管理者ガイド

AWS Service Catalog

AWS Service Catalog: 管理者ガイド

Table of Contents

Service Catalog とは

Service Catalog では、AWS 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。この IT サービスには、仮想マシンイメージ、サーバー、ソフトウェア、データベースなどから包括的な多層アプリケーションアーキテクチャまで、あらゆるものが含まれます。

Service Catalog により、組織は一般的にデプロイされる IT サービスを集中管理でき、一貫性のあるガバナンスを達成し、コンプライアンス要件を満たすうえで役立ちます。エンドユーザーは、組織によって設定された制約に従って、必要な承認済みの IT サービスのみをすばやくデプロイできます。

Service Catalog には次の利点があります。

- 標準化

  製品を起動できる場所、使用できるインスタンスのタイプ、およびその他の多くの設定オプションを制限することにより、承認済みのアセットを管理できます。その結果、組織全体で製品をプロビジョニングするために標準化された環境が実現します。
- セルフサービスの検出と起動

  ユーザーは、アクセスできる製品 (サービスまたはアプリケーション) のリストを参照し、使用する製品を見つけ、プロビジョニング済み製品としてすべて自分で起動できます。
- きめ細かなアクセスコントロール

  管理者は、カタログから製品のポートフォリオを生成し、プロビジョニングで使用する制約とリソースタグを追加して、AWS Identity and Access Management (IAM) のユーザーとグループを通じてポートフォリオにアクセス権限を付与します。
- 拡張性とバージョン管理

  管理者は、任意の数のポートフォリオに製品を追加し、別のコピーを作成することなく、それを制限できます。製品を新しいバージョンに更新すると、それを参照するすべてのポートフォリオで、すべての製品に対して更新が伝播されます。

詳細については、Service Catalog の詳細ページ を参照してください。

Service Catalog API では、AWS Management Console を使用する代わりに、すべてのエンドユーザーアクションに対する制御をプログラムで行うことができます。詳細については、「Service Catalog デベロッパーガイド」を参照してください。

ビデオ: AWS Service Catalog の概要

このビデオ (7:27) では、厳選された AWS 製品カタログを作成、整理、管理する方法と、権限レベルで製品を共有する方法について説明します。その結果、エンドユーザーは、基盤となる AWS サービスに直接アクセスしなくても、承認された IT リソースを迅速にプロビジョニングできます。

AWS Service Catalogへの概論

Service Catalog の概要

Service Catalog の使用を開始するにあたり、そのコンポーネントと、管理者とエンドユーザーの初期ワークフローについて理解しておくと役立ちます。

[ユーザー]

Service Catalog では、次のタイプのユーザーがサポートされています。

- カタログ管理者 (管理者) - 製品 (アプリケーションおよびサービス) のカタログを管理し、ポートフォリオに整理してエンドユーザーにアクセス権限を付与します。カタログ管理者は、AWS CloudFormation テンプレートの準備や制約の設定を行い、製品の IAM ロールを管理して、高度なリソース管理を提供します。
- エンドユーザー - IT 部門またはマネージャーから AWS 認証情報を受け取り、AWS Management Console を使用して、アクセス権限を付与されている製品を起動します。単純にユーザーと呼ばれることもあるエンドユーザーには、操作要件によって異なるアクセス許可を付与できます。たとえば、ユーザーに (使用する製品によって求められるすべてのリソースを起動および管理できるように) 最大限のアクセス許可レベルを付与することも、特定のサービス機能の使用に対するアクセス許可のみを付与することもできます。

製品

製品とは、AWS でのデプロイに利用できるようにする IT サービスのことです。製品は、EC2 インスタンス、ストレージボリューム、データベース、モニタリング設定、ネットワーキングコンポーネント、パッケージ化された AWS Marketplace 製品など、1 つ以上の AWS リソースで構成されます。製品には、AWS Linux を実行する 1 つのコンピューティングインスタンス、独自の環境で実行される完全に構成された多層ウェブアプリケーション、その中間に位置するものを使用できます。

AWS CloudFormation テンプレートをインポートして製品を作成します。AWS CloudFormation テンプレートでは、製品に必要な AWS リソース、リソース間の関係、エンドユーザーが製品を起動した

ときにセキュリティグループの設定、キーペアの作成、その他のカスタマイズを行うために組み込むことができるパラメータを定義します。

HashiCorp Terraform Open Source と Terraform Cloud のサポート

AWS Service Catalog は、 内の HashiCorp Terraform Open Source と Terraform Cloud の設定を管理する、迅速なセルフサービスプロビジョニングを可能にしますAWS。Service Catalog は、AWS の Terraform 構成を大規模に整理、管理、配布するための単一のツールとして使用できます。標準化され事前承認された Terraform テンプレートのカタログ作成、アクセス制御、最小権限のプロビジョニング、バージョン管理、タグ付け、数千の AWS アカウントへの共有など、Service Catalog の主要な機能にアクセスできます。エンドユーザーには、アクセスできる製品とバージョンの簡単なリストが表示され、それらの製品を 1 回のアクションでデプロイできます。

Terraform 製品のチュートリアルで詳細を確認したり、完成させたりするには、Terraform 製品の使用開始 をご覧ください。

プロビジョニングされた製品

AWS CloudFormation スタックにより、製品インスタンスを単一のユニットとしてプロビジョニング、タグ付け、更新、および終了できるので、製品のライフサイクルを管理しやすくなります。AWS CloudFormation スタックには、JSON 形式または YAML 形式で記述された AWS CloudFormation テンプレートとそれに関連するリソースのコレクションが含まれます。プロビジョニングされた製品はスタックです。エンドユーザーが製品を起動すると、Service Catalog によってプロビジョニングされる製品のインスタンスは、製品の実行に必要なリソースを伴うスタックになります。詳細については、AWS CloudFormationユーザーガイド を参照してください。

ポートフォリオ

ポートフォリオとは、製品の集合で、設定情報も含まれます。ポートフォリオは、特定の製品を使用できるユーザー、そのユーザーに許可される製品の使用方法の管理に役立ちます。Service Catalog では、組織のユーザータイプごとにカスタマイズしたポートフォリオを作成し、適切なポートフォリオへのアクセス権を選択的に付与できます。製品の新しいバージョンをポートフォリオに追加すると、そのバージョンは、現在のすべてのユーザーに対して自動的に利用可能になります。

また、自分のポートフォリオを他の AWS アカウントと共有して、そのアカウントの管理者がそのポートフォリオに制約 (ユーザーが作成できる EC2 インスタンスの制限など) を加えて配布できるようにすることができます。ポートフォリオ、アクセス権限、共有、制約を使用することで、組織のニーズおよび標準に合わせて適切に設定された製品をユーザーが起動するよう制御できます。

バージョニング

Service Catalog では、カタログで複数のバージョンの製品を管理できます。このアプローチにより、ソフトウェアの更新または設定の変更に基づいて新しいバージョンのテンプレートと関連するリソースを追加できます。

新しいバージョンの製品を作成すると、その製品にアクセスできるすべてのユーザーに更新が自動的に配信されるので、ユーザーは使用する製品のバージョンを選択できます。ユーザーは、製品の実行中のインスタンスを新しいバージョンにすばやく簡単に更新できます。

アクセス許可

ポートフォリオへのアクセス権をユーザーに付与すると、ユーザーはポートフォリオを閲覧して、それに含まれる製品を起動できます。AWS Identity and Access Management (IAM) アクセス許可を適用して、カタログを表示および変更できるユーザーを制御できます。IAM アクセス許可は IAM ユーザー、グループ、およびロールに割り当てることができます。

ユーザーが IAM ロールが割り当てられている製品を起動すると、Service Catalog では、そのロールで、AWS CloudFormation を使用して製品のクラウドリソースを起動します。IAM ロールを各製品に割り当てると、承認されていない操作を実行できるアクセス権限がユーザーに割り当てられないようにすることができます。また、ユーザーは、カタログを使用してリソースをプロビジョニングできます。

制約

制約によって、特定の AWS リソースを製品に対してデプロイできる方法を制御します。制約を使用して、製品に制限を適用し、ガバナンスまたはコスト管理を実現できます。AWS Service Catalog の制約にはさまざまなタイプがあります。起動の制約、通知の制約、テンプレートの制約です。

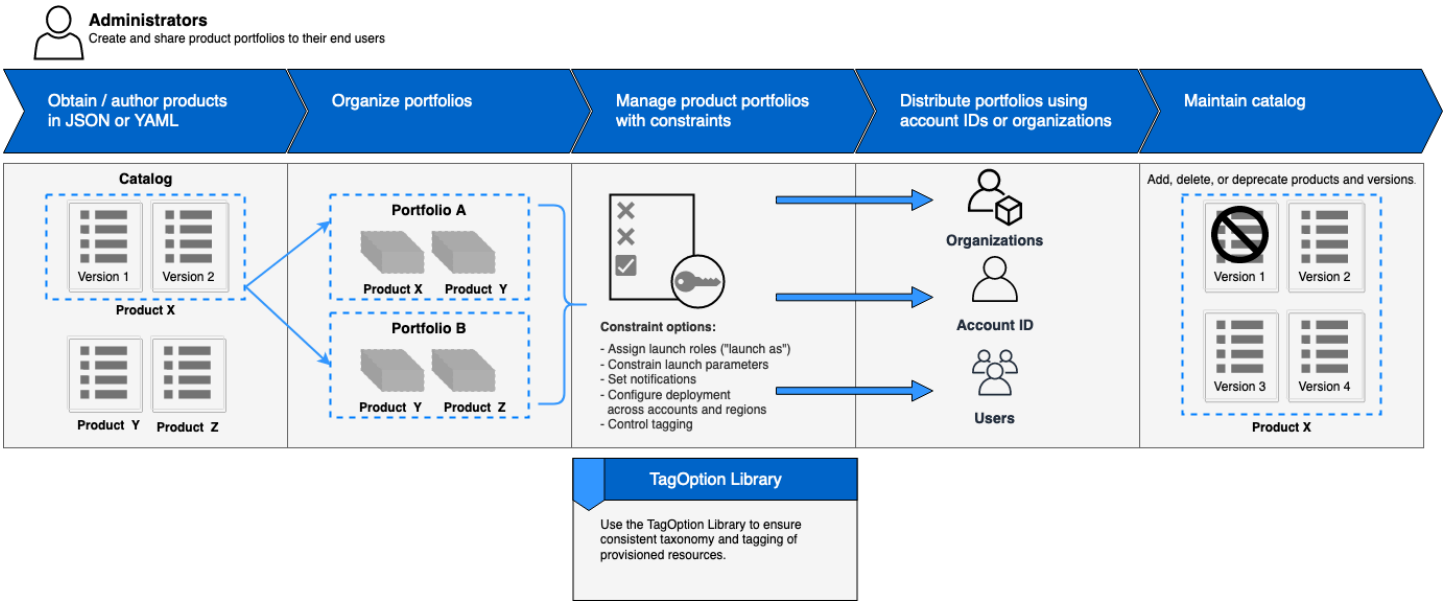
起動の制約では、ポートフォリオ内の製品に対してロールを指定します。このロールを使用して、起動時にリソースをプロビジョニングすると、ユーザーがカタログから製品をプロビジョニングする機能に影響を与えずに、ユーザーのアクセス許可を制限できます。

通知の制約は、Amazon SNS トピックを使用してスタックのイベントに関する通知を受けることができます。

テンプレート制約では、製品を起動したときにユーザーが使用できる設定パラメータ (EC2 インスタンスタイプ、IP アドレス範囲など) を制限します。テンプレート制約では、汎用 AWS CloudFormation テンプレートを製品に再利用して、製品単位またはポートフォリオ単位でテンプレートに制限を適用します。

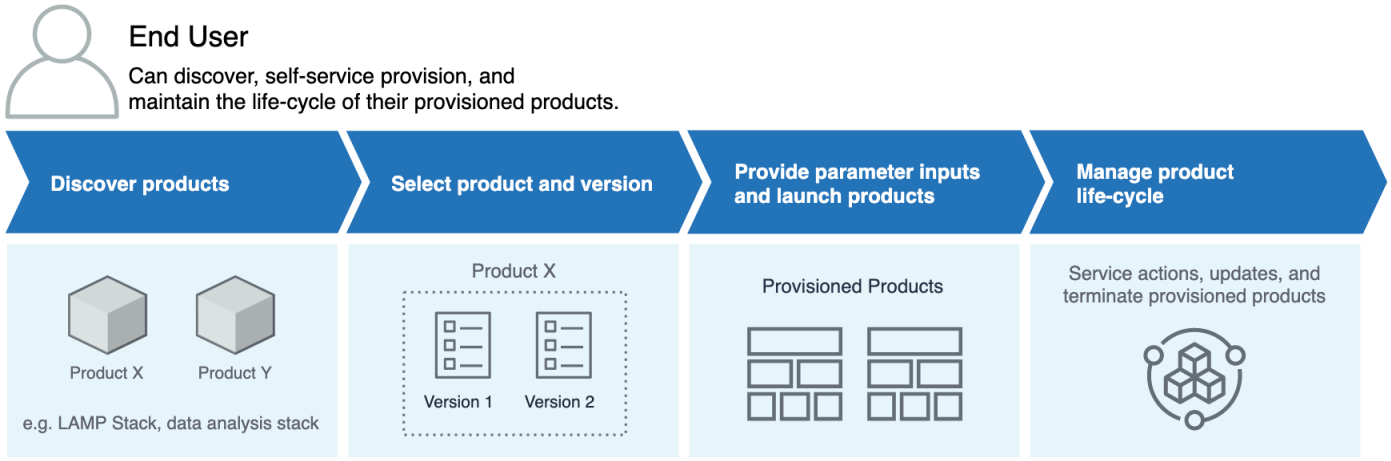
管理者の初期ワークフロー

次の図は、カタログを作成するときの管理者の初期ワークフローを示しています。



エンドユーザーの初期ワークフロー

次の図は、エンドユーザーの初期ワークフローを示しています。



AWS Service Catalog のデフォルトのサービスクォータ

AWS アカウントには、、制約AWS Organizations、ポートフォリオ、製品、プロビジョニング済み製品、リージョン、サービスアクション、および のデフォルトクォータがあります TagOptions。

Service Quotas を使用して、クォータを管理したり、クォータの増加をリクエストしたりできます。Service Quotas の詳細については、Service Quotas Service Quotas ユーザーガイドの「Service Quotas とは」を参照してください。クォータの増加をリクエストする方法については、「クォータ増加のリクエスト」を参照してください。

AWS Organizations

- 組織あたりの AWS Service Catalog 委任管理者数: 50

制約クォータ

- ポートフォリオ別の製品あたりの制約事項: 100

ポートフォリオのクォータ

- ポートフォリオあたりのユーザー、グループ、ロール: 100
- ポートフォリオあたりの製品: 150
- ポートフォリオあたりのタグ: 20
- ポートフォリオあたりの共有アカウント: 5000
- タグキーあたりのタグ値: 25

製品クォータ

- 製品あたりのユーザー、グループ、ロール: 200
- 製品あたりの製品バージョン: 100
- 製品あたりのタグ: 20
- タグキーあたりのタグ値: 25

プロビジョニング済み製品のクォータ

- プロビジョニング済み製品あたりのタグ: 50

リージョン別クォータ

- ポートフォリオ: 100
- 製品: 350

サービスアクションクォータ

- リージョンあたりのサービスアクション: 200
- 製品バージョンごとのサービスアクションの関連付け: 25

TagOptions クォータ

- TagOptions リソースあたり: 25
- あたりの値 TagOption: 25

AWS Service Catalogのセットアップ

の使用を開始する前に AWS Service Catalog、以下のタスクを完了してください。

トピック

にサインアップする AWS アカウント

がない場合は AWS アカウント、次のステップを実行して作成します。

にサインアップするには AWS アカウント

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

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

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

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

管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

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

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

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

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

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

管理アクセス権を持つユーザーとしてサインインする

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

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

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」を参照してください。

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

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

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

- のユーザーとグループ AWS IAM Identity Center：

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

- IAM 内で、ID プロバイダーによって管理されているユーザー:

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

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

AWS Service Catalog 管理者へのアクセス権限の付与

カタログ管理者には、AWS Service Catalog 管理者コンソールビューへのアクセスと、以下のようなタスクの実行を許可する IAM アクセス権限が必要です。

- ポートフォリオの作成と管理
- 製品の作成と管理
- 製品を起動するときにエンドユーザーが使用可能なオプションを管理するためのテンプレート制約の追加

- エンドユーザーが製品を起動するときに AWS Service Catalog が引き受ける IAM ロールを定義する起動制約の追加
- 製品へのエンドユーザーアクセス権の付与

このチュートリアルを完了するには、ユーザー、または IAM アクセス権限を管理する管理者は、IAM ユーザー、グループ、またはロールにポリシーをアタッチする必要があります。

カタログ管理者にアクセス権限を付与するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. ナビゲーションペインの [アクセス管理] を展開し、[ユーザー] を選択します。カタログ管理者として使用する IAM ユーザーをすでに作成している場合は、そのユーザー名を選択して [許可の追加] を選択します。それ以外の場合は、次のようにしてユーザーを作成します。
   a. [ユーザーを追加] を選択します。
   b. [User name] に、**ServiceCatalogAdmin** と入力します。
   c. [プログラムによるアクセス] と [AWS Management Console アクセス] を選択します。
   d. [次へ: アクセス許可] を選択します。
3. [既存のポリシーを直接添付する] を選択します。
4. [ポリシーの作成] を選択して、次の操作を行います。
   a. [JSON] タブを選択します。
   b. 次のポリシー例をコピーし、[Policy Document] に貼り付けます。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateKeyPair",
                "iam:AddRoleToInstanceProfile",
                "iam:AddUserToGroup",
                "iam:AttachGroupPolicy",
                "iam:CreateAccessKey",
                "iam:CreateGroup",
                "iam:CreateInstanceProfile",
                "iam:CreateLoginProfile",
```

```
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:Get*",
                "iam:List*",
                "iam:PutRolePolicy",
                "iam:UpdateAssumeRolePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

c. [次へ: タグ] を選択します。

d. (オプション) [タグを追加] を選択して、キーと値のペアをリソースに関連付けます。最大 50 個のタグを追加できます。

> **Note**
>
> タグは、リソースに追加できるキーと値のペアです。これにより、リソースの識別、整理、検索が容易になります。詳細については、 AWS 全般のリファレンス リファレンスガイドのAWS リソースのタグ付けを参照してください。

e. [次へ: 確認] を選択します。

f. [Policy Name] に「**ServiceCatalogAdmin-AdditionalPermissions**」と入力します。

> **Important**
>
> AWS Service Catalog が Amazon S3 に保存しているテンプレートにアクセスするには、管理者に Amazon S3 アクセス許可を付与する必要があります。詳細については、Amazon Simple Storage Service ユーザーガイドの「ユーザーポリシーの例」を参照してください。

g. [ポリシーの作成] を選択します。

5. アクセス権限ページのブラウザウィンドウに戻り、[Refresh] を選択します。

6. 検索フィールドに **ServiceCatalog** と入力してポリシーリストをフィルタリングします。

7. **AWSServiceCatalogAdminFullAccess** ポリシーと **ServiceCatalogAdmin-AdditionalPermissions** ポリシーのチェックボックスを選択し、[次へ: 確認] を選択します。
8. ユーザーを更新する場合は [Add permissions] を選択します。

   ユーザーを作成する場合は [Create user] を選択します。認証情報をダウンロードまたはコピーして、[Close] を選択できます。
9. カタログ管理者としてサインインするには、アカウント固有の URL を使用します。この URL を確認するには、ナビゲーションペインの [Dashboard] を選択し、[Copy Link] を選択します。ブラウザにリンクを貼り付け、この手順で更新または作成した IAM ユーザーの名前とパスワードを使用します。

AWS Service Catalog エンドユーザーへのアクセス権限の付与

エンドユーザーが AWS Service Catalog を使用できるようにする前に、AWS Service Catalog エンドユーザーコンソールビューへのアクセス権を付与する必要があります。アクセス権を付与するには、IAM ユーザー、グループ、またはエンドユーザーが使用するロールにポリシーをアタッチします。次の手順では、**AWSServiceCatalogEndUserFullAccess** ポリシーを IAM グループにアタッチします。

エンドユーザーグループにアクセス権限を付与するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. ナビゲーションペインで、[ユーザーグループ] を選択します。
3. [グループを作成] を選択して、次の操作を行います。
   a. [ユーザーグループ名] に「**Endusers**」と入力します。
   b. 検索フィールドに **AWSServiceCatalog** と入力してポリシーリストをフィルタリングします。
   c. **AWSServiceCatalogEndUserFullAccess** ポリシーのチェックボックスを選択します。また、代わりに **AWSServiceCatalogEndUserReadOnlyAccess** を選択することもできます。
   d. グループを作成 を選択します。
4. ナビゲーションペインで [Users (ユーザー)] を選択します。
5. [ユーザーを追加] を選択し、次の操作を行います。

a. [ユーザー名] にユーザーの名前を入力します。

b. 「パスワード - AWS 管理コンソールへのアクセス」を選択します。

c. [次へ: アクセス許可] を選択します。

d. [ユーザーをグループに追加] を選択します。

e. [Endusers] グループのチェックボックスをオンにし、[次へ: タグ]、[次へ: レビュー] の順に を選択します。

f. [確認] ページで、[ユーザーの作成] を選択します。認証情報をダウンロードまたはコピーして、[閉じる] を選択します。

Terraform プロビジョニングエンジンのインストールと設定

AWS Service Catalog で Terraform 製品を正常に使用するには、Terraform 製品を管理するのと同じアカウントに Terraform プロビジョニングエンジンをインストールして設定する必要があります。はじめに、AWS が提供する Terraform プロビジョニングエンジンを使用できます。このエンジンは、Terraform プロビジョニングエンジンが AWS Service Catalog と連携するために必要なコードとインフラストラクチャがインストールおよび設定されます。この 1 回限りのセットアップには約 30 分かかります。 AWS Service Catalogには、Terraform プロビジョニングエンジン をインストールして設定する手順を含む GitHub リポジトリが用意されています。

キューの決定

プロビジョニング操作を呼び出すと、AWS Service Catalog はプロビジョニングエンジンの関連キューに送信するペイロードメッセージを準備します。キューの ARN を構築するには、AWS Service Catalog は以下の前提条件を満たす必要があります。

- プロビジョニングエンジンはプロダクトオーナーのアカウントにあります
- プロビジョニングエンジンは、AWS Service Catalog への呼び出しが行われたのと同じリージョンにあります。
- プロビジョニングエンジンのキューは、以下に詳述する文書化された命名スキーマに従います。

例えば、アカウント 1111111111 によって作成された製品を使用してアカウント 0000000000000 us-east-1から が呼び ProvisionProduct 出された場合、 は正しい SQS ARN が であるAWS Service Catalogと仮定しますarn:aws:sqs:us-east-1:0000000000000:ServiceCatalogTerraformOSProvisionOperationQueue。

DescribeProvisioningParameters によって呼び出される Lambda 関数にも同じロジックが適用されます。

Confused Deputy を Terraform プロビジョニングエンジンに追加する

lambda:Invoke 操作のためのアクセスを制限するための、エンドポイントの Confused Deputy コンテキストキー

AWS Service Catalog が提供するエンジンによって作成されたパラメータパーサー Lambda関数には、AWS Service Catalog サービスプリンシパルのみにクロスアカウント lambda:Invoke 権限を付与するアクセスポリシーがあります。

```
{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "lambda:InvokeFunction",
          "Resource": "arn:aws:lambda:us-
east-1:account_id:function:ServiceCatalogTerraformOSParameterParser"
    }
  ]
}
```

AWS Service Catalog との統合が正しく機能するために必要な権限は、これだけです。ただし、aws:SourceAccount Confused Deputy コンテキストキーを使用すると、これをさらに制限できます。AWS Service Catalog がこれらのキューにメッセージを送信すると、AWS Service Catalog はキーにプロビジョニングアカウントの ID を入力します。これは、ポートフォリオ共有を通じて製品を配布する予定で、特定のアカウントだけがエンジンを使用できるようにしたい場合に役立ちます。

たとえば、以下の条件を使用して、000000000000 と 111111111111 を起点とするリクエストのみを許可するよう、エンジンに制限をかけることができます。

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

```
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "lambda:InvokeFunction",
          "Resource": "arn:aws:lambda:us-
east-1:account_id:function:ServiceCatalogTerraformOSParameterParser",
          "Condition": {
            "StringLike": {
              "aws:SourceAccount": ["000000000000", "111111111111"]
        }
      }
    }
  ]
}
```

sqs:SendMessage 操作のためのアクセスを制限するための、エンドポイントの Confused Deputy コンテキストキー

プロビジョニングオペレーションでは、AWS Service Catalogが提供したエンジンによって作成された Amazon SQS キューを取り込むためのアクセスポリシーがあり、AWS Service Catalog サービスプリンシパルのみにクロスアカウント sqs:SendMessage (および関連する KMS) 権限を付与します。

```
{
      "Version": "2008-10-17",
      "Statement": [
        {
          "Sid": "Enable AWS Service Catalog to send messages to the queue",
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": "sqs:SendMessage",
          "Resource": [
            "arn:aws:sqs:us-
east-1:account_id:ServiceCatalogTerraformOSProvisionOperationQueue"
          ]
        },
        {
          "Sid": "Enable AWS Service Catalog encryption/decryption permissions when
 sending message to queue",
```

```
          "Effect": "Allow",
          "Principal": {
            "Service": "servicecatalog.amazonaws.com"
          },
          "Action": [
            "kms:DescribeKey",
            "kms:Decrypt",
            "kms:ReEncrypt",
            "kms:GenerateDataKey"
          ],
          "Resource": "arn:aws:kms:us-east-1:account_id:key/key_id"
    }
  ]
}
```

AWS Service Catalog との統合が正しく機能するために必要な権限は、これだけです。ただし、aws:SourceAccount Confused Deputy コンテキストキーを使用すると、これをさらに制限できます。AWS Service Catalog がこれらのキューにメッセージを送信すると、AWS Service Catalog はキーにプロビジョニングアカウントの ID を入力します。これは、ポートフォリオ共有を通じて製品を配布する予定で、特定のアカウントだけがエンジンを使用できるようにしたい場合に役立ちます。

たとえば、以下の条件を使用して、000000000000 と 111111111111 を起点とするリクエストのみを許可するよう、エンジンに制限をかけることができます。

```
{
      "Version": "2008-10-17",
      "Statement": [
      {
        "Sid": "Enable AWS Service Catalog to send messages to the queue",
        "Effect": "Allow",
        "Principal": {
          "Service": "servicecatalog.amazonaws.com"
        },
        "Action": "sqs:SendMessage",
        "Resource": [
          "arn:aws:sqs:us-
east-1:account_id:ServiceCatalogTerraformOSProvisionOperationQueue"
        ],
        "Condition": {
          "StringLike": {
            "aws:SourceAccount": ["000000000000", "111111111111"]
```

```
          }
        }
      },
      {
        "Sid": "Enable AWS Service Catalog encryption/decryption permissions when
 sending message to queue",
        "Effect": "Allow",
        "Principal": {
          "Service": "servicecatalog.amazonaws.com"
        },
        "Action": [
          "kms:DescribeKey",
          "kms:Decrypt",
          "kms:ReEncrypt",
          "kms:GenerateDataKey"
        ],
        "Resource": "arn:aws:kms:us-east-1:account_id:key/key_id"
    }
  ]
}
```

開始方法

AWS Service Catalog の使用を開始するには、入門ライブラリにある適切に設計された製品テンプレートの 1 つを使用するか、入門チュートリアルの手順に従うことができます。

このチュートリアルでは、カタログ管理者およびエンドユーザーとしてタスクを実行します。カタログ管理者として、ポートフォリオを作成し、次に製品を作成します。エンドユーザーは、エンドユーザーコンソールにアクセスして製品を起動できることを確認します。製品は次のいずれかです。

- Amazon Linux 上で動作し、製品が使用できる AWS リソースを定義する AWS CloudFormation テンプレートに基づいているクラウド開発環境。
- Terraform プロビジョニングエンジン上で動作し、製品が使用できる AWS リソースを定義する tar.gz 設定ファイルに基づいているオープンソース環境。

> **Note**
>
> 開始する前に、必ず AWS Service Catalogのセットアップ のアクション項目を完了していることを確認してください。

トピック

入門ライブラリ

AWS Service Catalog には、Well-Architected 製品テンプレートの入門ライブラリが用意されているため、すぐに作業を開始できます。入門ライブラリポートフォリオの任意の製品を自分のアカウントにコピーし、ニーズに合わせてカスタマイズすることができます。

トピック

前提条件

入門ライブラリのテンプレートを使用する前に、次のものがあることを確認してください。

- AWS CloudFormation テンプレートを使用するために必要なアクセス許可。詳細については、「AWS Identity and Access Management を使用したユーザーアクセスの制御」を参照してください。
- AWS Service Catalog の管理に必要な管理者権限。詳細については、「the section called “アイデンティティとアクセスの管理”」を参照してください。

詳細はこちら

Well-Architected フレームワークの詳細については、「AWSWell-Architected」を参照してください。

AWS CloudFormation 製品を開始する

AWS Service Catalog の使用を開始するには、入門ライブラリにある適切に設計された製品テンプレートの 1 つを使用するか、入門チュートリアルの手順に従うことができます。

このチュートリアルでは、カタログ管理者およびエンドユーザーとしてタスクを実行します。カタログ管理者として、ポートフォリオを作成し、次に製品を作成します。エンドユーザーは、エンドユーザーコンソールにアクセスして製品を起動できることを確認します。製品は、Amazon Linux 上で動作するクラウド開発環境であり、製品が使用できる AWS リソースを定義する AWS CloudFormation テンプレートに基づいています。

> **Note**
>
> 開始する前に、必ず AWS Service Catalogのセットアップ のアクション項目を完了していることを確認してください。

トピック

- ステップ 1: AWS CloudFormation テンプレートのダウンロード
- ステップ 2: キーペアを作成する
- ステップ 3: ポートフォリオを作成する
- ステップ 4: ポートフォリオに新しい製品を作成する

- ステップ 5: テンプレート制約を追加してインスタンスサイズを制限する
- ステップ 6: IAM ロールを割り当てる起動制約を追加する
- ステップ 7: ポートフォリオへのアクセス権限のエンドユーザーへの付与
- ステップ 8: エンドユーザーのエクスペリエンスをテストする

ステップ 1: AWS CloudFormation テンプレートのダウンロード

AWS CloudFormation テンプレートを使用して、ポートフォリオと製品を設定およびプロビジョニングできます。これらのテンプレートは、JSON または YAML でフォーマットできるテキストファイルで、プロビジョニングするリソースを説明します。詳細については、AWS CloudFormation ユーザーガイドの「テンプレート フォーマット」を参照してください。AWS CloudFormation エディタまたはテキストエディタを使用して、テンプレートを作成し保存できます。このチュートリアルでは、開始するためのシンプルなテンプレートが用意されています。このテンプレートでは、SSH アクセス用に設定された 1 つの Linux インスタンスを起動します。

> **Note**
>
> AWS CloudFormation テンプレートを使用するには特別な権限が必要です。作業を開始する前に、正しい権限があることを確認してください。詳細については、入門ライブラリ の前提条件を参照してください。

テンプレートのダウンロード

このチュートリアルのサンプルテンプレート、`development-environment.template` は https://awsdocs.s3.amazonaws.com/servicecatalog/development-environment.template で利用可能です。

テンプレートの概要

サンプルテンプレートのテキストは次のとおりです。

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS Service Catalog sample template. Creates an Amazon EC2 instance
                   running the Amazon Linux AMI. The AMI is chosen based on the
 region
                   in which the stack is run. This example creates an EC2 security
                   group for the instance to give you SSH access. **WARNING** This
```

```
                    template creates an Amazon EC2 instance. You will be billed for
the
                    AWS resources used if you create a stack from this template.",

 "Parameters" : {
   "KeyName": {
     "Description" : "Name of an existing EC2 key pair for SSH access to the EC2
instance.",
     "Type": "AWS::EC2::KeyPair::KeyName"
   },

   "InstanceType" : {
     "Description" : "EC2 instance type.",
     "Type" : "String",
     "Default" : "t2.micro",
     "AllowedValues" : [ "t2.micro", "t2.small", "t2.medium", "m3.medium",
"m3.large",
       "m3.xlarge", "m3.2xlarge" ]
   },

   "SSHLocation" : {
     "Description" : "The IP address range that can SSH to the EC2 instance.",
     "Type": "String",
     "MinLength": "9",
     "MaxLength": "18",
     "Default": "0.0.0.0/0",
     "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
     "ConstraintDescription": "Must be a valid IP CIDR range of the form x.x.x.x/x."
  }
 },

 "Metadata" : {
   "AWS::CloudFormation::Interface" : {
     "ParameterGroups" : [{
       "Label" : {"default": "Instance configuration"},
       "Parameters" : ["InstanceType"]
     },{
       "Label" : {"default": "Security configuration"},
       "Parameters" : ["KeyName", "SSHLocation"]
     }],
     "ParameterLabels" : {
       "InstanceType": {"default": "Server size:"},
       "KeyName": {"default": "Key pair:"},
       "SSHLocation": {"default": "CIDR range:"}
```

```
      }
    }
  },

  "Mappings" : {
    "AWSRegionArch2AMI" : {
      "us-east-1"      : { "HVM64" : "ami-08842d60" },
      "us-west-2"      : { "HVM64" : "ami-8786c6b7" },
      "us-west-1"      : { "HVM64" : "ami-cfa8a18a" },
      "eu-west-1"      : { "HVM64" : "ami-748e2903" },
      "ap-southeast-1" : { "HVM64" : "ami-d6e1c584" },
      "ap-northeast-1" : { "HVM64" : "ami-35072834" },
      "ap-southeast-2" : { "HVM64" : "ami-fd4724c7" },
      "sa-east-1"      : { "HVM64" : "ami-956cc688" },
      "cn-north-1"     : { "HVM64" : "ami-ac57c595" },
      "eu-central-1"   : { "HVM64" : "ami-b43503a9" }
    }

  },

  "Resources" : {
    "EC2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "InstanceType" : { "Ref" : "InstanceType" },
        "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
        "KeyName" : { "Ref" : "KeyName" },
        "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" :
"AWS::Region" }, "HVM64" ] }
      }
    },

    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : { "Ref" : "SSHLocation"}
        } ]
      }
    }
```

```
  },

  "Outputs" : {
    "PublicDNSName" : {
      "Description" : "Public DNS name of the new EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "EC2Instance", "PublicDnsName" ] }
    },
    "PublicIPAddress" : {
      "Description" : "Public IP address of the new EC2 instance",
      "Value" : { "Fn::GetAtt" : [ "EC2Instance", "PublicIp" ] }
    }
  }
}
```

テンプレートのリソース

テンプレートでは、製品の起動時に作成されるリソースを宣言します。次のセクションがあります。

- AWSTemplateFormatVersion (オプション) – このAWSテンプレートの作成に使用されるテンプレート形式のバージョン。最新のテンプレートの形式バージョンは 2010-09-09 であり、現時点で唯一の有効な値です。
- 説明 (オプション) – テンプレートの説明。
- パラメータ (オプション) – 製品を起動するためにユーザーが指定する必要があるパラメータ。各パラメータについて、テンプレートには、入力する値が一致する必要がある説明と制約が含まれます。制約の詳細については、「AWS Service Catalog 制約の使用」を参照してください。

  KeyName パラメータでは、エンドユーザーが AWS Service Catalog を使用して製品を起動するときに指定する必要がある、Amazon Elastic Compute Cloud (Amazon EC2) キーペア名を指定することができます。次のステップでキーペアを作成します。
- メタデータ (オプション) – テンプレートに関する追加情報を提供するオブジェクト。AWS::CloudFormation:Interface キーは、エンドユーザーコンソールビューにパラメータを表示する方法を定義します。ParameterGroups プロパティは、パラメータのグループ化の方法と、それらのグループの見出しを定義します。ParameterLabels プロパティはフレンドリなパラメータ名を定義します。このテンプレートに基づいた製品を起動するパラメータをユーザーが指定すると、エンドユーザーコンソールビューには、見出し Server size: に Instance configuration というラベルのパラメータが表示され、見出し Key pair: に CIDR range: および Security configuration というラベルのパラメータが表示されます。
- マッピング (オプション) – ルックアップ テーブルと同様に、条件付きパラメーター値の指定に使用できるキーと関連する値のマッピング。リソースと出力セクションの Fn::FindInMap 組み

込み関数を使用して、キーを対応する値と一致させることができます。上記のテンプレートには、AWS リージョンのリストと、それぞれに対応する Amazon マシンイメージ (AMI) が含まれています。AWS Service Catalog は、このマッピングを使用して、ユーザーが AWS Management Console で選択した AWS リージョンに基づいて、どの AMI を使用するかを決定します。

- リソース (必須) — スタックリソースとそのプロパティ。テンプレートの「リソース」セクションと「出力」セクションでリソースを参照できます。上記のテンプレートでは、Amazon Linux を実行する EC2 インスタンスと、そのインスタンスへの SSH アクセスを許可するセキュリティ グループを指定します。EC2 インスタンスリソースの [プロパティ] セクションでは、ユーザーが入力した情報を使用して、SSH アクセス用のインスタンスタイプとキー名を構成します。

  AWS CloudFormation は、現在の AWS リージョンを使用して、前に定義されたマッピングから AMI ID を選択し、それにセキュリティグループを割り当てます。セキュリティグループは、ユーザーが指定する CIDR IP アドレス範囲からポート 22 でインバウンドアクセスを許可するように設定されます。
- 出力 (オプション) – 製品の発売が完了したことをユーザーに通知するテキスト。提供されたテンプレートは、起動されたインスタンスのパブリック DNS 名を取得し、それをユーザーに表示します。ユーザーが SSH を使用してインスタンスに接続するためには、DNS 名が必要です。

テンプレートの構造の詳細については、 AWS CloudFormationユーザーガイド の「テンプレートリファレンス」を参照してください。

ステップ 2: キーペアを作成する

エンドユーザーが、このチュートリアル用のサンプルテンプレートに基づいた製品を起動できるようにするには、Amazon EC2 キーペアを作成する必要があります。キーペアは、データを暗号化するために使用されるパブリックキーと、データを復号化するために使用されるプライベートキーの組み合わせです。キーペアの詳細については、 AWS コンソールにサインインし、Amazon EC2 ユーザーガイド」の「Amazon EC2 キーペア」を確認してください。 Amazon EC2

このチュートリアルの AWS CloudFormation テンプレート にはdevelopment-environment.template、 KeyNameパラメータが含まれています。

```
. . .
  "Parameters" : {
    "KeyName": {
      "Description" : "Name of an existing EC2 key pair for SSH access to the EC2
 instance.",
```

