開発者ガイド

# AWS Cloud Map

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Cloud Map: 開発者ガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は、Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

# Table of Contents

AWS Cloud Mapとは ..... 1
のコンポーネント AWS Cloud Map ..... 1
アクセス AWS Cloud Map ..... 2
AWS Identity and Access Management ..... 4
AWS Cloud Map 料金 ..... 4
AWS Cloud Map および AWS クラウドコンプライアンス ..... 5
使用を開始する ..... 6
セットアップする ..... 6
にサインアップする AWS ..... 6
API、 AWS CLIAWS Tools for Windows PowerShell、または にアクセスする AWS SDKs ..... 8
AWS Command Line Interface または をセットアップする AWS Tools for Windows PowerShell ..... 10
のダウンロード AWS SDK ..... 10
DNS クエリと API コール AWS Cloud Map で を使用する方法について説明します。 ..... 11
前提条件 ..... 11
ステップ 1: 名前空間を作成する ..... 12
ステップ 2: サービスを作成する ..... 12
ステップ 3: サービスインスタンスを作成する ..... 13
ステップ 4: サービスインスタンスを検出する ..... 14
ステップ 5 : クリーンアップ ..... 15
カスタム属性 AWS Cloud Map で を使用する方法について説明します。 ..... 16
前提条件 ..... 17
ステップ 1: 名前空間を作成する ..... 17
ステップ 2: DynamoDB テーブルを作成する ..... 17
ステップ 3: データサービスを作成する ..... 18
ステップ 4: 実行ロールを作成する ..... 19
ステップ 5: データを書き込む Lambda 関数を作成する ..... 19
ステップ 6: アプリサービスを作成する ..... 21
ステップ 7: データを読み取る Lambda 関数を作成する ..... 22
ステップ 8: サービスインスタンスを作成する ..... 23
ステップ 9: クライアントアプリケーションを作成して実行する ..... 24
ステップ 10: クリーンアップする ..... 26
名前空間 ..... 28
ネームスペースの作成 ..... 28

インスタンス検出オプション ..... 29
手順 ..... 32
次のステップ ..... 35
名前空間の一覧表示 ..... 35
名前空間の削除 ..... 37
サービス ..... 40
ヘルスチェックの設定 ..... 40
Route 53 ヘルスチェック ..... 41
カスタムヘルスチェック ..... 42
DNS 設定 ..... 42
ルーティングポリシー ..... 42
レコードタイプ ..... 43
サービスの作成 ..... 45
次のステップ ..... 51
サービスの更新 ..... 51
名前空間でのサービスの一覧表示 ..... 53
サービスの削除 ..... 55
サービスインスタンス ..... 57
サービスインスタンスの登録 ..... 57
サービスインスタンスの一覧表示 ..... 63
サービスインスタンスの更新 ..... 64
サービスインスタンスのカスタム属性の更新 ..... 65
サービスインスタンスの登録解除 ..... 65
セキュリティ ..... 68
Identity and Access Management ..... 68
対象者 ..... 69
アイデンティティを使用した認証 ..... 70
ポリシーを使用したアクセスの管理 ..... 73
と AWS Cloud Map の仕組み IAM ..... 76
アイデンティティベースポリシーの例 ..... 83
AWS マネージドポリシー ..... 90
AWS Cloud Map API アクセス許可リファレンス ..... 91
トラブルシューティング ..... 95
コンプライアンス検証 ..... 97
耐障害性 ..... 98
インフラストラクチャセキュリティ ..... 99

AWS PrivateLink ..... 99
モニタリング ..... 103
を使用した通話のログ AWS Cloud Map API記録 AWS CloudTrail ..... 103
データイベント ..... 105
管理イベント ..... 106
イベント例 ..... 106
リソースのタグ付け ..... 110
リソースのタグ付け方法 ..... 110
制限事項 ..... 111
AWS Cloud Map リソースのタグの更新 ..... 112
Service Quotas ..... 114
サービスクォータの管理 ..... 115
DiscoverInstances API リクエストのスロットリングを処理する ..... 116
スロットリングの適用方法 ..... 117
API スロットリングのクォータの調整 ..... 118
ドキュメント履歴 ..... 119
..... cxxi

# AWS Cloud Mapとは

AWS Cloud Map は、アプリケーションが依存するバックエンドサービスやリソースに論理名をマッピングするために使用できるフルマネージドソリューションです。また、アプリケーションがAWS SDKs、 RESTfulAPI呼び出し、または DNSクエリのいずれかを使用してリソースを検出するのにも役立ちます。 は、Amazon DynamoDB (DynamoDB ) テーブル、Amazon Simple Queue Service (Amazon SQS) キュー、Amazon Elastic Compute Cloud (Amazon EC2) インスタンスまたはAmazon Elastic Container Service (Amazon ECS) タスクを使用して構築された上位レベルのアプリケーションサービスなど、正常なリソースのみ AWS Cloud Map を提供します。

## のコンポーネント AWS Cloud Map

名前空間

開始するには、まずアプリケーションのサービスをグループ化する方法として機能する AWS Cloud Map 名前空間を作成します。名前空間は、 リソースの検索に使用する名前を識別し、API呼び出し、 のDNSクエリ、またはパブリックDNSクエリを使用して AWS Cloud Map DiscoverInstancesVPC、リソースの検索方法も指定します。通常、1 つの名前空間には 1 つのアプリケーション (例: 請求アプリケーション) のサービスがすべて含まれています。詳細については、「AWS Cloud Map 名前空間」を参照してください。

サービス

名前空間を作成したら、エンドポイントの検索に使用するリソースのタイプごとに AWS Cloud Map AWS Cloud Map サービスを作成します。たとえば、ウェブサーバーおよびデータベースサーバーのサービスを作成します。

サービスは、アプリケーションが別のウェブサーバーなどの別のリソースを追加するときに AWS Cloud Map が使用するテンプレートです。名前空間の作成DNS時に を使用してリソースを検索することを選択した場合、サービスにはウェブサーバーの検索に使用するレコードのタイプに関する情報が含まれます。サービスは、リソースの正常性を確認するかどうか、および Amazon Route 53 ヘルスチェックとサードパーティーのヘルスチェッカーのどちらを使用するかも示します。詳細については、「AWS Cloud Map サービス」を参照してください。

サービスインスタンス

アプリケーションがリソースを追加すると、コードで APIアクションを呼び出し AWS Cloud Map RegisterInstanceて、サービスに AWS Cloud Map サービスインスタンスを作成できます。

サービスインスタンスには、 DNSを使用するか、 APIアクションを使用する AWS Cloud Map DiscoverInstancesかにかかわらず、アプリケーションがリソースを見つける方法に関する情報が含まれています。

アプリケーションがリソースに接続する必要がある場合は、resource. AWS Cloud Map returns に関連付けられている名前空間とサービスを指定して、パブリッククエリDiscoverInstancesまたはプライベートDNSクエリを呼び出したり利用したりします。1 つ以上のリソースを見つける方法に関する情報を返します。サービスの作成時にヘルスチェックを指定した場合、 は正常なインスタンスのみ AWS Cloud Map を返します。詳細については、「AWS Cloud Map サービスインスタンス」を参照してください。

# アクセス AWS Cloud Map

には AWS Cloud Map 、次の方法でアクセスできます。

- AWS Management Console – このガイドの手順では、 を使用してタスク AWS Management Console を実行する方法について説明します。
- AWS SDKs – AWS を提供するプログラミング言語を使用している場合はSDK、 を使用して SDK にアクセスできます AWS Cloud Map。SDKs 認証を簡素化し、開発環境と簡単に統合し、 AWS Cloud Map コマンドへのアクセスを提供します。詳細については、「Amazon ウェブ サービスのツール」を参照してください。
- AWS Command Line Interface – 詳細については、「 ユーザーガイド」の「 の開始 AWS CLI方法AWS Command Line Interface 」を参照してください。
- AWS Tools for Windows PowerShell – 詳細については、「 ユーザーガイド」の「 の開始 AWS Tools for Windows PowerShell方法AWS Tools for Windows PowerShell 」を参照してください。
- AWS Cloud Map API – が利用できないプログラミング言語を使用している場合SDKは、AWS Cloud Map API「 リファレンス」でAPIアクションとAPIリクエストの作成方法に関する情報を参照してください。

> **Note**
>
> IPv6 クライアントサポート – 2023 年 6 月 22 日現在、すべての新しいリージョンで、IPv6クライアント AWS Cloud Map から に送信されるコマンドは新しいデュアルスタックエンドポイント () にルーティングされます`servicediscovery.<region>.api.aws`。 AWS Cloud Map IPv6専用ネットワークは、2023 年 6 月 22 日より前にリリースされた以下のリージョンのレガシー

(`servicediscovery.<region>.amazonaws.com`) とデュアルスタックの両方のエンドポイントでアクセスできます。

- 米国東部 (オハイオ) – us-east-2
- 米国東部 (バージニア北部) – us-east-1
- 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 (ハイデラバード) ap-south-2
- 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 (メルボルン) ap-southeast-4
- 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
- 欧州 (ミラノ) – eu-south-1
- 欧州 (パリ) – eu-west-3
- 欧州 (スペイン) eu-south-2
- 欧州 (ストックホルム) - eu-north-1
- 欧州 (チューリッヒ) eu-central-2
- 中東 (バーレーン) – me-south-1
- 中東 (UAE) – me-central-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

## AWS Identity and Access Management

AWS Cloud Map は、組織が以下のアクションを実行するために使用できるサービスである AWS Identity and Access Management（IAM) と統合されます。

- 組織のアカウントでユーザーとグループ AWS を作成する
- AWS アカウントのユーザー間でアカウントリソースを効率的に共有する
- 各ユーザーに一意のセキュリティ認証情報を割り当てる
- サービスやリソースに対するユーザーのアクセス権を細分化して制御する

例えば、 IAMで を使用して AWS Cloud Map 、 AWS アカウント内のどのユーザーが新しい名前空間を作成できるか、またはインスタンスを登録できるかを制御できます。

の一般的な情報についてはIAM、以下のリソースを参照してください。

- の ID とアクセスの管理 AWS Cloud Map
- AWS Identity and Access Management
- IAM ユーザーガイド

## AWS Cloud Map 料金

AWS Cloud Map の料金は、サービスレジストリに登録したリソースと、それらを検出するために行ったAPI呼び出しに基づいています。 AWS Cloud Map では前払いは行われず、使用した分だけお支払いいただきます。

オプションで、IP アドレスを持つリソースの DNSベースの検出を有効にできます。API 呼び出しまたはDNSクエリを使用してインスタンスを検出するかどうかにかかわらず、Amazon Route 53 ヘルスチェックを使用してリソースのヘルスチェックを有効にすることもできます。Route 53 DNSとヘルスチェックの使用に関連する追加料金が発生します。

詳細については、AWS Cloud Map 料金を参照してください。

# AWS Cloud Map および AWS クラウドコンプライアンス

さまざまなセキュリティコンプライアンス規制および監査標準への AWS Cloud Map 準拠については、以下のページを参照してください。

- AWS クラウドコンプライアンス
- AWS コンプライアンスプログラムによる対象範囲内のサービス

# の開始方法 AWS Cloud Map

以下のガイドでは、 AWS Cloud Map 名前空間を使用して一般的なタスクを使用 AWS Cloud Map および実行するように を設定する方法を示します。

| ガイドの概要 | 詳細はこちら |
| --- | --- |
| へのサインアップ AWS と使用の準備 AWS Cloud Map | を使用するようにセットアップする AWS Cloud Map |
| DNS クエリと API コールを使用してバックエンドサービスを検出します。 | DNS クエリと API コールで AWS Cloud Map サービス検出を使用する方法について説明します。 |
| サンプルアプリケーションを作成し、コードでカスタム属性を使用してリソースを検出します。 | カスタム属性で AWS Cloud Map サービス検出を使用する方法について説明します。 |

## を使用するようにセットアップする AWS Cloud Map

このセクションの概要と手順は、 の使用を開始し AWS 、 の使用を開始する準備をするのに役立つことを目的としています AWS Cloud Map。

トピック

- にサインアップする AWS
- API、 AWS CLIAWS Tools for Windows PowerShell、または にアクセスする AWS SDKs
- AWS Command Line Interface または をセットアップする AWS Tools for Windows PowerShell
- のダウンロード AWS SDK

## にサインアップする AWS

### 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、次の手順を実行して作成します。

にサインアップするには AWS アカウント

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

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

   にサインアップすると 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 は、サインアッププロセスが完了した後に確認 E メールを送信します。https://aws.amazon.com/ に移動し、マイアカウント を選択すると、いつでも現在のアカウントアクティビティを表示し、アカウントを管理できます。

## 管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

1. 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、「 ユーザーガイド」の AWS アカウント 「ルートユーザー (コンソール) の仮想MFAデバイスの有効化」を参照してください。 IAM

管理アクセスを持つユーザーを作成する

1. IAM Identity Center を有効にします。

手順については、「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 Identity Center で、ユーザーに管理アクセスを許可します。

   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、 AWS IAM Identity Center ユーザーガイドの「デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ」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 Identity Center ユーザーでサインインするには、IAMIdentity Center ユーザーの作成時に E メールアドレスにURL送信されたサインインを使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、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 Identity Center で、最小権限のアクセス許可を適用するベストプラクティスに従うアクセス許可セットを作成します。

   手順については、「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。

## API、 AWS CLIAWS Tools for Windows PowerShell、または にアクセスする AWS SDKs

API、 AWS CLI AWS Tools for Windows PowerShell、または を使用するには AWS SDKs、アクセスキー を作成する必要があります。これらのキーはアクセスキー ID とシークレットアクセスキーで構成されており、 AWSに行うプログラム的なリクエストに署名するために使用されます。

ユーザーが の AWS 外部とやり取りする場合は、プログラムによるアクセスが必要です AWS Management Console。プログラムによるアクセスを許可する方法は、 にアクセスするユーザーのタイプによって異なります AWS。

ユーザーにプログラマチックアクセス権を付与するには、以下のいずれかのオプションを選択します。

| プログラマチックアクセス権を必要とするユーザー | 目的 | 方法 |
| --- | --- | --- |
| ワークフォースアイデンティティ<br><br>（IAMアイデンティティセンターで管理されるユーザー） | 一時的な認証情報を使用して AWS CLI、 AWS SDKs、または へのプログラムによるリクエストに署名します AWS APIs。 | 使用するインターフェイス用の手引きに従ってください。<br><br>• については AWS CLI、AWS Command Line Interface ユーザーガイドの「 を使用する AWS CLI ように を設定する AWS IAM Identity Center」を参照してください。<br>• AWS SDKs、ツール、および については AWS APIs、「 およびツールリファレンスガイド」のIAM「アイデンティティセンター認証」を参照してください。 AWS SDKs |
| IAM | 一時的な認証情報を使用して AWS CLI、 AWS SDKs、または へのプログラムによるリクエストに署名します AWS APIs。 | IAM 「 ユーザーガイド」の「AWS リソースで一時的な認証情報を使用する」の手順に従います。 |
| IAM | (非推奨)<br>長期認証情報を使用して、AWS CLI、 AWS SDKsまたは へのプログラムによるリ | 使用するインターフェイス用の手引きに従ってください。 |

| プログラマチックアクセス権を必要とするユーザー | 目的 | 方法 |
| --- | --- | --- |
| | クエストに署名します AWS APIs。 | • については AWS CLI、AWS Command Line Interface 「 ユーザーガイド」の IAM「ユーザー認証情報を使用した認証」を参照してください。<br>• および ツールについては AWS SDKs、AWS SDKs「 および ツールリファレンスガイド」の「長期認証情報を使用した認証」を参照してください。<br>• については AWS APIs、「 ユーザーガイド」の IAM「ユーザーのアクセスキーの管理」を参照してください。 IAM |

## AWS Command Line Interface または をセットアップする AWS Tools for Windows PowerShell

AWS Command Line Interface（ AWS CLI) は、 AWS サービスを管理するための統合ツールです。をインストールして設定する方法については AWS CLI、 AWS Command Line Interface ユーザーガイドの「 のセットアップ方法 AWS Command Line Interface」を参照してください。

Windows の使用経験がある場合は PowerShell、 を使用することをお勧めします AWS Tools for Windows PowerShell。詳細については、AWS Tools for Windows PowerShell ユーザーガイドの「AWS Tools for Windows PowerShellのセットアップ」を参照してください。

## のダウンロード AWS SDK

SDK 用の AWS を提供するプログラミング言語を使用している場合は、 SDKの代わりに を使用することをお勧めします AWS Cloud Map API。の使用SDKにはいくつかの利点があります。SDKs 認証

を簡素化し、開発環境と簡単に統合し、 AWS Cloud Map コマンドへのアクセスを提供します。詳細については、Tools for Amazon Web Services を参照してください。

# DNS クエリと API コールで AWS Cloud Map サービス検出を使用する方法について説明します。

このチュートリアルでは、2 つのバックエンドサービスを使用したマイクロサービスアーキテクチャをシミュレートします。最初のサービスは、DNS クエリを使用して検出できます。2 番目のサービスは API を使用して AWS Cloud Map のみ検出できます。

> **Note**
>
> このチュートリアルでは、ドメイン名や IP アドレスなどのリソースの詳細はシミュレーションのみを目的としています。インターネット経由で解決することはできません。

## 前提条件

このチュートリアルを正常に完了するには、次の前提条件を満たす必要があります。

- 開始する前に、「を使用するようにセットアップする AWS Cloud Map」のステップを完了します。
- をまだインストールしていない場合は AWS Command Line Interface、「 の最新バージョンのインストールまたは更新 AWS CLI」の手順に従ってインストールします。

  このチュートリアルでは、コマンドを実行するためのコマンドラインターミナルまたはシェルが必要です。Linux および macOS では、任意のシェルとパッケージマネージャーを使用してください。

  > **Note**
  >
  > Windows では、Lambda でよく使用される一部の Bash CLI コマンド (`zip` など) が、オペレーティングシステムの組み込みターミナルでサポートされていません。Ubuntu および Bash の Windows 統合バージョンを取得するには、Windows Subsystem for Linux をインストールします。

- このチュートリアルでは、DNS `dig` ルックアップユーティリティコマンドを使用するローカル環境が必要です。`dig` コマンドの詳細については、「dig - DNS lookup utility」を参照してください。

## ステップ 1: AWS Cloud Map 名前空間を作成する

このステップでは、パブリック AWS Cloud Map 名前空間を作成します。 は、ユーザーに代わって、同じ名前で Route 53 ホストゾーン AWS Cloud Map を作成します。これにより、パブリック DNS レコードまたは AWS Cloud Map API コールを使用して、この名前空間で作成されたサービスインスタンスを検出できます。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. [名前空間の作成] を選択します。
3. 名前空間名 には、 を指定します`cloudmap-tutorial.com`。

   > **Note**
   > これを本番環境で使用する場合は、所有またはアクセス権のあるドメインの名前を指定する必要があります。ただし、このチュートリアルでは、実際に使用されているドメインである必要はありません。

4. ( オプション) 名前空間の説明 に、名前空間を使用する対象の説明を指定します。
5. インスタンス検出 で、API コールとパブリック DNS クエリ を選択します。
6. 残りのデフォルト値のままにして、名前空間の作成 を選択します。

