開発者ガイド

AWS Cloud Map

AWS Cloud Map: 開発者ガイド

Table of Contents

AWS Cloud Map とは?

AWS Cloud Map は、アプリケーションで使用されるバックエンドサービスやリソースのマップを作成および維持することができる完全マネージド型のサービスです。AWS Cloud Map の仕組みを以下に示します。

1. リソースの検索に使用する名前を識別し、リソースの検索方法を指定する名前空間を作成します。これを行うには、AWS Cloud Map DiscoverInstances API コール、VPC の DNS クエリ、またはパブリック DNS クエリを使用します。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。
2. AWS Cloud Map サービスは、AWS Cloud Map を使用してエンドポイントを検索するためのリソースのタイプごとに作成します。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、アプリケーションで別のリソース (例: 別のウェブサーバー) が作成される際に、AWS Cloud Map で使用されるテンプレートです。名前空間を作成したときに DNS 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種類に関する情報が含まれています。また、サービスは、リソースの状態をチェックするかどうか、またチェックする場合は Amazon Route 53 ヘルスチェックを使用するか、サードパーティー製のヘルスチェッカーを使用するかを示します。
3. アプリケーションでリソースが追加されると、AWS Cloud Map RegisterInstance API アクションが呼び出される場合があります。この場合、サービスインスタンスが作成されます。サービスインスタンスには、アプリケーションでリソースを検索する方法に関する情報が含まれます。この際、DNS、または AWS Cloud Map DiscoverInstances API アクションを使用するかどうかは関係ありません。
4. リソースに接続する必要がある場合、アプリケーションは DiscoverInstances を呼び出し、そのリソースに関連付けられている名前空間およびサービスを指定します。その後、AWS Cloud Map によって、1 つ以上のリソースの検索方法に関する情報が返ります。サービスの作成時にヘルスチェックを指定した場合は、AWS Cloud Map よりヘルスインスタンスのみが返ります。

AWS Cloud Map は、Amazon Elastic Container Service (Amazon ECS) と緊密に統合されています。新しいコンテナのタスクはスピンアップまたはスピンダウンするため、自動的に AWS Cloud Map に登録されます。Amazon Elastic Kubernetes Service を AWS Cloud Map に統合するには、Kubernetes ExternalDNS コネクタを使用します。また、AWS Cloud Map を使用して、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、Amazon API Gateway にデプロイされた API などのクラウドリ

ソースを登録および検索することもできます。サービスインスタンスの属性値を指定することで、クライアントはこのような属性を使用して、AWS Cloud Map より返るリソースをフィルタリングすることができます。たとえば、アプリケーションから、BETA や PROD 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。

トピック

- AWS Cloud Map へのアクセス
- AWS Identity and Access Management
- AWS Cloud Map 料金表
- AWS Cloud Map および AWS クラウドコンプライアンス

AWS Cloud Map へのアクセス

AWS Cloud Map には、以下の方法でアクセスできます。

- AWS Management Console – このガイドでは、AWS Management Console を使用してタスクを実行する方法について説明しています。
- AWS SDK - AWS が SDK 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 AWS Cloud Map にアクセスできます。SDK によって認証が簡素化され、開発環境との統合が容易になり、AWS Cloud Map コマンドにアクセスすることができます。詳細については、Tools for Amazon Web Services を参照してください。
- AWS Command Line Interface – 詳細については、AWS Command Line Interface ユーザーガイドの「AWS Command Line Interface でのセットアップ」を参照してください。
- 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。
- AWS Cloud Map API – SDK が提供されていないプログラミング言語を使用している場合、API アクションの情報と API リクエストの作成方法については、AWS Cloud Map API リファレンスを参照してください。

> **Note**
>
> IPv6 クライアントのサポート — 2023 年 6 月 22 日現在、すべての新しいリージョンで、 IPv6 から AWS Cloud Map のクライアントに送信されるコマンドはすべて新しいデュアルスタックエンドポイント (`servicediscovery.<region>.api.aws`) にルーティングされています。2023 年 6 月 22 日以前にリリースされた以下のリージョンでは、AWS Cloud Map IPv6 専用のネットワークから、レガシー

(`servicediscovery.<region>.amazonaws.com`) とデュアルスタックエンドポイントの両方にアクセスできます。

- 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 AWS Identity and Access Management (IAM) と統合されています。このサービスを組織で利用すると、以下のことを実行できます。

- 組織の AWS アカウントでユーザーとグループを作成する
- アカウント内のユーザー間で効率的な方法で AWS アカウントのリソースを共有する
- 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、AWS Cloud Map で IAM を使用すると、新しい名前空間の作成やインスタンスの登録を行うことができる AWS アカウントのユーザーを制御できます。

IAM の一般的な情報については、以下のリソースを参照してください。

- AWS Identity and Access Management に AWS Cloud Map
- AWS Identity and Access Management
- IAM ユーザーガイド

AWS Cloud Map 料金表

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う API コールによって異なります。AWS Cloud Map には前払いはなく、お支払いいただくのは使用した分のみです。

必要に応じて、IP アドレスを持つリソースに対して DNS ベースの検出を有効にできます。また、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。API コールまたは DNS クエリを使用してインスタンスを検出するかどうかは関係ありません。Route 53 DNS およびヘルスチェックの使用量に応じて、追加料金が発生します。

詳細については、「AWS Cloud Map の料金」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

AWS Cloud Map のさまざまなセキュリティコンプライアンス規制への準拠と監査標準の詳細については、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
- コンプライアンスプログラムによる AWS 対象範囲内のサービス

AWS Cloud Mapのセットアップ

このセクションの概要と手順は、 AWSを使い始める際の参考になります。

トピック

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

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/signup を開きます。
2. オンラインの手順に従います。

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

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

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

管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「ルートユーザーとしてサインインする」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

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

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

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

管理アクセス権を持つユーザーとしてサインインする

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

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

追加のユーザーにアクセス権を割り当てる

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

   手順については、「AWS IAM Identity Center ユーザーガイド」の「権限設定を作成する」を参照してください。
2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

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

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

API、、 AWS CLI AWS Tools for Windows PowerShellまたは AWS SDKsを使用するには、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 外部で を操作する場合は、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 (IAM Identity Center で管理されているユーザー)	一時的な認証情報を使用して、、 AWS SDKs AWS CLI、または 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。 • については AWS CLI、「 ユーザーガイド」の AWS CLI 「 を使用するための の設定 AWS IAM Identity CenterAWS Command Line

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
		Interface 」を参照してください。 • AWS SDKs 、ツール、AWS APIs「 SDK とツールのリファレンスガイド」の「IAM Identity Center 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 SDKs AWS CLI、または AWS APIs。	「IAM ユーザーガイド」の「 AWS リソースでの一時的な認証情報の使用」の手順に従います。
IAM	(非推奨) 長期認証情報を使用して、、AWS SDKs AWS CLI、または 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。 • 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。 • AWS SDKs「 SDK とツールのリファレンスガイド」の「長期的な認証情報を使用した認証」を参照してください。 AWS SDKs • 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。

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 の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、「 ユーザーガイド」の AWS Command Line Interface「 のセットアップ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。

AWS SDK をダウンロードする

SDK 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、 AWS Cloud Map API の代わりに SDK を使用することをお勧めします。SDK の使用には、いくつかの利点があります。SDK によって認証が簡素化され、開発環境との統合が容易になり、 AWS Cloud Map コマンドにアクセスすることができます。詳細については、Tools for Amazon Web Services を参照してください。

AWS Cloud Map を使用する

AWS Cloud Map は、論理名をアプリケーションのリソースにマッピングするために使用できる管理ソリューションです。また、AWS SDK、RESTful API 呼び出し、または DNS クエリのいずれかを使用してアプリケーションがリソースを検出するのにも役立ちます。AWS Cloud Map は、Amazon DynamoDB (DynamoDB) テーブル、Amazon Simple Queue Service (Amazon SQS) キュー、または 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高レベルのアプリケーションサービスなどの正常なリソースのみを提供します。

トピック

AWS Cloud Map の使用方法に関する概要

AWS Cloud Map の使用方法に関する概要は次のとおりです:

1. 名前空間 (サービスの論理グループ) を作成します。名前空間を作成する際、アプリケーションでインスタンスの検出に使用する名前を指定します。また、AWS Cloud Map に登録するサービスインスタンスを検出する方法を指定します。この際、API コールまたは DNS クエリを使用します。

   詳細については、次のトピックを参照してください。

   - AWS Cloud Map 名前空間の作成
   - AWS Cloud Map API リファレンスの CreatePublicDnsNamespace、CreatePrivateDnsNamespace、および CreateHttpNamespace

   パブリックまたはプライベートの DNS 名前空間を作成すると、名前空間と同じ名前を持つ Amazon Route 53 のパブリックまたはプライベートのホストゾーンが AWS Cloud Map によって自動的に作成されます。パブリックまたはプライベートの DNS 名前空間を使用する場合でも、AWS Cloud Map の DiscoverInstances リクエストを使用してインスタンスを検出することができます。

AWS Cloud MapAPIリクエストを送信できるエンドポイントのリストについては、Amazon Web Services 全般のリファレンスの「AWSリージョンとエンドポイント」の章のAWS Cloud Mapを参照のこと。

2. パブリック DNS 名前空間を作成した場合は、次のステップを実行して、ドメイン登録用のネームサーバーを、名前空間作成時に AWS Cloud Map によって作成された Route 53 ホストゾーンのネームサーバーに変更します。

   a. パブリック DNS 名前空間と同じ名前のドメインを登録済みの場合は、ステップ 2b に進みます。

      この名前空間と同じ名前のドメインをまだ登録していない場合は、ドメインを登録します。ドメイン登録に Route 53 を使用する場合は、、Amazon Route 53 デベロッパーガイド の「新しいドメインを登録する」を参照してください。ステップ 3 に進みます。

   b. 名前空間を作成し、名前空間 ID を取得した際に返った `OperationId` を使用します。詳細については、「GetOperation」を参照してください。

      > **Note**
      > プログラムによる方法を使用してこれらのステップを実行する場合は、プロセスの後半でもこの名前空間 ID を使用してサービスを作成します。

   c. ステップ 2b で取得した名前空間 ID を使用して、AWS Cloud Map によって作成された Route 53 ホストゾーンの ID を取得します。詳細については、AWS Cloud Map API リファレンスの「GetNamespace」を参照してください。

   d. ステップ 2c で取得したホストゾーン ID を使用して、Route 53 でホストゾーンに割り当てられたネームサーバーの名前を取得します。詳細については、「パブリックホストゾーンに対するネームサーバーの取得」を参照してください。

   e. ドメインに割り当てられているネームサーバーを変更します。ドメインが Route 53 に登録されている場合は、詳細について「ドメインのネームサーバーおよびグル–レコードを追加または変更する」を参照してください。

3. サービスを作成します。このサービスには、アプリケーションのリソースへのアクセス方法を識別するサービスインスタンス (例: ウェブサーバー、DynamoDB テーブル、Amazon S3 バケット) が含まれます。

   ステップ 1 でパブリックまたはプライベートの DNS 名前空間を作成した場合、サービスで指定する名前は、ステップ 1 で AWS Cloud Map によって自動的に作成された Route 53 のパブリッ

クまたはプライベートのホストゾーンのレコード名の一部になります。次のステップでインスタンスを登録すると、AWS Cloud Map によってレコードがホストゾーンに作成されます。レコード名は、サービスの名前 (例: `backend`) と名前空間の名前 (例: `example.com`) を組み合わせたもの `backend.example.com` になります。

サービスを作成する際、サービスインスタンスで指定するリソースの状態を確認するかどうかを選択することもできます。

- ヘルスチェックを設定しない場合は、対応するリソースの状態にかかわらず、AWS Cloud Map または Route 53 によりサービスインスタンスが返されます。
- Route 53 ヘルスチェック (パブリック DNS 名前空間の場合のみ) を選択した場合は、AWS Cloud Map によって自動的に Route 53 ヘルスチェックが作成され、対応する Route 53 レコードと関連付けられます。Route 53 は、正常なリソースのレコードのみを使用して DNS クエリに応答します。
- カスタムのヘルスチェックを選択する場合は、サードパーティーアプリケーションでリソースの状態を判断します。サードパーティーのヘルスチェックの結果に基づき、UpdateInstanceCustomHealthStatus リクエストを AWS Cloud Map に送信して、サービスインスタンスのステータスを更新します。