```
      "Type": "AWS::EC2::KeyPair::KeyName"
    },
. . .
```

エンドユーザーは、 AWS Service Catalog を使用してテンプレートに基づく製品を起動するときに、キーペアの名前を指定する必要があります。

使用したいキーペアがすでにアカウントにある場合は、「ステップ 3: ポートフォリオを作成する」に進むことができます。それ以外の場合は、以下の手順を完了します。

キーペアを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [Network & Security] で、[Key Pairs] を選択します。
3. [Key Pairs] ページで、[Create Key Pair] を選択します。
4. [Key pair name] に、覚えやすい名前を入力し、[作成] を選択します。
5. コンソールでプライベートキーファイルの保存を求められたら、安全な場所に保存します。

> **Important**
> プライベートキーのファイルを保存できるのは、このタイミングだけです。

ステップ 3: ポートフォリオを作成する

製品をユーザーに提供するには、最初に製品のポートフォリオを作成します。

ポートフォリオを作成するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションパネルで [ポートフォリオ] を選択し、[ポートフォリオの作成] を選択します。
3. 以下の値を入力します。
   - ポートフォリオ名 - **`Engineering Tools`**
   - ポートフォリオの説明 — **`Sample portfolio that contains a single product.`**
   - 所有者 - **`IT (it@example.com)`**

4. [作成] を選択します。

ステップ 4: ポートフォリオに新しい製品を作成する

ポートフォリオを作成したら、ポートフォリオ内に製品を作成する準備が整います。このチュートリアルでは、エンジニアリングツール ポートフォリオ内に、Amazon Linux 上で実行されるクラウド開発環境である Linux Desktop という製品を作成します。

ポートフォリオ内に製品を作成するには

1. 前のステップを完了すると、[Portfolios] ページがすでに表示されています。それ以外の場合は、[https://console.aws.amazon.com/servicecatalog/] を開きます。
2. ステップ 2 で作成したエンジニアリングツールポートフォリオを選択して開きます。
3. [Upload new product] (新しい製品のアップロード) を選択します。
4. 「製品の作成」ページの「製品詳細」セクションで、以下を入力します。
   - [製品名] - **Linux Desktop**
   - 製品の説明 – **Cloud development environment configured for engineering staff. Runs AWS Linux.**
   - 所有者 - **IT**
   - [ディストリビューター] – (空白)
5. バージョンの詳細ページで、 CloudFormation テンプレートの使用を選択します。[Amazon S3 テンプレートの URL を指定する] を選択したら、次のように入力します。
   - [テンプレートの選択] - **https://awsdocs.s3.amazonaws.com/servicecatalog/development-environment.template**
   - [バージョンタイトル] - **v1.0**
   - [説明] - **Base Version**
6. [サポート詳細] セクションで、次のように入力します。
   - [メールの連絡先] - **ITSupport@example.com**
   - [サポートリンク] - **https://wiki.example.com/IT/support**
   - [サポートの説明] -**Contact the IT department for issues deploying or connecting to this product.**
7. [製品の作成] を選択します。

ステップ 5: テンプレート制約を追加してインスタンスサイズを制限する

制約により、ポートフォリオレベルでの製品の制御が強化されます。制約では、製品の起動コンテキストを制御 (起動制約) したり、 AWS CloudFormation テンプレートにルールを追加 (テンプレート制約) したりできます。詳細については、「AWS Service Catalog 制約の使用」を参照してください。

これで、起動時にラージインスタンスタイプをユーザーが選択できないようにするテンプレート制約を Linux Desktop 製品に追加できます。development-environment テンプレートにより、ユーザーは 6 つのインスタンスタイプから選択できます。この制約では、有効なインスタンスタイプを 2 つの最小タイプ (`t2.micro` および `t2.small`) に制限します。詳細については、「Amazon EC2 ユーザーガイド」の「T2 インスタンス」を参照してください。 Amazon EC2

テンプレート制約を Linux Desktop 製品を追加するには

1. ポートフォリオの詳細ページで、[制約] を選択し、[制約の作成] を選択します。
2. 「制約の作成」ページの「製品」で、「Linux Desktop」を選択します。次に、[制約タイプ]で、[テンプレート] を選択します。
3. 「テンプレート制約」セクションで、「テキストエディタ」を選択します。
4. テキストエディタに、以下の内容を貼り付けます。

```
{
  "Rules": {
    "Rule1": {
      "Assertions": [
        {
          "Assert" : {"Fn::Contains": [["t2.micro", "t2.small"], {"Ref":
 "InstanceType"}]},
          "AssertDescription": "Instance type should be t2.micro or t2.small"
        }
      ]
    }
  }
}
```

5. [制約の説明] に、**`Small instance sizes`** と入力します。
6. [作成] を選択します。

ステップ 6: IAM ロールを割り当てる起動制約を追加する

起動制約は、エンドユーザーが製品を起動するときに AWS Service Catalog が引き受ける IAM ロールを指定します。

このステップでは、製品の AWS CloudFormation テンプレートの一部である IAM リソースを AWS Service Catalog が使用できるよう、Linux Desktop 製品に起動制約を追加します。

起動制約として製品に割り当てる IAM ロールには、以下のアクセス権限が必要です。

1. AWS CloudFormation
2. 製品用の AWS CloudFormation テンプレートのサービス。
3. サービス所有の Amazon S3 バケット内の AWS CloudFormation テンプレートへの読み取りアクセス。

この起動制約により、エンドユーザーは製品を起動し、起動後にそれをプロビジョニング済み製品として管理できるようになります。詳細については、「AWS Service Catalog 起動の設定」を参照してください。

起動制約がない場合、エンドユーザーが Linux Desktop 製品を使用するには、事前に追加の IAM アクセス権限をエンドユーザーに付与する必要があります。たとえば、`ServiceCatalogEndUserAccess` ポリシーにより、 AWS Service Catalog エンドユーザーコンソールビューにアクセスするために必要な最小の IAM アクセス権限が付与されます。

起動制約を使用すると、エンドユーザーの IAM 権限を最小限に抑えるという IAM のベストプラクティスに従うことができます。詳細については、IAM ユーザーガイドの「最小特権を認める」を参照してください。

起動の制約を追加するには

1. IAM ユーザーガイド の「JSON タブにある新しいポリシーを作成する」の指示に従ってください。
2. 以下の JSON ポリシードキュメントを貼り付けます。

   - `cloudformation`— AWS CloudFormation スタックを作成、読み取り、更新、削除、一覧表示、タグ付けするためのすべての権限が AWS Service Catalog に付与されます。
   - `ec2`— AWS Service Catalog 製品の一部である Amazon Elastic Compute Cloud (Amazon EC2) リソースの一覧表示、読み取り、書き込み、プロビジョニング、タグ付けを行うすべて

の権限を AWS Service Catalog に付与します。デプロイする AWS リソースによっては、この権限が変わる場合があります。

- ec2– AWS アカウントの新しい管理ポリシーを作成し、指定した IAM ロールに指定した管理ポリシーをアタッチします。
- s3— AWS Service Catalog が所有する Amazon S3 バケットへのアクセスを許可します。製品をデプロイするには、AWS Service Catalog はプロビジョニングアーティファクトへのアクセスが必要です。
- servicecatalog— エンドユーザーに代わってリソースの一覧表示、読み取り、書き込み、タグ付け、起動を行う AWS Service Catalog 権限を許可します。
- sns— 起動制約の対象となる Amazon SNS トピックの一覧表示、読み取り、書き込み、およびタグ付けを行う AWS Service Catalog 権限を許可します。

> **Note**
>
> デプロイする基盤となるリソースによっては、JSON ポリシーの例を変更する必要が生じることがあります。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStackEvents",
                "cloudformation:DescribeStacks",
                "cloudformation:GetTemplateSummary",
                "cloudformation:SetStackPolicy",
                "cloudformation:ValidateTemplate",
                "cloudformation:UpdateStack",
                "ec2:*",
                "servicecatalog:*",
                "sns:*"
            ],
            "Resource": "*"
        },
```

```
        {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "s3:ExistingObjectTag/servicecatalog:provisioning":"true"
            }
         }
      }
   ]
}
```

3. 次へ、タグ を選択します。
4. 次へ、レビューを選択します。
5. [ポリシーの確認] ページで、[名前] に「**linuxDesktopPolicy**」と入力します。
6. [ポリシーの作成] を選択します。
7. ナビゲーションペインで [ロール] を選択します。次に、[ロールの作成] を選択して、次の操作を行います。

   a. [信頼されたエンティティの選択] で [AWSサービス] を選択し、[その他の AWS サービスのユースケース] で [Service Catalog] を選択します。Service Catalog のユースケースを選択し、[次へ] を選択します。

   b. linuxDesktopPolicy ポリシーを検索し、チェックボックスをオンにします。

   c. [次へ] を選択します。

   d. [Role name](ロール名) に **linuxDesktopLaunchRole** を入力します。

   e. [Create role] (ロールの作成) を選択します。
8. AWS Service Catalog コンソールを開きます (https://console.aws.amazon.com/servicecatalog)。
9. [Engineering Tools] ポートフォリオを選択します。
10. ポートフォリオの詳細ページで、[制約] タブを選択し、[制約の作成] を選択します。
11. [製品]で、[Linux Desktop] を選択し、[制約タイプ]で、[起動] を選択します。
12. [IAM ロールの選択] を選択します。次に、linuxDesktopLaunchロール を選択し、作成 を選択します。

ステップ 7: ポートフォリオへのアクセス権限のエンドユーザーへの付与

これでポートフォリオを作成し、製品を追加したので、エンドユーザーにアクセス権限を付与することができます。

前提条件

エンドユーザーの IAM グループを作成していない場合は、「AWS Service Catalog エンドユーザーへのアクセス権限の付与」を参照してください。

ポートフォリオへのアクセス権限を提供するには

1. ポートフォリオの詳細ページで、[アクセス] タブを選択します。
2. [アクセス権の付与] を選択します。
3. [グループ] タブで、エンドユーザーの IAM グループのチェックボックスをオンにします。
4. [プロセスの追加] を選択します。

ステップ 8: エンドユーザーのエクスペリエンスをテストする

エンドユーザーが正常にエンドユーザーコンソールビューにアクセスし、製品を起動できることを確認するには、AWS にエンドユーザーとしてサインインしてこれらのタスクを実行します。

エンドユーザーがエンドユーザーコンソールにアクセスできることを確認するには

1. 手順については、「IAM ユーザーガイド」の「IAM ユーザーでサインイン」を参照してください。
2. メニューバーで、`Engineering Tools` ポートフォリオを作成した AWS リージョンを選択します。このチュートリアルでは、[us-east-1 リージョン] を選択します。
3. https://console.aws.amazon.com/servicecatalog/ で AWS Service Catalog コンソールを開いて、以下を確認します。

   - [製品] - ユーザーが使用できる製品。
   - [プロビジョニング済み製品] - ユーザーが起動したプロビジョニング済み製品。

エンドユーザーが Linux Desktop デスクトップ製品を起動できることを検証するには

このチュートリアルでは、[us-east-1 リージョン] を選択していることに注意してください。

1. コンソールの [製品] セクションで [Linux Desktop] を選択します。
2. [製品の起動] をクリックして、製品を構成するウィザードを開始します。
3. [起動: Linux Desktop] ページで、[プロビジョニング済み製品名] に「**Linux-Desktop**」と記入します。
4. [パラメータ] ページで、以下を入力し、[次へ] をクリックします。
   - [サーバーサイズ] - **t2.micro** を選択します。
   - [キーペア] - ステップ 2: キーペアを作成する で作成したキーペアを選択します。
   - [CIDR 範囲] - インスタンスへの接続する IP アドレスの有効な CIDR 範囲を入力します。任意の IP アドレスからのアクセスを許可するデフォルト値 (0.0.0.0/0) 、または自分の IP アドレスに **/32** を追加して自分の IP アドレスのみにアクセスを制限するか、またはその中間とすることができます。
5. [製品の起動] を選択してスタックを起動します。コンソールには Linux Desktop のスタックのスタック詳細ページが表示されます。製品の最初のステータスは [変更中] です。AWS Service Catalog が製品を起動するには数分間かかります。現在のステータスを表示するには、ブラウザを更新してください。製品が起動すると、ステータスは [Available] になります。

Terraform 製品の使用開始

AWS Service Catalog は、 内の HashiCorp Terraform 設定のガバナンスによる迅速なセルフサービスプロビジョニングを可能にしますAWS。AWS Service Catalog を単一のツールとして使用すると、AWS 内で Terraform 設定を大規模に整理、管理、配布できます。AWS Service Catalog は、標準化および事前承認された Terraform テンプレートのカタログ化、アクセス制御、バージョニング、タグ付け、他の AWS アカウントとの共有など、いくつかの主要な機能にわたって Terraform をサポートします。AWS Service Catalog では、エンドユーザーがアクセスできる製品とバージョンの簡単なリストが表示され、それらの製品を 1 回の操作でデプロイできます。

> **Note**
>
> 最近の Terraform へのライセンス変更の結果、 HashiCorp テクノロジーのサポートを継続するために、 は Terraform Open Source の以前の参照を外部 AWS Service Catalogに変更しました。External 製品タイプには、以前は Terraform オープンソースとして知られていた Terraform Community Edition のサポートが含まれます。既存の Terraform オープンソース製品およびプロビジョニング済み製品を外部製品タイプに移行する方法の詳細と手順について

は、既存の Terraform Open Source 製品およびプロビジョニング済み製品の外部製品タイプへの更新 をご覧ください。

以下のチュートリアルの手順は、AWS Service Catalog で Terraform 製品を使い始めるのに役立ちます。

カタログ管理者は、中央管理者アカウント (ハブアカウント) で作業します。Terrafrm Community Edition 製品と Terraform Cloud 製品のどちらにも Terraform プロビジョニングエンジンが必要です。詳細については、「Terraform Community Edition (External 製品タイプ) のプロビジョニングエンジン」と「Terraform クCloud 用のプロビジョニングエンジン 」を参照してください。

チュートリアルでは、管理者アカウントで以下のタスクを実行します。

- Terraform Cloud または External 製品タイプのいずれかを使用して Terraform 製品を作成します。Service Catalog は、[External] 製品タイプを使用して Terraform Community Edition 製品をサポートします。
- 製品をポートフォリオに関連付けます。
- エンドユーザーが製品をプロビジョニングできるように起動制約を作成します。
- 製品へのタグ
- ポートフォリオと Terraform 製品をエンドユーザーアカウント (スポークアカウント) と共有します。

チュートリアルでは、組織の管理アカウントでもある管理ハブアカウントから、組織共有オプションを使用してポートフォリオを共有します。組織共有の詳細については、「ポートフォリオの共有」を参照してください。

チュートリアルで作成した Terraform 製品に含まれる AWS リソースは、シンプルな Amazon S3 バケットです。

> **Note**
>
> 開始する前に、必ず AWS Service Catalogのセットアップ のアクション項目を完了していることを確認してください。

トピック

- 既存の Terraform Open Source 製品およびプロビジョニング済み製品の外部製品タイプへの更新

- 前提条件: Terraform プロビジョニングエンジンの設定
- ステップ 1: Terraform 設定ファイルをダウンロードする
- ステップ 2: Terraform 製品を作成する
- ステップ 3: AWS Service Catalog ポートフォリオを作成する
- ステップ 4: ポートフォリオに製品を追加する
- ステップ 5: 起動ロールを作成する
- ステップ 6: Terraform 製品に起動制約を追加する
- ステップ 7: エンドユーザーにアクセス許可を付与する
- ステップ 8: ポートフォリオをエンドユーザーと共有する
- ステップ 9: エンドユーザーのエクスペリエンスをテストする
- ステップ 10: Terraform プロビジョニング操作の監視

既存の Terraform Open Source 製品およびプロビジョニング済み製品の外部製品タイプへの更新

最近の Terraform へのライセンス変更の結果、 HashiCorp はテクノロジーのサポートを継続するために、Terraform Open Source の以前の参照を外部 AWS Service Catalogに変更しました。External 製品タイプには、以前は Terraform オープンソースとして知られていた Terraform Community Edition のサポートが含まれます。AWS Service Catalog は、「新規」製品またはプロビジョニングされた製品の有効な製品タイプとして Terraform オープンソースをサポートしなくなりました。実行できる操作は、製品バージョンとプロビジョニング済み製品を始めとする既存の Terraform オープンソースリソースの更新または終了のみです。

まだ移行していない場合は、このセクションの指示に従って、既存の Terraform オープンソース製品とプロビジョニング済み製品をすべて External 製品に移行する必要があります。

1. AWS Service Catalog の既存の Terraform Reference Engine を更新して、外部製品タイプと Terraform Open Source 製品タイプの両方をサポートするようにしてください。Terraform リファレンスエンジンの更新手順については、GitHub リポジトリ を参照してください。
2. 新しい外部製品タイプを使用して、既存の Terraform Open Source 製品を再作成します。
3. Terraform Open Source 製品タイプを使用する既存の製品をすべて削除します。
4. 新しい External 製品タイプを使用して、残りのリソースを再プロビジョニングします。
5. Terraform Open Source 製品タイプを使用する既存のプロビジョニング済み製品をすべて終了します。

既存の製品を移行した後は、tar.gz 設定ファイルを使用する新しい製品に External 製品タイプを使用してください。

AWS Service Catalog は、必要に応じて、この変更を通してお客様をサポートします。これらの変更によってお客様のアカウントに多大な労力が必要になる場合や、重要な製品ワークロードに影響が及ぶ場合は、アカウント担当者に連絡してサポートを依頼してください。

前提条件: Terraform プロビジョニングエンジンの設定

AWS Service Catalog で Terraform 製品を作成するための前提条件として、Service Catalog 管理者アカウント (ハブアカウント) にプロビジョニングエンジンをインストールして設定する必要があります。プロビジョニングエンジンは、Terraform Community Edition 製品 (External 製品タイプを使用) と Terraform Cloud 製品 (Terraform Cloud 製品タイプを使用) の両方に必要です。

> **Note**
>
> エンジン設定は 1 回限りの設定で、約 30 分かかります。

Terraform Community Edition (External 製品タイプ) のプロビジョニングエンジン

AWS Service Catalog は、External 製品タイプを使用して Terraform Community Edition 製品をサポートします。External 製品タイプは、プロビジョニングエンジンの設定に基づいて、Pulumi、Ansible、Chef などの他のプロビジョニングツールもサポートします。

HashiCorpの Terraform Community Edition で External 製品タイプを使用するAWS Service Catalog 製品の場合、AWS Service Catalog管理者アカウント (ハブアカウント) に Terraform プロビジョニングエンジンをインストールして設定する必要があります。 はこのエンジンとそのリソースAWSを管理します。

AWS Service Catalog は、 AWSが提供する Terraform プロビジョニングエンジン をインストールして設定する手順を含む GitHub リポジトリを提供します。リポジトリには次の情報が含まれています。

- 必要なインストールツール
- コードの構築
- AWS アカウントへのデプロイ
- プロビジョニングワークフロー、品質保証、制限に関する追加情報

Terraform クCloud 用のプロビジョニングエンジン

Terraform Cloud 製品タイプを HashiCorpの Terraform Cloud で使用するAWS Service Catalog製品の場合は、AWS Service Catalog管理者アカウント (ハブアカウント) に Terraform プロビジョニングエンジンをインストールして設定する必要があります。 HashiCorp は、このエンジンをリモート環境で管理します。

HashiCorp は、 用の Terraform Cloud エンジンAWS Service Catalogの設定手順を含む GitHub リポジトリを提供します。リポジトリには次の情報が含まれています。

- 必要なインストールツール
- コードの構築
- AWS アカウントへのデプロイ
- プロビジョニングワークフロー、品質保証、制限に関する追加情報

ステップ 1: Terraform 設定ファイルをダウンロードする

Terraform 設定ファイルを使用して、 HashiCorp Terraform 製品を作成およびプロビジョニングできます。これらの設定はプレーンテキストファイルで、プロビジョニングするリソースを記述します。任意のテキストエディターを使用して、設定を作成、更新、保存できます。プロダクトを作成するには、Terraform 設定を tar.gz ファイルとしてアップロードする必要があります。このチュートリアルでは、AWS Service Catalog がすぐに始められるように簡単な設定ファイルを用意しています。この設定により、Amazon S3 バケットが作成されます。

設定ファイルのダウンロード

AWS Service Catalog は、このチュートリアルで使用できるサンプル `simple-s3-bucket.tar.gz` 設定ファイルを提供します。

設定ファイルの概要

サンプル設定ファイルのテキストは次のとおりです。

```
variable "bucket_name" {
  type = string
}
provider "aws" {
}
```

```
resource "aws_s3_bucket" "bucket" {
  bucket = var.bucket_name
}
output regional_domain_name {
  value = aws_s3_bucket.bucket.bucket_regional_domain_name
}
```

リソース設定

設定ファイルには、AWS Service Catalog 製品のプロビジョニング時に作成されるリソースが宣言されています。次のセクションがあります。

- 変数 (オプション) — 管理者ユーザー (ハブアカウント管理者) が設定をカスタマイズするために割り当てることができる値の定義。変数は、特定の構成の動作を変更するための一貫したインターフェースを提供します。変数キーワードの後のラベルは変数の名前であり、同じモジュール内のすべての変数の中で一意である必要があります。この名前は、変数に外部値を割り当てたり、モジュール内から変数の値を参照したりするために使用されます。
- プロバイダー (オプション) — リソースをプロビジョニングするクラウドサービスプロバイダーAWS。 AWS Service Catalog は、AWS プロバイダーとしてのみサポートします。その結果、Terraform プロビジョニングエンジンは、リストされている他のプロバイダーを AWS にオーバーライドします。
- リソース (必須) — プロビジョニング用の AWS インフラストラクチャリソース。このチュートリアルでは、Terraform 設定ファイルで Amazon S3 を指定しています。
- 出力 (オプション) — プログラミング言語の戻り値と同様の、返される情報または値。出力データを使用して、自動化ツールでインフラストラクチャワークフローを設定できます。

ステップ 2: Terraform 製品を作成する

Terraform プロビジョニングエンジンをインストールしたら、 で HashiCorp Terraform 製品を作成する準備が整いますAWS Service Catalog。このチュートリアルでは、シンプルな Amazon S3 バケットを含む Terraform 製品を作成します。

新しい Terraform 製品を作成するには

1. https://console.aws.amazon.com/servicecatalog/ で AWS Service Catalog コンソールを開き、管理者ユーザーとしてサインインします。
2. 「管理」セクションに移動し、「製品リスト」を選択します。

3. [製品の作成] を選択します。
4. [製品の詳細] セクションの [製品の作成] ページで、[External]または [Terraform Cloud] の製品タイプを選択します。Service Catalog は、External 製品タイプを使用して Terraform Community Edition 製品をサポートします。
5. 次の製品の詳細を入力します。

   - [製品名] - **`Simple S3 bucket`**
   - [製品説明] — Amazon S3 バケットを含む Terraform 製品。
   - 所有者 - **`IT`**
   - [ディストリビューター] – (空白)
6. [バージョンの詳細] ペインで、テンプレートファイルのアップロード を選択し、[ファイルの選択] を選択します。ステップ 1: Terraform 設定ファイルをダウンロードする でダウンロードしたファイルを選択します。
7. 次のように入力します。

   - バージョン名 – **`v1.0`**
   - バージョンの説明 – **`Base Version`**
8. [サポートの詳細] セクションで、以下を入力し、[製品の作成] を選択します。

   - [メールの連絡先] - **`ITSupport@example.com`**
   - [サポートリンク] - **`https://wiki.example.com/IT/support`**
   - [サポートの説明] -**`Contact the IT department for issues deploying or connecting to this product.`**
9. [製品の作成] を選択します。

製品が正常に作成されると、AWS Service Catalog は製品ページに確認バナーを表示します。

ステップ 3: AWS Service Catalog ポートフォリオを作成する

AWS Service Catalog 管理者アカウント (ハブアカウント) にポートフォリオを作成すると、製品の整理とエンドユーザーアカウント (スポークアカウント) への配布が容易になります。

ポートフォリオを作成するには

1. https://console.aws.amazon.com/servicecatalog/ で AWS Service Catalog コンソールを開き、管理者としてサインインします。

2. 左側のナビゲーションパネルで [ポートフォリオ] を選択し、[ポートフォリオの作成] を選択します。
3. 次の値を入力します。
   - ポートフォリオ名 - **S3 bucket**
   - ポートフォリオの説明 — **Sample portfolio for Terraform configurations.**
   - 所有者 - **IT (it@example.com)**
4. [作成] を選択します。

ステップ 4: ポートフォリオに製品を追加する

ポートフォリオを作成したら、ステップ 2 で作成した HashiCorp Terraform 製品を追加できます。

製品をポートフォリオに追加するには

1. [製品リスト] ページに移動します。
2. ステップ 2 で作成したシンプルな S3 バケット Terraform 製品を選択し、[アクション] を選択します。ドロップダウンメニューから [製品をポートフォリオに追加] を選択します。 AWS Service Catalog に、[シンプルな S3 バケットをポートフォリオに追加] ペインが表示されます。
3. S3 バケットポートフォリオを選択し、[起動制約の作成] をオフにします。起動制約は、このチュートリアルの後半で作成します。
4. [製品をポートフォリオに追加] を選択します。

製品をポートフォリオに正常に追加すると、AWS Service Catalog は、製品リストページに確認バナーを表示します。

ステップ 5: 起動ロールを作成する

このステップでは、エンドユーザーが Terraform HashiCorp 製品を起動するときに Terraform プロビジョニングエンジンと が引き受けAWS Service Catalogることができるアクセス許可を指定する IAM ロール (起動ロール) を作成します。

起動制約としてシンプルな Amazon S3 バケット Terraform 製品に後で割り当てる IAM ロール (起動ロール) には、以下のアクセス許可が必要です。

- Terraform 製品の基盤となる AWS リソースへのアクセス。このチュートリアルでは、`s3:CreateBucket*`、`s3:DeleteBucket*`、`s3:Get*`、`s3:List*`、および `s3:PutBucketTagging` Amazon S3 オペレーションへのアクセスが含まれます。
- AWS Service Catalog が所有する Amazon S3 バケット内の Amazon S3 テンプレートへの読み取りアクセス
- `CreateGroup`、`ListGroupResources`、`DeleteGroup`、および `Tag` リソースグループオペレーションへのアクセス。これらの操作により、AWS Service Catalog はリソースグループとタグを管理できます。

AWS Service Catalog 管理者アカウントで起動ロールを作成するには

1. AWS Service Catalog 管理者アカウントにログインした状態で、IAM ユーザーガイドの「JSON タブにある新しいポリシーの作成」の指示に従います。
2. シンプルな Amazon S3 バケット Terraform 製品用のポリシーを作成します。このポリシーは、起動ロールを作成する前に作成する必要があり、以下の権限で構成されます。

   - `s3`— Amazon S3 製品の一覧表示、読み取り、書き込み、プロビジョニング、タグ付けを行うための完全な権限を AWS Service Catalog に付与します。
   - `s3`— AWS Service Catalog が所有する Amazon S3 バケットへのアクセスを許可します。製品をデプロイするには、AWS Service Catalog はプロビジョニングアーティファクトへのアクセスが必要です。
   - `resourcegroups`— AWS Service Catalog は、タグ AWS Resource Groups の作成、一覧表示、削除を行えるようになります。
   - `tag`— タグ付けの権限を AWS Service Catalog に付与します。

   > **Note**
   >
   > デプロイする基盤となるリソースによっては、サンプル JSON ポリシーを変更する必要がある場合があります。

   以下の JSON ポリシードキュメントを貼り付けます。

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

```
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/servicecatalog:provisioning": "true"
                }
            }
        },
        {
            "Action": [
                "s3:CreateBucket*",
                "s3:DeleteBucket*",
                "s3:Get*",
                "s3:List*",
                "s3:PutBucketTagging"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:ListGroupResources",
                "resource-groups:DeleteGroup",
                "resource-groups:Tag"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "tag:GetResources",
                "tag:GetTagKeys",
                "tag:GetTagValues",
                "tag:TagResources",
                "tag:UntagResources"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
```

```
}
```

3. a. 次へ、タグを選択します。

   b. 次へ、レビューを選択します。

   c. [ポリシーの確認] ページで、[名前] に「**S3ResourceCreationAndArtifactAccessPolicy**」と入力します。

   d. [Create policy] を選択します。

4. ナビゲーションペインで [Roles] を選択し、続いて [Create role] を選択します。

5. [信頼できるエンティティを選択] で [カスタム信頼ポリシー] を選択し、次の JSON ポリシーを入力します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GivePermissionsToServiceCatalog",
            "Effect": "Allow",
            "Principal": {
                "Service": "servicecatalog.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account_id:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringLike": {
                    "aws:PrincipalArn": [
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
TerraformExecutionRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogExternalParameterParserRole*",
                        "arn:aws:iam::accounti_id:role/TerraformEngine/
ServiceCatalogTerraformOSParameterParserRole*"
                    ]
                }
            }
        }
```

```
    ]
}
```

6. [次へ] を選択します。
7. 「ポリシー」リストで、作成したばかりの S3ResourceCreationAndArtifactAccessPolicy を選択します。
8. [次へ] を選択します。
9. [ロール名] には、「**SCLaunch-S3product**」と入力します。

> **Important**
>
> 起動ロール名は「SCLaunch」で始まり、その後に目的のロール名が続く必要があります。

10. ロールの作成 を選択します。

> **Important**
>
> AWS Service Catalog 管理者アカウントで起動ロールを作成したら、AWS Service Catalog エンドユーザーアカウントでも同じ起動ロールを作成する必要があります。エンドユーザーアカウントのロールは、管理者アカウントのロールと同じ名前で、同じポリシーが含まれている必要があります。

AWS Service Catalog エンドユーザーアカウントに起動ロールを作成するには

1. エンドユーザーアカウントに管理者としてログインし、IAM ユーザーガイドの JSON タブにある新しいポリシーの作成の指示に従います。
2. 上記の「AWS Service Catalog 管理者アカウントで起動ロールを作成するには」の手順 2 ~ 10 を繰り返します。

> **Note**
>
> AWS Service Catalog エンドユーザーアカウントで起動ロールを作成するときは、カスタム信頼ポリシーでも同じ管理者 **AccountId** を使用するようにしてください。

管理者アカウントとエンドユーザーアカウントの両方で起動ロールを作成したので、製品に起動制約を追加できます。

ステップ 6: Terraform 製品に起動制約を追加する

> **Important**
>
> HashiCorp Terraform 製品の起動制約を作成する必要があります。起動制約がない場合、エンドユーザーは製品をプロビジョニングできません。

管理者アカウントで起動ロールを作成したら、その起動ロールを External または Terraform Cloud 製品の起動制約に関連付けることができます。

この起動制約により、エンドユーザーは製品を起動し、起動後にそれをプロビジョニング済み製品として管理できるようになります。詳細については、「AWS Service Catalog 起動の設定」を参照してください。

起動制約を使用すると、エンドユーザーの IAM 権限を最小限に抑えるという IAM のベストプラクティスに従うことができます。詳細については、IAM ユーザーガイドの「最小特権を認める」を参照してください。

起動制限を製品に割り当てるには

1. AWS Service Catalog コンソールを開きます (https://console.aws.amazon.com/servicecatalog)。
2. 左側のナビゲーションコンソールから [ポートフォリオ] を選択します。
3. S3 バケットポートフォリオを選択します。
4. ポートフォリオの詳細ページで、[制約] タブを選択し、[制約の作成] を選択します。
5. [製品] には [シンプル S3 バケット] を選択します。AWS Service Catalog は 起動制約タイプを自動的に選択します。
6. [ロール名を入力] を選択し、[SCLaunch-S3product] を選択します。
7. [作成] を選択します。

> **Note**
>
> 指定されたロール名は、起動制約を作成したアカウントと、この起動制約を使用して製品を起動するユーザーのアカウントに存在している必要があります。

ステップ 7: エンドユーザーにアクセス許可を付与する

HashiCorp Terraform 製品に起動制約を適用したら、スポークアカウントのエンドユーザーにアクセス許可を付与する準備が整います。

このチュートリアルでは、プリンシパル名共有を使用してエンドユーザーにアクセス許可を付与します。プリンシパル名は、管理者がポートフォリオ内で指定してポートフォリオと共有できるグループ、ロール、およびユーザーの名前です。ポートフォリオを共有するときに、AWS Service Catalog は、それらのプリンシパル名がすでに存在するかどうかを確認します。存在する場合は、AWS Service Catalog は、一致する IAM プリンシパルを共有ポートフォリオに自動的に関連付けて、エンドユーザーにアクセス権を付与します。詳細については、「ポートフォリオの共有」を参照してください。

前提条件

エンドユーザーの IAM グループを作成していない場合は、「AWS Service Catalog エンドユーザーへのアクセス権限の付与」を参照してください。

ポートフォリオへのアクセス権限を提供するには

1. ポートフォリオページに移動し、S3 バケットポートフォリオを選択します。
2. [アクセス] タブを選択し、[アクセス権の付与] を選択します。
3. 「アクセスタイプ」ペインで、「プリンシパル名」を選択します。
4. プリンシパル名ペインで、プリンシパル名タイプを選択し、スポークアカウント内の目的のエンドユーザーのプリンシパル名を入力します。
5. [アクセス権の付与] を選択します。

ステップ 8: ポートフォリオをエンドユーザーと共有する

AWS Service Catalog 管理者は、共有またはAWS Organizations共有を使用して account-to-account 、ポートフォリオをエンドユーザーアカウントに配布できます。このチュートリアルでは、組織の管理アカウントでもある管理者アカウント (ハブアカウント) からポートフォリオを組織と共有します。

管理ハブアカウントからポートフォリオを共有するには

1. AWS Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。

2. ポートフォリオ ページで S3 バケットポートフォリオを選択します。[アクション] メニューで [共有] を選択します。
3. AWS Organizations を選択し、組織構造に絞り込みます。
4. AWS組織ペインで、エンドユーザーアカウント (スポークアカウント) を選択します。

   ルートノード を選択して、組織構造に基づいて、組織全体、組織内の親組織単位 (OU)、または子 OU とポートフォリオを共有することもできます。詳細については、「ポートフォリオの共有 」を参照してください。
5. 共有設定ペインで、「プリンシパル共有」を選択します。
6. [共有] を選択します。

ポートフォリオをエンドユーザーと共有できたら、次のステップはエンドユーザーエクスペリエンスを検証し、Terraform 製品をプロビジョニングすることです。

ステップ 9: エンドユーザーのエクスペリエンスをテストする

エンドユーザーが正常にエンドユーザーコンソールビューにアクセスし、**`Simple S3 bucket`** 製品を起動できることを確認するには、AWS にエンドユーザーとしてサインインしてこれらのタスクを実行します。

エンドユーザーがエンドユーザーコンソールにアクセスできることを確認するには

- https://console.aws.amazon.com/servicecatalog/ で AWS Service Catalog コンソールを開いて、以下を確認します。
  - [製品] - ユーザーが使用できる製品。
  - [プロビジョニング済み製品] - ユーザーが起動したプロビジョニング済み製品。

エンドユーザーが Terraform 製品を起動できることを確認するには

1. コンソールの [製品] セクションで、[シンプル S3 バケット] を選択します。
2. [製品の起動] をクリックして、製品を構成するウィザードを開始します。
3. 「シンプル S3 バケットの起動」ページで、プロビジョニングされた製品名を入力 **`Amazon S3 product`** します。
4. [パラメータ] ページで、以下を入力し、[次へ] をクリックします。

- [bucket_name] — Amazon S3 バケットの一意の名前を入力します。例えば、「**terraform-s3-product**」と入力します。

5. [製品の起動] を選択します。コンソールには Amazon S3 製品の起動に関するスタックの詳細ページが表示されます。製品の最初のステータスは [変更中] です。AWS Service Catalog が製品を起動するには数分間かかります。現在のステータスを表示するには、ブラウザを更新してください。製品の起動に成功すると、ステータスは [利用可能] になります。

   AWS Service Catalog は、**terraform-s3-product** という名前の新しい Amazon S3 バケットを作成します。

ステップ 10: Terraform プロビジョニング操作の監視

プロビジョニングオペレーションをモニタリングする場合は、Amazon CloudWatch ログと AWS Step Functions でプロビジョニングワークフローを確認できます。

プロビジョニングワークフローには次の 2 つのステートマシンがあります。

- `ManageProvisionedProductStateMachine` — AWS Service Catalog は、新しい Terraform 製品をプロビジョニングするとき、および既存の Terraform プロビジョニング済み製品を更新するときに、このステートマシンを呼び出します。
- `TerminateProvisionedProductStateMachine`— AWS Service Catalog は、既存の Terraform プロビジョニング済み製品を終了するときに、このステートマシンを呼び出します。

モニタリングステートマシンを実行するには

1. AWS 管理コンソールを開き、Terraform プロビジョニングエンジンがインストールされている管理ハブアカウントに管理者としてログインします。
2. AWS Step Functions を開きます。
3. 左側のナビゲーションパネルで、ステートマシンを選択します。
4. を選択しますManageProvisionedProductStateMachine。
5. 「実行」リストに、プロビジョニングされた製品 ID を入力して実行場所を特定します。

> **Note**
>
> AWS Service Catalog は、製品をプロビジョニングするときに、プロビジョニングされた製品 ID を作成します。プロビジョニングされた製品 ID の形式は次のとおりです。**pp-1111pwtn[ID number]**

6. 実行 ID を選択します。

表示される [実行詳細] ページでは、プロビジョニングワークフローのすべてのステップを確認できます。障害が発生した手順を確認して、障害の原因を特定します。

のセキュリティ AWS Service Catalog

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャからメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティとして説明しています。

- クラウドのセキュリティ — AWS クラウドで AWS サービスを実行するインフラストラクチャを保護する AWS 責任があります。 AWS また、 では、安全に使用できるサービスも提供しています。サードパーティーの監査人は、AWS コンプライアンスプログラム の一環として、セキュリティの有効性を定期的にテストおよび検証します。

  に適用されるコンプライアンスプログラムの詳細については AWS Service Catalog、「コンプライアンスAWS プログラムによる対象範囲内のサービス」を参照してください。
- クラウドのセキュリティ — お客様の責任は、使用する AWS サービスによって決まります。また、お客様は、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、 の使用時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 AWS Service Catalog。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ために AWS Service Catalog を設定する方法を示します。また、 AWS Service Catalog リソースのモニタリングや保護に役立つ他の AWS のサービスについても説明します。

トピック

- でのデータ保護 AWS Service Catalog
- AWS Service Catalogにおけるアイデンティティとアクセスの管理
- での口グ記録とモニタリング AWS Service Catalog
- のコンプライアンス検証 AWS Service Catalog
- の耐障害性 AWS Service Catalog
- のインフラストラクチャセキュリティ AWS Service Catalog
- のセキュリティのベストプラクティス AWS Service Catalog

でのデータ保護 AWS Service Catalog

責任 AWS 共有モデル、 でのデータ保護に適用されます AWS Service Catalog。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービス のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、データプライバシーのよくある質問を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿されたAWS 責任共有モデルおよび GDPRのブログ記事を参照してください。

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

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。
- AWS 暗号化ソリューションと、 内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

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

暗号化によるデータの保護

保管中の暗号化

AWS Service Catalog は、Amazon が管理するキーを使用して保管時に暗号化された Amazon S3 バケットと Amazon DynamoDB データベースを使用します。詳細については、Amazon S3 および Amazon DynamoDB で提供される保管時の暗号化に関する情報を参照してください。

転送中の暗号化

AWS Service Catalog は、Transport Layer Security (TLS) と、発信者と の間で転送中の情報のクライアント側の暗号化を使用します AWS。

VPC エンドポイントを作成することで、Amazon Virtual Private Cloud (Amazon VPC) から AWS Service Catalog APIs にプライベートにアクセスできます。VPC エンドポイントでは、VPC と 間のルーティング AWS Service Catalog は AWS 、インターネットゲートウェイ、NAT ゲートウェイ、または VPN 接続を必要とせずにネットワークによって処理されます。

で使用される最新世代の VPC エンドポイント AWS Service Catalog は AWS PrivateLink、 AWS VPC 内のプライベート IPs で Elastic Network Interface を使用するサービス間の AWS プライベート接続を可能にするテクノロジーである VPCs を利用しています。

AWS Service Catalogにおけるアイデンティティとアクセスの管理

へのアクセスには認証情報 AWS Service Catalog が必要です。これらの認証情報には、 AWS Service Catalog ポートフォリオや product. AWS Service Catalog integrates with AWS Identity and Access Management (IAM) などの AWS リソースにアクセスするためのアクセス許可が必要です。これにより、製品の作成と管理に必要なアクセス許可を AWS Service Catalog 管理者に付与したり、製品の起動とプロビジョニング済み製品の管理に必要なアクセス許可を AWS Service Catalog エンドユーザーに付与したりできます。これらのポリシーは、 によって作成および管理 AWS されるか、管理者とエンドユーザーによって個別に管理されます。アクセスを制御するには、 ユーザー、グループ、および AWS Service Catalogで使用するロールに、これらのポリシーをアタッチします。

対象者

AWS Identity and Access Management (IAM) で持つ権限は、 AWS Service Catalogで果たすロールによって異なります。

AWS Identity and Access Management (IAM) を介して持つ権限は、 AWS Service Catalogで果たすロールによって異なる場合もあります。

管理者 - AWS Service Catalog 管理者には、管理者コンソールへのフルアクセスと、ポートフォリオと製品の作成と管理、制約の管理、エンドユーザーへのアクセス許可の付与などのタスクを実行できるようにする IAM アクセス許可が必要です。

エンドユーザー - エンドユーザーが製品を使用する前に、 AWS Service Catalog エンドユーザーコンソールへのアクセスを許可するアクセス許可を付与する必要があります。また、製品を起動し、プロビジョニング済み製品を管理する権限も付与できます。

IAM 管理者 - 管理者は、 AWS Service Catalogへのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる AWS Service Catalog アイデンティティベースのポリシーの例を表示するには、「」を参照してくださいthe section called “AWS マネージドポリシー”。

のアイデンティティベースのポリシーの例 AWS Service Catalog

トピック

- エンドユーザーのコンソールアクセス
- エンドユーザー向け製品アクセス
- プロビジョニング済み製品を管理するためのポリシーの例

エンドユーザーのコンソールアクセス

AWSServiceCatalogEndUserFullAccess および **AWSServiceCatalogEndUserReadOnlyAccess** ポリシーにより、 AWS Service Catalog エンドユーザーコンソールビューへのアクセス権が付与されます。これらのポリシーのいずれかを持つユーザーが AWS Service Catalog で を選択すると AWS Management Console、エンドユーザーコンソールビューに起動するアクセス許可を持つ製品が表示されます。

エンドユーザーがアクセス権を付与 AWS Service Catalog する製品を起動するには、製品の AWS CloudFormation テンプレート内の基盤となる各 AWS リソースの使用を許可する追加の IAM アクセス許可を付与する必要があります。例えば、製品テンプレートに Amazon Relational Database Service (Amazon RDS) が含まれる場合、製品を起動する Amazon RDS アクセス許可をユーザーに付与する必要があります。

エンドユーザーが AWS リソースへの最小アクセス許可を適用しながら製品を起動できるようにする方法については、「」を参照してくださいthe section called “ 制約の使用”。

AWSServiceCatalogEndUserReadOnlyAccess ポリシーを適用すると、ユーザーはエンドユーザーコンソールにアクセスできますが、製品を起動し、プロビジョニング済み製品を管理するために必要なアクセス権限は与えられません。IAM を使用してこれらのアクセス許可をエンドユーザーに直接付与できますが、エンドユーザーが AWS リソースに対して持つアクセスを制限する場合は、ポリシーを起動ロールにアタッチする必要があります。次に、 AWS Service Catalog を使用して、製品の起動制約に起動ロールを適用します。起動ロールの適用、起動ロールの制限、サンプルの起動ロールの詳細については、「AWS Service Catalog の起動制約」を参照してください。

> **Note**
>
> AWS Service Catalog 管理者に IAM アクセス許可をユーザーに付与すると、代わりに管理者コンソールビューが表示されます。エンドユーザーが管理者コンソールビューにアクセス可能にする場合を除き、これらのアクセス権限をエンドユーザーに付与しないでください。

エンドユーザー向け製品アクセス

エンドユーザーがアクセス権を付与した製品を使用する前に、製品の AWS CloudFormation テンプレート内の基盤となる各 AWS リソースの使用を許可する追加の IAM アクセス許可を提供する必要があります。例えば、製品テンプレートに Amazon Relational Database Service (Amazon RDS) が含まれる場合、製品を起動する Amazon RDS アクセス許可をユーザーに付与する必要があります。

AWSServiceCatalogEndUserReadOnlyAccess ポリシーを適用すると、ユーザーはエンドユーザーコンソールビューにアクセスできますが、製品を起動し、プロビジョニング済み製品を管理するために必要なアクセス権限は与えられません。これらのアクセス許可は IAM のエンドユーザーに直接付与できますが、エンドユーザーが AWS リソースに対して持つアクセスを制限する場合は、ポリシーを起動ロールにアタッチする必要があります。次に、 AWS Service Catalog を使用して、製品の起動制約に起動ロールを適用します。起動ロールの適用、起動ロールの制限、サンプルの起動ロールの詳細については、「AWS Service Catalog の起動制約」を参照してください。

プロビジョニング済み製品を管理するためのポリシーの例

カスタムポリシーを作成して所属組織のセキュリティ要件を満たすことができます。以下では、ユーザーレベル、ロールレベル、アカウントレベルをサポートするアクションごとにアクセスレベルをカスタマイズする方法の例を示します。各自のロールまたは各自がログインしているアカウントで作成したプロビジョニング済み製品を表示、更新、終了、管理するためのアクセス権を付与できます。ユーザー自身が作成したもののみを操作することも、他のユーザーが作成したものも含めて操作することもできます。このアクセス権は階層状に順次適用されます。アカウントレベルのアクセス

を付与すると、ロールレベルとユーザーレベルのアクセスも付与されます。ロールレベルのアクセスを付与すると、ユーザーレベルのアクセスも付与されますが、アカウントレベルのアクセスは付与されません。これらのアクセス権は、Condition ブロックを accountLevel、roleLevel、または userLevel として使用して、ポリシー JSON で指定できます。

これらの例は、 AWS Service Catalog API 書き込みオペレーションのアクセスレベル、 UpdateProvisionedProduct および TerminateProvisionedProduct、読み取りオペレーションDescribeRecord、、 にも適用されますScanProvisionedProductsListRecordHistory。ScanProvisionedProducts および ListRecordHistory API オペレーションは AccessLevelFilterKey という入力を使用し、このキーの値は上で説明した Condition ブロックレベルに対応します (accountLevel は「アカウント」の AccessLevelFilterKey 値、roleLevel は「ロール」、userLevel は「ユーザー」に相当します)。詳細については、「Service Catalog デベロッパーガイド」を参照してください。

例

- プロビジョニング済み製品に対する完全な管理アクセス
- プロビジョニング済み製品へのエンドユーザーアクセス
- プロビジョニング済み製品に対する部分的な管理アクセス

プロビジョニング済み製品に対する完全な管理アクセス

次のポリシーでは、アカウントレベルで、カタログ内のプロビジョニング済み製品およびレコードに対する読み取りと書き込みのフルアクセスを許可します。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicecatalog:*"
         ],
         "Resource":"*",
         "Condition": {
            "StringEquals": {
               "servicecatalog:accountLevel": "self"
            }
         }
      }
```

```
   ]
}
```

