource Explorer は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップを実行できます。ここに示す手順は、統合の無効化に AWS Resource Explorer が提供するツールを使用できない場合にのみ実施してください。
> AWS Resource Explorer コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実施する必要はありません。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

次のコマンドを実行し、AWS Resource Explorer を Organizations で信頼されたサービスとして無効にすることができます。

```
$ aws organizations disable-aws-service-access \
    --service-principal resource-explorer-2.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

Resource Explorer 用の委任管理者アカウントの有効化

委任管理者アカウントを使用してマルチアカウントリソースビューを作成し、組織単位または組織全体にスコープします。マルチアカウントビューは、リソース共有を作成することで、AWS Resource Access Manager 経由で組織内の任意のアカウントと共有できます。

> **最小アクセス許可**
>
> 組織内の Resource Explorer 用の委任管理者としてメンバーアカウントを設定できるのは、以下の許可を持つ Organizations 管理アカウントのユーザーまたはロールのみです。`resource-explorer:RegisterAccount`

Resource Explorer 用の委任管理者アカウントを有効にする手順については、「AWS Resource Explorer ユーザーガイド」の「セットアップ」を参照してください。

AWS Resource Explorer コンソールを使用して委任管理者を設定する場合は、Resource Explorer がユーザーに代わって信頼されたアクセスを自動的に有効化します。

AWS CLI, AWS API

AWS CLI または AWS SDK を使用して委任管理者アカウントを設定するには、次のコマンドを使用します。

- AWS CLI:

```
$ aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal resource-explorer-2.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator` オペレーションおよびメンバーアカウントの ID 番号を呼び出して、アカウントサービス `resource-explorer-2.amazonaws.com` をパラメータとして識別します。

Resource Explorer 用の委任管理者の無効化

Resource Explorer 用の委任管理者を削除できるのは、Organizations の管理アカウント、または Resource Explorer の委任管理者アカウントの管理者のみです。信頼されたアクセスは、Organizations `DeregisterDelegatedAdministrator` CLI または SDK 操作を使用して無効にできます。

AWS Security Hub および AWS Organizations

AWS Security Hub は、 のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準とベストプラクティスに照らして環境をチェックするのに役立ちます。

Security Hub は、 全体 AWS アカウント、使用する AWS サービス、およびサポートされているサードパーティーパートナー製品からセキュリティデータを収集します。セキュリティの傾向を分析し、特に優先度の高いセキュリティ問題を特定するのに役立ちます。

Security Hub と の両方を同時に使用すると AWS Organizations 、追加された新しいアカウントを含め、すべてのアカウントに対して Security Hub を自動的に有効にできます。これにより、Security Hub のチェックと検出がより広範囲に行えるようになり、セキュリティ体制の全体像をより包括的かつ正確に把握できます。

Security Hub について詳しくは、AWS Security Hub ユーザーガイドを参照してください。

AWS Security Hub との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Security Hub はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Security Hub と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForSecurityHub`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Security Hub によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `securityhub.amazonaws.com`

Security Hub との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Security Hub に委任管理者を指定すると、Security Hub に対する信頼されたアクセスが Security Hub によって自動的に有効になります。

Security Hub での信頼されたアクセスの無効化

信頼されたアクセスを無効にするために必要なアクセス許可の詳細については、「 AWS Organizations ユーザーガイド」の「信頼されたアクセスを無効にするために必要なアクセス許可」を参照してください。

信頼されたアクセスを無効にする前に、必要に応じて組織の委任管理者に連絡して、メンバーアカウントの Security Hub を無効にし、それらのアカウントの Security Hub リソースをクリーンアップしてください。

信頼されたアクセスを無効にするには、 AWS Organizations コンソール、Organizations API、または を使用します AWS CLI。Security Hub との信頼されたアクセスを無効にすることができるのは、Organizations 管理アカウントの管理者のみです。

Security Hub で信頼されたアクセスを無効にする手順については、「 と Security Hub の統合を無効にする AWS Organizations」を参照してください。

Security Hub の委任管理者の有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、Security Hub の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Security Hub の管理を分離するのに有効です。

詳細については、AWS Security Hub ユーザーガイドの Security Hub 管理者アカウントの指定を参照してください。

メンバーアカウントを Security Hub の委任管理者として指定するには

1. Organizations の管理アカウントでサインインします。
2. 次のいずれかを実行します。
   - 管理アカウントで Security Hub が有効になっていない場合は、Security Hub コンソールで [Go to Security Hub] (Security Hub に移動) を選択します。
   - 管理アカウントで Security Hub が有効になっている場合は、Security Hub コンソールで、「一般的な設定」を選択します。
3. [Delegated Administrator] (委任管理者) に、アカウント ID を入力します。

Security Hub の委任管理者を無効にする

委任された Security Hub 管理者アカウントを削除できるのは、組織管理アカウントのみです。

委任された Security Hub 管理者を変更するには、まず現在の委任された管理者アカウントを削除してから、新しいアカウントを指定する必要があります。

Security Hub コンソールを使用して、あるリージョンの委任管理者を削除すると、その管理者はすべてのリージョンから自動的に削除されます。

Security Hub API は、委任された Security Hub 管理者アカウントを API コールまたはコマンドが発行されたリージョンからのみ削除します。他のリージョンでもこの操作を繰り返す必要があります。

Organizations API を使用して委任された Security Hub 管理者アカウントを削除すると、すべてのリージョンで自動的に削除されます。

委任された Security Hub 管理者を無効にする手順については、「委任された管理者 の削除または変更」を参照してください。

Amazon S3 Storage Lens と AWS Organizations

Amazon S3 Storage Lens に組織への信頼されたアクセスを許可することで、組織内のすべての でメトリクスを収集および集約 AWS アカウント できるようになります。S3 Storage Lens は、これを実行するにあたり、組織に属するアカウントのリストにアクセスします。そして、すべてのアカウントのストレージ、使用状況、アクティビティのメトリクスを収集して分析します。

詳細については、Amazon S3 Storage Lens ユーザーガイドの Amazon S3 Storage Lens でのサービスにリンクされたロールの使用を参照してください。

Amazon S3 Storage Lens を と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にし、Storage Lens の設定が組織に適用されていると、以下のサービスにリンクされたロールが組織の代理管理者アカウントに自動的に作成されます。このロールにより、Amazon S3 Storage Lens はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Amazon S3 Storage Lens と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForS3StorageLens`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Amazon S3 Storage Lens によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `storage-lens.s3.amazonaws.com`

Amazon S3 Storage Lens との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Amazon S3 Storage Lens コンソールまたは AWS Organizations コンソールを使用して、信頼されたアクセスを有効にできます。

> **Important**
>
> Organizations との統合の有効化には、可能な場合は常に Amazon S3 Storage Lens のコンソールまたはツールを使用することを強くお勧めします。そうすることで、サービスに必要

なリソースの作成など、必要な構成が Amazon S3 Storage Lens で実行可能になります。ここに示す手順は、統合の有効化に Amazon S3 Storage Lens が提供するツールを使用できない場合にのみ実施してください。詳細については、こちらの注意事項を参照してください。詳細については、この注意を参照してください。
Amazon S3 Storage Lens のコンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実施する必要はありません。

Amazon S3 コンソールを使用して信頼されたアクセスを有効にするには

「Amazon Simple Storage Service ユーザーガイド」のS3 Storage Lens の信頼されたアクセスの有効化」を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon S3 Storage Lens を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. Amazon S3 Storage Lens の信頼されたアクセスを有効にするダイアログボックスで、確認のために enable と入力し、信頼されたアクセスを有効にする を選択します。
6. のみの管理者である場合は AWS Organizations、Amazon S3 Storage Lens の管理者に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、Amazon S3 Storage Lens を Organizations で信頼できるサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal storage-lens.s3.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

Amazon S3 Storage Lens との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Amazon S3 Storage Lens ツールだけで、信頼されたアクセスを無効にできます。

Amazon S3 コンソール、、 AWS CLI または任意の AWS SDKsを使用して、信頼されたアクセスを無効にできます。

Amazon S3 コンソールを使用して信頼されたアクセスを無効にするには

「Amazon Simple Storage Service ユーザーガイド」のS3 Storage Lens の信頼されたアクセスの無効化」を参照してください。

Amazon S3 Storage Lens 用の代理管理者アカウントの有効化

メンバーアカウントを組織の代理管理者として指定すると、そのアカウントのユーザーおよびロールは、 Amazon S3 Storage Lens の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Amazon S3 Storage Lens の管理を分離するのに有効です。

> 最小アクセス許可
>
> 次の許可を持つ Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Amazon S3 ストレージレンズの委任管理者としてメンバーアカウントを設定できます。
> `organizations:RegisterDelegatedAdministrator`
> `organizations:DeregisterDelegatedAdministrator`

Amazon S3 Storage Lens は、組織内の代理管理者アカウントを最大で 5 つまでサポートしています。

Amazon S3 Storage Lens の代理管理者としてメンバーアカウントを指定するには

委任された管理者は、Amazon S3 コンソール、、 AWS CLI または任意の AWS SDKsを使用して登録できます。Amazon S3 コンソールを使用してメンバーアカウントを組織の委任管理者アカウントとして登録するには、Amazon Simple Storage Service ユーザーガイドのS3 Storage Lens の委任管理者の登録」を参照してください。

Amazon S3 Storage Lens の代理管理者の登録を解除するには

Amazon S3 コンソール、、 AWS CLI または任意の AWS SDKs を使用して、委任された管理者の登録を解除できます。Amazon S3 コンソールを使用して委任管理者の登録を解除するには、Amazon Simple Storage Service ユーザーガイドのS3 Storage Lens の委任管理者の登録解除」を参照してください。

Amazon Security Lake と AWS Organizations

Amazon Security Lake は、クラウド、オンプレミス、カスタムソースのセキュリティデータを、アカウントに保存されているデータレイクに一元化します。Organizations と統合することで、アカウント全体からログとイベントを収集するデータレイクを作成できます。詳細については、「Amazon Security Lake ユーザーガイド」の「AWS Organizationsで複数のアカウントを管理する」を参照してください。

Amazon Security Lake を と統合するには、次の情報を使用します AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

RegisterDataLakeDelegatedAdministrator API を呼び出すと、次のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Amazon Security Lake は、組織内の組織のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、Amazon Security Lake と Organizations 間の信頼されたアクセスを無効にした場合、または組織からメンバーアカウントを削除した場合のみです。

- `AWSServiceRoleForSecurityLake`

⚠ 推奨事項: Security Lake の RegisterDataLakeDelegatedAdministrator API を使用して、Security Lake に Organization へのアクセスを許可し、Organizations の委任された管理者を登録する

Organizations APIs を使用して委任された管理者を登録すると、Organizations のサービスにリンクされたロールが正常に作成されない場合があります。完全な機能を確保するには、Security Lake APIsを使用します。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Amazon Security Lake で使用されるサービスにリンクされたロールは、以下のサービスプリンシパルへのアクセスを許可します。

- `securitylake.amazonaws.com`

Amazon Security Lake での信頼されたアクセスの有効化

Security Lake との信頼されたアクセスを有効化すると、組織のメンバーシップに変更があった場合、Security Lake が自動的に対応するようになります。委任管理者は、任意の組織アカウントでサポートされている のサービスからの AWS ログ収集を有効にできます。詳細については、「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake のサービスにリンクされたロール」を参照してください。

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon Security Lake を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「Amazon Security Lake の信頼されたアクセスを有効にする」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、Amazon Security Lake の管理者に、コンソールを使用してそのサービスを有効にし、 と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、Amazon Security Lake を Organizations で信頼されたサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal securitylake.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

Amazon Security Lake での信頼されたアクセスの無効化

Amazon Security Lake との信頼されたアクセスを無効にすることができるのは、Organizations 管理アカウントの管理者のみです。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストで Amazon Security Lake を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. Amazon Security Lake の信頼されたアクセスを無効にするダイアログボックスで、無効にして確認します。次に、信頼されたアクセスを無効にするを選択します。
6. のみの管理者である場合は AWS Organizations、Amazon Security Lake の管理者に、コンソールまたはツールを使用してそのサービスを無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にできます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、Organizations で Amazon Security Lake を信頼されたサービスとすることを無効にできます。

```
$ aws organizations disable-aws-service-access \
```

```
    --service-principal securitylake.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Amazon Security Lake の委任管理者アカウントの有効化

Amazon Security Lake の委任管理者は、組織内の他のアカウントをメンバーアカウントとして追加します。委任管理者は Amazon Security Lake を有効にし、メンバーアカウントの Amazon Security Lake 設定を設定できます。委任管理者は、Amazon Security Lake が有効になっているすべての AWS リージョン (現在使用しているリージョンエンドポイントに関係なく) の組織全体のログを収集できます。

委任された管理者を設定して、組織の新しいアカウントをメンバーとして自動的に追加することもできます。Amazon Security Lake の委任管理者は、関連するメンバーアカウントのログとイベントにアクセスできます。したがって、関連するメンバーアカウントが所有するデータを収集するように Amazon Security Lake を設定できます。また、関連するメンバーアカウントが所有するデータを使用する権限をサブスクライバーに付与することもできます。

詳細については、「Amazon Security Lake ユーザーガイド」の「AWS Organizationsで複数のアカウントを管理する」を参照してください。

> **最小アクセス許可**
>
> Organizations 管理アカウントの管理者のみが、組織内の Amazon Security Lake の委任管理者としてメンバーアカウントを設定できます。

Amazon Security Lake コンソール、Amazon Security Lake `CreateDatalakeDelegatedAdmin` API アクション、または `create-datalake-delegated-admin` CLI コマンドを使用して、委任管理者アカウントを指定できます。または、Organizations `RegisterDelegatedAdministrator` CLI または SDK オペレーションを使用できます。Amazon Security Lake の委任された管理者アカウントを有効にする手順については、「Amazon Security Lake ユーザーガイド」の「委任された Security Lake 管理者の指定」および「メンバーアカウントの追加」を参照してください。

AWS CLI, AWS API

CLI または AWS SDKs のいずれかを使用して AWS 委任管理者アカウントを設定する場合は、次のコマンドを使用できます。

- AWS CLI:

```
$  aws organizations register-delegated-administrator \
    --account-id 123456789012 \ --service-principal securitylake.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator`オペレーションとメンバーアカウントの ID 番号を呼び出し、アカウントサービスプリンシパルをパラメータ`account.amazonaws.com`として識別します。

Amazon Security Lake の委任管理者を無効にする

Organizations 管理アカウントまたは Amazon Security Lake の委任された管理者アカウントの管理者のみが、組織から委任された管理者アカウントを削除できます。

委任された管理者アカウントを削除するには、Amazon Security Lake `DeleteDatalakeDelegatedAdmin` API `delete-datalake-delegated-admin` アクション、CLI コマンド、または Organizations CLI または SDK `DeregisterDelegatedAdministrator` オペレーションを使用します。Amazon Security Lake を使用して委任された管理者を削除するには、「Amazon Security Lake ユーザーガイド」の「Amazon Security Lake の委任された管理者の削除」を参照してください。

AWS Service Catalog および AWS Organizations

Service Catalog では、 AWSでの使用が承認された IT サービスのカタログを作成および管理できます。

Service Catalog と の統合により、組織全体でのポートフォリオの共有と製品のコピー AWS Organizations が簡素化されます。Service Catalog 管理者は、ポートフォリオを共有する AWS Organizations ときに で既存の組織を参照できます。また、組織のツリー構造内の信頼できる組織単位 (OU) とポートフォリオを共有できます。これにより、ポートフォリオ ID を共有する必要がなくなり、受信側アカウントはポートフォリオをインポートするときに手動でポートフォリオ ID を参照する必要がなくなります。このメカニズムで共有されるポートフォリオは、Service Catalog の管理者の [Imported Portfolio] (インポートされたポートフォリオ) ビューに、共有先アカウントとして表示されます。

Service Catalog の詳細については、「Service Catalog 管理者ガイド」を参照してください。

AWS Service Catalog との統合には、以下の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

AWS Service Catalog は、信頼されたアクセスの有効化の一環として、サービスにリンクされたロールを作成しません。

アクセス許可を付与するために使用されるサービスプリンシパル

信頼されたアクセスを有効にするには、次のサービスプリンシパルを指定する必要があります。

- `servicecatalog.amazonaws.com`

Service Catalog を使用して信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Service Catalog コンソールまたは AWS Organizations コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Service Catalog コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Service Catalog を実行できます。ここに示す手順は、統合の有効化に AWS Service Catalogが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Service Catalog コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

Service Catalog CLI または AWS SDK を使用して信頼されたアクセスを有効にするには

以下のいずれかのコマンドまたはオペレーションを呼び出します。

- AWS CLI: aws サービスカタログ enable-aws-organizations-access
- AWS SDKs: AWSServiceCatalog::EnableAWSOrganizationsAccess

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Service Catalogで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Service Catalog」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Service Catalog に、コンソールを使用してそのサービスを有効にして を操作できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、 を Organizations の信頼されたサービス AWS Service Catalog として有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal servicecatalog.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 有効AWSServiceAccess

Service Catalog を使用して信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Service Catalog の使用 AWS Organizations 中に を使用して信頼されたアクセスを無効にしても、現在の共有は削除されませんが、組織全体で新しい共有を作成することはできません。このアクションを呼び出した後で変更した場合、現在の共有は組織構造と同期されません。

信頼されたアクセスは、 AWS Service Catalog または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Service Catalog コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、 は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS Service Catalog を実行できます。ここに示す手順は、統合の無効化に AWS Service Catalogが提供するツールを使用できない場合にのみ実施してください。
> AWS Service Catalog コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

Service Catalog CLI または AWS SDK を使用して信頼されたアクセスを無効にするには

以下のいずれかのコマンドまたはオペレーションを呼び出します。

- AWS CLI: aws サービスカタログ disable-aws-organizations-access
- AWS SDKs 無効化AWSOrganizationsAccess

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Service Catalogで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。

5. 「 の信頼されたアクセスを無効にする AWS Service Catalog」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 のコンソールまたはツールを使用してそのサービスを無効にできるようになった AWS Service Catalog ことを の管理者に伝えます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にできます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Service Catalog としてを無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal servicecatalog.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。
- AWS API: 無効AWSServiceAccess

Service Quotas と AWS Organizations

Service Quotas は、一元的な場所からクォータを表示および管理できる AWS のサービスです。クォータ (制限とも呼ばれます) は、AWS アカウント のリソース、アクション、アイテムの最大値です。

Service Quotas が AWS Organizations に関連付けられている場合、クォータリクエストテンプレートを作成して、アカウントの作成時に自動的にクォータの引き上げをリクエストできます。

Service Quotas の詳細については、Service Quotas ユーザーガイドを参照してください。

Service Quotas と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Service Quotas はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Service Quotas と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForServiceQuotas`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Service Quotas によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `servicequotas.amazonaws.com`

Service Quotas との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Service Quotas だけで、信頼されたアクセスを有効にできます。

Service Quotas コンソール、AWS CLI、または SDK を使用して、信頼されたアクセスを有効にできます。

- Service Quotas コンソールを使用して信頼されたアクセスを有効にするには

  AWS Organizations の管理アカウントでサインインしてから、Service Quotas コンソールでテンプレートを設定します。詳細については、Service Quotas ユーザーガイドの Using the Service Quota Template を参照してください。
- Service Quotas の AWS CLI または SDK を使用して信頼されたアクセスを有効にするには

  以下のコマンドまたはオペレーションを呼び出します。

  - AWS CLI: aws service-quotas associate-service-quota-template

- AWS SDK: AssociateServiceQuotaTemplate

AWS IAM Identity Center および AWS Organizations

AWS IAM Identity Center は、すべての AWS アカウント およびクラウドアプリケーションにシングルサインオンアクセスを提供します。を介して Microsoft Active Directory に接続 AWS Directory Service し、そのディレクトリ内のユーザーが既存の Active Directory ユーザー名とパスワードを使用してパーソナライズされた AWS アクセスポータルにサインインできるようにします。 AWS アクセスポータルから、ユーザーはアクセス許可を持つすべての AWS アカウント およびクラウドアプリケーションにアクセスできます。

IAM Identity Center の詳細については、AWS IAM Identity Center ユーザーガイドを参照してください。

AWS IAM Identity Center との統合には、次の情報を参考にしてください AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、IAM Identity Center はサポートされているオペレーションを組織内の組織アカウントで実行できます。

このロールを削除または変更できるのは、IAM Identity Center と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForSSO`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。IAM Identity Center によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `sso.amazonaws.com`

IAM Identity Center との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼された AWS Organizations アクセスは、 AWS IAM Identity Center コンソールまたは コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS IAM Identity Center コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS IAM Identity Center を実行できます。ここに示す手順は、統合の有効化に AWS IAM Identity Centerが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS IAM Identity Center コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

IAM Identity Center を使用するには、 との信頼されたアクセスが必要です AWS Organizations 。IAM Identity Center をセットアップすると、信頼されたアクセスが有効になります。詳細については、AWS IAM Identity Center ユーザーガイドの Getting Started - Step 1: Enable AWS IAM Identity Center を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS IAM Identity Centerで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。

5. 「 の信頼されたアクセスを有効にする AWS IAM Identity Center」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS IAM Identity Center に、コンソールを使用してそのサービスを有効にして と連携できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS IAM Identity Center として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal sso.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

IAM Identity Center との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

IAM Identity Center を使用するには、 との信頼されたアクセスが必要です AWS Organizations 。IAM Identity Center を使用している AWS Organizations ときに を使用して信頼されたアクセスを無効にすると、組織にアクセスできないため、機能しなくなります。ユーザーは IAM Identity Center を使用してアカウントにアクセスできません。IAM Identity Center によって作成されるロールは残りますが、 サービスからアクセスすることはできません。IAM Identity Center のサービスにリンクされたロールは残ります。その後、信頼されたアクセスを再度有効にすると、IAM Identity Center は以前のように動作し続けます。サービスを再設定する必要はありません。