ヘルスチェックを構成する場合、AWS Cloud Map または Route53 は、DiscoverInstances リクエストまたは DNS クエリに応答して、正常なリソースのサービスインスタンスのみを返します。

詳細については、次のトピックを参照してください。

- AWS Cloud Map サービスの作成
- AWS Cloud Map API リファレンスの CreateService

4. 1 つ以上のサービスインスタンスを登録します。各サービスインスタンスには、アプリケーションの 1 つのリソースにアクセスする方法に関する情報が含まれます。

   詳細については、次のトピックを参照してください。

   - AWS Cloud Map サービスインスタンスの登録
   - AWS Cloud Map API リファレンスの RegisterInstance

5. アプリケーションを記述し、AWS Cloud Map の DiscoverInstances API アクション、または DNS クエリを使用してインスタンスを検出します。

- アプリケーションで DiscoverInstances を使用している場合は、AWS Cloud Map によって、指定された条件に合う利用可能なインスタンスに関する情報が返されます。
- アプリケーションで DNS クエリを使用している場合は、1 つ以上のレコードが Route 53 により返されます。

サービスの作成時にヘルスチェックの設定を指定した場合は、AWS Cloud Map または Route 53 により、ヘルスインスタンスの値のみが返されます。

6. リソースを使用して停止する場合は、対応するサービスインスタンスを登録解除します。関連する Route 53 レコードおよびヘルスチェックが AWS Cloud Map によって削除されます (存在する場合)。

   詳細については、次のトピックを参照してください。

   - AWS Cloud Map サービスインスタンスの登録解除
   - AWS Cloud MapAPI リファレンスの DeregisterInstance

7. サービスおよび名前空間が不要になった場合は、削除することができます。次の点に注意してください。

   - 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インスタンスはすべて、登録を解除する必要があります。
   - 名前空間を削除する前に、名前空間に作成したサービスはすべて削除する必要があります。

   詳細については、次のトピックを参照してください。

   - AWS Cloud Map サービスの削除
   - AWS Cloud Map 名前空間の削除
   - AWS Cloud Map API リファレンスの DeleteService
   - AWS Cloud MapAPI リファレンスの DeleteNamespace

の設定 AWS Cloud Map

以下のセクションでは、 AWS Cloud Map コンソールと を使用して名前空間とサービス AWS CLI を作成、表示、削除し、インスタンスの登録と登録解除を行う方法について説明します。

本番環境では、通常、ほとんどの AWS Cloud Map アクションをプログラムで実行します。へのプログラムによるアクセスの詳細については AWS Cloud Map、以下のドキュメントとダウンロードページを参照してください。

- AWS Cloud Mapのセットアップ
- Amazon Web Services のツールは、SDK、コマンドラインツールなどのデベロッパーリソースを示したものです。
- AWS Cloud Map API リファレンスは、 AWS が SDK を提供していないプログラミング言語を使用している場合の AWS Cloud Map API の使用に関する情報を提供します。

トピック

- AWS Cloud Map 名前空間の使用
- AWS Cloud Map サービスの使用
- AWS Cloud Map サービスインスタンスの使用
- AWS Cloud MapAWS Cloud Map コンソールで使用できない機能

AWS Cloud Map 名前空間の使用

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方法です。名前空間を作成するときは、API コールまたは DNS クエリ AWS Cloud Mapを使用して、 に登録するサービスインスタンスを検出する方法を指定します。また、アプリケーションでインスタンスの検出に使用する名前を指定します。

トピック

- AWS Cloud Map 名前空間の作成
- AWS Cloud Map 名前空間の表示
- AWS Cloud Map 名前空間の削除

AWS Cloud Map 名前空間の作成

名前空間を作成するには、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. [名前空間の作成] を選択します。
3. [名前空間の作成] ページに、適切な値を入力します。詳細については、「名前空間の作成時に指定する値」を参照してください。
4. [名前空間の作成] を選択します。

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
    ```

> **Note**
>
> 名前空間の要件:
>
> - 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 (.com など) で終わる必要があります。
> - 名前空間名は最大 1,024 文字で、先頭と末尾は文字でなければなりません。
> - 有効な文字は、A ~ Z、a ~ z、0 ~ 9、-(ハイフン)、_(アンダースコア)、.(ピリオド) です。

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

> **Note**
>
> 名前空間の要件:
>
> - 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 (.com など) で終わる必要があります。
> - 名前空間名は最大 1,024 文字で、先頭と末尾は文字でなければなりません。
> - 有効な文字は、A ~ Z、a ~ z、0 ~ 9、-(ハイフン)、_(アンダースコア)、.(ピリオド) です。

名前空間の作成時に指定する値

AWS Cloud Map 名前空間を作成するときは、次の値を指定します。

> **Note**
> 名前空間を作成すると、タグを変更できます。ただし、他の値を変更することはできません。

[値]

- Namespace name
- Namespace description
- Instance discovery
- Tags
- VPC

名前空間名

名前空間に指定する名前は、アプリケーションがインスタンスを検出する方法によって異なります。インスタンスの検出方法は、インスタンスの検出で選択したオプションによって決まります。オプションは、コンソールの現在のページの後半に表示されます。その内容は次のとおりです。

API コール

このオプションを選択した場合は、DiscoverInstances リクエストで名前空間名およびサービス名を指定することによって、アプリケーションでサービスインスタンスが検出されます。詳細については、AWS Cloud Map API リファレンスの DiscoverInstances を参照してください。

最大で 1,024 文字までの名前を指定できます。名前には、大文字および小文字の文字、数字、アンダースコア (_) 、またはハイフン (-) を含めることができます。

API コールと VPC の DNS クエリ

VPC 内のアプリケーションが DNS クエリを送信してインスタンスを検出する際に使用するドメイン名を入力します。 は、この名前の Amazon Route 53 プライベートホストゾーン AWS

Cloud Map を自動的に作成します。サービスインスタンスを登録すると、 AWS Cloud Map によって、次の形式の名前を持つ DNS レコードがホストゾーンに作成されます。

service-name.namespace-name

このオプションを選択した場合は、DiscoverInstances リクエストで名前空間名およびサービス名を指定することによって、アプリケーションでサービスインスタンスを検討することもできます。詳細については、AWS Cloud Map API リファレンスの DiscoverInstances を参照してください。

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。

API コールとパブリック DNS クエリ

パブリック DNS クエリを送信してインスタンスを検出する際にアプリケーションで使用するドメイン名を入力します。登録したドメイン名を使用します。名前空間を作成すると、 は同じ名前の Amazon Route 53 パブリックホストゾーン AWS Cloud Map を自動的に作成します。サービスインスタンスを登録すると、 AWS Cloud Map によって、次の形式の名前を持つ DNS レコードがホストゾーンに作成されます。

service-name.namespace-name

このオプションを選択した場合は、DiscoverInstances リクエストで名前空間名およびサービス名を指定することによって、アプリケーションでサービスインスタンスを検討することもできます。詳細については、AWS Cloud Map API リファレンスの DiscoverInstances を参照してください。

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。

名前空間の説明

名前空間の説明を入力します。ここに入力した値は、[名前空間] ページと、各名前空間の詳細ページに表示されます。

インスタンス検出

登録されたインスタンスをアプリケーションで検出する方法を選択します。

API コール

アプリケーションで API コールのみを使用して、登録されたインスタンスを検出する場合は、このオプションを選択します。

API コールと VPC の DNS クエリ

アプリケーションで API コール、または VPC の DNS クエリを使用して、インスタンスを検出できるようにするには、このオプションを選択します。両方の方法を使用する必要はありません。

API コールとパブリック DNS クエリ

アプリケーションで API コール、またはパブリック DNS クエリを使用して、インスタンスを検出できるようにするには、このオプションを選択します。両方の方法を使用する必要はありません。

SOA TTL

API コールと VPC の DNS クエリまたはAPI コールとパブリック DNS クエリの場合、名前空間で作成された Route 53 ホストゾーンの認証局の開始 (SOA) DNS レコードの存続可能時間 (TTL) の値。この値は、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。値を小さくすると、欠落しているエントリがキャッシュされる時間 (負のキャッシュ) が短縮され、その名前空間に対する追加のクエリが犠牲になります。

タグ

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。タグは、 AWS リソースに割り当てることができるオプションのラベルです。各タグは、キーと値から構成されます。例えば、Key = Environment および Value = Production のタグを定義できます。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。

名前空間のタグは、作成後に更新または削除できます。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。

VPC

インスタンス検出の値として VPCs で API コールと DNS クエリを選択すると、 は同じ名前の Amazon Route 53 プライベートホストゾーン AWS Cloud Map を作成します。VPC リストで選択した VPC をそのプライベートホストゾーンに AWS Cloud Map 関連付けます。

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。

VPC をさらにプライベートホストゾーンに関連付けることができます。詳細については、Amazon Route 53 API リファレンスの「AssociateVPCWithHostedZone」を参照してください。

AWS Cloud Map 名前空間の表示

作成した名前空間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。

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-xxxxxxxxxxxxxxx',
            'CreateDate': 1585354387.357,
            'Id': 'ns-xxxxxxxxxxxxxxx',
            'Name': 'myFirstNamespace',
            'Properties': {
                'DnsProperties': {
                    'HostedZoneId': 'Z06752353VBUDTC32S84S',
                },
                'HttpProperties': {
                    'HttpName': 'myFirstNamespace',
                },
            },
            'Type': 'DNS_PRIVATE',
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxxx',
            'CreateDate': 1586468974.698,
            'Description': 'My second namespace',
            'Id': 'ns-xxxxxxxxxxxxxxx',
            'Name': 'mySecondNamespace.com',
            'Properties': {
                'DnsProperties': {
                },
                'HttpProperties': {
                    'HttpName': 'mySecondNamespace.com',
                },
            },
            'Type': 'HTTP',
```

```
        },
        {
            'Arn': 'arn:aws::servicediscovery:us-west-2:123456789012:namespace/
ns-xxxxxxxxxxxxxxx',
            'CreateDate': 1587055896.798,
            'Id': 'ns-xxxxxxxxxxxxxxx',
            '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

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。次の点に注意してください。

- 名前空間を削除する前に、名前空間に作成したサービスをすべて削除する必要があります。詳細については、「AWS Cloud Map サービスの削除」を参照してください。
- 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、登録を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。
- 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 内の DNS クエリを使用してサービスインスタンスを検出するように指定すると、 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. 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 サービスの使用

サービスは、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。これにより、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アプリケーションのリソースを検索できます。

トピック

- AWS Cloud Map サービスの作成
- AWS Cloud Map サービスの更新
- 名前空間内のサービスの表示
- AWS Cloud Map サービスの削除

AWS Cloud Map サービスの作成

サービスを作成するには、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. [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. [サービスの作成] ページに、適切な値を入力します。詳細については、「サービスの作成時に指定する値」を参照してください。
6. [Create service (サービスの作成)] を選択します。