このポリシーは、次のポリシーと機能的に同じものです。

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

のポリシーでConditionブロックを指定しない場合 AWS Service Catalog 、"servicecatalog:accountLevel"アクセスの指定と同じものとして扱われます。accountLevel のアクセスには、roleLevel と userLevel のアクセスが含まれることに注意してください。

プロビジョニング済み製品へのエンドユーザーアクセス

次のポリシーでは、読み取りと書き込みのオペレーションへのアクセスを、現在のユーザーが作成したプロビジョニング済み製品または関連するレコードにのみ制限します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:DescribeRecord",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:ListRecordHistory",
                "servicecatalog:ProvisionProduct",
                "servicecatalog:ScanProvisionedProducts",
```

```
                "servicecatalog:SearchProducts",
                "servicecatalog:TerminateProvisionedProduct",
                "servicecatalog:UpdateProvisionedProduct"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:userLevel": "self"
                }
            }
        }
    ]
}
```

プロビジョニング済み製品に対する部分的な管理アクセス

次の 2 つのポリシーでは、両方が同じユーザーに適用された場合、読み取り専用のフルアクセスと書き込みの制限付きアクセスを提供することで、一種の「部分的な管理アクセス」を許可します。つまり、ユーザーはカタログのアカウント内にあるプロビジョニング済み製品または関連するレコードを表示することはできますが、そのユーザーが所有しないプロビジョニング済み製品やレコードに対しては一切の操作を実行できません。

最初のポリシーでは、ユーザーに許可されるアクセスは、現在のユーザーが作成したプロビジョニング済み製品に対する書き込みオペレーションのみとなり、他のユーザーが作成したプロビジョニング済み製品は対象外になります。2 番目のポリシーでは、すべて (ユーザー、ロール、またはアカウント) が作成したプロビジョニング済み製品に対する読み込みオペレーションへのフルアクセスを追加します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:ProvisionProduct",
                "servicecatalog:SearchProducts",
                "servicecatalog:TerminateProvisionedProduct",
                "servicecatalog:UpdateProvisionedProduct"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:userLevel": "self"
                }
            }
        }
    ]
}
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:DescribeRecord",
                "servicecatalog:ListRecordHistory",
                "servicecatalog:ScanProvisionedProducts"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:accountLevel": "self"
                }
            }
        }
    ]
}
```

AWS の マネージドポリシー AWS Service Catalog AppRegistry

AWS マネージドポリシー: **AWSServiceCatalogAdminFullAccess**

を IAM エンティティAWSServiceCatalogAdminFullAccessにアタッチできます。 AppRegistry は、ユーザーに代わって がアクションを実行できるようにするサービスロールにもこのポリシー AppRegistry をアタッチします。

このポリシーは、管理者コンソールビューへのフルアクセスを許可する###権限を付与し、製品とポートフォリオを作成および管理する権限を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- servicecatalog – プリンシパルに管理者コンソールビューへの完全なアクセス許可を付与し、ポートフォリオと製品の作成と管理、制約の管理、エンドユーザーへのアクセス許可の付与、 内でのその他の管理タスクの実行を許可します AWS Service Catalog。
- cloudformation— AWS CloudFormation スタックを一覧表示、読み取り、書き込み、タグ付けするための AWS Service Catalog 完全なアクセス許可を付与します。
- config– を介してポートフォリオ、製品、およびプロビジョニング済み製品に対する AWS Service Catalog 限定的なアクセス許可を付与します AWS Config。
- iam— 製品およびポートフォリオの作成と管理に必要なサービスユーザー、グループ、またはロールを表示および作成するためのすべての権限をプリンシパルに許可します。
- ssm – AWS Service Catalog 現在の AWS アカウントと AWS リージョンの Systems Manager ドキュメントを一覧表示および読み取る AWS Systems Manager ために を に許可します。

ポリシーを表示します: AWSServiceCatalogAdminFullAccess。

AWS マネージドポリシー: **AWSServiceCatalogAdminReadOnlyAccess**

を IAM エンティティAWSServiceCatalogAdminReadOnlyAccessにアタッチできます。AppRegistry は、ユーザーに代わって がアクションを実行できるようにするサービスロールにもこのポリシー AppRegistry をアタッチします。

このポリシーは、管理者コンソールビューへの完全なアクセスを許可する######権限を付与します。このポリシーは、製品とポートフォリオを作成または管理するためのアクセス権は付与しません。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- servicecatalog — 管理者コンソールビューへの読み取り専用アクセス権をプリンシパルに許可します。
- cloudformation– AWS CloudFormation スタックを一覧表示および読み取るための AWS Service Catalog 制限されたアクセス許可を許可します。
- config– を介してポートフォリオ、製品、およびプロビジョニング済み製品に対する AWS Service Catalog 限定的なアクセス許可を付与します AWS Config。

- iam— 製品およびポートフォリオの作成と管理に必要なサービスユーザー、グループ、または ロールを表示するための限定的な権限をプリンシパルに許可します。
- ssm – AWS Service Catalog 現在の AWS アカウントと AWS リージョンの Systems Manager ドキュメントを一覧表示および読み取る AWS Systems Manager ために を に許可します。

ポリシーを表示します: AWSServiceCatalogAdminReadOnlyAccess。

AWS マネージドポリシー: **AWSServiceCatalogEndUserFullAccess**

を IAM エンティティAWSServiceCatalogEndUserFullAccessにアタッチできます。 は、ユーザーに代わって がアクションを実行できるようにするサービスロール AppRegistry にもこのポリシー AppRegistry をアタッチします。

このポリシーは、エンドユーザーコンソールビューへの完全なアクセスを許可する###権限を付与し、製品を起動してプロビジョニングされた製品を管理する権限を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- servicecatalog - プリンシパルに、エンドユーザーコンソールビューに対する完全な権限と、製品を起動してプロビジョニングされた製品を管理する権限を許可します。
- cloudformation— AWS CloudFormation スタックを一覧表示、読み取り、書き込み、タグ付けするための AWS Service Catalog 完全なアクセス許可を付与します。
- config– ポートフォリオ、製品、プロビジョニング済み製品の詳細を 経由で一覧表示および読み取るための AWS Service Catalog 制限付きアクセス許可を付与します AWS Config。
- ssm – AWS Service Catalog 現在の AWS アカウントと AWS リージョンの Systems Manager ドキュメントを読み取る AWS Systems Manager ために を に許可します。

ポリシーを表示します: AWSServiceCatalogEndUserFullAccess。

AWS マネージドポリシー: **AWSServiceCatalogEndUserReadOnlyAccess**

を IAM エンティティAWSServiceCatalogEndUserReadOnlyAccessにアタッチできます。 は、ユーザーに代わって がアクションを実行できるようにするサービスロール AppRegistry にもこのポリシー AppRegistry をアタッチします。

このポリシーは、エンドユーザーのコンソールビューへの読み取り専用アクセスを許可する#####権限を付与します。このポリシーは、製品を起動し、プロビジョニング済み製品を管理する権限は付与しません。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- servicecatalog — エンドユーザーコンソールビューに対する読み取り専用の権限をプリンシパルに許可します。
- cloudformation– AWS CloudFormation スタックを一覧表示および読み取るための AWS Service Catalog 制限されたアクセス許可を許可します。
- config– ポートフォリオ、製品、プロビジョニング済み製品の詳細を 経由で一覧表示および読み取るための AWS Service Catalog 制限付きアクセス許可を付与します AWS Config。
- ssm – AWS Service Catalog 現在の AWS アカウントと AWS リージョンの Systems Manager ドキュメントを読み取る AWS Systems Manager ために を に許可します。

ポリシーを表示します: AWSServiceCatalogEndUserReadOnlyAccess。

AWS マネージドポリシー: **AWSServiceCatalogSyncServiceRolePolicy**

AWS Service Catalog は、このポリシーをAWSServiceRoleForServiceCatalogSyncサービスにリンクされたロール (SLR) AWS Service Catalog にアタッチし、 が外部リポジトリのテンプレートを AWS Service Catalog 製品に同期できるようにします。

このポリシーは、 AWS Service Catalog アクション (API コールなど) および AWS Service Catalog に依存する他の AWS サービスアクションへの制限付きアクセスを許可するアクセス許可を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- servicecatalog – AWS Service Catalog アーティファクト同期ロールにパブリック APIsへの AWS Service Catalog 制限付きアクセスを許可します。
- codeconnections– AWS Service Catalog アーティファクト同期ロールにCodeConnections パブリック APIsへの制限付きアクセスを許可します。
- cloudformation– AWS Service Catalog アーティファクト同期ロールにパブリック APIsへの AWS CloudFormation 制限付きアクセスを許可します。

ポリシーを表示します: AWSServiceCatalogSyncServiceRolePolicy。

サービスにリンクされたロールの詳細

AWS Service Catalog は、ユーザーが を使用する AWS Service Catalog 製品を作成または更新するときに作成されるAWSServiceRoleForServiceCatalogSyncサービスにリンクされたロールに対して上記のアクセス許可の詳細を使用します CodeConnections。このポリシーは、 AWS CLI、AWS API、または AWS Service Catalog コンソールを使用して変更できます。サービスにリンクされたロールを作成、編集、削除する方法の詳細については、「AWS Service Catalogのサービスにリンクされたロール (SLR) の使用」を参照してください。

AWSServiceRoleForServiceCatalogSync サービスにリンクされたロールに含まれるアクセス許可により AWS Service Catalog 、 は顧客に代わって次のアクションを実行できます。

- servicecatalog:ListProvisioningArtifacts — AWS Service Catalog アーティファクト同期ロールが、リポジトリ内のテンプレートファイルに同期されている特定の AWS Service Catalog 製品のプロビジョニングアーティファクトを一覧表示できるようにします。
- servicecatalog:DescribeProductAsAdmin — AWS Service Catalog アーティファクト同期ロールが DescribeProductAsAdmin API を使用して、リポジトリ内のテンプレートファイルに同期される AWS Service Catalog 製品および関連するプロビジョニングされたアーティファクトの詳細を取得できるようにします。アーティファクト同期ロールは、この呼び出しからの出力を使用して、プロビジョニングアーティファクトに対する製品の Service Quota 制限を検証します。
- servicecatalog:DeleteProvisioningArtifact — AWS Service Catalog アーティファクト同期ロールがプロビジョニングされたアーティファクトを削除できるようにします。
- servicecatalog:ListServiceActionsForProvisioningArtifact — AWS Service Catalog アーティファクト同期ロールが、サービスアクションがプロビジョニングアーティファクトに関連付けられているかどうかを判断し、サービスアクションが関連付けられている場合にプロビジョニングアーティファクトが削除されないようにします。
- servicecatalog:DescribeProvisioningArtifact — AWS Service Catalog アーティファクト同期ロールが DescribeProvisioningArtifact API から、SourceRevisionInfo出力で提供されるコミット ID などの詳細を取得できるようにします。
- servicecatalog:CreateProvisioningArtifact — 外部リポジトリのソーステンプレートファイルへの変更が検出された場合 (git-push がコミットされた場合など) 、アー AWS Service Catalog ティファクト同期ロールが新しいプロビジョニングされたアーティファクトを作成できるようにします。

- `servicecatalog:UpdateProvisioningArtifact` — AWS Service Catalog アーティファクト同期ロールが、接続済みまたは同期済み製品のプロビジョニング済みアーティファクトを更新できるようにします。
- `codeconnections:UseConnection` — AWS Service Catalog アーティファクト同期ロールが既存の接続を使用して製品を更新および同期できるようにします。
- `cloudformation:ValidateTemplate` — AWS Service Catalog アーティファクト同期ロールが に制限されたアクセスを許可 AWS CloudFormation して、外部リポジトリで使用されているテンプレートのテンプレート形式を検証し、 がテンプレートをサポート AWS CloudFormation しているかどうかを検証します。

AWS マネージドポリシー: **AWSServiceCatalogOrgsDataSyncServiceRolePolicy**

AWS Service Catalog は、このポリシーを`AWSServiceRoleForServiceCatalogOrgsDataSync`サービスにリンクされたロール (SLR) AWS Service Catalog にアタッチし、 が と同期できるようにします AWS Organizations。

このポリシーは、 AWS Service Catalog アクション (API コールなど) および AWS Service Catalog に依存する他の AWS サービスアクションへの制限付きアクセスを許可するアクセス許可を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `organizations`— AWS Service Catalog データ同期ロールにパブリック APIsへの AWS Organizations 制限付きアクセスを許可します。

ポリシーを表示します: AWSServiceCatalogOrgsDataSyncServiceRolePolicy。

サービスにリンクされたロールの詳細

AWS Service Catalog は、ユーザーが AWS Organizations 共有ポートフォリオアクセスを有効にするか、ポートフォリオ共有を作成したときに作成される`AWSServiceRoleForServiceCatalogOrgsDataSync`サービスにリンクされたロールに対して上記のアクセス許可の詳細を使用します。このポリシーは、 AWS CLI、 AWS API、または AWS Service Catalog コンソールを使用して変更できます。サービスにリンクされたロールを作成、編

集、削除する方法の詳細については、「AWS Service Catalogのサービスにリンクされたロール (SLR) の使用」を参照してください。

AWSServiceRoleForServiceCatalogOrgsDataSync サービスにリンクされたロールに含まれるアクセス許可により AWS Service Catalog 、 は顧客に代わって次のアクションを実行できます。

- organizations:DescribeAccount — AWS Service Catalog Organizations Data Sync ロールが、指定されたアカウントに関する AWS Organizations関連情報を取得できるようにします。
- organizations:DescribeOrganization — AWS Service Catalog Organizations Data Sync ロールが、ユーザーのアカウントが属する組織に関する情報を取得できるようにします。
- organizations:ListAccounts — AWS Service Catalog Organizations Data Sync ロールがユーザーの組織内のアカウントを一覧表示できるようにします。
- organizations:ListChildren — AWS Service Catalog Organizations Data Sync ロールが、指定された親 UOs) またはアカウントを一覧表示できるようにします。
- organizations:ListParents — AWS Service Catalog Organizations Data Sync ロールが、指定された子 OUs またはアカウントの直接の親として機能するルートまたは OU を一覧表示できるようにします。
- organizations:ListAWSServiceAccessForOrganization — AWS Service Catalog Organizations Data Sync ロールが、ユーザーが組織との統合を有効にした AWS サービスのリストを取得できるようにします。

非推奨ポリシー

次の管理ポリシーは廃止されました。

- ServiceCatalogAdminFullAccess — AWSServiceCatalogAdminFullAccess代わりに を使用します。
- ServiceCatalogAdminReadOnlyAccess — AWSServiceCatalogAdminReadOnlyAccess代わりに を使用します。
- ServiceCatalogEndUserFullAccess — AWSServiceCatalogEndUserFullAccess代わりに を使用します。
- ServiceCatalogEndUserAccess — AWSServiceCatalogEndUserReadOnlyAccess代わりに を使用します。

次の手順を使用して、現在のポリシーを使用して管理者とエンドユーザーにアクセス権限を確実に付与します。

廃止されたポリシーから現在のポリシーに移行するには、AWS Identity and Access Management ユーザーガイドの「IAM ID 権限の追加と削除」を参照してください。

AppRegistry AWS 管理ポリシーの更新

このサービスがこれらの変更の追跡を開始した AppRegistry 以降の の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動アラートを受け取るには、AppRegistry ドキュメント履歴ページの RSS フィードにサブスクライブしてください。

変更	説明	日付
AWSServiceCatalogSyncServiceRolePolicy – 管理ポリシーの更新	AWS Service Catalog は、`AWSServiceCatalogSyncServiceRolePolicy` ポリシーを `codestar-connections` に変更するように更新しましたcodeconnections 。	2024 年 5 月 7 日
AWSServiceCatalogAdminFullAccess – 管理ポリシーの更新	AWS Service Catalog は、AWS Service Catalog 管理者がアカウントに`AWSServiceRoleForServiceCatalogOrgsDataSync` サービスにリンクされたロール (SLR) を作成するために必要なアクセス許可を含めるように`AWSServiceCatalogAdminFullAccess` ポリシーを更新しました。	2023 年 4 月 14 日
AWSServiceCatalogOrgsDataSyncServiceRolePolicy – 新しいマネージドポリシー	AWS Service Catalog は`AWSServiceCatalogOrgsDataSyncServiceRolePolicy` 、`AWSServiceRoleForServiceCatalogOrgsDataSync` サービスにリン	2023 年 4 月 14 日

変更	説明	日付
	クされたロール (SLR) にアタッチされている を追加し、AWS Service Catalog と同期できるようになりました AWS Organizations。このポリシーは、 AWS Service Catalog アクション (API コールなど) および AWS Service Catalog に依存する他の AWS サービスアクションへの制限付きアクセスを許可します。	
AWSServiceCatalogAdminFullAccess – 管理ポリシーの更新	AWS Service Catalog は、AWS Service Catalog 管理者のすべてのアクセス許可を含めるようにAWSServiceCatalogAdminFullAccess ポリシーを更新し、との互換性を作成しましたAppRegistry。	2023 年 1 月 12 日
AWSServiceCatalogSyncServiceRolePolicy – 新しいマネージドポリシー	AWS Service Catalog は、AWSServiceRoleForServiceCatalogSync サービスにリンクされたロール (SLR) にアタッチされている AWSServiceCatalogSyncServiceRolePolicy ポリシーを追加しました。このポリシーにより AWS Service Catalog 、 は外部リポジトリのテンプレートを AWS Service Catalog 製品に同期できます。	2022 年 11 月 18 日

変更	説明	日付
AWSServiceRoleForServiceCatalogSync – 新しいサービスにリンクされたロール	AWS Service Catalog は、AWSServiceRoleForServiceCatalogSync サービスにリンクされたロール (SLR) を追加しました。このロールは、 が を使用し、製品の AWS Service Catalog プロビジョニングアーティファクト CodeConnections を作成、更新、および記述 AWS Service Catalog するために必要です。	2022 年 11 月 18 日

変更	説明	日付
AWSServiceCatalogAdminFullAccess – 管理ポリシーの更新	AWS Service Catalog は、AWS Service Catalog 管理者に必要なすべてのアクセス許可を含めるようにAWSServiceCatalogAdminFullAccess ポリシーを更新しました。このポリシーは、管理者が作成、説明、削除など、すべての AWS Service Catalog リソースに対して実行できる特定のアクションを識別します。さらに、 の属性ベースのアクセスコントロール (ABAC) という最近起動された機能をサポートするようにポリシーが変更されました AWS Service Catalog。ABAC では、AWSServiceCatalogAdminFullAccess ポリシーをテンプレートとして使用し、タグに基づいて AWS Service Catalog リソースに対するアクションを許可または拒否できます。ABAC の詳細については、「AWS Identity and Access Managementの AWSのABAC の概要」を参照してください。	2022 年 9 月 30 日
AppRegistry が変更の追跡を開始しました	AppRegistry が AWS マネージドポリシーの変更の追跡を開始しました。	2022 年 9 月 15 日

AWS Service Catalogのサービスにリンクされたロールの使用

AWS Service Catalog は AWS Identity and Access Management (IAM) サービスにリンクされたロール を使用します。サービスにリンクされたロールは、 に直接リンクされた一意のタイプの IAM ロールです AWS Service Catalog。サービスにリンクされたロールは によって事前定義 AWS Service Catalog されており、ユーザーに代わってサービスから他の AWS のサービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、 の設定 AWS Service Catalog が簡単になります。 は、サービスにリンクされたロールのアクセス許可 AWS Service Catalog を定義し、特に定義されている場合を除き、 のみがそのロールを引き受け AWS Service Catalog ることができます。定義したアクセス許可には、信頼ポリシーと許可ポリシーが含まれます。この許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールは、まずその関連リソースを削除しなければ削除できません。これにより、AWS Service Catalog リソースにアクセスするためのアクセス許可を誤って削除することがないため、リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスについては、「IAM と連動するAWS のサービス」を参照し、Service-linked roles (サービスにリンクされたロール) の列内で Yes (はい) と表記されたサービスを確認してください。そのサービスに関するサービスリンクロールのドキュメントを表示するには、リンクが設定されている [Yes (はい)] を選択します。

AWSServiceRoleForServiceCatalogSync のサービスリンクロールのアクセス許可

AWS Service Catalog は、 という名前のサービスにリンクされたロールを使用できます**AWSServiceRoleForServiceCatalogSync**。このサービスにリンクされたロールは、 AWS Service Catalog が を使用し、製品のプロビジョニングアーティファクトを作成、更新、および記述 CodeConnections AWS Service Catalog するために必要です。

AWSServiceRoleForServiceCatalogSync サービスにリンクされたロールは、ロールの引き受けについて以下のサービスを信頼します。

- sync.servicecatalog.amazonaws.com

という名前のロールアクセス許可ポリシーAWSServiceCatalogSyncServiceRolePolicyは AWS Service Catalog 、 が指定されたリソースに対して次のアクションを実行できるようにします。

- アクション: CodeConnections 上で Connection
- アクション: AWS Service Catalog 製品の ProvisioningArtifact Create, Update, and Describeに対する

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「Service-linked role permissions」を参照してください。

AWSServiceRoleForServiceCatalogSync サービスリンクロールの作成

AWSServiceRoleForServiceCatalogSync サービスにリンクされたロールを手動で作成する必要はありません。 AWS Service Catalog 、、 AWS CLIまたは AWS API CodeConnections で を確立すると AWS Management Console、 によって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。

> **Important**
>
> このサービスリンクロールは、このロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。また、 AWS Service Catalog 2022 年 11 月 18 日より前にサービスを使用していた場合、サービスにリンクされたロールのサポートが開始されると、 はアカウントにAWSServiceRoleForServiceCatalogSyncロール AWS Service Catalog を作成しました。詳細については、IAM アカウントに新しいロールが表示されるを参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ手順でアカウントにロールを再作成できます。を確立すると CodeConnections、 によってサービスにリンクされたロールが再度 AWS Service Catalog 作成されます。

IAM コンソールを使用して、同期された AWS Service Catalog 製品のユースケースでサービスにリンクされたロールを作成することもできます。 AWS CLI または AWS API で、サービス名を使用してsync.servicecatalog.amazonaws.comサービスにリンクされたロールを作成します。詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

AWSServiceRoleForServiceCatalogOrgsDataSync のサービスリンクロールのアクセス許可

AWS Service Catalog は、 という名前のサービスにリンクされたロールを使用できますAWSServiceRoleForServiceCatalogOrgsDataSync。このサービスにリンクされたロールは、 AWS Service Catalog 組織が と同期し続けるために必要です AWS Organizations。

AWSServiceRoleForServiceCatalogOrgsDataSync サービスにリンクされたロールは、ロールの引き受けについて以下のサービスを信頼します。

- orgsdatasync.servicecatalog.amazonaws.com

AWSServiceRoleForServiceCatalogOrgsDataSync サービスにリンクされたロールでは、AWSServiceCatalogOrgsDataSyncServiceRolePolicy マネージドポリシー に加えて以下の信頼ポリシーを使用する必要があります。

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

という名前のロールアクセス許可ポリシーAWSServiceCatalogOrgsDataSyncServiceRolePolicyは AWS Service Catalog 、 が指定されたリソースに対して次のアクションを実行できるようにします。

- アクション: Organizations accounts に対する DescribeAccount、DescribeOrganization および ListAWSServiceAccessForOrganization
- アクション: Organizations accounts に対する ListAccounts、ListChildren および ListParent

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するには、アクセス許可を設定する必要があります。詳細については、「IAM ユーザーガイド」の「Service-linked role permissions」を参照してください。

`AWSServiceRoleForServiceCatalogOrgsDataSync` サービスリンクロールの作成

`AWSServiceRoleForServiceCatalogOrgsDataSync`サービスにリンクされたロールを手動で作成する必要はありません。 AWS Service Catalog は、ユーザーに代わってバックグラウンドで SLR を作成する AWS Service Catalog 許可ポートフォリオの共有として、 AWS Organizations との共有または を有効にするアクションを考慮します。

AWS Service Catalog は、、、 AWS CLIまたは AWS API `CreatePortfolioShare`で `EnableAWSOrganizationsAccess`または をリクエストすると AWS Management Console、サービスにリンクされたロールを自動的に作成します。

> **Important**
>
> このサービスリンクロールは、このロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。詳細については、IAM アカウントに新しいロールが表示されるを参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は、同じ手順でアカウントにロールを再作成できます。`EnableAWSOrganizationsAccess` または `CreatePortfolioShare` をリクエストすると、 AWS Service Catalog はサービスにリンクされたロールを再度作成します。

AWS Service Catalogのサービスにリンクされたロールの編集

AWS Service Catalog では、 `AWSServiceRoleForServiceCatalogSync`または`AWSServiceRoleForServiceCatalogOrgsDataSync`サービスにリンクされたロールを編集することはできません。サービスリンクロールを作成した後は、多くのエンティティによってロールが参照される可能性があるため、ロール名を変更することはできません。ただし、IAM を使用したロールの説明の編集はできます。詳細については、IAM ユーザーガイドの「サービスリンクロールの編集」を参照してください。

AWS Service Catalogのサービスリンクロールの削除

IAM コンソール、 AWS CLI、または AWS API を使用して、`AWSServiceRoleForServiceCatalogSync` または

AWSServiceRoleForServiceCatalogOrgsDataSync SLR を手動で削除できます。これを行うには、まずサービスにリンクされたロールを使用しているすべてのリソース (外部リポジトリに同期されている AWS Service Catalog 製品など) を手動で削除してから、サービスにリンクされたロールを手動で削除する必要があります。

AWS Service Catalog のサービスリンクロールをサポートするリージョン

AWS Service Catalog は、サービスが利用可能なすべてのリージョンでサービスにリンクされたロールの使用をサポートします。詳細については、「AWS リージョンとエンドポイント」を参照してください。

リージョン名	リージョン識別子	でのサポート AWS Service Catalog
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい
米国西部 (オレゴン)	us-west-2	はい
アフリカ (ケープタウン)	af-south-1	はい
アジアパシフィック (香港)	ap-east-1	はい
アジアパシフィック (ジャカルタ)	ap-southeast-3	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
カナダ (中部)	ca-central-1	はい

リージョン名	リージョン識別子	でのサポート AWS Service Catalog
欧州 (フランクフルト)	eu-central-1	はい
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	はい
欧州 (ミラノ)	eu-south-1	はい
欧州 (パリ)	eu-west-3	はい
欧州 (ストックホルム)	eu-north-1	はい
中東 (バーレーン)	me-south-1	はい
南米 (サンパウロ)	sa-east-1	はい
AWS GovCloud （米国東部）	us-gov-east-1	なし
AWS GovCloud （米国西部）	us-gov-west-1	なし

AWS Service Catalog ID とアクセスのトラブルシューティング

次の情報は、 と IAM の使用時に発生する可能性がある一般的な問題の診断 AWS Service Catalog と修正に役立ちます。

トピック

- でアクションを実行する権限がない AWS Service Catalog
- iam:PassRole を実行する権限がない
- 自分の AWS アカウント以外のユーザーに自分の AWS Service Catalog リソースへのアクセスを許可したい