組織からアカウントを削除すると、サービスにリンクされたロールなど、すべてのメタデータとリソースが IAM Identity Center によって自動的にクリーンアップされます。スタンドアロンアカウントを組織から削除した場合は、IAM Identity Center で機能しなくなります。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS IAM Identity Centerで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS IAM Identity Center」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS IAM Identity Center に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、 Organizations の信頼されたサービス AWS IAM Identity Center として を無効にできます。

```
$ aws organizations disable-aws-service-access \
```

```
    --service-principal sso.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

IAM Identity Center 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、IAM Identity Center の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、IAM Identity Center の管理から組織の管理を分離するのに有効です。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で IAM Identity Center の委任管理者としてメンバーアカウントを設定できます。

IAM Identity Center の委任管理者アカウントを有効にする方法については、AWS IAM Identity Center ユーザーガイドの委任された管理を参照してください。

AWS Systems Manager および AWS Organizations

AWS Systems Manager は、 リソースの可視性と制御を可能にする機能のコレクションです AWS 。次の Systems Manager 機能は、組織内のすべての AWS アカウント にわたって組織と連携します。

- Systems Manager Explorer は、 AWS リソースに関する情報をレポートするカスタマイズ可能なオペレーションダッシュボードです。Organizations と Systems Manager Explorer を使用して、組織内のすべての AWS アカウント でオペレーションデータを同期できます。詳細については、AWS Systems Manager ユーザーガイドの Systems Manager Explorer を参照してください。
- Systems Manager Change Manager は、アプリケーションの設定とインフラストラクチャに対する運用上の変更をリクエスト、承認、実装、レポートするためのエンタープライズ変更管理フレームワークです。詳細については、「AWS Systems Manager ユーザーガイド」の「AWS Systems Manager Change Manager」を参照してください。
- Systems Manager OpsCenter は、オペレーションエンジニアや IT プロフェッショナルが AWS リソースに関連する運用作業項目 (OpsItems) を表示、調査、解決できる一元的な場所を提供します。Organizations OpsCenter で を使用すると、1 回のセッション中に管理アカウント

(Organizations 管理アカウントまたは Systems Manager の委任された管理者アカウント) と他の 1 つのアカウントからの の使用 OpsItems がサポートされます。設定が完了すると、ユーザーは次のタイプのアクションを実行できます。

- 別のアカウント OpsItems で作成、表示、更新します。
- 別のアカウントの OpsItems で指定されている AWS リソースに関する詳細情報を表示します。
- Systems Manager Automation ランブックを起動して、別のアカウントの AWS リソースに関する問題を修正します。

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

以下の情報は、 AWS Systems Manager との統合に役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Systems Manager はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Systems Manager と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForAmazonSSM_AccountDiscovery`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Systems Manager によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `ssm.amazonaws.com`

Systems Manager との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Organizations ツールだけで、信頼されたアクセスを有効にできます。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Systems Managerで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Systems Manager」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 コンソールを使用してそのサービスを有効に AWS Systems Manager して を操作できるようにすることを管理者に伝えてください AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、 を Organizations の信頼されたサービス AWS Systems Manager として有効にできます。

```
$ aws organizations enable-aws-service-access \
    --service-principal ssm.amazonaws.com
```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

Systems Manager との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Systems Manager では、組織 AWS アカウント 内の 間でオペレーションデータを同期 AWS Organizations するために、 との信頼されたアクセスが必要です。信頼されたアクセスを無効にすると、Systems Manager によるオペレーションデータの同期は失敗し、エラーが報告されます。

Organizations ツールだけで、信頼されたアクセスを無効にできます。

信頼されたアクセスを無効にするには、 AWS Organizations コンソールを使用するか、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを無効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストAWS Systems Managerで を選択します。
4. Disable trusted access (信頼されたアクセスを無効にする) を選択します。
5. 「 の信頼されたアクセスを無効にする AWS Systems Manager」ダイアログボックスで、「無効化」と入力して確認します。次に、「信頼されたアクセスを無効にする」を選択します。
6. のみの管理者である場合は AWS Organizations、 の管理者 AWS Systems Manager に、コンソールまたはツールを使用してそのサービスを無効にして、 の操作を無効にできるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Systems Manager として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal ssm.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Systems Manager 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、そのアカウントのユーザーおよびロールは、Systems Manager の管理アクションを実行できるようになります。通常この管理アクションは、組織の管理アカウントのユーザーとロールだけが実行できるものです。この手法は、組織の管理から Systems Manager の管理を分離するのに有効です。

組織全体で Change Manager を使用する場合は、委任管理者アカウントを使用します。これは、Change Manager で変更テンプレート、変更リクエスト、変更ランブック、承認ワークフローを管理するためのアカウントとして指定されている AWS アカウント です。この委任アカウントにより、組織全体の変更アクティビティが管理されます。Change Manager を使用するように組織をセットアップするときは、どのアカウントがこのロールを担うかを指定します。組織の管理アカウントである必要はありません。Change Manager を単一のアカウントのみで使用する場合、委任管理者アカウントは必要ありません。

メンバーアカウントを代理管理者として指定するには、「AWS Systems Manager ユーザーガイド」の以下のトピックを参照してください。

- Explorer と については OpsCenter、「委任された管理者の設定」を参照してください。
- 変更マネージャーについては、「Setting up an organization and delegated account for Change Manager」(の組織と委任されたアカウントの設定) を参照してください。

タグポリシーと AWS Organizations

タグポリシーは、組織のアカウント内のリソース間でタグを標準化する AWS Organizations のに役立つ のポリシーの一種です。ポリシーの詳細については、「タグポリシー」を参照してください。

次の情報は、タグポリシーを と統合するのに役立ちます AWS Organizations。

サービスにリンクされたロールで使用されるサービスプリンシパル

Organizations は、次のサービスプリンシパルを使用して、リソースにアタッチされたタグを操作します。

- `tagpolicies.tag.amazonaws.com`

タグポリシー用の信頼されたアクセスの有効化

信頼されたアクセスを有効にするには、組織でタグポリシーを有効にするか、 AWS Organizations コンソールを使用します。

> **Important**
>
> タグポリシーを有効にして、信頼されたアクセスを有効にすることを強くお勧めします。そうすることで、Organizations で必要な設定タスクが実行可能になります。

タグポリシー用に信頼されたアクセスを有効にするには、 AWS Organizations コンソールでタグポリシーのタイプを有効にします。詳細については、「ポリシータイプの有効化」を参照してください。

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。

2. ナビゲーションペインで [Services (サービス)] を選択します。
3. サービスのリストでタグポリシーを選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. タグポリシーの信頼されたアクセスを有効にするダイアログボックスで、確認のために有効化と入力し、信頼されたアクセスを有効にするを選択します。
6. のみの管理者である場合は AWS Organizations、タグポリシーの管理者に、コンソールを使用してそのサービスを有効にして と連携できるようになったことを知らせます AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、タグポリシーを Organizations で信頼されたサービスとして有効にすることができます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal tagpolicies.tag.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

タグポリシーとの信頼されたアクセスの無効化

AWS Organizations コンソールでタグポリシータイプを無効にすることで、タグポリシーの信頼されたアクセスを無効にできます。詳細については、「ポリシータイプの無効化」を参照してください。

AWS Trusted Advisor および AWS Organizations

AWS Trusted Advisor では、お客様の AWS 環境を検査し、コスト削減、システムの可用性とパフォーマンスの向上、セキュリティギャップの解消につながる推奨事項をお知らせしま

す。Organizations と統合すると、組織内のすべてのアカウントの Trusted Advisor チェック結果を受け取り、チェックや影響を受けるリソースについてまとめたレポートをダウンロードできます。

詳細については、AWS Support ユーザーガイドの Organizational view for AWS Trusted Advisor を参照してください。

AWS Trusted Advisor と AWS Organizations の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Trusted Advisor はサポートされているオペレーションを組織内のアカウントで実行できます。

このロールを削除または変更できるのは、Trusted Advisor と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForTrustedAdvisorReporting`

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Trusted Advisor によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `reporting.trustedadvisor.amazonaws.com`

Trusted Advisor との信頼されたアクセスの有効化

信頼されたアクセスの有効化に必要なアクセス許可に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

AWS Trusted Advisor だけで、信頼されたアクセスを有効にできます。

Trusted Advisor コンソールを使用して信頼されたアクセスを有効にするには

AWS Support ユーザーガイドの組織ビューの有効化を参照してください。

Trusted Advisor との信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス許可に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

この機能を無効にすると、組織内の他のすべてのアカウントで、Trusted Advisor によるチェック情報の記録が停止します。既存のレポートの表示やダウンロード、新しいレポートの作成はできなくなります。

AWS Trusted Advisor または AWS Organizations ツールを使用して信頼されたアクセスを無効にできます。

> **Important**
>
> Organizations との統合の無効化には、可能な場合は常に AWS Trusted Advisor コンソールまたはツールを使用することを強くお勧めします。そうすることで、AWS Trusted Advisor は、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップを実行できます。ここに示す手順は、統合の無効化に AWS Trusted Advisor が提供するツールを使用できない場合にのみ実施してください。
> AWS Trusted Advisor コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実施する必要はありません。

Trusted Advisor コンソールを使用して信頼されたアクセスを無効にするには

AWS Support ユーザーガイドの組織ビューの無効化を参照してください。

信頼されたアクセスの無効化には、Organizations の AWS CLI コマンドを実行する方法と、いずれかの AWS SDK で Organizations API オペレーションを呼び出す方法があります。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

サービスへの信頼されたアクセスを無効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、AWS Trusted Advisor を Organizations で信頼されたサービスとして無効にすることができます。

```
$ aws organizations disable-aws-service-access \
    --service-principal reporting.trustedadvisor.amazonaws.com
```

このコマンドが成功した場合、出力は生成されません。

- AWS API: DisableAWSServiceAccess

Trusted Advisor 用の委任管理者アカウントの有効化

メンバーアカウントを組織の委任管理者として指定すると、指定されたアカウントのユーザーおよびロールは組織のその他のメンバーアカウントの AWS アカウント メタデータを管理できるようになります。委任された管理者アカウントを有効にしない場合、これらのタスクは組織の管理アカウントによってのみ実行できます。この手法は、組織の管理からアカウントの詳細の管理を分離するのに有効です。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Trusted Advisor の委任管理者としてメンバーアカウントを設定できます

Trusted Advisor の委任された管理者アカウントを有効にする方法については、AWS Support ユーザーガイドの「委任された管理者」を参照してください。

AWS CLI, AWS API

AWS CLI または AWS SDK を使用して委任管理者アカウントを設定するには、次のコマンドを使用します。

- AWS CLI:

```
$  aws organizations register-delegated-administrator \
    --account-id 123456789012 \
    --service-principal reporting.trustedadvisor.amazonaws.com
```

- AWS SDK: Organizations `RegisterDelegatedAdministrator` オペレーションおよびメンバーアカウントの ID 番号を呼び出して、アカウントサービスプリンシパル `account.amazonaws.com` をパラメータとして識別します。

Trusted Advisor の委任された管理者の無効化

Trusted Advisor コンソール、あるいは Organizations `DeregisterDelegatedAdministrator` または SDK オペレーションを使用して、委任された管理者を削除できます。Trusted Advisor コンソールを使用して、委任された管理 Trusted Advisor アカウントを無効にする方法については、AWS Support ユーザーガイドの「Deregister delegated administrators」(委任された管理者の登録解除) を参照してください。

AWS Well-Architected Tool および AWS Organizations

AWS Well-Architected Tool は、ワークロードの状態を文書化し、最新の AWS アーキテクチャのベストプラクティスと比較するのに役立ちます。

Organizations AWS Well-Architected Tool で を使用すると、 AWS Well-Architected Tool と Organizations の両方のお客様は、組織の他のメンバーと AWS Well-Architected Tool リソースを共有するプロセスを簡素化できます。

詳しくは、AWS Well-Architected Tool ユーザーガイドの「AWS Well-Architected Tool リソースを共有する」を参照してください。

以下の情報は、 AWS Well-Architected Tool との統合に役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより AWS WA Tool 、 は組織内のアカウント内でサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、 AWS WA Tool と Organizations 間の信頼されたアクセスを無効にした場合か、組織から当該のメンバーアカウントを削除した場合だけです。

- `AWSServiceRoleForWellArchitected`

サービスロールポリシーは `AWSWellArchitectedOrganizationsServiceRolePolicy` です

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。で使用されるサービスにリンクされたロールは、次のサービスプリンシパルへのアクセス AWS WA Tool を許可します。

- wellarchitected.amazonaws.com

AWS WA Toolとの信頼されたアクセスの有効化

組織内の階層的な変更を反映する AWS WA Tool ために を更新できるようにします。

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

信頼された AWS Organizations アクセスは、 AWS Well-Architected Tool コンソールまたは コンソールを使用して有効にできます。

> **Important**
>
> 可能な限り、 AWS Well-Architected Tool コンソールまたはツールを使用して Organizations との統合を有効にすることを強くお勧めします。これにより、 はサービスに必要なリソースの作成など、必要な設定 AWS Well-Architected Tool を実行できます。ここに示す手順は、統合の有効化に AWS Well-Architected Toolが提供するツールを使用できない場合にのみ実施してください。詳細については、この注意を参照してください。
> AWS Well-Architected Tool コンソールまたはツールを使用して信頼されたアクセスを有効にする場合、これらのステップを実行する必要はありません。

AWS WA Tool コンソールを使用して信頼されたアクセスを有効にするには

「 ユーザーガイド」の「 AWS Well-Architected Tool リソースの共有」を参照してください。 AWS Well-Architected Tool

信頼されたアクセスを有効にするには、 AWS Organizations コンソールを使用するか、 AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサイン・ インします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. ナビゲーションペインで [Services (サービス)] を選択します。

3. サービスのリストAWS Well-Architected Toolで を選択します。
4. [Enable trusted access (信頼されたアクセスを有効にする)] を選択します。
5. 「 の信頼されたアクセスを有効にする AWS Well-Architected Tool」ダイアログボックスで、「有効化して確認します」と入力し、「信頼されたアクセスを有効にする」を選択します。
6. のみの管理者の場合は AWS Organizations、 の管理者 AWS Well-Architected Tool に、コンソールを使用してそのサービスを有効にして と連携できるようにしました AWS Organizations。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを有効にできます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行して、Organizations の信頼されたサービス AWS Well-Architected Tool として を有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal wellarchitected.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 有効AWSServiceAccess

AWS WA Toolとの信頼されたアクセスの無効化

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

信頼されたアクセスは、 AWS Well-Architected Tool または AWS Organizations ツールを使用して無効にできます。

> **Important**
>
> 可能な限り、 AWS Well-Architected Tool コンソールまたはツールを使用して Organizations との統合を無効にすることを強くお勧めします。これにより、サービスで不要になったリソースやアクセスロールの削除など、必要なクリーンアップ AWS Well-Architected Tool を実行できます。ここに示す手順は、統合の無効化に AWS Well-Architected Toolが提供するツールを使用できない場合にのみ実施してください。
> AWS Well-Architected Tool コンソールまたはツールを使用して信頼されたアクセスを無効にする場合、これらのステップを実行する必要はありません。

AWS WA Tool コンソールを使用して信頼されたアクセスを無効にするには

「 ユーザーガイド」の「 AWS Well-Architected Tool リソースの共有」を参照してください。 AWS Well-Architected Tool

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にできます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行して、 Organizations の信頼されたサービス AWS Well-Architected Tool として を無効にできます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal wellarchitected.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

Amazon VPC IP Address Manager (IPAM) および AWS Organizations

Amazon VPC IP Address Manager (IPAM) は、 AWS ワークロードの IP アドレスの計画、追跡、モニタリングを容易にする VPC 機能です。

AWS Organizations を使用すると、組織全体の IP アドレスの使用状況をモニタリングし、メンバーアカウント間で IP アドレスプールを共有できます。

詳細については、Amazon VPC IPAM ユーザーガイドの「Integrate IPAM with AWS Organizations」を参照してください。

次の情報は、Amazon VPC IP Address Manager (IPAM) を と統合するのに役立ちます AWS Organizations。

統合を有効にする際に作成されるサービスにリンクされたロール

IPAM コンソールまたは IPAM の `EnableIpamOrganizationAdminAccount` API のいずれかを使用して IPAM を AWS Organizations に統合すると、以下のサービスにリンクされたロールが組織の管理アカウントと各メンバーアカウント内に自動的に作成されます。

- `AWSServiceRoleForIPAM`

詳細については、Amazon VPC IPAM ユーザーガイドの「Service-linked roles for IPAM」を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。IPAM によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `ipam.amazonaws.com`

IPAM で信頼されたアクセスを有効にする

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

> **Note**
>
> IPAM の委任管理者を指定すると、組織の IPAM に対する信頼されたアクセスが自動的に有効になります。
> IPAM では、組織のこのサービスの委任管理者となるメンバーアカウントを指定する AWS Organizations 前に、 への信頼されたアクセスが必要です。

信頼されたアクセスを有効にするには、Amazon VPC IP Address Manager (IPAM) ツールのみを使用します。

IPAM コンソールまたは IPAM `EnableIpamOrganizationAdminAccount` API を使用して AWS Organizations IPAM を と統合すると、IPAM への信頼されたアクセスが自動的に付与されます。信頼されたアクセスを付与すると、サービスにリンクされたロール `AWS ServiceRoleForIPAM` が組織の管理アカウントおよびすべてのメンバーアカウントに作成されます。IPAM は、サービスにリンクされたロールを使用して、組織内の EC2 ネットワークリソースに関連付けられた CIDRs をモニタリングし、IPAM に関連するメトリクスを Amazon に保存します CloudWatch。詳細については、Amazon VPC IPAM ユーザーガイドの「Service-linked roles for IPAM」を参照してください。

信頼されたアクセスを有効にする手順については、Amazon VPC IP アドレス管理ユーザーガイドの「Integrate IPAM with AWS Organizations」を参照してください。

> **Note**
>
> AWS Organizations コンソールまたは `EnableAWSServiceAccess` API を使用して、IPAM で信頼されたアクセスを有効にすることはできません。

IPAM で信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

API を使用して IPAM との信頼されたアクセスを無効にすることができるのは、 AWS Organizations 管理アカウントの管理者のみです AWS Organizations `disable-aws-service-access`。

IP アドレス管理アカウントのアクセス許可を無効にし、サービスにリンクされたロールの削除の詳細については、Amazon VPC IPAM ユーザーガイドの「Service-linked roles for IPAM」を参照してください。

信頼されたアクセスを無効にするには、Organizations AWS CLI コマンドを実行するか、いずれかの AWS SDKs。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを無効にするには

次の AWS CLI コマンドまたは API オペレーションを使用して、信頼されたサービスアクセスを無効にすることができます。

- AWS CLI: disable-aws-service-access

  次のコマンドを実行し、Amazon VPC IP Address Manager (IPAM) を Organizations で信頼されたサービスとして無効にすることができます。

  ```
  $ aws organizations disable-aws-service-access \
      --service-principal ipam.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: 無効AWSServiceAccess

IPAM 用の委任管理者アカウントの有効化

IPAM の委任管理者アカウントは、IPAM および IP アドレスのプールの作成、組織での IP アドレスの使用状況のモニタリング、およびメンバーメンバーアカウント間の IP アドレスプールの共有を行います。詳細については、Amazon VPC IPAM ユーザーガイドの「Integrate IPAM with AWS Organizations」を参照してください。

IPAM の委任管理者を構成できるのは、組織管理アカウントの管理者のみです。

委任された管理者アカウントは、IPAM コンソールから指定するか、`enable-ipam-organization-admin-account` API を使用して指定します。詳細については、「 コマンドリファレンス」のenable-ipam-organization-admin「-account」を参照してください。 AWS AWS CLI

> 最小アクセス許可
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で IPAM の委任管理者としてメンバーアカウントを設定できます

IPAM コンソールを使用して委任管理者を設定するには、「Amazon VPC IPAM ユーザーガイド 」の「IPAM を AWS Organizationsと統合する」を参照してください。

IPAM の委任された管理者の無効化

IPAM の委任管理者を構成できるのは、組織管理アカウントの管理者のみです。

を使用して委任管理者を削除するには AWS AWS CLI、 コマンドリファレンスのdisable-ipam-organization-admin「-account」を参照してください。 AWS AWS CLI

IPAM コンソールを使用して委任管理者 IPAM アカウントを無効にするには、「Amazon VPC IPAM ユーザーガイド」の「IPAM を AWS Organizationsと統合する」を参照してください。

Amazon VPC Reachability Analyzer と AWS Organizations

Reachability Analyzer は、仮想プライベートクラウド (VPC) 内のソースリソースと送信先リソース間の接続テストを実行できるようにする設定分析ツールです。

Reachability Analyzer で AWS Organizations を使用すると、組織内のアカウント間のパスを追跡できます。

詳細については、「Reachability Analyzer ユーザーガイド」の「Cross-account analyses for Reachability Analyzer」(Reachability Analyzer のクロスアカウント分析) を参照してください。

AWS Organizations と Reachability Analyzer の統合には、次の情報を参考にしてください。

統合を有効にする際に作成されるサービスにリンクされたロール

信頼されたアクセスを有効にすると、以下のサービスにリンクされたロールが組織の管理アカウントに自動的に作成されます。このロールにより、Reachability Analyzer は組織内のアカウントでサポートされているオペレーションを実行できます。

このロールを削除または変更できるのは、Reachability Analyzer と Organizations 間の信頼されたアクセスを無効にする場合、または組織からメンバーアカウントを削除する場合だけです。

- `AWSServiceRoleForReachabilityAnalyzer`

詳細については、「Reachability Analyzer ユーザーガイド」の「Cross-account analyses for Reachability Analyzer」(Reachability Analyzer のクロスアカウント分析) を参照してください。

サービスにリンクされたロールで使用されるサービスプリンシパル

前のセクションで説明したサービスにリンクされたロールを引き受けることができるのは、ロールに定義された信頼関係によって承認されたサービスプリンシパルだけです。Reachability Analyzer によって使用されるサービスにリンクされたロールには、次のサービスプリンシパルへのアクセス許可が付与されます。