## ステップ 2: AWS Cloud Map サービスを作成する

このステップでは、2 つのサービスを作成します。最初のサービスは、パブリック DNS コールと API コールを使用して検出できます。2 番目のサービスは API コールを使用してのみ検出できます。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. 左側のナビゲーションペインで名前空間 を選択して、作成した名前空間を一覧表示します。
3. 名前空間のリストから`cloudmap-tutorial.com`名前空間を選択し、詳細を表示 を選択します。

4. 「サービス」セクションで「サービスの作成」を選択し、次の操作を実行して最初のサービスを作成します。

   a. [サービス名] に `public-service` と入力します。サービス名は、 が AWS Cloud Map 作成する DNS レコードに適用されます。使用される形式は です*`<service-name>`*.*`<namespace-name>`*。

   b. サービス検出設定 で、API と DNS を選択します。

   c. DNS 設定セクションのルーティングポリシー で、複数値回答ルーティング を選択します。

   > **Note**
   >
   > コンソールは、これを選択後に MULTIVALUE に変換します。使用可能なルーティングオプションの詳細については、Route 53 デベロッパーガイドの「ルーティングポリシーの選択」を参照してください。

   d. 残りのデフォルト値のままにして、名前空間の詳細ページに戻るサービスの作成を選択します。

5. 「サービス」セクションで「サービスの作成」を選択し、次の操作を実行して 2 番目のサービスを作成します。

   a. [サービス名] に `backend-service` と入力します。

   b. サービス検出設定 で、API のみを選択します。

   c. 残りのデフォルト値のままにして、サービスの作成 を選択します。

## ステップ 3: AWS Cloud Map サービスインスタンスを登録する

このステップでは、名前空間内のサービスごとに 1 つずつ、2 つのサービスインスタンスを作成します。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. 名前空間のリストから、ステップ 1 で作成した名前空間を選択し、詳細を表示 を選択します。
3. 名前空間の詳細ページで、サービスのリストから`public-service`サービスを選択し、詳細を表示 を選択します。
4. 「サービスインスタンス」セクションで「サービスインスタンスの登録」を選択し、次の操作を実行して最初のサービスインスタンスを作成します。

   a. サービスインスタンス ID には、 を指定しますfirst。
   b. IPv4 アドレス には、 を指定します192.168.2.1。
   c. 残りのデフォルト値のままにして、サービスインスタンスの登録 を選択します。
5. ページ上部のパンくずリストを使用して、cloudmap-tutorial.com を選択して名前空間の詳細ページに戻ります。
6. 名前空間の詳細ページで、サービスのリストからバックエンドサービスを選択し、詳細を表示を選択します。
7. 「サービスインスタンス」セクションで「サービスインスタンスの登録」を選択し、次の操作を実行して 2 番目のサービスインスタンスを作成します。
   a. サービスインスタンス ID には、これが 2 番目のサービスインスタンスであるsecondことを示す を指定します。
   b. インスタンスタイプ で、別のリソース の識別情報を選択します。
   c. カスタム属性 では、 をキーとして、 をbackend値service-nameとしてキーと値のペアを追加します。
   d. [サービスインスタンスの登録] を選択します。

## ステップ 4: AWS Cloud Map サービスインスタンスを検出する

AWS Cloud Map 名前空間、サービス、およびサービスインスタンスが作成されたら、インスタンスを検出することで、すべてが機能していることを確認できます。dig コマンドを使用してパブリック DNS 設定を確認し、 AWS Cloud Map API を使用してバックエンドサービスを確認します。dig コマンドの詳細については、「dig - DNS lookup utility」を参照してください。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/route53/ で Route 53 コンソールを開きます。
2. 左のナビゲーションで、[Hosted zones] (ホストゾーン) を選択します。
3. cloudmap-tutorial.com ホストゾーンを選択します。これにより、ホストゾーンの詳細が別のペインに表示されます。ホストゾーンに関連付けられているネームサーバーをメモしておきます。次のステップで使用します。
4. dig コマンドとホストゾーンの Route 53 ネームサーバーの 1 つを使用して、サービスインスタンスの DNS レコードをクエリします。

```
dig @hosted-zone-nameserver public-service.cloudmap-tutorial.com
```

出力ANSWER SECTIONの には、public-serviceサービスに関連付けた IPv4 アドレスが表示されます。

```
;; ANSWER SECTION:
public-service.cloudmap-tutorial.com. 300 IN A 192.168.2.1
```

5. を使用して AWS CLI、2 番目のサービスインスタンスの属性をクエリします。

```
aws servicediscovery discover-instances --namespace-name cloudmap-tutorial.com --
service-name backend-service --region region
```

出力には、サービスに関連付けた属性がキーと値のペアとして表示されます。

```
{
    "Instances": [
        {
            "InstanceId": "second",
            "NamespaceName": "cloudmap-tutorial.com",
            "ServiceName": "backend-service",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "service-name": "backend"
            }
        }
    ],
    "InstancesRevision": 71462688285136850
}
```

## ステップ 5: リソースをクリーンアップする

チュートリアルを完了したら、 resources. AWS Cloud Map requires を削除できます。逆の順序でクリーンアップし、最初にサービスインスタンス、次にサービス、最後に 名前空間をクリーンアップする必要があります。これらのステップを実行すると、 AWS Cloud Map はユーザーに代わって Route 53 リソースをクリーンアップします。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。

2. 名前空間のリストからcloudmap-tutorial.com名前空間を選択し、詳細を表示 を選択します。
3. 名前空間の詳細ページで、サービスのリストからpublic-serviceサービスを選択し、詳細を表示 を選択します。
4. 「サービスインスタンス」セクションで、firstインスタンスを選択し、「登録解除」を選択します。
5. ページ上部のパンくずリストを使用して、cloudmap-tutorial.com を選択して名前空間の詳細ページに戻ります。
6. 名前空間の詳細ページで、サービスのリストからパブリックサービスを選択し、削除を選択します。
7. に対してステップ 3～6 を繰り返しますbackend-service。
8. 左側のナビゲーションで、名前空間 を選択します。
9. cloudmap-tutorial.com 名前空間を選択し、「削除」を選択します。

> **Note**
>
> はユーザーに代わって Route 53 リソースを AWS Cloud Map クリーンアップしますが、Route 53 コンソールに移動して、cloudmap-tutorial.comホストゾーンが削除されたことを確認できます。

# カスタム属性で AWS Cloud Map サービス検出を使用する方法について説明します。

このチュートリアルでは、 を使用して検出可能なカスタム属性で AWS Cloud Map サービス検出を使用する方法を示します AWS Cloud Map API。このチュートリアルでは、 を使用してクライアントアプリケーションを作成および実行する手順を説明します AWS CloudShell。アプリケーションは 2 つの Lambda 関数を使用してデータを DynamoDB テーブルに書き込み、テーブルから読み取ります。Lambda 関数と DynamoDB テーブルは、サービスインスタンス AWS Cloud Map として に登録されます。クライアントアプリケーションと Lambda 関数のコードは AWS Cloud Map 、カスタム属性を使用して、ジョブの実行に必要なリソースを検出します。

> **Important**
>
> ワークショップ中に AWS リソースを作成し、アカウントに AWS コストが発生します。コストを最小限に抑えるため、ワークショップが終了したらすぐにリソースをクリーンアップすることをお勧めします。

## 前提条件

開始する前に、「を使用するようにセットアップする AWS Cloud Map」のステップを完了します。

## ステップ 1: AWS Cloud Map 名前空間を作成する

このステップでは、 AWS Cloud Map 名前空間を作成します。名前空間は、アプリケーションのサービスをグループ化するために使用されるコンストラクトです。名前空間を作成するときは、リソースの検出方法を指定します。このチュートリアルでは、この名前空間で作成されたリソースは、カスタム属性を使用した呼び出しで AWS Cloud Map API検出できます。これについては、後のステップで詳しく説明します。

1. にサインイン AWS Management Console し、 で AWS Cloud Map コンソールを開きますhttps://console.aws.amazon.com/cloudmap/。
2. [名前空間の作成] を選択します。
3. 名前空間名 には、 を指定しますcloudmap-tutorial。
4. ( オプション) 名前空間の説明 に、名前空間を使用する対象の説明を指定します。
5. インスタンス検出 で、 のAPI呼び出しを選択します。
6. 残りのデフォルト値のままにして、名前空間の作成 を選択します。

## ステップ 2: DynamoDB テーブルを作成する

このステップでは、このチュートリアルの後半で作成したサンプルアプリケーションのデータを保存および取得するために使用される DynamoDB テーブルを作成します。

DynamoDB を作成する方法については、「DynamoDB デベロッパーガイド」の「ステップ 1: テーブルを作成する」を参照してください。次の表を使用して、指定するオプションを決定します。

| オプション | 値 |
| --- | --- |
| テーブル名 | クラウドマップ |
| パーティションキー | id |

残りの設定のデフォルト値を保持し、テーブルを作成します。

## ステップ 3: AWS Cloud Map データサービスを作成し、DynamoDB テーブルをインスタンスとして登録する

このステップでは、 AWS Cloud Map サービスを作成し、最後のステップで作成した DynamoDB テーブルをサービスインスタンスとして登録します。

1. で AWS Cloud Map コンソールを開きます。 https://console.aws.amazon.com/cloudmap/
2. 名前空間のリストから`cloudmap-tutorial`名前空間を選択し、詳細を表示 を選択します。
3. サービス セクションで、サービスの作成 を選択し、以下を実行します。
   a. [サービス名] に `data-service` と入力します。
   b. 残りのデフォルト値のままにして、サービスの作成 を選択します。
4. サービス セクションで、`data-service`サービスを選択し、詳細を表示 を選択します。
5. 「サービスインスタンス」セクションで、「サービスインスタンスの登録」を選択します。
6. 「サービスインスタンスの登録」ページで、次の操作を行います。
   a. インスタンスタイプ で、別のリソース の識別情報を選択します。
   b. サービスインスタンス ID には、 を指定します`data-instance`。
   c. 「カスタム属性」セクションで、次のキーと値のペアを指定します。
      - キー = `name`、値 = `datatable`
      - キー = `tablename`、値 = `cloudmap`
   d. 属性が以下のイメージと一致することを確認し、サービスインスタンスの登録 を選択します。

## ステップ 4: AWS Lambda 実行ロールを作成する

このステップでは、次のステップで作成した AWS Lambda 関数が使用する IAMロールを作成します。このロールはこのチュートリアルでのみ使用され`cloudmap-tutorial-role`、後で削除できるため、IAMロールに名前を付けてアクセス許可の境界を省略できます。

Lambda のサービスロールを作成するには (IAM コンソール )

1. にサインイン AWS Management Console し、 でIAMコンソールを開きますhttps://console.aws.amazon.com/iam/。
2. IAM コンソールのナビゲーションペインで、ロール を選択し、ロールの作成 を選択します。
3. 信頼できるエンティティタイプ で、AWS のサービス を選択します。
4. サービスまたはユースケース で、Lambda を選択し、Lambda ユースケースを選択します。
5. [Next (次へ)] を選択します。
6. `PowerUserAccess` ポリシーを検索し、ポリシーの横にあるボックスを選択し、次へ を選択します。
7. [Next (次へ)] を選択します。
8. ロール名 には、 を指定します`cloudmap-tutorial-role`。
9. ロールを確認したら、[Create role] (ロールを作成) を選択します。

## ステップ 5: データを書き込む Lambda 関数を作成する

このステップでは、 を使用して AWS Cloud Map API作成した AWS Cloud Map サービスをクエリすることで、DynamoDB テーブルにデータを書き込む Lambda 関数を最初から作成します。

Lambda 関数の作成の詳細については、「デベロッパーガイド」の「コンソールを使用して Lambda 関数を作成する」を参照してください。次の表を使用して、指定または選択するオプションを決定します。 AWS Lambda

| オプション | 値 |
| --- | --- |
| 関数名 | 書き込み関数 |
| ランタイム | Python 3.12 |
| アーキテクチャ | x86_64 |
| アクセス許可 | 既存のロールを使用する |
| 既存のロール | cloudmap-tutorial-role |

関数を作成したら、次の Python コードを反映するようにサンプルコードを更新し、関数をデプロイします。DynamoDB テーブル用に作成した AWS Cloud Map サービスインスタンスに関連付けられた`datatable`カスタム属性を指定することに注意してください。関数は、1 から 100 までの乱数であるキーを生成し、呼び出されたときに関数に渡される値に関連付けます。

```
import json
import boto3
import random

def lambda_handler(event, context):

    serviceclient = boto3.client('servicediscovery')

    response = serviceclient.discover_instances(
        NamespaceName='cloudmap-tutorial',
        ServiceName='data-service',
        QueryParameters={ 'name': 'datatable' })

    tablename = response["Instances"][0]["Attributes"]["tablename"]

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

    response = table.put_item(
```

```
        Item={ 'id': str(random.randint(1,100)), 'todo': event })

    return {
        'statusCode': 200,
        'body': json.dumps(response)
    }
```

関数をデプロイした後、タイムアウトエラーを回避するには、関数のタイムアウトを 5 秒に更新します。詳細については、「 AWS Lambda デベロッパーガイド」の「Lambda 関数のタイムアウトを設定する」を参照してください。

## ステップ 6: AWS Cloud Map アプリケーションサービスを作成し、Lambda 書き込み関数をインスタンスとして登録する

このステップでは、 AWS Cloud Map サービスを作成し、Lambda 書き込み関数をサービスインスタンスとして登録します。

1. で AWS Cloud Map コンソールを開きます。 https://console.aws.amazon.com/cloudmap/
2. 左側のナビゲーションで、名前空間 を選択します。
3. 名前空間のリストから`cloudmap-tutorial`名前空間を選択し、詳細を表示 を選択します。
4. サービスセクションで、サービスの作成を選択し、以下を実行します。
   a. [サービス名] に `app-service` と入力します。
   b. 残りのデフォルト値のままにして、サービスの作成 を選択します。
5. サービス セクションで、`app-service`サービスを選択し、詳細を表示 を選択します。
6. 「サービスインスタンス」セクションで、「サービスインスタンスの登録」を選択します。
7. 「サービスインスタンスの登録」ページで、次の操作を行います。
   a. インスタンスタイプ で、別のリソース の識別情報を選択します。
   b. サービスインスタンス ID には、 を指定します`write-instance`。
   c. 「カスタム属性」セクションで、次のキーと値のペアを指定します。
      - キー = `name`、値 = `writeservice`
      - キー = `function`、値 = `writefunction`
   d. 属性が以下のイメージと一致することを確認し、サービスインスタンスの登録 を選択します。

## ステップ 7: データを読み取る Lambda 関数を作成する

このステップでは、作成した DynamoDB テーブルにデータを書き込む Lambda 関数を最初から作成します。

Lambda 関数の作成の詳細については、「デベロッパーガイド」の「コンソールを使用して Lambda 関数を作成する」を参照してください。次の表を使用して、指定または選択するオプションを決定します。 AWS Lambda

| オプション | 値 |
| --- | --- |
| 関数名 | 読み取り関数 |
| ランタイム | Python 3.12 |
| アーキテクチャ | x86_64 |
| アクセス許可 | 既存のロールを使用する |
| 既存のロール | cloudmap-tutorial-role |

関数を作成したら、次の Python コードを反映するようにサンプルコードを更新し、関数をデプロイします。関数はテーブルの amd をスキャンし、すべての項目を返します。

```
import json
import boto3

def lambda_handler(event, context):
    serviceclient = boto3.client('servicediscovery')
```

```
    response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
ServiceName='data-service', QueryParameters={ 'name': 'datatable' })

    tablename = response["Instances"][0]["Attributes"]["tablename"]

    dynamodbclient = boto3.resource('dynamodb')

    table = dynamodbclient.Table(tablename)

    response = table.scan(Select='ALL_ATTRIBUTES')

    return {
        'statusCode': 200,
        'body': json.dumps(response)
    }
```

関数をデプロイした後、タイムアウトエラーを回避するには、関数のタイムアウトを 5 秒に更新します。詳細については、「 AWS Lambda デベロッパーガイド」の「Lambda 関数のタイムアウトを設定する」を参照してください。

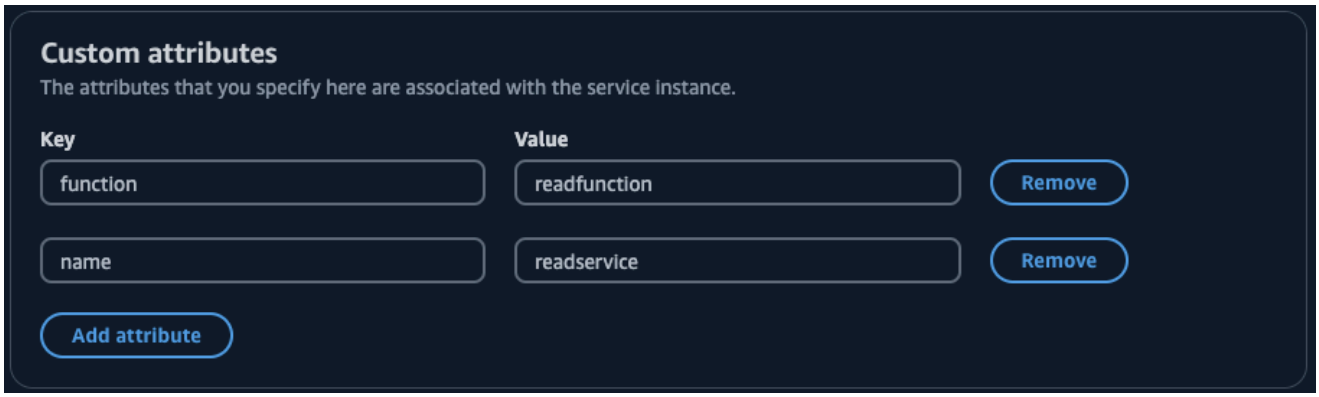
## ステップ 8: Lambda 読み取り関数を AWS Cloud Map サービスインスタンスとして登録する

このステップでは、Lambda 読み取り関数を、以前に作成したサービス内のapp-serviceサービスインスタンスとして登録します。

1. で AWS Cloud Map コンソールを開きます。 https://console.aws.amazon.com/cloudmap/
2. 左側のナビゲーションで、名前空間 を選択します。
3. 名前空間のリストからcloudmap-tutorial名前空間を選択し、詳細を表示 を選択します。
4. サービス セクションで、app-serviceサービスを選択し、詳細を表示 を選択します。
5. 「サービスインスタンス」セクションで、「サービスインスタンスの登録」を選択します。
6. 「サービスインスタンスの登録」ページで、次の操作を行います。
   a. インスタンスタイプ で、別のリソース の識別情報を選択します。
   b. サービスインスタンス ID には、 を指定しますread-instance。
   c. 「カスタム属性」セクションで、次のキーと値のペアを指定します。
      - キー = name、値 = readservice

- キー = `function`、値 = `readfunction`

d. 属性が以下のイメージと一致することを確認し、サービスインスタンスの登録 を選択します。




## ステップ 9: で読み取り/書き込みクライアントを作成して実行する AWS CloudShell

コード AWS CloudShell を使用して で設定したサービスを検出し、これらのサービスを呼び出すクライアントアプリケーションを作成 AWS Cloud Map および実行できます。

1. で AWS CloudShell コンソールを開く https://console.aws.amazon.com/cloudshell/
2. 次のコマンドを使用して、 という名前のファイルを作成します`writefunction.py`。

```
vim writeclient.py
```