AWS CLI

- `create-service` コマンドでサービスを更新します (*red* の値は独自の値で置き換え)。

```
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)

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. `create_service()`でサービスを作成します (*red* の値は独自の値で置き換え)。

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

> **Note**
>
> DNS クエリでアクセス可能なサービスの場合、大文字と小文字だけが異なる名前を持つ複数のサービス (例、EXAMPLE と example) を作成することはできません。そうしないと、これらのサービスは同じ 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。

サービスの作成時に指定する値

AWS Cloud Map サービスを作成するときは、次の値を指定します。

> **Note**
>
> サービス内のタグは、作成後にのみ変更できます。

[値]

- Service name
- Service description
- Service discovery configuration
- Routing policy
- Record type
- TTL
- Health check options
- Failure threshold
- Health check protocol
- Health check path
- Tags

サービス名

このサービスを使用して登録するインスタンスを表す名前を入力します。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。これ

は、名前空間の作成時に選択したインスタンス検出方法によって異なります。次のいずれかの方法を使用します。

- API コール – アプリケーションが を呼び出すとDiscoverInstances、API コールには名前空間名とサービス名が含まれます。
- API コールと VPC の DNS クエリまたは API コールとパブリック DNS クエリ - サービスインスタンスを登録し、名前空間を作成すると、 AWS Cloud Map によって Amazon Route 53 のプライベートまたはパブリックのホストゾーンが作成されます。また、そのホストゾーンに DNS レコードを作成します。レコードの名前の形式は次のとおりです。

  service-name.namespace-name

  アプリケーションから DNS クエリを送信してサービスインスタンスを検出すると、そのクエリは、レコード名にサービスの名前を含むレコードを対象とします。

> **Note**
>
> DNS クエリをサポートする名前空間にサービスを作成する場合、そのサービスのサービスインスタンスを検出可能にするには、DNS クエリではなく DiscoverInstances API オペレーションを呼び出します。Service discovery configuration を参照してください。

インスタンスの登録時に SRV レコード AWS Cloud Map 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する場合は、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

> **Note**
>
> DNS クエリでアクセス可能なサービスの場合、大文字と小文字だけが異なる名前を持つ複数のサービス (例、EXAMPLE と example) を作成することはできません。それ以外の場合、これらのサービスは同じ DNS 名を持ち、区別できません。

サービスの説明

サービスの説明を入力します。ここに入力した値は、[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。

サービスディスカバリ設定

名前空間が DNS クエリをサポートしている場合、 は次のサービス検出オプション AWS Cloud Map をサポートします。

API および DNS

AWS Cloud Map サービスのインスタンスを登録すると、 は SRV レコードを作成します。サービスインスタンスは、 DiscoverInstances API オペレーションを使用して検出することもできます。

API のみ

AWS Cloud Map は、サービスのインスタンスの SRV レコードを作成しません。サービスインスタンスは、 DiscoverInstances API オペレーションを使用してのみ検出できます。

ルーティングポリシー (パブリックおよびプライベートの DNS 名前空間のみ)

パブリックまたはプライベートの DNS 名前空間を使用してサービスを作成する場合、インスタンス登録時に AWS Cloud Map によって作成された DNS レコードの Amazon Route 53 ルーティングポリシーを選択します。(パブリック DNS 名前空間には、インスタンス検出用の API コールとパブリック DNS クエリの値があり、プライベート DNS 名前空間には、API コールと VPC での DNS クエリがあります。)

> **Note**
>
> インスタンスの登録時に、コンソールを使用して 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。インスタンス AWS Cloud Map をプログラムで登録するときに 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。

AWS Cloud Map では、次の Route 53 ルーティングポリシーがサポートされています。

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した 1 つのインスタンスの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、DNS クエリに対して、正常なインスタンスの中からランダムに選択した 1 つのインスタンスの IP アドレスを返します。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、DNS クエリに対して最大 8 の正常なインスタンスの IP アドレスを返します。正常なインスタンスが 8 未満の場合、Route 53 は DNS クエリに対してすべての正常なインスタンスの IP アドレスを返します。

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 (パブリックおよびプライベートの DNS 名前空間のみ)

パブリックまたはプライベートの DNS 名前空間を使用してサービスを作成する場合は、インスタンスの登録時に が AWS Cloud Map 作成するレコードの DNS レコードタイプを選択します。Amazon Route 53 は、登録インスタンスの DNS クエリに応答して、適切な値を返します。

以下のレコードタイプがサポートされています。

A

インスタンスを登録する場合は、リソースの IP アドレスを IPv4 形式 (例: 192.0.2.44) で指定します。

AAAA

インスタンスを登録する場合は、リソースの IP アドレスを IPv6 形式 (例: 2001:0db8:85a3:0000:0000:abcd:0001:2345) で指定します。

CNAME

インスタンスを登録する場合は、リソースのドメイン名 (例: www.example.com) を指定します。次の点に注意してください。

- CNAME を選択する場合は、ルーティングポリシーには加重ルーティングを選択する必要があります。
- CNAME を選択する場合、ヘルスチェックオプションには Route 53 ヘルスチェックを選べません。

SRV

SRV レコードの値には、次の値が使用されます。

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

この値に関して、以下の点に注意してください。

- `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 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。
  - インスタンス登録時に [サービスインスタンス ID] に指定した値
  - 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、サービスインスタンス ID にテストと指定したとします。サービスの名前はバックエンドであり、名前空間の名前は example.com です。AWS Cloud Map は、SRV レコードで `service-hostname` 属性に次の値を割り当てます。

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

SRV レコードの設定を指定する場合は、次の点に注意してください。

- [IPv4 アドレス]、[IPv6 アドレス]、または両方に値を指定した場合、SRV レコードの `service-hostname` の値と同じ名前を持つ A レコードや AAAA レコードが AWS Cloud Map によって自動的に作成されます。

- HAProxy など、特定の SRV 形式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」を参照し、正しい名前形式を指定する方法の詳細を確認してください。

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 および AAAA
- CNAME
- SRV

レコードタイプ [A] および [AAAA] を指定した場合は、インスタンス登録時に IPv4 IP アドレス、IPv6 IP アドレス、またはその両方を指定することができます。

TTL (パブリックおよびプライベート DNS 名前空間のみ)

パブリックまたはプライベートの DNS 名前空間を使用してサービスを作成する場合は、[有効期限 (TTL)] (time to live) に値を入力します。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。

ヘルスチェックオプション

ヘルスチェックなし

ヘルスチェックを設定しない場合、トラフィックは、正常かどうかに関わらずサービスインスタンスにルーティングされます。

Route 53 ヘルスチェック (プライベート DNS 名前空間ではサポートされません)

Amazon Route 53 ヘルスチェックの設定を指定した場合は、インスタンスを登録すると必ず、 AWS Cloud Map によって Route 53 ヘルスチェックが作成され、インスタンスの登録を解除すると、そのヘルスチェックは削除されます。

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。

API コールを使用してインスタンスを検出する名前空間の場合、 は Route 53 ヘルスチェック AWS Cloud Map を作成します。ただし、ヘルスチェックを関連付け AWS Cloud Map るの 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。

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 VPC にあるインスタンスがあるとします。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と同じ VPC 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。

失敗しきい値 (Route 53 ヘルスチェックのみ)

リソースの現在のステータスを正常から異常、または異常から正常に変更するために、Amazon Route 53 でリソースが合格または不合格になる必要がある、Route 53 ヘルスチェックの連続回数。詳細については、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 がヘルスチェックの正常性を判断する方法」を参照してください。

ヘルスチェックプロトコル(Route 53 ヘルスチェックのみ)

リソースの正常性をチェックする際に Amazon Route 53 が使用する方法は次のとおりです。

HTTP

Route 53 は TCP 接続を確立しようとします。成功した場合、 Route 53 は HTTP リクエストを送信し、2xx または 3xx 形式の HTTP ステータスコードを待機します。

HTTPS

Route 53 は TCP 接続を確立しようとします。成功した場合、 Route 53 は HTTP リクエストを送信し、2xx または 3xx 形式の HTTP ステータスコードを待機します。

> **Important**
>
> HTTPS を選択する場合は、リソースが TLS v1.0 以降をサポートしている必要があります。

[ヘルスチェックプロトコル] の値に HTTPS を選択すると、追加料金が適用されます。詳細については、「Route 53 料金表」を参照してください。

TCP

Route 53 は TCP 接続を確立しようとします。

詳細については、「Amazon Route 53 がヘルスチェックの正常性を判断する方法」を参照してください。

ヘルスチェックのパス (Route 53 HTTP および HTTPS のヘルスチェックのみ)

ヘルスチェックを実行するときに Amazon Route 53 がリクエストするパス。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。

タグ

1 つ以上のタグを指定して、サービスに追加できます。タグは、 AWS リソースに割り当てることができるオプションのラベルです。各タグは、キーと値から構成されます。例えば、Key = Environment および Value = Production のタグを定義できます。タグを使用して AWS リソースを分類すると、それらのリソースの管理が容易になります。

タグを作成したら、名前空間上のタグをいつでも更新または削除できます。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。

AWS Cloud Map サービスの更新

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、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、サービスを編集する名前空間を選択します。
4. 「名前空間: *#####*」のページで、編集するサービスを選択し、[編集] をクリックします。
5. 「サービス: *#####*」のページで、[編集] をクリックします。

6. [サービスの編集] ページに、適切な値を入力します。
7. [サービスの更新] をクリックします。

AWS CLI

- update-service コマンドでサービスを更新します (*red* の値は独自の値で置き換え)。

```
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. update_service()でサービスを作成します (*red* の値は独自の値で置き換え)。

```
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"
}
```

名前空間内のサービスの表示

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。

AWS CLI

- `list-services` コマンドでサービスを一覧表示します。

```
aws servicediscovery list-services
```

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-
xxxxxxxxxxxxxxxx',
            'CreateDate': 1587081768.334,
            'DnsConfig': {
                'DnsRecords': [
                    {
                        'TTL': 60,
                        'Type': 'A',
                    },
                ],
                'RoutingPolicy': 'MULTIVALUE',
            },
            'Id': 'srv-xxxxxxxxxxxxxxxx',
            '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。

サービスを削除するには、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. 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の使用

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 DiscoverInstances API アクションを使用してインスタンスを検索します。

トピック

- AWS Cloud Map サービスインスタンスの登録
- サービスインスタンスを登録または更新するときに指定する値
- 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表示
- AWS Cloud Map サービスインスタンスの登録解除

AWS Cloud Map サービスインスタンスの登録

サービスインスタンスを登録するには、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. [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:*namespace-name*] ページで、使用するサービスを選択します。
5. [Service:*service-name*] ページで、[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] ページに、適切な値を入力します。詳細については、「サービスインスタンスを登録または更新するときに指定する値」を参照してください。
7. [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:
  - `ServiceId` で指定したサービスで定義した DNS レコードごとに、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` コマンドでサービスインスタンスを登録します (*red* の値は独自の値で置き換え)。

```
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:
   - `ServiceId` で指定したサービスで定義した DNS レコードごとに、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()` コマンドでサービスインスタンスを登録します (*red* の値は独自の値で置き換え)。

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

サービスインスタンスを登録または更新するときに指定する値

サービスインスタンスを登録するときは、以下の値を指定します。

[値]

- Instance type
- Service instance ID
- IPv4 address
- IPv6 address
- Port
- EC2 instance ID
- Custom attributes

インスタンスタイプ

以下の各インスタンスタイプは、選択された設定でのみ使用することができます。

IP アドレス

サービスインスタンスに関連付けられているリソースが、IP アドレスを使用してアクセスできる場合は、このオプションを選択します。

このオプションは、3 つのすべてのタイプの名前空間 (HTTP、パブリック DNS、プライベート DNS) で選択できます。

EC2 インスタンス

サービスインスタンスに関連付けられているリソースが、EC2インスタンスからアクセス可能な場合は、このオプションを選択します。

このオプションは、HTTP で選択することができます。

別のリソースを特定するための情報

サービスインスタンスに関連付けられているリソースが、IPアドレスやEC2インスタンス以外の値を使用してアクセスできる場合は、このオプションを選択します。他の値は、[カスタム属性] で指定します。

このオプションは、3 つのすべてのタイプの名前空間 (HTTP、パブリック DNS、プライベート DNS) で選択できます。

サービスインスタンス ID

インスタンスに関連付ける識別子。次の点に注意してください。

- 新しいインスタンスを登録するには、同じサービスを使用して登録したインスタンス間で一意の値を指定する必要があります。
- [サービスインスタンス ID] で指定されているサービスに、SRV レコードの設定が含まれている場合は、SRV レコードの値の一部として、サービスインスタンス ID の値が自動的に含まれます。詳細については、セクション サービスの作成時に指定する値 の「レコードタイプ」を参照してください。
- 既存のインスタンスは、プログラムで更新できます。を呼び出しRegisterInstance、サービスインスタンス ID とサービス ID の値を指定し、サービスインスタンスの新しい設定を指定します。が最初にインスタンスを登録したときにヘルスチェック AWS Cloud Map を作成した場合、 は古いヘルスチェック AWS Cloud Map を削除し、新しいヘルスチェックを作成します。

> **Note**
>
> ヘルスチェックは即時に削除されないため、Amazon Route 53 `ListHealthChecks` リクエストを送信した場合など、しばらく表示されます。

IPv4 アドレス

IPv4 IP アドレス (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プリケーションからアクセスできる場合)。

IPv6 アドレス

IPv6 IP アドレス (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プリケーションからアクセスできる場合)。

ポート

ポート (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む必要がある)。[ポート] は、サービスに SRV レコード、または Amazon Route 53 ヘルスチェックが含まれている場合に必要です。

EC2のインスタンス ID

リソースの EC2 インスタンス ID 形式のインスタンス ID。

カスタム属性

リソースと関連付けるキーと値のペアを指定します (存在する場合)。

最大 30 のカスタム属性を追加できます。次の点に注意してください。

- [キー] と [値] はいずれも指定する必要があります。
- [キー] には、255 文字以内で指定し、a ~ z、A ~ Z、0 ~ 9、その他の表示可能な ASCII 文字 (33 ~ 126 の 10 進数値) を使用することができます。スペース、タブなどの空白文字は使用できません。
- [値] には、1,024 文字以内で指定し、a ~ z、A ~ Z、0 ~ 9、その他の表示可能な ASCII 文字 (33 ~ 126 の 10 進数値)、スペース、タブを使用することができます。

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。

- 任意の値の更新: カスタム属性を含め、サービスインスタンスの登録時に指定した値を更新する場合は、サービスインスタンスを再登録し、すべての値を再指定します。サービスインスタンスの詳細の更新 を参照してください。
- 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詳細の更新

サービスインスタンスを更新するには

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 ID をコピーします。
6. [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、ステップ 5 でコピーした ID を [サービスインスタンス ID] にペーストします。
8. サービスインスタンスに適用する他のすべての値を入力します。サービスインスタンスの以前の値は保持されません。詳細については、「サービスインスタンスを登録または更新するときに指定する値」を参照してください。
9. [サービスインスタンスの登録] を選択します。

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。

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. [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のオプションを選択します。
5. [Service:*service-name*] ページで、登録を解除するサービスインスタンスのオプションを選択します。
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': {
        '...': '...',
    },
}
```