でアクションを実行する権限がない AWS Service Catalog

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

す。次の例のエラーは、mateojackson ユーザーが コンソールを使用して架 my-example-widget空のリソースの詳細を表示しようとしているが、架空のaws:GetWidgetアクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 aws:GetWidget on resource: my-example-widget
```

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

iam:PassRole を実行する権限がない

iam:PassRole アクションを実行することが認可されていないというエラーが表示された場合、管理者に問い合わせ、サポートを依頼する必要があります。管理者は、ユーザー名とパスワードを提供した人です。そのユーザーにポリシーを更新して、 AWS Service Catalogにロールを渡すことができるようにするように依頼します。

一部の AWS サービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

次のエラー例は、marymajor という名前のユーザーがコンソールを使用して AWS Service Catalogでアクションを実行しようとすると発生します。ただし、アクションには、サービスロールによってサービスに許可が付与されている必要があります。メアリーには、ロールをサービスに渡す許可がありません。

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

この場合、Mary は管理者にポリシーを更新して iam:PassRole action の実行を許可するように依頼します。

自分の AWS アカウント以外のユーザーに自分の AWS Service Catalog リソースへのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- がこれらの機能 AWS Service Catalog をサポートしているかどうかを確認するには、 AWS Service Catalog 管理者ガイドAWS Identity and Access Management の AWS Service Catalog「」の「」を参照してください。
- 所有している AWS アカウント間で リソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有している別の AWS アカウントの IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティー AWS アカウントに提供する方法については、IAM ユーザーガイドの「サードパーティーが所有する AWS アカウントへのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いの詳細については、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

アクセス権限の制御

AWS Service Catalog ポートフォリオを使用すると、管理者はエンドユーザーのグループに対してレベルのアクセス制御を行うことができます。ユーザーをポートフォリオに追加すると、ユーザーは、ポートフォリオ内の任意の製品を閲覧および起動できるようになります。詳細については、「the section called “ポートフォリオの管理”」を参照してください。

制約

制約により、特定のポートフォリオから製品を起動するときにエンドユーザーに適用されるルールが制御されます。制約を使用して、製品に制限を適用し、ガバナンスまたはコスト管理を実現します。制約の詳細については、「the section called “ 制約の使用”」を参照してください。

AWS Service Catalog 起動制約により、エンドユーザーが必要とするアクセス許可をより詳細に制御できます。管理者がポートフォリオ内の製品の起動制約を作成すると、起動制約によって、エンドユーザーがそのポートフォリオから製品を起動するときに使用されるロール ARN が関連付けられます。このパターンを使用すると、 AWS リソース作成へのアクセスを制御できます。詳細については、「the section called “起動制約”」を参照してください。

でのログ記録とモニタリング AWS Service Catalog

AWS Service Catalog は、すべての AWS Service Catalog API コールをキャプチャし AWS CloudTrail、指定した Amazon S3 バケットにログファイルを配信するサービスである と統合します。詳細については、「 を使用した AWS Service Catalog API コールのログ記録」を参照してください CloudTrail。

通知制約を使用して、スタックイベントに関する Amazon SNS 通知を設定することもできます。詳細については、「the section called “通知の制約”」を参照してください。

のコンプライアンス検証 AWS Service Catalog

サードパーティーの監査者は、以下を含む複数の コンプライアンスプログラム AWS Service Catalog の一環として のセキュリティと AWS コンプライアンスを評価します。

- System and Organization Controls (SOC)
- Payment Card Industry Data Security Standard (PCI DSS)
- Federal Risk and Authorization Management Program (FedRAMP)
- Health Insurance Portability and Accountability Act (HIPAA)

特定のコンプライアンスプログラム AWS の対象となるサービスのリストについては、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。 一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

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、「Downloading reports in AWS Artifact」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 Service Catalog 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異なります。 AWS では、コンプライアンスに役立つ以下のリソースを提供しています。

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

- AWS コンプライアンスのリソース - このワークブックおよびガイド群には、貴社の所在地や属する業界に適用可能なものが含まれています。
- AWS Config – この AWS 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 Security Hub – この AWS サービスは、 内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への準拠を確認するのに役立ちます。

の耐障害性 AWS Service Catalog

AWS グローバルインフラストラクチャは、 AWS リージョンと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 リージョンは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続された、物理的に分離された複数のアベイラビリティーゾーンを提供します。アベイラビリティーゾーンでは、アベイラビリティーゾーン間で中断せずに、自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、およびスケーラビリティが優れています。

AWS 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 グローバルインフラストラクチャに加えて、 は AWS Service Catalog セルフサービスアクション AWS Service Catalog を提供します。セルフサービスアクションにより、お客様はコンプライアンスとセキュリティ対策に従いながら、管理メンテナンスやエンドユーザートレーニングを減らすことができます。セルフサービスアクションを使用すると、管理者は、 AWS Service Catalogでの運用タスク (バックアップや復元など) の実行、問題のトラブルシューティング、承認されたコマンドの実行、アクセス許可のリクエストをエンドユーザーに許可できます。詳細については、「the section called “サービスアクションの使用”」を参照してください。

のインフラストラクチャセキュリティ AWS Service Catalog

マネージドサービスである AWS Service Catalog 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 がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「 Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

が AWS 公開した API コールを使用して、ネットワーク AWS Service Catalog 経由で にアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。TLS 1.2 は必須で TLS 1.3 がお勧めです。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

を使用すると AWS Service Catalog、データを保存するリージョンを制御できます。ポートフォリオと製品は、それらを利用可能にしたリージョンでのみ利用できます。`CopyProduct` API を使用して、製品を別のリージョンにコピーできます。

のセキュリティのベストプラクティス AWS Service Catalog

AWS Service Catalog には、独自のセキュリティポリシーを開発および実装する際に考慮すべきいくつかのセキュリティ機能が用意されています。以下のベストプラクティスは一般的なガイドラインであり、完全なセキュリティソリューションを説明するものではありません。これらのベストプラクティスはお客様の環境に必ずしも適切または十分でない可能性があるので、処方箋ではなく、あくまで有用な考慮事項とお考えください。

製品の起動時にユーザーが入力するパラメータ値を制限するルールを定義できます。このルールは、製品の AWS CloudFormation テンプレートのデプロイ方法を制限するため、テンプレート制約と呼ばれます。簡単なエディターを使用してテンプレート制約を作成し、各製品に適用します。

AWS Service Catalog は、新しい製品をプロビジョニングするとき、または既に使用されている製品を更新するときに制約を適用します。常に、ポートフォリオと製品に適用されるすべての制約の中で、最も厳しい制約が適用されます。例えば、すべての Amazon EC2 インスタンスの起動を許可する製品と、2 つの制約 (GPU 以外のすべてのタイプの EC2 インスタンスの起動を許可する制約と、t1.micro と m1.small EC2 インスタンスのみの起動を許可する制約) を含んだポートフォリオがあるシナリオを考えてみます。この例では、 は 2 番目の制限 (t1.micro および m1.small) AWS Service Catalog を適用します。

IAM ポリシーを起動ロールにアタッチするときに、エンドユーザーが AWS リソースに対して持つアクセスを制限できます。次に、 AWS Service Catalog を使用して、製品の起動時に ロールを使用する起動制約を作成します。

の 管理ポリシーの詳細については AWS Service Catalog、「 の AWS 管理ポリシー」を参照してください AWS Service Catalog。

カタログの管理

AWS Service Catalog には、管理者コンソールからポートフォリオ、製品、および制約を管理するためのインターフェイスがあります。

> **Note**
>
> このセクションのタスクを実行するには、AWS Service Catalog の管理者権限が必要です。詳細については、「AWS Service Catalogにおけるアイデンティティとアクセスの管理」を参照してください。

タスク

- ポートフォリオの管理
- 製品の管理
- AWS Service Catalog 制約の使用
- AWS Service Catalog のサービスアクション
- ポートフォリオへの AWS Marketplace 製品の追加
- の使用 AWS CloudFormation StackSets
- 予算の管理

ポートフォリオの管理

ポートフォリオの作成、表示、および更新は、AWS Service Catalog 管理者コンソールの [ポートフォリオ] ページで行います。

タスク

- ポートフォリオの作成、表示、削除
- ポートフォリオの詳細の表示
- ポートフォリオの作成と削除
- 製品の追加
- 制約の追加
- ユーザーへのアクセス権限の付与
- ポートフォリオの共有

- ポートフォリオの共有とインポート

ポートフォリオの作成、表示、削除

[ポートフォリオ] ページには、現在のリージョンで作成したポートフォリオのリストが表示されます。このページを使用して、新しいポートフォリオの作成、ポートフォリオの詳細の表示、またはアカウントからのポートフォリオの削除を行います。

[ポートフォリオ] ページを表示するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 必要に応じて、別のリージョンを選択します。
3. AWS Service Catalog を初めて使用すると、AWS Service Catalog スタートページが表示されます。[Get started] を選択してポートフォリオを作成します。最初のポートフォリオを作成する手順に従い、[ポートフォリオ] ページに進みます。

AWS Service Catalog の使用中に、ナビゲーションバーの [Service Catalog]、[ポートフォリオ] を選択して、いつでも [ポートフォリオ] ページに戻ることができます。

ポートフォリオの詳細の表示

AWS Service Catalog 管理者コンソールで、[ポートフォリオの詳細] ページには、ポートフォリオの設定が一覧表示されます。このページでは、ポートフォリオ内の製品の管理、製品へのアクセス権のユーザーへの付与、 TagOptions および 制約の適用を行います。

[ポートフォリオの詳細] ページを表示するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 管理するポートフォリオを選択します。

ポートフォリオの作成と削除

[ポートフォリオ] ページを使用して、ポートフォリオを作成し、削除します。

新しいポートフォリオを作成するには:

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。

2. [ポートフォリオの作成] を選択します。
3. [ポートフォリオの作成] ページで、必要情報を入力します。
4. [作成] を選択します。AWS Service Catalog によってポートフォリオが作成され、ポートフォリオの詳細が表示されます。

ポートフォリオを削除するには

> **Note**
>
> 削除できるのはローカルポートフォリオのみです。インポートされた (共有) ポートフォリオは削除できますが、インポートされたポートフォリオは削除できません。

ポートフォリオを削除する前に、その製品、制約、グループ、ロール、ユーザー、共有、および をすべて削除する必要があります TagOptions。そのためには、ポートフォリオを開いて [ポートフォリオの詳細] を表示します。次に、タブを選択して削除します。

> **Note**
>
> エラーを回避するには、商品を削除する前にポートフォリオから制約を削除します。

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。
2. 削除したいポートフォリオを選択します。
3. [削除] を選択します。削除できるのはローカルポートフォリオのみです。インポートされた (共有) ポートフォリオを削除しようとすると、アクションメニューは使用できません。
4. 確認ウィンドウで、[Delete] を選択します。

製品の追加

ポートフォリオに製品を追加するには、新しい製品を既存のポートフォリオに直接アップロードするか、カタログの既存の製品をポートフォリオに関連付けることができます。

> **Note**
>
> AWS Service Catalog 製品を作成するときに、AWS CloudFormation テンプレートまたは Terraform 設定ファイルをアップロードできます。AWS CloudFormation テンプレートは

Amazon Simple Storage Service (Amazon S3) バケットに保存され、バケット名は「cf-templates-」で始まります。また、製品をプロビジョニングするときには、追加のバケットからオブジェクトを取得する権限も必要です。詳細については、「製品の作成」を参照してください。

新しい製品の追加

ポートフォリオの詳細 ページから新しい製品を直接追加します。このページから製品を作成すると、AWS Service Catalog により、現在選択されているポートフォリオに追加されます。

新しい製品を追加するには

1. [ポートフォリオ] ページに移動し、製品を追加するポートフォリオの名前を選択します。
2. [ポートフォリオの詳細] ページで、[製品] セクションを展開し、[新しい製品のアップロード] を選択します。
3. [製品の詳細を入力] に、以下のように入力します。

   - [製品名] – 製品の名前。
   - 製品説明 (オプション) — 製品説明。この説明は、正しい製品を選択するのに役立つように、製品リストに表示されます。
   - [説明] - 詳細な説明。この説明は、正しい製品を選択するのに役立つように、製品リストに表示されます。
   - 所有者またはディストリビューター — 所有者の名前またはメールアドレス。ディストリビューターの連絡先情報は任意です。
   - [ベンダー] (オプション) - アプリケーションの発行元の名前。このフィールドを使用すると、製品リストを並べ替えて、製品を見つけやすくすることができます。
4. [バージョンの詳細] ページに、以下のように入力します。

   - テンプレートの選択 — AWS CloudFormation 製品の場合は、独自のテンプレートファイル、ローカルドライブの AWS CloudFormation テンプレート、または Amazon S3 に保存されているテンプレート、既存の AWS CloudFormation Stack ARN テンプレート、または外部リポジトリに保存されているテンプレートファイルを指す URL を選択します。

     Teraform 製品では、独自のテンプレートファイル、ローカルドライブからの tar.gz 設定ファイル、Amazon S3 に保存されたテンプレートを指す URL、または外部リポジトリに保存された tar.gz 設定ファイルを選択します。

- バージョン名 (オプション) – 製品バージョンの名前 (例:「v1」、「v2beta」)。スペースは使用できません。
- [説明] (オプション) - このバージョンと前のバージョンとの違いを含む、製品バージョンの説明。

5. [Enter support details] に、以下のように入力します。

- [メール連絡先] (オプション) - 製品の問題を報告するためのメールアドレス。
- [サポートリンク] (オプション) - ユーザーがサポート情報またはファイルチケットを見つけることができるサイトの URL。URL は `http://`、または `https://` で始まる必要があります。管理者は、サポート情報の正確性とアクセスを維持する責任があります。
- [サポートの説明] (オプション) - ユーザーが [メール連絡先] および [サポートリンク] を使用する方法の説明。

6. [製品の作成] を選択します。

既存の製品の追加

[ポートフォリオ] リスト、[ポートフォリオの詳細] ページ、または [製品リスト] のページの 3 つの場所から既存の製品をポートフォリオに追加できます。

既存の製品をポートフォリオに追加するには

1. [ポートフォリオ] ページに移動します。
2. ポートフォリオを選択します。次に [アクション]-[ポートフォリオに製品を追加] を選択します。
3. 製品を選択し、[製品をポートフォリオへ追加] を選択します。

ポートフォリオからの製品の削除

ユーザーが製品を使用しないようにする場合は、ポートフォリオからその製品を削除します。製品は、[製品] ページからカタログでまだ使用でき、他のポートフォリオに追加できます。ポートフォリオから複数の製品を一度に削除できます。

ポートフォリオから製品を削除するには

1. [ポートフォリオ] ページに移動し、製品を含むポートフォリオを選択します。ポートフォリオの詳細ページが開きます。
2. [製品] セクションを展開します。

3. 1 つ以上の製品を選択し、[削除] を選択します。
4. 選択内容を確認します。

制約の追加

制約を追加して、ユーザーがどのように製品を使用するかを制御する必要があります。AWS Service Catalog でサポートされる制約事項のタイプの詳細については、「AWS Service Catalog 制約の使用」を参照してください。

製品に制約を追加するのは、ポートフォリオに配置された後です。

製品に制約を追加するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [ポートフォリオ] を選択し、ポートフォリオを選びます。
3. [ポートフォリオの詳細] ページで、[制約の作成] セクションを展開し、[制約の追加] を選択します。
4. [製品] で、制約事項を適用する製品を選択します。
5. [制約タイプ] で、次のいずれかのオプションを選択します。

   起動 — AWS リソースのプロビジョニングに使用される製品に IAM ロールを割り当てることができます。詳細については、「AWS Service Catalog の起動制約」を参照してください。

   通知 — 製品通知を Amazon SNS トピックにストリーミングできます。詳細については、「AWS Service Catalog 通知の制約」を参照してください。

   テンプレート – エンドユーザーが製品を起動するときに利用できるオプションを制限できます。テンプレートは、1 つ以上のルールを含む JSON 形式のテキストファイルで構成されます。ルールは、製品で使用される AWS CloudFormation テンプレートに追加されます。詳細については、「テンプレート制約のルール」を参照してください。

   スタックセット — を使用して、アカウントとリージョン間で製品のデプロイを設定できます AWS CloudFormation StackSets。詳細については、「AWS Service Catalog スタックセットの制約」を参照してください。

   [タグの更新] - 製品がプロビジョニングされた後にタグを更新できます。詳細については、「AWS Service Catalogタグ更新の制約」を参照してください。
6. [続行] を選択し、必要な情報を入力します。

制約を編集するには

1. AWS Management Console にサインインして [AWS Service Catalog 管理者コンソール] (https://console.aws.amazon.com/catalog/) を開きます。
2. [ポートフォリオ] を選択し、ポートフォリオを選びます。
3. 「ポートフォリオの詳細」ページで、「制約の作成」セクションを展開し、編集する制約を選択します。
4. [制約の編集] を選択します。
5. 必要に応じて制約を編集し、[保存] を選択します。

ユーザーへのアクセス権限の付与

グループまたはロールを通じてユーザーにポートフォリオへのアクセスを許可します。多くのユーザーにポートフォリオのアクセス権限を付与する最善の方法は、ユーザーを IAM グループに配置し、そのグループへのアクセス権限を付与することです。それにより、グループからユーザーを簡単に追加および削除して、ポートフォリオアクセスを管理することができます。詳細については、IAM ユーザーガイドの「IAM ユーザーとグループ」を参照してください。

ポートフォリオへのアクセスに加え、 ユーザーには、 AWS Service Catalog エンドユーザーコンソールへのアクセス権限も必要です。IAM でアクセス権限を適用することにより、コンソールへのアクセス権限を付与します。詳細については、「AWS Service Catalogにおけるアイデンティティとアクセスの管理」を参照してください。

ポートフォリオとそのプリンシパルを他のアカウントと共有したい場合は、プリンシパル名 (グループ、ロール、ユーザー) をポートフォリオに関連付けることができます。プリンシパル名はポートフォリオと共有され、エンドユーザーにアクセス権を付与するために受信者アカウントで使用されます。

ユーザーまたはグループにポートフォリオのアクセス権限を付与するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. ナビゲーションペインから [管理] を選択し、[ポートフォリオ] を選択します。
3. グループ、ロール、またはユーザーにアクセス権を付与するポートフォリオを選択します。AWS Service Catalog はポートフォリオの詳細ページに移動します。
4. ポートフォリオの詳細ページで、「アクセス」タブを選択します。
5. 「ポートフォリオアクセスで、「アクセス権の付与」を選択します。

6. [タイプ] で [プリンシパル名] を選択し、グループ/、ロール/、または ユーザー/、タイプを選択します。最大 9 個のプリンシパル名まで追加できます。
7. 「アクセス権の付与」を選択すると、プリンシパルが現在のポートフォリオに関連付けられます。

ポートフォリオへのアクセス権限を削除するには

1. ポートフォリオの詳細ページで、グループ、ロール、ユーザー名を選択します。
2. [アクセス権の削除] を選択します。

ポートフォリオの共有

別のAWSアカウントのAWS Service Catalog管理者が製品をエンドユーザーに配信できるようにするには、共有 または を使用してAWS Service Catalogポートフォリオ account-to-account を共有しますAWS Organizations。

account-to-account 共有または Organizations を使用してポートフォリオを共有する場合、そのポートフォリオのリファレンスを共有します。インポートされたポートフォリオの製品と制約は、共有した元のポートフォリオである共有ポートフォリオに対して行う変更と同期が維持されます。

受信者は、製品または制約を変更することはできませんが、エンドユーザーの (AWS Identity and Access Management) アクセス権を追加できます。

> **Note**
>
> 共有リソースを共有することはできません。これには、共有製品を含むポートフォリオが含まれます。

ccount-to-account 共有

これらのステップを完了するには、対象の AWS アカウント ID を取得する必要があります。ID は、対象のアカウントの AWS Management Console の [マイアカウント] ページにあります。

AWS アカウントとポートフォリオを共有するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。

2. 左側のナビゲーションメニューで、[ポートフォリオ] を選択し、共有するポートフォリオを選択します。「アクション」メニューで「共有」を選択します。
3. [アカウント ID を入力] に、共有する AWS アカウントのアカウント ID を入力します。(オプション) TagOption の共有 を選択します。次に、[共有] を選択します。
4. 対象のアカウントの AWS Service Catalog 管理者に URL を送信します。URL では、共有ポートフォリオの ARN が自動的に提供されて [ポートフォリオのインポート] ページが開きます。

ポートフォリオのインポート

別の AWS アカウントの AWS Service Catalog 管理者がポートフォリオを共有した場合、そのポートフォリオをアカウントにインポートし、その製品をエンドユーザーに配信できるようにします。

ポートフォリオが AWS Organizations を通じて共有されている場合は、ポートフォリオをインポートする必要はありません。

ポートフォリオをインポートするには、管理者からポートフォリオ ID を取得する必要があります。

インポートされたすべてのポートフォリオを表示するには、https://console.aws.amazon.com/servicecatalog/ で AWS Service Catalog コンソールを開きます。「ポートフォリオ」ページで、「インポート済み」タブを選択します。「インポートされたポートフォリオ」テーブルを確認します。

AWS Organizations との共有

AWS Organizations を使用して AWS Service Catalog ポートフォリオを共有できます。

まず、管理アカウントから共有するか、委任管理者アカウントから共有するかを決定する必要があります。管理アカウントから共有しない場合は、委任管理者アカウントを登録し、共有に使用してください。詳細については、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。

次に、共有する相手を決定する必要があります。次のエンティティと共有できます。

- 組織アカウント。
- 部門単位 (OU)。
- 組織そのもの。(これは、組織内のすべてのアカウントと共有されます)。

管理アカウントからの共有

組織構造を使用するか、組織ノードの ID を入力するときに、ポートフォリオを組織と共有できます。

組織構造を使用してポートフォリオを組織と共有するには

1. AWS Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [ポートフォリオ] ページで、共有するポートフォリオを選択します。「アクション」メニューで「共有」を選択します。
3. AWS Organizations を選択して組織構造に絞り込みます。

   ルートノードを選択して、ポートフォリオを組織全体、親組織単位 (OU)、子 OU、または組織内の AWS アカウントを共有できます。

   親 OU に共有すると、その親 OU 内のすべてのアカウントおよび子 OU にポートフォリオが共有されます。

   [AWS アカウントのみを表示] を選択すると、組織内のすべての AWS アカウントのリストを表示できます。

組織ノードの ID を入力して、ポートフォリオを組織と共有するには

1. AWS Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [ポートフォリオ] ページで、共有するポートフォリオを選択します。「アクション」メニューで「共有」を選択します。
3. [組織ノード] を選択します。

   組織全体、OU、または組織内の AWS アカウントと共有するかどうかを選択します。

   選択した組織ノードの ID を入力します。このノードは、AWS Organizations コンソール (https://console.aws.amazon.com/organizations/) で確認できます。

委任管理者アカウントからの共有

組織の管理アカウントは、他のアカウントを組織の委任管理者として登録および登録解除できます。

委任管理者は、管理アカウントと同じ方法で、組織内の AWS Service Catalog リソースを共有できます。ポートフォリオの作成、削除、共有などが許可されます。

委任管理者を登録または登録解除するには、マスターアカウントから API または CLI を使用する必要があります。詳細については、「AWS Organizations API リファレンス」の「RegisterDelegatedAdministrator」および「DeregisterDelegatedAdministrator」を参照してください。

> **Note**
>
> 管理者を委任する前に、管理者は、`EnableAWSOrganizationsAccess` を呼び出す必要があります。

委任管理者アカウントからポートフォリオを共有する手順は、前述の「the section called “管理アカウントからの共有”」で説明したように、マスターアカウントからの共有と同じです。

メンバーが委任管理者として登録解除されると、次のようになります。

- そのアカウントから作成されたポートフォリオ共有は削除されます。
- 新しいポートフォリオ共有を作成することはできません。

> **Note**
>
> 委任管理者が登録解除された後に、委任管理者によって作成されたポートフォリオと共有が削除されない場合は、委任管理者を再度登録して登録解除します。このアクションにより、そのアカウントで作成されたポートフォリオと共有が削除されます。

組織内のアカウントの移動

組織内でアカウントを移動すると、そのアカウントと共有されている AWS Service Catalog ポートフォリオが変更される可能性があります。

アカウントは、対象の組織または組織部門と共有されているポートフォリオにのみアクセスできます。

ポートフォリオを共有する TagOptions 際の共有

管理者として、共有を作成して を含めることができます TagOptions。管理者が次の操作を実行できるようにするキーと値のペア TagOptions です。

- タグの分類を定義し、適用します。
- タグオプションを定義し、製品やポートフォリオに関連付けます。
- ポートフォリオおよび製品に関連するタグオプションを他のアカウントと共有します。

メインアカウントでタグオプションを追加または削除すると、変更は自動的に受信者アカウントに表示されます。受信者アカウントでは、エンドユーザーが で製品をプロビジョニングするときに TagOptions、プロビジョニング済み製品のタグとなるタグの値を選択する必要があります。

受信者アカウントでは、管理者はインポートされたポートフォリオ TagOptions に追加のローカル を関連付けて、そのアカウントに固有のタグ付けルールを適用できます。

> **Note**
>
> ポートフォリオを共有するには、消費者の AWS アカウント ID が必要です。コンソールの [マイアカウント] で AWS アカウント ID を検索します。

> **Note**
>
> に単一の値 TagOption がある場合、 はプロビジョニングプロセス中にその値AWSを自動的に適用します。

ポートフォリオを共有する TagOptions ときに共有するには

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。
2. [ローカルポートフォリオ] で、ポートフォリオを選択し、開きます。
3. 上部のリストから [共有] を選択し、[共有] ボタンを選択します。
4. 別の AWS アカウントまたは Amazon の組織と共有することを選択します。
5. 12 桁のアカウント ID 番号を入力し、[有効化] を選択してから、[共有] を選択します。

共有したアカウントは、[共有したアカウント] セクションに表示されます。が有効 TagOptions であったかどうかを示します。

ポートフォリオ共有を更新して を含めることもできます TagOptions。ポートフォリオと製品に属するすべての TagOptions がこのアカウントと共有されるようになりました。

ポートフォリオ共有を更新して を含めるには TagOptions

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。
2. [ローカルポートフォリオ] で、ポートフォリオを選択し、開きます。
3. 上部のリストから [共有] を選択します。
4. [共有したアカウント] で、アカウント ID を選択してから、[アクション] を選択します。
5. [Update unshare] または [Unshare] を選択します。

   共有解除の更新 を選択すると、 を有効にして共有を開始します TagOptions。共有したアカウントは、[共有したアカウント] セクションに表示されます。

   [Unshare] を選択した場合、アカウントを共有する必要がなくなったことを確認します。

ポートフォリオを共有する際のプリンシパル名の共有

管理者は、プリンシパル名を含むポートフォリオ共有を作成できます。プリンシパル名は、管理者がポートフォリオで指定してポートフォリオと共有できるグループ、ロール、ユーザーの名前です。ポートフォリオを共有するときに、AWS Service Catalog は、それらのプリンシパル名がすでに存在するかどうかを確認します。存在する場合は、AWS Service Catalog は、一致する IAM プリンシパルを共有ポートフォリオに自動的に関連付けて、ユーザーにアクセス権を付与します。

> **Note**
>
> プリンシパルをポートフォリオに関連付けると、そのポートフォリオが他のアカウントと共有されたときに、権限昇格の過程が生じる可能性があります。受信者アカウントのユーザーが AWS Service Catalog 管理者ではないものの、プリンシパル (ユーザー/ロール) を作成する権限を持っている場合、そのユーザーはポートフォリオのプリンシパル名の関連付けと一致する IAM プリンシパルを作成できます。このユーザーは、AWS Service Catalog を介してどのプリンシパル名が関連付けられているのかわからない場合がありますが、ユーザーを推測できる場合があります。この潜在的な昇格の過程が懸念される場合は、AWS

Service Catalog は `PrincipalType` を `IAM` として使用することをお勧めします。この設定では、`PrincipalARN` が既に受信者アカウントに存在していなければ関連付けることはできません。

メインアカウントでタグオプションを追加または削除すると、AWS Service Catalog は、これらの変更を受信者のアカウントに自動的に適用します。受信者アカウントのユーザーは、その役割に基づいてタスクを実行できます。

- エンドユーザー はポートフォリオの製品をプロビジョニング、更新、終了できます。
- 管理者 は、インポートされたポートフォリオに追加の IAM プリンシパルを関連付けて、そのアカウントに固有のエンドユーザーにアクセス権を付与できます。

Note

プリンシパル名の共有は AWS Organizations でのみ使用できます。

ポートフォリオを共有する際にプリンシパル名を共有するには

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。
2. ローカルポートフォリオで、共有したいポートフォリオを選択します。
3. [アクション] メニューで [共有] を選択します。
4. AWS Organizations 内の組織を選択します。
5. 組織ルート全体、組織ユニット (OU)、または組織メンバーを選択します。
6. 共有設定で、プリンシパルの共有オプションを有効にします。

また、ポートフォリオ共有を更新して、プリンシパル名の共有を含めることもできます。これにより、そのポートフォリオに属するすべてのプリンシパル名が受信者アカウントと共有されます。

ポートフォリオ共有を更新して、プリンシパル名を有効または無効にするには

1. 左側のナビゲーションメニューから [ポートフォリオ] を選択します。
2. ローカルポートフォリオで、更新するポートフォリオを選択します。
3. [共有] タブを選択します。
4. 更新する共有を選択し、[共有] を選択します。

5. [共有を更新] を選択し、[有効化] を選択してプリンシパルの共有を開始します。その後、AWS Service Catalog は受信者アカウントのプリンシパル名を共有します。

受信者アカウントとのプリンシパル名の共有を停止したい場合は、プリンシパル共有を無効にします。

プリンシパル名を共有する場合はワイルドカードを使用する

AWS Service Catalog は、「*」や「?」などのワイルドカードを使用して IAM プリンシパル (ユーザー、グループ、ロール) 名へのポートフォリオアクセスを許可できます。ワイルドカードパターンを使用すると、複数の IAM プリンシパル名を一度に扱うことができます。ARN パスとプリンシパル名には、無制限のワイルドカード文字を使用できます。

許容できるワイルドカード ARN の例:

- **arn:aws:iam:::role/ResourceName_***
- **arn:aws:iam:::role/*/ResourceName_?**

許容できないワイルドカード ARN の例:

- **arn:aws:iam:::*/ResourceName**

IAM プリンシパル ARN 形式 (**arn:partition:iam:::resource-type/resource-path/resource-name**) では、有効な値には user/、group/、または role/ が含まれます。「?」 「*」と「*」は、resource-id セグメントのリソースタイプの後にのみ使用できます。特殊文字はリソース ID 内のどこでも使用できます。

「*」文字は「/」文字とも一致するため、リソース ID 内 にパスを作成できます。例:

arn:aws:iam:::role/*/ResourceName_? は **arn:aws:iam:::role/pathA/pathB/ResourceName_1** と **arn:aws:iam:::role/pathA/ResourceName_1** の両方に一致します。

ポートフォリオの共有とインポート

AWS Service Catalog 製品を、AWS アカウント に属していないユーザー (他の組織のユーザー、または組織内の他の AWS アカウント に属しているユーザーなど) が利用できるようにするには、ポートフォリオをそれらのユーザーと共有します。共有は、 account-to-account 共有、組織共有、スタックセットを使用したカタログのデプロイなど、いくつかの方法で共有できます。

製品とポートフォリオを他のアカウントと共有する前に、カタログの参照を共有するか、カタログのコピーを各受信者アカウントにデプロイするかを決定する必要があります。コピーをデプロイする場合、受信者アカウントに反映する更新が発生したら再デプロイする必要があります。

スタックセットを使用して、同時に複数のアカウントにカタログをデプロイできます。リファレンス (インポートされたバージョンのポートフォリオが元のバージョンと同期している場合) を共有する場合は、 account-to-account 共有を使用するか、 を使用して共有できますAWS Organizations。

スタックセットを使用してカタログのコピーを展開するには、「企業標準 AWS Service Catalog 製品のマルチリージョン、マルチアカウントカタログをセットアップする方法」を参照してください。

account-to-account 共有 または を使用してポートフォリオを共有する場合AWS Organizations、別のAWSアカウントのAWS Service Catalog管理者がポートフォリオをアカウントにインポートし、そのアカウントのエンドユーザーに製品を配布できるようにします。

このインポートされたポートフォリオは独立コピーではありません。インポートされたポートフォリオの製品と制約は、共有した元のポートフォリオである共有ポートフォリオに対して行う変更と同期が維持されます。ポートフォリオを共有する管理者である受信者管理者は、製品または制約を変更することはできませんが、エンドユーザーに対して AWS Identity and Access Management (IAM) アクセス権限を追加できます。詳細については、「ユーザーへのアクセス権限の付与」を参照してください。

受信者管理者は、次の方法で、自身の AWS アカウントに属しているエンドユーザーに製品を配信できます。

- インポートされたポートフォリオにユーザー、グループ、ロールを追加します。
- インポートされたポートフォリオからローカルポートフォリオに製品を追加することにより、受信者管理者が作成し、自身の AWS アカウントに属する別のポートフォリオが作成されます。次に、受信者の管理者は、そのローカルポートフォリオにユーザー、グループ、およびロールを追加します。共有ポートフォリオで製品に適用した制約は、ローカルポートフォリオにも存在します。ローカルポートフォリオの受信者管理者は、制限を追加することができますが、共有ポートフォリオから最初にインポートされた制約を削除することはできません。

共有ポートフォリオに製品または制約を追加または削除すると、変更はポートフォリオのすべてのインポートされたインスタンスに伝搬されます。たとえば、共有ポートフォリオから製品を削除する場合、その製品はインポートされたポートフォリオからも削除されます。また、製品が追加されたすべてのローカルポートフォリオからも削除されます。削除する前にエンドユーザーが製品を起動した場合、エンドユーザーのプロビジョニング済み製品は実行し続けますが、それ以降の起動では使用できなくなります。

共有ポートフォリオで製品に起動制約を適用する場合、製品のすべてのインポートされたインスタンスに伝搬されます。この起動制約を上書きするには、受信者管理者はローカルポートフォリオに製品を追加し、別の起動制約を適用します。有効な起動制約により、製品の起動ロールが設定されます。

起動ロールは、エンドユーザーが製品を起動するときに、AWS リソース (Amazon EC2 インスタンスや Amazon RDS データベースなど) をプロビジョニングするために AWS Service Catalog が使用する IAM ロールです。管理者は、特定の起動ロール ARN またはローカルロール名を指定できます。ロール ARN を使用する場合、エンドユーザーが起動ロールを所有するアカウントとは異なる AWS アカウントに属している場合でも、そのロールが使用されます。ローカルロール名を使用する場合、エンドユーザーのアカウントでその名前を持つ IAM ロールが使用されます。

起動制約と起動ロールの詳細については、「AWS Service Catalog の起動制約」を参照してください。起動ロールを所有する AWS アカウントが、AWS リソースをプロビジョニングし、このアカウントにより、これらのリソースの使用料金が発生します。詳細については、「AWS Service Catalog の料金」を参照してください。

この動画では、AWS Service Catalog のアカウント間でポートフォリオを共有する方法を説明します。

AWS Service Catalog のアカウント間でポートフォリオを共有 (https://www.youtube.com/embed/BVSohYOppjk%22%3EShare) します。

> **Note**
>
> インポートまたは共有されたポートフォリオの製品を再共有することはできません。

> **Note**
>
> ポートフォリオのインポートは、管理アカウントと依存アカウント間の同じリージョンで実行する必要があります。

共有ポートフォリオとインポートされたポートフォリオの関係

この表に、インポートされたポートフォリオと共有ポートフォリオの関係、およびポートフォリオをインポートする管理者が、そのポートフォリオとポートフォリオ内の製品で実行できることと、実行できないことの概要を示します。

共有ポートフォリオの要素	インポートされたポートフォリオとの関係	受信者の管理者が実行できること	受信者の管理者が実行できないこと
製品と製品バージョン	継承されます。 ポートフォリオ作成者が共有ポートフォリオに製品を追加または削除すると、変更はインポートされたポートフォリオに伝播されます。	インポートされた製品をポートフォリオに追加する。製品は、共有ポートフォリオとの同期を維持します。	インポートされたポートフォリオに製品をアップロード、追加、または削除する。
起動制約	継承されます。 ポートフォリオ作成者が共有製品に起動制約を追加または削除すると、変更は製品のすべてのインポートされたインスタンスに伝播されます。 受信者の管理者がインポートされた製品をローカルポートフォリオに追加した場合、そのインポートされた起動制約は共有ポートフォリオには引き継がれません。	ローカルポートフォリオでは、管理者は製品のローカルリリースに影響する起動制約を適用できます。	インポートされたポートフォリオとの間で、起動制約を追加または削除する。
テンプレート制約	継承されます。	ローカルポートフォリオでは、管理者はローカル製品を制約	インポートされたテンプレートの制約を削除する。

共有ポートフォリオの要素	インポートされたポートフォリオとの関係	受信者の管理者が実行できること	受信者の管理者が実行できないこと
	ポートフォリオ作成者がテンプレート制約を共有製品に追加または削除すると、変更は製品のすべてインポートされたインスタンスに伝搬されます。 受信者管理者がインポートされた製品をローカルポートフォリオに追加した場合、インポートされたテンプレートの制約はローカルポートフォリオに引き継がれません。	するテンプレート制約を追加できます。	
ユーザー、グループ、およびロール	継承されません。	管理者の AWS アカウントに属するユーザー、グループ、およびロールを追加する。	該当しません。

製品の管理

製品を作成したり、更新されたテンプレートに基づいて新しいバージョンを作成して製品を更新したり、製品をポートフォリオにグループ化してユーザーに配布したりできます。

新しいバージョンの製品は、ポートフォリオを通じて製品にアクセスできるすべてのユーザーに伝播されます。更新を配信すると、エンドユーザーは既存のプロビジョニングされた製品をアップデートできます。

タスク

- [製品] ページの表示
- 製品の作成
- ポートフォリオへの製品の追加
- 製品の更新
- 製品 GitHub、 GitHub エンタープライズ、または Bitbucket のテンプレートファイルとの同期
- 製品の削除
- バージョンの管理

[製品] ページの表示

AWS Service Catalog 管理者コンソールの [製品リスト] ページから製品を管理します。

[製品リスト] ページを表示するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [製品リスト] を選択します。

製品の作成

AWS Service Catalog 管理者コンソールの [製品] ページから製品を作成します。

> **Note**
>
> Terraform 製品を作成するには、Terraform プロビジョニングエンジンや起動ロールなどの追加設定が必要です。詳細については、「Terraform 製品の使用開始 」を参照してください。

新しい AWS Service Catalog 製品を作成するには

1. [製品リスト] ページに移動します。
2. [製品の作成] を選択し、[製品の作成] を選択します。
3. 製品の詳細 — 作成する製品の種類を選択できます。AWS Service Catalog は AWS CloudFormation、Terraform Cloud、External (Terraform Community Edition をサポート) の各製

品タイプをサポートします。製品詳細には、リストまたは詳細ページで製品を検索して表示したときに表示されるメタデータも含まれています。次のように入力します。

- [製品名] – 製品の名前。
- 商品説明 — 説明は商品リストに表示され、正しい商品を選択するのに役立ちます。
- 所有者 — この製品を公開する個人または組織。所有者は IT 組織または管理者の名前である可能性があります。
- ディストリビュータ (オプション) – アプリケーションのパブリッシャーの名前。このフィールドを使用すると、製品リストを並べ替えて、製品を見つけやすくすることができます。

4. バージョン詳細 により、テンプレートファイルを追加して製品を構築できます。次のように入力します。

- 方法の選択 — テンプレートファイルを追加するには 4 つの方法があります。
  - ローカルテンプレートファイルを使用 - AWS CloudFormation テンプレートまたは Terraform tar.gz 設定ファイルをローカルドライブからアップロードします。
  - Amazon S3 URL を使用 - Amazon S3 に保存されている AWS CloudFormation テンプレートまたは Terraform tar.gz 設定ファイルを指す URL を指定します。Amazon S3 の URL を指定する場合、`https://` で始まる必要があります。
  - 外部リポジトリを使用する - GitHub、 GitHub エンタープライズ、または Bitbucket コードリポジトリを指定します。 AWS Service Catalogを使用すると、製品をテンプレートファイルに同期できます。Terraform 製品の場合、テンプレートファイル形式は Tar にアーカイブされ、Gzip で圧縮された単一のファイルである必要があります。
  - 既存の CloudFormation スタックを使用する - 既存の CloudFormation スタックの ARN を入力します。このメソッドは Terraform Cloud 製品および External 製品をサポートしていません。
- バージョン名 (オプション) – 製品バージョンの名前 (例: 「v1」、「v2beta」)。スペースは使用できません。
- [説明] (オプション) - このバージョンと他のバージョンとの違いを含む、製品バージョンの説明。
- ガイダンス — 製品詳細ページの「バージョン」タブで管理されます。製品作成ワークフロー中に製品バージョンが作成されると、そのバージョンのガイダンスがデフォルトに設定されます。ガイダンスの詳細については、「バージョン管理」を参照してください。

5. サポートの詳細は、貴社の組織を特定し、サポートの窓口となります。次のように入力します。

- [メール連絡先] (オプション) - 製品の問題を報告するためのメールアドレス。

- [サポートリンク] (オプション) - ユーザーがサポート情報またはファイルチケットを見つけることができるサイトの URL。URL は `http://`、または `https://` で始まる必要があります。管理者は、サポート情報の正確性とアクセスを維持する責任があります。
- [サポートの説明] (オプション) - ユーザーが [メール連絡先] および [サポートリンク] を使用する方法の説明。

6. タグの管理 (オプション) — タグを使用してリソースを分類するだけでなく、このリソースを作成する権限を認証するためにも使用できます。
7. 製品の作成 — フォームの入力が完了したら、[製品の作成] を選択します。数秒後、製品が [製品のリスト] ページに表示されます。製品を表示するには、ブラウザの更新が必要になる場合があります。

CodePipeline を使用して、製品テンプレートを にデプロイし、ソースリポジトリで行った変更を配信するパイプラインを作成AWS Service Catalogおよび設定することもできます。詳細については、「チュートリアル: AWS Service Catalog にデプロイするパイプラインを作成する」を参照してください。

AWS CloudFormation および Terraform テンプレートでパラメータプロパティを定義し、プロビジョニング中にこれらのルールを適用できます。これらのプロパティには、最小長と最大長、最小値と最大値、許容値、および値の正規表現を定義する機能があります。提供された値がパラメータプロパティに準拠していない場合、AWS Service Catalog はプロビジョニング中に警告します。パラメータプロパティの詳細については、AWS CloudFormationユーザーガイドの「パラメータ」を参照してください。

トラブルシューティング

Amazon S3 バケットからオブジェクトを取得するためのアクセス許可が必要です。そうしないと、製品の起動または更新中に次のエラーが発生する場合があります。

```
Error: failed to process product version s3 access denied exception
```

このメッセージが表示された場合は、以下のバケットからオブジェクトを取得する権限があることを確認してください。

- プロビジョニングアーティファクトテンプレートが保存されているバケット。
- 「cf-templates-*」で始まり、AWS Service Catalog がプロビジョニングアーティファクトテンプレートを格納しているバケット。

- 「sc-*」で始まり、AWS Service Catalog がメタデータを格納している内部バケット。アカウントからこのバケットを表示することはできません。

次のポリシー例は、前述のバケットからオブジェクトを取得するために必要な最小限のアクセス許可を示しています。

```
{
          "Sid": "VisualEditor1",
          "Effect": "Allow",
          "Action": "s3:GetObject*",
          "Resource": [
              "arn:aws:s3:::YOUR_TEMPLATE_BUCKET",
              "arn:aws:s3:::YOUR_TEMPLATE_BUCKET/*",
              "arn:aws:s3:::cf-templates-*",
              "arn:aws:s3:::cf-templates-*/*",
              "arn:aws:s3:::sc-*",
              "arn:aws:s3:::sc-*/*"
          ]
      }
```

ポートフォリオへの製品の追加

任意の数のポートフォリオに製品を追加できます。製品が更新されると、その製品を含むすべてのポートフォリオ (共有ポートフォリオを含む) が自動的に新しいバージョンを受け取ります。

カタログからポートフォリオに製品を追加するには

1. [製品リスト] ページに移動します。
2. 製品を選択し、[アクション] を選択します。ドロップダウンメニューから [ポートフォリオに製品を追加] を選択します。「ポートフォリオ*name-of-product*に追加」ページに移動します。
3. ポートフォリオを選択し、[製品をポートフォリオへ追加] を選択します。

Terraform 製品をポートフォリオに追加する場合、その製品には起動制限が必要です。アカウントから IAM ロールを選択するか、IAM ロール ARN を入力するか、ロール名を入力する必要があります。ロール名を指定すると、アカウントが起動制約を使用する場合、アカウントは IAM ロールのその名前を使用します。これにより、起動ロールの制約がアカウントに依存しないようになり、共有アカウントごとに作成できるリソースの数が確実に減ります。詳細と手順については、「ステップ 6: Terraform 製品に起動制約を追加する」を参照してください。

ポートフォリオには、AWS CloudFormation と Terraform 製品タイプの混合である多くの製品を含めることができます。

製品の更新

製品のテンプレートを更新する必要がある場合は、製品の新しいバージョンを作成します。新しい製品バージョンは、製品を含むポートフォリオにアクセスできるすべてのユーザーに自動的に提供されます。

> **Note**
>
> 既存の製品を更新する場合、製品タイプ (AWS CloudFormation または Teraform) を変更することはできません。たとえば、AWS CloudFormation 製品を更新する場合、既存の AWS CloudFormation テンプレートを Terraform tar.gz 設定ファイルに置き換えることはできません。既存の AWS CloudFormation テンプレートファイルを新しい AWS CloudFormation テンプレートファイルで更新する必要があります。

以前の製品バージョンのプロビジョニングされた製品を現在実行しているエンドユーザーは、プロビジョニングされた製品を新しいバージョンに更新できます。製品の新しいバージョンが利用できる場合、ユーザーは [プロビジョニング済み製品リスト] または[プロビジョニング済み製品の詳細] ページで [プロビジョニング済み製品の更新] コマンドを使用できます。

AWS Service Catalog では、製品の新しいバージョンを作成する前に、AWS CloudFormation または Terraform エンジン内で製品の更新を テストして、機能することを確認することをお勧めします。

新しい製品バージョンを作成するには

1. 製品リスト ページに移動します。
2. 更新する製品を選択します。製品詳細ページに移動します。
3. [製品の詳細] ページで、[バージョン] タブを展開し、[新バージョンの作成] を選択します。
4. [バージョンの詳細] で、以下を実行します。

   - テンプレートの選択 — テンプレートファイルを追加するには 4 つの方法があります。

     ローカルテンプレートファイルを使用 - AWS CloudFormation テンプレートまたは Terraform tar.gz 設定ファイルをローカルドライブからアップロードします。

Amazon S3 URL を使用 - Amazon S3 に保存されている AWS CloudFormation テンプレートまたは Terraform tar.gz 設定ファイルを指す URL を指定します。Amazon S3 の URL を指定する場合、https:// で始まる必要があります。

外部リポジトリを使用する - GitHub、 GitHub エンタープライズ、または Bitbucket コードリポジトリを指定します。 AWS Service Catalogを使用すると、製品をテンプレートファイルに同期できます。Terraform 製品の場合、テンプレートファイル形式は Tar にアーカイブされ、Gzip で圧縮された単一のファイルである必要があります。

既存の CloudFormation スタックを使用する - 既存の CloudFormation スタックの ARN を入力します。このメソッドは Terraform Cloud 製品および External 製品をサポートしていません。

- [バージョンタイトル] – 製品バージョンの名前 (例えば「v1」、「v2beta」など)。スペースは使用できません。
- [説明] (オプション) - このバージョンと前のバージョンとの違いを含む、製品バージョンの説明。

5. [製品バージョンを作成] を選択します。

CodePipeline を使用して、製品テンプレートを にデプロイしAWS Service Catalog、ソースリポジトリに変更を配信するパイプラインを作成および設定することもできます。詳細については、「チュートリアル: AWS Service Catalog にデプロイするパイプラインを作成する」を参照してください。

製品 GitHub、 GitHub エンタープライズ、または Bitbucket のテンプレートファイルとの同期

AWS Service Catalog 外部リポジトリプロバイダを通じて管理されているテンプレートファイルに製品を同期できます。 AWS Service Catalog このタイプのテンプレート接続を使用する製品を Git 同期製品と呼びます。リポジトリオプションには GitHub、[GitHub エンタープライズ] または [Bitbucket] があります。 AWS アカウント 外部リポジトリアカウントで認証すると、 AWS Service Catalog 新しい製品を作成したり、既存の製品を更新してリポジトリ内のテンプレートファイルと同期したりできます。テンプレートファイルに変更が加えられ、リポジトリにコミットされると (git-push を使用するなど)、 AWS Service Catalog 自動的に変更が検出され、新しい製品バージョン (アーティファクト) が作成されます。

トピック

- 製品を外部のテンプレートファイルと同期させるために必要な権限
- アカウント接続を作成する
- Git と同期された製品接続の表示
- Git 同期製品接続の更新
- Git 同期製品接続を削除する
- Terraform 製品を GitHub、 GitHub エンタープライズ、または Bitbucket のテンプレートファイルと同期する
- AWS リージョン Git 同期製品のサポート

製品を外部のテンプレートファイルと同期させるために必要な権限

次の AWS Identity and Access Management (IAM) ポリシーをテンプレートとして使用すると、AWS Service Catalog 管理者は外部リポジトリのテンプレートファイルに製品を同期できます。このポリシーには、 CodeConnections との両方から必要な権限が含まれています。 AWS Service Catalog AWS Service Catalog 以下のテンプレートポリシーをコピーし、 AWS Service Catalog `AWSServiceCatalogAdminFullAccess`リポジトリ同期製品を有効にするときには管理ポリシーを使用することをおすすめします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CodeStarAccess",
            "Effect": "Allow",
            "Action": [
            "codestar-connections:UseConnection",
            "codestar-connections:PassConnection",
            "codestar-connections:CreateConnection",
            "codestar-connections:DeleteConnection",
            "codestar-connections:GetConnection",
            "codestar-connections:ListConnections",
            "codestar-connections:ListInstallationTargets",
            "codestar-connections:GetInstallationUrl",
            "codestar-connections:StartOAuthHandshake",
            "codestar-connections:UpdateConnectionInstallation",
            "codestar-connections:GetIndividualAccessToken"
            ],
            "Resource": "arn:aws:codestar-connections:*:*:connection/*"
        },
```

```
        {
        "Sid": "CreateSLR",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/
sync.servicecatalog.amazonaws.com/AWSServiceRoleForServiceCatalogArtifactSync",
        "Condition": {
        "StringLike": {
        "iam:AWSServiceName": "sync.servicecatalog.amazonaws.com"
                }
            }
        }
    ]
}
```