3. `writeclient.py` ファイルで、 `i` ボタンを押して挿入モードを入力します。次に、次のコードをコピーして貼り付けます。このコードは、`app-service`サービス`name=writeservice`内のカスタム属性を検索してデータを書き込む Lambda 関数を検出します。DynamoDB テーブルへのデータの書き込みを担当する Lambda 関数の名前が返されます。次に、Lambda 関数が呼び出され、値としてテーブルに書き込まれるサンプルペイロードが渡されます。

```
import boto3

serviceclient = boto3.client('servicediscovery')

response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
 ServiceName='app-service', QueryParameters={ 'name': 'writeservice' })

functionname = response["Instances"][0]["Attributes"]["function"]
```

```
lambdaclient = boto3.client('lambda')

resp = lambdaclient.invoke(FunctionName=functionname, Payload='"This is a test
 data"')

print(resp["Payload"].read())
```

4. エスケープキーを押し、 と入力し`:wq`、Enter キーを押してファイルを保存し、終了します。
5. 次のコマンドを使用して Python コードを実行します。

```
python3 writeclient.py
```

出力は、次のような`200`レスポンスである必要があります。

```
b'{"statusCode": 200, "body": "{\\"ResponseMetadata\\": {\\"RequestId\\": \
\"Q0M038IT0BPBVBJK8OCKK6I6M7VV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Wed, 06
 Mar 2024 22:46:09 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\\",
 \\"content-length\\": \\"2\\", \\"connection\\": \\"keep-alive\\", \\"x-amzn-
requestid\\": \\"Q0M038IT0BPBVBJK8OCKK6I6M7VV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"x-amz-
crc32\\": \\"2745614147\\"}, \\"RetryAttempts\\": 0}}"}'
```

6. 前のステップで書き込みが成功したことを確認するには、読み取りクライアントを作成します。

   a. 次のコマンドを使用して、 という名前のファイルを作成します`readfunction.py`。

   ```
   vim readclient.py
   ```

   b. `readclient.py` ファイルで、 `i` ボタンを押して挿入モードにします。次に、次のコードをコピーして貼り付けます。このコードはテーブルをスキャンし、前のステップでテーブルに書き込んだ値を返します。

   ```
   import boto3

   serviceclient = boto3.client('servicediscovery')

   response = serviceclient.discover_instances(NamespaceName='cloudmap-tutorial',
    ServiceName='app-service', QueryParameters={ 'name': 'readservice' })

   functionname = response["Instances"][0]["Attributes"]["function"]
   ```

```
lambdaclient = boto3.client('lambda')

resp = lambdaclient.invoke(FunctionName=functionname,
 InvocationType='RequestResponse')

print(resp["Payload"].read())
```

c. エスケープキーを押し、 と入力し`:wq`、Enter キーを押してファイルを保存し、終了します。

d. 次のコマンドを使用して Python コードを実行します。

```
python3 readclient.py
```

出力は、 を実行してテーブルに書き込まれた値`writefunction.py`と、Lambda 書き込み関数で生成されたランダムキーを一覧表示する、次のようになります。

```
  b'{"statusCode": 200, "body": "{\\"Items\\": [{\\"id\\": \\"45\
\", \\"todo\\": \\"This is a test data\\"}], \\"Count\\": 1, \
\"ScannedCount\\": 1, \\"ResponseMetadata\\": {\\"RequestId\\": \
\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Thu, 25
 Jul 2024 20:43:33 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\
\", \\"content-length\\": \\"91\\", \\"connection\\": \\"keep-alive\\", \\"x-
amzn-requestid\\": \\"9JF8J6SFQCKR6IDT5JG5NOM3CNVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"1163081893\\"}, \\"RetryAttempts\\": 0}}"}'
```

## ステップ 10: リソースをクリーンアップする

チュートリアルを完了したら、追加料金が発生しないようにリソースを削除します。 AWS Cloud Map では、リソースを逆の順序でクリーンアップし、サービスインスタンスを最初にクリーンアップし、次にサービス、最後に 名前空間をクリーンアップする必要があります。次の手順では、このチュートリアルで使用する AWS Cloud Map リソースをクリーンアップする手順を説明します。

AWS Cloud Map リソースを削除するには

1. にサインイン AWS Management Console し、 で AWS Cloud Map コンソールを開きますhttps://console.aws.amazon.com/cloudmap/。
2. 名前空間のリストから`cloudmap-tutorial`名前空間を選択し、詳細を表示 を選択します。

3. 名前空間の詳細ページで、サービスのリストからdata-serviceサービスを選択し、詳細を表示を選択します。
4. 「サービスインスタンス」セクションで、data-instanceインスタンスを選択し、「登録解除」を選択します。
5. ページ上部のパンくずリストを使用して、cloudmap-tutorial.com を選択して名前空間の詳細ページに戻ります。
6. 名前空間の詳細ページで、 サービスのリストからデータサービスを選択し、「削除」を選択します。
7. app-service サービス、 および read-instanceサービスインスタンスについて、ステップ 3～6 write-instance を繰り返します。
8. 左側のナビゲーションで、名前空間 を選択します。
9. cloudmap-tutorial 名前空間を選択し、削除を選択します。

次の表に、チュートリアルで使用されている他のリソースを削除するために使用できる手順を示します。

| リソース | ステップ | |
|---|---|---|
| DynamoDB テーブル | ステップ 8: Amazon DynamoDB デベロッパーガイドの (オプション) リソースをクリーンアップする DynamoDB | |
| Lambda 関数と関連するIAM実行ロール | 「 デベロッパーガイド」の「クリーンアップAWS Lambda ►► | |

# AWS Cloud Map 名前空間

名前空間は、アプリケーションのサービスを共通の名前と検出可能性のレベルにグループ化するために使用される の論理エンティティ AWS Cloud Map です。名前空間を作成するときは、以下を指定します。

- アプリケーションがインスタンスを検出するために使用する名前。
- に登録したサービスインスタンスを検出 AWS Cloud Map できる方法。リソースをインターネット経由でパブリックに検出する必要があるか、特定の仮想プライベートクラウド (VPC) でプライベートに検出する必要があるか、または API コールでのみ検出する必要があるかを決定できます。

以下は、名前空間に関する一般的な概念です。

- 名前空間は AWS リージョン 、作成された に固有です。複数のリージョン AWS Cloud Map で を使用するには、リージョンごとに名前空間を作成する必要があります。
- VPC 内の DNS クエリによるインスタンス検出を許可する名前空間を作成すると、 AWS Cloud Map は自動的にプライベート Route 53 ホストゾーンを作成します。このホストゾーンは、複数の VPCsに関連付けることができます。詳細については、「Amazon Route 53 API AssociateVPCWithHostedZone」を参照してください。

トピック

- アプリケーションサービスをグループ化する AWS Cloud Map 名前空間の作成
- AWS Cloud Map 名前空間の一覧表示
- AWS Cloud Map 名前空間の削除

## アプリケーションサービスをグループ化する AWS Cloud Map 名前空間の作成

名前空間を作成して、API コールまたは DNS クエリを介してアプリケーションリソースを検出できるわかりやすい名前でアプリケーションのサービスをグループ化できます。

# インスタンス検出オプション

次の表は、 のさまざまなインスタンス検出オプション AWS Cloud Map と、アプリケーションのサービスと設定に応じて作成できる対応する名前空間タイプをまとめたものです。

| 名前空間タイプ | インスタンス検出方法 | 仕組み | 追加情報 |
| --- | --- | --- | --- |
| HTTP | API コール | アプリケーションのリソースは、 `DiscoverInstances` API のみを呼び出すことで他のリソースを検出できます。 | • DiscoverInstances<br>• CreateHttpNamespace |
| プライベート DNS | VPC での API コールと DNS クエリ | アプリケーションのリソースは、 `DiscoverInstances` API を呼び出し、 AWS Cloud Map が自動的に作成するプライベート Route 53 ホストゾーンのネームサーバーをクエリすることで、他のリソースを検出できます。<br><br>によって作成されたホストゾーンの名前 AWS Cloud Map は名前空間と同じで、*service-name* .*namespace-name* という形式の名 | • DiscoverInstances<br>• CreatePrivateDnsNamespace |

| 名前空間タイプ | インスタンス検出方法 | 仕組み | 追加情報 |
| --- | --- | --- | --- |
| | | 前の DNS レコードが含まれています。<br><br>**Note**<br>Route 53 リゾルバーは、プライベートホストゾーンのレコードを使用して、VPC で発生した DNS クエリを解決します。DNS クエリのドメイン名に一致するレコードが、プライベートホストゾーンに含まれていない場合、Route 53 は、`NXDOMAIN` (存在しないドメイン) でクエリに応答します。 | |

| 名前空間タイプ | インスタンス検出方法 | 仕組み | 追加情報 |
| --- | --- | --- | --- |
| パブリック DNS | API コールとパブリック DNS クエリ | アプリケーションのリソースは、`DiscoverInstances` API を呼び出し、AWS Cloud Map が自動的に作成するパブリック Route 53 ホストゾーンのネームサーバーをクエリすることで、他のリソースを検出できます。<br><br>パブリックホストゾーンの名前は名前空間と同じで、service-*name*.*namespace* -name という形式の名前の DNS レコードが含まれています。<br><br>**Note**<br>この場合の名前空間名は、登録したドメイン名である必要があります。 | • DiscoverInstances<br>• CreatePublicDnsNamespace |

## 手順

AWS CLI、、または SDK for Python を使用して名前空間を作成するには AWS Management Console、次の手順に従います。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. [名前空間の作成] を選択します。
3. 名前空間名 に、インスタンスの検出に使用される名前を入力します。

   > **Note**
   >
   > - パブリック DNS クエリ用に設定された名前空間は、最上位ドメインで終わる必要があります。例えば `.com` です。
   > - まず名前を Punycode に変換する場合は、国際化ドメイン名 (IDN) を指定します。オンラインコンバーターについては、インターネットで「punycode コンバーター」を検索してください。
   >
   > プログラムで名前空間を作成する際、国際化ドメイン名 (IDN) を Punycode に変換することもできます。たとえば、Java を使用する場合は、java.net.IDN ライブラリの `toASCII` メソッドを使って Unicode 値を Punycode に変換できます。

4. ( オプション) 名前空間の説明 に、名前空間ページと名前空間情報 の下に表示される名前空間に関する情報を入力します。この情報を使用して、名前空間を簡単に識別できます。
5. インスタンス検出 では、API コール 、VPCs 内の API コールと DNS クエリ、API コールとパブリック DNS クエリのいずれかを選択して、それぞれ HTTP、プライベート DNS、またはパブリック DNS 名前空間を作成できます。詳細については、「インスタンス検出オプション」を参照してください。

   選択内容に基づいて、以下のステップに従います。

   - VPCs で API コールと DNS クエリを選択した場合、VPC では、名前空間を関連付ける仮想プライベートクラウド (VPC) を選択します。
   - VPCs の API コールと DNS クエリ、または API コールとパブリック DNS クエリを選択した場合は、TTL に数値を秒単位で指定します。有効期限 (TTL) 値は、名前空間で作成さ

れた Route 53 ホストゾーンの認証開始 (SOA) DNS レコードの情報を DNS リゾルバーがキャッシュする時間を決定します。TTL の詳細については、「Amazon Route 53 デベロッパーガイド」の「TTL (秒 )」を参照してください。

6. ( オプション) タグ でタグを追加 を選択し、名前空間にタグを付けるキーと値を指定します。1 つ以上のタグを指定して、名前空間に追加することができます。タグを使用すると、AWS リソースをより簡単に管理できるようにリソースを分類できます。詳細については、「AWS Cloud Map リソースのタグ付け」を参照してください。
7. [名前空間の作成] を選択します。オペレーションのステータスは、 を使用して表示できますListOperations。詳細については、 API リファレンスListOperationsの「」を参照してください。 AWS Cloud Map

AWS CLI

- 希望するインスタンスディスカバリタイプのコマンドで名前空間を作成します ( *red* の値は独自の値に置換):
  - `create-http-namespace` を使用して HTTP 名前空間を作成します。HTTP 名前空間を使用して登録したサービスインスタンスは、`DiscoverInstances` リクエストを使用して検出できますが、DNS を使用して検出することはできません。

    ```
    aws servicediscovery create-http-namespace --name name-of-namespace
    ```

  - DNS に基づいてプライベート名前空間を作成します。これは、`create-private-dns-namespace`を使用して、指定した Amazon VPC 内でのみ表示されます。 リクエストまたは DNS`DiscoverInstances` を使用して、プライベート DNS 名前空間に登録されたインスタンスを検出できます。

    ```
    aws servicediscovery create-private-dns-namespace --name name-of-namespace --
    vpc vpc-xxxxxxxxx
    ```

  - `create-public-dns-namespace` を使用して DNS に基づいてパブリック名前空間を作成します。これは、インターネットで表示されます。`DiscoverInstances` リクエストまたは DNS を使用して、プライベート DNS 名前空間で登録したインスタンスを検出できます。

    ```
    aws servicediscovery create-public-dns-namespace --name name-of-namespace
    ```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. 希望するインスタンスディスカバリタイプのコマンドで名前空間を作成します ( *red* の値は独自の値に置換):

   - create_http_namespace() を使用して HTTP 名前空間を作成します。HTTP 名前空間を使用して登録したサービスインスタンスは、discover_instances() を使用して検出できますが、DNS を使用して検出することはできません。

     ```
     response = client.create_http_namespace(
         Name='name-of-namespace',
     )
     # If you want to see the response
     print(response)
     ```

   - DNS に基づいてプライベート名前空間を作成します。これは、create_private_dns_namespace()を使用して、指定した Amazon VPC 内でのみ表示されます。discover_instances()または DNS を使用して、プライベート DNS 名前空間に登録されたインスタンスを検出できます。

     ```
     response = client.create_private_dns_namespace(
         Name='name-of-namespace',
         Vpc='vpc-1c56417b',
     )
     # If you want to see the response
     print(response)
     ```

   - create_public_dns_namespace() を使用して DNS に基づいてパブリック名前空間を作成します。これは、インターネットで表示されます。discover_instances() リクエストまたは DNS を使用して、プライベート DNS 名前空間で登録したインスタンスを検出できます。

     ```
     response = client.create_public_dns_namespace(
     ```

```
    Name='name-of-namespace',
)
# If you want to see the response
print(response)
```

- レスポンスオブジェクトの例

```
{
    'OperationId': 'gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

## 次のステップ

名前空間を作成したら、名前空間にサービスを作成して、アプリケーション内の特定の目的にまとめて対応するアプリケーションリソースをグループ化できます。サービスは、アプリケーションリソースをインスタンスとして登録するためのテンプレートとして機能します。 AWS Cloud Map サービスの作成の詳細については、「」を参照してください[アプリケーションコンポーネントの AWS Cloud Map サービスの作成](#)。

# AWS Cloud Map 名前空間の一覧表示

名前空間を作成したら、以下の手順に従って作成した名前空間のリストを表示できます。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで名前空間 を選択して、名前空間のリストを表示します。名前空間は、名前、説明、インスタンス検出モード、または名前空間 ID で注文できます。検索フィールドに名前空間名または ID を入力して、特定の名前空間を検索して表示することもできます。

AWS CLI

- `list-namespaces` コマンドで名前空間を一覧表示します。

```
aws servicediscovery list-namespaces
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. list_namespaces() で名前空間を一覧表示します。

```
response = client.list_namespaces()
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'Namespaces': [
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1585354387.357,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myFirstNamespace',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z06752353VBUDTC32S84S',
                },
                'HttpProperties': {
                    'HttpName': 'myFirstNamespace',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
        {
```

```
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1586468974.698,
            'Description': 'My second namespace',
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'mySecondNamespace.com',
            'Properties': {
                'DnsProperties': {
                },
                'HttpProperties': {
                    'HttpName': 'mySecondNamespace.com',
                },
            },
            'Type': 'HTTP',
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxx',
            'CreateDate': 1587055896.798,
            'Id': 'ns-xxxxxxxxxxxxxx',
            'Name': 'myThirdNamespace.com',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z09983722P0QME1B3KC8I',
                },
                'HttpProperties': {
                    'HttpName': 'myThirdNamespace.com',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map 名前空間の削除

名前空間の使用が完了したら、削除できます。削除した名前空間は、サービスインスタンスの登録または検出に使用できなくなります。

> **Note**
>
> 名前空間を作成するときに、パブリック DNS クエリまたは VPCs、 は Amazon Route 53 パブリックホストゾーンまたはプライベートホストゾーン AWS Cloud Map を作成します。名前空間を削除すると、 は対応するホストゾーン AWS Cloud Map を削除します。

名前空間を削除する前に、すべてのサービスインスタンスを登録解除してから、名前空間で作成されたすべてのサービスを削除する必要があります。詳細については、「AWS Cloud Map サービスインスタンスの登録解除」および「AWS Cloud Map サービスの削除」を参照してください。

名前空間で作成されたインスタンスの登録を解除し、サービスを削除したら、以下の手順に従って名前空間を削除します。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. 削除する名前空間を選択し、「削除」を選択します。
4. 削除を再度選択して、サービスを削除することを確認します。

AWS CLI

- 名前空間を `delete-namespace` コマンドで削除します (*red* の値は独自の値で置き換え)。名前空間に 1 つ以上のサービスが含まれている場合、リクエストは失敗します。

```
aws servicediscovery delete-namespace --id ns-xxxxxxxxxxx
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
```

```
client = boto3.client('servicediscovery')
```

3. 名前空間を delete_namespace()で削除します (*red* の値は独自の値で置き換え)。名前空間に 1 つ以上のサービスが含まれている場合、リクエストは失敗します。

```
response = client.delete_namespace(
    Id='ns-xxxxxxxxxxx',
)
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'OperationId': 'gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map サービス

AWS Cloud Map サービスは、該当する場合、サービスの名前とDNS設定で構成されるサービスインスタンスを登録するためのテンプレートです。ヘルスチェックを設定して、サービス内のインスタンスのヘルスステータスを判断し、異常なリソースを除外することもできます。サービスはアプリケーションのコンポーネントを表すことができます。例えば、アプリケーションでの支払いを処理するリソース用のサービスを作成し、ユーザーを管理するリソース用のサービスを作成できます。

サービスを使用すると、リソースへの接続に使用できる 1 つ以上のエンドポイントを取得して、アプリケーションのリソースを見つけることができます。リソースの場所は、名前空間の設定方法に応じて、DNSクエリまたは AWS Cloud Map DiscoverInstances APIアクションを使用して行われます。 AWS Cloud Map コンソールを使用して、サービスレベルでインスタンス検出の範囲を絞り込むことができます。

以下のトピックでは、 サービスのヘルスチェックとDNS設定について説明し、サービスの作成、一覧表示、更新、削除の手順を示します。

トピック


- AWS Cloud Map サービスヘルスチェックの設定
- AWS Cloud Map サービスDNS設定
- アプリケーションコンポーネントの AWS Cloud Map サービスの作成
- AWS Cloud Map サービスの更新
- 名前空間での AWS Cloud Map サービスの一覧表示
- AWS Cloud Map サービスの削除