- `reachabilityanalyzer.networkinsights.amazonaws.com`

Reachability Analyzer で信頼されたアクセスを有効にするには

信頼されたアクセスの有効化に必要な権限に関しては、信頼されたアクセスを有効にするために必要なアクセス許可 を参照してください。

Reachability Analyzer の委任管理者を指定すると、組織の Reachability Analyzer に対する信頼されたアクセスが自動的に有効になります。

組織でこのサービスの委任管理者にするメンバーアカウントを指定するにあたり、Reachability Analyzer には AWS Organizations への信頼されたアクセスが必要です。

> **Important**
>
> - Reachability Analyzer コンソールまたは Organizations コンソールを使用して、信頼されたアクセスを有効にできます。ただし、Reachability Analyzer コンソールまたは `EnableMultiAccountAnalysisForAwsOrganization` API を使用して、Organizations との統合を有効にすることを強くお勧めします。そうすることで、サービスに必要なリソースの作成などの設定が Reachability Analyzer で実行可能になります。
> - 信頼されたアクセスを付与すると、サービスにリンクされたロール `AWSServiceRoleForReachabilityAnalyzer` が組織の管理アカウントおよびすべてのメンバーアカウントに作成されます。Reachability Analyzer はサービスにリンクされたロールを使用して管理を許可し、委任管理者は組織内の任意のリソース間の接続分析を実行できるようになります。Reachability Analyzer は接続に関するクエリに応答するため、組織内のアカウントにおけるネットワーク要素のスナップショットを取得できます。
> - 詳細および Reachability Analyzer で信頼されたアクセスを有効にする手順については、「Reachability Analyzer ユーザーガイド」の「Cross-account analyses for Reachability Analyzer」(Reachability Analyzer のクロスアカウント分析) を参照してください。

信頼されたアクセスの有効化には、AWS Organizations コンソールを使用する方法、AWS CLI コマンドを実行する方法、いずれかの AWS SDK で API オペレーションを呼び出す方法があります。

AWS Management Console

Organizations コンソールを使用して信頼されたアクセスを有効にするには

1. AWS Organizations コンソールにサインインします。組織の管理アカウントで、IAM ユーザーとしてサインインするか、IAM ロールを引き受けるか、ルートユーザーとしてサインインする推奨されません必要があります。
2. [サービス] ページで、[VPC Reachability Analyzer] の行を探し、サービスの名前を選択してから [信頼されたアクセスを有効化] を選択します。
3. 確認ダイアログボックスで、[Show the option to enable trusted access] (信頼されたアクセスを有効にするオプションを表示する) を有効にし、ボックスに「**enable**」と入力してから、[Enable trusted access] (信頼されたアクセスを有効にする) を選択します。
4. AWS Organizations だけの管理者である場合は、Reachability Analyzer の管理者に、コンソールを使用してそのサービスを有効にし、AWS Organizations と連携させて使用できるようになったことを知らせます。

AWS CLI, AWS API

Organizations CLI/SDK を使用して信頼されたアクセスを有効にするには

信頼されたサービスのアクセスを有効にするには、次の AWS CLI コマンドまたは API オペレーションを使用できます。

- AWS CLI: enable-aws-service-access

  次のコマンドを実行し、Reachability Analyzer を Organizations で信頼されたサービスとして有効にできます。

  ```
  $ aws organizations enable-aws-service-access \
      --service-principal reachabilityanalyzer.networkinsights.amazonaws.com
  ```

  このコマンドが成功した場合、出力は生成されません。

- AWS API: EnableAWSServiceAccess

Reachability Analyzer で信頼されたアクセスを無効にするには

信頼されたアクセスの無効化に必要なアクセス権限に関しては、信頼されたアクセスを無効にするために必要なアクセス許可 を参照してください。

Reachability Analyzer コンソール (推奨) または Organizations コンソールを使用して、信頼されたアクセスを無効にできます。Reachability Analyzer コンソールを使用して信頼されたアクセスを無効にするには、「Reachability Analyzer ユーザーガイド」の「Cross-account analyses for Reachability Analyzer」(Reachability Analyzer のクロスアカウント分析) を参照してください。

Reachability Analyzer 用の委任管理者アカウントの有効化

委任管理者アカウントは、組織内のどのリソースでも接続分析を実行できます。詳細については、「Reachability Analyzer ユーザーガイド」の「Reachability Analyzer を AWS Organizations と統合する」を参照してください。

Reachability Analyzer の委任管理者を設定できるのは、組織の管理アカウントの管理者のみです。

委任管理者アカウントは、Reachability Analyzer コンソールから、または `RegisterDelegatedAdministrator` API を使用して指定できます。詳細については、「Organizations Command Reference」(Organizations コマンドリファレンス) で、「RegisterDelegatedAdministrator」を参照してください。

> **最小アクセス許可**
>
> Organizations 管理アカウントのユーザーまたはロールのみが、組織内で Reachability Analyzer の委任管理者としてメンバーアカウントを設定できます

Reachability Analyzer コンソールを使用して委任管理者を設定するには、「Reachability Analyzer ユーザーガイド」の「Reachability Analyzer を AWS Organizations と統合する」を参照してください。

Reachability Analyzer 用の委任管理者の無効化

Reachability Analyzer の委任管理者を設定できるのは、組織の管理アカウントの管理者のみです。

Reachability Analyzer コンソールまたは API、あるいは Organizations `DeregisterDelegatedAdministrator` CLI または SDK オペレーションを使用して、委任管理者アカウントを削除できます。

Reachability Analyzer コンソールを使用して委任管理者の Reachability Analyzer アカウントを無効にするには、「Reachability Analyzer ユーザーガイド」の「Cross-account analyses for Reachability Analyzer」(Reachability Analyzer のクロスアカウント分析) を参照してください。

Organizations と連携する AWS サービスの委任管理者

AWS Organizations 管理アカウントとそのユーザーおよびロールは、そのアカウントで実行する必要のあるタスクのみに使用することをおすすめします。また、すべての AWS リソースを組織内の他のメンバーアカウントに保存し、管理アカウントからは切り離すことをおすすめします。これは、Organizations のサービスコントロールポリシー (SCP) などのセキュリティ機能は、管理アカウントのユーザーやロールに制限を加えることができないためです。また、リソースを管理アカウントから分離することで、請求書に記載される請求額が理解しやすくなります。

Organizations と統合される AWS サービスでは、多くの場合管理アカウントの使用量を減らすことができます。これらのサービスでは、サービスで使用される組織のアカウントをすべて管理できる管理者として 1 つ以上のメンバーアカウントを登録できます。これらのアカウントは、その特定のサービスの委任管理者と呼ばれます。メンバーアカウントを AWS サービスの委任管理者として登録すると、そのアカウントに、そのサービスに対する一部の管理者権限と、Organizations の読み取り専用アクションの権限が付与されます。

アカウントをあるサービスの委任管理者として登録する前に:

- そのサービスが委任管理者をサポートしていることを確認します。どのサービスが委任管理者をサポートしているかについては、AWS で使用できる のサービス AWS Organizations の表を参照してください。
- そのサービスでの信頼されたアクセスを有効にします。

> **Note**
>
> 委任管理者にサービスを有効にする方法については、AWS で使用できる のサービス AWS Organizations の表を参照して、そのサービスの [委任管理者のサポート] 列にある [詳細] リンクを選択してください。

委任管理者アカウントに付与された権限

各サービス固有の委任管理者アカウントには、そのサービスによって権限が付与されます。詳細については、AWS で使用できる のサービス AWS Organizations の表を参照して、そのサービスの [委任管理者のサポート] 列にある [詳細] リンクを選択してください。

委任管理者アカウントには、次の読み取り専用権限もあります。

- `DescribeAccount`
- `DescribeCreateAccountStatus`
- `DescribeEffectivePolicy`
- `DescribeHandshake`
- `DescribeOrganization`
- `DescribeOrganizationalUnit`
- `DescribePolicy`
- `DescribeResourcePolicy`
- `ListAccounts`
- `ListAccountsForParent`
- `ListAWSServiceAccessForOrganization`
- `ListChildren`
- `ListCreateAccountStatus`
- `ListDelegatedAdministrators`
- `ListDelegatedServicesForAccount`
- `ListHandshakesForAccount`
- `ListHandshakesForOrganization`
- `ListOrganizationalUnitsForParent`
- `ListParents`
- `ListPolicies`
- `ListPoliciesForTarget`
- `ListRoots`
- `ListTagsForResource`
- `ListTargetsForPolicy`

これらの権限により、次のコンソール項目を表示できますが、変更はできません。

- 組織構造、すべてのアカウントと OU、組織ポリシー
- メンバーシップ
- すべてのアカウントと OU
- 組織方針

のセキュリティ AWS Organizations

AWS クラウドセキュリティは最優先事項です。 AWS 顧客は、最もセキュリティに敏感な組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャの恩恵を受けることができます。

セキュリティは、 AWS お客様とお客様との間で共有される責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ — AWS AWS AWS クラウド内で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があります。 AWS また、安全に使用できるサービスも提供します。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。に適用されるコンプライアンスプログラムについて詳しくは AWS Organizations、「AWS コンプライアンスプログラム別の対象サービス」を参照してください。
- クラウドにおけるセキュリティ — お客様の責任は、 AWS 使用するサービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、Organizations.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 Organizations を設定する方法を説明します。また、Organizations AWS のリソースの監視と保護に役立つ他のサービスの使い方についても学びます。

トピック

- AWS PrivateLink にとって AWS Organizations
- AWS Identity and Access Management と AWS Organizations
- での口グ記録とモニタリング AWS Organizations
- AWS Organizationsのコンプライアンス検証
- AWS Organizations での耐障害性
- AWS Organizations でのインフラストラクチャセキュリティ

AWS PrivateLink にとって AWS Organizations

AWS PrivateLink for を使用すると AWS Organizations、パブリックインターネットを経由しなくても、Virtual Private Cloud (VPC) AWS Organizations 内からサービスにアクセスできます。

Amazon VPC では、 AWS カスタム仮想ネットワークでリソースを起動できます。VPCを使用して、IP アドレス範囲、サブネット、ルートテーブル、ネットワークゲートウェイなどのネットワーク設定を制御できます。Amazon VPC の詳細については、「Amazon VPC ユーザーガイド」を参照してください。

Amazon VPC を接続するには AWS Organizations、まずインターフェイス VPC エンドポイント (インターフェイスエンドポイント) を定義する必要があります。インターフェイスエンドポイントは、VPC 内のサブネットからプライベート IP アドレスが割り当てられた 1 つ以上の Elastic Network Interface (ENI) で表されます。VPC AWS Organizations からインターフェイスエンドポイントへのリクエストは、Amazon ネットワークに残ります。

インターフェイスエンドポイントに関する一般的な情報については、Amazon VPC ユーザーガイドの「インターフェイス VPC AWS エンドポイントを使用してサービスにアクセスする」を参照してください。

トピック

- for の制限と制約 AWS PrivateLinkAWS Organizations
- VPC エンドポイントの作成
- AWS Organizations用の VPC エンドポイントポリシーの作成

for の制限と制約 AWS PrivateLinkAWS Organizations

には VPC AWS PrivateLink AWS Organizationsの制限が適用されます。詳細については、Amazon VPC ユーザーガイドの「インターフェイス VPC エンドポイント、 AWSAWS PrivateLink クォータを使用してサービスにアクセスする」を参照してください。また、以下の制限も適用されます。

- このリージョンでのみ利用可能です。`us-east-1`
- トランスポート層セキュリティ (TLS) 1.1 はサポートしていません

VPC エンドポイントの作成

Amazon VPC コンソール AWS Command Line Interface (AWS CLI) またはを使用して VPC AWS Organizations にエンドポイントを作成できます。 AWS CloudFormation

Amazon VPC コンソールまたはを使用してエンドポイントを作成および設定する方法については AWS CLI、Amazon VPC ユーザーガイドの「VPC エンドポイントの作成」を参照してください。を使用してエンドポイントを作成および設定する方法については AWS CloudFormation、ユーザーガイドの AWS:: EC2:: VpcEndpoint リソースを参照してください。AWS CloudFormation

AWS Organizations エンドポイントを作成するときは、以下をサービス名として使用します。

```
com.amazonaws.us-east-1.organizations
```

アクセス時に FIPS 140-2 検証済みの暗号モジュールが必要な場合は AWS、次の AWS Organizations FIPS サービス名を使用してください。

```
com.amazonaws.us-east-1.organizations-fips
```

AWS Organizations用の VPC エンドポイントポリシーの作成

Organizations へのアクセスを制御する VPC エンドポイントにエンドポイントポリシーをアタッチできます。このポリシーでは、以下の情報を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、Amazon VPC ユーザーガイドの「エンドポイントポリシーを使用して VPC エンドポイントへのアクセスを制御する」を参照してください。

例: AWS Organizations アクション用の VPC エンドポイントポリシー

```
{
   "Statement":[
      {
         "Principal":"*",
         "Effect":"Allow",
```

```
         "Action":[
            "Organizations:DescribeAccount"
         ],
         "Resource":"*"
      }
   ]
}
```

AWS Identity and Access Management と AWS Organizations

AWS Organizations へのアクセスには、認証情報が必要です。これらの認証情報には、Amazon Simple Storage Service (Amazon S3) バケット、Amazon Elastic Compute Cloud (Amazon EC2) インスタンス、AWS Organizations 組織単位 (OU) などの AWS リソースへのアクセス許可がある必要があります。次のセクションでは、AWS Identity and Access Management (IAM) を使用して組織に安全にアクセスし、組織を管理できるユーザーを制御する方法の詳細を説明します。

組織のどの部分を誰が管理するかを決定するために、AWS Organizations は他の AWS サービスと同じ IAM ベースのアクセス許可モデルを使用します。組織の管理アカウントの管理者は、IAM ベースのアクセス許可を付与して、マスターアカウントのユーザー、グループ、ロールにポリシーをアタッチすることで AWS Organizations タスクを実行できます。これらのポリシーは、それらのプリンシパルが実行できるアクションを指定します。そのユーザーがメンバーとして含まれているグループ、またはユーザーかロールに直接、IAM アクセス許可ポリシーをアタッチします。ベストプラクティスとして、ユーザーではなくグループにポリシーをアタッチすることをおすすめします。完全な管理者アクセス許可を他のユーザーに付与することもできます。

AWS Organizations のほとんどの管理者のオペレーションでは、管理アカウントのユーザーまたはグループにアクセス許可をアタッチする必要があります。メンバーアカウントのユーザーが組織の管理オペレーションを実行する必要がある場合は、メンバーアカウントのユーザーがロールを引き受けられるように、AWS Organizations アクセス許可を管理アカウントの IAM ロールに付与します。IAM アクセス許可ポリシーの一般的な情報については、IAM ユーザーガイドの「IAM ポリシーの概要」を参照してください。

トピック

- 認証
- アクセスコントロール
- AWS 組織へのアクセス許可の管理
- AWS Organizations でアイデンティティベースのポリシー (IAM ポリシー) を使用する

- タグおよび AWS Organizations による属性ベースのアクセスコントロール

認証

AWS には、次のいずれかのタイプの ID でアクセスできます。

- AWS アカウント のルートユーザー - AWS にサインアップする際は、AWS アカウント に関連付けられた E メールアドレスとパスワードを指定します。これらは [ルート認証情報]であり、これらの情報を使用すると、すべての AWS リソースに完全にアクセスできるようになります。

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

- IAM ユーザー - IAM ユーザーは、特定のカスタムアクセス許可 (例: Amazon Elastic File System でファイルシステムを作成するアクセス許可) を持つ AWS アカウント 内の ID です。IAM のユーザー名とパスワードは、AWS Management Console、AWS ディスカッションフォーラム、または AWS サポートセンターなどのセキュアな AWS ウェブページへのサインインに使用できます。

  ユーザー名とパスワードに加えて、各ユーザーのアクセスキーを生成できます。複数の SDK の 1 つを通して、または AWS Command Line Interface (AWS CLI) を使用して、プログラムで AWS のサービスにアクセスするときに、これらのキーを使用します。SDK と AWS CLI ツールでは、アクセスキーを使用してリクエストが暗号で署名されます。AWS ツールを使用しない場合は、リクエストを自分で署名する必要があります。AWS Organizations では、署名バージョン 4 がサポートされています。これは、インバウンド API リクエストを認証するためのプロトコルです。リクエストの認証の詳細については、IAM ユーザーガイドの API AWSリクエストの署名を参照してください。

- IAM ロール - IAM ロールは、特定のアクセス許可を持ち、アカウントで作成できるもう 1 つの IAM ID 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。IAM ロールでは、AWS のサービスおよびリソースにアクセスできる一時的なアクセスキーを取得することができます。IAM ロールと一時的な認証情報は、次の状況で役立ちます。

  - フェデレーティッドユーザーアクセス – IAM ユーザーを作成するのではなく、AWS Directory Service、エンタープライズユーザーディレクトリ、またはウェブアイデンティティプロバイ

ダーの既存のユーザーアイデンティティを使用することもできます。このようなユーザーはフェデレーティッドユーザーと呼ばれます。AWS では、ID プロバイダーを通じてアクセスがリクエストされたとき、フェデレーティッドユーザーにロールを割り当てます。フェデレーションユーザーの詳細については、「IAM ユーザーガイド」の「フェデレーションユーザーとロール」を参照してください。

- クロスアカウントアクセス - アカウントの IAM ロールを使用して、アカウントのリソースにアクセスするための別の AWS アカウント アクセス許可を付与できます。この例については、IAM ユーザーガイドの「チュートリアル: AWS アカウント 間の IAM ロールを使用したアクセス許可の委任」を参照してください。
- AWS のサービスのアクセス - アカウントで IAM ロールを使用して、アカウントのリソースにアクセスするための AWS のサービスのアクセス許可を付与できます。例えば、Amazon Redshift がお客様に代わって Amazon S3 バケットにアクセスし、バケットに保存されたデータを Amazon Redshift クラスターにロードすることを許可するロールを作成できます。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。
- Amazon EC2 で実行されるアプリケーション - インスタンス上で実行されるアプリケーションが AWS API リクエストを実行する際に使用するアクセスキーを EC2 インスタンスに保存する代わりに、IAM ロールを使用して、これらのアプリケーション用の一時的認証情報を管理できます。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認証情報を取得することができます。詳細については、IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」を参照してください。

アクセスコントロール

有効な認証情報があればリクエストを認証できますが、アクセス許可が付与されている場合を除き、AWS Organizations リソースを管理またはアクセスすることはできません。たとえば、OU を作成したり、サービスコントロールポリシー (SCP) をアカウントにアタッチするには、アクセス許可が必要です。

次のセクションでは、AWS Organizations の許可を管理する方法について説明します。

- AWS 組織へのアクセス許可の管理
- AWS Organizations でアイデンティティベースのポリシー (IAM ポリシー) を使用する

- タグおよび AWS Organizations による属性ベースのアクセスコントロール

AWS 組織へのアクセス許可の管理

組織のルート、OU、アカウント、ポリシーなどの AWS リソースはすべて、AWS アカウント によって所有され、リソースの作成またはアクセスを行うアクセス許可は、アクセス許可ポリシーによって管理されます。組織では、管理アカウントはすべてのリソースを所有します。アカウント管理者は、IAM アイデンティティ (ユーザー、グループ、ポリシー) にアクセス許可ポリシーをアタッチして、AWS リソースへのアクセスを制御できます。

> **Note**
>
> アカウント管理者 (または管理者ユーザー) は、管理者アクセス許可を持つユーザーです。詳細については、「IAM ユーザーガイド」の「IAM でのセキュリティベストプラクティス」を参照してください。

アクセス許可を付与する場合、アクセス許可を取得するユーザー、取得するアクセス許可の対象となるリソース、およびそれらのリソースに対して許可される特定のアクションを決定します。

デフォルトでは、IAM ユーザー、グループ、ロールにはアクセス許可がありません。組織の管理アカウントの管理者として管理タスクを行うか、管理アカウントの他の IAM ユーザーまたはロールに管理者のアクセス許可を委任できます。そのためには、IAM ユーザー、グループ、またはロールに IAM アクセス許可ポリシーをアタッチします。デフォルトでは、ユーザーにこれらを行う権限はありません。これは、暗黙的な拒否と呼ばれます。このポリシーによって、暗黙的な拒否は、ユーザーが実行するアクションや、アクションを実行できるリソースを指定する明示的な許可に上書きされます。ロールにアクセス許可が付与されている場合は、組織内の他のアカウントのユーザーはそのロールを引き受けることができます。

AWS Organizations リソースおよびオペレーション

本セクションでは、AWS Organizations 概念を IAM と同等の概念とどのようにマッピングされるかを説明します。

リソース

AWS Organizations では、次のリソースへのアクセスをコントロールできます。

- 組織の階層構造を構成するルートおよび OU

- 組織のメンバーであるアカウント
- 組織のエンティティにアタッチするポリシー
- 組織の状態の変更に使用するハンドシェイク

このようなリソースにはそれぞれ、一意の Amazon リソースネーム (ARN) が関連付けられています。IAM アクセス許可ポリシーの Resource 要素の ARN を指定して、リソースへのアクセスをコントロールします。で使用されるリソースの ARN 形式の詳細なリストについてはAWS Organizations、「サービス認証リファレンス」の「 で定義されるリソースタイプAWS Organizations」を参照してください。

オペレーション

AWS には、組織のリソースを操作する一連のオペレーションが用意されています。そのため、リソースの作成、一覧表示、変更、内容へのアクセス、削除などを行うことができます。ほとんどのオペレーションは、オペレーションの実行権限を制御する IAM ポリシーの Action 要素で参照できます。IAM ポリシーのアクセス許可として使用できるAWS Organizationsオペレー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AWS「 Organizations で定義されるアクション」を参照してください。

Action と Resource を 1 つのアクセス許可ポリシー Statement で組み合わせると、特定のアクションを使用できるリソースが正確に制御されます。

条件キー

AWS は特定のアクションにおいて、より詳細な制御を実現するクエリを実行できる条件キーを提供しています。IAM ポリシーの Condition 要素でこれらの条件キーを参照し、ステートメントが一致しているとみなされる条件を満たすように追加条件を指定することができます。