アカウント接続を作成する

AWS Service Catalog テンプレートファイルを製品と同期する前に、1 回限りの接続を作成して承認する必要があります。 account-to-account この接続を使用して、目的のテンプレートファイルを含むリポジトリの詳細を指定します。 AWS Service Catalog コンソール、 CodeConnections コンソール、 AWS Command Line Interface (CLI)、または CodeConnections API を使用して接続を作成できます。

接続を確立したら、 AWS Service Catalog コンソール、 AWS Service Catalog API、または CLI AWS Service Catalog を使用して同期された製品を作成できます。 AWS Service Catalog 管理者は、リポジトリとブランチにあるテンプレートファイルに基づいて、 AWS Service Catalog 新しい製品を作成したり、既存の製品を更新したりできます。リポジトリに変更がコミットされると、AWS Service Catalog その変更が自動的に検出され、新しい製品バージョンが作成されます。以前の製品バージョンは規定のバージョン制限まで維持され、非推奨ステータスが割り当てられます。

さらに、 AWS Service Catalog 接続が作成されると、サービスにリンクされたロール (SLR) が自動的に作成されます。この SLR により、 AWS Service Catalog はリポジトリにコミットされたテンプレートファイルの変更をすべて検出できます。SLR では、 AWS Service Catalog 同期した製品の新しい製品バージョンを自動的に作成することもできます。SLR の権限と機能について詳しくは、AWS Service Catalogの「サービスにリンクされたロール」を参照してください。

Git と同期された製品を作成するには

1. ナビゲーションパネルで、[製品のリスト] を選択し、[製品の作成] を選択します。
2. 製品の詳細を入力します。

3. [バージョンの詳細] で [AWS CodeStar プロバイダーを使用してコードリポジトリを指定] を選択し、[AWS CodeStar 新しい接続を作成] リンクを選択します。
4. 接続を作成したら、接続リストを更新し、新しい接続を選択します。リポジトリ、分岐、テンプレートファイルパス などのリポジトリの詳細を指定します。

   Terraform 設定ファイルの使用方法の詳細については、「Terraform 製品を GitHub、 GitHub エンタープライズ、または Bitbucket のテンプレートファイルと同期する 」を参照してください。

   a. (AWS Service Catalog 新しい製品リソースを作成する場合はオプション) Support 詳細セクションで、製品のメタデータを追加します。
   b. (AWS Service Catalog 新しい製品リソースを作成する場合はオプション)「タグ」セクションで「Add new tag」を選択し、「Key」 と「Value」のペアを入力します。
5. [新しい商品を作成] を選択します。

Git 同期製品を複数作成するには

1. AWS Service Catalog コンソールの左側のナビゲーションパネルで、「製品リスト」を選択し、「複数の Git 管理製品の作成」を選択します。
2. 「共通製品の詳細」を入力します。
3. 「外部リポジトリの詳細」で、AWS CodeStar 接続を選択し、リポジトリと分岐を指定します。
4. 「製品の追加」ペインで、「テンプレートファイルパス」と「製品名」を入力します。[新しいアイテムを追加] を選択し、必要に応じて製品の追加を続けます。
5. 必要な製品をすべて追加したら、[製品の一括作成] を選択します。

AWS Service Catalog 既存の製品を外部リポジトリに接続するには

1. AWS Service Catalog コンソールの左側のナビゲーションパネルで、[製品リスト] を選択し、[製品を外部リポジトリにConnect] を選択します。
2. 「製品を選択」ページで、外部リポジトリに接続する製品を選択し、「次へ」を選択します。
3. 「Specify source details」ページで、 AWS CodeStar 既存の接続を選択し、リポジトリ、ブランチ、テンプレートファイルパスを指定します。
4. [次へ] を選択します。
5. [確認して送信] ページで、接続の詳細を確認し、[製品を外部リポジトリに接続する] を選択します。

Git と同期された製品接続の表示

AWS Service Catalog コンソール、API、 AWS CLI またはを使用して、リポジトリ接続の詳細を表示できます。 AWS Service Catalog テンプレートファイルにリンクされている製品では、リポジトリ接続に関する情報と、前回テンプレートが製品と同期された時刻を Last Sync Status から取得できます。

> **Note**
>
> リポジトリ情報と最終同期ステータスは製品レベルで表示できます。リポジトリの詳細を表示するには、ユーザーは CodeConnections API の IAM 権限を持っている必要があります。これらの IAM 権限に必要なポリシーの詳細については、「 AWS Service Catalog 製品をテンプレートファイルに同期するために必要な権限」を参照してください。

接続とリポジトリの詳細を表示するには、 AWS Management Console

1. 左のナビゲーションパネルで製品リスト を選択します。
2. リストから製品を選択します。
3. 製品ページで、[製品ソースの詳細] セクションに移動します。
4. 製品バージョンのソースリビジョン ID を表示するには、「最終作成バージョン」リンクを選択します。「バージョン詳細」セクションには、ソースリビジョン ID が表示されます。

接続とリポジトリの詳細を表示するには: AWS CLI

から AWS CLI、以下のコマンドを実行します。

```
$ aws servicecatalog describe-product-as-admin

$ aws servicecatalog describe-provisioning-artifact

$ aws servicecatalog search-product-as-admin

$ aws servicecatalog list-provisioning-artifacts
```

Git 同期製品接続の更新

AWS Service Catalog コンソール、 AWS Service Catalog API、またはを使用して、既存のアカウント接続と GIT 同期製品を更新できます。 AWS CLI

AWS Service Catalog 既存の製品をテンプレートファイルに接続する方法については、「Git と同期した製品接続の新規作成」を参照してください。

既存の製品を Git 同期商品に更新するには

1. 左側のナビゲーションパネルで [製品リスト] を選択し、次のいずれかのオプションを選択します。
   - 1 つの製品 を更新するには、製品を選択し、「製品ソースの詳細」セクションに移動して「詳細を編集」を選択します。
   - 複数の製品 を更新するには、[製品を外部リポジトリに接続する] を選択し、最大 10 個の製品を選択してから [次へ] を選択します。
2. 「製品ソースの詳細」セクションで、以下の更新を行います。
   - 接続を指定します。
   - リポジトリを指定します。
   - 分岐を指定します。
   - テンプレートファイルに名前を付けます。
3. [変更の保存] を選択します。

> **Note**
> 外部リポジトリにまだ接続されていない製品の場合は、製品を選択した後、製品情報ページ上部のアラートに表示される [外部リポジトリに接続] オプションを使用できます。

AWS Service Catalog コンソールまたは TO を使用することもできます。 AWS CLI

- AWS Service Catalog 既存の製品を外部リポジトリのテンプレートファイルConnect する
- 製品名、説明、タグなどの製品メタデータを更新します。
- 以前接続していた AWS Service Catalog 製品の接続を再設定 (別のリポジトリソースを使用するように同期を更新) します。

AWS Service Catalog コンソールを使用して接続とリポジトリの詳細を更新するには

1. AWS Service Catalog コンソールの左側のナビゲーションパネルで [製品リスト] を選択し、現在外部リポジトリに接続されている製品を選択します。
2. 「製品ソース詳細」セクションで、「製品ソースを編集」を選択します。
3. 「製品ソース詳細」セクションで、希望する新しいリポジトリを指定します。
4. [変更の保存] を選択します。

接続とリポジトリの詳細を更新するには、 AWS CLI

から、 AWS CLI `$ aws servicecatalog update-product` `$ aws servicecatalog update-provisioning-artifact`およびコマンドを実行します。

Git 同期製品接続を削除する

AWS Service Catalog コンソール、 CodeConnections API、またはを使用して、 AWS Service Catalog 製品とテンプレートファイル間の接続を削除できます AWS CLI。製品をテンプレートファイルから切り離すと、 AWS Service Catalog 同期された製品は通常管理されている製品に切り替わります。製品を切断した後、以前に接続したリポジトリでテンプレートファイルを変更してコミットしても、変更は反映されません。 AWS Service Catalog 製品を外部リポジトリのテンプレートファイルに再接続するには、「接続と同期された製品の更新」を参照してください。 AWS Service Catalog

コンソールを使用して GIT 同期製品の接続を解除するには AWS Service Catalog

1. で AWS Management Console、左側のナビゲーションパネルから [製品リスト] を選択します。
2. リストから製品を選択します。
3. 製品ページで、[製品ソースの詳細] セクションに移動します。
4. [切断] を選択します。
5. アクションを確認し、[切断] を選択します。

以下を使用して GIT 同期製品の接続を解除するには AWS CLI

から AWS CLI、コマンドを実行します。`$ aws servicecatalog update-productConnectionParameters` 入力で、指定した接続を削除します。

CodeConnections API を使用して接続を削除するには、または AWS CLI

CodeConnections API またはで AWS CLI、`$ aws codestar-connections delete-connection`コマンドを実行します。

Terraform 製品を GitHub、 GitHub エンタープライズ、または Bitbucket のテンプレートファイルと同期する

Terraform 設定ファイルを使用して Git 同期製品を作成する場合、ファイルパスは tar.gz 形式のみを受け入れます。Terraform フォルダ形式はファイルパスでは使用できません。

AWS リージョン Git 同期製品のサポート

AWS Service Catalog AWS リージョン 以下の表に示すように、Git 同期製品をサポートします。

AWS リージョン 名前	AWS リージョン アイデンティティ	Git 同期製品のサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい
米国西部 (オレゴン)	us-west-2	はい
アフリカ (ケープタウン)	af-south-1	いいえ
アジアパシフィック (香港)	ap-east-1	いいえ
アジアパシフィック (ジャカルタ)	ap-southeast-3	いいえ
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	いいえ
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい

AWS リージョン 名前	AWS リージョン アイデンティティ	Git 同期製品のサポート
カナダ (中部)	ca-central-1	はい
欧州 (フランクフルト)	eu-central-1	はい
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	はい
欧州 (ミラノ)	eu-south-1	いいえ
欧州 (パリ)	eu-west-3	はい
欧州 (ストックホルム)	eu-north-1	はい
中東 (バーレーン)	me-south-1	いいえ
南米 (サンパウロ)	sa-east-1	Yes
AWS GovCloud (米国東部)	us-gov-east-1	No
AWS GovCloud (米国西部)	us-gov-west-1	No

製品の削除

製品を削除すると、AWS Service Catalog は、その製品を含むすべてのポートフォリオからすべての製品バージョンを削除します。

AWS Service Catalog は、AWS Service Catalog コンソールまたは AWS CLI を使用して製品を削除できます。製品を正常に削除するには、まず製品に関連付けられているすべてのリソースの関連付けを解除する必要があります。製品リソースの関連付けの例には、ポートフォリオの関連付け、予算 TagOptions、サービスアクションなどがあります。

> **Important**
>
> 製品が削除された後は、その製品を復元することはできません。

AWS Service Catalog コンソールを使用して製品を削除するには

1. ポートフォリオ ページに移動し、削除する製品を含むポートフォリオを選択します。
2. 削除する製品を選択してから、製品ペインの右上にある [削除] を選択します。
3. 関連リソースのない製品の場合は、テキストボックスに「削除」と入力して削除する製品を確認し、[削除] を選択します。

   関連リソースのある製品の場合は、ステップ 4 に進みます。
4. 「製品を削除」ウィンドウで、製品の関連リソースがすべて表示されている「関連付け」テーブルを確認します。AWS Service Catalog は、製品を削除したときに、これらのリソースの関連付けを解除しようとします。
5. テキストボックスに「削除」と入力して、製品を削除して関連するリソースをすべて削除することを確認します。
6. [関連付けを解除して削除] を選択します。

AWS Service Catalog が、製品のすべてのリソースの関連付けを解除できない場合、その製品は削除されません。「製品を削除」ウィンドウには、関連付け解除に失敗した回数と失敗ごとの説明が表示されます。製品削除時に失敗したリソースの関連付け解除の解決について詳しくは、以下の「製品削除時に失敗したリソースの関連付け解除の解決」を参照してください。

トピック

- AWS CLI を使用して製品を削除する
- 製品削除時に失敗したリソースの関連付け解除の解決

AWS CLI を使用して製品を削除する

AWS Service Catalog は、AWS Command Line Interface (AWS CLI) を使用してポートフォリオから製品を削除できます。AWS CLI は、コマンドラインシェルでコマンドを使用して AWS サービスとやり取りするためのオープンソースツールです。AWS Service Catalog 強制削除機能には AWS CLIエイリアスが必要です。エイリアスは、頻繁に使用するコマンドやスクリプトを短縮するために、AWS CLI で作成するショートカットです。

前提条件

- AWS CLI をインストールして設定します。詳細については、「AWS CLI の最新バージョンのインストールまたは更新」および「設定の基本」を参照してください。1.11.24 または 2.0.0 の最小 AWS CLI バージョンを使用してください。

- 製品の CLI エイリアスを削除するには、Bash 互換のターミナルと JQ コマンドライン JSON プロセッサが必要です。コマンドライン JSON プロセッサのインストールの詳細については、jq のダウンロード を参照してください。
- AWS CLI エイリアスを作成して Disassociation API 呼び出しをバッチ処理することで、1 つのコマンドで製品を削除できるようになります。

製品を正常に削除するには、まず製品に関連付けられているすべてのリソースの関連付けを解除する必要があります。製品リソースの関連付けの例としては、ポートフォリオの関連付け、予算、タグオプション、サービスアクションなどがあります。CLI を使用して製品を削除する場合、CLI `force-delete-product` エイリアスを使用して `Disassociate` APIを呼び出し、`DeleteProduct` APIを妨げるリソースの関連付けを解除できます。これにより、個別の関連付け解除を個別に呼び出す必要がなくなります。

> **Note**
>
> 以下の手順で示すファイルパスは、これらの操作を実行するために使用するオペレーティングシステムによって異なる場合があります。

AWS CLI エイリアスを作成して AWS Service Catalog 製品を削除する

AWS CLI を使用して AWS Service Catalog 製品を削除する場合、CLI `force-delete-product` エイリアスを使用して `Disassociate` API を呼び出し、`DeleteProduct` 呼び出しを妨げていたリソースの関連付けを解除できます。

AWS CLI 設定フォルダーに **`alias`** ファイルを作成します。

1. AWS CLI コンソールで、設定フォルダに移動します。デフォルトでは、設定フォルダは Linux または macOS では「`~/.aws/`」、Windows 上は「`%USERPROFILE%\.aws\`」にあります。
2. ファイルナビゲーションを使用するか、任意のターミナルで以下のコマンドを入力して、`cli` という名前のサブフォルダを作成します。

   ```
   $ mkdir -p ~/.aws/cli
   ```

   作成されるフォルダ「`cli`」のデフォルトパスは、Linux または macOS では「`~/.aws/cli/`」、Windows では「`%USERPROFILE%\.aws\cli`」上にあります。

3. 新しい cli フォルダで、ファイル拡張子なしの alias という名前のテキストファイルを作成します。ファイルナビゲーションを使用するか、任意のターミナルで以下のコマンドを入力して alias ファイルを作成できます。

```
$ touch ~/.aws/cli/alias
```

4. 1 行目に「[toplevel]」と 入力します。
5. ファイルを保存します。

次に、エイ force-delete-product リアススクリプトをaliasファイルに手動で貼り付けるか、ターミナルウィンドウで コマンドを使用して、エイリアスをファイルに追加できます。

alias ファイルに force-delete-product エイリアスを手動で追加する

1. AWS CLI コンソールで AWS CLI 設定フォルダーに移動し、alias ファイルを開きます。
2. ファイルの [toplevel] 行の下に次のコードエイリアスを入力します。

```
 [command servicecatalog]
force-delete-product =
  !f() {
    if [ "$#" -ne 1 ]; then
        echo "Illegal number of parameters"
        exit 1
    fi

    if [[ "$1" != prod-* ]]; then
       echo "Please provide a valid product id."
       exit 1
    fi

    productId=$1
    describeProductAsAdminResponse=$(aws servicecatalog describe-
product-as-admin --id $productId)
    listPortfoliosForProductResponse=$(aws servicecatalog list-
portfolios-for-product --product-id $productId)

    tagOptions=$(echo "$describeProductAsAdminResponse" | jq -r
 '.TagOptions[].Id')
```

```
                budgetName=$(echo "$describeProductAsAdminResponse" | jq -r
 '.Budgets[].BudgetName')
                portfolios=$(echo "$listPortfoliosForProductResponse" | jq -r
 '.PortfolioDetails[].Id')
                provisioningArtifacts=$(echo "$describeProductAsAdminResponse" | jq
 -r '.ProvisioningArtifactSummaries[].Id')
                provisioningArtifactServiceActionAssociations=()

                for provisioningArtifactId in $provisioningArtifacts; do
                  listServiceActionsForProvisioningArtifactResponse=$(aws
 servicecatalog list-service-actions-for-provisioning-artifact --product-id
 $productId --provisioning-artifact-id $provisioningArtifactId)
                  serviceActions=$(echo
 "$listServiceActionsForProvisioningArtifactResponse" | jq -r
 '[.ServiceActionSummaries[].Id] | join(",")')
                  if [[ -n "$serviceActions" ]]; then
                    provisioningArtifactServiceActionAssociations
+=("${provisioningArtifactId}:${serviceActions}")
                  fi
                done

                echo "Before deleting a product, the following associated resources
 must be disassociated. These resources will not be deleted. This action may take
 some time, depending on the number of resources being disassociated."

                echo "Portfolios:"
                for portfolioId in $portfolios; do
                  echo "\t${portfolioId}"
                done

                echo "Budgets:"
                if [[ -n "$budgetName" ]]; then
                  echo "\t${budgetName}"
                fi

                echo "Tag Options:"
                for tagOptionId in $tagOptions; do
                  echo "\t${tagOptionId}"
                done

                echo "Service Actions on Provisioning Artifact:"
                for association in
 "${provisioningArtifactServiceActionAssociations[@]}"; do
                  echo "\t${association}"
```

```
                done

                read -p "Are you sure you want to delete ${productId}? y,n "
                if [[ ! $REPLY =~ ^[Yy]$ ]]; then
                   exit
                fi

                for portfolioId in $portfolios; do
                  echo "Disassociating ${portfolioId}"
                  aws servicecatalog disassociate-product-from-portfolio --product-
id $productId --portfolio-id $portfolioId
                done

                if [[ -n "$budgetName" ]]; then
                  echo "Disassociating ${budgetName}"
                  aws servicecatalog disassociate-budget-from-resource --budget-
name "$budgetName" --resource-id $productId
                fi

                for tagOptionId in $tagOptions; do
                  echo "Disassociating ${tagOptionId}"
                  aws servicecatalog disassociate-tag-option-from-resource --tag-
option-id $tagOptionId --resource-id $productId
                done

                for association in
 "${provisioningArtifactServiceActionAssociations[@]}"; do
                  associationPair=(${association//:/ })
                  provisioningArtifactId=${associationPair[0]}
                  serviceActionsList=${associationPair[1]}
                  serviceActionIds=${serviceActionsList//,/ }
                  for serviceActionId in $serviceActionIds; do
                    echo "Disassociating ${serviceActionId} from
 ${provisioningArtifactId}"
                    aws servicecatalog disassociate-service-action-from-
provisioning-artifact --product-id $productId --provisioning-artifact-id
 $provisioningArtifactId --service-action-id $serviceActionId
                  done
                done

                echo "Deleting product ${productId}"
                aws servicecatalog delete-product --id $productId

              }; f
```

3. ファイルを保存します。

ターミナルウィンドウを使用してエイ force-delete-product リアスを **alias** ファイルに追加する

1. ターミナルウィンドウを開いて、次のコマンドを実行します。

   ```
   $ cat >> ~/.aws/cli/alias
   ```

2. エイリアススクリプトをターミナルウィンドウに貼り付け、CTRL+D キーを押して cat コマンドを終了します。

force-delete-product エイリアスを呼び出す

1. ターミナルウィンドウで次のコマンドを実行し、削除製品エイリアスを呼び出します

   ```
   $ aws servicecatalog force-delete-product {product-id}
   ```

   次の例は、force-delete-product エイリアスコマンドとその結果の応答を示しています。

   ```
   $ aws servicecatalog force-delete-product prod-123
   ```

   ```
   Before deleting a product, the following associated resources must
   be disassociated. These resources will not be deleted. This action may take some
   time, depending on the number of resources being disassociated.
   Portfolios:
     port-123
   Budgets:
       budgetName
   Tag Options:
       tag-123
   Service Actions on Provisioning Artifact:
       pa-123:act-123
   Are you sure you want to delete prod-123? y,n
   ```

2. y を入力し、製品を削除することを確認します。

製品を正常に削除すると、ターミナルウィンドウに次の結果が表示されます。

```
Disassociating port-123
Disassociating budgetName
Disassociating tag-123
Disassociating act-123 from pa-123
Deleting product prod-123
```

追加リソース

AWS CLI、エイリアスの使用、AWS Service Catalog 製品の削除の詳細については、次のリソースを参照してください。

- AWS CLI エイリアスの作成と使用は AWS Command Line Interface (CLI) ユーザーガイドに記載されています。
- AWS CLIエイリアスリポジトリ git リポジトリ。
- AWS Service Catalog 製品の削除。
- AWS re:Invent 2016: の実用AWS CLIユーザーYouTube。

製品削除時に失敗したリソースの関連付け解除の解決

リソースの関連付け解除の例外が原因で以前に製品を削除しようとして失敗した場合は、以下の例外リストとその解決策を確認してください。

> **Note**
>
> リソース関連付けの解除に失敗したというメッセージが表示される前に [製品の削除] ウィンドウを閉じた場合は、「製品の削除」セクションの手順 1 ~ 3 を実行して、ウィンドウを再度開くことができます。

リソースの関連付け解除に失敗した問題を解決するには

「製品を削除」ウィンドウで、「アソシエーション」テーブルの「ステータス」列を確認します。リソースの関連付け解除に失敗した例外と推奨される解決策を特定してください。

ステータス例外の種類	原因	解決方法
製品-****	AWS Service Catalog 製品にまだ TagOptions、予算、少なくとも 1 つのアクションProvisioningArtifact が関連付けられている、製品がポートフォリオにまだ割り当てられている、製品にユーザーが存在する、または製品に制約があるため、 は製品を削除できませんでした。	製品を再度削除してみます。
ユーザー: username には以下を実行する権限がありません:	製品を削除しようとしているユーザーには、製品のリソースの関連付けを解除するのに必要な権限がありません。	AWS Service Catalog では、現在関連付けを解除する権限を持たない製品リソースの関連付けの解除の詳細について、アカウント管理者に問い合わせることをお勧めします。

バージョンの管理

製品を作成するときに製品バージョンを割り当てます。製品バージョンはいつでも更新できます。

バージョンには、AWS CloudFormation テンプレート、タイトル、説明、ステータス、ガイダンスがあります。

バージョンステータス

バージョンには、次の 3 つのステータスのいずれかを指定できます。

- アクティブ - アクティブなバージョンがバージョンリストに表示され、ユーザーがそのバージョンを起動できるようにします。

- 非アクティブ - 非アクティブバージョンは、バージョンリストで非表示になります。このバージョンから起動された既存のプロビジョニング済み製品は影響を受けません。
- 削除済み - バージョンが削除されると、バージョンリストから削除されます。バージョンの削除は元に戻せません。

バージョンガイダンス

バージョンガイダンスを設定して、製品バージョンに関する情報をエンドユーザーに提供することができます。バージョンガイダンスは、アクティブな製品バージョンにのみ影響します。

バージョンガイダンスには、次の 2 つのオプションがあります。

- なし - デフォルトでは、製品バージョンにはガイダンスはありません。エンドユーザーはそのバージョンを使用して、プロビジョニングされた製品を更新および起動できます。
- 非推奨 - ユーザーは、非推奨の製品バージョンを使用して新しいプロビジョニング済み製品を起動することはできません。以前にリリースされたプロビジョニング済み製品が、現在廃止されたバージョンを使用している場合、ユーザーはそのプロビジョニング済み製品を既存のバージョンまたは新しいバージョンを使用してのみ更新できます。

バージョンの更新

製品を作成するときに製品バージョンを割り当てます。また、バージョンはいつでも更新できます。製品の作成に関する詳細については、「製品の作成」を参照してください。

製品バージョンを更新するには

1. AWS Service Catalog コンソールで、[製品] を選択します。
2. 製品リストから、バージョンを更新する製品を選択します。
3. [製品詳細] ページで、[バージョン] タブを選択し、更新するバージョンを選択します。
4. [バージョンの詳細] ページで、製品バージョンを編集し、[変更の保存] を選択します。

AWS Service Catalog 制約の使用

制約を適用して、エンドユーザーがポートフォリオを起動したときに特定のポートフォリオ内の製品に適用されるルールを制御します。エンドユーザーが製品を起動すると、制約を使用して適用したルールが表示されます。製品をポートフォリオに入れたら、製品に制約を適用できます。制約は、作成するとすぐに有効になり、起動されていない製品の現在のバージョンすべてに適用されます。

制約

- AWS Service Catalog の起動制約
- AWS Service Catalog 通知の制約
- AWS Service Catalog タグの更新の制約
- AWS Service Catalog スタックセットの制約
- AWS Service Catalog テンプレート制約

AWS Service Catalog の起動制約

起動制約は、エンドユーザーが製品を起動、更新、または終了するときに AWS Service Catalog が引き受ける AWS Identity and Access Management (IAM) ロールを指定します。IAM ロールは、ユーザーや AWS のサービスが、AWS のサービスを使用するために一時的に引き受けることができるアクセス権限のコレクションです。簡単な例については、以下を参照してください。

- AWS CloudFormation 製品タイプ: ステップ 6: IAM ロールを割り当てる起動制約を追加する
- Terraform Open Source または Terraform Cloud 製品タイプ: ステップ 5: 起動ロールを作成する

起動制約は、ポートフォリオ (製品 - ポートフォリオの関連付け) 内の製品に適用されます。起動制約は、ポートフォリオレベルやすべてのポートフォリオにわたる製品に対して適用されません。起動の制約をポートフォリオ内のすべての製品に関連付けるには、起動の制約を各製品に個別に適用する必要があります。

起動制約がない場合、エンドユーザーは各自の IAM 認証情報を使用して製品を起動し、管理する必要があります。そのためには、AWS CloudFormation、製品で使用される AWS のサービス、および AWS Service Catalog のアクセス許可が必要です。起動ロールを使用することにより、エンドユーザーのアクセス権限をその製品に必要な最小限のものに制限することができます。エンドユーザーのアクセス権限の管理の詳細については、「AWS Service Catalogにおけるアイデンティティとアクセスの管理」を参照してください。

IAM ロールを作成して割り当てるには、以下の IAM 管理者権限が必要です。

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:PassRole`
- `iam:Get*`

- `iam:List*`

起動ロールの設定

起動制約として製品に割り当てる IAM ロールには、以下を使用するためのアクセス権限が必要です。

Cloudformation 製品用

- `arn:aws:iam::aws:policy/AWSCloudFormationFullAccess` AWS CloudFormation マネージドポリシー
- 製品用の AWS CloudFormation テンプレートのサービス。
- サービス所有の Amazon S3 バケット内の AWS CloudFormation テンプレートへの読み取りアクセス。

Terraform 製品用

- 製品用の Amazon S3 テンプレートのサービス。
- サービス所有の Amazon S3 バケット内の Amazon S3 テンプレートへの読み取りアクセス。
- Amazon EC2 インスタンスでの `resource-groups:Tag` タグ付け用 (プロビジョニング操作の実行時に Terraform プロビジョニングエンジンが引き受けます)
- `resource-groups:CreateGroup` リソースグループのタグ付け用 (AWS Service Catalog がリソースグループを作成してタグを割り当てることで引き受けます)

IAM ロールには、AWS Service Catalog がロールを引き受けることができる信頼ポリシーが必要です。以下の手順では、ロールタイプとして AWS Service Catalog を選択すると信頼ポリシーが自動的に設定されます。コンソールを使用していない場合は、「IAM ロールでの信頼ポリシーの使用方法」の「ロールを引き受ける AWS サービスの信頼ポリシーの作成」セクションを参照してください。

> **Note**
>
> `servicecatalog:ProvisionProduct`、`servicecatalog:TerminateProvisionedProduct`、のアクセス権限を起動ロールで割り当てることはできません。「AWS Service Catalogエンドユーザーにアクセス許可を付与する」セクションのインラインポリシーの手順に示されているように、IAM ロールを使用する必要があります。

> **Note**
>
> プロビジョニングされた Cloudformation 製品とリソースを AWS Service Catalog コンソールに表示するには、エンドユーザーには AWS CloudFormation 読み取りアクセス権が必要です。プロビジョニングされた製品とリソースをコンソールで表示しても、起動ロールは使用されません。

起動ロールを作成するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。

   Terraform 製品には追加の起動ロール設定が必要です。詳細については、Terraform Open Source 製品入門 の「ステップ 5: 起動ロールの作成」を参照してください。
2. [ロール] を選択します。
3. [Create New Role (新しいロールを作成)] を選択します。
4. ロール名を入力し、[Next Step] を選択します。
5. [AWS Service Catalog] の隣の [AWSサービスロール] で、[選択] を選択します。
6. [Attach Policy] ページで、[Next Step] を選択します。
7. ロールを作成するには、[Create Role] を選択します。

ポリシーを新しいロールにアタッチするには

1. 作成したロールを選択して、[role details] ページを表示します。
2. [Permissions] タブを選択して、[Inline Policies] セクションを展開します。次に、[click here] を選択します。
3. [Custom Policy] を選択し、[Select] を選択します。
4. ポリシーの名前を入力し、[Policy Document] エディタに次のように貼り付けます。

```
     "Statement":[
{
   "Effect":"Allow",
   "Action":[
      "s3:GetObject"
   ],
   "Resource":"*",
```

```
            "Condition":{
               "StringEquals":{
                  "s3:ExistingObjectTag/servicecatalog:provisioning":"true"
               }
            }
      ]
}
```

> **Note**
>
> 起動制約の起動ロールを設定する場合は、`"s3:ExistingObjectTag/servicecatalog:provisioning":"true"` の文字列を使用する必要があります。

5. 製品で使用する追加のサービスごとに、ポリシーに行を追加します。例えば、Amazon Relational Database Service (Amazon RDS) のアクセス許可を追加するには、`Action` リストの最後の行の末尾にカンマを入力し、次の行を追加します。

```
"rds:*"
```

6. [ポリシーを適用] を選びます。

起動制約の適用

起動ロールを設定したら、起動制約として製品にロールを割り当てます。このアクションにより、エンドユーザーが製品を起動した際にロールを引き受けるよう AWS Service Catalog に指示します。

製品にロールに割り当てるには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 製品を含むポートフォリオを選択します。
3. [制約] タブを選択して、[制約の作成] を選択します。
4. [製品] から製品を選択し、[制約タイプ] の [起動] を選択します。[続行] を選択します。
5. [起動の制約] セクションでは、アカウントから IAM ロールを選択して IAM ロール ARN を入力するか、ロール名を入力できます。

ロール名を指定すると、アカウントが起動制約を使用する場合、アカウントは IAM ロールのその名前を使用します。このアプローチにより、起動ロールの制約をアカウントに依存しないようにできます。共有アカウントごとに作成するリソースを減らすことができます。

> **Note**
>
> 指定されたロール名は、起動制約を作成したアカウントと、この起動制約を使用して製品を起動するユーザーのアカウントに存在している必要があります。

6. IAM ロールを指定したら、[作成] を選択します。

混同代理人を起動制約に追加する

AWS Service Catalog は、Assume Role リクエストで実行される API 混同代理 保護をサポートします。起動制約を追加すると、起動ロールの信頼ポリシーの `sourceAccount` および `sourceArn` の条件を使用して、起動ロールのアクセスを制限できます。これにより、信頼できるソースから起動ロールが呼び出されるようになります。

次の例では、AWS Service Catalog エンドユーザーはアカウント 111111111111 に属しています。AWS Service Catalog 管理者が製品の `LaunchConstraint` を作成すると、エンドユーザーは起動ロールの信頼ポリシーに次の条件を指定して、引き受けロールをアカウント 111111111111 に制限できます。

```
"Condition":{
   "ArnLike":{
      "aws:SourceArn":"arn:aws:servicecatalog:us-east-1:111111111111:*"
   },
   "StringEquals":{
      "aws:SourceAccount":"111111111111"
   }

}
```

`LaunchConstraint` を使用して製品をプロビジョニングするユーザーは、同じ `AccountId` (111111111111) を持っている必要があります。そうでない場合、操作は `AccessDenied` エラーで失敗し、起動ロールの誤用を防ぐことができます。

混同代理による保護のため、 次の AWS Service Catalog API が保護されています。

- LaunchConstraint
- ProvisionProduct
- UpdateProvisionedProduct
- TerminateProvisionedProduct
- ExecuteProvisionedProductServiceAction
- CreateProvisionedProductPlan
- ExecuteProvisionedProductPlan

sourceArn 保護は、「arn:<aws-partition>:servicecatalog:<region>:<accountId>:」などのテンプレート化された ARN AWS Service Catalog のみをサポートします。特定のリソース ARN はサポートしていません。

起動制約の検証

AWS Service Catalog がロールを使用して製品を起動し、製品のプロビジョニングを正常に行えることを検証するには、AWS Service Catalog コンソールから製品を起動します。ユーザーに公開する前に制約をテストするには、同じ製品を含むテストポートフォリオを作成し、そのポートフォリオで制約をテストします。

製品を起動するには

1. AWS Service Catalog コンソールのメニューで、[Service Catalog]、[エンドユーザー] の順に選択します。
2. 製品を選択して、[製品の詳細] ページを開きます。[起動オプション] テーブルで、ロールの Amazon リソースネーム (ARN) が表示されることを確認します。
3. [製品の起動] を選択します。
4. 起動手順を続行して必要な情報を入力します。
5. 製品が正常に起動することを確認します。

AWS Service Catalog 通知の制約

> **Note**
>
> AWS Service Catalog は、Terraform Open Source 製品または Terraform Cloud 製品の通知制約をサポートしていません。

通知制約は、スタックのイベントに関する通知を受ける Amazon SNS トピックを指定します。

以下の手順を使用して、SNS トピックを作成しそれをサブスクライブします。

SNS トピックを作成しサブスクリプションするには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. [Create topic] (トピックの作成) を選択します。
3. トピック名を入力し、[Create topic] を選択します。
4. [サブスクリプションの作成] を選択します。
5. [Protocol] で [Email] を選択します。[Endpoint] では、通知を受信するために使用できる E メールアドレスを入力します。[サブスクリプションの作成] を選択します。
6. `AWS Notification - Subscription Confirmation` という件名の確認用 E メールを受信します。E メールを開き、指示に従ってサブスクリプションを完了します。

次の手順を使用して、前の手順で作成された SNS トピックを使用する通知の制約を適用します。

製品に通知の制約を適用するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 製品を含むポートフォリオを選択します。
3. [制約] を展開し、[制約を追加] を選択します。
4. [製品] で製品を選択し、[制約タイプ] を [通知] に設定します。[続行] を選択します。
5. [Choose a topic from your account] を選択して、[Topic Name] から作成した SNS トピックを選択します。
6. [送信] を選択します。

AWS Service Catalog タグの更新の制約

> **Note**
>
> AWS Service Catalog は、Terraform Open Source 製品のタグ更新制約をサポートしていません。

タグの更新の制約を使用すると、AWS Service Catalog 管理者は、プロビジョニング済み製品に関連付けられたリソースのタグの更新をエンドユーザーに許可または禁止できます。タグの更新が許可されている場合、 製品またはポートフォリオに関連付けられた新しいタグは、プロビジョニング済み製品の更新中にプロビジョニングされたリソースに適用されます。

製品に対するタグの更新を有効にするには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 更新する製品を含むポートフォリオを選択します。
3. [制約] タブを選択して、[制約の追加] を選択します。
4. [制約タイプ] で、[タグの更新] を選択します。
5. [製品] から製品を選択して、[続行] を選択します。
6. [タグの更新ページ] で、[タグの更新を有効にする] を選択します。
7. [送信] を選択します。

AWS Service Catalog スタックセットの制約

> **Note**
>
> - AWS Service Catalog は、Terraform Open Source 製品のスタックセット制約をサポートしません。
> - AutoTags は現在、 ではサポートされていませんAWS CloudFormation StackSets。

スタックセットの制約により、 AWS CloudFormation を使用して製品のデプロイオプションを設定できます StackSets。製品の起動で複数のアカウントおよびリージョンを指定できます。エンドユーザーは、これらのアカウントを管理し、製品のデプロイ場所とデプロイ順序を決定できます。

製品にスタックセットの制約を適用するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 必要な製品を含むポートフォリオを選択します。
3. [制約] タブを選択して、[制約の作成] を選択します。
4. [製品] で、製品を選択します。[制約タイプ] で、[スタックセット] を選択します。
5. スタックセットの制約のアカウント、リージョン、およびアクセス許可を設定します。

- [アカウント設定　] で、製品を作成するアカウントを指定します。
- [リージョンの設定] で、製品をデプロイする地理的リージョンと、それらの製品をそれらのリージョンにデプロイする順序を選択します。
- アクセス許可 で、ターゲットアカウントを管理する IAM StackSet 管理者ロールを選択します。ロールを選択しない場合、 StackSets はデフォルトの ARN を使用します。スタックセットのアクセス許可の設定の詳細については、こちらを参照してください。

6. [作成] を選択します。

AWS Service Catalog テンプレート制約

> **Note**
>
> AWS Service Catalog は、Terraform Open Source 製品または Terraform Cloud 製品のテンプレート制約をサポートしていません。

ユーザーが製品を起動するときに使用可能なオプションを制限するには、テンプレート制約を適用します。テンプレート制約を適用し、エンドユーザーが組織のコンプライアンス要件に違反することなく製品を使用できるようにします。製品へのテンプレート制約の適用は、AWS Service Catalog ポートフォリオで行います。テンプレート制約を定義するには、ポートフォリオに 1 つ以上の製品が含まれている必要があります。

テンプレート制約は 1 つ以上のルールで構成されます。これらのルールでは、製品の基盤となる AWS CloudFormation テンプレートで定義されているパラメータで許可される値の範囲を絞り込みます。AWS CloudFormation テンプレートのパラメータでは、スタックを作成するときにユーザーが指定できる値のセットを定義します。たとえば、パラメータで、EC2 インスタンスを含むスタックを起動するときにユーザーが選択できるさまざまなインスタンスタイプを定義します。

テンプレートのパラメータ値のセットが、ポートフォリオの対象者に対して広範囲すぎる場合、製品を起動するときにユーザーが選択できる値を制限するようテンプレート制約を定義できます。たとえば、テンプレートパラメータに、スモールインスタンスタイプ (`t2.micro` や `t2.small` など) のみを使用するユーザーにとって大きすぎる EC2 インスタンスタイプが含まれている場合は、エンドユーザーが選択できるインスタンスタイプを制限するテンプレート制約を追加できます。AWS CloudFormation テンプレートパラメータの詳細については、 AWS CloudFormation ユーザーガイドの「Parameters」を参照してください。