## AWS Cloud Map サービスヘルスチェックの設定

ヘルスチェックは、サービスインスタンスが正常かどうかを判断するのに役立ちます。サービスの作成時にヘルスチェックを設定しない場合、トラフィックはインスタンスのヘルスステータスに関係なくサービスインスタンスにルーティングされます。ヘルスチェックを設定すると、 はデフォルトで正常なリソース AWS Cloud Map を返します。の `HealthStatus`パラメータを使用して`DiscoverInstances`API、ヘルスステータスでリソースをフィルタリングし、異常なリソースのリストを取得できます。を使用して`GetInstancesHealthStatus`API、特定のサービスインスタンスのヘルスステータスを取得することもできます。

AWS Cloud Map サービスの作成時に、Route 53 ヘルスチェックまたはカスタムのサードパーティーヘルスチェックを設定できます。

## Route 53 ヘルスチェック

Amazon Route 53 ヘルスチェックの設定を指定すると、 はインスタンスを登録するたびに Route 53 ヘルスチェック AWS Cloud Map を作成し、インスタンスの登録を解除するとヘルスチェックを削除します。

パブリックDNS名前空間の場合、 はインスタンスの登録時に が AWS Cloud Map 作成する Route 53 レコードにヘルスチェックを AWS Cloud Map 関連付けます。サービスDNSの設定で Aと の両方のAAAAレコードタイプを指定すると、 IPv4 アドレスを使用してリソースのヘルスをチェックするヘルスチェック AWS Cloud Map を作成します。IPv4 アドレスで指定されたエンドポイントが異常である場合、Route 53 は レコードAと AAAAレコードの両方を異常と見なします。サービスDNSの設定でCNAMEレコードタイプを指定した場合、Route 53 ヘルスチェックを設定することはできません。

API 呼び出しを使用してインスタンスを検出する名前空間の場合、 AWS Cloud Map は Route 53 ヘルスチェックを作成します。ただし、 がヘルスチェックを関連付け AWS Cloud Map るためのDNSレコードはありません。ヘルスチェックが正常かどうかを判断するには、Route 53 コンソールまたは Amazon を使用してモニタリングを設定できます CloudWatch。Route 53 コンソールの使用方法の詳細については、Amazon Route 53 デベロッパーガイドの「ヘルスチェックが失敗した場合に通知を取得する」を参照してください。の使用の詳細については CloudWatch、「Amazon CloudWatch API リファレンスPutMetricAlarm」の「」を参照してください。

> **Note**
>
> - プライベートDNS名前空間で作成されたサービスの Amazon Route 53 ヘルスチェックを設定することはできません。
> - 各ヘルスチェックの Route 53 ヘルスチェッカーは、30 秒ごとにヘルスチェックリクエストをエンドポイント AWS リージョン に送信します。平均して、エンドポイントは約 2 秒ごとにヘルスチェックリクエストを受け取ることになります。ただし、ヘルスチェッカーは互いに調整しません。そのため、1 秒以内に複数のリクエストを受け取った後で数秒間、ヘルスチェックを受け取らないという状況が発生する場合があります。ヘルスチェックリージョンのリストについては、「リージョン」を参照してください。

Route 53 ヘルスチェックの料金については、「Route 53 の料金」を参照してください。

## カスタムヘルスチェック

インスタンスの登録時にカスタムヘルスチェックを使用する AWS Cloud Map ように を設定する場合は、サードパーティーのヘルスチェッカーを使用してリソースのヘルスを評価する必要があります。カスタムヘルスチェックは、以下の状況で役立ちます。

- インターネット経由でリソースにアクセスできないため、Route 53 ヘルスチェックを使用することができません。例えば、Amazon にあるインスタンスがあるとしますVPC。このインスタンスにはカスタムヘルスチェックを使用できます。ただし、ヘルスチェックが機能するには、ヘルスチェッカーもインスタンスVPCと同じ にある必要があります。
- リソースの場所に関係なく、サードパーティーのヘルスチェッカーを使用します。

カスタムヘルスチェックを使用する場合、特定のリソースのヘルスを直接チェック AWS Cloud Map しないでください。代わりに、サードパーティーのヘルスチェッカーはリソースのヘルスをチェックし、アプリケーションにステータスを返します。その後、アプリケーションは、このステータスを に中継する`UpdateInstanceCustomHealthStatus`リクエストを送信する必要があります AWS Cloud Map。リレーされた初期ステータスが でUNHEALTHY、30 秒`UpdateInstanceCustomHealthStatus`以内に が のステータスをリレーする別の がない場合HEALTHY、リソースは異常であることが確認されます。 はそのリソースへのトラフィックのルーティングを AWS Cloud Map 停止します。

## AWS Cloud Map サービスDNS設定

DNS クエリによるインスタンス検出をサポートする名前空間にサービスを作成すると、 は Route 53 DNSレコード AWS Cloud Map を作成します。が AWS Cloud Map 作成するすべての Route 53 DNS レコードに適用される Route 53 ルーティングポリシーとDNSレコードタイプを指定する必要があります。

### ルーティングポリシー

ルーティングポリシーは、Route 53 がサービスインスタンスの検出に使用されるDNSクエリにどのように応答するかを決定します。サポートされているルーティングポリシーと、その関連 AWS Cloud Map は次のとおりです。

加重ルーティング

Route 53 は、同じサービスを使用して登録したインスタンスの中からランダムに選択された AWS Cloud Map 1 つの AWS Cloud Map サービスインスタンスから該当する値を返します。レ

コードの加重はすべて同じであるため、トラフィック量を増減してインスタンスにルーティングすることはできません。

例えば、サービスに A レコード 1 つとヘルスチェックが含まれており、そのサービスを使用して、10 のインスタンスを登録するとします。Route 53 は、正常なインスタンスの中からランダムに選択された 1 つのインスタンスの IP アドレスを使用してDNSクエリに応答します。正常なインスタンスがない場合、Route 53 はすべてのインスタンスが正常であるかのようにDNSクエリに応答します。

サービスでヘルスチェックを定義していない場合、Route 53 はすべてのインスタンスが正常であると仮定し、ランダムに選択した 1 つのインスタンスの該当する値を返します。

詳細については、Amazon Route 53 デベロッパーガイドの加重ルーティングを参照してください。

複数値回答ルーティング

サービスでヘルスチェックを定義し、ヘルスチェックが正常であれば、Route 53 は最大 8 のインスタンスについて該当する値を返します。

例えば、サービスに 1 つのA レコードとヘルスチェックの設定が含まれているとします。このサービスを使用して、10 個のインスタンスを登録します。Route 53 は、最大 8 つの正常なインスタンスに対して IP アドレスを持つDNSクエリに応答します。正常なインスタンスが 8 つ未満の場合、Route 53 はすべての正常なインスタンスの IP アドレスですべてのDNSクエリに応答します。

サービスでヘルスチェックを定義していない場合、Route 53 はすべてのインスタンスが正常であると仮定し、最大 8 個のインスタンスの値を返します。

詳細については、Amazon Route 53 デベロッパーガイドのマルチバリューアンサールーティングを参照してください。

## レコードタイプ

Route 53 DNSレコードタイプは、サービスインスタンスの検出に使用されるDNSクエリに応答して Route 53 が返す値のタイプを決定します。指定できるさまざまなDNSレコードタイプと、クエリに応答して Route 53 によって返される関連する値は次のとおりです。

A

このタイプを指定すると、Route 53 は 192.0.2.44 などのIPv4形式のリソースの IP アドレスを返します。

AAAA

このタイプを指定すると、Route 53 は 2001:0db8:85a3:0000:0000:abcd:0001:2345 などのIPv6形式でリソースの IP アドレスを返します。

CNAME

このタイプを指定すると、Route 53 はリソースのドメイン名 (www.example.com など) を返します。

> **Note**
>
> - CNAME DNS レコードを設定するには、加重ルーティングルーティングポリシーを指定する必要があります。
> - CNAME DNS レコードを設定する場合、Route 53 ヘルスチェックを設定することはできません。

SRV

このタイプを指定すると、Route 53 は SRVレコードの値を返します。SRV レコードの値は、次の値を使用します。

```
priority weight port service-hostname
```

以下の点を考慮します。

- `priority` および `weight` の値は 1 に設定され、変更することはできません。
- の場合`port`、インスタンスの登録時にポート (AWS_INSTANCE_PORT) に指定した値 AWS Cloud Map を使用します。
- 値 `service-hostname` は、次の値を連結したものです。
  - インスタンスの登録時にサービスインスタンス ID (InstanceID) に指定する値
  - サービスの名前
  - 名前空間の名前

例えば、インスタンスを登録するときにインスタンス ID としてテストを指定するとします。サービスの名前はバックエンドで、名前空間の名前は example.com です。 AWS Cloud Map はSRVレコードの `service-hostname` 属性に次の値を割り当てます。

```
test.backend.example.com
```

> **Note**
>
> インスタンスの登録時にIPv4アドレス、IPv6アドレス、またはその両方の値を指定すると、 はAAAAレコード`service-hostname`内の の値と同じ名前の A および/またはSRVレコード AWS Cloud Map を自動的に作成します。

レコードタイプは、次の組み合わせで指定することができます。

- A
- AAAA
- A と AAAA
- CNAME
- SRV

A レコードタイプと AAAAレコードタイプを指定する場合は、インスタンスの登録時に IPv4 IP アドレス、IPv6IP アドレス、またはその両方を指定できます。

# アプリケーションコンポーネントの AWS Cloud Map サービスの作成

名前空間を作成したら、特定の目的を果たすアプリケーションのさまざまなコンポーネントを表すサービスを作成できます。例えば、支払いを処理するアプリケーション内のリソースのサービスを作成できます。

> **Note**
>
> DNS クエリによってアクセス可能な複数のサービスを、大文字と小文字だけが異なる名前で作成することはできません (例: EXAMPLE や example)。そうしようとすると、これらの

サービスが同じ DNS 名になります。API コールでのみアクセス可能な名前空間を使用する場合は、大文字と小文字によってのみ異なる名前を持つサービスを作成できます。

AWS Management Console、、および SDK for Python を使用してサービスを作成するには AWS CLI、次の手順に従います。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. [名前空間] ページで、サービスを追加する名前空間を選択します。
4. [名前空間: *namespace-name*] ページで、[サービスの作成] を選択します。
5. サービス名 には、このサービスの使用時に登録するインスタンスを説明する名前を入力します。値は、API コールまたは DNS クエリで AWS Cloud Map サービスインスタンスを検出するために使用されます。

   > **Note**
   >
   > インスタンスの登録時に SRV AWS Cloud Map レコードを作成し、特定の SRV 形式 (HAProxyなど) を必要とするシステムを使用している場合は、サービス名 に以下を指定します。
   >
   > - 例えば、アンダースコア (_) で名前を開始します。例、_exampleservice。
   > - *._protocol* で名前を終わらせる。例、._tcp。
   >
   > インスタンスを登録すると、 は SRV レコード AWS Cloud Map を作成し、サービス名と名前空間名を連結して名前を割り当てます。次に例を示します。_exampleservice._tcp.example.com

6. ( オプション) サービスの説明 に、サービスの説明を入力します。ここで入力した説明は、サービスページと各サービスの詳細ページに表示されます。
7. 名前空間が DNS クエリをサポートしている場合、サービス検出設定 で、サービスレベルで検出可能性を設定できます。API コールと DNS クエリの両方を許可するか、このサービス内のインスタンスを検出するための API コールのみを許可するかを選択します。

> **Note**
>
> API コール を選択した場合、インスタンスの登録時に は SRV レコードを作成し AWS Cloud Map ません。

API と DNS を選択した場合は、以下の手順に従って DNS レコードを設定します。DNS レコードを追加または削除できます。

1. ルーティングポリシー で、インスタンスの登録時に が作成する DNS レコード AWS Cloud Map の Amazon Route 53 ルーティングポリシーを選択します。加重ルーティングと複数値回答ルーティングのどちらかを選択できます。詳細については、「ルーティングポリシー」を参照してください。

   > **Note**
   >
   > コンソールを使用して、インスタンスの登録時に Route 53 エイリアスレコードを作成する AWS Cloud Map ように を設定することはできません。プログラム AWS Cloud Map でインスタンスを登録するときに Elastic Load Balancing ロードバランサーのエイリアスレコードを作成する場合は、ルーティングポリシー の加重ルーティング を選択します。

2. レコードタイプ で、 によって DNS クエリに応答して Route 53 が返すものを決定する DNS レコードタイプを選択します AWS Cloud Map。詳細については、「レコードタイプ」を参照してください。
3. TTL には、数値を指定して、サービスレベルで有効期限 (TTL) 値を秒単位で定義します。TTL の値は、リゾルバーが別の DNS クエリを Amazon Route 53 に転送して、更新された設定を取得するまでに、DNS リゾルバーがこのレコードの情報をキャッシュする期間を決定します。

8. ヘルスチェック設定 のヘルスチェックオプション で、サービスインスタンスに適用されるヘルスチェックのタイプを選択します。ヘルスチェックを設定しないか、インスタンスの Route 53 ヘルスチェックまたは外部ヘルスチェックのいずれかを選択できます。詳細については、「AWS Cloud Map サービスヘルスチェックの設定」を参照してください。

> **Note**
>
> Route 53 ヘルスチェックは、パブリック DNS 名前空間内のサービスに対してのみ設定できます。

Route 53 ヘルスチェック を選択した場合は、次の情報を入力します。

1. 障害しきい値 には、サービスインスタンスがヘルスステータスを変更するために合格または不合格にする必要がある連続する Route 53 ヘルスチェックの数を定義する 1 ~ 10 の数値を指定します。
2. ヘルスチェックプロトコル で、Route 53 がサービスインスタンスのヘルスをチェックするために使用する方法を選択します。
3. HTTP または HTTPS ヘルスチェックプロトコルを選択した場合、ヘルスチェックパスに、ヘルスチェックの実行時に Amazon Route 53 がリクエストするパスを指定します。パスには、ファイル `/docs/route53-health-check.html` などの任意の値を指定できます。リソースが正常である場合、2xx または 3xx 形式の HTTP ステータスコードが返されます。クエリ文字列パラメータ (`/welcome.html?language=jp&login=y` など) を含めることもできます。 AWS Cloud Map コンソールでは、先行するスラッシュ (/) 文字が自動的に追加されます。

Route 53 ヘルスチェックの詳細については、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 がヘルスチェックが正常かどうかを判断する方法」を参照してください。

9. ( オプション) タグ でタグを追加 を選択し、名前空間にタグを付けるキーと値を指定します。1 つ以上のタグを指定して、名前空間に追加することができます。タグを使用すると、AWS リソースをより簡単に管理できるようにリソースを分類できます。詳細については、「AWS Cloud Map リソースのタグ付け」を参照してください。
10. [Create service (サービスの作成)] を選択します。

AWS CLI

- `create-service` コマンドを使用してサービスを作成します。### の値を自分の値に置き換えてください。

```
aws servicediscovery create-service \
    --name service-name \
    --namespace-id  ns-xxxxxxxxxxx \
    --dns-config "NamespaceId=ns-
xxxxxxxxxxx,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

出力:

```
{
        "Service": {
        "Id": "srv-xxxxxxxxxxx",
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxx",
        "Name": "service-name",
        "NamespaceId": "ns-xxxxxxxxxxx",
        "DnsConfig": {
            "NamespaceId": "ns-xxxxxxxxxxx",
            "RoutingPolicy": "MULTIVALUE",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 60
                }
            ]
        },
        "CreateDate": 1587081768.334,
        "CreatorRequestId": "567c1193-6b00-4308-bd57-ad38a8822d25"
    }
}
```

AWS SDK for Python (Boto3)

まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。

1. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

2. を使用してサービスを作成しますcreate_service()。### の値を自分の値に置き換えてください。詳細については、「create_service」を参照してください。

```
response = client.create_service(
    DnsConfig={
        'DnsRecords': [
            {
                'TTL': 60,
                'Type': 'A',
            },
        ],
        'NamespaceId': 'ns-xxxxxxxxxxx',
        'RoutingPolicy': 'MULTIVALUE',
    },
    Name='service-name',
    NamespaceId='ns-xxxxxxxxxxx',
)
```

レスポンスオブジェクトの例

```
{
    'Service': {
        'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxx',
        'CreateDate': 1587081768.334,
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 60,
                    'Type': 'A',
                },
            ],
            'NamespaceId': 'ns-xxxxxxxxxxx',
            'RoutingPolicy': 'MULTIVALUE',
        },
        'Id': 'srv-xxxxxxxxxxx',
        'Name': 'service-name',
        'NamespaceId': 'ns-xxxxxxxxxxx',
    },
    'ResponseMetadata': {
        '...': '...',
    },
```

```
}
```

## 次のステップ

サービスを作成したら、アプリケーションリソースを、アプリケーションがリソースを見つける方法に関する情報を含むサービスインスタンスとして登録できます。 AWS Cloud Map サービスインスタンスの登録の詳細については、「」を参照してください[リソースを AWS Cloud Map サービスインスタンスとして登録する](#)。

# AWS Cloud Map サービスの更新

サービスの設定に応じて、リDNSゾルバーのタグ、Route 53 ヘルスチェックの失敗しきい値、有効期限 (TTL) を更新できます。サービスインスタンスを更新するには、次の手順を使用します。

AWS Management Console

1. にサインイン AWS Management Console し、 で AWS Cloud Map コンソールを開きますhttps://console.aws.amazon.com/cloudmap/。
2. ナビゲーションペインで [名前空間] を選択します。
3. 名前空間 ページで、サービスが作成される名前空間を選択します。
4. 名前空間で：***namespace-name*** ページ、編集するサービスを選択し、詳細の表示 を選択します。
5. サービスの場合：***service-name*** ページ、編集 を選択します。

   > **Note**
   >
   > 編集ボタンのワークフローを使用して、インスタンス検出のAPI呼び出しのみを許可するサービスの値を編集することはできません。ただし、サービスでタグを追加または削除することはできます。***service-name*** ページ。

6. 「サービスの編集」ページの「サービスの説明」で、以前に設定したサービスの説明を更新したり、新しい説明を追加したりできます。DNS リゾルバーTTLのタグを追加したり、更新したりすることもできます。
7. DNS 設定 ではTTL、 の更新期間を秒単位で指定できます。これにより、DNSリゾルバーが別のDNSクエリを Amazon Route 53 に転送して更新された設定を取得するまでに、リゾルバーがこのレコードの情報をキャッシュする時間を決定できます。

8. Route 53 ヘルスチェックを設定している場合、障害しきい値 に、サービスインスタンスがヘルスステータスを変更するために合格または不合格にする必要がある連続する Route 53 ヘルスチェックの数を定義する 1 ~ 10 の新しい数を指定できます。
9. サービスの更新 を選択します。

AWS CLI

- update-service コマンドを使用してサービスを更新します ( *red* 独自の の値 ) 。

```
aws servicediscovery update-service \
    --id  srv-xxxxxxxxxxx \
    --service "Description=new
 description,DnsConfig={DnsRecords=[{Type=A,TTL=60}]}"
```

出力:

```
{
    "OperationId": "l3pfx7f4ynndrbj3cfq5fm2qy2z37bms-5m6iaoty"
}
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. でサービスを更新する update_service() ( *red* 独自の の値 ) 。

```
response = client.update_service(
    Id='srv-xxxxxxxxxxx',
    Service={
        'DnsConfig': {
            'DnsRecords': [
                {
                    'TTL': 300,
                    'Type': 'A',
```

```
                },
            ],
        },
        'Description': "new description",
    }
)
```

レスポンスオブジェクトの例

```
{
    "OperationId": "l3pfx7f4ynndrbj3cfq5fm2qy2z37bms-5m6iaoty"
}
```