次の条件キーは、AWS Organizations を使用する場合に特に役立ちます。

- aws:PrincipalOrgID - リソースベースのポリシーの Principal 要素の指定を簡素化します。このグローバルキーは、組織内のすべての AWS アカウントのすべてのアカウント ID を一覧表示する代わりに使用できます。組織のメンバーであるすべてのアカウントを一覧表示せずに、Condition 要素に組織 ID を指定することができます。

> **Note**
>
> このグローバル条件は、組織の管理アカウントにも適用されます。

詳細については、「IAM ユーザーガイドPrincipalOrgID」のAWS「 グローバル条件コンテキストキーの の説明」を参照してください。

- aws:PrincipalOrgPaths - この条件キーを使用して、特定の組織ルート、OU、またはその子のメンバーを照合します。リクエストを行うプリンシパル (ルートユーザー、IAM ユーザー、またはロール) が指定された組織パスにある場合、aws:PrincipalOrgPaths 条件キーは true を返します。パスとは、AWS Organizations エンティティの構造をテキストで表記したものです。パスの詳細については、「IAM ユーザーガイド」のAWS Organizations「エンティティパスを理解する」を参照してください。この条件キーの使用の詳細については、「IAM ユーザーガイド」の「aws:PrincipalOrgPaths」を参照してください。

  例えば次の条件要素は、同じ組織内の 2 つの OU のいずれかのメンバーと一致します。

```
            "Condition": {
                "ForAnyValue:StringLike": {
                    "aws:PrincipalOrgPaths": [
                        "o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbbb/",
                        "o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-jkl0-awsddddd/"
                    ]
                }
            }
```

- organizations:PolicyType - この条件キーを使用して、Organization ポリシー関連 API オペレーションを、指定したタイプの Organization ポリシーでのみ動作するように制限できます。この条件キーは、Organizations ポリシーとやり取りするアクションを含むポリシーステートメントに適用できます。

  この条件キーでは、次の値を使用できます。

  - AISERVICES_OPT_OUT_POLICY
  - BACKUP_POLICY
  - SERVICE_CONTROL_POLICY
  - TAG_POLICY

  例えば、次のポリシー例では、ユーザーが任意の Organizations オペレーションを実行できます。ただし、ポリシー引数を取るオペレーションをユーザーが実行した場合、指定したポリシーがタグ付けポリシーである場合にのみオペレーションが許可されます。ユーザーが他のタイプのポリシーを指定した場合、オペレーションは失敗します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "IfTaggingAPIThenAllowOnOnlyTaggingPolicies",
            "Effect": "Allow",
            "Action": "organizations:*",
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "organizations:PolicyType": [ "TAG_POLICY" ]
                }
            }
        }
    ]
}
```

- `organizations:ServicePrincipal` – 他の AWSのサービスとの信頼されたアクセスを有効AWSServiceAccessまたは無効にするAWSServiceAccess操作を使用する場合、条件として使用できます。`organizations:ServicePrincipal` を使用して、これらのオペレーションからのリクエストを、承認されたサービスプリンシパル名のリストに制限できます。

  例えば以下のポリシーでは、ユーザーは、AWS Firewall Manager を使用して信頼されたアクセスを有効および無効にする場合のみ、AWS Organizations を指定することができます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowOnlyAWSFirewallIntegration",
            "Effect": "Allow",
            "Action": [
                "organizations:EnableAWSServiceAccess",
                "organizations:DisableAWSServiceAccess"
            ],
            "Resource": "*",
            "Condition": {
                "StringLikeIfExists": {
                    "organizations:ServicePrincipal": [ "fms.amazonaws.com" ]
                }
            }
        }
```

```
    ]
}
```

IAM ポリシーでアクセス許可として使用できる AWS Organizations固有の条件キーの一覧については、「サービス認証リファレンス」の「 の条件キーAWS Organizations」を参照してください。

リソース所有権について

AWS アカウント は、誰がリソースを作成したかにかかわらず、アカウントで作成されたリソースを所有します。具体的には、リソース所有者は、リソース作成リクエストを認証する「プリンシパルエンティティ」(つまり、ルートユーザー、IAM ユーザー、IAM ロール) の AWS アカウント です。AWS の組織の場合は、これは常に管理アカウントになります。組織のリソースの作成またはアクセスを行うほとんどのオペレーションは、メンバーアカウントから呼び出すことができません。次の例は、この仕組みを示しています。

- 管理アカウントのルートアカウントの認証情報を使用して OU を作成する場合の管理アカウントは、リソースの所有者です。(AWS Organizations で、リソースは OU です。)
- 管理アカウントに IAM ユーザーを作成し、そのユーザーに OU を作成するためのアクセス権限を付与する場合、そのユーザーは OU を作成できます。ただし、OU リソースを所有しているのは、このユーザーが属する管理アカウントです。
- OU を作成するためのアクセス権限を持つ管理アカウントに IAM ロールを作成する場合は、ロールを引き受けることのできるユーザーはいずれも OU を作成できます。OU リソースを所有するのは、ロール (引き受けるユーザーではない) が属する管理アカウントです。

リソースへのアクセスの管理

アクセス権限ポリシー では、誰が何にアクセスできるかを記述します。次のセクションで、アクセス許可ポリシーを作成するために使用可能なオプションについて説明します。

> **Note**
>
> このセクションでは、AWS Organizations のコンテキストでの IAM の使用について説明します。これは、IAM サービスに関する詳細情報を取得できません。完全な IAM ドキュメントについては、「IAM ユーザーガイド」を参照してください。IAM ポリシーの構文と説明については、「IAM ユーザーガイド」の「IAM JSON ポリシーリファレンス」を参照してください。

IAM 認証情報にアタッチされたポリシーは、アイデンティティベースのポリシー (IAM ポリシー) と呼ばれます。リソースにアタッチされたポリシーを、リソースベースのポリシーと呼びます。AWS Organizations は、ID ベースのポリシー (IAMポリシー) のみをサポートします。

トピック

- ID ベースのアクセス許可ポリシー (IAM ポリシー)
- リソースベースのポリシー

ID ベースのアクセス許可ポリシー (IAM ポリシー)

ポリシーを IAM ID にアタッチして、その ID が AWS リソースに対してオペレーションを実行できるようにすることができます。例えば、次の操作を実行できます。

- アカウントのユーザーまたはグループにアクセス許可ポリシーをアタッチする - サービスコントロールポリシー (SCP) または OU などの AWS Organizations リソースを作成するための権限をユーザーに付与するには、ユーザーまたはユーザーが所属するグループにアクセス許可ポリシーをアタッチします。ユーザーまたはグループを組織の管理アカウントにする必要があります。
- アクセス許可ポリシーをロールにアタッチする (クロスアカウントのアクセス許可を付与する) - アイデンティティベースのアクセス許可ポリシーを IAM ロールにアタッチして、クロスアカウントのアクセス許可を組織に付与することができます。例えば、管理アカウントの管理者はロールを作成し、次のようにクロスアカウントのアクセス許可をメンバーアカウントのユーザーに付与できます。

  1. 管理アカウントの管理者は IAM ロールを作成し、ロールにアクセス許可ポリシーをアタッチして組織のリソースにアクセス許可を付与します。
  2. 管理アカウントの管理者は、そのロールを引き受ける `Principal` として、メンバーアカウントの ID を識別するロールに信頼ポリシーをアタッチします。
  3. その後、メンバーアカウント管理者は、ロールを引き受けるアクセス権限をメンバーアカウントのユーザーに委任できます。これにより、メンバーアカウントのユーザーは、管理アカウントや組織のリソースを作成し、アクセスできるようになります。ロールを引き受けるアクセス許可を AWS サービスに付与する場合は、信頼ポリシーのプリンシパルも AWS サービスのプリンシパルにできます。

  IAM を使用した許可の委任の詳細については、「IAM ユーザーガイド」の「アクセス管理」を参照してください。

以下に示しているのは、組織の CreateAccount アクションの実行をユーザーに許可するポリシーの例です。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"Stmt1OrgPermissions",
         "Effect":"Allow",
         "Action":[
            "organizations:CreateAccount"
         ],
         "Resource":"*"
      }
   ]
}
```

ポリシーの Resource 要素に部分的な ARN を指定して、リソースのタイプを示すこともできます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowCreatingAccountsOnResource",
         "Effect":"Allow",
         "Action":"organizations:CreateAccount",
         "Resource":"arn:aws:organizations::*:account/*"
      }
   ]
}
```

作成するアカウントに特定のタグを含まないアカウントの作成を拒否することもできます。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyCreatingAccountsOnResourceBasedOnTag",
         "Effect":"Deny",
         "Action":"organizations:CreateAccount",
         "Resource":"*",
```

```
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/key":"value"
            }
         }
      }
   ]
}
```

ユーザー、グループ、ロール、アクセス許可の詳細については、IAM ユーザーガイドの「IAM ID (ユーザー、ユーザーグループ、ロール)」を参照してください。

リソースベースのポリシー

Amazon S3 などの一部のサービスでは、リソースベースのアクセス許可ポリシーもサポートされています。例えば、ポリシーを Amazon S3 バケットにアタッチして、そのバケットに対するアクセス許可を管理できます。AWS Organizations では現在、リソースベースのポリシーはサポートされていません。

ポリシー要素の指定: アクション、条件、効果、リソース

AWS Organizations リソース については、サービスは、一連の API オペレーション、またはアクションを定義し、何らかの形でリソースと相互作用するか、リソースを操作します。これらのオペレーションを実行するためのアクセス権限を付与するために、AWS Organizations ではポリシーに一連のアクションを定義できます。例えば、OU リソースの場合は、AWS Organizations によって次のようなアクションが定義されています。

- `AttachPolicy` および `DetachPolicy`
- `CreateOrganizationalUnit` および `DeleteOrganizationalUnit`
- `ListOrganizationalUnits` および `DescribeOrganizationalUnit`

場合によっては、API オペレーションを実行する際、アクションまたはリソースのアクセス許可が 2 つ以上必要になることがあります。

IAM アクセス許可ポリシーで使用できる最もベーシックなポリシーは、次のとおりです。

- Action - このキーワードを使用して、許可または拒否するオペレーション (アクション) を識別します。例えば、指定した `Effect` に応じて、`organizations:CreateAccount` では、AWS Organizations の `CreateAccount` オペレーションを実行するユーザーのアクセス権限を許可また

は拒否します。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシーエレメント: アクション」を参照してください。

- Resource - このキーワードを使用して、ポリシー構文が適用されるリソースの ARN を指定します。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシーエレメント: リソース」を参照してください。
- Condition - ポリシーステートメントを満たす必要がある条件を指定するために、このキーワードを使用します。Condition は通常、ポリシーが一致するために満たす必要がある追加条件を指定します。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。
- Effect - このキーワードを使用して、ポリシー構文でリソースのアクションを許可または拒否するかどうかを指定します。リソースへのアクセス権を明示的に付与 (または許可) していない場合、アクセスは暗黙的に拒否されます。また、リソースへのアクセスを明示的に拒否することもできます。これにより、別のポリシーによってアクセスが許可されていても、ユーザーは、指定のリソースで指定のアクションを実行できないことがあります。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシーエレメント: 効果」を参照してください。
- Principal - アイデンティティベースのポリシー (IAM ポリシー) では、ポリシーがアタッチされているユーザーが自動的かつ暗黙的にプリンシパルとなります。リソースベースのポリシーでは、権限 (リソースベースのポリシーにのみ適用) を受け取りたいユーザー、アカウント、サービス、またはその他のエンティティを指定します。現在、AWS Organizations では、アイデンティティベースのポリシーのみサポートしており、リソースベースのポリシーはサポートしていません。

IAM ポリシーの構文と説明の詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシーリファレンス」を参照してください。

AWS Organizations でアイデンティティベースのポリシー (IAM ポリシー) を使用する

組織の管理アカウントの管理者として、組織内の AWS (IAM) アイデンティティ (ユーザー、グループ、ロール) にアクセス許可ポリシーをアタッチすることで、AWS Identity and Access Management リソースへのユーザーのアクセスをコントロールすることができます。アクセス許可を付与する場合、アクセス許可を取得するユーザー、取得するアクセス許可の対象となるリソース、およびそれらのリソースに対して許可される特定のアクションを決定します。ロールにアクセス権限が付与されている場合は、組織内の他のアカウントのユーザーがそのロールを引き受けることができます。

デフォルトでは、ユーザーには何もアクセス権限が与えられていません。すべてのアクセス権限は、ポリシーで明示的に付与されている必要があります。アクセス許可が明示的に付与されていない場合

は、暗黙的に拒否されます。アクセス許可が明示的に拒否されている場合、許可されている可能性があるその他のポリシーより優先されます。つまり、ユーザーは明示的に付与されていて、明示的に拒否されていないアクセス許可だけを持つことになります。

このトピックで説明する基本的な方法に加えて、組織内のリソースに適用されるタグ (組織ルート、組織単位 (OU)、アカウント、およびポリシー) を使用して、組織へのアクセスをコントロールできます。詳細については、「タグおよび AWS Organizations による属性ベースのアクセスコントロール」を参照してください。

完全な管理者権限をユーザーに付与する

組織の IAM ユーザーに、完全な AWS Organizations 管理者権限を付与する IAM ポリシーを作成できます。これを行うには、IAM コンソールの JSON ポリシーエディタを使用します。

JSON ポリシーエディタでポリシーを作成するには

1. AWS Management Console にサインインして、IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. 左側のナビゲーションペインで、[ポリシー] を選択します。

   初めて [ポリシー] を選択する場合には、[管理ポリシーにようこそ] ページが表示されます。[今すぐ始める] を選択します。
3. ページの上部で、[ポリシーを作成] を選択します。
4. [ポリシーエディタ] セクションで、[JSON] オプションを選択します。
5. 次の JSON ポリシードキュメントを入力します。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "organizations:*",
        "Resource": "*"
    }
}
```

6. [次へ] をクリックします。

> **Note**
>
> いつでも [Visual] と [JSON] エディタオプションを切り替えることができます。ただし、[Visual] エディタで [次] に変更または選択した場合、IAM はポリシーを再構成して visual エディタに合わせて最適化することがあります。詳細については、『IAM ユーザーガイド』の「ポリシーの再構成」を参照してください。

7. [確認と作成] ページで、作成するポリシーの [ポリシー名] と [説明] (オプション) を入力します。[このポリシーで定義されているアクセス許可] を確認して、ポリシーによって付与されたアクセス許可を確認します。
8. [ポリシーの作成] をクリックして、新しいポリシーを保存します。

IAM ポリシーの作成の詳細については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

制限付きのアクセス許可をアクションごとに付与する

完全なアクセス許可ではなく制限されたアクセス許可を付与する場合は、IAM アクセス許可ポリシーの `Action` 要素で許可する個々のアクセス許可をリストするポリシーを作成できます。次の例に示すように、ワイルドカード (*) 文字を使用して `Describe*` および `List*` のアクセス権限のみを付与することができます。この方法では通常、読み取り専用アクセスが組織に付与されます。

> **Note**
>
> サービスコントロールポリシー (SCP) では、`Action` 要素のワイルドカード (*) 文字は、ポリシー自体、または文字列の末尾にのみ使用できます。文字列の先頭または中間には表示されません。そのため、`"servicename:action*"` は有効ですが、`"servicename:*action"` と `"servicename:some*action"` はいずれも、SCP で無効です。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "organizations:Describe*",
```

```
            "organizations:List*"
        ],
        "Resource": "*"
    }
}
```

IAM ポリシーで割り当てることができるすべてのアクセス許可のリストについては、「サービス認証リファレンス」のAWS「 Organizations で定義されるアクション」を参照してください。

特定のリソースへのアクセス許可の付与

特定のアクションへのアクセスの制限に加えて、組織内の特定のエンティティへのアクセスを制限できます。前のセクションの例の Resource 要素は、両方ワイルドカード文字 (*) を指定します。つまり、「アクションがアクセスできる任意のリソース」を意味します。代わりに、「*」をアクセスを許可する特定のエンティティの Amazon リソースネーム (ARN) で置き換えることができます。

例: 単一の OU にアクセス許可を付与する

次のポリシーの最初のステートメントは、IAM ユーザーに組織全体への読み取りアクセスを許可していますが、2 番目のステートメントでは、ユーザーは指定された単一の組織単位 (OU) 内でのみ AWS Organizations 管理アクションの実行が許可されます。これは、子 OU には適用されません。請求へのアクセスは付与されません。ただし、OU 内の AWS アカウント への管理アクセスはできません。指定された OU 内のアカウントで AWS Organizations オペレーションを実行するためのアクセス許可のみが付与されます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "organizations:Describe*",
        "organizations:List*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "organizations:*",
      "Resource": "arn:aws:organizations::<masterAccountId>:ou/o-<organizationId>/ou-
<organizationalUnitId>"
    }
```

```
  ]
}
```

OU と組織の ID は AWS Organizations コンソールから取得するか、または `List*` API を呼び出して取得します。このポリシーを適用されたユーザーまたはグループは、OU に直接含まれるエンティティに対して任意のアクション (`"organizations:*"`) を実行できます。OU は Amazon リソースネーム (ARN) によって識別されます。

さまざまなリソースの ARNs「サービス認証リファレンス」の「 で定義されるリソースタイプAWS Organizations」を参照してください。

制限付きサービスプリンシパルに信頼されたアクセスを有効にするための権限を付与する

ポリシーステートメントの `Condition` 要素を使用して、ポリシーステートメントの一致をさらに制限することができます。

例: 特定した 1 つのサービスに信頼されたアクセスを有効にするための権限を付与する

次のステートメントは、特定したサービスのみに信頼されたアクセスを有効にするための制限方法を示しています。AWS IAM Identity Center のものではなく、別のサービスプリンシパルをユーザーが使用して API を呼び出す場合、このポリシーは一致せずリクエストが拒否されます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "organizations:EnableAWSServiceAccess",
            "Resource": "*",
            "Condition": {
                "StringEquals" : {
                    "organizations:ServicePrincipal" : "sso.amazonaws.com"
                }
            }
        }
    ]
}
```

さまざまなリソースの ARNs「サービス認証リファレンス」の「 で定義されるリソースタイプAWS Organizations」を参照してください。

タグおよび AWS Organizations による属性ベースのアクセスコントロール

属性ベースのアクセスコントロールでは、AWS リソースと AWS アイデンティティの両方にアタッチされたタグなどの管理者によって管理される属性を使用して、それらのリソースへのアクセスをコントロールすることができます。例えば、ユーザーとリソースの両方に特定のタグの同じ値がアタッチされている場合には、ユーザーがリソースにアクセスできるように指定できます。

AWS Organizations のタグ付け可能なリソースには、AWS アカウント、組織ルート、組織単位 (OU)、またはポリシーがあります。Organizations リソースにタグをアタッチすると、そのタグを使用して、それらのリソースにアクセスできるユーザーを制御できます。これを行うには、アクションを許可する前に、特定のタグキーと値が存在するかどうかを確認するための `Condition` 要素を AWS Identity and Access Management (IAM) アクセス許可ポリシーステートメントに追加します。これにより、「キー X と値 Y を含むタグがアタッチされた OU のみの管理をユーザに許可する」または「ユーザにアタッチされたタグキー Z と同じ値を含むキー Z でタグ付けされた OU のみの管理をユーザに許可する」という効果的な IAM ポリシーを作成することができます。

IAM ポリシーのさまざまなタイプのタグリファレンスに基づいて `Condition` テストを行うことができます。

- リクエストによって指定されたリソースにアタッチされたタグのチェック
- リクエストを行う IAM ユーザーまたはロールにアタッチされているタグを確認する
- リクエストのパラメータとして含まれているタグをチェックする

ポリシーによるアクセスコントロールでタグを使用する方法の詳細については、「リソースタグを使用した IAM ユーザーおよびロールへのアクセスのコントロール」を参照してください。IAM アクセス許可ポリシーの完全な構文については、「IAM JSON ポリシーリファレンス」を参照してください。

リクエストによって指定されたリソースにアタッチされたタグのチェック

AWS Management Console、AWS Command Line Interface (AWS CLI)、または AWS いずれかの SDK を使用してリクエストを行う場合は、そのリクエストによってアクセスするリソースを指定します。利用可能な特定のタイプのリソースの一覧表示、リソースの読み取り、リソースの書き込み、変更、更新のいずれかを行う場合は、アクセスするリソースをリクエストのパラメータとして指定します。これらのリクエストは、ユーザーとロールにアタッチする IAM アクセス許可ポリシーによって制御されます。これらのポリシーでは、リクエストされたリソースにアタッチされているタグを比較し、それらのタグのキーと値に応じてアクセスを許可または拒否できます。

リソースにアタッチされているタグを確認するには、タグキー名の前に aws:ResourceTag/ という文字列を付けて、Condition 要素内のタグを参照します。

例えば、以下のポリシー例では、リソースにキー department と値 security のタグがアタッチされない限り、ユーザーまたはロールはすべての AWS Organizations オペレーションを実行することができます。そのキーと値が存在する場合、ポリシーによって UntagResource オペレーションが明示的に拒否されます。

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
            "Effect" : "Allow",
            "Action" : "organizations:*",
            "Resource" : "*"

        },
        {
            "Effect" : "Deny",
            "Action" : "organizations:UntagResource",
            "Resource" : "*",
            "Condition" : {
                "StringEquals" : {
                    "aws:ResourceTag/department" : "security"
                }
            }
        }
    ]
}
```

この要素の使用方法の詳細については、IAM ユーザーガイドの「リソースへのアクセスのコントロール」および「aws:ResourceTag」を参照してください。

リクエストを行う IAM ユーザーまたはロールにアタッチされているタグを確認する

リクエストを行うユーザー (プリンシパル) が実行できるオペレーションを、そのユーザーの IAM ユーザーまたはロールにアタッチされたタグに応じて制御できます。これを行うには、aws:PrincipalTag/*key-name* 条件キーを使用して、呼び出し元のユーザーまたはロールにアタッチする必要のあるタグと値を指定します。

次の例では、指定されたタグ (cost-center) が、オペレーションを呼び出すプリンシパルと、オペレーションによってアクセスされるリソースの両方に同じ値が含まれる場合にのみ、アクションを許