テンプレート制約はポートフォリオ内にバインドされます。1 つのポートフォリオで製品にテンプレート制約を適用し、別のポートフォリオに製品を含める場合、制約は 2 番目のポートフォリオの製品には適用されません。

既にユーザーと共有されている製品にテンプレート制約を適用する場合、制約は後続のすべての製品の起動と、ポートフォリオ内の製品のすべてのバージョンに対してすぐに有効になります。

ルールエディタを使用するか、AWS Service Catalog 管理者コンソールで JSON テキストとしてルールを書き込むことで、テンプレート制約ルールを定義します。構文と例を含むルールの詳細については、「テンプレート制約のルール」を参照してください。

ユーザーに公開する前に制約をテストするには、同じ製品を含むテストポートフォリオを作成し、そのポートフォリオで制約をテストします。

製品にテンプレート制約を適用するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [ポートフォリオ] ページで、テンプレート制約を適用する製品を含むポートフォリオを選択します。
3. [制約] セクションを展開し、[制約の追加] を選択します。
4. [商品とタイプの選択] ウィンドウの [製品] で、テンプレート制約を定義する製品を選択します。次に、[制約タイプ]で、[テンプレート] を選択します。[続行] を選択します。
5. [テンプレート制約ビルダー] ページで、JSON エディタまたはルールビルダーのインターフェイスを使用して制約ルールを編集します。

   - ルールの JSON コードを編集するには、[制約テキストエディタ] タブを選択します。このタブには、使用を開始するためにいくつかの例が含まれています。

     ルールビルダーのインターフェイスを使用してルールを構築するには、[ルールビルダー] タブを選択します。このタブで、製品向けにテンプレートで指定される任意のパラメータを選択し、そのパラメータで許可される値を指定できます。パラメータの種類に応じて、チェックリストで項目を選択する、数を指定する、またはカンマ区切りリストで値のセットを指定することで、許可される値を指定します。

     ルールの構築を終了したら、[ルールの追加] を選択します。[ルールビルダー] タブのテーブルにルールが表示されます。JSON 出力を確認して編集するには、[制約テキストビルダー] タブを選択します。
6. 制約のルールの編集を終了したら、[送信] を選択します。制約を表示するには、ポートフォリオの詳細ページに移動し、[制約] を展開します。

テンプレート制約のルール

AWS Service Catalog ポートフォリオでテンプレート制約を定義するルールでは、エンドユーザーがテンプレートをいつ使用できるかと、使用を試みている製品を作成するために使用される AWS CloudFormation テンプレートで宣言されるパラメータ用に指定できる値を示します。ルールは、エンドユーザーが意図せずに不適切な値を指定することを防ぐために役立ちます。たとえば、エンドユーザーが、特定の VPC で有効なサブネットを指定したかどうかや、テスト環境用に `m1.small` インスタンスタイプを使用したかどうかを確認するルールを追加できます。AWS CloudFormation は、ルールを使用して、製品のリソースを作成する前にパラメータ値を確認します。

各ルールは、ルール条件 (オプション) とアサーション (必須) の 2 つのプロパティで構成されます。ルール条件では、ルールがいつ有効になるかを決定します。アサーションでは、特定のパラメータにユーザーが指定できる値を示します。ルール条件を定義しない場合、ルールのアサーションが常に有効になります。ルール条件とアサーションを定義するには、ルール固有の組み込み関数を使用します。これは、テンプレートの `Rules` セクションでのみ使用できる関数です。関数をネストすることができますが、ルール条件またはアサーションの最終結果は、true または false である必要があります。

例として、`Parameters` セクションで VPC とサブネットパラメータを宣言したとします。特定のサブネットが特定の VPC 内にあることを検証するルールを作成できます。したがって、ユーザーが VPC を指定するときに、AWS CloudFormation はアサーションを評価して、サブネットのパラメータ値がその VPC にあるかどうか確認してから、スタックを作成または更新します。パラメータ値が無効の場合、AWS CloudFormation はスタックの作成または更新にすぐに失敗します。ユーザーが VPC を指定しない場合、AWS CloudFormation はサブネットのパラメータ値を確認しません。

構文

テンプレートの `Rules` セクションは、キーの名前 `Rules` とそれに続く単一のコロンで構成されます。ルールの宣言全体を中括弧で囲みます。複数のルールを宣言する場合は、カンマで区切ります。ルールごとに、引用符で囲んだ論理名、単一のコロン、およびルール条件とアサーションを囲む中括弧から成る形式で宣言します。

ルールには `RuleCondition` プロパティを含めることができ、`Assertions` プロパティを含める必要があります。ルールごとに、1 つのルール条件のみを定義できます。`Assertions` プロパティ内に 1 つ以上のアサーションを定義できます。次の擬似テンプレートに示すように、ルール固有の組み込み関数を使用してルール条件とアサーションを定義します。

```
"Rules":{
   "Rule01":{
```

```
      "RuleCondition":{
         "Rule-specific intrinsic function"
      },
      "Assertions":[
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
         },
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
         }
      ]
   },
   "Rule02":{
      "Assertions":[
         {
            "Assert":{
               "Rule-specific intrinsic function"
            },
            "AssertDescription":"Information about this assert"
         }
      ]
   }
}
```

擬似テンプレートには、Rules および Rule01 という 2 つのルールを含む Rule02 セクションが表示されます。Rule01 には、ルール条件と 2 つのアサーションが含まれます。ルール条件の関数が true に評価される場合、各アサーションの両方の関数が評価および適用されます。ルール条件が false の場合、ルールは有効になりません。Rule02 にはルール条件がないため、常に有効になります。つまり、1 つのアサーションが常に評価および適用されます。

ルール条件とアサーションを定義するルール固有の組み込み関数については、AWS CloudFormation ユーザーガイド の「AWSルール関数」を参照してください。

例: パラメータ値の条件付きの確認

次の 2 つのルールでは、InstanceType パラメータの値を確認します。Environment パラメータの値 (test または prod) に応じて、ユーザーは m1.small パラメータに対して m1.large また

は InstanceType を指定する必要があります。InstanceType および Environment パラメータは、同じテンプレートの Parameters セクションで宣言する必要があります。

```
"Rules" : {
  "testInstanceType" : {
    "RuleCondition" : {"Fn::Equals":[{"Ref":"Environment"}, "test"]},
    "Assertions" : [
      {
        "Assert" :  { "Fn::Contains" : [ ["m1.small"], {"Ref" : "InstanceType"} ] },
        "AssertDescription" : "For the test environment, the instance type must be
 m1.small"
      }
    ]
  },
  "prodInstanceType" : {
    "RuleCondition" : {"Fn::Equals":[{"Ref":"Environment"}, "prod"]},
    "Assertions" : [
      {
        "Assert" :  { "Fn::Contains" : [ ["m1.large"], {"Ref" : "InstanceType"} ] },
        "AssertDescription" : "For the prod environment, the instance type must be
 m1.large"
      }
    ]
  }
}
```

AWS Service Catalog のサービスアクション

> **Note**
>
> AWS Service Catalog は、Terraform Open Source または Terraform Cloud 製品のサービスアクションはサポートしていません。

AWS Service Catalog により、コンプライアンスとセキュリティ対策に従いながら、管理メンテナンスとエンドユーザートレーニングを減らすことができます。サービスアクションを使用すると、管理者は、AWS Service Catalog での運用タスクの実行、問題のトラブルシューティング、承認されたコマンドの実行、アクセス許可のリクエストをエンドユーザーに許可できます。AWS Systems Manager ドキュメントを使用して、サービスアクションを定義します。AWS Systems Manager ド

キュメントでは、Amazon EC2 の停止や再起動などの AWS ベストプラクティスを実装する事前定義されたアクションへのアクセスが提供されており、カスタムアクションを定義することもできます。

このチュートリアルでは、Amazon EC2 インスタンスの再起動をエンドユーザーに許可します。必要なアクセス許可を追加し、サービスアクションを定義して製品に関連付けたら、プロビジョニングされた製品でそのアクションを使用して、エンドユーザーのエクスペリエンスをテストします。

前提条件

このチュートリアルでは、AWS のフル管理アクセス許可を持ち、すでに AWS Service Catalog の使用に慣れていて、製品、ポートフォリオ、ユーザーの基本セットを所有していることを前提とします。AWS Service Catalog の使用に慣れていない場合は、このチュートリアルを使用する前に「設定」と「開始方法」のタスクを完了してください。

トピック

ステップ 1: エンドユーザーのアクセス許可を設定する

エンドユーザーは、特定のサービスアクションを表示および実行するアクセス許可が必要です。この例では、AWS Service Catalog のサービスアクション機能にアクセスして Amazon EC2 を再起動するアクセス許可がエンドユーザーに必要です。

アクセス許可を更新するには

1. AWS Identity and Access Management IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. メニューからユーザーグループを見つけます。
3. エンドユーザーが AWS Service Catalog リソースへのアクセスに使用するグループを選択します。この例では、エンドユーザーグループを選択します。独自の実装では、該当するエンドユーザーによって使用されるグループを選択します。

4. グループの詳細ページの [アクセス許可] タブで、新しいポリシーを作成するか、既存のポリシーを編集します。この例では、グループの AWS Service Catalog プロビジョニングおよび終了アクセス許可用に作成されたカスタムポリシーを選択して、既存のポリシーにアクセス許可を追加します。
5. [ポリシー] ページで、[ポリシーの編集] を選択して必要なアクセス許可を追加します。ビジュアルエディタまたは JSON エディタを使用してポリシーを編集できます。この例では、JSON エディタを使用してアクセス許可を追加します。このチュートリアルでは、以下のアクセス許可をポリシーに追加します。

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "Stmt1536341175150",
   "Action": [
    "servicecatalog:ListServiceActionsForProvisioningArtifact",
    "servicecatalog:ExecuteprovisionedProductServiceAction",
    "ssm:DescribeDocument",
    "ssm:GetAutomationExecution",
    "ssm:StartAutomationExecution",
    "ssm:StopAutomationExecution",
    "cloudformation:ListStackResources",
    "ec2:DescribeInstanceStatus",
    "ec2:StartInstances",
    "ec2:StopInstances"
   ],
   "Effect": "Allow",
   "Resource": "*"
  }
 ]
}
```

6. ポリシーを編集した後、ポリシーの変更を確認して承認します。これで、エンドユーザーグループのユーザーに、AWS Service Catalog で Amazon EC2 の再起動アクションを実行するアクセス許可が付与されました。

ステップ 2: サービスアクションを作成する

次は、Amazon EC2 インスタンスを再起動するサービスアクションを作成します。

1. AWS Service Catalog コンソール (https://console.aws.amazon.com/s3/) を開きます。
2. メニューの [サービスアクション] を選択します。
3. [サービスアクション] ページで、[アクションの作成] を選択します。
4. [Create action] ページで、サービスアクションを定義する AWS Systems Manager ドキュメントを選択します。Amazon EC2 インスタンスの再起動アクションは AWS Systems Manager ドキュメントによって定義されているため、ドロップダウンメニューのデフォルトのオプションである [Amazon ドキュメント] をそのまま使用します。
5. AWS-RestartEC2Instance アクションを検索して選択します。
6. お客様の環境とチームに合ったアクションの名前と説明を指定します。この説明はエンドユーザーに表示されるため、アクションの内容を理解するのに役立つものを選択してください。
7. [Parameter and target configuration] で、アクションのターゲットとなる SSM ドキュメントパラメータ ([インスタンス ID] など) を選択し、パラメータのターゲットを選択します。パラメータを追加するには、[パラメータの追加] を選択します。
8. [許可] で、ロールを選択します。この例では、デフォルトのアクセス許可を使用します。他のアクセス許可の設定も可能で、このページで定義します。
9. 設定を確認したら、[アクションの作成] を選択します。
10. 次のページでは、アクションが作成されて使用可能になると確認メッセージが表示されます。

ステップ 3: サービスアクションを製品バージョンに関連付ける

アクションを定義したら、そのアクションを製品に関連付ける必要があります。

1. [サービスアクション] ページで、[AWS-RestartEC2instance]を選択して、[アソシエイトアクション] の順に選択します。
2. [アクションの関連付け] ページで、エンドユーザーによってサービスアクションが実行されるようにする製品を選択します。この例では、[Linux Desktop] を選択します。
3. 製品バージョンを選択します。1 番上のチェックボックスを使用して、すべてのバージョンを選択できます。
4. [アクションの関連付け] を選択します。
5. 次のページで、確認メッセージが表示されます。

これで、AWS Service Catalog でサービスアクションが作成されました。このチュートリアルの次のステップは、エンドユーザーとしてサービスアクションを使用することです。

ステップ 4: エンドユーザーのエクスペリエンスをテストする

エンドユーザーはプロビジョニングされた製品に対してサービスアクションを実行できます。このチュートリアルの目的上、エンドユーザーには少なくとも 1 つのプロビジョニングされた製品が必要です。プロビジョニングされた製品は、前のステップでサービスアクションに関連付けた製品バージョンから起動する必要があります。

エンドユーザーとしてサービスアクションにアクセスするには

1. エンドユーザーとして AWS Service Catalog コンソールにログインします。
2. AWS Service Catalog ダッシュボードのナビゲーションペインで、[プロビジョニングされた製品のリスト] を選択します。このリストには、エンドユーザーのアカウント用にプロビジョンされた製品が表示されます。
3. [プロビジョニングされた製品のリスト] ページで、プロビジョニングされたインスタンスを選択します。
4. [プロビジョニングされた製品の詳細] ページで、右上の [アクション] を選択してから、[AWS-RestartEC2instance] アクションを選択します。
5. カスタムアクションを実行することを確認します。アクションが送信されたという確認メッセージが表示されます。

ステップ 5: AWS CloudFormation によるサービスアクションの管理

サービスアクションと AWS CloudFormation リソースとの関連付けを作成できます。詳細については、AWS CloudFormationユーザーガイドで次を参照してください。

- AWS::ServiceCatalog::CloudFormation製品 ProvisioningArtifactProperties
- AWS::ServiceCatalog::ServiceAction関連付け

> **Note**
>
> サービスアクションと AWS CloudFormation リソースとの関連付けを管理する場合は、AWS Command Line Interface または AWS Management Console を使用してサービスアクションを追加したり削除したりしないでください。スタックの更新を実行すると、AWS CloudFormation の外部で行われたサービスアクションへの変更はすべて置き換えられます。

ステップ 6: トラブルシューティング

サービスアクションの実行が失敗した場合、[プロビジョニング済み製品] ページのサービスアクション実行イベントの [出力] セクションにエラーメッセージが表示されます。以下に、よくあるエラーメッセージの説明を示します。

> **Note**
>
> エラーメッセージの正確なテキストは変更される可能性があるため、いずれの種類の自動化プロセスでも使用しないでください。

内部エラー

AWS Service Catalog で内部エラーが発生しました。後でもう一度お試しください。問題が解決しない場合は、カスタマーサポートまでお問い合わせください。

StartAutomationExecution オペレーションの呼び出し時にエラー (ThrottlingException) が発生しました

サービスアクションの実行が SSM などのバックエンドサービスによって制限されました。

ロールの引き受け中にアクセスが拒否されました

AWS Service Catalog は、サービスアクション定義で指定されたロールを引き受けることができませんでした。servicecatalog.amazonaws.com プリンシパル、または servicecatalog.us-east-1.amazonaws.com などのリージョン別プリンシパルがロールの信頼ポリシーで許可リストに登録されていることを確認してください。

StartAutomationExecution オペレーションを呼び出すときにエラー (AccessDeniedException) が発生しました: ユーザーには、リソースStartAutomationExecution に対して ssm: を実行する権限がありません。

サービスアクション定義で指定されたロールに、ssm: を呼び出すアクセス許可がありません StartAutomationExecution。ロールに適切な SSM アクセス許可があることを確認してください。

プロビジョニング済み製品*TargetType*でタイプ のリソースが見つからない

プロビジョニングされた製品に、SSM ドキュメントで指定されたターゲットタイプと一致するリソース (AWS::EC2::Instance など) が含まれていません。プロビジョニングされた製品でこれらのリソースを確認するか、ドキュメントが正しいことを確認してください。

その名前のドキュメントは存在しません

サービスアクション定義で指定されたドキュメントが存在しません。

SSM オートメーションドキュメントの定義の取得に失敗しました

指定したドキュメントの定義を取得しようとしたときに、AWS Service Catalog が SSM からの不明な例外を検出しました。

ロールの認証情報の取得に失敗しました

AWS Service Catalog は指定されたロールの引き受け中に不明なエラーを検出しました。

パラメータの値***InvalidValue***「」が ***{ValidValue1}#{ValidValue2}*** にありません

SSM に渡されたパラメータ値がドキュメントの許容値のリストにありません。渡されたパラメータが有効であることを確認し、再試行してください。

パラメータのタイプに誤りがあります。に指定された値は有効な文字列***ParameterName***ではありません。

SSM に渡されたパラメータの値がドキュメントのこのタイプでは無効です。

パラメータがサービスアクション定義で定義されていません

サービスアクション定義で定義されていないパラメータが AWS Service Catalog に渡されました。使用できるのは、サービスアクション定義で定義されたパラメータのみです。

アクションの実行中またはキャンセル中にステップが失敗します。***Error message.*** 診断の詳細については、「オートメーションサービストラブルシューティングガイド」を参照してください。

SSM オートメーションドキュメントのステップが失敗しました。さらにトラブルシューティングするには、メッセージ内のエラーを参照してください。

パラメータの次の値は、プロビジョニング済み製品に含まれていないため、使用できません。***InvalidResourceId***

プロビジョニングされた製品にないリソースに対するアクションをユーザーがリクエストしました。

TargetType SSM オートメーションドキュメントに対して定義されていません

サービスアクションでは、SSM オートメーションドキュメントに が TargetType 定義されている必要があります。SSM オートメーションドキュメントを確認してください。

ポートフォリオへの AWS Marketplace 製品の追加

AWS Marketplace 製品をポートフォリオに追加し、それらの製品を AWS Service Catalog のエンドユーザーが使用するようにできます。

AWS Marketplace は、さまざまなソフトウェアやサービスを見つけ、サブスクライブし、すぐに使用を開始できるオンラインストアです。AWS Marketplace の製品のタイプには、データベース、アプリケーションサーバー、テストツール、モニタリングツール、コンテンツ管理ツール、ビジネスインテリジェンスソフトウェアなどがあります。AWS Marketplace は https://aws.amazon.com/marketplace で使用できます。Software as a Service (SaaS) 製品を AWS Marketplace から AWS Service Catalog に追加することはできないことに注意してください。

AWS Marketplace 製品を AWS Service Catalog エンドユーザーに配布するには、AWS CloudFormation テンプレートを含む製品を AWS Service Catalog にコピーし、製品をポートフォリオに追加します。

> **Note**
>
> AWS Service Catalog は、Terraform Open Source または Terraform Cloud 製品テンプレートを使用した AWS Service Catalog エンドユーザーへの AWS Marketplace 製品の配布をサポートしていません。

AWS Marketplace は、AWS Service Catalog を直接サポートすることも、登録して手動で追加することもできます。AWS Service Catalog 用に特別に設計された機能を使用して製品を追加することをお勧めします。

AWS Service Catalog を使用した AWS Marketplace 製品の管理

カスタムインターフェイスを使用して、登録した AWS Marketplace を AWS Service Catalog に直接追加できます。AWS Marketplace で、[サービスカタログ] を選択します。詳細については、AWS Marketplaceヘルプと FAQ の「AWS Service Catalog への製品のコピー」を参照してください。

手動での AWS Marketplace 製品の管理と追加

AWS Marketplace 製品にサブスクライブするには、次のステップを完了し、その製品を AWS CloudFormation テンプレートで定義して、AWS Service Catalog ポートフォリオにテンプレートを追加します。

AWS Marketplace 製品をサブスクライブするには

1. AWS Marketplace (https://aws.amazon.com/marketplace) に移動します。
2. 製品を参照するか、AWS Service Catalog ポートフォリオに追加する製品を検索します。製品を選択して、製品の詳細ページを表示します。
3. [続ける] を選択して受理ページを表示し、[手動で起動] タブを選択します。

   受理ページの情報には、サポートされる Amazon Elastic Compute Cloud (Amazon EC2) インスタンスタイプ、サポートされる AWS リージョン、および製品が各 AWS リージョン用に使用する Amazon マシンイメージ (AMI) ID が含まれます。選択内容によってはコストに影響する場合もあります。この情報を使用して、後のステップで AWS CloudFormation テンプレートをカスタマイズします。
4. 製品をサブスクライブするには、[Accept Terms] を選択します。

   製品をサブスクライブすると、[Your Software] を選択し、製品を選択することにより AWS Marketplace の製品の受理ページの情報にいつでもアクセスできます。

AWS CloudFormation テンプレートで AWS Marketplace 製品を定義するには

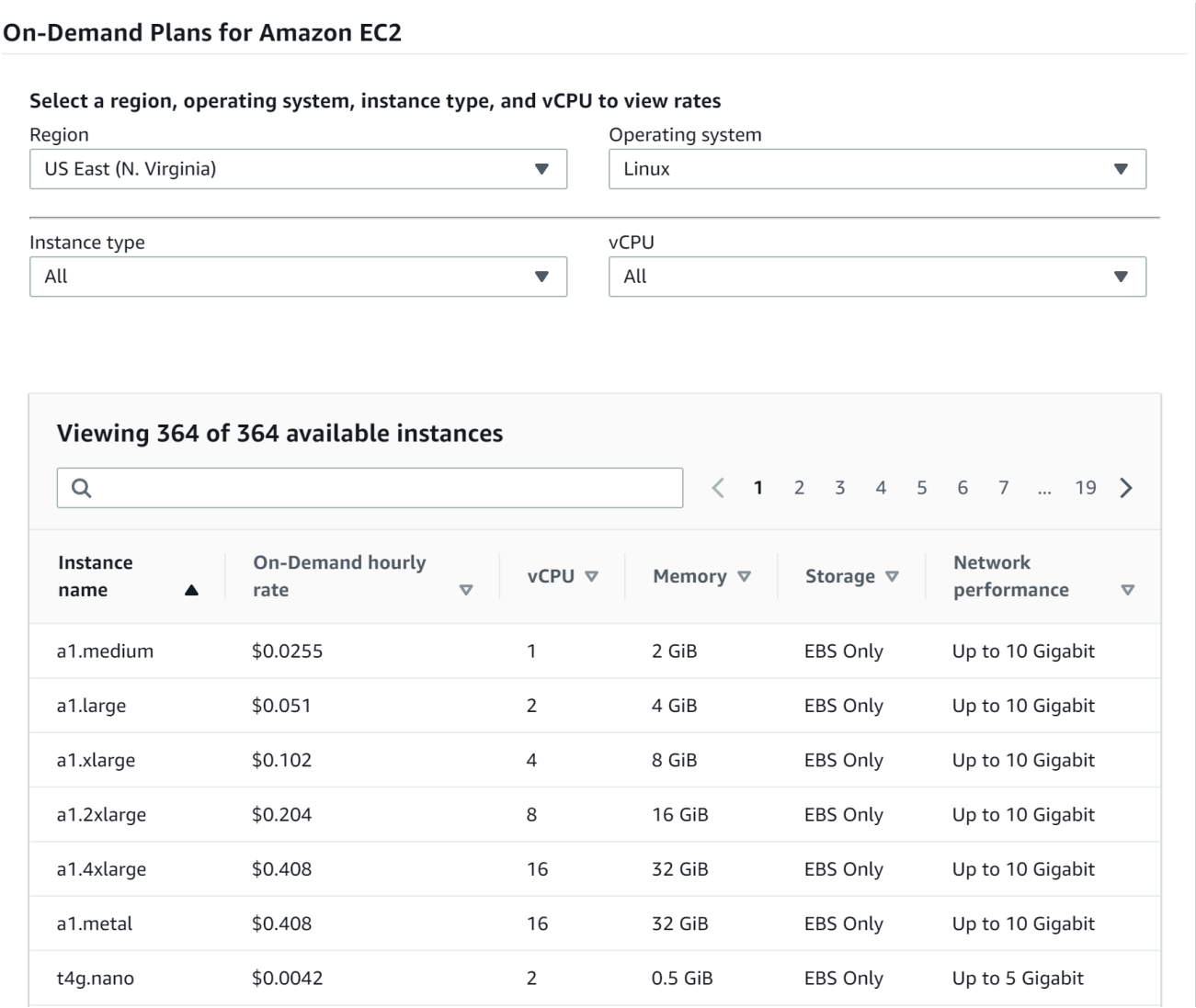
次のステップを完了するには、開始点として AWS CloudFormation サンプルテンプレートの 1 つを使用し、テンプレートをカスタマイズして、AWS Marketplace 製品を表すようにします。サンプルテンプレートにアクセスするには、AWS CloudFormationユーザーガイドの「サンプルテンプレート」を参照してください。

1. AWS CloudFormation ユーザーガイドの「サンプルテンプレート」ページで、製品の AWS リージョンを選択します。AWS リージョンは AWS Marketplace 製品でサポートされている必要があります。サポートされているリージョンは、AWS Marketplace の製品受理のページで表示できます。
2. リージョンに適したサービスサンプルテンプレートのリストを表示するには、[サービス] リンクを選択します。
3. 開始点として、ニーズに適している任意のサンプルを使用できます。この手順のステップでは、[Amazon EC2 instance in a security group] テンプレートを使用します。サンプルテンプレートを表示するには、[View] を選択し、テンプレートのコピーをローカルに保存して、編集できるようにします。ローカルファイルには、`.template` 拡張子が必要です。
4. テキストエディタでテンプレートを開きます。
5. テンプレートの上部の説明をカスタマイズします。ダッシュボードの外観は以下の例のようになっています。

```
"Description": "Launches a LAMP stack from AWS Marketplace",
```

6. InstanceType パラメータをカスタマイズし、製品でサポートされる EC2 インスタンスタイプのみを含むようにします。サポートされていない EC2 インスタンスタイプがテンプレートに含まれる場合、製品はエンドユーザーに対して起動しません。

   a. AWS Marketplace の製品受理のページの [料金詳細] セクションで、サポートされる EC2 インスタンスタイプを表示します。



On-Demand Plans for Amazon EC2

Select a region, operating system, instance type, and vCPU to view rates

Region: US East (N. Virginia)

Operating system: Linux

Instance type: All

vCPU: All

Viewing 364 of 364 available instances

1 2 3 4 5 6 7 ... 19

Instance name	On-Demand hourly rate	vCPU	Memory	Storage	Network performance
a1.medium	$0.0255	1	2 GiB	EBS Only	Up to 10 Gigabit
a1.large	$0.051	2	4 GiB	EBS Only	Up to 10 Gigabit
a1.xlarge	$0.102	4	8 GiB	EBS Only	Up to 10 Gigabit
a1.2xlarge	$0.204	8	16 GiB	EBS Only	Up to 10 Gigabit
a1.4xlarge	$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
a1.metal	$0.408	16	32 GiB	EBS Only	Up to 10 Gigabit
t4g.nano	$0.0042	2	0.5 GiB	EBS Only	Up to 5 Gigabit

   b. テンプレートで、デフォルトのインスタンスタイプを、サポートされている任意の EC2 インスタンスタイプに変更します。

   c. AllowedValues リストを編集し、製品でサポートされている EC2 インスタンスタイプのみを含むようにします。

d. エンドユーザーが AllowedValues リストから製品を起動するときに、使用しないようにする EC2 インスタンスタイプを削除します。

InstanceType パラメータの編集を終了した例を次に示します。

```
    "InstanceType" : {
      "Description" : "EC2 instance type",
      "Type" : "String",
      "Default" : "m1.small",
      "AllowedValues" : [ "t1.micro", "m1.small", "m1.medium", "m1.large",
"m1.xlarge", "m2.xlarge", "m2.2xlarge", "m2.4xlarge", "c1.medium", "c1.xlarge",
"c3.large", "c3.large", "c3.xlarge", "c3.xlarge", "c3.4xlarge", "c3.8xlarge" ],
      "ConstraintDescription" : "Must be a valid EC2 instance type."
    },
```

7. テンプレートの Mappings セクションで、サポートされる EC2 インスタンスタイプとアーキテクチャのみが含まれるように、AWSInstanceType2Arch マッピングを編集します。

a. AllowedValues パラメータの InstanceType リストに含まれていないすべての EC2 インスタンスタイプを削除して、マッピングのリストを編集します。

b. 各 EC2 インスタンスタイプの Arch の値を編集し、製品でサポートされるアーキテクチャータイプとなるようにします。有効な値は、PV64、HVM64、HVMG2 です。製品でサポートされるアーキテクチャの詳細については、「AWS Marketplace」のサポート製品詳細ページを参照してください。EC2 インスタンスファミリーでサポートされるアーキテクチャについては、「Amazon Linux AMI インスタンスタイプのマトリックス」を参照してください。

AWSInstanceType2Arch マッピングの編集が完了した例を次に示します。

```
    "AWSInstanceType2Arch" : {
      "t1.micro"    : { "Arch" : "PV64"  },
      "m1.small"    : { "Arch" : "PV64"  },
      "m1.medium"   : { "Arch" : "PV64"  },
      "m1.large"    : { "Arch" : "PV64"  },
      "m1.xlarge"   : { "Arch" : "PV64"  },
      "m2.xlarge"   : { "Arch" : "PV64"  },
      "m2.2xlarge"  : { "Arch" : "PV64"  },
      "m2.4xlarge"  : { "Arch" : "PV64"  },
      "c1.medium"   : { "Arch" : "PV64"  },
```

```
      "c1.xlarge"   : { "Arch" : "PV64"  },
      "c3.large"    : { "Arch" : "PV64"  },
      "c3.xlarge"   : { "Arch" : "PV64"  },
      "c3.2xlarge"  : { "Arch" : "PV64"  },
      "c3.4xlarge"  : { "Arch" : "PV64"  },
      "c3.8xlarge"  : { "Arch" : "PV64"  }
    }
,
```

8. テンプレートの Mappings セクションで、AWSRegionArch2AMI マッピングを編集し、各 AWS リージョンを、製品の対応するアーキテクチャと AMI ID に関連付けます。

   a. AWS Marketplace の製品受理のページで、次の例のように、製品が AWS の各リージョンで使用する AMI ID を表示します。

Region	ID	
US East (N. Virginia)	ami-	Launch with EC2 Console
US West (Oregon)	ami-	Launch with EC2 Console
US West (N. California)	ami-	Launch with EC2 Console
EU (Frankfurt)	ami-	Launch with EC2 Console
EU (Ireland)	ami-	Launch with EC2 Console
Asia Pacific (Singapore)	ami-	Launch with EC2 Console
Asia Pacific (Sydney)	ami-	Launch with EC2 Console
Asia Pacific (Tokyo)	ami-	Launch with EC2 Console
South America (Sao Paulo)	ami-	Launch with EC2 Console

   b. テンプレートで、サポートしないすべての AWS リージョンのマッピングを削除します。

   c. 各リージョンのマッピングを編集し、サポートされないアーキテクチャ (PV64、HVM64、または HVMG2) と、関連する AMI ID を削除します。

   d. 残りの各 AWS リージョンとアーキテクチャのマッピングで、AWS Marketplace の製品詳細ページから、対応する AMI ID を指定します。

   AWSRegionArch2AMI マッピングの編集が完了したコード例を次に示します。

```
    "AWSRegionArch2AMI" : {
      "us-east-1"        : {"PV64" : "ami-nnnnnnnn"},
      "us-west-2"        : {"PV64" : "ami-nnnnnnnn"},
      "us-west-1"        : {"PV64" : "ami-nnnnnnnn"},
      "eu-west-1"        : {"PV64" : "ami-nnnnnnnn"},
```

```
    "eu-central-1"     : {"PV64" : "ami-nnnnnnnn"},
    "ap-northeast-1"   : {"PV64" : "ami-nnnnnnnn"},
    "ap-southeast-1"   : {"PV64" : "ami-nnnnnnnn"},
    "ap-southeast-2"   : {"PV64" : "ami-nnnnnnnn"},
    "sa-east-1"        : {"PV64" : "ami-nnnnnnnn"}
  }
```

これで、テンプレートを使用して製品を AWS Service Catalog ポートフォリオに追加できます。追加の変更が必要な場合は、テンプレートの詳細について「AWS CloudFormation テンプレートの使用」を参照してください。

AWS Marketplace 製品を AWS Service Catalog ポートフォリオに追加するには

1. AWS Management Console にサインインし、AWS Service Catalog 管理者コンソール (https://console.aws.amazon.com/servicecatalog/) に移動します。
2. [ポートフォリオ] ページで、AWS Marketplace 製品を追加するポートフォリオを選択します。
3. ポートフォリオの詳細ページで、[新製品の更新] を選択します。
4. リクエストされた製品とサポートの詳細を入力します。
5. [Version details] ページで、[Upload a template file]、[Browse]、テンプレートファイルの順に選択します。
6. バージョンタイトルと説明を入力します。
7. [次へ] を選択します。
8. [Review] ページで、概要が正確であることを確認し、[確認とアップロード] を選択します。製品がポートフォリオに追加されます。これで、ポートフォリオにアクセスするエンドユーザーが製品を使用できるようになりました。

の使用 AWS CloudFormation StackSets

> **Note**
>
> AutoTags は現在、 ではサポートされていませんAWS CloudFormation StackSets。

を使用してAWS CloudFormation StackSets 、複数の AWS リージョンおよびアカウントでAWS Service Catalog製品を起動できます。AWS リージョン内で製品を連続してデプロイする順序を指定することができます。製品はアカウント間で並列にデプロイされます。ユーザーは起動時に、

障害耐性と、同時にデプロイするアカウントの最大数を指定できます。詳細については、「AWS CloudFormation の使用 StackSets」を参照してください。

スタックセットとスタックインスタンス

スタックセットでは、1 つの AWS CloudFormation テンプレートを使用して、複数の AWS リージョンの AWS アカウントにスタックを作成できます。

スタックインスタンスは、AWS リージョン内のターゲットアカウントのスタックのことであり、1 つのスタックセットのみと関連付けられます。

詳細については、「StackSets の概念」を参照してください。

スタックセットの制約

AWS Service Catalog では、スタックセットの制約を使用して製品のデプロイオプションを設定できます。

AWS Service Catalog は、 (米国西部) と AWS GovCloud (US) Regions AWS GovCloud （ AWS GovCloud 米国東部) の 2 つの の製品でスタックセット制約をサポートします。

詳細については、「AWS Service Catalog スタックセットの制約」を参照してください。

予算の管理

AWS Budgets を使用して、AWS Service Catalog 内のサービスのコストと使用状況を追跡できます。予算を AWS Service Catalog 製品やポートフォリオに関連付けることができます。

> **Note**
>
> AWS Service Catalog は、Terraform Open Source 製品の予算をサポートしていません。

AWS Budgets には、カスタム予算を設定して、コストまたは使用量が予算額や予算量を超えたとき (あるいは、超えると予測されたとき) にアラートを発信できる機能が用意されています。AWS Budgets の詳細については、「https://aws.amazon.com/aws-cost-management/aws-budgets」を参照してください。

タスク

前提条件

AWS Budgets を使用する前に、AWS Billing and Cost Management コンソールでコスト配分タグをアクティブ化する必要があります。詳細については、AWS Billing and Cost Management ユーザーガイドの「ユーザー定義のコスト配分タグのアクティブ化」を参照してください。

> **Note**
>
> タグがアクティブ化されるまでに最大 24 時間かかります。

また、Budgets 機能を使用するすべてのユーザーまたはグループに対して、AWS Billing and Cost Management コンソールへのユーザーアクセスを有効にする必要があります。これを行うには、ユーザー用の新しいポリシーを作成します。

ユーザーが予算を作成できるようにするには、請求情報の表示もユーザーに許可する必要があります。Amazon SNS 通知を使用する場合は、以下のポリシー例に示すように、ユーザーに Amazon SNS 通知を作成する権限を与えることができます。

予算ポリシーを作成するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. ナビゲーションペインで、ポリシー を選択します。
3. コンテンツペインで、[ポリシーの作成] を選択します。
4. [JSON] タブを選択し、以下の JSON ポリシードキュメントからテキストをコピーします。このテキストを [JSON] ボックスに貼り付けます。

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