# 名前空間での AWS Cloud Map サービスの一覧表示

名前空間に作成したサービスのリストを表示するには、次の手順を実行します。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. 一覧表示するサービスを含む名前空間の名前を選択します。すべてのサービスのリストをサービスで表示し、検索フィールドにサービス名または ID を入力して、特定のサービスを検索できます。

AWS CLI

- `list-services` コマンドでサービスを一覧表示します。次のコマンドは、名前空間 ID をフィルターとして使用して、名前空間内のすべてのサービスを一覧表示します。赤の値を独自の値に置き換えます。

```
aws servicediscovery list-services --filters
 Name=NAMESPACE_ID,Values=ns-1234567890abcdef,Condition=EQ
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. list_services() でサービスを一覧表示する。

```
response = client.list_services()
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'Services': [
        {
            'Arn': 'arn:aws:servicediscovery:us-west-2:123456789012:service/srv-
xxxxxxxxxxxxxxx',
            'CreateDate': 1587081768.334,
            'DnsConfig': {
                'DnsRecords': [
                    {
                        'TTL': 60,
                        'Type': 'A',
                    },
                ],
                'RoutingPolicy': 'MULTIVALUE',
            },
            'Id': 'srv-xxxxxxxxxxxxxxx',
            'Name': 'myservice',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map サービスの削除

サービスを削除する前に、サービスを使用して登録されたサービスインスタンスはすべて、登録解除する必要があります。詳細については、「AWS Cloud Map サービスインスタンスの登録解除」を参照してください。

サービスを使用して登録されたすべてのインスタンスを登録解除したら、次の手順を実行してサービスを削除します。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. 削除するサービスを含む名前空間のオプションを選択します。
4. [名前空間:*namespace-name*] ページで、削除するサービスのオプションを選択します。
5. [削除] をクリックします。
6. サービスを削除することを確認します。

AWS CLI

- `delete-service` コマンドでサービスを削除します (*red* の値は独自の値で置き換え)。

```
aws servicediscovery delete-service --id srv-xxxxxx
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. `delete_service()`でサービスを削除します (*red* の値は独自の値で置き換え)。

```
response = client.delete_service(
```

```
    Id='srv-xxxxxx',
)
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map サービスインスタンス

サービスインスタンスには、アプリケーションのリソース (例: ウェブサーバー) を検索する方法に関する情報が含まれます。インスタンスを登録したら、DNSクエリまたは AWS Cloud Map DiscoverInstancesAPIアクションを使用してインスタンスを検索します。登録できるリソースには以下が含まれますが、これらに限定されません。

- Amazon EC2インスタンス
- Amazon DynamoDB テーブル
- Amazon S3 バケット
- Amazon Simple Queue Service (Amazon SQS) キュー
- APIs Amazon API Gateway 上にデプロイされた

サービスインスタンスの属性値を指定でき、クライアントはこれらの属性を使用して が AWS Cloud Map 返すリソースをフィルタリングできます。例えば、アプリケーションは BETAや などの特定のデプロイステージでリソースをリクエストできますPROD。バージョニングに属性を使用することもできます。

次の手順では、アプリケーション内のリソースをサービスインスタンスとして登録する方法、サービス内の登録済みインスタンスのリストを表示する方法、特定のインスタンスパラメータを編集する方法、およびインスタンスの登録を解除する方法について説明します。

トピック


- リソースを AWS Cloud Map サービスインスタンスとして登録する
- AWS Cloud Map サービスインスタンスの一覧表示
- AWS Cloud Map サービスインスタンスの更新
- AWS Cloud Map サービスインスタンスの登録解除


## リソースを AWS Cloud Map サービスインスタンスとして登録する

アプリケーションのリソースを AWS Cloud Map サービス内のインスタンスとして登録できます。例えば、ユーザーデータを管理するすべてのアプリケーションリソース`users`に対して というサービスを作成しているとします。その後、ユーザーデータをこのサービスのインスタンスとして保存するために使用する DynamoDB テーブルを登録できます。

> **Note**
>
> AWS Cloud Map コンソールでは、以下の機能は使用できません。
>
> - コンソールを使用してサービスインスタンスを登録する場合、トラフィックを Elastic Load Balancing (ELB) ロードバランサーにルーティングするエイリアスレコードを作成することはできません。インスタンスを登録する場合は、`AWS_ALIAS_DNS_NAME` 属性を含める必要があります。詳細については、「 AWS Cloud Map APIリファレンスRegisterInstance」の「」を参照してください。
> - カスタムのヘルスチェックを含むサービスを使用してインスタンスを登録する場合、カスタムのヘルスチェックの初期ステータスは指定できません。デフォルトでは、カスタムのヘルスチェックの初期ステータスは [正常] です。初期のヘルスステータスを [異常] にするには、プログラムでインスタンスを登録し、`AWS_INIT_HEALTH_STATUS` 属性を含めます。詳細については、「 AWS Cloud Map APIリファレンスRegisterInstance」の「」を参照してください。

サービスにインスタンスを登録するには、次の手順に従います。

AWS Management Console

1. にサインイン AWS Management Console し、 で AWS Cloud Map コンソールを開きますhttps://console.aws.amazon.com/cloudmap/。
2. ナビゲーションペインで [名前空間] を選択します。
3. [名前空間] ページで、サービスインスタンスを登録するためのテンプレートとして使用するサービスを含む名前空間を選択します。
4. 名前空間で：***namespace-name*** ページ、使用するサービスを選択します。
5. サービスの場合：***service-name*** 「」ページ、「サービスインスタンスの登録」を選択します。
6. サービスインスタンスの登録ページで、インスタンスタイプ を選択します。名前空間インスタンス検出設定に応じて、IP アドレス、Amazon EC2インスタンス ID、または IP アドレスを持たないリソースのその他の識別情報を指定できます。

> **Note**
>
> EC2 インスタンスはHTTP名前空間でのみ選択できます。

7. サービスインスタンス ID には、サービスインスタンスに関連付けられた識別子を指定します。

> **Note**
>
> 既存のインスタンスを更新する場合は、更新するインスタンスに関連付けられた識別子を指定します。次に、次のステップを使用して値を更新し、インスタンスを再登録します。

8. インスタンスタイプ の選択に基づいて、次の手順を実行します。

| インスタンスタイプ | ステップ | |
|---|---|---|
| IP アドレス | a. 「標準属性IPv4」で、アドレス に、アプリケーションがこのサービスインスタンスに関連付けられているリソースにアクセスできるIPv4アドレスがあれば指定します。<br>b. IPv6 アドレス には、アプリケーションがこのサービスインスタンスに関連付けられているリソースにアクセスできる IPv6 IP アドレスがあれば指定します。<br>c. ポート には、このサービスインスタンスに関連付けられているリソースにアクセスするためにアプリケーションに含める必要があるポートを指定します。サービスに SRVレコードまたは Amazon Route 53 ヘル | |

| インスタンスタイプ | ステップ | |
|---|---|---|
| | スチェックが含まれている場合、ポートが必要です。<br>d. （オプション) カスタム属性 で、リソースに関連付けるキーと値のペアを指定します。 | |
| EC2 インスタンス | a. EC2 インスタンス ID で、 AWS Cloud Map サービスEC2インスタンスとして登録する Amazon インスタンスの ID を選択します。<br>b. （オプション) カスタム属性 で、リソースに関連付けるキーと値のペアを指定します。 | |

| インスタンスタイプ | ステップ | |
|---|---|---|
| 別のリソースを特定するための情報 | a. 標準属性 で、サービス設定にCNAMEDNSレコードが含まれている場合は、 CNAMEフィールドが表示されます。にはCNAME、Route 53 がDNSクエリに応答して返すドメイン名を指定します (例: `example.com` )。<br>b. カスタム属性 で、IP アドレスまたは Amazon EC2インスタンス ID ではないリソースの識別情報をキーと値のペアとして指定します。例えば、 というキーを指定`function`し、Lambda 関数の名前を値として指定することで、Lambda 関数を登録できます。というキーを指定`name`し、プログラムによるインスタンス検出に使用できる名前を指定することもできます。 | |

9. [サービスインスタンスの登録] を選択します。

AWS CLI

- `RegisterInstance` リクエストを送信すると:
  - で指定されたサービスで定義したDNSレコードごとに`ServiceId`、対応する名前空間に関連付けられたホストゾーンでレコードが作成または更新されます。

- サービスに HealthCheckConfig が含まれる場合、ヘルスチェック設定の設定に基づいてヘルスチェックが作成されます。
- ヘルスチェックは、新しいレコードまたは更新された各レコードに関連付けられます。

register-instance コマンドを使用してサービスインスタンスを登録する ( *red* 独自の値 ) 。

```
aws servicediscovery register-instance \
    --service-id srv-xxxxxxxxx \
    --instance-id myservice-xx \
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. RegisterInstance リクエストを送信すると:
   - で指定されたサービスで定義したDNSレコードごとにServiceId、対応する名前空間に関連付けられたホストゾーンでレコードが作成または更新されます。
   - サービスに HealthCheckConfig が含まれる場合、ヘルスチェック設定の設定に基づいてヘルスチェックが作成されます。
   - ヘルスチェックは、新しいレコードまたは更新された各レコードに関連付けられます。

   サービスインスタンスを に登録する register_instance() ( *red* 独自の 値 ) 。

```
response = client.register_instance(
    Attributes={
        'AWS_INSTANCE_IPV4': '172.2.1.3',
        'AWS_INSTANCE_PORT': '808',
    },
    InstanceId='myservice-xx',
```

```
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'OperationId': '4yejorelbukcjzpnr6tlmrghsjwpngf4-k95yg2u7',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map サービスインスタンスの一覧表示

サービスを使用して登録したサービスインスタンスのリストを表示するには、次の手順を実行します。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. サービスインスタンスを一覧表示するサービスを含む名前空間の名前を選択します。
4. サービスインスタンスの作成に使用したサービスの名前を選択します。サービスインスタンス の下にインスタンスのリストが表示されます。検索フィールドにインスタンス ID を入力して、特定のインスタンスを一覧表示できます。

AWS CLI

- `list-instances` コマンドでサービスインスタンスを一覧表示します (*red* の値は独自の値で置き換え)。

```
aws servicediscovery list-instances --service-id srv-xxxxxxxxx
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. list_instances() コマンドでサービスインスタンスを一覧表示します (*red* の値は独自の値で置き換え)。

```
response = client.list_instances(
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

レスポンスオブジェクトの例

```
{
    'Instances': [
        {
            'Attributes': {
                'AWS_INSTANCE_IPV4': '172.2.1.3',
                'AWS_INSTANCE_PORT': '808',
            },
            'Id': 'i-xxxxxxxxxxxxxxxxx',
        },
    ],
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# AWS Cloud Map サービスインスタンスの更新

更新する値に応じて、次の 2 つの方法でサービスインスタンスを更新できます。

- 値の更新: カスタム属性を含め、登録時にサービスインスタンスに指定した値のいずれかを更新する場合は、サービスインスタンスを再登録し、すべての値を再指定する必要があります。「」の手順に従ってリソースを AWS Cloud Map サービスインスタンスとして登録する、サービスインスタンス ID の既存のサービスインスタンスのインスタンス ID を指定します。

  または、 `RegisterInstance` API を使用することもできます。および `InstanceIdServiceId`パラメータを使用して既存のインスタンスとサービスの ID を指定し、他の値を再指定できます。
- カスタム属性のみの更新: サービスインスタンスのカスタム属性のみを更新する場合は、インスタンスを再登録する必要はありません。これらの値のみを更新できます。サービスインスタンスのカスタム属性の更新 を参照してください。

## サービスインスタンスのカスタム属性の更新

サービスインスタンスのカスタム属性のみを更新するには

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. [名前空間] ページで、サービスインスタンスを登録するために最初に使用したサービスを含む名前空間を選択します。
4. [Namespace:***namespace-name***] ページで、サービスインスタンスの登録に使用したサービスを選択します。
5. [Service: ***service-name***] ページで、更新するサービスインスタンスの名前を選択します。
6. [Custom attributes (カスタム属性)] セクションで、[Edit (編集)] を選択します。
7. [Edit service instance: ***instance-name***] ページで、カスタム属性の追加、削除、更新を行います。既存の属性のキーと値の両方を更新できます。
8. [Update service instance (サービスインスタンスを更新)] を選択します。

# AWS Cloud Map サービスインスタンスの登録解除

サービスを削除する前に、サービスを使用して登録されたサービスインスタンスはすべて、登録解除する必要があります。

サービスインスタンスの登録を解除するには、次の手順を使用します。

AWS Management Console

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. ナビゲーションペインで [名前空間] を選択します。
3. 登録を解除するサービスインスタンスを含む名前空間のオプションを選択します。
4. 名前空間: ***namespace-name*** ページで、サービスインスタンスの登録に使用したサービスを選択します。
5. サービス: ***service-name*** ページで、登録解除するサービスインスタンスを選択します。
6. [Deregister] (登録解除) を選択します。
7. サービスインスタンスの登録を解除することを確認します。

AWS CLI

- `deregister-instance` コマンドでサービスインスタンスを登録解除します (*red* の値は独自の値で置き換え)。このコマンドは、Amazon Route 53 DNS レコードと、指定されたインスタンス用に が AWS Cloud Map 作成したヘルスチェックを削除します。

```
aws servicediscovery deregister-instance \
    --service-id srv-xxxxxxxxx \
    --instance-id myservice-53
```

AWS SDK for Python (Boto3)

1. まだBoto3がインストールしていない場合は、[こちら]のインストール、設定、使用に関する説明をBoto3参照してください。
2. Boto3をインポートしてサービスとしてservicediscoveryを使用してください。

```
import boto3
client = boto3.client('servicediscovery')
```

3. deregister-instance()でサービスインスタンスを登録解除します (*red* の値は独自の値で置き換え)。このコマンドは、Amazon Route 53 DNS レコードと、指定されたインスタンス用に が AWS Cloud Map 作成したヘルスチェックを削除します。

```
response = client.deregister_instance(
```

```
    InstanceId='myservice-53',
    ServiceId='srv-xxxxxxxxx',
)
# If you want to see the response
print(response)
```

## レスポンスオブジェクトの例

```
{
    'OperationId': '4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq',
    'ResponseMetadata': {
        '...': '...',
    },
}
```

# のセキュリティ AWS Cloud Map

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 — クラウドで AWS サービスを実行するインフラストラクチャを保護する責任 AWS は AWS にあります。 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。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。に適用されるコンプライアンスプログラムの詳細については AWS Cloud Map、「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。
- クラウド内のセキュリティ — 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 Cloud Map。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 AWS Cloud Map を達成するために を設定する方法を示します。また、 AWS Cloud Map リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック


- の ID とアクセスの管理 AWS Cloud Map
- のコンプライアンス検証 AWS Cloud Map
- の耐障害性 AWS Cloud Map
- のインフラストラクチャセキュリティ AWS Cloud Map


## の ID とアクセスの管理 AWS Cloud Map

AWS Identity and Access Management （ IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰を認証 (サインイン) し、誰に AWS

Cloud Map リソースの使用を許可する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで AWS のサービス 使用できる です。

トピック


- 対象者
- アイデンティティを使用した認証
- ポリシーを使用したアクセスの管理
- と AWS Cloud Map の仕組み IAM
- のアイデンティティベースのポリシーの例 AWS Cloud Map
- AWS の マネージドポリシー AWS Cloud Map
- AWS Cloud Map API アクセス許可リファレンス
- AWS Cloud Map ID とアクセスのトラブルシューティング


## 対象者

AWS Identity and Access Management （ IAM) の使用方法は、 で行う作業によって異なります AWS Cloud Map。

サービスユーザー – AWS Cloud Map サービスを使用してジョブを実行する場合、管理者は必要な認証情報とアクセス許可を提供します。より多くの AWS Cloud Map 機能を使用して作業を行う際には、追加のアクセス許可が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切な権限をリクエストするのに役に立ちます。 AWS Cloud Map機能にアクセスできない場合は、「AWS Cloud Map ID とアクセスのトラブルシューティング」を参照してください。

サービス管理者 – 社内の AWS Cloud Map リソースを担当している場合は、 へのフルアクセスがある可能性があります AWS Cloud Map。サービスユーザーがどの AWS Cloud Map 機能やリソースにアクセスできるかを判断するのは、お客様の仕事です。その後、IAM管理者にリクエストを送信して、サービスユーザーのアクセス許可を変更する必要があります。このページの情報を確認して、の基本概念を理解しますIAM。会社IAMで を使用する方法の詳細については AWS Cloud Map、「」を参照してくださいと AWS Cloud Map の仕組み IAM。

IAM 管理者 - IAM管理者の場合は、 へのアクセスを管理するためのポリシーの作成方法の詳細を知りたい場合があります AWS Cloud Map。で使用できる AWS Cloud Map ID ベースのポリシーの例を表示するにはIAM、「」を参照してくださいのアイデンティティベースのポリシーの例 AWS Cloud Map。

# アイデンティティを使用した認証

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

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

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

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

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

## AWS アカウント ルートユーザー

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

## フェデレーティッドアイデンティティ

ベストプラクティスとして、 では、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用してアクセスするために ID プロバイダーとのフェデレーション AWS のサービス の使用を要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブアイデンティティプロバイダー、 AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを通じて提供された認証情報 AWS のサービス を使用してアクセスするユーザーです。フェデレーティッド ID が にアクセスすると AWS アカウント、それらはロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成するか、独自の ID ソース内のユーザーとグループのセットに接続して同期し、すべての AWS アカウント とアプリケーションで使用できます。IAM Identity Center の詳細については、AWS IAM Identity Center 「 ユーザーガイド」のIAM「Identity Center とは」を参照してください。

## IAM ユーザーとグループ

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

IAM グループは、IAMユーザーのコレクションを指定する ID です。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、という名前のグループがありIAMAdmins、そのグループにIAMリソースを管理するアクセス許可を付与できます。

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

## IAM ロール

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

IAM 一時的な認証情報を持つ ロールは、以下の状況で役立ちます。

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールの詳細については、 IAM ユーザーガイドの「サードパーティー ID プロバイダーのロールの作成」を参照してください。IAM Identity Center を使用する場合は、アクセス許可セットを設定します。ID が認証された後にアクセスできる内容を制御するために、IAMIdentity Center はアクセス許可セットを のロールに関連付けますIAM。アクセス許可セットの詳細については、「AWS IAM Identity Center ユーザーガイド」の「アクセス許可セット」を参照してください。
- 一時的なIAMユーザーアクセス許可 – IAM ユーザーまたはロールは、特定のタスクに対して異なるアクセス許可を一時的に引き受けるIAMロールを引き受けることができます。
- クロスアカウントアクセス — IAMロールを使用して、別のアカウントの誰か (信頼できるプリンシパル) が自分のアカウントのリソースにアクセスすることを許可できます。クロスアカウントアクセスを許可する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスのロールとリソースベースのポリシーの違いについては、IAM「 ユーザーガイド」の「 のクロスアカウントリソースアクセスIAM」を参照してください。
- クロスサービスアクセス — 他の の機能 AWS のサービス を使用するものもあります AWS のサービス。例えば、 サービスで呼び出しを行う場合、そのサービスが Amazon でアプリケーションを実行EC2したりAmazon S3にオブジェクトを保存したりするのが一般的です。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS ) – IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストを使用します。FAS リクエストは、サービスが他の

AWS のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「アクセスセッションの転送」を参照してください。

- サービスロール – サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受けるIAMロールです。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、IAM「 ユーザーガイド」の「 にアクセス許可を委任するロールの作成 AWS のサービス」を参照してください。
- サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示することはできますが、編集することはできません。

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

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

## ポリシーを使用したアクセスの管理

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

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

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

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

## アイデンティティベースのポリシー

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

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

## リソースベースのポリシー

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

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

## アクセスコントロールリスト (ACLs )

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

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

## その他のポリシータイプ

AWS は、追加の低頻度ポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 – アクセス許可の境界は、アイデンティティベースのポリシーがIAMエンティティ (IAMユーザーまたはロール) に付与できる最大アクセス許可を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、IAM「 ユーザーガイド」のIAM「エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs ) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定するJSONポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の をグループ化して一元管理するためのサービス AWS アカウント です。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCPs) を任意のアカウントまたはすべてのアカウントに適用できます。は、各 を含むメンバーアカウントのエンティティのアクセス許可SCPを制限します AWS アカウントのルートユーザー。Organizations と の詳細についてはSCPs、AWS Organizations 「 ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「 ユーザーガイド」の「セッションポリシー」を参照してください。 IAM

### 複数のポリシータイプ

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

## と AWS Cloud Map の仕組み IAM

IAM を使用して へのアクセスを管理する前に AWS Cloud Map、 で使用できるIAM機能について説明します AWS Cloud Map。

| IAM 機能 | AWS Cloud Map サポート |
| --- | --- |
| アイデンティティベースのポリシー | あり |
| リソースベースのポリシー | なし |
| ポリシーアクション | あり |
| ポリシーリソース | はい |
| ポリシー条件キー (サービス固有) | 可能 |
| ACLs | 不可 |
| ABAC (ポリシーのタグ ) | 可能 |
| 一時的な認証情報 | 可能 |
| 転送アクセスセッション (FAS ) | 可能 |
| サービスロール | いいえ |
| サービスリンクロール | 不可 |

AWS Cloud Map および他の AWS のサービスがほとんどのIAM機能とどのように連携するかの概要については、 IAM ユーザーガイドのAWS 「 と連携する のサービスIAM」を参照してください。

## のアイデンティティベースのポリシー AWS Cloud Map

アイデンティティベースのポリシーのサポート: あり

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

IAM ID ベースのポリシーでは、許可または拒否されたアクションとリソース、およびアクションが許可または拒否される条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、 IAM ユーザーガイドのIAMJSON「ポリシー要素リファレンス」を参照してください。

のアイデンティティベースのポリシーの例 AWS Cloud Map

AWS Cloud Map ID 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のアイデンティティベースのポリシーの例 AWS Cloud Map。

## 内のリソースベースのポリシー AWS Cloud Map

リソースベースのポリシーのサポート: なし

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

クロスアカウントアクセスを有効にするには、リソースベースのポリシーのプリンシパルとして、別のアカウントのアカウントまたはIAM エンティティ全体を指定できます。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる にある場合 AWS アカウント、信頼されたアカウントのIAM管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにア

タッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーをさらに付与する必要はありません。詳細については、IAM「 ユーザーガイド」の「 のクロスアカウントリソースアクセスIAM」を参照してください。

## のポリシーアクション AWS Cloud Map

ポリシーアクションのサポート: あり

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

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

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

AWS Cloud Map アクションのリストを確認するには、「サービス認証リファレンス」の「 で定義されるアクション AWS Cloud Map」を参照してください。

のポリシーアクションは、アクションの前に次のプレフィックス AWS Cloud Map を使用します。