可する方法について説明します。この例では、呼び出し元ユーザーは、インスタンスにユーザーと同じ cost-center 値がタグ付けされている場合にのみ、Amazon EC2 インスタンスを開始および停止できます。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "ec2:startInstances",
            "ec2:stopInstances"
        ],
        "Resource": "*",
        "Condition": {"StringEquals":
            {"ec2:ResourceTag/cost-center": "${aws:PrincipalTag/cost-center}"}}
    }
}
```

この要素の使用方法の詳細については、IAM ユーザーガイドの「IAM プリンシパルのアクセスのコントロール」および「aws:PrincipalTag」を参照してください。

リクエストのパラメータとして含まれているタグをチェックする

いくつかのオペレーションでは、リクエストの一部としてタグを指定できます。例えば、リソースを作成する際に、新しいリソースにアタッチされるタグを指定できます。aws:TagKeys を使用する Condition 要素を指定し、特定のタグキーまたはキーのセットがリクエストに含まれているかどうかによって、オペレーションを許可または拒否できます。この比較演算子では、タグにどのような値が含まれていても問題ありません。指定されたキーのタグが存在するかどうかだけがチェックされます。

タグキーまたはキーのリストを確認するには、次の構文で Condition 要素を指定します。

```
"aws:TagKeys": [ "tag-key-1", "tag-key-2", ... , "tag-key-n" ]
```

比較演算子の前に ForAllValues: を使用することで、リクエストに含まれるすべてのキーが、ポリシーで指定されたキーの 1 つと確実に一致しているかどうかを確認できます。例えば次のポリシー例では、指定した 3 つすべてのタグキーがリクエストに存在する場合にのみ、Organizations のすべてのオペレーションを許可します。

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

```
    "Statement": {
        "Effect": "Allow",
        "Action": "organizations:*",
        "Resource": "*",
        "Condition": {
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "department",
                    "costcenter",
                    "manager"
                ]
            }
        }
    }
}
```

あるいは、ForAnyValue: を比較演算子の前に記述して使用し、リクエスト内の少なくとも 1 つのキーが、ポリシーで指定されたキーの 1 つと確実に一致しているかどうかを確認できます。例えば、次のポリシーでは、指定されたタグキーの少なくとも 1 つがリクエスト内に存在する場合にのみ、Organizations のオペレーションを許可します。

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": "organizations:*",
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals": {
                "aws:TagKeys": [
                    "stage",
                    "region",
                    "domain"
                ]
            }
        }
    }
}
```

複数のオペレーションによって、リクエストのタグを指定できます。例えば、リソースを作成する際に、新しいリソースにアタッチされるタグを指定できます。ポリシー内のタグキーと値のペアと、リクエストに含まれるキーと値のペアを比較できます。これを行うには、タグのキー名の前に

aws:RequestTag/*key-name* という文字列を付けて Condition 要素内のタグを参照し、含まれるべきタグの値を指定します。

例えば、次のポリシー例では、ユーザーまたはロールが AWS アカウント を作成するためのリクエストに costcenter タグが含まれていない場合、またはタグに 1、2、3 以外の値が指定されている場合に、そのリクエストが拒否されます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "organizations:CreateAccount",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/costcenter": "true"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": "organizations:CreateAccount",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "aws:RequestTag/costcenter": [
                        "1",
                        "2",
                        "3"
                    ]
                }
            }
        }
    ]
}
```

これらの要素を使用する方法の詳細については、IAM ユーザーガイドの「aws:TagKeys」および「aws:RequestTag」を参照してください。

でのログ記録とモニタリング AWS Organizations

ベストプラクティスとして、変更がログに記録されることを確実にするために組織を監視する必要があります。これにより、予期しない変更を調査でき、不要な変更をロールバックできます。AWS Organizations は現在、組織とその内部で発生するアクティビティをモニタリングできる 2 つの AWS サービスをサポートしています。

トピック

- を使用した AWS Organizations API コールのログ記録 AWS CloudTrail
- Amazon EventBridge

を使用した AWS Organizations API コールのログ記録 AWS CloudTrail

AWS Organizations は、 と統合されています。これは AWS CloudTrail、 のユーザー、ロール、または AWS サービスによって実行されたアクションを記録するサービスです AWS Organizations。は、 AWS Organizations コンソールからの呼び出しや API へのコード呼び出しを含む、 のすべての API 呼び出しをイベント AWS Organizations として CloudTrail キャプチャします。 AWS Organizations APIs 証跡を作成する場合は、 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 AWS Organizations。証跡を設定しない場合でも、 CloudTrail コンソールのイベント履歴 で最新のイベントを表示できます。で収集された情報を使用して CloudTrail、 に対するリクエスト AWS Organizations、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。

> **Important**
>
> のすべての CloudTrail 情報は、米国東部 (バージニア北部) リージョン AWS Organizations でのみ表示できます。 CloudTrail コンソールに AWS Organizations アクティビティが表示されない場合は、右上隅のメニューを使用して、コンソールを米国東部 (バージニア北部) に設定します。 CloudTrail AWS CLI または SDK ツールを使用してクエリを実行する場合は、クエリを米国東部 (バージニア北部) エンドポイントに転送します。

AWS Organizations の情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 で が有効になります。でアクティビティが発生すると AWS Organizations、そのアクティビティは CloudTrail イベント履歴 の他の AWS サー

ビスイベントとともにイベントに記録されます。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「イベント履歴を使用した CloudTrail イベントの表示」を参照してください。

AWS Organizationsのイベントなど、 AWS アカウントのイベントの継続的な記録に対して、追跡を作成します。証跡により CloudTrail 、 はログファイルを Amazon S3 バケットに配信できます。で CloudTrail ログ記録が有効になっている場合 AWS アカウント、 AWS Organizations アクションに対して行われた API コールは CloudTrail ログファイルで追跡され、他の AWS サービスレコードとともに書き込まれます。 CloudTrail ログで収集されたイベントデータをさらに分析して処理するように、他の AWS サービスを設定できます。詳細については、次を参照してください:

- 証跡の作成のための概要
- CloudTrail サポートされているサービスと統合
- の Amazon SNS 通知の設定 CloudTrail

すべての AWS Organizations アクションは によってログに記録 CloudTrail され、 AWS Organizations API リファレンス に記載されています。例えば、 `CreateAccount` (`CreateAccountResult`イベントを含む) `ListHandshakesForAccount`、、、 を呼び出すと`CreatePolicy`、 CloudTrail ログファイルにエントリ`InviteAccountToOrganization`が生成されます。

各ログエントリには、リクエストの生成者に関する情報が含まれます。ログエントリのユーザーアイデンティティ情報は、次のことを確認するのに役立ちます。

- リクエストが、ルートユーザーまたは IAM ユーザーのどちらの認証情報を使用して送信されたかどうか
- リクエストが、IAM ロールの一時的なセキュリティ認証情報またはフェデレーティッドユーザーによって行われたかどうか
- リクエストが別の AWS サービスによって行われたかどうか

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

AWS Organizations ログファイルエントリについて

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。 CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラ

メータなどの情報を含みます。 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。

ログエントリの例: CloseAccount

次の例は、API がCloseAccount呼び出され、アカウントを閉鎖するワークフローがバックグラウンドで処理を開始するときに生成されるサンプル呼び出しの CloudTrail ログエントリを示しています。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAMVNPBQA3EXAMPLE:my-admin-role",
        "arn": "arn:aws:sts::111122223333:assumed-role/my-admin-role/my-session-id",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDAMVNPBQA3EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/my-admin-role",
                "accountId": "111122223333",
                "userName": "my-session-id"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2022-03-18T18:17:06Z"
            }
        }
    },
    "eventTime": "2022-03-18T18:17:06Z",
    "eventSource": "organizations.amazonaws.com",
    "eventName": "CloseAccount",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.168.0.1",
    "userAgent":  "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)...",
    "requestParameters": {
        "accountId": "555555555555"
    },
    "responseElements": null,
    "requestID": "e28932f8-d5da-4d7a-8238-ef74f3d5c09a",
```

```
    "eventID": "19fe4c10-f57e-4cb7-a2bc-6b5c30233592",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

次の例は、アカウントを正常に閉じるためのバックグラウンドワークフローが完了した後のCloseAccountResult呼び出しの CloudTrail ログエントリを示しています。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "organizations.amazonaws.com"
  },
  "eventTime": "2022-03-18T18:17:06Z",
  "eventSource": "organizations.amazonaws.com",
  "eventName": "CloseAccountResult",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "organizations.amazonaws.com",
  "userAgent": "organizations.amazonaws.com",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "closeAccountStatus": {
      "accountId": "555555555555",
      "state": "SUCCEEDED",
      "requestedTimestamp": "Mar 18, 2022 6:16:58 PM",
      "completedTimestamp": "Mar 18, 2022 6:16:58 PM"
    }
   },
   "eventCategory": "Management"
}
```

ログエントリの例: CreateAccount

次の例は、API がCreateAccount呼び出され、アカウントを作成するワークフローがバックグラウンドで処理を開始するときに生成されるサンプル呼び出しの CloudTrail ログエントリを示しています。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAMVNPBQA3EXAMPLE:my-admin-role",
        "arn": "arn:aws:sts::111122223333:assumed-role/my-admin-role/my-session-id",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDAMVNPBQA3EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/my-admin-role",
                "accountId": "111122223333",
                "userName": "my-session-id"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-09-16T21:16:45Z"
            }
        }
    },
    "eventTime": "2018-06-21T22:06:27Z",
    "eventSource": "organizations.amazonaws.com",
    "eventName": "CreateAccount",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.168.0.1",
    "userAgent":  "Mozilla/5.0 (Windows NT 10.0; Win64; x64)...",
    "requestParameters": {
        "tags": [],
        "email": "****",
        "accountName": "****"
    },
    "responseElements": {
        "createAccountStatus": {
            "accountName": "****",
```

```
            "state": "IN_PROGRESS",
            "id": "car-examplecreateaccountrequestid111",
            "requestedTimestamp": "Sep 16, 2020 9:20:50 PM"
        }
    },
    "requestID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```

次の例は、アカウントを正常に作成するためのバックグラウンドワークフローが完了した後のCreateAccount呼び出しの CloudTrail ログエントリを示しています。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "..."
  },
  "eventTime": "2020-09-16T21:20:53Z",
  "eventSource": "organizations.amazonaws.com",
  "eventName": "CreateAccountResult",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "....",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "createAccountStatus": {
      "id": "car-examplecreateaccountrequestid111",
      "state": "SUCCEEDED",
      "accountName": "****",
      "accountId": "444455556666",
      "requestedTimestamp": "Sep 16, 2020 9:20:50 PM",
      "completedTimestamp": "Sep 16, 2020 9:20:53 PM"
    }
  }
}
```

次の例は、CreateAccountバックグラウンドワークフローがアカウントの作成に失敗した後に生成される CloudTrail ログエントリを示しています。

```
  {
  "eventVersion": "1.06",
  "userIdentity": {
    "accountId": "111122223333",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2018-06-21T22:06:27Z",
  "eventSource": "organizations.amazonaws.com",
  "eventName": "CreateAccountResult",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
  "readOnly": false,
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "111122223333",
  "serviceEventDetails": {
    "createAccountStatus": {
      "id": "car-examplecreateaccountrequestid111",
      "state": "FAILED",
      "accountName": "****",
      "failureReason": "EMAIL_ALREADY_EXISTS",
      "requestedTimestamp": Jun 21, 2018 10:06:27 PM,
      "completedTimestamp": Jun 21, 2018 10:07:15 PM
    }
  }
}
```

ログエントリの例: CreateOrganizationalUnit

次の例は、サンプルCreateOrganizationalUnit呼び出しの CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "AIDAMVNPBQA3EXAMPLE",
        "arn": "arn:aws:iam::111111111111:user/diego",
        "accountId": "111111111111",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "diego"
    },
    "eventTime": "2017-01-18T21:40:11Z",
    "eventSource": "organizations.amazonaws.com",
    "eventName": "CreateOrganizationalUnit",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36",
    "requestParameters": {
        "name": "OU-Developers-1",
        "parentId": "r-a1b2"
    },
    "responseElements": {
        "organizationalUnit": {
            "arn": "arn:aws:organizations::111111111111:ou/o-aa111bb222/ou-
examplerootid111-exampleouid111",
            "id": "ou-examplerootid111-exampleouid111",
            "name": "test-cloud-trail"
        }
    },
    "requestID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```

ログエントリの例: InviteAccountToOrganization

次の例は、サンプルInviteAccountToOrganization呼び出しの CloudTrail ログエントリを示しています。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAMVNPBQA3EXAMPLE",
        "arn": "arn:aws:iam::111111111111:user/diego",
        "accountId": "111111111111",
```

```
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "diego"
    },
    "eventTime": "2017-01-18T21:41:17Z",
    "eventSource": "organizations.amazonaws.com",
    "eventName": "InviteAccountToOrganization",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36",
    "requestParameters": {
        "notes": "This is a request for Mary's account to join Diego's organization.",
        "target": {
            "type": "ACCOUNT",
            "id": "111111111111"
        }
    },
    "responseElements": {
        "handshake": {
            "requestedTimestamp": "Jan 18, 2017 9:41:16 PM",
            "state": "OPEN",
            "arn": "arn:aws:organizations::111111111111:handshake/o-aa111bb222/invite/
h-examplehandshakeid111",
            "id": "h-examplehandshakeid111",
            "parties": [
                {
                    "type": "ORGANIZATION",
                    "id": "o-aa111bb222"
                },
                {
                    "type": "ACCOUNT",
                    "id": "222222222222"
                }
            ],
            "action": "invite",
            "expirationTimestamp": "Feb 2, 2017 9:41:16 PM",
            "resources": [
                {
                    "resources": [
                        {
                            "type": "MASTER_EMAIL",
                            "value": "diego@example.com"
                        },
                        {
```

```
                            "type": "MASTER_NAME",
                            "value": "Management account for organization"
                        },
                        {
                            "type": "ORGANIZATION_FEATURE_SET",
                            "value": "ALL"
                        }
                    ],
                    "type": "ORGANIZATION",
                    "value": "o-aa111bb222"
                },
                {
                    "type": "ACCOUNT",
                    "value": "222222222222"
                },
                {
                    "type": "NOTES",
                    "value": "This is a request for Mary's account to join Diego's
 organization."
                }
            ]
        }
    },
    "requestID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```

ログエントリの例: AttachPolicy

次の例は、サンプルAttachPolicy呼び出しの CloudTrail ログエントリを示しています。このレスポンスは、リクエストされたポリシータイプが、アタッチが試行された root で有効でないため、呼び出しが失敗したことを示します。