```
        {
            "Sid": "Stmt1435216493000",
            "Effect": "Allow",
            "Action": [
                "aws-portal:ViewBilling",
                "aws-portal:ModifyBilling",
                "budgets:ViewBudget",
                "budgets:ModifyBudget"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Stmt1435216552000",
            "Effect": "Allow",
            "Action": [
                "sns:*"
            ],
            "Resource": [
                "arn:aws:sns:us-east-1"
            ]
        }
    ]
}
```

5. 完了したら、[ポリシーの確認] (ポリシーの確認) を選択します。構文エラーがある場合は、Policy Validator (ポリシー検証) によってレポートされます。
6. [確認] ページで、ポリシーに名前を付けます。ポリシーの [Summary (概要)] で、ポリシーによって割り当てられたアクセス許可を確認し、[ポリシーの作成] を選択して作業を保存します。

   新しいポリシーが管理ポリシーの一覧に表示され、ユーザーやグループにアタッチできるようになります。詳細については、AWS Identity and Access Management ユーザーガイドの「カスタマー管理ポリシーの作成と添付」を参照してください。

予算の作成

AWS Service Catalog 管理者コンソールの [製品リスト] ページと [ポートフォリオ] ページには、既存の製品とポートフォリオに関する情報が表示され、それらに対してアクションを実行できます。予算を作成するには、まず予算を関連付ける製品またはポートフォリオを決定します。

予算を作成するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [製品リスト] または [ポートフォリオ] を選択します。
3. 予算を追加する製品またはポートフォリオを選択します。
4. [アクション] メニューを開き、[予算を作成] を選択します。
5. [予算の作成] ページで、1 つのタグタイプを予算に関連付けます。

   タグには、 AutoTags と の 2 つのタイプがあります TagOptions。 AutoTags は、製品を起動したポートフォリオ、製品、およびユーザーを識別します。 は、これらのタグをプロビジョニングされたリソースに自動的にAWS Service Catalog適用します。 TagOption は、 で管理される管理者定義のキーと値のペアですAWS Service Catalog。

   ポートフォリオまたは製品で発生する支出が関連付けられた予算に反映されるには、両方に同じタグが必要です。初めて使用されたタグキーは、アクティブ化されるまでに 24 時間かかることがあります。詳細については、「the section called “前提条件”」を参照してください。
6. [AWS Budgets で作成] を選択します。[予算の設定] ページに移動します。「予算を作成」の手順に従って、予算の設定を続行します。

> **Note**
>
> 予算を作成したら、それを製品またはポートフォリオに関連付ける必要があります。

予算の関連付け

各ポートフォリオまたは製品には 1 つの予算を関連付けることができます。各予算は複数のポートフォリオや製品に関連付けることができます。

予算をポートフォリオまたは製品に関連付けると、そのポートフォリオまたは製品の詳細ページから予算に関する情報を表示できます。ポートフォリオまたは製品で発生した支出を予算に反映するには、予算とポートフォリオまたは製品に同じタグを関連付ける必要があります。

> **Note**
>
> AWS Budgets から予算を削除しても、AWS Service Catalog 製品およびポートフォリオとの既存の関連付けは引き続き存在します。AWS Service Catalog は、削除された予算に関する情報を表示できなくなります。

予算を関連付けるには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [製品リスト] または [ポートフォリオ] を選択します。
3. 予算を関連付ける製品またはポートフォリオを選択します。
4. [アクション] メニューを開き、[予算を関連付ける] を選択します。
5. [予算の関連付け] ページで、既存の予算を選択し、[続行] を選択します。
6. 製品またはポートフォリオのテーブルに、追加したばかりの予算のデータが含まれるようになりました。

予算の表示

予算が製品に関連付けられている場合は、[製品の詳細] または [製品のリスト] ページで予算に関する情報を表示できます。予算がポートフォリオに関連付けられている場合は、[ポートフォリオ] および [ポートフォリオの詳細] ページで予算に関する情報を表示できます。

[ポートフォリオ] と [製品のリスト] のページに、既存のリソースの予算情報が表示されます。[現状対予算] および [予測対予算] を表示する列を表示できます。

製品またはポートフォリオを選択すると、詳細ページに移動します。[ポートフォリオの詳細] と [製品の詳細] ページには、関連付けられた予算に関する詳細情報を含むセクションがあります。予算額、現在の使用額、および予測使用額を確認できます。また、予算の詳細を表示し、予算を編集するオプションもあります。

予算の関連付けの解除

ポートフォリオまたは製品から予算の関連付けを解除できます。

> **Note**
>
> AWS Budgets から予算を削除しても、AWS Service Catalog 製品およびポートフォリオとの既存の関連付けは引き続き存在します。 AWS Service Catalog は、削除された予算に関する情報を表示できなくなります。

予算の関連付けを解除するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. [製品リスト] または [ポートフォリオ] を選択します。
3. 予算の関連付けを解除する製品またはポートフォリオを選択します。
4. [アクション] を選択します。ドロップダウンから [予算の関連付けを解除] を選択します。確認のアラートが表示されます。
5. 製品またはポートフォリオから予算の関連付けを解除することを確認したら、[確認] を選択します。

プロビジョニング済み製品の管理

AWS Service Catalog は、プロビジョニング済み製品を管理するためのインターフェイスを提供します。アクセスレベルに基づいてカタログのすべてのプロビジョニング済み製品を表示、更新、終了できます。手順の例については、以下のセクションを参照してください。

トピック

プロビジョニングされた製品を管理者として管理する

アカウントのプロビジョニングされた製品をすべて管理するには、プロビジョニングされた製品の書き込み操作にアクセスするための `AWSServiceCatalogAdminFullAccess`、または同等の IAM 権限が必要です。詳細については、「AWS Service Catalogにおけるアイデンティティとアクセスの管理」を参照してください。

> **Tip**
>
> 静的なプロビジョニング済み製品のチェーニングでは、プロビジョニング済み製品をプロビジョニングする前に、製品アーティファクトテンプレート内のプロビジョニング済み製品の出力を参照する必要があります。例を含む詳細については、次を参照してください。
>
> - 「AWS::ServiceCatalog::CloudFormationProvisionedProduct」(AWS CloudFormation ユーザーガイド) を参照してください。
> - 「 AWS Service Catalogデベロッパーガイド」のDescribeProvisioningParameters「 (ProvisioningArtifactOutputKeys) 」。

すべてのプロビジョニング済み製品を表示および管理するには

1. AWS Service Catalog コンソールを開きます (https://console.aws.amazon.com/servicecatalog/)。

   AWS Service Catalog コンソールにすでにログインしている場合、[Service Catalog]、[エンドユーザー] の順に選択します。
2. 必要に応じて、[プロビジョニング済み商品] セクションまで下方へスクロールします。
3. [プロビジョニング済み製品] セクションで、[表示: リスト] を選択して、表示するアクセスのレベルを [ユーザー]、[ロール] または [アカウント] から選択します。このアクションにより、カタログ内のすべてのプロビジョニング済み製品が表示されます。
4. 表示、更新、または終了するプロビジョニング済み製品を選択します。このビューに表示される情報の詳細については、「プロビジョニング済み製品に関する情報を表示する」を参照してください。

プロビジョニング済み製品所有者の変更

プロビジョニング済み製品の所有者はいつでも変更できます。新しい所有者として設定するユーザーまたはロールの ARN を知っている必要があります。

デフォルトでは、この機能は、`AWSServiceCatalogAdminFullAccess` 管理ポリシーを使用する管理者が使用できます。AWS Identity and Access Management (IAM) の `servicecatalog:UpdateProvisionedProductProperties` のアクセス許可をエンドユーザーに付与することで、エンドユーザーに対して有効にできます。

プロビジョニング済み製品の所有者を変更するには

1. AWS Service Catalog コンソールで、[プロビジョニング済み製品リスト] を選択します。
2. 更新するプロビジョニング済み製品を見つけ、その横にある 3 つのドットを選択して、[Change provisioned product owner (プロビジョニング済み製品所有者の変更)] を選択します。また、[アクション] メニューのプロビジョニング済み商品の詳細ページに [所有者の変更] オプションもあります。
3. ダイアログボックスで、新しい所有者として設定するユーザーまたはロールの ARN を入力します。ARN は `arn:` で始まり、コロンやスラッシュで区切られたその他の情報 (`arn:aws:iam::123456789012:user/NewOwner` など) を含みます。
4. [送信] を選択します。所有者が更新されると、成功メッセージが表示されます。

以下の資料も参照してください。

- UpdateProvisionedProductProperties

プロビジョニング済み製品のテンプレートの更新

プロビジョニング済み製品の現在のテンプレートを別のテンプレートに変更できます。例えば、Service Catalog に EC2 製品がある場合、その EC2 製品を更新して、同じプロビジョニング済み製品 ID を保持し、テンプレートを S3 バケットに変更できます。

> **Note**
>
> プロビジョニング済みの Terraform Open Source 製品または Terraform Cloud 製品では、テンプレートの更新はサポートされていません。既存の Terraform 製品に別のテンプレートを使用する場合は、製品を削除し、目的のテンプレートを使用して新しい製品を作成する必要があります。

プロビジョニング済み製品のテンプレートを更新するには

1. 左側のナビゲーションメニューで、[プロビジョニング済み製品] を選択します。
2. [プロビジョニング済み製品] で、プロビジョニング済み製品を選択し、[アクション]、[更新] を選択します。

   また、プロビジョニング済み製品の詳細ページで [アクション] 、[更新] を選択することもできます。
3. (オプション) [製品の詳細] で、[製品の変更] を選択します。

   [製品の変更] では、次のことに注意してください。

   製品を変更すると、このプロビジョニング済み製品が別の製品テンプレートに更新されます。これにより、リソースが終了し、新しいリソースが作成される可能性があります。

   プロビジョニング済み製品を同じ製品内の別のバージョンに更新できます。
4. (オプション) [製品] で、別のテンプレートで更新する製品を選択します。次に [変更] を選択します。

   [商品の詳細] では、次のことに注意してください。

[製品名] が [現在のテンプレート名] から [新しいテンプレート名] に更新されます。ただし、プロビジョニングされた製品の名前 [プロビジョニングされた製品名] は変更されません。

プロビジョニング済み製品を同じ製品内の別のバージョンに更新できます。

5. [製品バージョン] で、使用する製品のバージョンを選択します。
6. [パラメータ] で、該当するパラメータを選択します。
7. [更新] を選択します。

[プロビジョニング済み製品の詳細] では、更新の詳細が表示されます。プロビジョニングされた製品名は変更されませんが、プロビジョニングされた製品には別のテンプレートが含まれるようになりました。

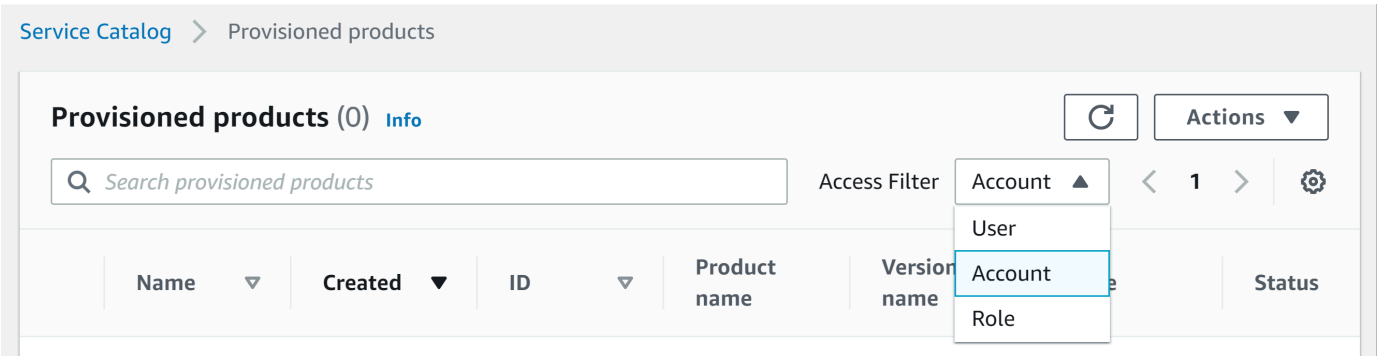
チュートリアル：ユーザーのリソース割り当ての確認

AWS Service Catalog コンソールを使用して、製品をプロビジョニングしたユーザーとその製品に関連付けられているリソースを確認できます。このチュートリアルでは、次の例をユーザー独自の特定のプロビジョニング済み製品に応用できるようにします。

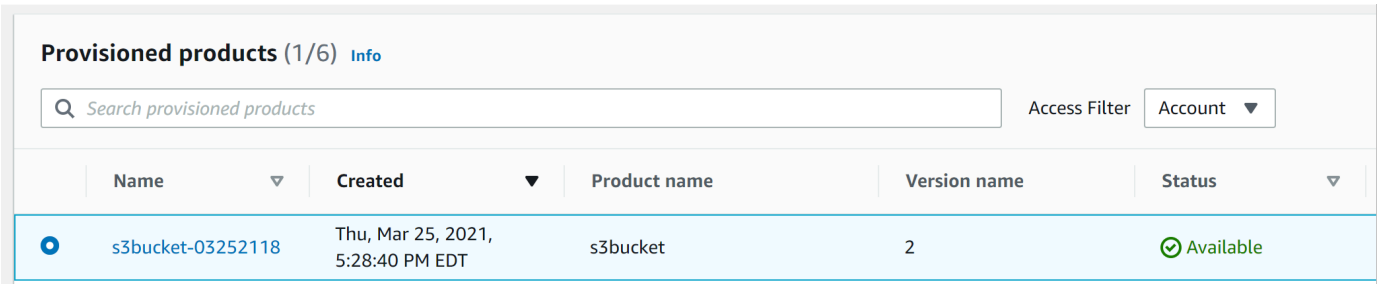
アカウントのすべてのプロビジョニング済み製品を管理するには、プロビジョニング済み製品に対する書き込みオペレーションへの `AWSServiceCatalogAdminFullAccess` または同等のアクセスが必要です。詳細については、AWS Service Catalog 管理者ガイドの「Identity and Access Management」を参照してください。

製品をプロビジョニングしたユーザーおよび関連するリソースを確認するには

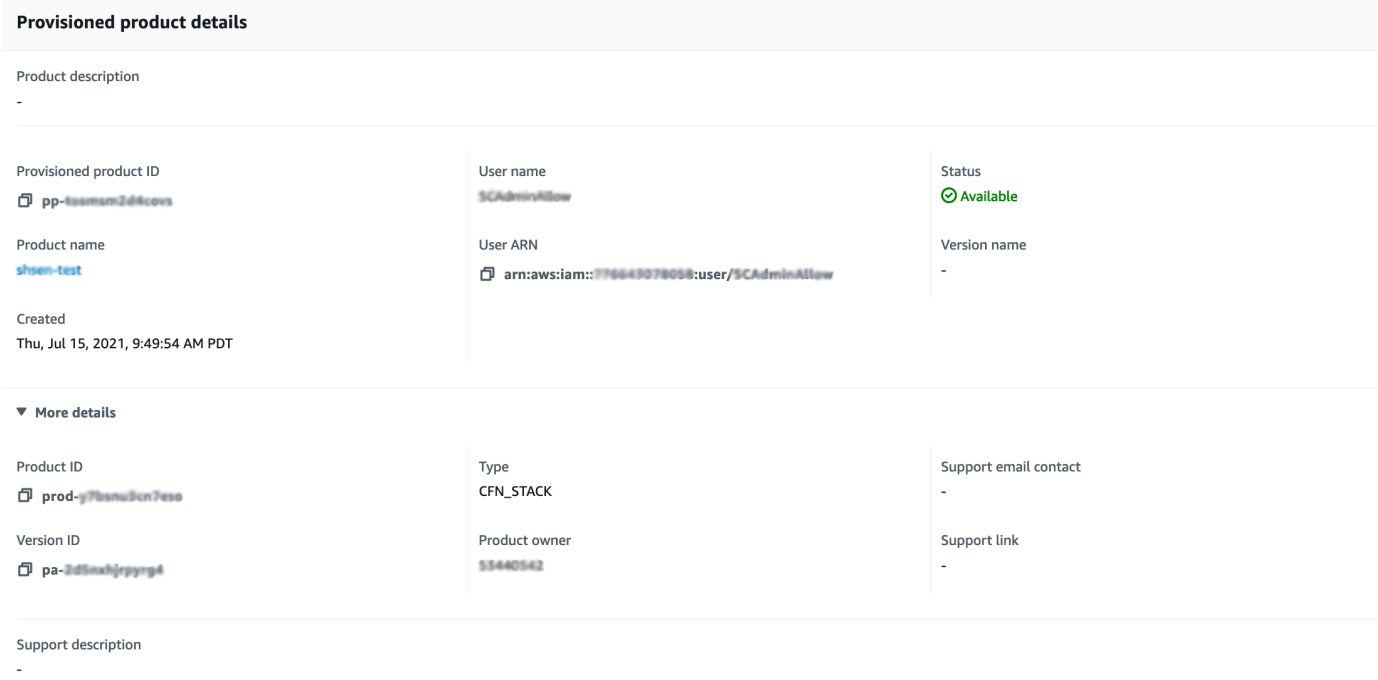
1. https://console.aws.amazon.com/servicecatalog を開きます。
2. 左側のナビゲーションメニューで、[プロビジョニング済み製品] を選択します。
3. [アクセスフィルター] ドロップダウンメニューで、[アカウント] を選択します。



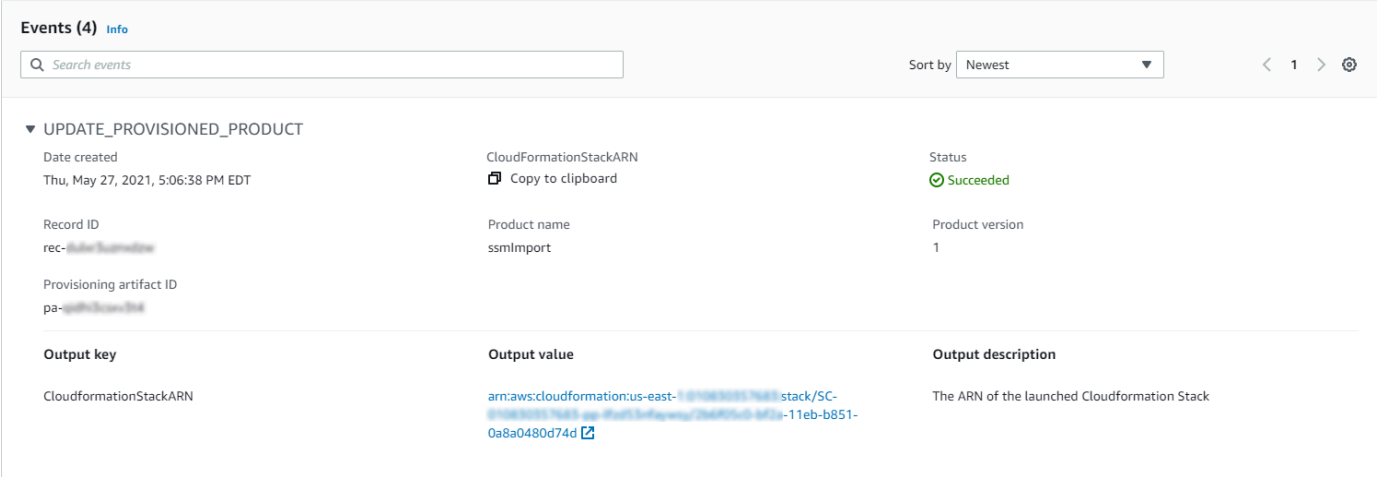
4. [アカウント] ビューで、プロビジョニング済み製品を表示、選択、開き、詳細を表示します。



プロビジョニング済み製品の詳細が表示されます。



5. 下にスクロールして [イベント] セクションを展開します。`Provisioned product ID` と `CloudformationStackARN` の値に注意してください。



6. プロビジョニング済み製品 ID を使用して、この起動に対応する AWS CloudTrail レコードを確認し、リクエストしているユーザー (通常は、フェデレーション時に入力した E メールアドレス) を確認します。この例では「steve」です。

```
{
  "eventVersion":"1.03","userIdentity":
  {
    "type":"AssumedRole",
    "principalId":"[id]:steve",
    "arn":"arn:aws:sts::[account number]:assumed-role/SC-usertest/steve",
    "accountId":[account number],
    "accessKeyId":[access key],
    "sessionContext":
    {
      "attributes":
      {
        "mfaAuthenticated":[boolean],
        "creationDate":[timestamp]
      },
      "sessionIssuer":
      {
        "type":"Role",
        "principalId":"AROAJEXAMPLELH3QXY",
        "arn":"arn:aws:iam::[account number]:role/[name]",
        "accountId":[account number],
        "userName":[username]
      }
    }
  },
  "eventTime":"2016-08-17T19:20:58Z","eventSource":"servicecatalog.amazonaws.com",
```

```
  "eventName":"ProvisionProduct",
  "awsRegion":"us-west-2",
  "sourceIPAddress":[ip address],
  "userAgent":"Coral/Netty",
  "requestParameters":
  {
    "provisioningArtifactId":[id],
    "productId":[id],
    "provisioningParameters":[Shows all the parameters that the end user entered],
    "provisionToken":[token],
    "pathId":[id],
    "provisionedProductName":[name],
    "tags":[],
    "notificationArns":[]
  },
  "responseElements":
  {
    "recordDetail":
    {
      "provisioningArtifactId":[id],
      "status":"IN_PROGRESS",
      "recordId":[id],
      "createdTime":"Aug 17, 2016 7:20:58 PM",
      "recordTags":[],
      "recordType":"PROVISION_PRODUCT",
      "provisionedProductType":"CFN_STACK",
      "pathId":[id],
      "productId":[id],
      "provisionedProductName":"testSCproduct",
      "recordErrors":[],
      "provisionedProductId":[id]
    }
  },
  "requestID":[id],
  "eventID":[id],
  "eventType":"AwsApiCall",
  "recipientAccountId":[account number]
}
```

7. CloudformationStackARN の値を使用して AWS CloudFormation イベントを識別し、作成されたリソースに関する情報を見つけます。AWS CloudFormation API を使用して、この情報を取得することもできます。詳細については、「AWS CloudFormation API リファレンス」を参照してください。

ステップ 1〜4 は、AWS Service Catalog API または AWS CLI を使って行うことができます。詳細については、「AWS Service Catalogデベロッパーガイド」および「AWS Service Catalogコマンドラインリファレンス」を参照してください。

Terraform Open Source 製品のステータスエラーの管理

Terraform Open Source ProvisionProduct の障害は TAINTED 状態にルーティングされ、プロビジョニングされた各製品が UpdateProvisionedProduct に進むことができます。この問題が発生した場合:

- UpdateProvisionedProduct は、タグの更新や修正、またはリソースグループの作成や変更を試みません。
- UpdateProvisionedProduct は、プロビジョニングされた製品を AVAILABLE または TAINTED に設定する必要があるかどうかを決定するときに、以前のプロビジョニング操作による失敗を考慮しません。

AWS Service Catalog は ProvisionProduct 中にのみタグを適用します。ProvisionProduct 操作の失敗によるタグ付けの失敗は、自動的には解決されません。

ステータスエラーの例

例 1: AWS Service Catalog は **ProvisionProduct** 中にリソースグループを作成しない

以下のシナリオでは、サポートするリソースグループがなく、リソースにタグが適用されていなくても、プロビジョニングされた製品がその AVAILABLE 状態になっています。

1. ProvisionProduct でアクションが開始されます。
2. Terraform プロビジョニングエンジンはワークフロー障害で ProvisionProduct に応答しますが、ResourceIdentifier を提供しません。
3. ProvisionProduct ワークフローはリソースグループを作成せず、プロビジョニングされた製品の状態を ERROR に設定します。
4. その後、Updateprovisionedproduct 操作を開始します。
5. Terraform プロビジョニングエンジンが「成功」と応答します。
6. その結果、UpdateprovisionedProduct ワークフローはプロビジョニングされた製品の状態を AVAILABLE に設定しますが、リソースグループを作成したり、タグの適用を試みたりすることはありません。

例 2: AWS Service Catalog は **UpdateProvisionedProduct** 中に新しいリソースを作成します

以下のシナリオでは、新しいリソースにタグが適用されていなくても、プロビジョニングされた製品がその AVAILABLE 状態になっています。

1. ProvisionProduct でアクションが開始されます。
2. Terraform プロビジョニングエンジンが「成功」と応答し、「ResourceIdentifier」を提供します。
3. ProvisionProduct ワークフローはリソースグループを作成し、特定されたすべてのリソースにタグを適用します。
4. 新しいリソースを作成する新しいアーティファクトで UpdateProvisionedProduct を開始します。
5. Terraform プロビジョニングエンジンが「成功」と応答します。
6. UpdateProvisionedProduct ワークフローはプロビジョニングされた製品の状態を AVAILABLE に設定しますが、新しいリソースに追加のタグを適用しようとはしません。

ステータスエラーの解決策

AWS Service Catalog は、ProvisionProduct から TAINTED に設定された、プロビジョニングされたすべての製品に対してリソースグループが作成されていることを確認します。Terraform プロビジョニングエンジンが ResourceIdentifier を返さない場合、AWS Service Catalog がリソースグループの作成に失敗した場合、プロビジョニング済み製品はその ERROR 状態に設定され、強制終了されます。

Terraform Open Source 製品ステートファイルの管理

Terraform Open Source のプロビジョニング済み製品にはすべて、単一状態のファイルがあります。プロビジョニングされた製品とその状態ファイルには 1:1 の関係があります。このファイルは、sc-terraform-engine-state-${AWS::AccountId}-${AWS::Region} という名前の Amazon S3 バケットに保存されています。状態ファイルは AccountID または ProvisionedProductID オブジェクトキーで保存されます。

状態ファイルへのアクセスは、GetStateFileAWS Lambda と Amazon EC2 起動テンプレートに限定されます。 AWS Service Catalog 管理者は Amazon S3 の状態ファイルに直接アクセスすることはできません。管理者は Amazon EC2 を使用してファイルにアクセスする必要があります。デフォルトでは、AWS Service Catalog 管理者はステートファイルのリストを表示できますが、ファイルの

内容を読み書きすることはできません。Terraform プロビジョニングエンジンのみがファイルの内容を読み書きできます。

AWS Service Catalog でのタグの管理

AWS Service Catalog には、リソースを分類できるように、タグが用意されています。タグには、AutoTags と の 2 つのタイプがあります TagOptions。

AutoTags は、 のプロビジョニングされたリソースのオリジンに関する情報を識別するタグAWS Service Catalogで、 によってAWS Service Catalogプロビジョニングされたリソースに自動的に適用されます。

TagOptions は、 で管理AWS Service Catalogされるキーと値のペアで、AWSタグを作成するためのテンプレートとして機能します。

トピック

AWS Service Catalog AutoTags

> **Note**
>
> AWS Service Catalog は Terraform Open Source 製品の AutoTags をサポートしていません。

AutoTags は、 のプロビジョニングされたリソースのオリジンに関する情報を識別するタグAWS Service Catalogで、 によってAWS Service Catalogプロビジョニングされたリソースに自動的に適用されます。

AutoTags には、ポートフォリオ、製品、ユーザー、製品バージョン、およびプロビジョニング済み製品の一意の識別子のタグが含まれます。これにより、ユーザーがカタログで設定した AWS Service Catalog 構造を反映する一連のタグが提供されます。 AutoTags は、お客様の 50 タグの制限にはカウントされません。

> **Note**
>
> AWS Service Catalog は Terraform Open Source 製品の AutoTags をサポートしていません。

AWS Service Catalog AutoTags は、 リソースに一貫したタグ付けを提供するのに役立ちます。これは、ポートフォリオ、製品、またはユーザーの予算を設定するときに便利です。を使用して AutoTags 、AWS Configルールの設定など、起動後のオペレーションのリソースを識別することもできます。プロビジョニングされたリソース AutoTags の はAWS CloudFormation、、Amazon EC2、Amazon S3 など、プロビジョニングに使用されるダウンストリームサービスのタグセクションで表示できます。

> **Note**
>
> AWS Service Catalog は、プロビジョニングされたリソース AutoTags に適用 AutoTags した後は更新されません。プロビジョニングされた製品を別の製品、プロビジョニングされたアーティファクト、または新しい起動パスに更新しても、既存の には元の値 AutoTags が表示されます。

AutoTag の詳細

- aws:servicecatalog:portfolioArn - プロビジョニング済み製品の起動元のポートフォリオの ARN。
- aws:servicecatalog:productArn - プロビジョニング済み製品の起動元の製品の ARN。
- aws:servicecatalog:provisioningPrincipalArn - プロビジョニング済み製品を作成したプロビジョニングプリンシパル (ユーザー) の ARN。
- aws:servicecatalog:provisionedProductArn - プロビジョニング済み製品 ARN。
- aws:servicecatalog:provisioningArtifactIdentifier - 元のプロビジョニングアーティファクト (製品バージョン) の ID。

AWS Service Catalog TagOption ライブラリ

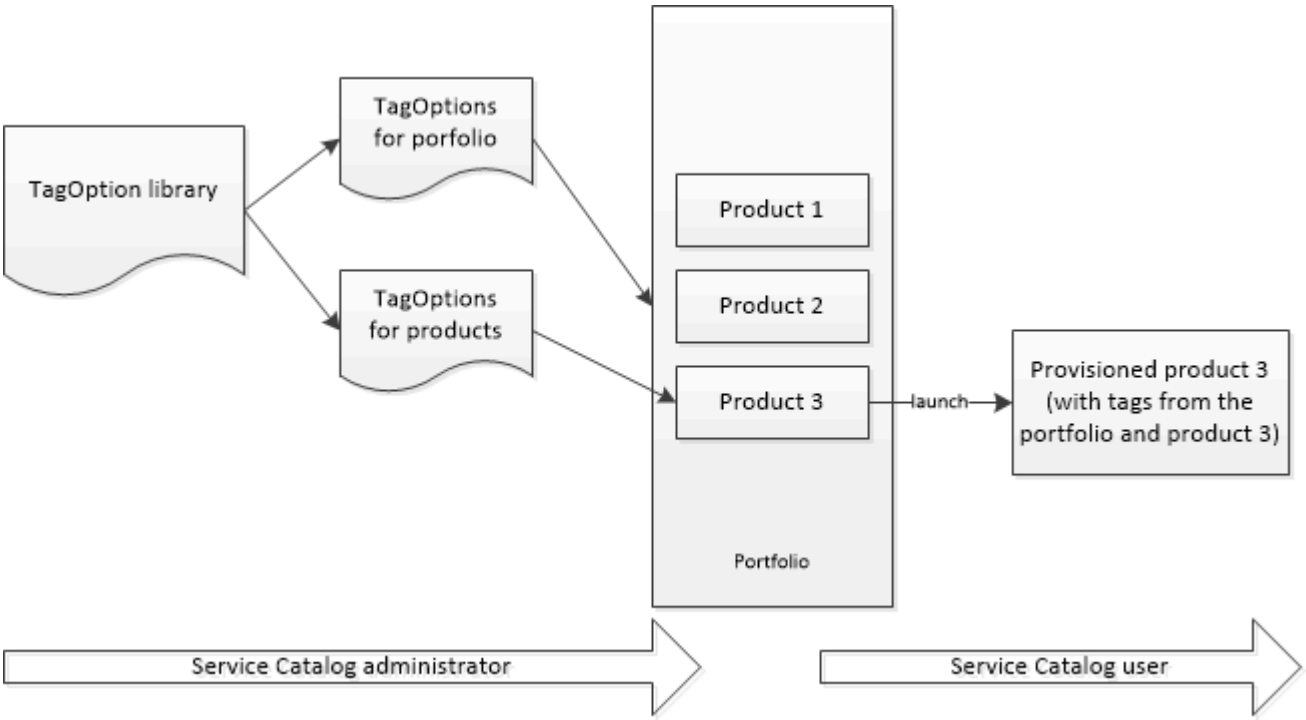
管理者がプロビジョニング済み製品のタグを簡単に管理できるように、 は TagOption ライブラリ AWS Service Catalogを提供します。 TagOption は、 で管理されるキーと値のペアですAWS Service Catalog。これは AWS タグではありませんが、 に基づいて AWS タグを作成するためのテンプレートとして機能します TagOption。

AWS Service Catalog は TagOptions 、Terraform Open Source または Terraform Cloud 製品のサポートしていません。

TagOption ライブラリを使用すると、以下を簡単に適用できます。

- 整合性のある分類
- AWS Service Catalog リソースの適切なタグ付け
- 許可されたタグのためのユーザーが選択可能な定義済みのオプション

管理者はポートフォリオや製品 TagOptions に関連付けることができます。製品の起動 (プロビジョニング) 中、 は関連するポートフォリオと製品 をAWS Service Catalog集約し TagOptions、次の図に示すように、プロビジョニングされた製品に適用します。



TagOption ライブラリを使用すると、ポートフォリオまたは製品との関連付けを非アクティブ化 TagOptions して保持し、必要に応じて再アクティブ化できます。このアプローチは、ライブラリの整合性を維持するだけでなく、断続的に使用されるか、特別な状況でのみ使用される可能性のある TagOptions を管理することもできます。

は、 AWS Service Catalogコンソールまたは TagOption ライブラリ API TagOptions を使用して管理します。詳細については、「Service Catalog API リファレンス」を参照してください。

コンテンツ

での製品の起動 TagOptions

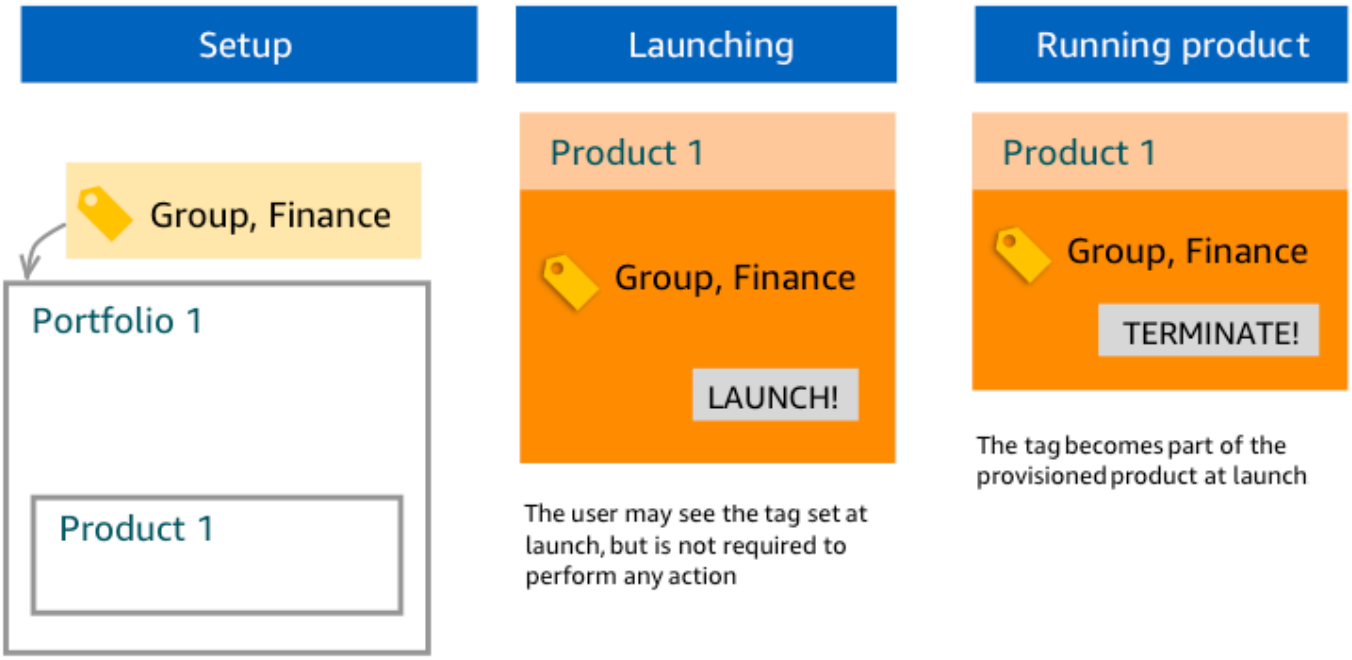
ユーザーが を持つ製品を起動すると TagOptions、 AWS Service Catalogはユーザーに代わって次のアクションを実行します。

- 製品と起動ポートフォリオ TagOptions のすべての を収集します。
- プロビジョニング済み製品のタグでは、一意のキー TagOptions のみが使用されます。ユーザーは、キーの複数選択値リストを取得します。ユーザーが値を選択すると、プロビジョニング済み製品のタグになります。
- ユーザーは、プロビジョニング中に競合しないタグを製品に追加することができます。

以下のユースケースは、起動時の TagOptions 動作を示しています。

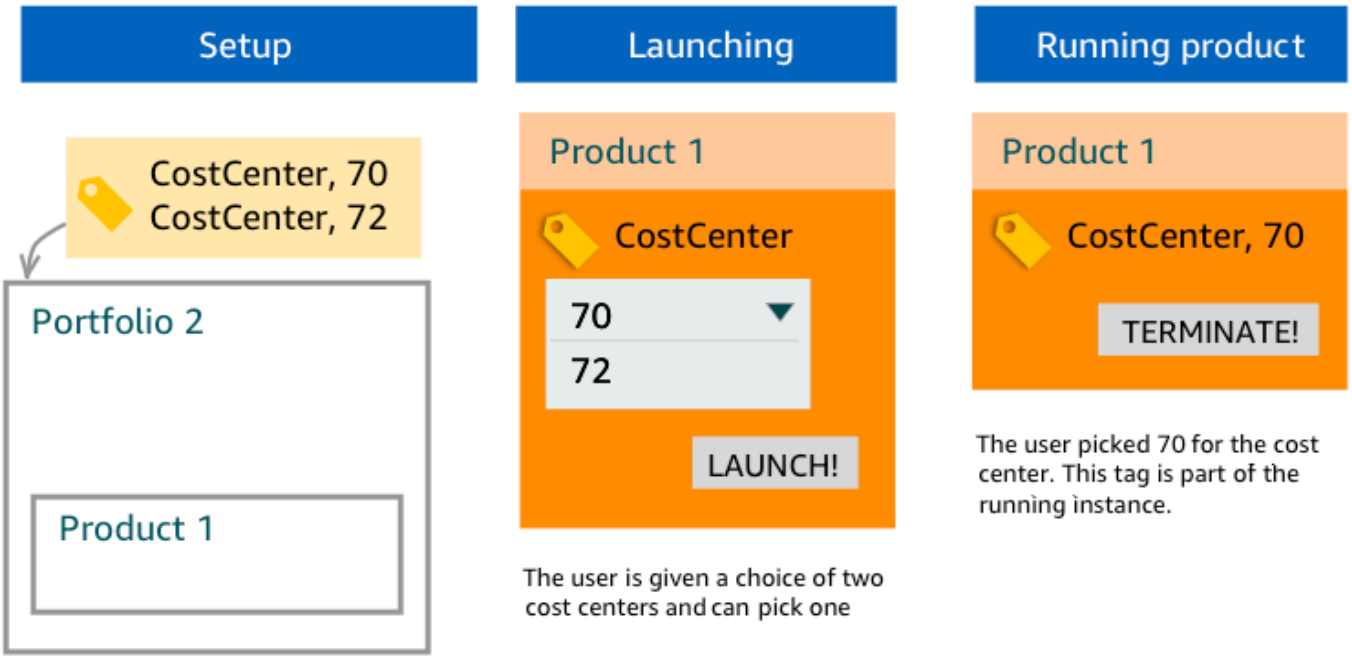
例 1: 一意の TagOption キー

管理者は、TagOption[Group=Finance] を作成し、それをPortfolio1 に関連付けます。ポートフォリオ 1 には、Product1 があり、 はありません TagOptions。ユーザーがプロビジョニング済み製品を起動すると、次のように、単一の が Tag[Group=Finance] TagOption になります。



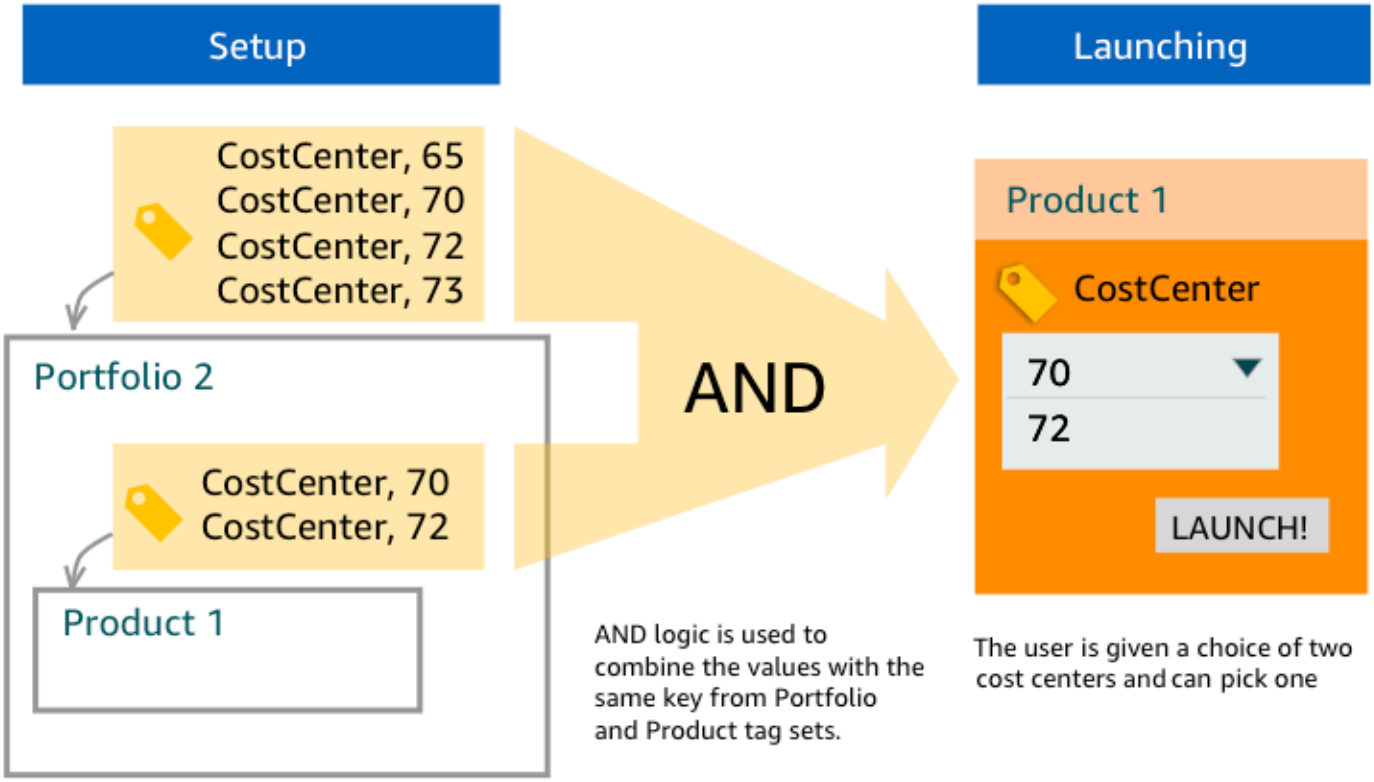
例 2: ポートフォリオで同じキー TagOptions を持つ のセット

管理者は、同じキー TagOptions を持つ 2 つの をポートフォリオに配置し、そのポートフォリオ内の製品に同じキー TagOptions を持つ はありません。起動時に、ユーザーはキーに関連付けられた 2 つの値のいずれかを選択する必要があります。プロビジョニング済みの製品には、キーとユーザーが選択した値でタグが付けられます。



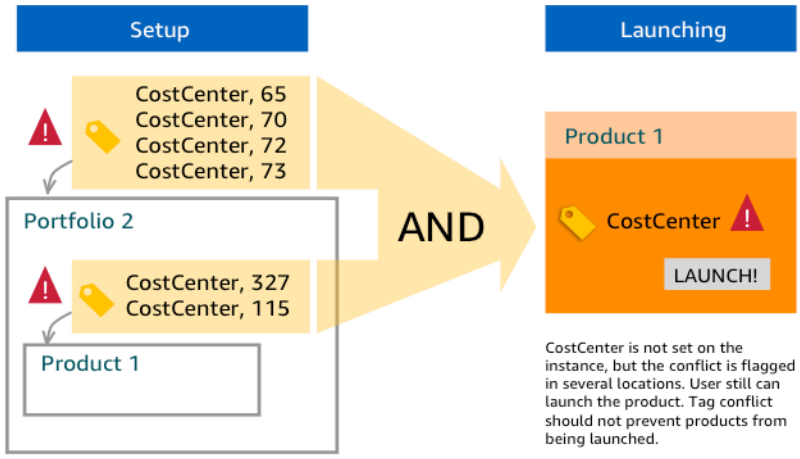
例 3: ポートフォリオとそのポートフォリオ内の製品の両方で同じキー TagOptions を持つ のセット

管理者は、ポートフォリオに同じキー TagOptions を持つ複数の を配置し、そのポートフォリオ内の製品にも同じキー TagOptions を持つ複数の を配置しています。 は、 の集約 (論理 AND オペレーション) から一連の値AWS Service Catalogを作成します TagOptions。ユーザーが製品を起動すると、ユーザーはこの値のセットを見て選択します。プロビジョニング済みの製品に、キーとユーザーが選択した値でタグが付けられます。



例 4: 同じキーと競合する値 TagOptions を持つ複数の

管理者は、ポートフォリオに同じキー TagOptions を持つ複数の を配置し、そのポートフォリオ内の製品にも同じキー TagOptions を持つ複数の を配置しています。 は、 の集約 (論理 AND オペレーション) から一連の値AWS Service Catalogを作成します TagOptions。集約でキーの値が見つからない場合、AWS Service Catalog は同じキーと `sc-tagconflict-`*`portfolioid`*`-`*`productid`* という値を持つタグを作成します。ここで、*`portfolioid`* と *`productid`* は、ポートフォリオと製品の ARN です。これにより、プロビジョニング済みの製品に、正しいキーと、管理者が検索して修正できる値のタグが付けられます。



の管理 TagOptions

管理者は、次のアクションを実行して TagOptions 、ライブラリ TagOptions で を管理できます。

- 作成と削除
- アクティブ化または非アクティブ化
- 関連付け/関連付け解除
- [Edit (編集)]

コンソール TagOptions で を作成するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションメニューで、TagOptionsライブラリ を選択します。
3. 「新しい を作成する TagOption」で、キーと値を入力し、「 を追加」を選択します。

   新しい TagOption が作成されると、キーと値のペアでグループ化され、TagOptionsリスト内でアルファベット順にソートされます。

   AWS Service Catalog API TagOption を使用して を作成するには、「」を参照してくださいCreateTagOption。

コンソール TagOptions で を削除するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。

2. 左側のナビゲーションメニューで、TagOptions ライブラリ を選択し、アクション を選択します。
3. [削除] を選択して、削除を確定します。

コンソール TagOptions で 1 つ以上の を有効または無効にするには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションメニューで、TagOptions ライブラリ を選択し、アクション を選択します。
3. アクティブ化するには、 TagOption 使用する非アクティブ を選択します。次に、[アクション] を選択し、ドロップダウンメニューから [有効化] を選択し、選択を確定します。

   非アクティブ化するには、 TagOption 目的のアクティブな を選択します。次に、[アクション] を選択し、ドロップダウンメニューから [非アクティブ化] を選択し、選択を確定します。

コンソールで 1 つ以上の をポートフォリオ TagOptions に関連付けたり、関連付けを解除したりするには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションメニューで、[ポートフォリオ] から、関連付けするポートフォリオまたは関連付けを解除するポートフォリオを開きます。
3. TagOptions タブを選択し、ポートフォリオとの関連付けまたは関連付け解除 TagOptions を行う 1 つ以上の を選択します。
4. [アクション] を選択します。次に、[関連付け] または [関連付け解除] を選択し、選択を確認します。

コンソールで 1 つ以上の を製品 TagOptions に関連付けたり、関連付けを解除したりするには

1. AWS Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションメニューの [管理] で、[製品] を選択します。次に、関連付けを行う、または関連付けを解除する製品を開きます。
3. TagOptions タブを選択し、ポートフォリオとの関連付けまたは関連付け解除 TagOptions を行う 1 つ以上の を選択します。

4. [アクション] を選択します。次に、[関連付け] または [関連付け解除] を選択し、選択を確認します。

> **Note**
>
> AWS Service Catalog API を使用してポートフォリオまたは製品 TagOptions に関連付けるには、「」を参照してくださいAssociateTagOptionWithResource。
> AWS Service Catalog API TagOptions を使用して削除 (関連付け解除) するには、「」を参照してくださいDisassociateTagOptionFromResource。

コンソール TagOptions で の値を編集するには

1. Service Catalog コンソール (https://console.aws.amazon.com/servicecatalog/) を開きます。
2. 左側のナビゲーションメニューで、TagOptionsライブラリ を選択します。
3. TagOption を選択し、値を開きます。(値はハイパーリンクされています)。次に、[編集] を選択します。
4. [値] フィールドで、値を編集し、[変更の保存] を選択します。

AWS Organizations タグポリシー TagOptions での の使用

このトピックでは、 AWS Organizationsおよび のタグポリシーの概要 TagOptions を説明しますAWS Service Catalog。また、両方の機能を同時に使用する場合に、タグ付け競合を防ぐ方法についても説明します。

TagOptions のタグポリシーAWS Service CatalogはAWSアカウントと組織単位 (CloudFormationOU) または組織ルートに適用されますが、 の はプロビジョニング済み製品 (スタック) AWS Organizationsに適用されます。例えば、OU にタグポリシーをアタッチすると、同じタグポリシーはその OU 内のすべてのアカウントに適用されます。両方のタグ付け機能を同時に使用する場合は、競合しないように構成する必要があります。

タグポリシー

タグポリシーを使用すると、AWS Organizations のアカウントの AWS リソースでのタグの使用方法に関するルールを定義できます。タグポリシーを使用して、アカウントレベルで AWS リソースの一貫性のあるタグ付けアプローチを作成し、維持できます。

タグポリシーは、ユーザーが一貫したタグを適用し、タグ付きリソースを監査し、適切なリソース分類を維持するための簡単な方法を提供します。また、タグキーを大文字にする方法、および許可する値を定義することもできます。例えば、アカウント内のすべての EC2 インスタンスに、タグキーを **CostCenter** に、タグの値を **Data Insights** または **Marketing** に設定することを要求できます。

タグポリシーを使用すると、タグ付けルールを適用するオプション、タグの非準拠操作を防ぎ、適用するリソースタイプを指定するためのオプションを選択できます。強制オプションを選択しない場合、タグポリシーでは、非準拠タグを作成または変更できますが、それらのタグは AWS Organizations コンソールで非準拠としてレポートされます。

アカウントレベルのタグ適用をセットアップする方法の詳細については、「AWS Organizations でのタグポリシー」を参照してください。

TagOptions

TagOptions は、関連付けられた製品AWS Service Catalogに適用されている場合、 CloudFormation スタックレベルでプロビジョニング済み製品に適用されるタグ付け機能です。 AWS Service Catalog には、AWS Service Catalog製品に関連付けるキーと値のペアを定義できる TagOptions ライブラリが用意されています。AWS Service Catalog 製品を起動するときは、そのポートフォリオまたは製品に関連付けられた既存の TagOption キー TagOption の値を選択して、その製品を起動する必要があります。 TagOptions はポートフォリオレベルまたは製品レベルで設定するため、アカウントとリージョン間で共有されるポートフォリオに一貫した分類を適用できます。

TagOptions で を設定する方法の詳細についてはAWS Service Catalog、AWS Service Catalog TagOption 「ライブラリ」を参照してください。

AWS Organizations タグポリシーと 間の競合を回避する AWS Service Catalog TagOptions

組織内のアカウントの AWS Organizations タグポリシーを設定する場合は、以下を推奨します。

- 適合タグの要件を、AWS Service Catalogポートフォリオや製品に関して も管理する TagOptions 管理者と共有します。
- 適合タグの要件を AWS Service Catalog で製品を起動する可能性のあるエンドユーザーと共有し、オプションのエンドユーザータグを製品の起動に追加してください。

TagOption キー AWS Service Catalogを使用する で製品を起動し`city`、、**`AtlantaSan Francisco`**、または などの米国の都市 のタグ値を持つ`city`タグキーを要求するタグポリシーがあ

るとします**Austin**。 AWS Service Catalog では、製品に必要な TagOption キー TagOption の値を選択せずに製品を起動することはできません。

この場合、 **Rio de Janeiro**や など、南米の都市 cityを含む TagOption キー TagOption の値があると、 **Buenos Aires**AWS Service Catalogは製品を起動しません。代わりに、タグポリシーに準拠するには、起動時に米国の都市を含む TagOption 値を選択する必要があります。

次の表は、タグポリシーの使用時および TagOptions 同時に発生する可能性のあるタグ付けの競合の問題を解決する方法を説明するシナリオを示しています。

シナリオ	理由	ソリューション
タグポリシーでタグの適用のチェックがオンになっている場合、非準拠のタグが原因で製品の起動に失敗します。	タグポリシーの準拠タグの許可リストに追加していないキーと値 TagOptions で を指定します。 タグポリシーに準拠していないオプションのカスタムタグを追加する。	タグポリシータグキーの大文字と小文字の区別の適用で特定の大文字と小文字スキーマを設定する場合は、TagOptions タグキーとオプションのカスタムタグキーが、タグポリシーで指定したものと一致していることを確認してください。 タグポリシーでタグキーの大文字と小文字の区別の適用ボックスがオフになっていると、すべての小文字のタグキーが準拠し、 TagOptions タグキーとオプションのカスタムタグキーがタグポリシーで要求したものと一致する (すべて小文字など) ことに注意してください。
タグキーの大文字と小文字の区別により、製品の起動に失敗しました。	タグポリシーの大文字と小文字の区別の適用ルールと一致しない TagOptions キーに大文字と小文字を指定します。	タグポリシーを正しく構成してください。タグキーの大文字と小文字の区別のコンプライアンスを規定しない場合、

シナリオ	理由	ソリューション
		デフォルトでタグキーはすべて小文字になります。 さらに、タグポリシーでタグキーの大文字と小文字の区別コンプライアンスを指定しない場合は、強制ルールに準拠するために、 の TagOptions タグキーAWS Service Catalog がすべて小文字であることを確認してください。 大文字と小文字の区別のコンプライアンスが有効になっていないタグポリシーを使用する場合、そのタグポリシーでは、すべての小文字のタグキーのみが準拠していると見なされます。
互換性のないタグ値のため、製品の起動に失敗しました。	タグポリシーの TagOptions タグ値コンプライアンスの許可リストに含まれていない製品起動のタグ値を選択します。	リストタグポリシータグ値コンプライアンスで許可されているタグ値で要求されているものと一致する製品とポートフォリオ TagOptions に関連付けます。

の外部エンジン AWS Service Catalog

では AWS Service Catalog、外部エンジンはEXTERNAL製品タイプを通じて表されます。 EXTERNAL 製品タイプを使用すると、Terraform などのサードパーティーのプロビジョニングエンジンを統合できます。外部エンジンを使用すると、Service Catalog の機能をネイティブ AWS CloudFormation テンプレート以外にも拡張できるため、他の Instructure as Code (IaC) ツールを使用できます。

EXTERNAL 製品タイプを使用すると、選択した IaC ツールの特定の機能と構文を活用しながら、Service Catalog の使い慣れたインターフェイスを使用してリソースを管理およびデプロイできます。

Service Catalog でEXTERNAL製品タイプを有効にするには、アカウントで一連の標準リソースを定義する必要があります。これらのリソースはエンジン と呼ばれます。Service Catalog は、アーティファクト解析およびプロビジョニングオペレーションの特定の時点で、タスクをエンジンに委任します。

プロビジョニングアーティファクトは、Service Catalog 内の製品の特定のバージョンを表し、一貫したリソースを管理およびデプロイできるようにします。

EXTERNAL 製品タイプのプロビジョニングアーティファクトに対して AWS Service CatalogDescribeProvisioningArtifactまたは DescribeProvisioningParametersオペレーションを呼び出すと、Service Catalog はエンジンで AWS Lambda 関数を呼び出します。これは、提供されたプロビジョニングアーティファクトからパラメータのリストを抽出し、 に返すために必要です AWS Service Catalog。これらのパラメータは、後でプロビジョニングプロセスの一部として使用されます。

を呼び出してEXTERNALプロビジョニングアーティファクトをプロビジョニングするとProvisionProduct、Service Catalog はまずいくつかのアクションを内部で実行し、次にエンジンの Amazon SQS キューにメッセージを送信します。次に、エンジンは提供された起動ロール (起動制約として製品に割り当てる IAM ロール) を引き受け、提供されたプロビジョニングアーティファクトに基づいてリソースをプロビジョニングし、NotifyProvisionProductEngineWorkflowResultAPI を呼び出して成功または失敗を報告します。

UpdateProvisionedProduct と への呼び出しTerminateProvisionedProductも同様に処理され、それぞれに個別のキューと通知 APIsがあります。

- NotifyProvisionProductEngineWorkflowResult
- NotifyUpdateProvisionedProductEngineWorkflowResult

- NotifyTerminateProvisionedProductEngineWorkflowResult.

トピック

- 考慮事項
- パラメータ解析
- プロビジョニング
- [更新中]
- 終了中
- タグ付け

考慮事項

ハブアカウントあたり 1 つの外部エンジンの制限

Service Catalog ハブアカウントごとに使用できるEXTERNALプロビジョニングエンジンは 1 つだけです。Service Catalog hub-and-spokeモデルを使用すると、ハブアカウントはベースライン製品を作成し、ポートフォリオを共有できます。一方、スポークアカウントはポートフォリオをインポートして製品を活用します。

この制限は、 がアカウント内の 1 つのエンジンにのみルーティングEXTERNALできるためです。管理者が複数の外部エンジンを使用したい場合は、異なるハブアカウントで外部エンジンを (ポートフォリオや製品とともに) 設定する必要があります。

外部エンジンは、起動制約のある起動ロールのみをサポートします

EXTERNAL プロビジョニングアーティファクトは、起動制約 を使用して指定された起動ロールによるプロビジョニングのみをサポートします。起動制約は、エンドユーザーが製品を起動、更新、または終了するときに Service Catalog が引き受ける IAM ロールを指定します。起動制約の詳細については、AWS Service Catalog 「起動制約」を参照してください。

パラメータ解析

EXTERNAL プロビジョニングアーティファクトは任意の形式にすることができます。つまり、EXTERNAL製品タイプを作成する場合、エンジンは提供されたプロビジョニングアーティファクトからパラメータのリストを抽出し、Service Catalog に返す必要があります。これは、次のリクエ

スト形式を受け入れ、プロビジョニングアーティファクトを処理し、次のレスポンス形式を返すことができる Lambda 関数をアカウントに作成することによって行われます。

> **Important**
>
> Lambda 関数には という名前を付ける必要がありますServiceCatalogExternalParameterParser。

リクエストの構文:

```
{
    "artifact": {
        "path": "string",
        "type": "string"
    },
    "launchRoleArn": "string"
}
```

フィールド	タイプ	必須	説明
アーティファクト	オブジェクト	はい	解析するアーティファクトの詳細。
アーティファクト/パス	文字列	はい	パーサーがアーティファクトをダウンロードする場所。例えば、 の場合AWS_S3、これは Amazon S3 URI です。
アーティファクト/タイプ	文字列	はい	アーティファクトのタイプ。使用できる値: AWS_S3。
launchRole	string	いいえ	アーティファクトのダウンロード時に引き受ける起動ロールの Amazon リソース

フィールド	タイプ	必須	説明
			ネーム (ARN)。起動ロールが指定されていない場合は、Lambda の実行ロールが使用されます。

レスポンスの構文:

```
{
    "parameters": [
        {
            "key": "string"
            "defaultValue": "string",
            "type": "string",
            "description": "string",
            "isNoEcho": boolean
        },
    ]
}
```

フィールド	タイプ	必須	説明
parameters	list	はい	製品のプロビジョニング時またはプロビジョニング済み製品の更新時に、Service Catalog がエンドユーザーに提供するように要求するパラメータのリスト。アーティファクトにパラメータが定義されていない場合、空のリストが返されます。
キー	文字列	はい	パラメータキー。

フィールド	タイプ	必須	説明
defaultValue	string	いいえ	エンドユーザーが値を指定しない場合のパラメータのデフォルト値。
type	文字列	はい	エンジンのパラメータ値の想定されるタイプ。例えば、文字列、ブール値、マップなどです。使用できる値は、各エンジンに固有です。Service Catalog は、各パラメータ値を文字列としてエンジンに渡します。
説明	string	いいえ	パラメータの説明。これはユーザーフレンドリであることが推奨されます。
isNoEcho	ブール値	なし	パラメータ値がログにエコーされないかどうかを決定します。デフォルト値は false (パラメータ値はエコーされます) です。

プロビジョニング

ProvisionProduct オペレーションの場合、Service Catalog はリソースの実際のプロビジョニングをエンジンに委任します。エンジンは、アーティファクトで定義されているリソースをプロビジョニン

グするために、選択した IaC ソリューション (Terraform など) とやり取りする責任があります。エンジンは、結果を Service Catalog に通知する責任も負います。

Service Catalog は、すべてのプロビジョニングリクエストを という名前のアカウントの Amazon SQS キューに送信しますServiceCatalogExternalProvisionOperationQueue。

リクエストの構文:

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "productId": "string",
    "provisioningArtifactId": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "artifact": {
        "path": "string",
        "type": "string"
    },
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    },
    "parameters": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "tags": [
        {
            "key": "string",
            "value": "string"
        }
    ]
}
```

フィールド	タイプ	必須	説明
トークン	文字列	はい	このオペレーションを識別するトークン。実行結果を通知するために、トークンを Service Catalog に返す必要があります。
オペレーション	文字列	はい	このフィールドは、このオペレーションPROVISION_PRODUCT の である必要があります。
provisionedProductId	文字列	はい	プロビジョニング済み製品の ID。
provisionedProductName	文字列	はい	プロビジョニングされた製品の名前。
productId	文字列	はい	製品の ID。
provisioningArtifactId	文字列	はい	プロビジョニングアーティファクトの ID。
recordId	文字列	はい	このオペレーションの Service Catalog レコードの ID。
launchRoleArn	文字列	はい	リソースのプロビジョニングに使用する IAM ロールの Amazon リソースネーム (ARN)。

フィールド	タイプ	必須	説明
アーティファクト	オブジェクト	はい	リソースのプロビジョニング方法を定義するアーティファクトの詳細。
アーティファクト/パス	文字列	はい	エンジンがアーティファクトをダウンロードする場所。例えば、 の場合AWS_S3、これは Amazon S3 URI です。
アーティファクト/タイプ	文字列	はい	アーティファクトのタイプ。使用できる値: AWS_S3。
identity	string	いいえ	フィールドは現在使用されていません。
parameters	list	はい	ユーザーがこのオペレーションの入力として Service Catalog に入力したパラメータのキーと値のペアのリスト。
タグ	list	はい	プロビジョニングされたリソースに適用するタグとして Service Catalog に入力した key-value-pairs ユーザーのリスト。

ワークフロー結果通知：

NotifyProvisionProductEngineWorkflowResult API の詳細ページで指定されたレスポンスオブジェクトを使用して API を呼び出します。

[更新中]

UpdateProvisionedProduct オペレーションの場合、Service Catalog はリソースの実際の更新をエンジンに委任します。エンジンは、アーティファクトで定義されているリソースを更新するために、選択した IaC ソリューション (Terraform など) とインターフェイス接続する責任があります。エンジンは、結果を Service Catalog に通知する責任も負います。

Service Catalog は、 という名前のアカウントの Amazon SQS キューにすべての更新リクエストを送信しますServiceCatalogExternalUpdateOperationQueue。

リクエストの構文:

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "productId": "string",
    "provisioningArtifactId": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "artifact": {
        "path": "string",
        "type": "string"
    },
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    },
    "parameters": [
        {
            "key": "string",
            "value": "string"
        }
    ],
    "tags": [
        {
            "key": "string",
```

```
            "value": "string"
        }
    ]
}
```

フィールド	タイプ	必須	説明
トークン	文字列	はい	このオペレーションを識別するトークン。実行結果を通知するために、トークンを Service Catalog に返す必要があります。
オペレーション	文字列	はい	このフィールドは、このオペレーションUPDATE_PROVISION_PRODUCT の である必要があります。
provisionedProductId	文字列	はい	プロビジョニング済み製品の ID。
provisionedProductName	文字列	はい	プロビジョニングされた製品の名前。
productId	文字列	はい	製品の ID。
provisioningArtifactId	文字列	はい	プロビジョニングアーティファクトの ID。
recordId	文字列	はい	このオペレーションの Service Catalog レコードの ID。

フィールド	タイプ	必須	説明
launchRoleArn	文字列	はい	リソースのプロビジョニングに使用する IAM ロールの Amazon リソースネーム (ARN)。
アーティファクト	オブジェクト	はい	リソースのプロビジョニング方法を定義するアーティファクトの詳細。
アーティファクト/パス	文字列	はい	エンジンがアーティファクトをダウンロードする場所。例えば、 の場合AWS_S3、これは Amazon S3 URI です。
アーティファクト/タイプ	文字列	はい	アーティファクトのタイプ。使用できる値: AWS_S3。
identity	string	いいえ	フィールドは現在使用されていません。
parameters	list	はい	ユーザーがこのオペレーションの入力として Service Catalog に入力したパラメータのキーと値のペアのリスト。

フィールド	タイプ	必須	説明
タグ	list	はい	プロビジョニングされたリソースに適用するタグとして Service Catalog に入力した key-value-pairs ユーザーのリスト。

ワークフロー結果通知：

NotifyUpdateProvisionedProductEngineWorkflowResult API の詳細ページで指定されたレスポンスオブジェクトを使用して API を呼び出します。

終了中

TerminateProvisionedProduct オペレーションの場合、Service Catalog はリソースの実際の終了をエンジンに委任します。エンジンは、アーティファクトで定義されているリソースを終了するために、選択した IaC ソリューション (Terraform など) とやり取りする責任があります。エンジンは、結果を Service Catalog に通知する責任も負います。

Service Catalog は、すべての終了リクエストを という名前のアカウントの Amazon SQS キューに送信しますServiceCatalogExternalTerminateOperationQueue。

リクエストの構文:

```
{
    "token": "string",
    "operation": "string",
    "provisionedProductId": "string",
    "provisionedProductName": "string",
    "recordId": "string",
    "launchRoleArn": "string",
    "identity": {
        "principal": "string",
        "awsAccountId": "string",
        "organizationId": "string"
    }
```

```
}
```

フィールド	タイプ	必須	説明
トークン	文字列	はい	このオペレーションを識別するトークン。実行結果を通知するために、トークンを Service Catalog に返す必要があります。
オペレーション	文字列	はい	このフィールドは、このオペレーションTERMINATE_PROVISIONED_PRODUCT の である必要があります。
provisionedProductId	文字列	はい	プロビジョニング済み製品の ID。
provisionedProductName	文字列	はい	プロビジョニングされた製品の名前。
recordId	文字列	はい	このオペレーションの Service Catalog レコードの ID。
launchRoleArn	文字列	はい	リソースのプロビジョニングに使用する IAM ロールの Amazon リソースネーム (ARN)。
identity	string	いいえ	フィールドは現在使用されていません。

ワークフロー結果通知：

NotifyTerminateProvisionedProductEngineWorkflowResult API の詳細ページで指定されたレスポンスオブジェクトを使用して API を呼び出します。

タグ付け

Resource Groups でタグを管理するには、起動ロールに次の追加のアクセス許可ステートメントが必要です。

```
{
    "Effect": "Allow",
    "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "tag:GetResources",
        "tag:GetTagKeys",
        "tag:GetTagValues",
        "tag:TagResources",
        "tag:UntagResources"
    ],
    "Resource": "*"
}
```

> **Note**
>
> 起動ロールには、 などのアーティファクト内の特定のリソースに対するタグ付けアクセス許可も必要ですec2:CreateTags。

AWS Service Catalog におけるモニタリング

Amazon を使用してAWS Service Catalogリソースをモニタリングすることで CloudWatch、 から raw データを収集し、読み取り可能なメトリクスAWS Service Catalogに加工することができます。これらの統計情報は 2 週間記録されるため、履歴情報にアクセスしてサービスの動作をより的確に把握できます。AWS Service Catalog メトリクスデータは 1 分間隔で CloudWatch に自動的に送信されます。の詳細については CloudWatch、「Amazon CloudWatch ユーザーガイド」を参照してください。

使用可能なメトリクスとディメンションのリストについては、「AWS Service Catalog CloudWatch メトリクス」を参照してください。

モニタリングは、AWS Service Catalog と AWSソリューションの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。ただし、AWS Service Catalog のモニタリングをスタートする前に、以下の質問に対する回答を反映したモニタリング計画を作成する必要があります。

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

モニタリングツール

AWS は、AWS Service Catalog のモニタリングに使用できるさまざまなツールを提供します。これらのツールの一部はモニタリングを行うように設定できますが、一部のツールは手動による介入が必要です。モニタリングタスクをできるだけ自動化することをお勧めします。

自動モニタリングツール

Amazon CloudWatch アラームを使用して、 の中断をモニタリングAWS Service Catalogおよびレポートできます。

CloudWatch アラームは、指定した期間にわたって 1 つのメトリクスを監視し、複数の期間にわたる特定のしきい値に対するメトリクスの値に基づいて 1 つ以上のアクションを実行します。アクションは、Amazon Simple Notification Service (Amazon SNS) トピックまたは Amazon EC2 Auto Scaling ポリシーに送信される通知です。 CloudWatch alarms は、単に特定の状態にあるというだけではアクションを呼び出しません。状態が変わり、指定された期間にわたって持続している必要があります。アラームの作成方法については、「Amazon CloudWatch アラームの作成」を参照してください。で Amazon CloudWatch メトリクスを使用する方法の詳細についてはAWS Service Catalog、「」を参照してくださいAWS Service Catalog CloudWatch メトリクス。

AWS Service Catalog CloudWatch メトリクス

Amazon を使用してAWS Service Catalogリソースをモニタリングすることで CloudWatch、 から raw データを収集し、読み取り可能なメトリクスAWS Service Catalogに加工することができます。これらの統計情報は 2 週間記録されるため、履歴情報にアクセスしてサービスの動作をより的確に把握できます。AWS Service Catalog メトリクスデータは 1 分間隔で CloudWatch に自動的に送信されます。の詳細については CloudWatch、「Amazon CloudWatch ユーザーガイド」を参照してください。

トピック

- CloudWatch メトリクスの有効化
- 使用できるメトリクスとディメンション
- AWS Service Catalog メトリクスの表示

CloudWatch メトリクスの有効化

Amazon CloudWatch メトリクスはデフォルトで有効になっています。

使用できるメトリクスとディメンション

が Amazon AWS Service Catalogに送信するメトリクスとディメンション CloudWatch を以下に示します。

AWS Service Catalog メトリクス

`AWS/ServiceCatalog` 名前空間には、次のメトリクスが含まれます。

メトリクス	説明
ProvisionedProductLaunch	指定された期間内に特定の製品およびプロビジョニングアーティファクトに対して起動されたプロビジョニング済みの製品の数。 単位: カウント 有効な統計: Minimum、Maximum、Sum、Average

AWS Service Catalog メトリクスのディメンション

AWS Service Catalog は次のディメンションを Amazon に送信します CloudWatch。

ディメンション	説明
State	このディメンションは、この指定された状態で起動されたすべてのプロビジョニング済み製品に対してリクエストしたデータをフィルタリングします。これにより、起動の状態別にデータを分類するのに役立ちます。 有効な状態: SUCCEEDED、FAILED
ProductId	このディメンションは、識別された製品 ID に対してのみ、リクエストしたデータをフィルタリングします。これにより、起動予定の商品を正確に特定することができます。
ProvisioningArtifactId	このディメンションは、識別されたプロビジョニングアーティファクト ID のみに対して、リクエストしたデータをフィルタリングします。これにより、起動予定の製品のバージョンを正確に特定することができます。

AWS Service Catalog メトリクスの表示

Amazon コンソールで Amazon CloudWatch メトリクスを表示できます。Amazon CloudWatch コンソールでは、リソースを詳細でカスタマイズして表示でき、サービスで実行中のタスクの数も表示できます。

トピック

- Amazon CloudWatch コンソールでのAWS Service Catalogメトリクスの表示

Amazon CloudWatch コンソールでのAWS Service Catalogメトリクスの表示

Amazon CloudWatch コンソールでAWS Service Catalogメトリクスを表示できます。Amazon CloudWatch コンソールにはAWS Service Catalogメトリクスの詳細ビューがあり、ニーズに合わせてビューを調整できます。Amazon の詳細については CloudWatch、「Amazon CloudWatch ユーザーガイド」を参照してください。

Amazon CloudWatch コンソールでメトリクスを表示するには

1. https://console.aws.amazon.com/cloudwatch/ で Amazon CloudWatch コンソールを開きます。
2. 左のナビゲーションの [メトリクス] セクションで、[Service Catalog] を選択します。
3. 表示するメトリクスを選択します。

AWS Service Catalogを使用したAWS CloudTrailAPI コールのログ記録

AWS Service Catalog は、 のユーザーAWS CloudTrail、ロール、または AWSの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ある と統合されていますAWS Service Catalog。 は、のすべての API コールをイベントAWS Service Catalogとして CloudTrail キャプチャします。キャプチャされたコールには、AWS Service Catalog コンソールのコールと、AWS Service Catalog API オペレーションへのコードのコールが含まれます。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 Service Catalog。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 Service Catalog、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。

AWS Service Catalog 内の情報 CloudTrail

CloudTrail は、AWSアカウントの作成時に有効になります。でアクティビティが発生するとAWS Service Catalog、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。AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、「イベント履歴 での CloudTrail イベントの表示」を参照してください。

AWSのイベントなど、AWS Service Catalogアカウントのイベントの継続的なレコードについては、追跡を作成します。証跡により、 はログファイル CloudTrail を Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョン に適用されます。証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、 CloudTrail ログで収集されたデータをより詳細に分析し、それに基づく対応を行うように他の AWSサービスを設定できます。詳細については、次を参照してください:

- 「証跡作成の概要」
- AWS CloudTrail でサポートされるサービスと統合
- AWS CloudTrail の Amazon SNS 通知の設定
- 「複数のリージョンから AWS CloudTrail ログファイルを受け取る」および「複数のアカウントから AWS CloudTrail ログファイルを受け取る」

はすべてのAWS Service CatalogアクションCloudTrail を記録します。例えば、 および `UpdateProvisionedProduct`アクションを呼び出す`CreateProduct`と`CreatePortfolio`、CloudTrail ログファイルにエントリが生成されます。

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

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS サービスによって送信されたかどうか。

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

AWS Service Catalog ログファイルエントリについて

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、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に関する情報が含まれます。 CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。次の例は、`CreateApplication` API を示す CloudTrail ログエントリを示しています。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "account",
        "arn": "arn:aws:iam::12345789012:user/dev-haw",
        "accountId": "12345789012",
        "accessKeyId": "keyId",
        "userName": "dev-haw"
    },
    "eventTime": "2020-09-23T21:07:58Z",
    "eventSource": "servicecatalog-appregistry.amazonaws.com",
    "eventName": "CreateApplication",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "205.251.233.48",
    "userAgent": "aws-cli/1.18.140 Python/3.6.11
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.63",
    "requestParameters": {
        "name": "hawTestCT",
        "clientToken": "6f36d650-a086-47cf-810a-fbfab2f8ad33"
    },
    "responseElements": {
        "application": {
            "applicationArn": "arn:aws:servicecatalog:us-
east-1:12345789012:application/app-02ocuq2cie2328pv64ya78e22f",
            "applicationId": "app-02ocuq2cie2328pv64ya78e22f",
            "creationTime": 1600895277.775,
            "lastUpdateTime": 1600895277.775,
            "name": "hawTestCT",
            "tags": {}
        }
    },
    "requestID": "1b6ad353-3b06-421b-bcb4-00075a782762",
```

```
    "eventID": "0a2ca224-cdfd-4c4b-a4ed-163218ff5e2d",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "12345789012"
}
```

コンソールのブランドの設定

AWS Service Catalog は、管理者がアカウントのコンソールブランド設定を指定できるようにします。管理者はコンソールブランディングを使用して、さまざまなサイトコンポーネントの会社名、ロゴ画像、プライマリカラーとセカンダリ (アクセント) カラーを指定できます。これらのブランド設定は、コンソールを使用するときに管理者とエンドユーザーの両方に表示されます。

コンソールのブランド設定により、アカウントの外観が向上し、次のことが可能になります。

- コンソールと内部アプリケーションの間をシームレスかつ視覚的に移行できます。
- 同じ会社の異なる社内チームが使用しているアカウントを区別します。
- 開発、ステージング、本番環境など、複数の環境にわたってアカウントを区別します

> **Note**
>
> 管理者はアカウントレベルでコンソールのブランド設定を指定します。

コンソールのブランド設定を指定するには

1. 左側のナビゲーションメニューで、[設定] を選択します。
2. ライトモードまたはダークモードのブランド設定で [編集] を選択します。
3. ロゴをアップロードし、ブランド名を入力して、「プライマリカラー」と「セカンダリカラー」を選択します。
4. [保存] を選択します。

AWS Service Catalog がコンソールのブランディングをサポートしているリージョンのリストについては、AWS リージョン のコンソールブランディングのサポート を確認してください。

コンソールのブランド設定に関する AWS リージョン のサポート

AWS Service Catalog は、以下の表に示す AWS リージョン においてコンソールのブランド設定をサポートします。

AWS リージョン 名	AWS リージョン アイデンティティ	
米国東部 (バージニア北部)	us-east-1	
米国東部 (オハイオ)	us-east-2	
米国西部 (北カリフォルニア)	us-west-1	
米国西部 (オレゴン)	us-west-2	
アフリカ (ケープタウン)	af-south-1	
アジアパシフィック (香港)	ap-east-1	
アジアパシフィック (ジャカルタ)	ap-southeast-3	
アジアパシフィック (ムンバイ)	ap-south-1	
アジアパシフィック (大阪)	ap-northeast-3	
アジアパシフィック (ソウル)	ap-northeast-2	
アジアパシフィック (シンガポール)	ap-southeast-1	
アジアパシフィック (シドニー)	ap-southeast-2	
アジアパシフィック (東京)	ap-northeast-1	
カナダ (中部)	ca-central-1	
欧州 (フランクフルト)	eu-central-1	
欧州 (アイルランド)	eu-west-1	
欧州 (ロンドン)	eu-west-2	
ヨーロッパ (ミラノ)	eu-south-1	
欧州 (パリ)	eu-west-3	
欧州 (ストックホルム)	eu-north-1	

AWS リージョン 名	AWS リージョン アイデンティティ	
中東 (バーレーン)	me-south-1	
南米 (サンパウロ)	sa-east-1	
AWS GovCloud （米国東部）	us-gov-east-1	
AWS GovCloud （米国西部）	us-gov-west-1	

ドキュメント履歴

次の表は、 のドキュメントに対する重要な変更点を示しています AWS Service Catalog。このドキュメントの更新に関する通知を受け取るには、RSS フィードにサブスクライブできます。

- API バージョン: 2014-11-12
- ドキュメントの最終更新日: 2024 年 5 月 16 日

変更	説明	日付
の外部エンジン AWS Service Catalog	AWS Service Catalog は、外部エンジンに関する新しいドキュメントを追加します。外部エンジンはEXTERNAL製品タイプで表されます。EXTERNAL 製品タイプを使用すると、Terraform などのサードパーティーのプロビジョニングエンジンを統合できます。外部エンジンを使用すると、Service Catalog の機能をネイティブ AWS CloudFormation テンプレート以外にも拡張できるため、他の Instructure as Code (IaC) ツールを使用できます。詳細については、「 の外部エンジン AWS Service Catalog」を参照してください。	2024 年 5 月 16 日
セキュリティ IAM 更新	AWS Service Catalog は、AWSServiceCatalogSyncServiceRolePolicy ポリシーを更新して codestar-connections に変更しますcodeconne	2024 年 5 月 7 日

	ctions 。詳細については、「AWS Service Catalog AppRegistryのAWS マネージドポリシー」を参照してください。	

以前の更新

次の表は、2024 年 4 月 25 日 AWS Service Catalog 以前の のドキュメントリリース履歴を示しています。

機能	説明	リリース日
AWS Service Catalog	Hashicorp による Terraform ライセンスの変更と外部製品タイプへの更新については、既存の Terraform Open Source 製品およびプロビジョニング済み製品の外部製品タイプへの更新 をご覧ください。	2023 年 10 月 20 日
AWS Service Catalog	とポートフォリオを共有 AWS Organizationsし、 と同期 AWS Service Catalog できるようにする方法については AWS Organizations、AWSServiceCatalogOrgsDataSyncServiceRolePolicyポリシーとAWSServiceRoleForServiceCatalogOrgsDataSyncサービスにリンクされたロールを参照してください。	2023 年 4 月 14 日
AWS Service Catalog	git 接続製品の管理と、外部リポジトリ内のテン	2022 年 11 月 18 日

機能	説明	リリース日
	プレートを製品と同期する の許可については、「AWSServiceCatalogSyncServiceRolePolicyポリシーとAWSServiceRoleForServiceCatalogSyncサービスにリンクされたロール」を参照してください。 AWS Service Catalog AWS Service Catalog	
AWS Service Catalog AppRegistry	AppRegistry がアプリケーション、関連するリソースコレクション AWS 、およびアプリケーション属性グループを保存する方法については、「」を参照してくださいAWS Service Catalog AppRegistry。	2022 年 6 月 15 日
AWS Service Management Connector	Connectors for Jira Service Management と の詳細については ServiceNow、AWS「Service Management Connector」を参照してください。	2022 年 6 月 9 日
Connector for Jira Service Management	Connector for Jira Service Management の更新については、「AWS Service Management Connector for Jira Service Management」を参照してください。	2021 年 5 月 25 日

機能	説明	リリース日
用のコネクタ ServiceNow	Connector for の更新については ServiceNow、AWS「Service Management Connector for 」を参照してください ServiceNow。	2021 年 4 月 7 日
用のコネクタ ServiceNow	Connector for の更新については ServiceNow、AWS「Service Management Connector for 」を参照してください ServiceNow。	2020 年 9 月 24 日
AWS Service Quotas	が Service Quotas と AWS Service Catalog どのように連携するかについては、AWS Service Catalog「デフォルトの Service Quotas」を参照してください。 AWS Service Quotas	2020 年 3 月 24 日
入門ライブラリ	が提供する適切に設計された製品テンプレートのライブラリについては AWS Service Catalog、「」を参照してください。 入門ライブラリ	2020 年 3 月 10 日
バージョンガイダンス	製品バージョンガイダンスの詳細については、「バージョンガイダンス」を参照してください。	2019 年 12 月 17 日

機能	説明	リリース日
Connector for Jira Service Desk	Connector for Jira Service Desk の使用を開始するには、「AWS Service Management Connector for Jira Service Desk」を参照してください。	2019 年 11 月 21 日
用のコネクタ ServiceNow	Connector for の更新については ServiceNow、AWS「Service Management Connector for 」を参照してください ServiceNow。	2019 年 11 月 18 日
セキュリティに関する新しい章	のセキュリティの詳細については AWS Service Catalog、「 のセキュリティ」を参照してください AWS Service Catalog。	2019 年 10 月 31 日
プロビジョニング済み製品所有者の変更	プロビジョニング済み製品の所有者を変更する方法については、「プロビジョニング済み製品の所有者の変更」を参照してください。	2019 年 10 月 31 日
新しいリソースの更新の制約	プロビジョニング済み製品でタグを更新するために RESOURCE_UPDATE の制約を使用する方法の詳細については、「AWS Service Catalog タグ更新の制約」を参照してください。	2019 年 4 月 17 日

機能	説明	リリース日
用のコネクタ ServiceNow	Connector for の使用を開始するには ServiceNow、AWS「 のサービス管理コネクタ」を参照してください ServiceNow。	2019 年 3 月 19 日
のサポート AWS CloudFormation StackSets	の使用を開始するには AWS CloudFormation StackSets 、「 の使用」を参照してください AWS CloudFormation StackSets。	2018 年 11 月 14 日
セルフサービスアクション	セルフサービスアクションの使用を開始するには、「AWS CloudFormation サービスアクション」を参照してください。	2018 年 10 月 17 日
Amazon CloudWatch メトリクス	Amazon CloudWatch メトリクスの詳細については、「」を参照してくださいAWS Service CatalogAmazon CloudWatch。	2018 年 9 月 26 日
のサポート TagOptions	タグを管理するには、「ライブラリAWS Service Catalog TagOption」を参照してください。	2017 年 28 月 6 日
ポートフォリオのインポート	別の AWS アカウントから共有されているポートフォリオをインポートするには、「ポートフォリオのインポート」を参照してください。	2016 年 2 月 16 日

機能	説明	リリース日
アクセス権限情報の更新	エンドユーザーコンソールビューへのアクセスを許可するには、「エンドユーザー向けのコンソールアクセス」を参照してください。	2016 年 2 月 16 日
初回リリース	これは AWS Service Catalog 管理者ガイドの最初のリリースです。	2015 年 7 月 9 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。