```
servicediscovery
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
      "servicediscovery:action1",
      "servicediscovery:action2"
         ]
```

AWS Cloud Map ID 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のアイデンティティベースのポリシーの例 AWS Cloud Map。

## のポリシーリソース AWS Cloud Map

ポリシーリソースのサポート: あり

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

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

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

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

AWS Cloud Map リソースタイプとその のリストを確認するにはARNs、「サービス認証リファレンス」の「 で定義されるリソース AWS Cloud Map」を参照してください。各リソースARNの を指定できるアクションについては、「 で定義されるアクション AWS Cloud Map」を参照してください。

AWS Cloud Map ID 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のアイデンティティベースのポリシーの例 AWS Cloud Map。

## のポリシー条件キー AWS Cloud Map

サービス固有のポリシー条件キーのサポート: あり

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

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

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

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

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

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

AWS Cloud Map 条件キーのリストを確認するには、「サービス認証リファレンス」の「 の条件キー AWS Cloud Map」を参照してください。条件キーを使用できるアクションとリソースについては、「 で定義されるアクション AWS Cloud Map」を参照してください。

AWS Cloud Map は、IAMポリシーにきめ細かなフィルタリングを提供するために使用できる以下のサービス固有の条件キーをサポートしています。

**`servicediscovery:NamespaceArn`**

関連する名前空間の Amazon リソースネーム (ARN) を指定してオブジェクトを取得できるフィルター。

**`servicediscovery:NamespaceName`**

関連する名前空間の名前を指定することで、オブジェクトの取得を可能にするフィルター。

**`servicediscovery:ServiceArn`**

関連サービスの Amazon リソースネーム (ARN) を指定してオブジェクトを取得できるフィルター。

**`servicediscovery:ServiceName`**

関連するサービスの名前を指定することで、オブジェクトの取得を可能にするフィルター。

AWS Cloud Map ID 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のアイデンティティベースのポリシーの例 AWS Cloud Map。

## ACLs の AWS Cloud Map

をサポートACLs： なし

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

## ABAC と AWS Cloud Map

サポート ABAC (ポリシーのタグ): はい

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認証戦略です。では AWS、これらの属性はタグ と呼ばれます。IAM エンティティ (ユーザーまたはロール) と多くの AWS リソースにタグをアタッチできます。エンティティとリソースのタグ付けは、 の最初のステップですABAC。次に、プリンシパルのタグがアクセスしようとしているリソースのタグと一致するときに、オペレーションを許可するABACポリシーを設計します。

ABAC は、急速に成長している環境で役立ち、ポリシー管理が面倒になる状況に役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

の詳細についてはABAC、 IAM ユーザーガイドの「 とはABAC」を参照してください。を設定する手順を含むチュートリアルを表示するにはABAC、 IAM ユーザーガイドの「属性ベースのアクセスコントロールを使用する (ABAC ) 」を参照してください。

## での一時的な認証情報の使用 AWS Cloud Map

一時的な認証情報のサポート: あり

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する方法などの詳細については、 IAM ユーザーガイドの AWS のサービス を使用する方法IAMを参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成

されます。ロールの切り替えの詳細については、IAM「 ユーザーガイド」の「ロールへの切り替え (コンソール ) 」を参照してください。

AWS CLI または を使用して、一時的な認証情報を手動で作成できます AWS API。その後、これらの一時的な認証情報を使用して にアクセスします AWS。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「」の「一時的なセキュリティ認証情報IAM」を参照してください。

## の転送アクセスセッション AWS Cloud Map

転送アクセスセッションをサポート (FAS): はい

ユーザーIAMまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可と AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストを使用します。FAS リクエストは、サービスが他の AWS のサービス または リソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するための権限が必要です。FAS リクエストを行う際のポリシーの詳細については、「アクセスセッションの転送」を参照してください。

## AWS Cloud Mapのサービスロール

サービスロールのサポート: なし

サービスロールは、ユーザーに代わってアクションを実行するためにサービスが引き受けるIAMロールです。IAM 管理者は、 内からサービスロールを作成、変更、削除できますIAM。詳細については、IAM「 ユーザーガイド」の「 にアクセス許可を委任するロールの作成 AWS のサービス」を参照してください。

> **Warning**
> サービスロールの権限を変更すると、 AWS Cloud Map の機能が破損する可能性があります。 AWS Cloud Map がガイダンスを提供する場合にのみ、サービスロールを編集します。

## のサービスにリンクされたロール AWS Cloud Map

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、 サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示することはできますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、AWS 「 と連携するサービス IAM」を参照してください。表の中から、[Service-linked role] (サービスにリンクされたロール) 列に Yes と記載されたサービスを見つけます。サービスリンクロールに関するドキュメントをサービスで表示するには、はい リンクを選択します。

## のアイデンティティベースのポリシーの例 AWS Cloud Map

デフォルトでは、 ユーザーおよびロールには、 AWS Cloud Map リソースを作成または変更する権限はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または を使用してタスクを実行することはできません AWS API。必要なリソースに対してアクションを実行するアクセス許可をユーザーに付与するには、IAM管理者はIAMポリシーを作成できます。その後、管理者はIAMポリシーをロールに追加し、ユーザーはロールを引き受けることができます。

これらのポリシードキュメント例を使用して IAM ID ベースのJSONポリシーを作成する方法については、IAM「 ユーザーガイド」のIAM「ポリシーの作成」を参照してください。

各リソースタイプの の形式など AWS Cloud Map、 で定義されるアクションとリソースタイプの詳細については、「サービス認証リファレンス」のARNs「 のアクション、リソース、および条件キー AWS Cloud Map」を参照してください。

トピック

- ポリシーのベストプラクティス
- AWS Cloud Map コンソールを使用する
- AWS Cloud Map コンソールアクセスの例
- AWS Cloud Map ユーザーに独自のアクセス許可の表示を許可する
- すべての AWS Cloud Map リソースへの読み取りアクセスを許可する
- AWS Cloud Map サービスインスタンスの例
- AWS Cloud Map サービスの作成例
- AWS Cloud Map 名前空間の作成の例

## ポリシーのベストプラクティス

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

- AWS マネージドポリシーを開始し、最小権限のアクセス許可に移行 – ユーザーとワークロードへのアクセス許可の付与を開始するには、多くの一般的なユースケースのアクセス許可を付与するAWS マネージドポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM「 ユーザーガイド」の「 管理AWS ポリシー」またはジョブ機能の 管理ポリシーを参照してください。 AWS
- 最小権限のアクセス許可を適用する – IAMポリシーでアクセス許可を設定する場合、タスクの実行に必要なアクセス許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用してアクセス許可を適用する方法の詳細については、IAM「 ユーザーガイド」の「 のポリシーとアクセス許可IAM」を参照してください。
- IAM ポリシーの条件を使用してアクセスをさらに制限する – ポリシーに条件を追加して、アクションとリソースへのアクセスを制限できます。例えば、ポリシー条件を記述して、すべてのリクエストを を使用して送信する必要があることを指定できますSSL。また、 などの特定の を通じてサービスアクションが使用されている場合 AWS のサービス、 条件を使用してサービスアクションへのアクセスを許可することもできます AWS CloudFormation。詳細については、IAM「 ユーザーガイド」のIAMJSON「ポリシー要素: 条件」を参照してください。
- IAM Access Analyzer を使用してIAMポリシーを検証し、安全で機能的なアクセス許可を確保する – IAM Access Analyzer は、ポリシーがポリシー言語 (JSON) とIAMベストプラクティスに準拠するように、新規および既存のIAMポリシーを検証します。IAM Access Analyzer には、安全で機能的なポリシーの作成に役立つ 100 を超えるポリシーチェックと実用的なレコメンデーションが用意されています。詳細については、IAM「 ユーザーガイド」のIAM「Access Analyzer ポリシーの検証」を参照してください。
- 多要素認証が必要 (MFA ) – でIAMユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、 をオンにMFAしてセキュリティを強化します。API オペレーションが呼び出されるMFAタイミングを要求するには、ポリシーにMFA条件を追加します。詳細については、IAM「 ユーザーガイド」のMFA「保護APIアクセスの設定」を参照してください。

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

## AWS Cloud Map コンソールを使用する

AWS Cloud Map コンソールにアクセスするには、最小限のアクセス許可のセットが必要です。これらのアクセス許可により、 内の AWS Cloud Map リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または のみを呼び出すユーザーに対して、最小限のコンソールアクセス許可を付与する必要はありません AWS API。代わりに、実行しようとしているAPIオペレーションに一致するアクションのみへのアクセスを許可します。

ユーザーとロールが AWS Cloud Map 引き続きコンソールを使用できるようにするには、 または *ReadOnly* AWS 管理ポリシーを AWS Cloud Map *ConsoleAccess*エンティティにアタッチします。詳細については、「 ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。 IAM

## AWS Cloud Map コンソールアクセスの例

AWS Cloud Map コンソールへのフルアクセスを許可するには、次のアクセス許可ポリシーでアクセス許可を付与します。

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:*",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "route53:CreateHostedZone",
            "route53:DeleteHostedZone",
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
```

```
            "ec2:DescribeInstances",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
      }
   ]
}
```

アクセス許可が必要な理由は次のとおりです。

**servicediscovery:***

すべての AWS Cloud Map アクションを実行できます。

**route53:CreateHostedZone**, **route53:GetHostedZone**, **route53:ListHostedZonesByName**, **route53:DeleteHostedZone**

パブリックおよびプライベートDNS名前空間を作成および削除するときに、ホストゾーン AWS Cloud Map を管理できます。

**route53:CreateHealthCheck**, **route53:GetHealthCheck**, **route53:DeleteHealthCheck**, **route53:UpdateHealthCheck**

サービスの作成時に Amazon Route 53 ヘルスチェックを含めると、ヘルスチェック AWS Cloud Map を管理できます。

**ec2:DescribeVpcs** および **ec2:DescribeRegions**

プライベートホストゾーン AWS Cloud Map を管理できるようにします。

## AWS Cloud Map ユーザーに独自のアクセス許可の表示を許可する

この例では、IAMユーザーがユーザー ID にアタッチされているインラインポリシーとマネージドポリシーを表示できるようにするポリシーを作成する方法を示します。このポリシーには、コンソールで、または AWS CLI または を使用してプログラムでこのアクションを実行するアクセス許可が含まれています AWS API。

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

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

## すべての AWS Cloud Map リソースへの読み取りアクセスを許可する

次のアクセス許可ポリシーでは、すべての AWS Cloud Map リソースへの読み取り専用アクセスをユーザーに付与します。

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:Get*",
            "servicediscovery:List*",
            "servicediscovery:DiscoverInstances"
         ],
```

```
         "Resource":"*"
      }
   ]
}
```

## AWS Cloud Map サービスインスタンスの例

次の例は、サービスインスタンスを登録、登録解除、検出するアクセス許可をユーザーに付与するアクセス許可ポリシーを示しています。`Sid` (ステートメント ID) はオプションです。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid" : "AllowInstancePermissions",
         "Effect": "Allow",
         "Action": [
            "servicediscovery:RegisterInstance",
            "servicediscovery:DeregisterInstance",
            "servicediscovery:DiscoverInstances",
            "servicediscovery:Get*",
            "servicediscovery:List*",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "route53:ChangeResourceRecordSets",
            "route53:CreateHealthCheck",
            "route53:GetHealthCheck",
            "route53:DeleteHealthCheck",
            "route53:UpdateHealthCheck",
            "ec2:DescribeInstances"
         ],
         "Resource": "*"
      }
   ]
}
```

このポリシーでは、サービスインスタンスの登録と管理に必要なアクションに対するアクセス許可が付与されます。パブリックまたはプライベートDNS名前空間を使用している場合は、Route 53 アクセス許可が必要です。インスタンスを登録および登録解除するときに Route 53 レコードとヘルスチェックが AWS Cloud Map によって作成、更新、および削除されます。のワイルドカード文字 (*) は、すべての AWS Cloud Map インスタンス、および現在の AWS アカウントが所有する Route 53 レコードとヘルスチェックへのアクセス`Resource`を許可します。

## AWS Cloud Map サービスの作成例

ID がサービスIAMを作成 AWS Cloud Map できるようにするアクセス許可ポリシーを追加する場合は、リソースフィールドに名前空間とサービスの両方 AWS Cloud Map の Amazon リソースネーム (ARN) を指定する必要があります。ARN には、リージョン、アカウント ID、名前空間 ID が含まれます。サービスのサービス ID がまだわからないため、ワイルドカードを使用することをお勧めします。以下はポリシースニペットの例です。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:CreateService"
         ],
         "Resource":[
            "arn:aws:servicediscovery:region:111122223333:namespace/ns-p32123EXAMPLE",
            "arn:aws:servicediscovery:region:111122223333:service/*"
         ]
      }
   ]
}
```

## AWS Cloud Map 名前空間の作成の例

次のアクセス許可ポリシーでは、ユーザーはすべてのタイプの AWS Cloud Map 名前空間を作成できます。

```
{
   "Version": "2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "servicediscovery:CreateHttpNamespace",
            "servicediscovery:CreatePrivateDnsNamespace",
            "servicediscovery:CreatePublicDnsNamespace",
            "route53:CreateHostedZone",
            "route53:GetHostedZone",
            "route53:ListHostedZonesByName",
            "ec2:DescribeVpcs",
```