```
{
    "eventVersion": "1.06",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAMVNPBQA3EXAMPLE",
        "arn": "arn:aws:iam::111111111111:user/diego",
        "accountId": "111111111111",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "userName": "diego"
    },
    "eventTime": "2017-01-18T21:42:44Z",
    "eventSource": "organizations.amazonaws.com",
    "eventName": "AttachPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/55.0.2883.95 Safari/537.36",
    "errorCode": "PolicyTypeNotEnabledException",
    "errorMessage": "The given policy type ServiceControlPolicy is not enabled on the
 current view",
    "requestParameters": {
        "policyId": "p-examplepolicyid111",
        "targetId": "ou-examplerootid111-exampleouid111"
    },
    "responseElements": null,
    "requestID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventID": "EXAMPLE8-90ab-cdef-fedc-ba987EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111111111111"
}
```


Amazon EventBridge

AWS Organizations は EventBridge、以前の Amazon CloudWatch Events である Amazon と連携して、管理者が指定したアクションが組織内で発生したときにイベントを発生させることができます。例えば、アクションの重要性のため、ほとんどの管理者は、組織内に誰かが新しいアカウントを作成するたびに、またはメンバーアカウントの管理者が組織を離れようとするたびに警告を受けたいと考えています。これらのアクションを検索し、生成されたイベントを管理者定義のターゲットに送信する EventBridge ルールを設定できます。受信者に E メールまたはテキストメッセージを送信する Amazon SNS トピックをターゲットに指定できます。また、後で確認するためにアクションの詳細をログに記録する AWS Lambda 関数を作成することもできます。

で組織内のキーアクティビティを EventBridge モニタリングする方法を示すチュートリアルについては、「」を参照してくださいチュートリアル: Amazon EventBridge を使用して、組織の重要な変更をモニタリングする。

> **Important**
>
> 現在、 AWS Organizations は米国東部 (バージニア北部) リージョンでのみホストされています (グローバルに利用可能です) 。このチュートリアルのステップを実行するには、そのリージョンを使用する AWS Management Console ように を設定する必要があります。

の設定と有効化の方法など EventBridge、 の詳細については、「Amazon EventBridge ユーザーガイド」を参照してください。

AWS Organizationsのコンプライアンス検証

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

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

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

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

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

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。

- AWS カスタマーコンプライアンスガイド — コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に把握できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- Amazon GuardDuty – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。 GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件への対応に役立ちます。
- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

AWS Organizations での耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。AWS リージョン には、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立· 隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、アベイラビリティーゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョン とアベイラビリティ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。

AWS Organizations でのインフラストラクチャセキュリティ

マネージドサービスである AWS Organizations は AWS グローバルネットワークセキュリティで保護されています。AWSセキュリティサービスと AWS がインフラストラクチャを保護する方法については、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected Framework」の「インフラストラクチャ保護」を参照してください。

Organizations 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。

- Transport Layer Security (TLS) TLS 1.2 および TLS 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です。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時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。

コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140−2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、連邦情報処理規格 (FIPS) 140-2を参照してください。

AWS Organizations リファレンス

このセクションのトピックを使用して、AWS Organizations のさまざまな側面に関する詳細な参考資料を検索します。

トピック

- のクォータ AWS Organizations
- AWS で使用できる マネージドポリシー AWS Organizations

のクォータ AWS Organizations

このセクションでは、 AWS Organizationsに影響を与えるクォータを指定します。

命名ガイドライン

以下は、アカウント、組織単位 (OUs) AWS Organizations、ルート、ポリシーの名前など、 で作成する名前のガイドラインです。

- 名前は Unicode 文字で構成すること
- 名前の最大文字列長は、オブジェクトによって異なります。それぞれの実際の制限を確認するには、AWS Organizations API リファレンスでオブジェクトを作成する API オペレーションを検索してください。そのオペレーションの `Name` パラメータの詳細を確認します。例: アカウント名または OU 名。

最大値および最小値

のエンティティのデフォルトの最大値を次に示します AWS Organizations。

> **Note**
>
> Service Quotas コンソールを使用して、これらの値の一部を引き上げるよう要求できます。Organizations は、米国東部 (バージニア北部) リージョン (`us-east-1`) で物理的にホストされているグローバルサービスです。したがって、Service Quotas コンソール、、 AWS CLIまたは AWS SDK を使用する場合は、 `us-east-1`を使用して Organizations クォータにアクセスする必要があります。

組織 AWS アカウント 内の の数	10 — 組織で許容されるアカウントのデフォルトの最大数。より多く必要な場合は、Service Quotas コンソールを使用して数を増やすようリクエストできます。 アカウントに送信された招待はこのクォータに対してカウントされます。招待されたアカウントが拒否された場合、管理アウントが招待をキャンセルした場合、または招待状の有効期限が切れた場合は、カウントが返されます。 新しく作成されたアカウントや組織では、デフォルトの 10 アカウントを下回るクォータが発生する可能性があります。
組織の root の数	1
組織の OU の数	1,000
組織の各タイプのポリシーの数	AI サービスのオプトアウトポリシー: 1000 バックアップポリシー: 1000 サービスコントロールポリシー: 2000 タグポリシー: 1000
ポリシードキュメントの最大サイズ	AI サービスのオプトアウトポリシー: 2500 文字 バックアップポリシー: 10,000 文字 サービスコントロールポリシー: 5120 文字 タグポリシー: 10,000 文字 注： を使用してポリシーを保存する場合 AWS Management Console、JSON 要素と引用符の外側の間に余分な空白 (スペースや改行など) は削除され、カウントされません。SDK オペレーションまたは を使用してポリシーを保存すると AWS CLI、ポリシーは指定したとおりに保存され、文字の自動削除は行われません。
ルート内の OU 最大ネスト数	ルート以下に 5 レベルの OU 深。

24 時間以内に試行できる招待の最大回数	20 または組織で許可される最大アカウント数のいずれか大きい方です。承諾済みの招待は、このクォータに対してカウントされません。1 つの招待が承諾されるとすぐ、同じ日に別の招待を送信できます。 組織で許可される最大アカウント数が 20 未満の場合、組織に含めることができるアカウント数を超えてアカウントを招待しようとすると、「アカウント制限を超えました」という例外が発生します。ただし、招待をキャンセルして、1 日に最大 20 回まで新しい招待を送信することができます。
同時に作成できるメンバーアカウントの数	5 - 1 つが終了するとすぐに他を開始できますが、一度に進行できるのは 5 つのみです。
30 日間に閉鎖できるメンバーアカウントの数	組織内のメンバーアカウントの 10%、最大 1000 個。 • 100 アカウント未満 — 最大 10 人のメンバーアカウントを閉鎖できます • 100 ~ 10,000 アカウント – メンバーアカウントの最大 10% を閉鎖できます • > 10,000 アカウント – 最大 1,000 のメンバーアカウントを閉鎖できます 例えば、10,500 個のメンバーアカウントがある場合、30 日間に最大 1,000 個の (1,050 個ではない) アカウントを閉鎖できます。この上限に達すると、AWS Billing コンソールで追加のアカウントを閉鎖できます。また、上限数がリセットされるまで待ちます。詳細については、「アカウント管理ガイド」の「アカウントを閉鎖する前に知っておくべきこと」を参照してください。 AWS
同時に閉鎖できるメンバーアカウントの数	3 – 同時に実行できるアカウント閉鎖は 3 つだけです。1 つが終了するとすぐに、別のアカウントを閉鎖できます。
ポリシーをアタッチできるエンティティの数	無制限

ルート、OU、またはアカウントにアタッチできるタグの数	50
リソースベースの委任ポリシーの最大サイズ	40,000 文字

ハンドシェイクの有効期限

でのハンドシェイクのタイムアウトは次のとおりです AWS Organizations。

組織に参加するための招待	15 日間
組織内のすべての機能を有効にするリクエスト	90 日間
ハンドシェイクが削除され、リストに表示されなくなる	ハンドシェイクの完了から 30 日後

1 つのエンティティにアタッチできるポリシーの数

最小値および最大値は、ポリシータイプとポリシーをアタッチするエンティティよって異なります。次の表では、各ポリシータイプおよび各タイプにアタッチできるエンティティの数を示します。

> **Note**
>
> これらの数は、OU またはアカウントに直接アタッチされたポリシーにのみ適用されます。継承によって OU またはアカウントに影響を与えるポリシーは、これらの制限にはカウントされません。すべてのポリシー制限はハード制限です。

ポリシータイプ	エンティティにアタッチされる最小数	ルートにアタッチされる最大値	OU あたりの最大アタッチ数	アカウントあたりの最大アタッチ数
サービスコントロールポリシー	1 - 各エンティティには常に少なくとも 1 つの SCP がアタッチされる必要があります。エンティティから最後の SCP を削除することはできません。	5	5	5
AI サービスのオプトアウトポリシー	0	5	5	5
バックアップポリシー	0	10	10	10
タグポリシー	0	10	10	10

> **Note**
>
> 現在、ルートは組織に 1 つのみ持つことができます。

スロットリングの制限

次の表は、管理カテゴリ別の AWS Organizations APIs を一覧表示し、アカウントレベルと組織レベルでそれぞれのスロットルレートを示しています。

AWS Organizations API	アカウントあたりの制限 (レート、バースト)	組織あたりの制限 (レート、バースト)
アカウント管理		

AWS Organizations API	アカウントあたりの制限 (レート、バースト)	組織あたりの制限 (レート、バースト)
CloseAccount	.05、1	
CreateAccount, CreateGovCloudAccount	0.1、3	
DescribeAccount	20、30	24、36
DescribeCreateAccountStatus	2、2	2、3
LeaveOrganization	1、1	
ListCreateAccountStatus	5、8	6、10
ハンドシェイク管理		
AcceptHandshake, DescribeHandshake	1、1	
CancelHandshake	2、3	
DeclineHandshake	1、3	
InviteAccountToOrganization	3、5	
ListHandshakesForAccount, ListHandshakesForOrganization	5、8	6、10
組織管理		
CreateOrganization, DeleteOrganization, EnableFullControl	1、1	
CreateOrganizationalUnit, DescribeOrganization	1、2	

AWS Organizations API	アカウントあたりの制限 (レート、バースト)	組織あたりの制限 (レート、バースト)
MoveAccount, UpdateOrganizationalUnit, DeleteOrganizationalUnit	2、3	
DescribeOrganizationalUnit	2、2	2、3
ListAccounts	8、12	9、15
ListChildren	6、10	7、12
ListParents, ListAccountsForParent, ListOrganizationalUnitsForParent	5、8	6、10
ListRoots	1、2	1、3
ListTagsForResource	10、15	12、18
RemoveAccountFromOrganization	2、2	
TagResource, UntagResource	4、6	
ポリシー管理		
CreatePolicy, DeletePolicy, AttachPolicy, DetachPolicy	2、3	
DescribePolicy	2、2	2、3
DisablePolicyType, EnablePolicyType	1、1	
ListPolicies, ListPoliciesForTarget, ListTargetsForPolicy	5、8	6、10
UpdatePolicy	2、3	

AWS Organizations API	アカウントあたりの制限 (レート、バースト)	組織あたりの制限 (レート、バースト)
サービス管理		
を有効化AWSServiceAccess、無効化AWSServiceAccess	1、2	
AWSServiceAccessForOrganization、 を一覧表示します。 ListDelegatedServicesForAccount	1、3	1、4
ListDelegatedAdministrators	5、8	6、10
RegisterDelegatedAdministrator, DeregisterDelegatedAdministrator	1、2	

AWS で使用できる マネージドポリシー AWS Organizations

このセクションでは、組織の管理に使用する AWSマネージドポリシーについて説明します。 AWS管理ポリシーを変更または削除することはできませんが、必要に応じて組織内のエンティティにアタッチまたはデタッチできます。

AWS OrganizationsAWS Identity and Access Management (IAM) で使用する マネージドポリシー

IAM管理ポリシーは、 AWSによって提供され、維持されます。管理ポリシーは、管理ポリシーを適切な IAM ユーザーまたはロールオブジェクトにアタッチすることでユーザーに割り当てることができる、一般的なタスクに対するアクセス許可を提供します。ポリシーを自分で記述する必要はありません。また、 が新しいサービスをサポートするために必要に応じてポリシー AWS を更新すると、自動的にすぐに更新のメリットが得られます。 AWS 管理ポリシーのリストは、IAM コンソールの [Policies] (ポリシー) ページで確認できます。[Filter policies] (フィルターポリシー) のドロップダウンを使用して、[AWS managed] (マネージド) を選択します。

以下の管理ポリシーを使用して、組織のユーザーにアクセス許可を付与できます

ポリシー名	説明	ARN
AWSOrganizationsFullAccess	組織を作成し、完全に管理するために必要なすべてのアクセス許可を提供します。次のスニペットは、このポリシーステートメントの内容を示しています。 { "Version": "2012-10-17", "Statement": [{ "Sid": "AWSOrganizationsFullAccess", "Effect": "Allow", "Action": "organizations:*", "Resource": "*" }, { "Sid": "AWSOrganizationsFullAccessAccount", "Effect": "Allow", "Action": ["account:PutAlternateContact", "account:DeleteAlternateContact", "account:GetAlternateContact", "account:GetContactInformation", "account:PutContactInformation", "account:ListRegions", "account:EnableRegion", "account:DisableRegion"],	arn:aws:iam::aws:policy/AWSOrganizationsFullAccess

ポリシー名	説明	ARN

	"Resource": "*"
	},
	{
	"Sid": "AWSOrganizationsFullAccessCreateSLR",
	"Effect": "Allow",
	"Action": "iam:CreateServiceLinkedRole",
	"Resource": "*",
	"Condition": {
	"StringEquals": {
	"iam:AWSServiceName": "organizations.amazonaws.com"
	}
	}
	}
	]
	}	

ポリシー名	説明	ARN

AWSOrganizationsReadOnlyAccess	組織に関する情報への読み取り専用アクセスを提供します。ユーザーがこれに変更を加えることはできません。次のスニペットは、このポリシーステートメントの内容を示しています。
	
	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Sid": "AWSOrganizationsReadOnly",
	"Effect":"Allow",
	"Action":[
	"organizations:Describe*",
	"organizations:List*"
	],
	"Resource": "*"
	},
	{
	"Sid": "AWSOrganizationsReadOnlyAccount",
	"Effect":"Allow",
	"Action":[
	"account:GetAlternateContact",
	"account:GetContactInformation",
	"account:GetPrimaryEmail",
	"account:GetRegionOptStatus",
	"account:ListRegions"
	],
	"Resource": "*"
	}
		arn:aws:iam::aws:policy/AWSOrganizationsReadOnlyAccess

ポリシー名	説明	ARN

	<pre>]
	}</pre>	

Organizations AWS 管理ポリシーの更新

次の表は、このサービスがこれらの変更の追跡を開始してからの AWS マネージドポリシーの更新の詳細を示しています。このページの変更に関する自動通知については、AWS Organizations ドキュメントの履歴ページの RSS フィードをサブスクライブしてください。

変更	説明	日付
AWSOrganizationsReadOnlyAccess – ルートユーザーの E メールアドレスを表示するために必要なアカウント API アクセス許可を許可するように更新されました。	Organizations は、組織内の任意のメンバーアカウントのルートユーザーの E メールアドレスを表示するアクセスを有効にするaccount:GetPrimaryEmail アクションと、組織内の任意のメンバーアカウントの有効なリージョンを表示するアクセスを有効にするaccount:GetRegionOptStatus アクションを追加しました。	2024 年 6 月 6 日
AWSOrganizationsFullAccess – ポリシーステートメントを記述するSid要素を含めるように更新されました。	Organizations が AWSOrganizationsFullAccess マネージドポリシーのSid要素を追加しました。	2024 年 2 月 6 日
AWSOrganizationsReadOnlyAccess – ポリシーステートメントを記述するSid要素を含めるように更新されました。	Organizations は、 AWSOrganizationsReadOnlyAccess マネージドポリシーのSid要素を追加しました。	2024 年 2 月 6 日
AWSOrganizationsFullAccess – Organizations コンソール AWS リージョン で有効または無効にするた	Organizations では、ポリシーに account:ListRegions 、account:EnableRegion 、	2022 年 12 月 22 日

変更	説明	日付
めに必要なアカウント API アクセス許可を許可するように更新されました。	および account:DisableRegion アクションを追加して、アカウントのリージョンを有効または無効にするための書き込みアクセスを有効にしました。	
AWSOrganizationsReadOnlyAccess – Organizations コンソール AWS リージョン を介して一覧表示するために必要なアカウント API アクセス許可を許可するように更新されました。	Organizations では、ポリシーに account:ListRegions アクションを追加して、アカウントのリージョンを表示するためのアクセスを有効にしました。	2022 年 12 月 22 日
AWSOrganizationsFullAccess – Organizations コンソールを介してアカウントの連絡先を追加または編集するために必要なアカウント API アクセス許可を許可するように更新されました。	Organizations では、ポリシーに account:GetContactInformation および account:PutContactInformation アクションを追加して、アカウントの連絡先を変更するための書き込みアクセスを有効にしました。	2022 年 10 月 21 日
AWSOrganizationsReadOnlyAccess – Organizations コンソールを介してアカウントの連絡先を表示するために必要なアカウント API アクセス許可を付与するように更新されました。	Organizations では、ポリシーに account:GetContactInformation アクションを追加して、アカウントの連絡先を表示するためのアクセスを有効にしました。	2022 年 10 月 21 日

変更	説明	日付
AWSOrganizationsFullAccess – 組織の作成を許可するように更新されました。	Organizations では、組織の作成に必要なサービスリンクロールの作成できるようにするために、ポリシーに CreateServiceLinkedRole アクセス許可を追加しました。アクセス許可は、organizations.amazonaws.com のサービスのみで使用できるロールの作成に制限されています。	2022 年 8 月 24 日
AWSOrganizationsFullAccess – Organizations コンソールを介してアカウントの代替連絡先を追加、編集、または削除するために必要なアカウント API アクセス許可を付与するように更新されました。	Organizations では、アカウントの代替連絡先を変更するための書き込みアクセスを有効にする account:GetAlternateContact 、account:DeleteAlternateContact 、および account:PutAlternateContact アクションがポリシーに追加されました。	2022 年 2 月 7 日
AWSOrganizationsReadOnlyAccess – Organizations コンソールを介してアカウントの代替連絡先を表示するために必要なアカウント API アクセス許可を付与するように更新されました。	Organizations では、アカウントの代替連絡先を表示するためのアクセスを有効にする account:GetAlternateContact アクションがポリシーに追加されました。	2022 年 2 月 7 日

AWS Organizations マネージドサービスコントロールポリシー

サービスコントロールポリシー (SCPsは IAM アクセス許可ポリシーに似ていますが、IAM AWS Organizations ではなく の機能です。SCP を使用して、有効化されるエンティティの最大アクセス権限を指定します。SCP は、組織内のルート、組織単位 (OU)、またはアカウントにアタッチできます。自分で作成することも、IAM で定義したポリシーを使用することもできます。Organizations コンソール の [Policies] (ポリシー) ページに、組織内のポリシーのリストが表示されます。

> **Important**
>
> すべての root、OU、アカウントには常に少なくとも 1 つの SCP がアタッチされていなければなりません。

ポリシー名	説明	ARN
フルAWSAccess	メンバーアカウントへの AWS Organizations 管理アカウントアクセスを提供します。	arn:aws:organizations::aws:policy/service_control_policy/p-FullAWSAccess

AWS Organizations のトラブルシューティング

AWS Organizations の操作中に問題が発生した場合は、このセクションのトピックを参照してください。

トピック

一般的な問題のトラブルシューティング

この情報を使用して、アクセス拒否された問題や、AWS Organizations を操作するときに発生する可能性のある他の一般的な問題の診断や修復を行います。

トピック

AWS Organizations にリクエストを送信すると、「アクセス拒否」というメッセージが表示される

- 要求したアクションとリソースを呼び出す権限を持っているかを確認します。管理者は、ユーザー、グループ、ロールに IAM ポリシーをアタッチし、許可を付与する必要があります。ポリシーが時間帯または IP アドレス制限などの条件を含む権限を付与する記述をしている場合は、リクエストを送信する際にそれらの条件を満たす必要もあります。ユーザー、グループ、ロールのポリシーの表示または修正に関する詳細については、「IAM ユーザーガイド」の「ポリシーの使用」を参照してください。
- 手動で API リクエストに署名する (AWS SDK を使用しない) 場合は、正確にリクエストに署名していることを確認します。

一時的なセキュリティ認証情報を使用してリクエストを送信すると「アクセスが拒否されました」というメッセージが表示される

- リクエストの作成に使用している ユーザーまたはロールに適切なアクセス権限があることを確認します。一時的なセキュリティ認証情報のアクセス権限は ユーザーまたはロールから生じるものであるため、 ユーザーまたはロールに付与されたアクセス権限に制限されます。一時的なセキュリティ認証情報のアクセス権限がどのように決定されるかについては、「IAM ユーザーガイド」の「Controlling Permissions for Temporary Security Credentials」を参照してください。
- リクエストが正しく署名されており、そのリクエストの形式が正しいことを確認します。詳細については、選択した SDK の ツールキット ドキュメント、または IAM ユーザーガイドの「AWS リソースへのアクセスをリクエストするための一時的なセキュリティ認証情報の発行」を参照してください。
- 一時的な認証情報が失効していないことを確認します。詳細については、「IAM ユーザーガイド」の「Requesting Temporary Security Credentials」を参照してください。

組織をメンバーアカウントとして残したり、メンバーアカウントを管理アカウントとして削除しようとすると、「アクセスが拒否されました」というメッセージが表示されます。

- メンバーアカウントを削除できるのは、メンバーアカウントでの請求を IAM ユーザーアクセスで有効にした後のみです。詳細については、AWS Billing ユーザーガイドの「Billing and Cost Management コンソールへのアクセスを有効にする」を参照してください。

- アカウントがスタンドアロンアカウントとして動作するために必要な情報を持っている場合にのみ、組織からアカウントを削除できます。AWS Organizations コンソール、API、AWS CLI コマンドを使用して組織内にアカウントを作成した場合、その情報が自動的に収集されるわけではありません。スタンドアロンとして使用する各アカウントについて、まず AWS カスタマーアグリーメントに同意してサポートプランを選択し、必須の連絡先情報を入力および確認して、現在のお支払い方法を入力する必要があります。この支払い方法は、アカウントが組織に関連付けられていない間に発生する AWS の課金対象 (AWS 無料利用枠外) のアクティビティに対して課金するために AWS によって使用されます。詳細については、「メンバーアカウントから組織を退会する」を参照してください。

組織にアカウントを追加しようとすると「クォータを超えました」というメッセージが表示される

組織内で保持できるアカウントの数には上限があります。削除したアカウントや閉じたアカウントは、引き続きこのクォータに対してカウントされます。

参加の招待は、組織内のアカウントの上限数に対してカウントされます。招待されたアカウントが拒否された場合、管理アウントが招待をキャンセルした場合、または招待状の有効期限が切れた場合は、カウントが返されます。

- AWS アカウント を閉じたり、削除したりする前に、そのアカウントを組織から除外して、以後クォータにカウントされないようにしてください。
- クォータ引き上げのリクエストの詳細については、「最大値および最小値」を参照してください。

アカウントを追加または削除するときに「このオペレーションでは待機期間が必要です」というメッセージが表示される

一部のアクションでは待機期間が必要です。たとえば、作成したばかりのアカウントをすぐに削除することはできません。数日後にアクションを再試行してください。アカウントを追加または削除するときにアカウントのクォータに関する問題が発生した場合は、「最大値および最小値」でクォータの引き上げをリクエストする方法について確認してください。

組織にアカウントを追加しようとすると「組織がまだ初期化中です」というメッセージが表示される

このエラーが表示され、組織を作成してから 1 時間以上が経過している場合は、AWS Support までお問い合わせください。

組織にアカウントを招待しようとすると、「招待は無効になっています」というメッセージが表示される。

これは、組織内のすべての機能を有効にする場合に発生します。このオペレーションには時間がかかり、すべてのメンバーアカウントが応答する必要があります。オペレーションが完了するまで、新しいアカウントを組織サイトに加入するよう招待することはできません。

変更がすぐに表示されない

世界中のデータセンター内のコンピューターを介してアクセスされるサービスとして、AWS Organizations は、結果整合性と呼ばれる分散コンピューティングモデルを採用しています。AWS Organizations で行った変更は、すべての可能なエンドポイントから認識されるまでに時間がかかります。この遅延は、サーバー間、レプリケーションゾーン間、世界中のリージョン間でのデータ送信にかかる時間から発生している場合もあります。AWS Organizations ではパフォーマンス向上のためにキャッシュも使用しているため、これが原因で遅延が発生することがあります。変更は、以前にキャッシュされたデータがタイムアウトになるまで反映されない場合があります。

発生する可能性のあるこれらの遅延を考慮して、グローバルなアプリケーションを設計します。ある場所で行われた変更が別の場所にすぐに表示されない場合でも、適切に動作することを確認します。

AWS の他のいくつかのサービスがこの遅延からどのような影響を受けるかの詳細については、以下のリソースを参照してください。

- Amazon Redshift データベースデベロッパーガイドの「データの整合性の管理」
- 「Amazon Simple Storage Service ユーザーガイド」の「Amazon S3 Data Consistency Model」
- AWS Big Data Blog の「ETL ワークフローに Amazon S3 および Amazon Elastic MapReduce を使用する場合の整合性の確保」
- Amazon EC2 API リファレンスの「EC2 の結果整合性」

AWS Organizations ポリシーのトラブルシューティング

AWS Organizations ポリシーで検出された共通のエラーを診断して変更するために、この情報を使用します。

サービスコントロールポリシー

のサービスコントロールポリシー (SCPs AWS Organizations は IAM ポリシーと似ており、共通の構文を共有します。この構文は Object JavaScript Notation (JSON) のルールで始まります。JSON では、オブジェクトおよびオブジェクトを構成する名前と値のペアを指定します。IAM ポリシーの文法は、アクセス許可を付与するためにポリシーを使用する AWS のサービスに対して名前と値がどのような意味を持ち、どのように解釈されるかを定義するものです。

AWS Organizations は、IAM 構文と文法のサブセットを使用します。詳細については、「SCP 構文」を参照してください。

一般的なポリシーエラー

- 複数のポリシーオブジェクト
- 複数の Statement 要素
- ポリシードキュメントが最大サイズを超えています

複数のポリシーオブジェクト

SCP は、1 つの JSON オブジェクトのみで構成する必要があります。オブジェクトは括弧 ({ }) で囲んで示します。外側の { } のペア内に追加の { } を埋め込むことによって JSON オブジェクト内で他のオブジェクトをネストすることができますが、ポリシーに含めることができるのは一番外側の { } のペアのみです。次の例は、最上位に 2 つのオブジェクト (#で示した箇所) が含まれているため、誤りです。

```
{
  "Version": "2012-10-17",
  "Statement":
  {
     "Effect":"Allow",
     "Action":"ec2:Describe*",
     "Resource":"*"
  }
}
```

```
{
  "Statement": {
     "Effect": "Deny",
     "Action": "s3:*",
     "Resource": "*"
  }
}
```

ただし、正しいポリシーの文法を使用して、前述の例の目的を果たすことができます。それぞれに独自の Statement 要素を含む 2 つのポリシーオブジェクトを含める代わりに、1 つの Statement 要素に 2 つのブロックを組み合わせて使用することができます。Statement 要素には、次の例に示すように 2 つのオブジェクトの配列を値として指定します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource":" *"
    },
    {
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*"
    }
  ]
}
```

この例では、2 つの要素による効果は異なるため、1 つの要素を含む Statement に圧縮することはできません。構文は、各構文の Effect や、Resource 要素が同一の場合にのみ、組み合わせることができます。

複数の Statement 要素

このエラーは、一見、前のセクションのエラーのバリエーションのように見えますが、構文上はエラーの種類が異なります。次の例では、最上位の 1 ペアの { } で示された 1 つのポリシーオブジェクトのみが存在します。そのオブジェクト内に 2 つの Statement 要素が含まれています。

SCP には、名前 (Statement) の後にコロン、その後に値という形式で構成される 1 つの Statement 要素のみを指定する必要があります。Statement 要素の値は、{} で示され、1 つの

Effect 要素、1 つの Action 要素、および 1 つの Resource 要素を含むオブジェクトである必要があります。次の例は、ポリシーオブジェクトに 2 つの Statement 要素が含まれているため、誤りです。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "ec2:Describe*",
    "Resource": "*"
  },
  "Statement": {
    "Effect": "Deny",
    "Action": "s3:*",
    "Resource": "*"
  }
}
```

値オブジェクトは複数の値オブジェクトの配列にすることができるため、次の例に示すように、2 つの Statement 要素を組み合わせて 1 つの要素にすることで、この問題を解決できます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe*",
      "Resource":"*"
    },
    {
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": "*"
    }
 ]
}
```

Statement 要素の値はオブジェクト配列です。例の配列は 2 つのオブジェクトで構成され、各オブジェクトに Statement 要素の正しい値が指定されています。配列内の各オブジェクトはカンマで区切ります。

ポリシードキュメントが最大サイズを超えています

SCP ドキュメントの最大サイズは 5,120 文字です。この最大サイズには、空白を含むすべての文字が含まれます。SCP のサイズを減らすには、引用符の外側にあるすべての空白文字 (スペースや改行など) を削除できます。

> **Note**
>
> を使用してポリシーを保存する場合 AWS Management Console、JSON 要素と引用符の外側の間の余分な空白は削除され、カウントされません。SDK オペレーションまたは を使用してポリシーを保存すると AWS CLI、ポリシーは指定したとおりに保存され、文字の自動削除は行われません。