AWS Cloud MapAWS Cloud Map コンソールで使用できない機能

次の AWS Cloud Map 機能は、 AWS Cloud Map コンソールでは使用できません。これらの機能を使用するには、プログラムによる方法を使用して にアクセスする必要があります AWS Cloud Map。

サービス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

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は、 Elastic Load Balancing (ELB) ロードバランサーにトラフィックをルーティングするエイリアスレコードは作成できません。次の点に注意してください。

- サービスを作成する際、`RoutingPolicy` に `WEIGHTED` を指定する必要があります。これを行うには、コンソールを使用します。詳細については、「AWS Cloud Map サービスの作成」を参照してください。

  AWS Cloud Map API を使用してサービスを作成する方法については、 API リファレンスのCreateService「」を参照してください。 AWS Cloud Map
- 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 リファレンスの RegisterInstance を参照してください。

カスタムヘルスチェックの初期のヘルスステータスを指定する

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 リファレンスの RegisterInstance を参照してください。

未完了オペレーションのステータスの取得

名前空間を作成し、その作成処理が完了する前にブラウザウィンドウを閉じた場合は、コンソールで現在のステータスを確認することはできません。ステータスを取得するには、ListOperations を使用します。詳細については、AWS Cloud Map API リファレンスの ListOperations を参照してください。

チュートリアル

以下のチュートリアルでは、 AWS Cloud Map 名前空間を使用して一般的なタスクを実行する方法を示しています。

トピック

- チュートリアル: DNS クエリで AWS Cloud Map サービス検出を使用する
- チュートリアル: カスタム属性で AWS Cloud Map サービス検出を使用する

チュートリアル: DNS クエリで AWS Cloud Map サービス検出を使用する

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。

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/signup を開きます。
2. オンラインの手順に従います。

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

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

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

管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

   ルートユーザーを使用してサインインする方法については、AWS サインイン ユーザーガイドの「ルートユーザーとしてサインインする」を参照してください。

2. ルートユーザーの多要素認証 (MFA) を有効にします。

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

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

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

管理アクセス権を持つユーザーとしてサインインする

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

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

追加のユーザーにアクセス権を割り当てる

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

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

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

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

のインストール AWS Command Line Interface

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。

dig ユーティリティへのアクセス権がある

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、以下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 では、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作成する

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、詳細の表示を選択します。
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service-nameとして、 を値backend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た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られた 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られ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。

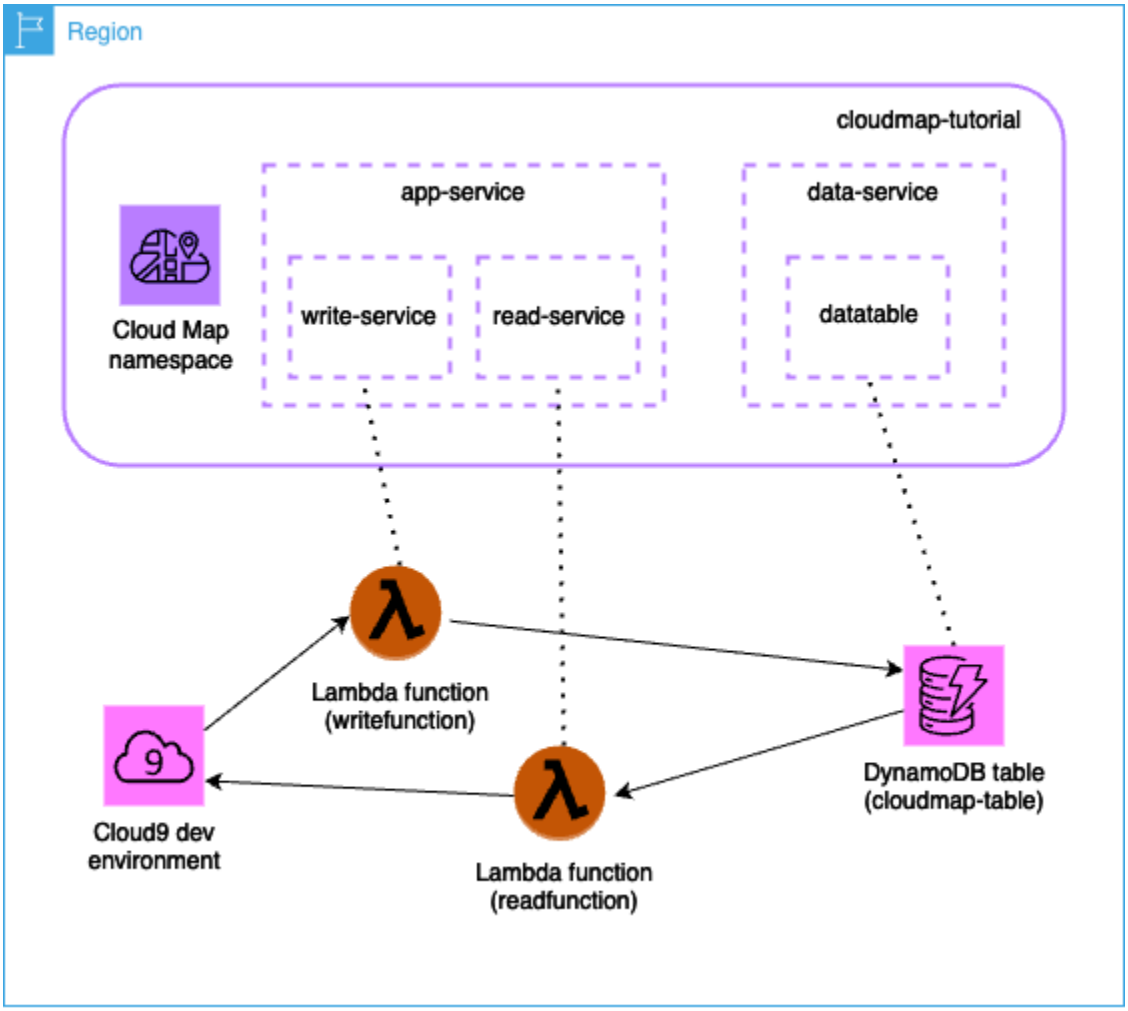
1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloudmap/ で AWS Cloud Map コンソールを開きます。
2. 名前空間のリストから`cloudmap-tutorial.com`名前空間を選択し、詳細を表示 を選択します。
3. 名前空間の詳細ページで、サービスのリストから`public-service`サービスを選択し、詳細の表示を選択します。
4. 「サービスインスタンス」セクションで、`first`インスタンスを選択し、「登録解除」を選択します。
5. ページ上部のパンくずリストを使用して、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。

チュートリアル: カスタム属性で AWS Cloud Map サービス検出を使用する

このチュートリアルでは、 AWS Cloud Map API 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。このチュートリアルでは、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る AWS Cloud9 環境内でクライアントアプリケーションを作成する手順を説明します。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。

次の図は、このチュートリアルで使用する高レベルのアーキテクチャを示しています。



> **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

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。

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/signup を開きます。
2. オンラインの手順に従います。

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

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

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

管理アクセスを持つユーザーを作成する

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

のセキュリティ保護 AWS アカウントのルートユーザー

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

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

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

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

手順については、「AWS IAM Identity Center ユーザーガイド」の「AWS IAM Identity Centerの有効化」を参照してください。

2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

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

管理アクセス権を持つユーザーとしてサインインする

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

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

追加のユーザーにアクセス権を割り当てる

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

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

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

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

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 AWS Cloud Map API コールで検出できます。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、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。
4. (オプション) 名前空間の説明 で、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 を選択します。
6. 残りのデフォルト値はそのままにして、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。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/dynamodb/ で DynamoDB 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、テーブル 、テーブル の作成 を選択します。
3. テーブルの作成ページで、次の操作を行います。
   a. テーブル名 には、 を指定しますcloudmap-table。
   b. パーティションキー には、 を指定しますid。
   c. 残りのデフォルト値のままにして、テーブルの作成 を選択します。