```
            "ec2:DescribeRegions"
         ],
         "Resource":"*"
      }
   ]
}
```

## AWS の マネージドポリシー AWS Cloud Map

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) が更新されます。 AWS のサービス は、新しい AWS が起動されたとき、または既存のサービスで新しいAPIオペレーションが使用可能になったときに、 AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については、「 ユーザーガイド」の「 AWS 管理ポリシーIAM」を参照してください。

### AWS マネージドポリシー： AWSCloudMapDiscoverInstanceAccess

IAM エンティティ`AWSCloudMapDiscoverInstanceAccess`に をアタッチできます。 AWS Cloud Map Discovery へのアクセスを提供しますAPI。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSCloudMapDiscoverInstanceAccess」の「」を参照してください。 AWS

### AWS マネージドポリシー: AWSCloudMapReadOnlyAccess

IAM エンティティ`AWSCloudMapReadOnlyAccess`に をアタッチできます。すべての AWS Cloud Map アクションへの読み取り専用アクセスを許可します。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSCloudMapReadOnlyAccess」の「」を参照してください。 AWS

### AWS マネージドポリシー： AWSCloudMapRegisterInstanceAccess

IAM エンティティ`AWSCloudMapRegisterInstanceAccess`に をアタッチできます。名前空間とサービスへの読み取り専用アクセス権を付与し、サービスインスタンスの登録と登録解除を行うアクセス許可を付与します。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSCloudMapRegisterInstanceAccess」の「」を参照してください。 AWS

### AWS マネージドポリシー: AWSCloudMapFullAccess

IAM エンティティ`AWSCloudMapFullAccess`に をアタッチできます。すべての AWS Cloud Map アクションへのフルアクセスを提供します

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSCloudMapFullAccess」の「」を参照してください。 AWS

### AWS Cloud MapAWS 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 AWS Cloud Map 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、AWS Cloud Map 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ます。

| 変更 | 説明 | 日付 |
| --- | --- | --- |
| AWSCloudMapDiscoverInstanceAccess、AWSCloudMapRegisterInstanceAccess、AWSCloudMapReadOnlyAccess – 既存のポリシーの更新。 | AWS Cloud Map は、これらのポリシーを更新して、新しい AWS Cloud Map `DiscoverInstanceRevision` APIオペレーションへのアクセスを提供しました。 | 2023 年 8 月 15 日 |

## AWS Cloud Map API アクセス許可リファレンス

アクセスコントロールを設定し、IAMアイデンティティにアタッチできるアクセス許可ポリシー (アイデンティティベースのポリシー) を記述するときは、次のリストを参照として使用できます。リストには、アクセス許可を付与する必要がある各 AWS Cloud Map APIアクションとアクションが含ま

れます。ポリシーの Action フィールドにアクションを指定します。Resource フィールドまたはIAMポリシーで指定する必要があるリソース値の詳細については、「サービス認証リファレンス」の「 のアクション、リソース、および条件キー AWS Cloud Map」を参照してください。

一部のオペレーションでは、 AWS Cloud Map- 特定の条件キーをIAMポリシーで使用できます。詳細については、「サービス認証リファレンス」の「 の条件キー AWS Cloud Map」を参照してください。

アクションを指定するには、servicediscoveryプレフィックスの後にAPIアクション名、例えば servicediscovery:CreatePublicDnsNamespaceと を使用しますroute53:CreateHostedZone。

## AWS Cloud Map アクションに必要なアクセス許可

### CreateHttpNamespace

必要なアクセス許可 (API アクション):

- servicediscovery:CreateHttpNamespace

### CreatePrivateDnsNamespace

必要なアクセス許可 (API アクション):

- servicediscovery:CreatePrivateDnsNamespace
- route53:CreateHostedZone
- route53:GetHostedZone
- route53:ListHostedZonesByName
- ec2:DescribeVpcs
- ec2:DescribeRegions

### CreatePublicDnsNamespace

必要なアクセス許可 (API アクション):

- servicediscovery:CreatePublicDnsNamespace
- route53:CreateHostedZone
- route53:GetHostedZone
- route53:ListHostedZonesByName

### CreateService

必要なアクセス許可 (API アクション): servicediscovery:CreateService

DeleteNamespace

必要なアクセス許可 (API アクション):

- `servicediscovery:DeleteNamespace`

DeleteService

必要なアクセス許可 (API アクション): `servicediscovery:DeleteService`

DeregisterInstance

必要なアクセス許可 (API アクション):

- `servicediscovery:DeregisterInstance`
- `route53:GetHealthCheck`
- `route53:DeleteHealthCheck`
- `route53:UpdateHealthCheck`
- `route53:ChangeResourceRecordSets`

DiscoverInstances

必要なアクセス許可 (API アクション): `servicediscovery:DiscoverInstances`

GetInstance

必要なアクセス許可 (API アクション): `servicediscovery:GetInstance`

GetInstancesHealthStatus

必要なアクセス許可 (API アクション): `servicediscovery:GetInstancesHealthStatus`

GetNamespace

必要なアクセス許可 (API アクション): `servicediscovery:GetNamespace`

GetOperation

必要なアクセス許可 (API アクション): `servicediscovery:GetOperation`

GetService

必要なアクセス許可 (API アクション): `servicediscovery:GetService`

ListInstances

必要なアクセス許可 (API アクション): `servicediscovery:ListInstances`

ListNamespaces

必要なアクセス許可 (API アクション): `servicediscovery:ListNamespaces`

ListOperations

必要なアクセス許可 (API アクション): `servicediscovery:ListOperations`

ListServices

必要なアクセス許可 (API アクション): `servicediscovery:ListServices`

ListTagsForResource

必要なアクセス許可 (API アクション): `servicediscovery:ListTagsForResource`

RegisterInstance

必要なアクセス許可 (API アクション):

- `servicediscovery:RegisterInstance`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`
- `route53:UpdateHealthCheck`
- `route53:ChangeResourceRecordSets`
- `ec2:DescribeInstances`

TagResource

必要なアクセス許可 (API アクション): `servicediscovery:TagResource`

UntagResource

必要なアクセス許可 (API アクション): `servicediscovery:UntagResource`

UpdateHttpNamespace

必要なアクセス許可 (API アクション): `servicediscovery:UpdateHttpNamespace`

UpdateInstanceCustomHealthStatus

必要なアクセス許可 (API アクション): `servicediscovery:UpdateInstanceCustomHealthStatus`

UpdatePrivateDnsNamespace

必要なアクセス許可 (API アクション):

- `servicediscovery:UpdatePrivateDnsNamespace`

- `route53:ChangeResourceRecordSets`

UpdatePublicDnsNamespace

必要なアクセス許可 (API アクション):

- `servicediscovery:UpdatePublicDnsNamespace`
- `route53:ChangeResourceRecordSets`

UpdateService

必要なアクセス許可 (API アクション):

- `servicediscovery:UpdateService`
- `route53:GetHealthCheck`
- `route53:CreateHealthCheck`
- `route53:DeleteHealthCheck`
- `route53:UpdateHealthCheck`
- `route53:ChangeResourceRecordSets`

## AWS Cloud Map ID とアクセスのトラブルシューティング

AWS Cloud Map 以下の情報は、 および の使用時に発生する可能性のある一般的な問題を診断して修正するのに役立ちますIAM。

トピック

- でアクションを実行する権限がありません AWS Cloud Map
- iam を実行する権限がありません。PassRole
- 自分の外部のユーザーに自分の AWS Cloud Map リソース AWS アカウント 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ありません AWS Cloud Map

「I am not authorized to perform an action in Amazon Bedrock」というエラーが表示された場合、そのアクションを実行できるようにポリシーを更新する必要があります。

次のエラー例は、`mateojacksonIAM`ユーザーがコンソールを使用して架空の*`my-example-widget`*リソースの詳細を表示しようとしたときに、架空の`servicediscovery:`*`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:
 servicediscovery:GetWidget on resource: my-example-widget
```

この場合、servicediscovery:*GetWidget* アクションを使用して *my-example-widget*リソースへのアクセスを許可するように、mateojackson ユーザーのポリシーを更新する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

## iam を実行する権限がありません。PassRole

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

一部の AWS のサービス では、新しいサービスロール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、 という名前のIAMユーザーがコンソールを使用して marymajor でアクションを実行しようとすると発生します AWS Cloud Map。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。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 のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン認証情報を提供した担当者が管理者です。

## 自分の外部のユーザーに自分の AWS Cloud Map リソース AWS アカウント 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 (ACLs) をサポートするサービスでは、これらのポリシーを使用して、リソースへのアクセスをユーザーに許可できます。

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 Cloud Map をサポートしているかどうかについては、「」を参照してください[と AWS Cloud Map の仕組み IAM](#)。
- 所有 AWS アカウント している 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 AWS アカウント している別の 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。
- サードパーティー にリソースへのアクセスを提供する方法については AWS アカウント、 IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する 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「外部認証されたユーザーへのアクセスを提供する (ID フェデレーション )」を参照してください。
- クロスアカウントアクセスにロールとリソースベースのポリシーを使用する違いについては、IAM ユーザーガイドの「 のクロスアカウントリソースアクセスIAM」を参照してください。

## のコンプライアンス検証 AWS Cloud Map

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

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

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

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

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

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

## の耐障害性 AWS Cloud Map

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、物理的に分離および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 Cloud Map は主にグローバルサービスです。ただし、 AWS Cloud Map を使用して、Amazon EC2インスタンスや Elastic Load Balancing ロードバランサーなど、特定のリージョンのリソースのヘルスをチェックする Route 53 ヘルスチェックを作成できます。

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。

## のインフラストラクチャセキュリティ AWS Cloud Map

マネージドサービスである AWS Cloud Map は、 AWS グローバルネットワークセキュリティによって保護されています。 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呼び出しを使用して、ネットワーク AWS Cloud Map 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 ) 。1TLS.2 が必要で、1.3 TLS をお勧めします。
- (Ephemeral Diffie-HellmanPFS) や DHE (Elliptic Curve Ephemeral Diffie-Hellman) などの完全前方秘匿性 ECDHE () を備えた暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

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

インターフェイスVPCエンドポイントを使用する AWS Cloud Map ように を設定VPCすることで、のセキュリティ体制を改善できます。詳細については、「インターフェイスエンドポイント (AWS PrivateLink) AWS Cloud Map を使用した へのアクセス」を参照してください。

## インターフェイスエンドポイント (AWS PrivateLink) AWS Cloud Map を使用した へのアクセス

を使用して AWS PrivateLink 、VPC と の間にプライベート接続を作成できます AWS Cloud Map。インターネットゲートウェイ、NAT デバイス、VPN 接続、または AWS Direct Connect 接続を使用

せずに、VPC 内にある AWS Cloud Map かのように にアクセスできます。VPC内のインスタンスはAWS Cloud MapにアクセスするためにパブリックIPアドレスを必要としません。

このプライベート接続を確立するには、 AWS PrivateLinkを利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、 AWS Cloud Map宛てのトラフィックのエントリポイントとして機能するリクエスタ管理型ネットワークインターフェイスです。

詳細については、『AWS PrivateLink ガイド』の「AWS のサービス でアクセスする」を参照してください。

## に関する考慮事項 AWS Cloud Map

のインターフェイスエンドポイントを設定する前に AWS Cloud Map、「 AWS PrivateLink ガイド」の「考慮事項」を参照してください。

Amazon VPC にインターネットゲートウェイがなく、タスクが`awslogs`ログドライバーを使用してログ情報を CloudWatch ログに送信する場合は、ログのインターフェイス VPC CloudWatch エンドポイントを作成する必要があります。詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「インターフェイス VPC エンドポイントでの CloudWatch ログの使用」を参照してください。

VPC エンドポイントは AWS クロスリージョンリクエストをサポートしていません。 AWS Cloud Mapに対して API コールを発行するリージョンと同じリージョンにエンドポイントを作成してください。

VPC エンドポイントでは、Amazon Route 53 を介して Amazon 提供の DNS のみがサポートされています。独自の DNS を使用したい場合は、条件付き DNS 転送を使用できます。詳細については、Amazon VPC ユーザーガイド の「DHCP オプション設定」を参照してください。

VPC エンドポイントにアタッチされたセキュリティグループは、Amazon VPC のプライベートサブネットからのポート443での着信接続を許可する必要があります。

## のインターフェイスエンドポイントを作成する AWS Cloud Map

Amazon VPC コンソールまたは AWS Command Line Interface () AWS Cloud Map を使用して、 用のインターフェイスエンドポイントを作成できますAWS CLI。詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントを作成」を参照してください。

次のサービス名 AWS Cloud Map を使用して、 用のインターフェイスエンドポイントを作成します。

> **Note**
>
> `DiscoverInstances` API は、これら 2 つのエンドポイントでは使用できません。

```
com.amazonaws.region.servicediscovery
```

```
com.amazonaws.region.servicediscovery-fips
```

次のサービス名を使用して AWS Cloud Map 、データプレーンが `DiscoverInstances` API にアクセスするためのインターフェイスエンドポイントを作成します。

```
com.amazonaws.region.data-servicediscovery
```

```
com.amazonaws.region.data-servicediscovery-fips
```

> **Note**
>
> データプレーンエンドポイントのリージョンまたはゾーンごとの VPCE DNS 名で呼び出す場合は、`DiscoverInstances` ホストプレフィックスインジェクションを無効にする必要があります。 AWS CLI および AWS SDKs、各 API オペレーションを呼び出すときにサービスエンドポイントにさまざまなホストプレフィックスを付加します。これにより、VPC エンドポイントを指定すると無効な URL が生成されます。

インターフェース· エンドポイントのプライベートDNSを有効にすると、デフォルトの地域DNS名を使用して AWS Cloud Map へのAPI要求を行うことができます。例えば `servicediscovery.us-east-1.amazonaws.com` です。

VPCE AWS PrivateLink 接続は、 が AWS Cloud Map サポートされているすべてのリージョンでサポートされていますが、エンドポイントを定義する前に、VPCE をサポートするアベイラビリティーゾーンを確認する必要があります。リージョン内のインターフェイス VPC エンドポイントでサポートされているアベイラビリティーゾーンを確認するには、 describe-vpc-endpoint-services コマンドを使用するか、 を使用します AWS Management Console。たとえば、次のコマンドは、米国東部 (オハイオ) リージョン内の AWS Cloud Map インターフェイス VPC エンドポイントを配置できるアベイラビリティ ゾーンを返します：

```
aws --region us-east-2 ec2 describe-vpc-endpoint-services --query 'ServiceDetails[?
ServiceName==`com.amazonaws.us-east-2.servicediscovery`].AvailabilityZones[]'
```

# モニタリング AWS Cloud Map

モニタリングは、 AWS ソリューションの信頼性、可用性、および性能を維持するうえで重要な部分です。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、モニタリングを開始する前に、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。

- どのような目的でモニタリングしますか?
- 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?

トピック

- を使用した通話のログ AWS Cloud Map API記録 AWS CloudTrail

## を使用した通話のログ AWS Cloud Map API記録 AWS CloudTrail

AWS Cloud Map は、ユーザーAWS CloudTrail、ロール、または 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 のサービス。 は、 のすべてのAPI呼び出しをイベント AWS Cloud Map として CloudTrail キャプチャします。キャプチャされた呼び出しには、AWS Cloud Map コンソールからの呼び出しと、 オペレーションへのコード呼び出しが含まれます AWS Cloud Map API。によって収集された情報を使用して CloudTrail、 に対して行われたリクエスト AWS Cloud Map、リクエスト元の IP アドレス、リクエスト日時などの詳細を確認できます。

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、以下を判別するのに役立ちます:

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- 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 のサービスによって送信されたかどうか。

CloudTrail アカウント AWS アカウント を作成すると、 は でアクティブになり、 CloudTrail イベント履歴 に自動的にアクセスできます。 CloudTrail イベント履歴は、 に記録された過去 90 日間の管理イベントの表示可能、検索可能、ダウンロード可能、およびイミュータブルなレコードを提供します AWS リージョン。詳細については、「 ユーザーガイド」の CloudTrail 「イベント履歴の使用AWS CloudTrail 」を参照してください。イベント履歴を表示するための料金はかかりません CloudTrail。

AWS アカウント 過去 90 日間のイベントの継続的な記録については、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

証跡により CloudTrail 、 はログファイルを Amazon S3 バケットに配信できます。を使用して作成された証跡はすべてマルチリージョン AWS Management Console です。 AWS CLIを使用する際は、単一リージョンまたは複数リージョンの証跡を作成できます。 AWS リージョン アカウントのすべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成 CloudTrail することで、 から Amazon S3 バケットに継続的な管理イベントのコピーを 1 つ無料で配信できますが、Amazon S3 ストレージ料金が発生します。 CloudTrail 料金の詳細については、「 の料金AWS CloudTrail」を参照してください。Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

CloudTrail Lake では、イベントに対して SQLベースのクエリを実行できます。 CloudTrail Lake は、既存のイベントを行ベースのJSON形式で Apache ORC 形式に変換します。ORC は、データを迅速に取得できるように最適化された列指向ストレージ形式です。イベントはイベントデータストアに集約されます。イベントデータストアは、高度なイベントセレクタを適用することによって選択する条件に基いた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。 CloudTrail Lake の詳細については、「 ユーザーガイド」の AWS CloudTrail 「Lake の使用AWS CloudTrail 」を参照してください。

CloudTrail Lake イベントデータストアとクエリにはコストがかかります。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプショ

ンによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。 CloudTrail 料金の詳細については、「 の料金AWS CloudTrail」を参照してください。

## AWS Cloud Map のデータイベント CloudTrail

データイベントは、リソースで、またはリソースで実行されるリソースオペレーションに関する情報を提供します (例えば、名前空間内の登録済みインスタンスの検出 ) 。これらのイベントは、データプレーンオペレーションとも呼ばれます。データイベントは、多くの場合、高ボリュームのアクティビティです。デフォルトでは、 CloudTrail はデータイベントを記録しません。 CloudTrail イベント履歴にはデータイベントは記録されません。

追加の変更がイベントデータに適用されます。 CloudTrail 料金の詳細については、「 の料金AWS CloudTrail」を参照してください。

CloudTrail コンソール、、または CloudTrail API オペレーションを使用して AWS CLI、 AWS Cloud Map リソースタイプのデータイベントをログに記録できます。データイベントをログに記録する方法の詳細については、「 AWS CloudTrail ユーザーガイド」の「 AWS Management Consoleを使用したデータイベントのログ記録」および「AWS Command Line Interfaceを使用したデータイベントのログ記録」を参照してください。

次の表に、データイベントをログ記録できる AWS Cloud Map リソースタイプを示します。データイベントタイプ (コンソール ) 列には、 CloudTrail コンソールのデータイベントタイプリストから選択する値が表示されます。resources.type 値列には、 AWS CLI または を使用して高度なイベントセレクタを設定するときに指定する `resources.type`値が表示されます CloudTrail APIs。にAPIs記録されたデータ CloudTrail列には、リソースタイプの CloudTrail に記録されたAPI呼び出しが表示されます。

| データイベントタイプ (コンソール) | resources.type 値 | にAPIs記録されたデータ CloudTrail |
| --- | --- | --- |
| AwsApiCall | `AWS::ServiceDiscovery::Namespace` | • DiscoverInstances<br>• DiscoverInstancesRevision |
| AwsApiCall | `AWS::ServiceDiscovery::Service` | • DiscoverInstances<br>• DiscoverInstancesRevision |

eventName、readOnly、および resources.ARN フィールドでフィルタリングして、自分にとって重要なイベントのみをログに記録するように高度なイベントセレクタを設定できます。これらのフィールドの詳細については、「 AWS CloudTrail APIリファレンスAdvancedFieldSelector」の「」を参照してください。

次の例は、すべての AWS Cloud Map データイベントをログに記録するように高度なイベントセレクタを設定する方法を示しています。

```
"AdvancedEventSelectors":
[
    {
        "Name": "Log all AWS Cloud Map data events",
        "FieldSelectors": [
            { "Field": "eventCategory", "Equals": ["Data"] },
            { "Field": "resources.type", "Equals":
 ["AWS::ServiceDiscovery::Namespace"] }
        ]
    }
]
```

## AWS Cloud Map の管理イベント CloudTrail

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します AWS アカウント。これらのイベントは、コントロールプレーンオペレーションとも呼ばれます。デフォルトでは、 は管理イベントを CloudTrail ログに記録します。

AWS Cloud Map は、すべての AWS Cloud Map コントロールプレーンオペレーションを管理イベントとしてログに記録します。に AWS Cloud Map ログ記録される AWS Cloud Map コントロールプレーンオペレーションのリストについては CloudTrail、「 AWS Cloud Map APIリファレンス」を参照してください。

## AWS Cloud Map イベントの例

イベントは任意のソースからの単一のリクエストを表し、リクエストされたAPIオペレーション、オペレーションの日時、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、イベントは特定の順序で表示されません。

次の例は、 CreateHTTPNamespaceオペレーションを示す CloudTrail 管理イベントを示しています。

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
        "arn": "arn:aws:sts::111122223333:assumed-role/users/alejandro_rosalez",
        "accountId": "111122223333",
        "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA123456789EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/readonly-role",
                "accountId": "111122223333",
                "userName": "alejandro_rosalez"
            },
            "attributes": {
                "creationDate": "2024-03-19T16:15:37Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2024-03-19T19:23:13Z",
    "eventSource": "servicediscovery.amazonaws.com",
    "eventName": "CreateHttpNamespace",
    "awsRegion": "eu-west-3",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36",
    "requestParameters": {
        "name": "example-namespace",
        "creatorRequestId": "eda8b524-ca14-4f68-a176-dc4dfd165c26",
        "tags": []
    },
    "responseElements": {
        "operationId": "7xm4i7ghhkaalma666nrg6itf2eylcbp-gwipo38o"
    },
    "requestID": "641274d0-dbbe-4e64-9b53-685769a086c7",
    "eventID": "4a1ab076-ef1b-4bcf-aa95-cec5fb64f2bd",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "servicediscovery.eu-west-3.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