HTTP クエリリクエストを作成して API を呼び出す

このセクションは、AWS Organizations に対して Query API を使用する場合についての一般的な情報を提供します。API オペレーションとエラーの詳細については、「AWS Organizations API リファレンス」を参照してください。

> **Note**
>
> AWS Organizations Query API を直接呼び出す代わりに、いずれかの AWS SDK を使用することができます。AWS SDK は、さまざまなプログラム言語およびプラットフォームのライブラリやサンプルコード (Java、Ruby、.NET、iOS、Android など) から成ります。SDK は、AWS Organizations や AWS へのプログラムによるアクセス許可を作成するのに役立ちます。例えば、SDK は要求への暗号を使用した署名、エラーの管理、要求の自動的な再試行などのタスクを処理します。AWS SDK のダウンロードやインストールなどの詳細については、「Amazon Web Services のツール」を参照してください。

AWS Organizations 用 Query API により、サービスアクションを呼び出すことができます。Query API リクエストは、HTTPS リクエストであり、実行するべき操作を示す `Action` パラメータを含める必要があります。AWS Organizations は、すべての操作の GET リクエストおよび POST リクエストをサポートしています。つまり、この API では、あるアクションに対しては GET を、他のアクションに対しては POST をといった使い分けを必要としません。しかしながら、GET リクエストは URL のサイズに制限があります。この制限はブラウザによって異なり、通常は 2048 バイトです。したがって、大きなサイズを必要とする Query API リクエストにおいては、POST リクエストを使用する必要があります。

レスポンスは XML 文書です。レスポンスの詳細については、「AWS Organizations API リファレンス」の個別のアクションページを参照してください。

トピック

- エンドポイント
- HTTPS の必要性
- AWS Organizations API リクエストの署名

エンドポイント

AWS Organizations には、米国東部 (バージニア北部) リージョンでホストされる単一のグローバル API エンドポイントがあります。

すべてのサービスのAWSエンドポイントとリージョンの詳細については、「」の「リージョンエンドポイント」を参照してくださいAWS 全般のリファレンス。

HTTPS の必要性

Query API は、セキュリティ認証情報などの機密情報を返すため、必ず HTTPS を使用してすべての API リクエストを暗号化する必要があります。

AWS Organizations API リクエストの署名

リクエストには、アクセスキー ID およびシークレットアクセスキーによる署名が必要です。AWS Organizations での日々の作業には、AWS アカウントのルートユーザー 認証情報を使用しないことを強くお勧めします。ユーザーまたはロールに認証情報を使用できます。

API リクエストに署名するには、AWS 署名バージョン 4 を使用する必要があります。Signature Version 4 の詳細については、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。

AWS Organizations では、署名バージョン 2 などの以前のバージョンをサポートしていません。

詳細については、次を参照してください。

- AWS セキュリティ認証情報 - AWS へのアクセスに使用できる認証情報の種類に関する一般的な情報を提供します。
- IAM でのセキュリティのベストプラクティス - AWS Organizations に含まれるリソースなど、AWS リソースの保護に役立つ IAM サービスの使用に関するアドバイスを提供します。
- IAM での一時的なセキュリティ認証情報 - 一時的なセキュリティ認証情報の作成方法と使用方法を説明します。

AWS SDKs を使用する Organizations のコード例

次のコード例は、 AWS Software Development Kit (SDK) で Organizations を使用する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、関連するシナリオやサービス間の例ではアクションのコンテキストが確認できます。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

コードの例

- AWS SDKsアクション
  - AWS SDK または CLI AttachPolicyで を使用する
  - AWS SDK または CLI CreateAccountで を使用する
  - AWS SDK または CLI CreateOrganizationで を使用する
  - AWS SDK または CLI CreateOrganizationalUnitで を使用する
  - AWS SDK または CLI CreatePolicyで を使用する
  - AWS SDK または CLI DeleteOrganizationで を使用する
  - AWS SDK または CLI DeleteOrganizationalUnitで を使用する
  - AWS SDK または CLI DeletePolicyで を使用する
  - AWS SDK または CLI DescribePolicyで を使用する
  - AWS SDK または CLI DetachPolicyで を使用する
  - AWS SDK または CLI ListAccountsで を使用する
  - AWS SDK または CLI ListOrganizationalUnitsForParentで を使用する
  - AWS SDK または CLI ListPoliciesで を使用する

AWS SDKsアクション

次のコード例は、 AWS SDKs を使用して個々の Organizations アクションを実行する方法を示しています。これらの抜粋は Organizations API を呼び出し、コンテキスト内で実行する必要がある大規

模なプログラムからのコードの抜粋です。各例には へのリンクが含まれており GitHub、コードの設定と実行の手順を確認できます。

以下の例には、最も一般的に使用されるアクションのみ含まれています。詳細な一覧については、「AWS Organizations API リファレンス」を参照してください。

例

- AWS SDK または CLI AttachPolicyで を使用する
- AWS SDK または CLI CreateAccountで を使用する
- AWS SDK または CLI CreateOrganizationで を使用する
- AWS SDK または CLI CreateOrganizationalUnitで を使用する
- AWS SDK または CLI CreatePolicyで を使用する
- AWS SDK または CLI DeleteOrganizationで を使用する
- AWS SDK または CLI DeleteOrganizationalUnitで を使用する
- AWS SDK または CLI DeletePolicyで を使用する
- AWS SDK または CLI DescribePolicyで を使用する
- AWS SDK または CLI DetachPolicyで を使用する
- AWS SDK または CLI ListAccountsで を使用する
- AWS SDK または CLI ListOrganizationalUnitsForParentで を使用する
- AWS SDK または CLI ListPoliciesで を使用する

AWS SDK または CLI **AttachPolicy**で を使用する

以下のコード例は、`AttachPolicy` の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Shows how to attach an AWS Organizations policy to an organization,
    /// an organizational unit, or an account.
    /// </summary>
    public class AttachPolicy
    {
        /// <summary>
        /// Initializes the Organizations client object and then calls the
        /// AttachPolicyAsync method to attach the policy to the root
        /// organization.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();
            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new AttachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.AttachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully attached Policy ID {policyId} to
Target ID: {targetId}.");
            }
            else
            {
                Console.WriteLine("Was not successful in attaching the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスAttachPolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントにポリシーをアタッチするには

例 1

次の例は、サービスコントロールポリシーを OU にアタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

例 2

次の例は、サービスコントロールポリシーをアカウントに直接アタッチする方法を示しています。

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- API の詳細については、「 コマンドリファレンスAttachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def attach_policy(policy_id, target_id, orgs_client):
```

```
    """
    Attaches a policy to a target. The target is an organization root, account,
or
    organizational unit.

    :param policy_id: The ID of the policy to attach.
    :param target_id: The ID of the resources to attach the policy to.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.attach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Attached policy %s to target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't attach policy %s to target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 AttachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **CreateAccount**で を使用する

以下のコード例は、CreateAccount の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Creates a new AWS Organizations account.
    /// </summary>
    public class CreateAccount
    {
        /// <summary>
        /// Initializes an Organizations client object and uses it to create
        /// the new account with the name specified in accountName.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();
            var accountName = "ExampleAccount";
            var email = "someone@example.com";

            var request = new CreateAccountRequest
            {
                AccountName = accountName,
                Email = email,
            };

            var response = await client.CreateAccountAsync(request);
            var status = response.CreateAccountStatus;

            Console.WriteLine($"The staus of {status.AccountName} is
{status.State}.");
        }
    }
```

- API の詳細については、「 API リファレンスCreateAccount」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

自動的に組織の一部となるメンバーアカウントを作成するには

次の例は、組織のメンバーアカウントを作成する方法を示しています。メンバーアカウントは、「プロダクションアカウント」という名前と E メールアドレス (susan@example.com) で構成されます。roleName パラメータが指定されていない OrganizationAccountAccessRole ため、Organizations は のデフォルト名を使用して IAM ロールを自動的に作成します。roleName また、アカウントの請求データにアクセスするための十分なアクセス許可を持つ IAM ユーザーまたはロールを許可する設定は、 IamUserAccessToBilling パラメータが指定されていないため、デフォルト値の ALLOW に設定されます。Organizations は、スーザンに「ようこそ」という AWS E メールを自動的に送信します。

```
aws organizations create-account --email susan@example.com --account-name
 "Production Account"
```

出力には、ステータスが現在の IN_PROGRESS 状態であることを示すリクエストオブジェクトが含まれます。

```
{
        "CreateAccountStatus": {
                "State": "IN_PROGRESS",
                "Id": "car-examplecreateaccountrequestid111"
        }
}
```

後で、 create-account-request-id パラメータの値として describe-create-account-status コマンドに Id レスポンス値を指定することで、リクエストの現在のステータスをクエリできます。

詳細については、「 Organizations ユーザーガイド AWS 」の「組織でのアカウントの作成」を参照してください。 AWS

- API の詳細については、「 コマンドリファレンスCreateAccount」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **CreateOrganization**で を使用する

以下のコード例は、CreateOrganization の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;

    /// <summary>
    /// Creates an organization in AWS Organizations.
    /// </summary>
    public class CreateOrganization
    {
        /// <summary>
        /// Creates an Organizations client object and then uses it to create
        /// a new organization with the default user as the administrator, and
        /// then displays information about the new organization.
        /// </summary>
        public static async Task Main()
        {
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var response = await client.CreateOrganizationAsync(new
CreateOrganizationRequest
            {
                FeatureSet = "ALL",
            });
```

```
            Organization newOrg = response.Organization;

            Console.WriteLine($"Organization: {newOrg.Id} Main Accoount:
 {newOrg.MasterAccountId}");
        }
    }
```

- API の詳細については、「 API リファレンスCreateOrganization」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

例 1: 新しい組織を作成するには

Bill は、アカウント 111111111111 の認証情報を使用して組織を作成したいと考えています。次の例は、このアカウントが新しい組織のマスターアカウントになることを示しています。Bill は機能セットを指定していないため、新しい組織ではデフォルトですべての機能が有効になり、サービスコントロールポリシーがルート上で有効になります。

```
aws organizations create-organization
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
        "Organization": {
                "AvailablePolicyTypes": [
                        {
                                "Status": "ENABLED",
                                "Type": "SERVICE_CONTROL_POLICY"
                        }
                ],
                "MasterAccountId": "111111111111",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/
o-exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "FeatureSet": "ALL",
```

```
                "Id": "o-exampleorgid",
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid"
        }
}
```

例 2: 一括決済機能のみを有効にした新しい組織を作成するには

次の例では、一括決済機能のみをサポートする組織を作成します。

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

出力には、新しい組織に関する詳細を含む組織オブジェクトが含まれます。

```
{
        "Organization": {
                "Arn": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
                "AvailablePolicyTypes": [],
                "Id": "o-exampleorgid",
                "MasterAccountArn": "arn:aws:organizations::111111111111:account/
o-exampleorgid/111111111111",
                "MasterAccountEmail": "bill@example.com",
                "MasterAccountId": "111111111111",
                "FeatureSet": "CONSOLIDATED_BILLING"
        }
}
```

詳細については、「AWS Organizations ユーザーガイド」の「Creating an Organization」を参照してください。

- API の詳細については、「 コマンドリファレンスCreateOrganization」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **CreateOrganizationalUnit**で を使用する

以下のコード例は、CreateOrganizationalUnit の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates a new organizational unit in AWS Organizations.
/// </summary>
public class CreateOrganizationalUnit
{
    /// <summary>
    /// Initializes an Organizations client object and then uses it to call
    /// the CreateOrganizationalUnit method. If the call succeeds, it
    /// displays information about the new organizational unit.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var orgUnitName = "ProductDevelopmentUnit";

        var request = new CreateOrganizationalUnitRequest
        {
            Name = orgUnitName,
            ParentId = "r-0000",
        };

        var response = await client.CreateOrganizationalUnitAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
```

```
                Console.WriteLine($"Successfully created organizational unit:
 {orgUnitName}.");
                Console.WriteLine($"Organizational unit {orgUnitName} Details");
                Console.WriteLine($"ARN: {response.OrganizationalUnit.Arn} Id:
 {response.OrganizationalUnit.Id}");
            }
            else
            {
                Console.WriteLine("Could not create new organizational unit.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ルート OU または親 OU に OU を作成するには

次の例は、AccountingOU という名前の OU を作成する方法を示しています。

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

出力には、新しい OU に関する詳細を含む organizationalUnit オブジェクトが含まれます。

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- API の詳細については、「 コマンドリファレンスCreateOrganizationalUnit」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **CreatePolicy**で を使用する

以下のコード例は、CreatePolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Creates a new AWS Organizations Policy.
/// </summary>
public class CreatePolicy
{
    /// <summary>
    /// Initializes the AWS Organizations client object, uses it to
    /// create a new Organizations Policy, and then displays information
    /// about the newly created Policy.
    /// </summary>
    public static async Task Main()
    {
        IAmazonOrganizations client = new AmazonOrganizationsClient();
        var policyContent = "{" +
            "   \"Version\": \"2012-10-17\"," +
            " \"Statement\" : [{" +
                " \"Action\" : [\"s3:*\"]," +
                " \"Effect\" : \"Allow\"," +
                " \"Resource\" : \"*\"" +
```

```
                "}]" +
            "}";

            try
            {
                var response = await client.CreatePolicyAsync(new
CreatePolicyRequest
                {
                    Content = policyContent,
                    Description = "Enables admins of attached accounts to
delegate all Amazon S3 permissions",
                    Name = "AllowAllS3Actions",
                    Type = "SERVICE_CONTROL_POLICY",
                });

                Policy policy = response.Policy;
                Console.WriteLine($"{policy.PolicySummary.Name} has the following
content: {policy.Content}");
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }
    }
```

- API の詳細については、「 API リファレンスCreatePolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

例 1: JSON ポリシーのテキストソースファイルを使用してポリシーを作成するには

次の例は、AllowAllS3Actions という名前のサービスコントロールポリシーを作成する方法を示しています。ポリシーの内容は、policy.json というローカルコンピューター上のファイルから取得されます。

```
aws organizations create-policy --content file://policy.json --name
 AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

出力には、新しいポリシーの詳細を含むポリシーオブジェクトが含まれます。

```
{
        "Policy": {
                "Content": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*\"]}]}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::o-exampleorgid:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Allows delegation of all S3 actions",
                        "Name": "AllowAllS3Actions",
                        "Type":"SERVICE_CONTROL_POLICY"
                }
        }
}
```

例 2: JSON ポリシーをパラメータとしてポリシーを作成するには

次の例は、ポリシーの内容を JSON 文字列としてパラメータに埋め込むことで、同じ SCP を作成する方法を示しています。文字列は、パラメータ内でリテラルとして扱われるように、二重引用符の前にバックスラッシュを付けて エスケープする必要があります。パラメータ自体も二重引用符で囲みます。

```
aws organizations create-policy --content "{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"*
\"]}]}" --name AllowAllS3Actions --type SERVICE_CONTROL_POLICY --description
 "Allows delegation of all S3 actions"
```

Organizations でのポリシーの作成と使用の詳細については、「AWS Organizations ユーザーガイド」の「AWS Organizations のポリシーの管理」を参照してください。

- API の詳細については、「 コマンドリファレンスCreatePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def create_policy(name, description, content, policy_type, orgs_client):
    """
    Creates a policy.

    :param name: The name of the policy.
    :param description: The description of the policy.
    :param content: The policy content as a dict. This is converted to JSON
 before
                    it is sent to AWS. The specific format depends on the policy
 type.
    :param policy_type: The type of the policy.
    :param orgs_client: The Boto3 Organizations client.
    :return: The newly created policy.
    """
    try:
        response = orgs_client.create_policy(
            Name=name,
            Description=description,
            Content=json.dumps(content),
            Type=policy_type,
        )
        policy = response["Policy"]
        logger.info("Created policy %s.", name)
    except ClientError:
        logger.exception("Couldn't create policy %s.", name)
        raise
    else:
        return policy
```

- API の詳細については、 CreatePolicy AWS SDK for Python (Boto3) API リファレンス の「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださ いAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **DeleteOrganization**で を使用する

以下のコード例は、DeleteOrganization の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to delete an existing organization using the AWS
/// Organizations Service.
/// </summary>
public class DeleteOrganization
{
    /// <summary>
    /// Initializes the Organizations client and then calls
    /// DeleteOrganizationAsync to delete the organization.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();
```

```
            var response = await client.DeleteOrganizationAsync(new
DeleteOrganizationRequest());

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine("Successfully deleted organization.");
            }
            else
            {
                Console.WriteLine("Could not delete organization.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeleteOrganization」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

組織を削除するには

次の例は、組織を削除する方法を示しています。この操作を実行するには、組織のマスターアカウントの管理者である必要があります。この例では、組織からメンバーアカウント、OU、ポリシーをすべて削除済みであることを前提としています。

```
aws organizations delete-organization
```

- API の詳細については、「 コマンドリファレンスDeleteOrganization」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **DeleteOrganizationalUnit**で を使用する

以下のコード例は、DeleteOrganizationalUnit の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to delete an existing AWS Organizations organizational unit.
/// </summary>
public class DeleteOrganizationalUnit
{
    /// <summary>
    /// Initializes the Organizations client object and calls
    /// DeleteOrganizationalUnitAsync to delete the organizational unit
    /// with the selected ID.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var orgUnitId = "ou-0000-00000000";

        var request = new DeleteOrganizationalUnitRequest
        {
            OrganizationalUnitId = orgUnitId,
        };

        var response = await client.DeleteOrganizationalUnitAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
```

```
                Console.WriteLine($"Successfully deleted the organizational unit
with ID: {orgUnitId}.");
            }
            else
            {
                Console.WriteLine($"Could not delete the organizational unit with
ID: {orgUnitId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeleteOrganizationalUnit」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

OU を削除するには

次の例は、OU を削除する方法を示しています。この例では、OU からすべてのアカウントと他の OU を削除済みであることを前提としています。

```
aws organizations delete-organizational-unit --organizational-unit-id ou-
examplerootid111-exampleouid111
```

- API の詳細については、「 コマンドリファレンスDeleteOrganizationalUnit」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **DeletePolicy**で を使用する

以下のコード例は、DeletePolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Deletes an existing AWS Organizations policy.
/// </summary>
public class DeletePolicy
{
    /// <summary>
    /// Initializes the Organizations client object and then uses it to
    /// delete the policy with the specified policyId.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var policyId = "p-00000000";

        var request = new DeletePolicyRequest
        {
            PolicyId = policyId,
        };

        var response = await client.DeletePolicyAsync(request);

        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            Console.WriteLine($"Successfully deleted Policy: {policyId}.");
        }
```

```
            else
            {
                Console.WriteLine($"Could not delete Policy: {policyId}.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDeletePolicy」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

ポリシーを削除するには

次の例は、組織からポリシーを削除する方法を示しています。この例では、ポリシーをすべてのエンティティから事前にデタッチしたことを前提としています。

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDeletePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def delete_policy(policy_id, orgs_client):
    """
    Deletes a policy.
```

```
    :param policy_id: The ID of the policy to delete.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.delete_policy(PolicyId=policy_id)
        logger.info("Deleted policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't delete policy %s.", policy_id)
        raise
```

- API の詳細については、 DeletePolicy AWS SDK for Python (Boto3) API リファレンス の「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださ いAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **DescribePolicy**で を使用する

以下のコード例は、DescribePolicy の使用方法を示しています。

CLI

AWS CLI

ポリシーに関する情報を取得するには

次の例は、ポリシーに関する情報をリクエストする方法を示しています。

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

出力には、ポリシーの詳細を含むポリシーオブジェクトが含まれます。

```
{
        "Policy": {
                "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement
\": [\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n
 \"Resource\": \"*\"\n    }\n  ]\n}",
```

```
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::111111111111:policy/o-
exampleorgid/service_control_policy/p-examplepolicyid111",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-examplepolicyid111",
                        "AwsManaged": false,
                        "Name": "AllowAllS3Actions",
                        "Description": "Enables admins to delegate S3
 permissions"
                }
        }
}
```

- API の詳細については、「 コマンドリファレンスDescribePolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def describe_policy(policy_id, orgs_client):
    """
    Describes a policy.