ステップ 3: AWS Cloud Map データサービスを作成する

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. 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 IAM ロールはこのチュートリアルにのみ使用され、後で削除できるため、ロールに名前を付けてアクセス許可の境界をcloudmap-role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、https://console.aws.amazon.com/iam/ で IAM コンソールを開きます。
2. IAM コンソールのナビゲーションペインで、[ロール]、[ロールを作成] 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. [次へ] をクリックします。
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. [次へ] をクリックします。
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、 AWS Cloud Map API を使用して作成した AWS Cloud Map サービスのクエリを実行して、DynamoDB テーブルにデータを書き込む Lambda 関数を作成します。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/lambda/ で AWS Lambda コンソールを開きます。
2. 左側のナビゲーションで、関数 、関数 の作成 を選択します。
3. 関数の作成ページで、次の操作を行います。
   a. [ゼロから作る] を選択します。
   b. 関数名 には、 を指定しますwritefunction。
   c. ランタイム で、 を選択しますPython 3.12。
   d. アーキテクチャ で、 を選択しますx86_64。
   e. 「アクセス許可」セクションで、次の操作を行います。
      i. デフォルトの実行ロールの変更オプションを展開し、既存のロールを使用する を選択します。
      ii. 既存のロール では、ドロップダウンメニューを使用して、 で作成した IAM ロールを選択しますステップ 4: AWS Lambda 実行ロールを作成する。
      iii. 残りのデフォルト値はそのままにして、関数の作成 を選択します。
   f. コードタブの「コードソース」セクションで、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。

```
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('cloudmap-table')

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

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

g. デプロイ を選択して関数を更新します。

ステップ 6: AWS Cloud Map アプリサービスを作成する

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. 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 関数を作成します。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/lambda/ で AWS Lambda コンソールを開きます。
2. 左側のナビゲーションで、関数 、関数 の作成 を選択します。
3. 関数の作成ページで、次の操作を行います。

   a. [ゼロから作る] を選択します。

   b. 関数名 には、 を指定しますreadfunction。

   c. ランタイム で、 を選択しますPython 3.12。

   d. アーキテクチャ で、 を選択しますx86_64。

   e. 「アクセス許可」セクションで、次の操作を行います。

      i. 「デフォルトの実行ロールの変更」オプションを展開し、「既存のロールを使用する」を選択します。

      ii. 既存のロール では、ドロップダウンメニューを使用して、 で作成した IAM ロールを選択しますステップ 4: AWS Lambda 実行ロールを作成する。

iii. 残りのデフォルト値はそのままにして、関数の作成 を選択します。

f. コードタブの「コードソース」セクションで、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('cloudmap-table')

    response = table.get_item(Key={'id': event})

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

g. デプロイ を選択して関数を更新します。

ステップ 8: AWS Cloud Map サービスインスタンスを作成する

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. 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。

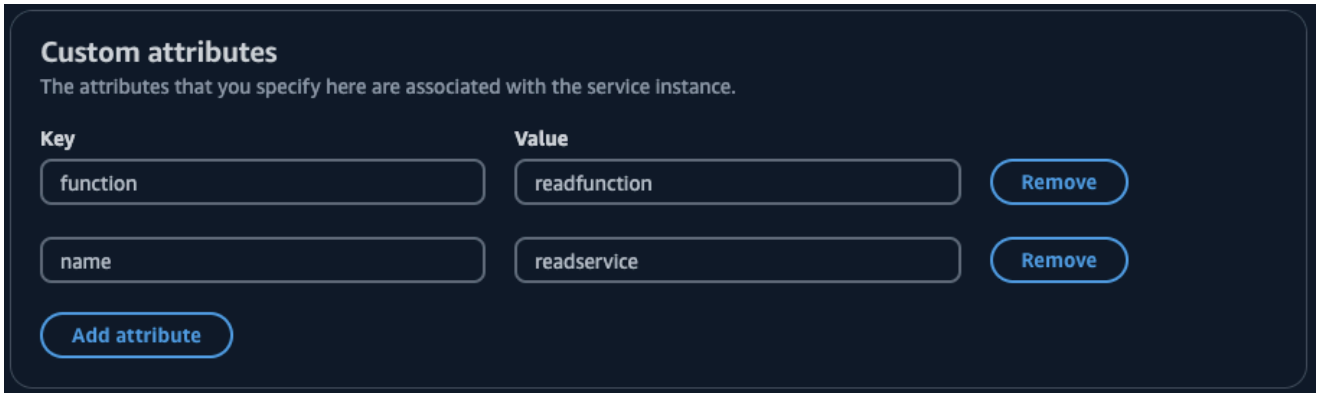
a. インスタンスタイプ で、別のリソース の識別情報を選択します。

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: 開発環境を作成する

AWS Cloud9 は、 によって管理される統合開発環境 (IDE) です AWS。 AWS Cloud9 IDE は、動的プログラミングに必要なソフトウェアとトゥーリングを提供します。このステップでは、 AWS Cloud9 環境を作成し、 AWS API でプログラミング AWS SDK for Python (Boto3) する を使用して環境を設定します。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloud9/ で AWS Cloud9 コンソールを開きます。
2. 左側のナビゲーションメニューで、環境 を選択し、環境の作成 を選択します。
3. 「環境の作成」ページで、次の操作を実行して開発環境を作成します。

   a. 名前 には、 を使用しますcloudmap-tutorial。

   b. 環境タイプ で、新しい EC2 インスタンス を選択します。

   c. インスタンスタイプ で、t2.micro を選択します。

   d. プラットフォーム では、ドロップダウンメニューを使用して Ubuntu Server 22.04 LTS を選択します。

e. 残りのデフォルト選択のままにして、 の作成を選択します。

4. AWS Cloud9 環境が作成されたら、`cloudmap-tutorial`環境を選択し、Cloud9 で開くを選択します。これにより、開発環境が新しいタブで開き、使用する bash シェルが表示されます。

> **Important**
>
> AWS Cloud9 環境を開く際に問題が発生した場合は、「 AWS Cloud9 ユーザーガイドAWS Cloud9 」の「トラブルシューティング: 環境を開くことができません」を参照してください。

5. bash シェルを使用して、次のコマンドを実行して環境を設定します。

   a. 環境を更新します。

   ```
   sudo apt-get -y update
   ```

   b. `python3` がインストールされていることを確認します。

   ```
   python3 --version
   ```

   c. Boto3 パッケージを環境にインストールします。

   ```
   sudo apt install -y python3-boto3
   ```

ステップ 10: フロントエンドクライアントを作成する

前のステップで作成した AWS Cloud9 開発環境を使用して、 で設定したサービスを検出 AWS Cloud Map し、これらのサービスを呼び出すコードを使用するフロントエンドクライアントを作成します。

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloud9/ で AWS Cloud9 コンソールを開きます。
2. 左側のナビゲーションメニューで、マイ環境を選択し、`cloudmap-tutorial`環境を選択し、Cloud9 で開くを選択します。
3. AWS Cloud9 環境のファイルメニューで、 という名前のファイルを作成する新しいファイルを選択します`Untitled1`。

4. Untitled1 ファイルで、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。Lambda 関数の名前が返され、DynamoDB テーブルへのデータの書き込みを担当します。次に、Lambda 関数が呼び出され、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())
```

5. ファイルメニューから名前を付けて保存... を選択し、ファイルを として保存しますwriteclient.py。
6. AWS Cloud9 環境の bash シェルから、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}}"}'
```

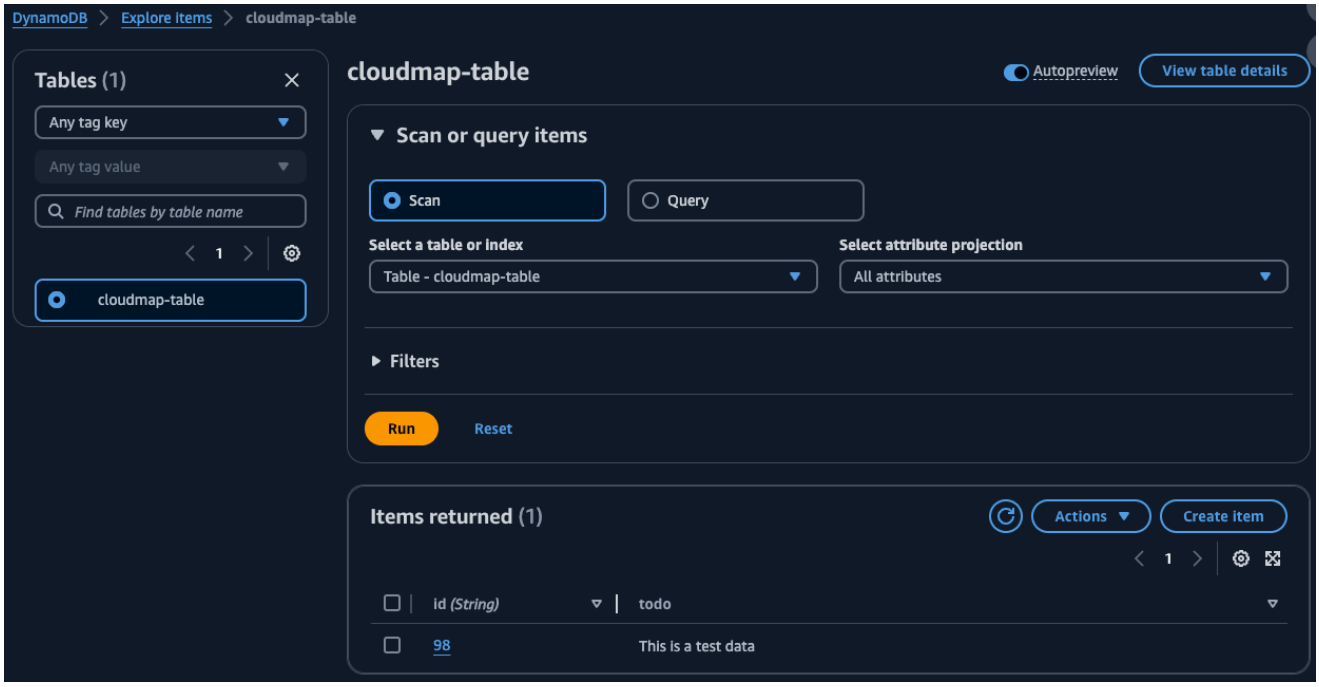
7. 前のステップで書き込みが成功したことを確認するには、読み取りクライアントを作成します。
   a. にサインイン AWS Management Console し、https://console.aws.amazon.com/dynamodb/ で DynamoDB コンソールを開きます。

b. 左のナビゲーションペインで、[テーブル] を選択します。

c. テーブルのリストから、cloudmap-table を選択し、アクションメニューを使用してアイテムの探索 を選択します。

d. Items returned セクションで、id(String) 列の数値を書き留めます。

以下は、id(String) 値が である例を示しています98。



e. AWS Cloud9 環境のファイルメニューで、 という名前のファイルを作成する新しいファイルを選択しますUntitled1。

f. Untitled1 ファイルで、次のコードをコピーして貼り付けます。Payload 値を前のステップの DynamoDB テーブルid（String)の値に置き換えます。このコードはテーブルから読み取り、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', Payload='"98"')

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

g. ファイルメニューから名前を付けて保存... を選択し、ファイルを として保存しますreadclient.py。

h. AWS Cloud9 環境の bash シェルから、次のコマンドを使用して Python コードを実行します。