次の例は、 DiscoverInstancesオペレーションを示す CloudTrail データイベントを示しています。

```
{
            "eventVersion": "1.09",
            "userIdentity": {
                "type": "AssumedRole",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE:alejandro_rosalez",
                "arn": "arn:aws:sts::111122223333:assumed-role/role/Admin",
                "accountId": "111122223333",
                "accessKeyId": "AIDACKCEVSQ6C2EXAMPLE",
                "sessionContext": {
                    "sessionIssuer": {
                        "type": "Role",
                        "principalId": "AROA123456789EXAMPLE",
                        "arn": "arn:aws:iam::"111122223333":role/Admin",
                        "accountId": "111122223333",
                        "userName": "Admin"
                    },
                    "attributes": {
                        "creationDate": "2024-03-19T16:15:37Z",
                        "mfaAuthenticated": "false"
                    }
                }
            },
            "eventTime": "2024-03-19T21:19:12Z",
            "eventSource": "servicediscovery.amazonaws.com",
            "eventName": "DiscoverInstances",
            "awsRegion": "eu-west-3",
            "sourceIPAddress": "13.38.34.79",
            "userAgent": "Boto3/1.20.34 md/Botocore#1.34.60 ua/2.0 os/linux#6.5.0-1014-
aws md/arch#x86_64 lang/python#3.10.12 md/pyimpl#CPython cfg/retry-mode#legacy
 Botocore/1.34.60",
            "requestParameters": {
```

```
                "namespaceName": "example-namespace",
                "serviceName": "example-service",
                "queryParameters": {"example-key": "example-value"}
            },
            "responseElements": null,
            "requestID": "e5ee36f1-edb0-4814-a4ba-2e8c97621c79",
            "eventID": "503cedb6-9906-4ee5-83e0-a64dde27bab0",
            "readOnly": true,
            "resources": [
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Namespace",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:namespace/
ns-vh4nbmhEXAMPLE"
                },
                {
                    "accountId": "111122223333",
                    "type": "AWS::ServiceDiscovery::Service",
                    "ARN": "arn:aws:servicediscovery:eu-west-3:111122223333:service/
srv-h46op6ylEXAMPLE"
                }
            ],
            "eventType": "AwsApiCall",
            "managementEvent": false,
            "recipientAccountId": "111122223333",
            "eventCategory": "Data",
            "tlsDetails": {
                "tlsVersion": "TLSv1.3",
                "cipherSuite": "TLS_AES_128_GCM_SHA256",
                "clientProvidedHostHeader": "data-servicediscovery.eu-
west-3.amazonaws.com"
            },
            "sessionCredentialFromConsole": "true"
        }
```


CloudTrail レコードの内容の詳細については、「 ユーザーガイド」のCloudTrail「レコードの内容AWS CloudTrail 」を参照してください。

# AWS Cloud Map リソースのタグ付け

タグは、 AWS リソースに割り当てるラベルです。タグはそれぞれ、1 つのキーとオプションの 1 つの値で構成されており、どちらもお客様側が定義します。

タグを使用すると、目的、所有者、環境などで AWS リソースを分類できます。同じ型のリソースが多い場合に、割り当てたタグに基づいて特定のリソースをすばやく識別できます。例えば、 AWS Cloud Map サービスのタグのセットを定義して、各サービスの所有者とスタックレベルを追跡できます。リソースタイプごとに一貫した一連のタグキーを考案することをお勧めします。

タグは自動的にはリソースに割り当てられません。タグを追加したら、いつでもタグキーと値は編集でき、タグはリソースからいつでも削除できます。リソースを削除すると、リソースのタグも削除されます。

タグには意味論的な意味はなく AWS Cloud Map 、文字列として厳密に解釈されます。タグの値を空の文字列に設定することはできますが、タグの値を null に設定することはできません。特定のリソースについて既存のタグと同じキーを持つタグを追加した場合、以前の値は新しい値によって上書きされます。

タグは、 AWS Management Console、、 AWS CLIおよび AWS Cloud Map API を使用して操作できます。

AWS Identity and Access Management (IAM) を使用している場合は、 AWS アカウント内のどのユーザーがタグを作成、編集、または削除するためのアクセス許可を持っているかを制御できます。

## リソースのタグ付け方法

新規または既存の AWS Cloud Map 名前空間とサービスにタグを付けることができます。

AWS Cloud Map コンソールを使用している場合は、新しいリソースの作成時にタグを適用したり、関連するリソースページのタグタブを使用して既存のリソースにタグを適用したりできます。

AWS Cloud Map API、、 AWS CLIまたは AWS SDK を使用している場合は、関連する API アクションの `tags`パラメータを使用して新しいリソースに、または API TagResourceアクションを使用して既存のリソースにタグを適用できます。詳細については、「」を参照してくださいTagResource。

リソース作成アクションによっては、リソースの作成時にリソースのタグを指定できます。リソースの作成時にタグを適用できない場合、リソースの作成プロセスは失敗します。これにより、作成時にタグ付けするリソースが、指定したタグで作成されるか、まったく作成されないことが確認されま

す。作成時にリソースにタグを付ける場合、リソースの作成後にカスタムのタグ付けスクリプトを実行する必要はありません。

次の表は、タグ付けできる AWS Cloud Map リソースと、作成時にタグ付けできるリソースを示しています。

AWS Cloud Map リソースのタグ付けのサポート

| リソース | タグをサポート | タグの伝播をサポート | 作成時のタグ付けをサポート (AWS Cloud Map API、 AWS CLI、 AWS SDK) |
|---|---|---|---|
| AWS Cloud Map 名前空間 | あり | いいえ。名前空間タグは、名前空間に関連付けられた他のリソースには伝達されません。 | あり |
| AWS Cloud Map サービス | あり | いいえ。サービスタグは、サービスに関連付けられた他のリソースには伝達されません。 | あり |

# 制限事項

タグには以下のベーシックな制限があります。

- それぞれのリソースに付けることができるタグの最大数は 50 です。
- タグキーは、リソースごとにそれぞれ一意である必要があります。また、各タグキーに設定できる値は 1 つのみです。
- キーの最大長 - UTF-8 の 128 Unicode 文字
- 値の最大長 - UTF-8 の 256 Unicode 文字
- タグ付けスキーマが複数の AWS サービスやリソースで使用されている場合は、他のサービスで許可される文字に制限がある可能性があることに注意してください。一般的に使用が許可される文字は、UTF-8 で表現できる文字、数字、スペース、および +、-、=、.、_、:、/、@。

- タグのキーと値では、大文字と小文字が区別されます。
- キーまたは値のプレフィックスなどaws:AWS:、、、または大文字または小文字の組み合わせは使用しないでください。使用のために予約されています AWS 。このプレフィックスを持つタグのキーや値を編集または削除することはできません。このプレフィックスを持つタグは tags-per-resource 、制限にはカウントされません。

# AWS Cloud Map リソースのタグの更新

リソースのタグを追加、更新、一覧表示、削除するには、次の AWS CLI コマンドまたは AWS Cloud Map API オペレーションを使用します。

AWS Cloud Map リソースのタグ付けのサポート

| タスク | API アクション | AWS CLI | AWS Tools for Windows PowerShell |
|---|---|---|---|
| 1 つ以上のタグを追加、または上書きします。 | TagResource | タグリソース | SD を追加ResourceTag |
| 1 つ以上のタグを削除します。 | UntagResource | タグなしリソース | SD の削除ResourceTag |
| リソースのタグの一覧表示 | ListTagsForResource | list-tags-for-resource | SD を取得するResourceTag |

以下の例では、 AWS CLIを使用して、リソースに対してタグ付けまたはタグ削除する方法を示しています。

例 1: 既存のリソースにタグ付けする

次のコマンドでは、既存のリソースにタグ付けします。

```
aws servicediscovery tag-resource --resource-arn resource_ARN --tags team=devs
```

例 2: 既存のリソースからタグを削除する

次のコマンドでは、既存のリソースからタグを削除します。

```
aws servicediscovery untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

例 3: リソースのタグのリスト取得

次のコマンドは、既存のリソースに関連付けられているタグのリストを取得します。

```
aws servicediscovery list-tags-for-resource --resource-arn resource_ARN
```

一部のリソース作成アクションでは、リソースの作成時にタグを指定できます。以下のアクションでは、作成時のタグ付けがサポートされます。

| タスク | API アクション | AWS CLI | AWS Tools for Windows PowerShell |
| --- | --- | --- | --- |
| HTTP 名前空間を作成する | CreateHttpNamespace | create-http-namespace | 新規 SDHttpNamespace |
| DNS に基づくプライベート名前空間を作成する | CreatePrivateDnsNamespace | create-private-dns-namespace | 新規 SDPrivateDnsNamespace |
| DNS に基づくパブリック名前空間を作成する | CreatePublicDnsNamespace | create-public-dns-namespace | 新規 SDPublicDnsNamespace |
| サービスを作成する | CreateService | サービスの作成 | New-SDService |

# AWS Cloud Map サービスクォータ

AWS Cloud Map リソースには、以下のアカウントレベルのサービスクォータが適用されます。リストされている各クォータは、 AWS Cloud Map リソースを作成する各 AWS リージョンに適用されます。

| 名前 | デフォルト | 引き上げ可能 | 説明 |
| --- | --- | --- | --- |
| インスタンスあたりのカスタム属性 | サポートされている各リージョン: 30 | なし | インスタンス登録時に指定できるカスタム属性の最大数。 |
| DiscoverInstances アカウントあたりのオペレーションのバーストレート | サポートされている各リージョン: 2,000 | はい | 1 つのアカウントから DiscoverInstances オペレーションを呼び出す最大バーストレート。 |
| DiscoverInstances アカウントあたりのオペレーションの定常レート | サポートされている各リージョン: 1,000 | はい | 1 つのアカウントから DiscoverInstances オペレーションを呼び出す最大定常レート。 |
| DiscoverInstancesRevision アカウントあたりの オペレーションレート | サポートされている各リージョン: 3,000 | はい | 1 つのアカウントから DiscoverInstancesRevision オペレーションを呼び出す最大レート。 |
| 名前空間あたりのインスタンス | サポートされている各リージョン: 2,000 | はい | 同じ名前空間を使用して登録できるサービスインスタンスの最大数。 |

| 名前 | デフォルト | 引き上げ可能 | 説明 |
| --- | --- | --- | --- |
| サービスあたりのインスタンス | サポートされている各リージョン: 1,000 | なし | 同じサービスを使用してリージョンに登録できるインスタンスの最大数。 |
| リージョンあたりの名前空間 | サポートされている各リージョン: 50 | はい | リージョンごとに作成できる名前空間の最大数。 |

* 名前空間を作成すると、Amazon Route 53 ホストゾーンが自動的に作成されます。このホストゾーンは、 AWS アカウントで作成できるホストゾーンの数のクォータに対してカウントされます。詳細については、Amazon Route 53 デベロッパーガイドのホストゾーンのクォータを参照してください。

** AWS Cloud Map 用のDNS 名前空間あたりのインスタンスを増やすには、Route 53 ホストゾーンあたりのレコード数の制限を増やす必要があり、これには追加料金が発生します。

# AWS Cloud Map サービスクォータの管理

AWS Cloud Map は、Service Quotas と統合されています。Service Quotas は、クォータを一元的に表示および管理できる AWS サービスです。詳細については、「Service Quotas ユーザーガイド」の「Service Quotas とは」を参照してください。

Service Quotas を使用すると、 AWS Cloud Map Service Quotas の値を簡単に検索できます。

AWS Management Console

を使用して AWS Cloud Map Service Quotas を表示するには AWS Management Console

1. https://console.aws.amazon.com/servicequotas/ で Service Quotas コンソールを開きます。
2. ナビゲーションペインで、[AWS サービス] を選択します。
3. [AWS サービス] リストから、[AWS Cloud Map]]] を探して選択します。

4. のサービスクォータリストには AWS Cloud Map、サービスクォータ名、適用された値 (使用可能な場合 ) 、 AWS デフォルトのクォータ、およびクォータ値が調整可能かどうかが表示されます。

   説明など、サービスクォータに関する追加情報を表示するには、クォータ名を選択してクォータの詳細を表示します。
5. ( オプション) クォータの引き上げをリクエストするには、引き上げるクォータを選択し、アカウントレベルで引き上げをリクエストする を選択します。

を使用して Service Quotas をさらに操作するには、Service Quotas AWS Management Console 」を参照してください。

AWS CLI

を使用して AWS Cloud Map Service Quotas を表示するには AWS CLI

次のコマンドを実行して、デフォルトの AWS Cloud Map クォータを表示します。

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code AWSCloudMap \
    --output table
```

次のコマンドを実行して、適用された AWS Cloud Map クォータを表示します。

```
aws service-quotas list-service-quotas \
    --service-code AWSCloudMap
```

を使用した Service Quotas の操作の詳細については AWS CLI、Service Quotas AWS CLI コマンドリファレンス」を参照してください。クォータの引き上げをリクエストするには、「AWS CLI コマンド リファレンス」で `request-service-quota-increase` コマンドを参照してください。

# AWS Cloud Map DiscoverInstances API リクエストスロットリングの処理

AWS Cloud Map は、リージョンごとに各 AWS アカウントの DiscoverInstances API リクエストを調整します。スロットリングは、サービスのパフォーマンスを向上させ、すべての AWS Cloud

Map お客様に公平な使用を提供するのに役立ちます。スロットリングにより、 AWS Cloud Map DiscoverInstancesAPI への呼び出しが API DiscoverInstancesリクエストの最大許容クォータを超えないようになります。DiscoverInstances 次のいずれかのソースから発信される API コールには、リクエストクォータが適用されます。

- サードパーティーのアプリケーション
- コマンドラインツール
- AWS Cloud Map コンソール

API スロットリングクォータを超えると、`RequestLimitExceeded` エラーコードが返されます。詳細については、「the section called “リクエストレート制限”」を参照してください。

## スロットリングの適用方法

AWS Cloud Map は、トークンバケットアルゴリズムを使用して API スロットリングを実装します。このアルゴリズムでは、アカウントには、特定の数のトークンを保持するバケットがあります。バケット内のトークンの数は、特定の秒におけるスロットリングクォータを表します。単一のリージョンに対して 1 つのバケットがあり、リージョン内のすべてのエンドポイントに適用されます。

### リクエストレート制限

スロットリングは、実行できる DiscoverInstances API リクエストの数を制限します。各リクエストは、バケットから 1 つのトークンを削除します。例えば、DiscoverInstancesAPI オペレーションのバケットサイズは 2,000 トークンであるため、1 秒で最大 2,000 DiscoverInstancesリクエストを行うことができます。1 秒で 2,000 リクエストを超えると、スロットルされ、その秒以内の残りのリクエストは失敗します。

バケットは設定されたレートで自動的に補充されます。バケットが容量に達していない場合、バケットが容量に達するまで、設定された数のトークンが毎秒追加されます。リフィルトークンが到着したときにバケットが容量に達している場合、これらのトークンは破棄されます。DiscoverInstances API オペレーションのバケットサイズは 2,000 トークンで、リフィルレートは 1 秒あたり 1,000 トークンです。1 秒あたり 2,000 件の DiscoverInstances API リクエストを行うと、バケットはすぐにゼロ (0) トークンに減少します。バケットは、最大容量の 2000 トークンに達するまで、毎秒最大 1,000 トークンで補充されます。

トークンはバケットに追加されたときに使用できます。API リクエストを行う前に、バケットが最大容量になるのを待つ必要はありません。1 秒間に 2,000 件の DiscoverInstances API リクエストを実行してバケットを使い果たしても、その後は必要な期間、毎秒最大 1,000 件の DiscoverInstances

API リクエストを行うことができます。つまり、バケットに追加されたリフィルトークンをすぐに使用できます。バケットは、リフィルレートよりも毎秒少ない API リクエスト数を作成する場合にのみ、最大容量への補充を開始します。

### 再試行またはバッチ処理

API リクエストが失敗した場合、アプリケーションでリクエストを再試行する必要がある場合があります。API リクエストの数を下げるには、連続するリクエストの間に適切なスリープ間隔を使用します。最良の結果を得るには、漸増または可変スリープ間隔を使用します。

### スリープ間隔の計算

API リクエストをポーリングまたは再試行する必要がある場合は、エクスポネンシャルバックオフアルゴリズムを使用して API コール間のスリープ間隔を計算することをお勧めします。連続したエラーレスポンスの再試行間隔の待機時間を徐々に長くすることで、失敗リクエストの数を減らすことができます。このアルゴリズムの詳細と実装例については、「 SDK とツールのリファレンスガイド」の「再試行動作」を参照してください。 AWS SDKs

## API スロットリングのクォータの調整

AWS アカウントの API スロットリングクォータの引き上げをリクエストできます。クォータの調整をリクエストするには、AWS Support センターまでお問い合わせください。

# のドキュメント履歴 AWS Cloud Map

次の表は、「AWS Cloud Map デベロッパーガイド」の主な更新や新機能の一覧です。また、お客様からいただいたフィードバックに対応するために、ドキュメントを頻繁に更新しています。

| 変更 | 説明 | 日付 |
| --- | --- | --- |
| チュートリアルの追加 | の使用に関する一般的なユースケースを示す 2 つのチュートリアル AWS Cloud Map が追加されました。 | 2024 年 3 月 27 日 |
| CloudTrail 統合ドキュメントが更新されました | アクティビティをログに記録する CloudTrail ための と AWS Cloud Map の統合について説明するドキュメントが更新されAPIました。 | 2024 年 3 月 20 日 |
| マネージドポリシーの更新 | `AWSCloudMapDiscoverInstanceAccess` 、`AWSCloudMapRegisterInstanceAccess` 、`AWSCloudMapReadOnlyAccess` ポリシーが更新されました。 | 2023 年 9 月 20 日 |
| クラウドマップと AWS PrivateLink | を使用して AWS PrivateLink 、 VPCと の間にプライベート接続を作成できるようになりました AWS Cloud Map。 | 2023 年 9 月 15 日 |
| マネージドポリシーの更新 | `AWSCloudMapDiscoverInstanceAccess` ポリシーが更新されました。 | 2023 年 8 月 15 日 |

| | | |
|---|---|---|
| AWS SDK for Python | Python コマンドラインの例を追加しました。 | 2022 年 9 月 13 日 |
| IPv6 サポート | API エンドポイントが IPv6のみのネットワークで使用できるようになりました。 | 2022 年 1 月 28 日 |
| サービスインスタンスの検出 | AWS Cloud Map は、DiscoverInstancesAPIオペレーションのみを使用して検出でき、DNSクエリを使用しないDNSクエリをサポートする名前空間でサービスを作成するためのサポートを追加しました。 | 2021 年 3 月 24 日 |
| リソースへのタグ付け | AWS Cloud Map は、 を使用して名前空間とサービスにメタデータタグを追加するためのサポートを追加しました AWS Management Console。 | 2021 年 2 月 8 日 |
| リソースへのタグ付け | AWS Cloud Map は、 AWS CLI および を使用して名前空間とサービスにメタデータタグを追加するためのサポートを追加しましたAPIs。 | 2020年6月22日 |
| 初回リリース | これは AWS Cloud Map デベロッパーガイドの初回リリースです。 | 2018 年 11 月 28 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。