    :param policy_id: The ID of the policy to describe.
    :param orgs_client: The Boto3 Organizations client.
    :return: The description of the policy.
    """
    try:
        response = orgs_client.describe_policy(PolicyId=policy_id)
        policy = response["Policy"]
        logger.info("Got policy %s.", policy_id)
    except ClientError:
        logger.exception("Couldn't get policy %s.", policy_id)
        raise
```

```
    else:
        return policy
```

- API の詳細については、 DescribePolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **DetachPolicy**で を使用する

以下のコード例は、DetachPolicy の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to detach a policy from an AWS Organizations organization,
/// organizational unit, or account.
/// </summary>
public class DetachPolicy
{
    /// <summary>
    /// Initializes the Organizations client object and uses it to call
```

```
        /// DetachPolicyAsync to detach the policy.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var policyId = "p-00000000";
            var targetId = "r-0000";

            var request = new DetachPolicyRequest
            {
                PolicyId = policyId,
                TargetId = targetId,
            };

            var response = await client.DetachPolicyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Successfully detached policy with Policy Id:
{policyId}.");
            }
            else
            {
                Console.WriteLine("Could not detach the policy.");
            }
        }
    }
```

- API の詳細については、「 API リファレンスDetachPolicy」の「」を参照してください。AWS SDK for .NET

CLI

AWS CLI

root、OU、またはアカウントからポリシーをデタッチするには

次のコード例は、OU からポリシーをデタッチする方法を示しています。

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111
 --policy-id p-examplepolicyid111
```

- API の詳細については、「 コマンドリファレンスDetachPolicy」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def detach_policy(policy_id, target_id, orgs_client):
    """
    Detaches a policy from a target.

    :param policy_id: The ID of the policy to detach.
    :param target_id: The ID of the resource where the policy is currently
 attached.
    :param orgs_client: The Boto3 Organizations client.
    """
    try:
        orgs_client.detach_policy(PolicyId=policy_id, TargetId=target_id)
        logger.info("Detached policy %s from target %s.", policy_id, target_id)
    except ClientError:
        logger.exception(
            "Couldn't detach policy %s from target %s.", policy_id, target_id
        )
        raise
```

- API の詳細については、 DetachPolicy AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **ListAccounts**で を使用する

以下のコード例は、ListAccounts の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Uses the AWS Organizations service to list the accounts associated
/// with the default account.
/// </summary>
public class ListAccounts
{
    /// <summary>
    /// Creates the Organizations client and then calls its
    /// ListAccountsAsync method.
    /// </summary>
    public static async Task Main()
    {
        // Create the client object using the default account.
        IAmazonOrganizations client = new AmazonOrganizationsClient();

        var request = new ListAccountsRequest
        {
            MaxResults = 5,
        };
```

```
            var response = new ListAccountsResponse();
            try
            {
                do
                {
                    response = await client.ListAccountsAsync(request);
                    response.Accounts.ForEach(a => DisplayAccounts(a));
                    if (response.NextToken is not null)
                    {
                        request.NextToken = response.NextToken;
                    }
                }
                while (response.NextToken is not null);
            }
            catch (AWSOrganizationsNotInUseException ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        /// <summary>
        /// Displays information about an Organizations account.
        /// </summary>
        /// <param name="account">An Organizations account for which to display
        /// information on the console.</param>
        private static void DisplayAccounts(Account account)
        {
            string accountInfo = $"{account.Id}
{account.Name}\t{account.Status}";

            Console.WriteLine(accountInfo);
        }
    }
```

- API の詳細については、「 API リファレンスListAccounts」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

組織内のすべてのアカウントのリストを取得するには

次の例は、組織内のアカウントのリストをリクエストする方法を示しています。

```
aws organizations list-accounts
```

出力には、アカウントサマリーオブジェクトのリストが含まれます。

```
{
        "Accounts": [
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/111111111111",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481830215.45,
                        "Id": "111111111111",
                        "Name": "Master Account",
                        "Email": "bill@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/222222222222",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835741.044,
                        "Id": "222222222222",
                        "Name": "Production Account",
                        "Email": "alice@example.com",
                        "Status": "ACTIVE"
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/333333333333",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835795.536,
                        "Id": "333333333333",
                        "Name": "Development Account",
                        "Email": "juan@example.com",
                        "Status": "ACTIVE"
```

```
                },
                {
                        "Arn": "arn:aws:organizations::111111111111:account/o-
exampleorgid/444444444444",
                        "JoinedMethod": "INVITED",
                        "JoinedTimestamp": 1481835812.143,
                        "Id": "444444444444",
                        "Name": "Test Account",
                        "Email": "anika@example.com",
                        "Status": "ACTIVE"
                }
        ]
}
```

- API の詳細については、「 コマンドリファレンスListAccounts」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **ListOrganizationalUnitsForParent**で を使用する

以下のコード例は、ListOrganizationalUnitsForParent の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.Organizations;
    using Amazon.Organizations.Model;
```

```
    /// <summary>
    /// Lists the AWS Organizations organizational units that belong to an
    /// organization.
    /// </summary>
    public class ListOrganizationalUnitsForParent
    {
        /// <summary>
        /// Initializes the Organizations client object and then uses it to
        /// call the ListOrganizationalUnitsForParentAsync method to retrieve
        /// the list of organizational units.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            var parentId = "r-0000";

            var request = new ListOrganizationalUnitsForParentRequest
            {
                ParentId = parentId,
                MaxResults = 5,
            };

            var response = new ListOrganizationalUnitsForParentResponse();
            try
            {
                do
                {
                    response = await
client.ListOrganizationalUnitsForParentAsync(request);
                    response.OrganizationalUnits.ForEach(u =>
DisplayOrganizationalUnit(u));
                    if (response.NextToken is not null)
                    {
                        request.NextToken = response.NextToken;
                    }
                }
                while (response.NextToken is not null);
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
```

```
            }
        }

        /// <summary>
        /// Displays information about an Organizations organizational unit.
        /// </summary>
        /// <param name="unit">The OrganizationalUnit for which to display
        /// information.</param>
        public static void DisplayOrganizationalUnit(OrganizationalUnit unit)
        {
            string accountInfo = $"{unit.Id} {unit.Name}\t{unit.Arn}";

            Console.WriteLine(accountInfo);
        }
    }
```

- API の詳細については、「 API リファレンスListOrganizationalUnitsForParent」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

親 OUs またはルート内の OU のリストを取得するには

次の例は、指定したルートの OUsのリストを取得する方法を示しています。

```
aws organizations list-organizational-units-for-parent --parent-id r-
examplerootid111
```

出力は、指定されたルートに 2 つの OUsを示し、それぞれの詳細を示します。

```
{
        "OrganizationalUnits": [
                {
                        "Name": "AccountingDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid111"
                },
                {
```

```
                        "Name": "ProductionDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid222"
                }
        ]
}
```

- API の詳細については、「 コマンドリファレンスListOrganizationalUnitsForParent」の「」を参照してください。 AWS CLI

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK または CLI **ListPolicies**で を使用する

以下のコード例は、ListPolicies の使用方法を示しています。

.NET

AWS SDK for .NET

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
using System;
using System.Threading.Tasks;
using Amazon.Organizations;
using Amazon.Organizations.Model;

/// <summary>
/// Shows how to list the AWS Organizations policies associated with an
/// organization.
/// </summary>
public class ListPolicies
{
    /// <summary>
    /// Initializes an Organizations client object, and then calls its
```

```
        /// ListPoliciesAsync method.
        /// </summary>
        public static async Task Main()
        {
            // Create the client object using the default account.
            IAmazonOrganizations client = new AmazonOrganizationsClient();

            // The value for the Filter parameter is required and must must be
            // one of the following:
            //     AISERVICES_OPT_OUT_POLICY
            //     BACKUP_POLICY
            //     SERVICE_CONTROL_POLICY
            //     TAG_POLICY
            var request = new ListPoliciesRequest
            {
                Filter = "SERVICE_CONTROL_POLICY",
                MaxResults = 5,
            };

            var response = new ListPoliciesResponse();
            try
            {
                do
                {
                    response = await client.ListPoliciesAsync(request);
                    response.Policies.ForEach(p => DisplayPolicies(p));
                    if (response.NextToken is not null)
                    {
                        request.NextToken = response.NextToken;
                    }
                }
                while (response.NextToken is not null);
            }
            catch (AWSOrganizationsNotInUseException ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        /// <summary>
        /// Displays information about the Organizations policies associated
        /// with an organization.
        /// </summary>
        /// <param name="policy">An Organizations policy summary to display
```

```
        /// information on the console.</param>
        private static void DisplayPolicies(PolicySummary policy)
        {
            string policyInfo = $"{policy.Id}
 {policy.Name}\t{policy.Description}";

            Console.WriteLine(policyInfo);
        }
    }
```

- API の詳細については、「 API リファレンスListPolicies」の「」を参照してください。 AWS SDK for .NET

CLI

AWS CLI

特定のタイプの組織のすべてのポリシーのリストを取得するには

次の例は、フィルターパラメータで指定された SCP のリストを取得する方法を示しています。

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

出力には、ポリシーのリストと概要情報が含まれます。

```
{
        "Policies": [
                {
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Name": "AllowAllS3Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid111",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid111",
                        "Description": "Enables account admins to delegate
 permissions for any S3 actions to users and roles in their accounts."
                },
                {
                        "Type": "SERVICE_CONTROL_POLICY",
```

```
                        "Name": "AllowAllEC2Actions",
                        "AwsManaged": false,
                        "Id": "p-examplepolicyid222",
                        "Arn": "arn:aws:organizations::111111111111:policy/
service_control_policy/p-examplepolicyid222",
                        "Description": "Enables account admins to delegate
 permissions for any EC2 actions to users and roles in their accounts."
                },
                {
                        "AwsManaged": true,
                        "Description": "Allows access to every operation",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-FullAWSAccess",
                        "Arn": "arn:aws:organizations::aws:policy/
service_control_policy/p-FullAWSAccess",
                        "Name": "FullAWSAccess"
                }
        ]
}
```

- API の詳細については、「 コマンドリファレンスListPolicies」の「」を参照してください。 AWS CLI

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、設定と実行の方法を確認してください。

```
def list_policies(policy_filter, orgs_client):
    """
    Lists the policies for the account, limited to the specified filter.

    :param policy_filter: The kind of policies to return.
    :param orgs_client: The Boto3 Organizations client.
    :return: The list of policies found.
    """
```

```
    try:
        response = orgs_client.list_policies(Filter=policy_filter)
        policies = response["Policies"]
        logger.info("Found %s %s policies.", len(policies), policy_filter)
    except ClientError:
        logger.exception("Couldn't get %s policies.", policy_filter)
        raise
    else:
        return policies
```

- API の詳細については、 ListPolicies AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパーガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK AWS Organizations での の使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

のドキュメント履歴 AWS Organizations

次の表は、 AWS Organizationsのドキュメントの主な更新をまとめたものです。

- API バージョン: 2016-11-28
- ドキュメントの最終更新日: 2024 年 6 月 6 日

変更	説明	日付
AWSOrganizationsReadOnlyAccess 管理ポリシーを更新しました。	組織内の任意のメンバーアカウントのルートユーザーの E メールアドレスを表示するためのアクセスを許可する `account:GetPrimaryEmail` アクションを AWSOrganizationsReadOnlyAccess ポリシーに追加し、組織内の任意のメンバーアカウントの有効なリージョンを表示するためのアクセスを有効にする `account:GetRegionOptStatus` アクションを追加しました。	2024 年 6 月 6 日
新しいルートユーザーの E メールアドレスの更新に関するトピック	Organizations では、組織内の任意のメンバーアカウントのルートユーザーの E メールアドレスを一元的に更新できるようになりました。	2024 年 6 月 6 日
更新されたポリシーステートメント	AWS Organizations マネージドポリシーステートメントに新しい`Sid`要素を追加しました。	2024 年 2 月 6 日

新しい管理アカウントの閉鎖に関するトピック	管理アカウントを閉鎖する方法を説明する考慮事項と詳細な手順へのリンクを追加しました。	2024 年 2 月 1 日
ベストプラクティスの更新	IAM のベストプラクティスとの整合に役立つ新しい情報をベストプラクティスのセクションに追加しました。	2023 年 6 月 12 日
AWSOrganizationsFullAccess および AWSOrganizationsReadOnlyAccess マネージドポリシーを更新しました	両方の管理ポリシーが更新され、アカウントの連絡先への書き込みまたは読み取りアクセスが可能になりました。	2022 年 10 月 21 日
AWSOrganizationsFullAccess 管理ポリシーを更新しました	マネージドポリシーが更新されました。これにより、新しい組織に必要なサービスにリンクされたロールを作成するためのアクセス許可を追加することで、組織の作成ができるようになりました。	2022 年 8 月 24 日
Organizations が AWS Organizations コンソールからアカウント機能を閉じる	管理アカウントのプリンシパルは、 AWS Organizations コンソールからメンバーアカウントを閉鎖することができ、IAM ポリシーを使用して、メンバーアカウントが誤って閉鎖されないように保護できます。	2022 年 3 月 29 日

AWS Organizations コンソールで別の連絡先を更新するためのお知らせを更新しました	Organizations では、 AWS Organizations コンソールを使用して組織内のアカウントの代替連絡先を更新できるようになりました。新しい機能と手順の「Account Management リファレンス」を追記しました。	2022 年 2 月 8 日
Organizations 管理ポリシーの更新 – 既存のポリシーへの更新	AWSOrganizationsFullAccess および AWSOrganizationsReadOnlyAccess 管理ポリシーを更新して、 AWS Organizations コンソールを介してアカウントの代替連絡先を更新または表示するために必要なアカウント API アクセス許可を付与しました。	2022 年 2 月 7 日
Organizations と Amazon DevOpsGuru の統合	Amazon DevOpsGuru を と統合することで AWS Organizations 、すべての組織アカウントでアプリケーションのヘルスを包括的にモニタリングし、インサイトを得ることができます。	2022 年 1 月 3 日
Amazon Detective との組織の統合	Amazon Detective を と統合 AWS Organizations して、Detective 動作グラフがすべての組織アカウントのアクティビティを可視化できるようにします。	2021 年 12 月 16 日

Organizations と の統合で、マルチアカウントマルチリージョンのデータ集約がサポートされ AWS Config るようになりました。	代理管理者アカウントを使用し、組織のすべてのメンバーアカウントを対象に、リソース構成とコンプライアンスデータを集約できます。詳しくは、AWS Config デベロッパーガイドの Multi-account multi-region data aggregation を参照してください。	2021年6月16日
Organizations と の統合に、委任された管理者のサポートが含まれる AWS Firewall Manager ようになりました	組織内のメンバーアカウントを、組織全体の Firewall Manager 管理者として指定できるようになりました。これにより、組織の管理アカウントからアクセス許可をより適切に分離できます。	2021 年 4 月 30 日
組織のバックアップ ポリシーで継続的なバックアップがサポートされるようになりました	AWS Backup 継続的バックアップ機能は、組織のバックアップポリシーで使用できます。	2021 年 3 月 10 日
Organizations と の統合に、委任された管理者のサポートが含まれる AWS CloudFormation StackSets ようになりました	組織内のメンバーアカウントを組織全体の AWS CloudFormation StackSets 管理者に指定できるようになりました。これにより、組織の管理アカウントからアクセス許可をより適切に分離できます。	2021 年 2 月 18 日

すべての機能を有効にしつつアカウントの招待を継続する	AWS は、組織内のすべての機能を有効にするプロセスを更新しました。既存のアカウントが招待に応答するのを待っている間も、引き続き新しいアカウントを組織に参加するよう招待できるようになりました。	2021 年 2 月 3 日
AWS Organizations コンソールのバージョン 2.0 を導入	AWS は、 AWS コンソールの新しいバージョンを導入しました。すべてのドキュメントが更新され、タスクの実行に新しい方法が適用されます。	2021 年 1 月 21 日
Organizations が との統合をサポートするようになりました AWS Marketplace	を有効に AWS Marketplace して、組織内のすべてのアカウント間でソフトウェアライセンスをより簡単に共有できるようになりました。	2020年12月3日
Organizations が Amazon S3 Lens との統合のサポートを開始	Amazon S3 Lens は、Organizations の信頼されたアクセスと代理管理者の両方をサポートしています。詳しくは、Amazon Simple Storage Service ユーザーガイドの「Amazon S3 Storage Lens」を参照してください。	2020 年 11 月 18 日
クロスアカウントバックアップコピー	バックアップポリシーを使用して組織内のリソースをバックアップする場合、バックアップのコピーを組織 AWS アカウント 内の他の に保存できるようになりました。	2020 年 11 月 18 日

AWS リージョン 中国の が Organizations の信頼されたサービス AWS Resource Access Manager として をサポートするようになりました	Organizations と中国 AWS RAM で を使用する場合、信頼できるサービスとして Organizations と統合する AWS RAM 機能を使用できるようになりました。	2020 年 11 月 18 日
Organizations が との統合をサポートするようになりました AWS Security Hub	組織内のすべてのアカウントで Security Hub を有効にし、組織のメンバーアカウント 1 つを Security Hub の委任管理者アカウントとして指定できます。	2020 年 11 月 12 日
マスターアカウントの名称の変更	AWS Organizations は、「マスターアカウント」の名前を「管理アカウント」に変更しました。変更されたのは名称だけです。機能に変更はありません。	2020 年 10 月 20 日
ベストプラクティスに関する新しいセクションとトピック	AWS Organizationsに関するベストプラクティスの新しいセクションを追加しました。この新しいセクションには、管理アカウント、メンバーアカウントのルートユーザー、パスワード管理のベストプラクティスについて説明するトピックが含まれています。	2020 年 10 月 6 日

ベストプラクティスのセクションを新設し、最初の 2 ページを追加	AWS Organizationsに関するベストプラクティスを説明するトピックの新しいセクションができました。今回の追加には、組織の管理アカウントのベストプラクティスのトピックと、メンバーアカウントのベストプラクティスのトピックが含まれています。	2020 年 10 月 2 日
組織のバックアップポリシーは、VSS (ボリュームシャドウコピーサービス) を使用して、Windows EC2 インスタンスでのアプリケーションコンシステントバックアップをサポートするようになりました	バックアップポリシーのサポートに、「`advanced_backup_settings`　」セクションが新たに追加されます。この新しいセクションに最初に追加されるのは、ユーザーによる有効と無効の切り替えが可能な `WindowsVSS` と呼ばれる `ec2` 設定です。詳しくは、AWS Backup デベロッパーガイドの Creating a VSS-Enabled Windows Backup を参照してください。	2020 年 9 月 24 日
Organizations が tag-on-create とタグベースのアクセスコントロールをサポート	Organizations リソースを作成するときにリソースにタグを追加できます。タグポリシーを使用して、Organizations リソースでのタグの使用を標準化できます。特定のタグのキーと値があるリソースだけにアクセスを制限する IAM ポリシーを使用できます。	2020 年 9 月 15 日
信頼されたサービス AWS Health として を追加	組織内のアカウント間で AWS Health イベントを集約できます。	2020 年 8 月 4 日

人工知能 (AI) サービスのオプトアウトポリシー	AI サービスのオプトアウトポリシーを使用して、 AWS AI サービスが AI サービスおよび AWS テクノロジーの開発と継続的な改善のために、それらのサービスによって処理された顧客コンテンツ (AI コンテンツ) を保存および使用するかどうかを制御できます。	2020年7月8日
バックアップポリシーと との統合を追加 AWS Backup	バックアップポリシーを使用して、バックアップポリシーを作成し、組織内のすべてのアカウントに適用できます。	2020 年 6 月 24 日
IAM Access Analyzer の委任管理をサポート	組織内のアクセスアナライザーの管理アクセスを、指定したメンバーアカウントに委任できます。	2020 年 3 月 30 日
との統合 AWS CloudFormation StackSets	サービス管理スタックセットを作成して、 AWS Organizationsによって管理されるアカウントにスタックインスタンスをデプロイできます。	2020 年 2 月 11 日
Compute Optimizer との統合	組織のアカウントで操作できるサービスとして Compute Optimizer が追加されました。	2020 年 2 月 4 日
タグポリシー	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。	2019 年 11 月 26 日

Systems Manager との統合	Systems Manager Explorer では、組織内のすべての AWS アカウント でオペレーションデータを同期できます。	2019 年 11 月 26 日
aws:PrincipalOrgPaths	新しいグローバル条件キーは、リクエストを行う IAM ユーザー、IAM ロール、または AWS アカウント ルートユーザーの AWS Organizations パスをチェックします。	2019 年 11 月 20 日
AWS Config ルールとの統合	AWS Config API オペレーションを使用して、組織内のすべての AWS アカウント で AWS Config ルールを管理できます。	2019 年 7 月 8 日
信頼されたアクセスに対応した新しいサービス	組織のアカウントで操作できるサービスとして Service Quotas を追加しました。	2019 年 6 月 24 日
AWS Control Tower との統合	AWS Control Tower が、組織内のアカウントと連携できるサービスとして追加されました。	2019 年 6 月 24 日
との統合 AWS Identity and Access Management	IAM は、組織のエンティティ (組織のルート、OU、アカウント) に対してサービスの最終アクセスデータを提供します。このデータを使用して、必要な AWS サービスのみにアクセスを制限できます。	2019 年 6 月 20 日

アカウントへのタグ付け	組織内のアカウントへのタグ付けとタグ解除を行い、組織内のアカウントのタグを表示できます。	2019 年 6 月 6 日
リソース、条件、サービスコントロールポリシー (SCP) 内の `NotAction` 要素	リソース、条件および SPC 内の `NotAction` 要素を指定して、組織や組織単位 (OU) 内のアカウント間のアクセスを拒否できるようになりました。	2019 年 3 月 25 日
信頼されたアクセスに対応した新しいサービス	AWS License Manager および Service Catalog が、組織内のアカウントと連携できるサービスとして追加されました。	2018 年 12 月 21 日
信頼されたアクセスに対応した新しいサービス	AWS CloudTrail と が、組織内のアカウントと連携できるサービスとして AWS RAM 追加されました。	2018 年 12 月 4 日
信頼されたアクセスに対応した新しいサービス	AWS Directory Service は、組織内のアカウントと連携できるサービスとして を追加しました。	2018 年 9 月 25 日
E メールアドレスの検証	組織に既存のアカウントを招待する前に、管理アカウントに関連付けられた E メールアカウントを所有していることを確認する必要があります。	2018 年 9 月 20 日
CreateAccount 通知	`CreateAccount` 通知は管理アカウントの CloudTrail 口グに発行されます。	2018 年 6 月 28 日

信頼されたアクセスに対応した新しいサービス	AWS Artifact は、組織内のアカウントと連携できるサービスとして を追加しました。	2018 年 6 月 20 日
信頼されたアクセスに対応した新しいサービス	AWS Config および は、組織内のアカウントと連携できるサービスとして AWS Firewall Manager 追加されました。	2018 年 4 月 18 日
信頼されたサービスへのアクセス	組織内のアカウントで動作する一部の AWS サービスへのアクセスを有効または無効にできるようになりました。IAM Identity Center は、最初にサポートされている信頼されたサービスです。	2018 年 3 月 29 日
アカウントの削除がセルフサービスになりました	に連絡 AWS Organizations せずに、 内から作成されたアカウントを削除できるようになりました AWS Support。	2017 年 12 月 19 日
新しいサービスのサポートを追加 AWS IAM Identity Center	AWS Organizations で AWS IAM Identity Center (IAM Identity Center) との統合がサポートされるようになりました。	2017 年 12 月 7 日
AWS は、すべての組織アカウントにサービスにリンクされたロールを追加しました	という名前のサービスにリンクされたロール`AWSServiceRoleForOrganizations` が組織内のすべてのアカウントに追加され、 AWS Organizations と他の AWS のサービスの統合が可能になります。	2017 年 10 月 11 日

作成したアカウントを削除できるようになりました	作成したアカウントを組織から削除するには、 AWS Supportまでお問い合わせください。	2017 年 6 月 15 日
サービスの起動	新しいサービスの起動に付随する AWS Organizations ドキュメントの初版。	2017 年 2 月 17 日

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