```
python3 readclient.py
```

出力は以下の例のようになります。

```
b'{"statusCode": 200, "body": "{\\"Item\\": {\\"id\\": \\"98\\", \\"todo\
\": \\"This is a test data\\"}, \\"ResponseMetadata\\": {\\"RequestId\\": \
\"JSO5DLRGF0JUPQN4NCH369ABMBVV4KQNSO5AEMVJF66Q9ASUAAJG\\", \\"HTTPStatusCode\
\": 200, \\"HTTPHeaders\\": {\\"server\\": \\"Server\\", \\"date\\": \\"Wed, 06
 Mar 2024 23:03:38 GMT\\", \\"content-type\\": \\"application/x-amz-json-1.0\
\", \\"content-length\\": \\"61\\", \\"connection\\": \\"keep-alive\\", \\"x-
amzn-requestid\\": \\"JSO5DLRGF0JUPQN4NCH369ABMBVV4KQNSO5AEMVJF66Q9ASUAAJG\\",
 \\"x-amz-crc32\\": \\"3104232745\\"}, \\"RetryAttempts\\": 0}}"}'
```

ステップ 11: 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、 リソースを削除できます。 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、Lambda、DynamoDB、および AWS Cloud9 リソースをクリーンアップする手順を説明します。

AWS Cloud9 リソースを削除するには

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/cloud9/ で AWS Cloud9 コンソールを開きます。
2. 左側のナビゲーションメニューで、マイ環境 を選択します。
3. `cloudmap-tutorial` 環境を選択し、「削除」を選択します。
4. 削除を確定するには、「削除」と入力し`Delete`、「」を選択します。

Lambda 関数を削除するには

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/lambda/ で AWS Lambda コンソールを開きます。
2. 左側のナビゲーションで、関数 を選択します。
3. 関数`writefunction`と `readfunction`関数の両方を選択します。
4. [アクション] メニューから、[削除] を選択します。
5. 削除を確定するには、 と入力し`delete`、「削除」を選択します。

DynamoDB テーブルを削除する

1. にサインイン AWS Management Console し、https://console.aws.amazon.com/dynamodb/ で DynamoDB コンソールを開きます。
2. 左のナビゲーションペインで、[テーブル] を選択します。
3. `cloudmap-table` テーブルを選択し、削除を選択します。
4. 削除を確定するには、 と入力し`confirm`、「削除」を選択します。

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、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`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 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。

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。 AWS のお客様は、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利点を得られます。

セキュリティは、 AWS とユーザー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 を担います AWS 。また、 は、ユーザーが安全に使用できるサービス AWS も提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。に適用されるコンプライアンスプログラムの詳細については 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、 を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます 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

AWS Identity and Access Management に AWS Cloud Map

AWS Identity and Access Management (IAM) では、ドメインの登録やレコードの更新など、 AWS Cloud Map リソースに対して何らかのアクションを実行するには、承認されたユーザーであることを認証する必要があります。 AWS AWS Cloud Map コンソールを使用している場合は、 AWS ユーザー名とパスワードを入力して ID を認証します。 AWS Cloud Map プログラムでアクセスする場合、アプリケーションはアクセスキーを使用するか、リクエストに署名することで、ユーザーの ID を認証します。

ID を認証すると、IAM AWS はアクションを実行したりリソースにアクセスしたりする権限があることを確認して、ユーザーへのアクセスを制御します。アカウント管理者である場合、IAM を使用して、アカウントに関連付けられたリソースへの他のユーザーのアクセスをコントロールできます。

この章では、IAM AWS Cloud Map の使用方法とリソースの保護方法について説明します。

トピック

- 認証
- アクセスコントロール

認証

次のいずれかでアクセスできます AWS 。

- AWS アカウントのルートユーザー – AWS アカウントを初めて作成する場合は、このアカウントのすべての AWS サービスとリソースに対して完全なアクセス権限を持つシングルサインインアイデンティティで始めます。このアイデンティティはAWS アカウントのルートユーザーアカウントと呼ばれ、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のサインインによりアクセスします。を作成するときは AWS アカウント、 AWS のサービス アカウント内のすべてのリソースに完全にアクセスできる 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 root ユーザーと呼ばれ、アカウントの作成に使用したメールアドレスとパスワードでサインインすることでアクセスされます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、『IAM ユーザーガイド』の「ルートユーザー認証情報が必要なタスク」を参照してください。
- IAM ユーザー — IAM ユーザーは、特定のカスタム権限 (たとえば、で HTTP 名前空間を作成する権限) AWS を持つアカウント内の ID です。 AWS Cloud Map IAMサインインの認証情報を使用し

て、AWS Management Console、AWS ディスカッションフォーラム、 AWS Support センターなどの AWS ウェブページにサインインします。

また、サインイン認証情報に加えて、各ユーザーのアクセスキーを生成することができます。これらのキーは、複数の SDK のいずれかを介して、 AWS またはを使用してプログラムでサービスにアクセスする場合に使用できます。AWS Command Line InterfaceSDK と CLI ツールでは、アクセスキーを使用してリクエストが暗号で署名されます。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。 AWS Cloud Map インバウンド API リクエストを認証するためのプロトコルである署名バージョン 4 をサポートします。リクエストの認証の詳細については、『』の「署名バージョン 4 の署名プロセスAmazon Web Services 全般のリファレンス」を参照してください。

- IAM ロール - IAM ロールは、アカウントで作成して特定の許可を付与できる IAM ID です。IAM ロールは、ID で実行できることとできないことを決定するアクセス権限ポリシーを備えた AWS ID であるという点で IAM ユーザーと似ています。 AWSただし、ユーザーは 1 人の特定の人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。また、ロールには標準の長期認証情報 (パスワードやアクセスキーなど) も関連付けられません。代わりに、ロールを引き受けると、ロールセッション用の一時的なセキュリティ認証情報が提供されます。IAM ロールと一時的な認証情報は、次の状況で役立ちます:

  - フェデレーションユーザーアクセス — IAM ユーザーを作成する代わりに、エンタープライズユーザーディレクトリ、またはウェブ ID AWS Directory Serviceプロバイダーの既存のユーザー ID を使用できます。これらはフェデレーティッドユーザーと呼ばれます。 AWS ID プロバイダーを通じてアクセスが要求されたときに、フェデレーティッドユーザーにロールを割り当てます。フェデレーションユーザーの詳細については、「IAM ユーザーガイド」の「フェデレーションユーザーとロール」を参照してください。
  - AWS サービスアクセス — アカウントの IAM ロールを使用して、 AWS アカウントのリソースにアクセスするためのアクセス権限をサービスに付与できます。例えば、Amazon Redshift がユーザーに代わって Simple Storage Service (Amazon S3) バケットにアクセスし、そのバケットのデータを Amazon Redshift クラスターにロードすることを許可するロールを作成できます。詳細については、『IAM ユーザーガイド』の「AWS サービスにアクセス権限を委任するロールの作成」を参照してください。
  - Amazon EC2 で実行されるアプリケーション — IAM ロールを使用して、Amazon EC2 インスタンスで実行され、 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、Amazon EC2 インスタンス内でのアクセスキーの保存に推奨されます。Amazon EC2 AWS インスタンスにロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされる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、Amazon EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得することができます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して権限を付与する」を参照してください。

アクセスコントロール

AWS Cloud Map リソースを作成、更新、削除、または一覧表示するには、アクションを実行する権限と、対応するリソースにアクセスする権限が必要です。また、プログラムでアクションを実行するには、有効なアクセスキーが必要です。

以下のセクションでは、の権限を管理する方法について説明します AWS Cloud Map。最初に概要のセクションを読むことをお勧めします。

- AWS Cloud Map リソースへのアクセス許可の管理の概要
- アイデンティティベースのポリシー (IAM ポリシー) の使用方法 AWS Cloud Map
- AWS Cloud Map API 権限:アクション、リソース、条件リファレンス

AWS Cloud Map リソースへのアクセス許可の管理の概要

AWS AWS すべてのリソースはアカウントが所有し、リソースを作成またはアクセスする権限はアクセス権限ポリシーによって管理されます。

> **Note**
>
> アカウント管理者 (または管理者ユーザー) は、管理者権限を持つユーザーです。管理者の詳細については、IAM ユーザーガイドの「IAM ベストプラクティス」を参照してください。

アクセス許可を付与するときは、アクセス許可を取得するユーザー、アクセス許可を取得する対象のリソース、およびアクセス許可を取得して実行するアクションを決定します。

トピック

- リソースの ARN AWS Cloud Map
- リソース所有権について
- リソースへのアクセスの管理

- ポリシー要素の指定: リソース、アクション、効果、プリンシパル
- IAM ポリシーでの条件の指定

リソースの ARN AWS Cloud Map

選択したオペレーションの名前空間およびサービスにリソースレベルのアクセス許可を付与または拒否することができます。詳細については、「AWS Cloud Map API 権限:アクション、リソース、条件リファレンス」を参照してください。

リソース所有権について

AWS アカウントは、誰がリソースを作成したかにかかわらず、そのアカウントで作成されたリソースを所有します。具体的には、リソース所有者は、リソース作成リクエストを認証するプリンシパルエンティティ (つまり、ルートユーザーアカウント、IAM ユーザー、または IAM ロール) のアカウントです。 AWS

次の例は、この仕組みを示しています。

- アカウントのルートユーザーアカウント認証情報を使用して HTTP ネームスペースを作成すると、 AWS そのアカウントがリソースの所有者になります。 AWS
- AWS アカウントに IAM ユーザーを作成し、そのユーザーに HTTP 名前空間を作成する権限を付与すると、そのユーザーは HTTP 名前空間を作成できます。ただし、ユーザーが属する AWS アカウントが HTTP 名前空間リソースを所有します。
- HTTP ネームスペースを作成する権限を持つ IAM AWS ロールをアカウント内に作成すると、そのロールを引き受けることができるユーザーなら誰でも HTTP ネームスペースを作成できます。ロールが属する AWS アカウントが HTTP 名前空間リソースを所有しています。

リソースへのアクセスの管理

アクセス許可のポリシーでは、誰が何にアクセスできるかを指定します。このセクションでは、AWS Cloud Mapのアクセス権限のポリシーを作成するために使用可能なオプションについて説明します。IAM ポリシー構文の概説については、IAM ユーザーガイドの「IAM ポリシーリファレンス」を参照してください。

IAM アイデンティティに添付されたポリシーはアイデンティティベースのポリシー (IAM ポリシー) と呼ばれ、リソースに添付されたポリシーはリソースベースのポリシーと呼ばれます。 AWS Cloud Map はアイデンティティベースのポリシー (IAM ポリシー) のみをサポートします。

トピック

- アイデンティティベースのポリシー (IAM ポリシー)
- リソースベースのポリシー

アイデンティティベースのポリシー (IAM ポリシー)

ポリシーを IAM アイデンティティにアタッチできます。例えば、次のオペレーションを実行できます。

- アカウントのユーザーまたはグループに許可ポリシーを添付する - アカウント管理者は、特定のユーザーに関連付けられる許可ポリシーを使用して、そのユーザーに AWS Cloud Map リソースを作成する許可を付与することができます。
- ロールにアクセス権限ポリシーをアタッチする (クロスアカウント権限を付与) — 別のアカウントが作成したユーザーに、 AWS Cloud Map アクションを実行する権限を付与できます。 AWS そのためには、IAM ロールにアクセス許可ポリシーをアタッチし、他のアカウントのユーザーがそのロールを引き受けられるようにします。次の例では、これがアカウント A とアカウント B の 2 つの AWS アカウントでどのように機能するかを説明します。

  1. アカウント A の管理者は、IAM ロールを作成して、アカウント A が所有するリソースの作成や操作をするアクセス許可を付与するアクセス許可ポリシーをロールにアタッチします。
  2. アカウント A の管理者は、ロールに信頼ポリシーをアタッチします。信頼ポリシーは、ロールを引き受けることのできるプリンシパルとしてアカウント B を識別します。
  3. 次に、アカウント B の管理者は、ロールを引き受けるアクセス許可をアカウント B のユーザーまたはグループに委任できます。これにより、アカウント B のユーザーはアカウント A でリソースを作成したり、リソースにアクセスしたりできます。

  別の AWS アカウントのユーザーに許可を委任する方法については、IAM ユーザーガイドの「アクセス管理」を参照してください。

以下のポリシー例では、 AWS ユーザーは任意のアカウントのパブリック DNS CreatePublicDnsNamespace名前空間を作成するアクションを実行できます。パブリック DNS ネームスペースを作成すると、Route 53 AWS Cloud Map ホストゾーンも作成されるため、Amazon Route 53 のアクセス権限が必要です。

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

```
        {
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreatePublicDnsNamespace",
                "route53:CreateHostedZone",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName"
            ],
            "Resource":"*"
        }
    ]
}
```

代わりにポリシーをプライベート DNS 名前空間に適用したい場合は、アクションを使用するためのアクセス権限を付与する必要があります。 AWS Cloud Map CreatePrivateDnsNamespaceさらに、Route 53 AWS Cloud Map のプライベートホストゾーンが作成されるため、前の例と同じ Route 53 アクションを使用する権限を付与します。また、`DescribeVpcs` と `DescribeRegions` の 2 つの Amazon EC2 アクションを使用するための許可も付与します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreatePrivateDnsNamespace",
                "route53:CreateHostedZone",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName"
            ],
            "Resource":"*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeRegions"
            ],
            "Resource":"*"
        }
    ]
}
```

の ID にポリシーをアタッチする方法の詳細については AWS Cloud Map、「」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 (IAM ポリシー) の使用方法 AWS Cloud Mapユーザー、グループ、ロール、許可の詳細については、「IAM ユーザーガイド」の「アイデンティティ (ユーザー、グループ、ロール)」を参照してください。

リソースベースのポリシー

Amazon S3 などの他のサービスでは、リソースへのアクセス許可ポリシーのアタッチもサポートされています。たとえば、S3 バケットにポリシーをアタッチして、そのバケットへのアクセス権限を管理できます。 AWS Cloud Map リソースへのポリシーのアタッチはサポートされていません。

ポリシー要素の指定: リソース、アクション、効果、プリンシパル

AWS Cloud Map には、 AWS Cloud Map 各リソースで使用できる AWS Cloud Map API アクション (「API リファレンス」を参照) が含まれています (「」を参照リソースの ARN AWS Cloud Map)。これらのアクションの一部またはすべてを実行するアクセス許可を、ユーザーまたはフェデレーティッドユーザーに付与できます。パブリック DNS 名前空間の作成など、一部の API アクションでは複数のアクションを実行するアクセス許可が必要な点に注意してください。

以下は、基本的なポリシーの要素です。

- リソース - Amazon リソースネーム (ARN) を使用して、ポリシーを適用するリソースを識別します。詳細については、「リソースの ARN AWS Cloud Map」を参照してください。
- アクション - アクションのキーワードを使用して、許可または拒否するリソースアクションを識別します。たとえば、指定された内容に応じて`Effect`、`servicediscovery:CreateHttpNamespace` AWS Cloud Map CreateHttpNamespace権限によってアクションの実行がユーザーに許可または拒否されます。
- 効果 - ユーザーが指定されたリソースでアクションの実行を試みた場合の、許可または拒否の効果を指定します。アクションへのアクセスを明示的に許可していない場合、アクセスは暗黙的に拒否されます。また、明示的にリソースへのアクセスを拒否すると、別のポリシーによってアクセスが許可されている場合でも、ユーザーはそのリソースにアクセスできなくなります。
- プリンシパル – ID ベースのポリシー (IAM ポリシー) で、ポリシーがアタッチされているユーザーが黙示的なプリンシパルとなります。リソースベースのポリシーでは、アクセス許可を受け取りたいユーザー、アカウント、サービス、またはその他のエンティティを指定します (リソースベースのポリシーにのみ適用)。 AWS Cloud Map では、リソースベースのポリシーはサポートされていません。

IAM ポリシー構文の詳細と説明については、IAM ユーザーガイドのIAM ポリシーリファレンスを参照してください。

AWS Cloud Map API アクションとそれらが適用されるリソースのリストについては、を参照してくださいAWS Cloud Map API 権限:アクション、リソース、条件リファレンス。

IAM ポリシーでの条件の指定

アクセス許可を付与するとき、IAM ポリシー言語を使用して、いつポリシーが有効になるかを指定できます。たとえば、指定した日付の後にのみポリシーを適用、または指定した名前空間にのみポリシーを適用する場合などです。

条件を表現するには、定義済みの条件キーを使用します。 AWS Cloud Map 独自の条件キーセットを定義し、一部のグローバル条件キーの使用もサポートしています。詳細については、次のトピックを参照してください。

- AWS Cloud Map 条件キーの詳細については、を参照してくださいAWS Cloud Map API 権限:アクション、リソース、条件リファレンス。
- AWS グローバル条件キーについては、IAM ユーザーガイドの「AWS グローバル条件コンテキストキー」を参照してください。
- ポリシー言語での条件の指定の詳細については、IAM ユーザーガイドの「IAM JSON ポリシーの要素: Condition」を参照してください。

アイデンティティベースのポリシー (IAM ポリシー) の使用方法 AWS Cloud Map

このトピックでは、アカウント管理者が IAM アイデンティティ (ユーザー、グループ、ロール) にアクセス権限ポリシーをアタッチして、リソースに対してアクションを実行するアクセス権限を付与する方法を示すアイデンティティベースのポリシーの例を示します。 AWS Cloud Map

> **Important**
>
> リソースへのアクセスを管理するための基本概念とオプションを説明する概要トピックを最初に確認することをお勧めします。 AWS Cloud Map 詳細については、「AWS Cloud Map リソースへのアクセス許可の管理の概要」を参照してください。

トピック

- AWS Cloud Map コンソールを使用するために必要なアクセス権限

以下の例は、サービスインスタンスを登録または登録解除するための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 DNS 名前空間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 の権限が必要です。インスタンスを登録および登録解除すると、Route 53 AWS Cloud Map のレコードとヘルスチェックが作成、更新、削除されるためです。のワイルドカード文字 (*) は、 AWS Cloud Map 現在のアカウントが所有するすべてのインスタンス、Route 53 のレコード、Resourceヘルスチェックへのアクセスを許可します。 AWS

各アクションを使用するアクセス許可を付与または拒否するために指定するアクションおよび ARN のリストについては、「AWS Cloud Map API 権限:アクション、リソース、条件リファレンス」を参照してください。

AWS Cloud Map コンソールを使用するために必要なアクセス権限

AWS Cloud Map コンソールへのフルアクセスを許可するには、以下のアクセス権限ポリシーで権限を付与します。

```
{
   "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:*`

AWS Cloud Map すべての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 DNS 名前空間とプライベート DNS 名前空間を作成および削除するときに、 AWS Cloud Map ホストゾーンを管理できます。

`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、 AWS Cloud Map ヘルスチェックを管理しましょう。

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

AWS Cloud Map プライベートホストゾーンの管理にお任せください。

AWS の AWS Cloud Map 管理ポリシー

AWS マネージドポリシーは、AWS が作成および管理するスタンドアロンポリシーです。AWS マネージドポリシーは、多くの一般的なユースケースでアクセス許可を提供できるように設計されているため、ユーザー、グループ、ロールへのアクセス許可の割り当てを開始できます。

AWS マネージドポリシーは、ご利用の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ご注意ください。AWS のすべてのお客様が使用できるようになるのを避けるためです。ユースケース別にカスタマー管理ポリシーを定義することで、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義したアクセス権限は変更できません。AWS が AWS マネージドポリシーに定義されているアクセス許可を更新すると、更新はポリシーがアタッチされているすべてのプリンシパルアイデンティティ (ユーザー、グループ、ロール) に影響します。新しい AWS のサービスを起動するか、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ると、AWS が AWS マネージド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、「IAM ユーザーガイド」の「AWS 管理ポリシー」を参照してください。

AWS 管理ポリシー: AWSCloudMapDiscoverInstanceAccess

IAM エンティティに `AWSCloudMapDiscoverInstanceAccess` をアタッチできます。ディスカバリー API AWS Cloud Map へのアクセスを提供します。

このポリシーに対する許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSCloudMapDiscoverInstanceAccess」を参照してください。

AWS 管理ポリシー: AWSCloudMapReadOnlyAccess

IAM エンティティに `AWSCloudMapReadOnlyAccess` 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。

このポリシーに対する許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSCloudMapReadOnlyAccess」を参照してください。

AWS 管理ポリシー: AWSCloudMapRegisterInstanceAccess

IAM エンティティに `AWSCloudMapRegisterInstanceAccess` 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。

このポリシーに対する許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSCloudMapRegisterInstanceAccess」を参照してください。

AWS 管理ポリシー: AWSCloudMapFullAccess

IAM エンティティに `AWSCloudMapFullAccess` 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許可

このポリシーに対する許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSCloudMapFullAccess」を参照してください。

AWS Cloud Map マネージドポリシーの AWS 更新

このサービスがこれらの変更の追跡を開始してからの、AWS の AWS Cloud Map 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通知については、AWS Cloud Map [Document history] (ドキュメントの履歴) ページの RSS フィードをサブスクライブしてください。

変更	説明	日付
AWSCloudMapDiscoverInstanceAccess、AWSCloudMapRegisterInstanceAccess、AWSCloudMapReadOnlyAccess — 既存のポリシーの更新。	AWS Cloud Map 新しい AWS Cloud Map `DiscoverInstanceRevision` API オペレーションにアクセスできるように、これらのポリシーを更新しました。	2023 年 8 月 15 日

お客様が管理するポリシーの例

独自のカスタム IAM ポリシーを作成して、AWS Cloud Map アクションに許可を付与することもできます。これらのカスタムポリシーは、指定されたアクセス許可が必要な IAM ユーザーまたはグループにアタッチできます。これらのポリシーは、AWS Cloud Map API、AWS SDK、または AWS CLI の使用時に適用されます。次の例では、いくつかの一般的ユースケースのアクセス許可を示します。AWS Cloud Map へのフルアクセスをユーザーに許可するポリシーについては、「AWS Cloud Map コンソールを使用するために必要なアクセス権限」を参照してください。

例

- 例 1: 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
- 例 2: すべてのタイプの名前空間の作成を許可する

例 1: 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":"*"
      }
   ]
}
```

例 2: すべてのタイプの名前空間の作成を許可する

次のアクセス許可ポリシーでは、すべてのタイプの名前空間の作成をユーザーに許可します。

```
{
   "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":"*"
        }
    ]
}
```

アクセスを提供するには、ユーザー、グループ、またはロールにアクセス許可を追加します。

- AWS IAM Identity Center のユーザーとグループ:

  アクセス許可セットを作成します。「AWS IAM Identity Center ユーザーガイド」の「シークレットの作成と管理」の手順に従ってください。

- ID プロバイダーを通じて IAM で管理されているユーザー:

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

- IAM ユーザー:
  - ユーザーが設定できるロールを作成します。手順については、「IAM ユーザーガイド」の「IAM ユーザー用ロールの作成」を参照してください。
  - (非推奨) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。「IAM ユーザーガイド」の「ユーザー (コンソール) へのアクセス許可の追加」の指示に従います。

AWS Cloud Map API 権限:アクション、リソース、条件リファレンス

アクセスコントロール をセットアップし、IAM ID にアタッチできるアクセス許可ポリシー (ID ベースのポリシー) を作成するときは、以下のリストをリファレンスとして使用できます。リストには、各 AWS Cloud Map API アクション、アクセス権限を付与する必要があるアクション、 AWS および

アクセスを許可する必要があるリソースが含まれます。ポリシーの `Action` フィールドでアクションを指定し、ポリシーの `Resource` フィールドでリソースの値を指定します。

一部のオペレーションでは AWS Cloud Map、IAM ポリシーで特定の条件キーを使用できます。詳細については、「AWS Cloud Map 条件キーリファレンス」を参照してください。 AWS ワイドコンディションキーも使用できます。 AWS ワイドキーの全リストについては、IAM ユーザーガイドの「使用可能なキー」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、API アクション名の順の文字列を使用します (例: `servicediscovery:CreatePublicDnsNamespace` および `route53:CreateHostedZone`)。

トピック

- AWS Cloud Map アクションに必要な権限
- AWS Cloud Map 条件キーリファレンス

AWS Cloud Map アクションに必要な権限

CreateHttpNamespace

必要な許可 (API アクション):

- `servicediscovery:CreateHttpNamespace`

リソース: *

CreatePrivateDnsNamespace

必要な許可 (API アクション):

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

リソース: *

CreatePublicDnsNamespace

必要な許可 (API アクション):

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

リソース: *

CreateService

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

リソース: *

DeleteNamespace

必要なアクセス許可 (API アクション):

- `servicediscovery:DeleteNamespace`

リソース: *, `arn:aws:servicediscovery:`*`region`*`:`*`account-id`*`:namespace/`*`namespace-id`*

DeleteService

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

リソース: *, `arn:aws:servicediscovery:`*`region`*`:`*`account-id`*`:service/`*`service-id`*

DeregisterInstance

必要な許可 (API アクション):

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

リソース: *

DiscoverInstances

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

リソース: *

GetInstance

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

リソース: *

GetInstancesHealthStatus

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

リソース: *

GetNamespace

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

リソース: *, `arn:aws:servicediscovery:`*`region`*`:`*`account-id`*`:namespace/`*`namespace-id`*

GetOperation

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

リソース: *

GetService

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

リソース: *, `arn:aws:servicediscovery:`*`region`*`:`*`account-id`*`:service/`*`service-id`*

ListInstances

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

リソース: *

ListNamespaces

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

リソース: *

ListOperations

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

リソース: *

ListServices

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

リソース: *

ListTagsForResource

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

リソース: *

RegisterInstance

必要な許可 (API アクション):

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

リソース: *

TagResource

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

リソース: *

UntagResource

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

リソース: *

UpdateHttpNamespace

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

リソース: *, arn:aws:servicediscovery:*region*:*account-id*:namespace/*namespace-id*

UpdateInstanceCustomHealthStatus

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

リソース: *

UpdatePrivateDnsNamespace

必要なアクセス許可 (API アクション):

- servicediscovery:UpdatePrivateDnsNamespace
- route53:ChangeResourceRecordSets

リソース: *, arn:aws:servicediscovery:*region*:*account-id*:namespace/*namespace-id*

UpdatePublicDnsNamespace

必要なアクセス許可 (API アクション):

- servicediscovery:UpdatePublicDnsNamespace
- route53:ChangeResourceRecordSets

リソース: *, arn:aws:servicediscovery:*region*:*account-id*:namespace/*namespace-id*

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

リソース: *, arn:aws:servicediscovery:*region*:*account-id*:service/*service-id*

AWS Cloud Map 条件キーリファレンス

AWS Cloud Map IAM `Condition` AWS Cloud Map ポリシーの要素で特定のアクションに使用できる以下の条件キーを定義しています。これらのキーを使用して、ポリシーステートメントが適用される条件をさらに絞り込むことができます。 AWS Cloud Map どのアクションがこれらの条件キーを受け入れるかについての詳細は、「AWS Cloud Mapによって定義されたアクション」を参照してください。条件キー全般の詳細については、を参照してくださいIAM ポリシーでの条件の指定。

`servicediscovery:NamespaceArn`

関連する名前空間の Amazon リソースネーム (ARN) 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: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`

関連するサービスの Amazon リソースネーム (ARN) 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: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。

でのログ記録とモニタリング 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、パフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。ただし、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?

のコンプライアンス検証 AWS Cloud Map

のセキュリティとコンプライアンス AWS Cloud Map は、Health Insurance Portability and Accountability Act (HIPAA)、Payment Card Industry Data Security Standard (PCI DSS)、ISO、FIPS など、複数の 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人によって評価されます。

特定のコンプライアンスプログラムの対象となる AWS のサービスのリストについては、「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。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。

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

AWS サービスを使用する際のお客様のコンプライアンス責任は、お客様のデータの機密性、企業のコンプライアンス目的、適用法規によって決まります。 AWS はコンプライアンスに役立つリソースを提供します。

- セキュリティおよびコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境を にデプロイするための手順を説明します AWS。
- Architecting for HIPAA Security and Compliance ホワイトペーパー – このホワイトペーパーでは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。
- AWS コンプライアンスのリソース – このワークブックとガイドのコレクションは、お客様の業界や場所に適用される場合があります。
- AWS Config – この AWS サービスは、自社プラクティス、業界ガイドライン、規制に対するリソース設定の準拠状態を評価します。
- AWS Security Hub — この AWS サービスは、 内のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準およびベストプラクティスへの準拠を確認するのに役立ちます。

の耐障害性 AWS 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、および高度の冗長ネットワークで接続されている複数の物理的に独立および隔離された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の特定のリージョンのリソースの正常性をチェックする 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。

AWS Cloud Map でのインフラストラクチャセキュリティ

マネージドサービスである AWS Cloud Map は AWS グローバルネットワークセキュリティで保護されています。AWSセキュリティサービスと AWS がインフラストラクチャを保護する方法については、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected Framework」の「インフラストラクチャ保護」を参照してください。

AWS の発行済み API コールを使用して、ネットワーク経由で AWS Cloud Map にアクセスします。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS) TLS 1.2 および TLS 1.3 をお勧めします。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートです。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

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

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

インターフェイスエンドポイントを使用して AWS Cloud Map API にアクセス (AWS PrivateLink)

AWS PrivateLink を使用して、VPC と AWS Cloud Mapの間にプライベート接続を作成できます。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内にあるかのようにAWS Cloud Mapにアクセスすることができます。VPC のインスタンスは、パブリック IP アドレスがなくても AWS Cloud Mapにアクセスできます。

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 Guide (AWS PrivateLink ガイド)」の「Access an AWS のサービス using an interface VPC endpoint (インターフェイス VPC エンドポイントを使用して にアクセスする)」を参照してください。

AWS Cloud Mapに関する考慮事項

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

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 Logs に送信する場合、CloudWatch Logs 用のインターフェース VPC 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 Logs の使用」を参照してください。

VPC エンドポイントでは、AWS クロスリージョンリクエストはサポートされていません。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。

AWS Cloud Map 用のインターフェイスエンドポイントの作成

Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) を使用して、AWS Cloud Map API のインターフェイスエンドポイントを作成できます。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の作成」を参照してください。

以下のサービス名を使用して、AWS Cloud Map API 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
```

以下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ース・エンドポイントを作成する：

```
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のSDKでは、各API操作を呼び出すと、サービス・エンドポイントの前にさまざまなホスト接頭辞が付加されるため、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。

インターフェイス・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リクエストを行うことができる。例えば、`servicediscovery.us-east-1.amazonaws.com` です。

VPCE AWS PrivateLinkの接続は、AWS Cloud Mapが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。ただし、エンドポイントを定義する前に、どのアベイラビリティーゾーンがVPCE をサポートしているか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、AWS Management Console を使用します。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[]'
```

を使用した AWS Cloud Map API コールのログ記録 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 コールをイベント 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 AWS Cloud Map API オペレーションへのコード呼び出しが含まれます。で収集された情報を使用して CloudTrail、 に対するリクエスト AWS 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、以下を判別するために役立ちます。

- 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、 が 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、検索、ダウンロード、およびイミュータブルな記録を提供します 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。イベント履歴の表示には料金はかかりません CloudTrail。

AWS アカウント 過去 90 日間のイベントを継続的に記録するには、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、 はログファイル CloudTrail 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、マルチリージョンの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、 から進行中の管理イベントのコピーを 1 つ無料で Amazon S3 バケットに配信できますが、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、AWS CloudTrail 「 の料金」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、データを高速に取得するために最適化された単票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発生します。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、「 AWS CloudTrail の料金」を参照してください。

AWS Cloud Map での 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た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、AWS CloudTrail「 の料金」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に記録できる 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 CloudTrail APIs 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。列にログ記録された Data APIs CloudTrail には、リソースタイプ CloudTrail について にログ記録された API コールが表示されます。

データイベントタイプ (コンソール)	resources.type 値	にログ記録APIs CloudTrail
AwsApiCall	`AWS::ServiceDiscovery::Namespace`	• DiscoverInstances • DiscoverInstancesRevision
AwsApiCall	`AWS::ServiceDiscovery::Service`	• DiscoverInstances • 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。オブジェクトの詳細については、「AWS CloudTrail API リファレンス」の「AdvancedFieldSelector」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 での 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 CloudTrail を記録します。

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記録します。が に記録する 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 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

イベントは任意の送信元からの単一のリクエストを表し、リクエストされた 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は表示されません。

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

AWS Cloud Map リソースのタグ付け

AWS Cloud Map リソースを管理しやすくするために、タグ形式で各リソースに独自のメタデータを割り当てることができます。このトピックでは、タグとその作成方法について説明します。

目次

タグの基本

タグとは、AWS リソースに付けるラベルです。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、 AWS リソースを目的、所有者、環境などで分類できます。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。たとえば、AWS Cloud Map サービスに一連のタグを定義して、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、AWS Cloud Map に関連する意味はなく、完全に文字列として解釈されます。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。

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

AWS Identity and Access Management (IAM) を使用している場合は、タグを作成、編集、削除する許可を持つ AWS アカウントのユーザーを制御できます。

リソースのタグ付け

新規または既存の AWS Cloud Map 名前空間とサービスにタグ付けできます。

AWS Cloud Map コンソールを使用している場合、新規リソースには作成時にタグを適用でき、既存のリソースには関連するリソースページの [Tags] (タグ) タブを使用していつでもタグを適用できます。

AWS Cloud Map API、AWS CLI、または AWS SDK を使用している場合、関連する API アクションで `tags` パラメータを使用して新規リソースにタグを適用でき、TagResource API 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、「TagResource」を参照してください。

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。

次の表では、タグ付け可能な AWS Cloud Map リソースと、作成時にタグ付け可能なリソースについて説明します。

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	Yes	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。	Yes
AWS Cloud Map のサービス	Yes	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	Yes

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)
		ソースには伝達されません。	

タグの制限

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
- 複数の AWS サービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスでも許可される文字に制限が適用されることがあるため注意ください。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 +、-、=、.、_、:、/、@。
- タグのキーと値は大文字と小文字が区別されます。
- `aws:`、`AWS:`、またはその大文字または小文字の組み合わせを、キーまたは値のプレフィックスとして使用しないでください。これらの文字列は AWS による使用のために予約されています。このプレフィックスが含まれる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、リソースあたりのタグ数の制限にはカウントされません。

CLI または API でのタグの操作

リソースのタグの追加、更新、リスト表示、および削除には、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	Add-SDResourceTag
1 つ以上のタグを削除します。	UntagResource	タグなしリソース	Remove-SDResourceTag
リソースのタグを一覧表示します	ListTagsForResource	list-tags-for-resource	Get-SDResourceTag

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	New-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	New-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	New-SDPublicDnsNamespace
[Create a service] (サービスを作成)	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、次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	いいえ	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定常率	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 アカウントレートあたりの オペレーション	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	いいえ	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 のサービスです。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では 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を確認できます。

   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。

を使用してさらにサービスクォータを操作するには、「Service Quotas ユーザーガイド 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 リクエストのスロットリング

AWS Cloud Map は、各 AWS アカウントの DiscoverInstances API リクエストをリージョンごとにスロットリングします。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 お客様に平等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 への呼び出しが許容される DiscoverInstances API リクエストの最大クォータを超えないようにします。DiscoverInstances 次のいずれかのソースから発信される API コールは、リクエストクォータの対象となります。

- 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、 DiscoverInstances API オペレーションのバケットサイズは 2,000 トークンであるため、1 秒あたり最大 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、リフィルレートは毎秒 1,000 トークンです。1 秒あたり 2,000 の DiscoverInstances API リクエストを行うと、バケットはすぐにゼロ (0) トークンに削減されます。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、必要な期間にわたって毎秒最大 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。詳細およびこのアルゴリズムの実装の例については、「AWS でのエラーの再試行とエクスポネンシャルバックオフ」を参照してください。

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。

関連情報

AWS Cloud Map を利用する際に役立つ関連リソースは以下の通りです。

トピック

- AWS リソース
- サードパーティ製ツールとライブラリ

AWS リソース

このサービスを利用する際に役立つ関連リソースは次のとおりです。

- クラスとワークショップ – AWS のスキルを磨き、実践的な経験が得るために役立つセルフペースラボに加えて、ロールベースのコースと特別コースへのリンクです。
- AWS デベロッパーセンター – チュートリアルの検索、ツールのダウンロード、AWS デベロッパーイベントの確認を行います。
- AWS デベロッパーツール - AWS アプリケーションを開発および管理するためのデベロッパーツール、SDK、IDE ツールキット、およびコマンドラインツールへのリンクです。
- ご利用開始のためのリソースセンター — AWS アカウント をセットアップする方法、AWS コミュニティに参加する方法、最初のアプリケーションを起動する方法を説明します。
- ハンズオンチュートリアル - ステップ バイ ステップのチュートリアルに従って、最初のアプリケーションを AWS で起動します。
- AWS ホワイトペーパー – アーキテクチャ、セキュリティ、エコノミクスなどのトピックについて、AWS のソリューションアーキテクトや他の技術エキスパートが記述した AWS の技術ホワイトペーパーの包括的なリストへのリンクです。
- AWS Support Center – AWS Support のケースを作成して管理するためのハブです。フォーラム、技術上のよくある質問、サービスヘルスステータス、AWS Trusted Advisor など、他の役立つリソースへのリンクも含まれています。
- AWS Support – AWS Support に関する情報のメインウェブページです。クラウド内でのアプリケーションの構築および実行を支援するために 1 対 1 での迅速な対応を行うサポートチャネルとして機能します。
- お問い合わせ - AWS の請求、アカウント、イベント、不正使用、その他の問題などに関するお問い合わせの受付窓口です。

- AWS サイトの利用規約 – 当社の著作権、商標、お客様のアカウント、ライセンス、サイトへのアクセス、その他のトピックに関する詳細情報。

サードパーティ製ツールとライブラリ

AWS リソースに加えて、次のサードパーティー製ツールとライブラリが AWS Cloud Map で稼働します。

- クラウドアプリケーションフレームワーク (AWS Cloud Map) - 一般的なクラウドプラットフォームタスクを処理するライブラリ (AWS Cloud Map を使用したメッセージのキュー、イベントの発行やクラウド機能の呼び出しなど)。
- ExternalDNS for Kubernetes - 外部の DNS サービスを設定するためのツール。Amazon Route 53、Kubernetes Ingresses and Services 用の AWS Cloud Map を含みます。

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
チュートリアルが追加されました。	使用の一般的な使用例を示す 2 つのチュートリアルが追加されました。 AWS Cloud Map	2024 年 3 月 27 日
CloudTrail 統合ドキュメントが更新されました。	API AWS Cloud Map CloudTrail アクティビティを記録するための統合について説明しているドキュメントが更新されました。	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
Cloud Map と AWS PrivateLink	これで、を使用して VPC との間にプライベート接続を作成できます。 AWS PrivateLink 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 Python 用 SDK	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 専用のネットワークで使用可能になりました。	2022 年 1 月 28 日
サービスインスタンスディスカバリー	AWS Cloud Map DNS クエリをサポートする名前空間にサービスを作成するサポートが追加されました。DNS クエリは DiscoverInstancesAPI オペレーションでのみ検出可能で、DNS クエリは使用できません。	2021 年 3 月 24 日
リソースへのタグ付け	AWS Cloud Map を使用してネームスペース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が追加されました。 AWS Management Console	2021 年 2 月 8 日
リソースへのタグ付け	AWS Cloud Map および API 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サポートが追加されました。 AWS CLI	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

AWS 用語集

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

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